A Gaming Approach to Collaborative Modelling

Master's Thesis Information Sciences

by

Ilona Wilmont

Radboud University Nijmegen
Faculty of Science
Institute of Computing and Information Sciences

March 4, 2009

Supervisors:
Dr. Stijn Hoppenbrouwers, Radboud University Nijmegen
ing. Mark Mastop, MSc., Everest B.V.

Thesis number: 91 IK

Abstract

Formal modelling is often considered a boring, complicated task. This thesis proposes a different perspective on modelling, in which the interactions between participants is central. When using this interaction-oriented perspective, the goal-driven, collaborative, challenging aspects of modelling become more prominent, and modelling then shows a strong parallel to gaming.

Whereas in gaming, collaboration happens naturally, in modelling this seems very difficult to accomplish. Modellers often struggle with their problems alone. Therefore, if a gaming approach is used to look at project development, of which modelling is a key part, the question is if collaboration will be encouraged due to the playful nature of the approach.

This thesis examines the underlying psychological nature of gaming, why it is so attractive to many people, the benefits that can be gained from using an online game to do project development, and finally proposes a concept and design for a project development game.

The game has been designed with a real industrial environment in mind.

Acknowledgments

First of all, I would like to thank dr. Stijn Hoppenbrouwers, my supervisor from the Radboud University of Nijmegen, for all the feedback, ideas and discussions during the entire course of the project.
Second, my external supervisor at Everest B.V., Mark Mastop, for making the time to read through my thesis and all the creative ideas for the eventual game.
Also, many thanks to Ingrid van Baast, Dick van Soest, Vincent Knoop Pathuis, Rick Fleuren, Stef Demeester, Rob Gense, Friso van der Meer and Vincent Jansen, who all attended my game workshops with wonderful ideas, and laid the foundation for the Aquimatization game. Moreover, for finding time to do interviews and answering any questions I had.
Finally, to Casper Harteveld, for letting me participate last-minute in his game design workshop, and to dr. Penny de Byl, for helping me out with my game design.

Ilona Wilmont
March 4, 2009

Contents

Chapter 1

Introduction

Formal modelling is often perceived as a troublesome, complicated process. However, when abstracting from methods and modelling languages and considering only the underlying behavioural interactions, it appears to be a highly intuitive, collaborative process. Interaction and modelling elements used are very basic. For people without knowledge of modelling techniques, a logical approach is to have discussions and to draw diagrams constituted mainly of lines and blocks.
Collaborative modelling can thus be characterised as a "goal-driven interactive activity that requires freedom of action and decision within clearly set boundaries" [Hopp 09].

This is a description which typically applies to games as well. It is well-known that games are very successful in engaging and entertaining users, in some cases even leading to severe addiction. A major success are *massive multiplayer online role-playing games* (MMORPGs), which offer players roles to take on. Embodying these roles, they perform certain tasks, or quests, in order to earn some type of reward. Entire virtual communities emerge, and they are attractive enough to make people continuously return to pursue the game.
When put before a very difficult assignment which can typically not be achieved individually, players tend to find each other, organise themselves and collaborate in order to complete the task. This happens in first place because players ask for it, but also because there is some reward or even just a challenge in it for them.

In modelling, experts placed before difficult tasks usually try to solve things alone and this often tends to result in erroneous or invalid models. Using the gaming mindset to engage in modelling could be a promising approach. It may lower the threshold of the modelling process, provide insight into interactive modelling, help develop support tools and last but not least may serve to add a dimension of challenge to modelling. The latter might encourage modellers to become more actively involved in the creation of the model and challenge them to optimize it as well as they can.

This thesis aims to investigate how gaming aspects can be used in business environments to achieve the aforementioned promises. Research is both theoretical and practical, in the context of a real industrial environment. Collaboration with Everest B.V. has made this possible. I have used one of their key applications, Aquima Studio (a key component of Everest's Aquima Business Engineering Suite for application specification and execution), to test the hypotheses.

1.1 Research questions

Despite modelling's basic intuitiveness, formal modelling in practice is very difficult. Business processes and structures are complex, and for a technical expert who knows nothing of the business the challenge to capture and express them correctly is huge. Knowledge of underlying business models is typically contained in the people working in a particular area, and the problem is that they know nothing of modelling methods, have a completely different point of view and focus, and often do not speak the same language as technical experts.

The main question guiding this research is the following:

> *How can gaming aspects be used in business environments to provide increased efficiency of modelling, better quality of models, and a motivating challenge for modellers?*

In order to provide an answer to the main question, several subquestions have been formulated.

1. What aspects of gaming are essential for a successful and motivating game?
2. What HCI aspects have to be taken into consideration to make modelling more accessible?
3. Where in Aquima Studio is the implementation of gaming aspects most desirable?
4. How may such an implementation be realised?

My project can be divided in two parts. First of all I investigated why it is that in games collaborative activities such as discussion, seeking help and taking on a challenge together happen naturally. I tried to find which factors bring about the enormously attractive power games have for engaging players, so these aspects could be applied in a possible business game. From the theoretical results I have drawn up a list of essential aspects which make up a good game. I have also researched how serious games are set up, and made recommendations for evaluating modelling game playability.
Secondly, the concepts had to be tested in practice. If a modelling environment were to be much more like a gaming environment, would this encourage users to model and collaborate no matter what background they have? The results of my theoretical research have been used to conduct practical game design workshops with Everest employees, establishing their views on how a modelling game might take shape and how they would feel more motivated by using game aspects. The results have been processed into a conceptual game design, which makes use of Aquima Studio and focuses on the entire work process, rather than being specifically applied to Aquima Studio. The final results are therefore the

design document and first graphical impressions.
Only as the game gets used in practice, can it be tested to prove its true value.

1.2 Company background

1.2.1 General information

Everest is a Dutch software company with its main office located in ’s Hertogenbosch. The company was founded in 1995, and has been growing at a very fast rate, having up to around 170 employees at the time of writing. They have a few international offices as well, the largest one being in Madrid.
Everest has its roots in artificial intelligence and knowledge technology.

The Everest method for software development is unique within the world of software engineering. Their philosophy assumes that a good model of an application contains all the information necessary to build the application. Human code writing, then, is merely an additional point of interpretation where more errors can be introduced. Consequently, they directly interpret the model itself and automatically generate and execute the application from there. This completely skips the step of (automated) code generation.
Due to this method of development, the focus shifts from code generation and testing to model validation. As a result, the solutions Everest delivers are highly flexible and can be delivered, according to Everest, up to 3 times faster and at lower costs than by using the traditional code writing way.
Everest’s target branch is the financial sector. This is a field continuously in motion. New products appear on the market, old products change, laws are adapted. The Everest methodology is therefore perfectly suited for banks, insurance companies and governmental organisations.
Recently, market analyst company Gartner has named Everest Cool Vendor in Business Process Management. A Cool Vendor is a company employing an innovative software solution allowing organisations to do things not previously possible.

1.2.2 Aquima Business Engineering Suite

The Aquima Business Engineering Suite is specifically designed to create dynamic business solutions which can offer customised services in a flexible way. The basic principle underlying Aquima is a separation of the business logic (knowledge en policy) and the program code (control mechanism). In this way, business logic can be easily adjusted while the program and the running solution stay up. Changes can thus be processed quickly and smoothly and can immediately be made available to users.

Aquima consists of several components which cover all aspects of the Business Process Management Cycle [Ever 09]:

- **Aquima Interactions**
  The component controling intelligent user interaction covering several channels.

- **Aquima Processes**
  Delivers functionality to realise processes aimed at front- and mid-office.
- **Aquima Documents**
  The solution to model and perform complex document models and output processes.
- **Aquima Services**
  Ensures that modelled processes are made available as service for use within external systems.
- **Aquima Foundation**
  Based on inference mechanisms and extensive domain models, this is the core of the Aquima suite.
- **Aquima Studio**
  This is the component that allows for specification of applications through models.

Aquima Studio

The component I will be focusing on is Aquima Studio. In this component, the actual modelling of the application takes place, so this is also where the implementation of gaming techniques for collaborative modelling is most relevant.
Aquima Studio represents business knowledge in a structured, visually appealing way, and does not require knowledge of programming in order to perform maintenance operations on the business knowledge. It can be used as a simulation and testing environment before applications are launched. Flexibility is achieved through localisation, tailored authorisation and context-sensitive support.

The main reason for introducing gaming aspects in Aquima Studio is because modelling is often perceived as boring. Using an approach that will change this state of mind into a more motivated one can be extremely beneficial both for the quality of models and the productivity of the participants.

1.3 Relevance

Lots of research has been done on improving the quality of models, but this research has mainly focused on the methods and languages employed to create models, resulting in the births of many modelling techniques, the one only slightly different from the other. This is often referred to as the 'YAMA-syndrome' (Yet Another Modelling Approach). Few research projects have yet systematically investigated the interaction-based, human aspects of modelling, or taken the process itself into consideration [Hopp 08a, Hopp 09]. This is exactly where metaphors using appealing concepts such as gaming can be of great help in clarifying and structuring things.
For instance, a typical example is someone playing an online role-playing game and being confronted with a task too difficult to accomplish on his own. The player is very much aware of this, and will immediately seek out other competent players who can help him accomplish the task. This is an action that happens

naturally and requires no specific instruction. People from very different backgrounds and with completely different interests come together, collaborate, and achieve success.

In modelling activities, such things typically do not happen. People are instructed to do a task, and will try everything within their power to accomplish it. Collaboration usually does not go much further than discussing some difficult issues with colleagues and asking the business people for feedback. But this is not true collaboration, since the people being questioned are not truly involved in the modelling task and always seem to be separated by their differences in background and interests. Most importantly, the interaction is temporary, typically being high at the start of a project, reducing to practically none after the initial phase [D].

Consequently, finished models and resulting applications do not exactly meet the customer's expectations. It may also happen that models are not constructed at all. Invalid models are a consequence of improper communication and understanding between technical and business experts. Failure to construct models happens when technical experts are too few or have more important things to do. Business people too refuse to do the task because it is not their job. Since models are the foundation of applications and structured business processes, it is essential that a solution to these problems is found.

One of the key aspects of finding a good solution is to focus on the model development process instead of the modelling language. First of all, collaboration has to become so obvious that there is no way around it. A game interface with built-in avatars and chatbox seems a promising solution. With collaboration functioning effectively, modelling is likely to become more accessible to both technical and business experts, and last but not least more motivating because it will become a collaborative, in-game activity.
Adequate tooling in the form of an all-encompassing *business process progress-monitoring and development game* has to support the process. If properly designed, the game can serve as the virtual meeting point and working place for the whole team.

As suggested above, project development and gaming have much in common. Both are goal-driven, interactive activities requiring freedom of action and decision [Hopp 09], be it within clearly defined boundaries. Even, each phase of project development can be described in terms of different game genres [Wode 06], [5.1]. This is why the gaming approach applied to project development, of which modelling is a key part, can result in a process which emphasizes the creative, innovative aspects of modelling and enhances it by adding a strong motivational factor. Also, by defining a clear set of modelling game rules, the restrictions of modelling become much more prominent and understandable, and hopefully more easy to apply.
Moreover, creating a modelling process that appeals to a user can make modelling accessible not only to technical experts, but also to regular business users to whom it is of great importance that the final product conforms to their interests and demands.

Another important aspect of modelling which can be facilitated by a gaming metaphor is collaboration. Currently, modellers and business experts collaborate far too little on complex modelling tasks, as Everest business engineers have pointed out during an interview. When collaboration does take place, the problem is that technical modellers and business experts often cannot express and communicate their points of view clearly, inevitably resulting in misinterpretations and faulty models. Add to that the fact that modellers are few and not uncommonly struggle alone on the complex tasks without seeking timely help. All these things make that modelling is perceived as a boring, time-consuming process without any essential added value.

Within Everest, these problems are recognised and confirmed. By applying gaming concepts directly to the modelling tool and engaging both modellers and business experts in a gaming mindset, there is a promising prospect that participants in a modelling process will be challenged to face a common focus; the model, and be more motivated to accomplish the task by means of true collaboration. Hopefully, combining the knowledge of many by using a gaming approach will result in model quality improvement.

In summary, a short overview of the main benefits of the gaming approach for modelling [Hopp 09]:

- Motivating modellers
- Designing improved interactive tools

There are some ambitious goals hoping to be achieved by a modelling game [Hopp 09]:

- Lowering the threshold of the modelling process
- Improving model quality

By defining clear rules, sharing knowledge and systematically structuring specific concerns, the business process progress-monitoring and development game proposed in this thesis may help achieve these goals.

1.4 Related fields of research

The gaming approach provides a clarifying and structured view on the design of an operational interaction system. Such a view calls for an interdisciplinary approach, involving areas of science such as Method Engineering, Human-Computer Interaction and Collaboration Engineering.
First of all, I will explain the relation between Method Engineering and gaming. For this we first need a clear definition of terminology.

Method Engineering (ME):
A research field investigating the construction of information systems (IS) development methods and tools. It aims to design, construct and adapt models, techniques and tools for the development of ISs [Brin 96].

Situational Methods (SM):
This involves the configuration of IS development methods explicitly adapted to the situation of the currently active project [Brin 96]. Engineering a situational method requires standardised building blocks and guidelines (the so-called *meta-methods*) to assemble those building blocks.

Situational Method Engineering (SME):
This is a field that aims at the construction of methods for project-specific contexts of use [Mirb 06], possibly using method fragments from existing methods [Brin 99]. It covers both the language and process aspects of methods. SME works towards tool support and operationalisation [Hopp 08a].

While both ME and my current work aim to construct models for IS development, tailored to suit the needs of the corresponding project [Mirb 06, Brin 99], there is a difference in focus. Traditional ME focuses mostly on conceptual aspects of modelling languages and requirements of methods [Brin 96], whereas I will approach the whole activity of modelling as an interaction system based upon operational modelling methods. Such a process-oriented view, considering basic interactions, will provide steps in the direction of deeper understanding, support and adequate tooling [Hopp 09]. When taking methods and their tool support, and combining them with participants engaged in the modelling process, an operational interaction system results [Dijk 08]. This system can be approached from a Human Computer Interaction (HCI) point of view, offering possibilities to solve the problems with reference to the same variables and emphasis points as usability engineering.
The importance of this change is emphasised by the fact that increasingly, formal models are being used as input for business IT-applications. Often, just as is the case with Aquima Studio, these models are interpreted and software is generated from this interpretation. This means that interaction with the model directly affects the generated application. Therefore, when modelling, human factors concerning the final product have to be kept in mind from the very start. Another consequence is that the whole operational system bringing forth or manipulating the models will have to be taken into account. Factors such as usability, user acceptance, motivation and efficiency will play important roles when designing such systems.
The game metaphor serves to provide a different way of framing a modelling tool. Tools should become more process oriented and collaborative, and designing a system according to the principles of game design will facilitate exactly that.
For example, a very important element will be a clearly defined rule set. Rules constrain the participant's behaviour and guide the process, making the achievement of the main goal more inefficient than it might seem. In gaming, this is part of the challenge [Sale 04]. A parallel can be drawn with modelling. In natural language, anything one wants to say can just be expressed verbally or in writing. As soon as it is formalised, the expression is constrained by the rules of the method and may not appear to be so straightforward to express anymore. Certain preconditions will have to be reckoned with. So in modelling, part of challenge is to create the best possible model within the limits the rules present.
Another field of knowledge showing parallels to this research is **Collaboration Engineering**. This is defined as *the development of repeatable collaborative*

processes that are conducted by practitioners themselves [Brig 03]. They state that under circumstances, people using group support systems have proven to be more productive than people who do not make use of such systems.
Moreover, in traditional modelling processes, in-depth stakeholder involvement happens through a facilitator. This is both costly and inefficient, as it produces an extra interpretation point, which is prone to error introduction. In Collaboration Engineering, an important goal is to automate the facilitator [Hopp 09].

1.5 To conclude...

In the next chapter I will explore what makes a game motivating and how these factors are reflected in the design of games as applied in the gaming industy.
The third chapter elaborates on the relation between modelling, gaming and Human Computer Interaction, and which factors are of importance in operational modelling games. I describe an interaction-oriented perspective on modelling, and examine how modelling and games can be assessed from an HCI point of view, something which so far has never gotten much attention [Hopp 09]. I will also pay attention to collaboration and how to assess it.
Chapter 4 describes the Everest situation, and provide ideas for how gaming can be used to turn Aquima Studio into a playable modelling game environment. I will investigate how essential gaming aspects can be tailored to suit the needs of Aquima Studio.
The fifth chapter translates these ideas into a more concrete concept for game design in Aquima, based on ideas conjured up during design workshop sessions. I describe a game setup which incorporates the entire development process, as proposed during the game design workshops, to provide not only motivation for modellers, but also more insight for project leaders.
The sixth and final chapter concludes my thesis and envisions the future of gaming development for business applications.

Chapter 2

Gaming: motivation and design

People love to play. Human beings engage in play even before they can walk and talk. Just think of babies finding hours of entertainment just messing around with coloured blocks. While this type of play is too much unstructured to be considered a game, it does show that play is like a second nature to us. And the popularity of play stays as we grow older, the games becoming more formal along the way.

This development is reflected very much in the computer game industry. Players voluntarily invest huge amounts of time, money and energy in games. So much, in fact, that video gaming is the fastest growing form of human entertainment, and annual revenues from this industry have surpassed those of Hollywood [Yi 04]. This makes them the largest entertainment medium in the world.
The game experience is something that greatly appeals to people. Interactive, representational, social and cultural aspects contribute to experience. Consider early games such as Pong. They were revolutionary, because people were transformed from one-way interactivity viewers into players. Unlike watching television, they were enabled to play with the media object. The popularity of Pong remains unscathed.

So why are games so extremely popular? In this chapter, I will try and create an understanding of gaming: why they are so successful in engaging people for hours on end, and what aspects are necessary to create such a game. I will explore the available literature on motivation for gaming, game design approaches, and finally distill all the common elements into a list of essential game design concepts.

2.1 Understanding games

> ”(Play) is a significant meaningful function - that is to say, there is some sense to it. In play there is something ”at play” which transcends the immediate needs of life and imparts meaning to the action. All play means something.” **Johann Huizinga, *Homo Ludens***

2.1.1 Meaningful play

Play in games can take a huge variety of forms, from a physical duel between two soccer teams to an intellectual chess match to a multiplayer online computer game. Each of these situations places play in the context of a game. Play itself, however, does not just come from the game, but from the interaction between the players, the context and the game system. During the course of play, players continuously make trade-offs, choices and decisions. Understanding this interaction helps to understand what goes on when a game is played [Sale 04]. Each action has an outcome which changes the overall game state, creating a new meaning every time a player's decision takes effect.

Therefore it can be said that play is not simply about unstructured, purposeless interaction between individuals or groups. Quite the contrary: play is meaningful interaction with a clear goal to be achieved. From this notion the concept of *meaningful play* arises.

Meaningful play, as defined by [Sale 04], can be viewed in two ways. Firstly, they refer to the way game actions result in game outcomes to create meaning. This view basically describes what happens in a game. The relationship between player action and system outcome is essential: it is the process of the player taking an action within the defined game system and the system responding to that action. The meaning of the action resides in this relationship, which is unique for every game.

Whereas every game should be based on meaningful play, there are some games which create more meaningful play than others. Such games are typically the most successful ones. This does imply that a more selective, evaluative criterion, determining *when* meaningful play takes place, is necessary. The relationships between actions and outcomes in a game have to be both *discernable* and *integrated* into the larger gaming context. The creation of meaningful play is then the ultimate goal of successful game design.

Meaningful in this sense is mostly about emotional and psychological player experiences in a well-designed system of play. Discernable means that the result of the game action is communicated to the player in a perceivable way. This allows the player to realise what has happened and to decide how to continue playing. Without this property, a game would be no more than random button pressing. Discernability forms the building blocks of meaningful play.

For another, the relationship between action and outcome requires integration into the whole of the gaming context. Player actions must have both immediate visible effects and effects on the long term. For example, killing a monster gets the player out of immediate danger, but killing all the monsters in a level must get the player to the next level. If there is no next level, what is the point of killing all the monsters?

Meaningful play constitutes the core goal of successful game design. It can occur on different layers, from a formal, mathematical move in Chess to a social level where several teams battle for victory, across different cultures and even for different game purposes. Understanding meaningful play is the first step in understanding what a game is.

2.1.2 Defining games

There are two ways in which the relation between game and play can be seen. The first is a typological approach, defining the relationship according to the forms play and games take in the world. For instance, children playing on a jungle gym is not organised enough to be called a game, but definitely is a form of play. The more formal and organised forms of play constitute games. Games are thus a subset of play.

The second approach is conceptual. Within the field of game design, play is only a component of a game, namely the interaction part. Besides play, there is a whole lot more that makes up the entire game. In this case, play is a subset of the game.

Salen and Zimmerman [Sale 04] compare different definitions of 'a game' before taking the most important elements and coming to their own definition. All definitions save one agree that a game is based on a predefined rule set, but apart from that the emphasised elements differed widely. Their definition is as follows:

> *"A game is a system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome".*

However, there are some elements appearing in the other definitions which are of importance for this thesis as well. Therefore, for the purpose of this thesis, I would like to propose a slight modification of the definition given above:

> *A game is a system in which players* ***voluntarily*** *engage in a* ***goal-oriented****, artificial conflict, that results in a quantifiable outcome.* ***The activity takes the form of a process which is*** *defined by rules,* ***yet offers freedom of action****.*

The main ideas behind this definition are the following:

System

A system consists of four basic elements: objects, attributes, internal relationships between objects and the environment surrounding the system. Any game can be framed as a system. The content of the elements depends on the type of game.

Voluntary

People engage in games on a voluntary basis. Often, people have to pay in order to facilitate game play, or even face disapproval from parents or friends. This implies a strong intrinsic motivational factor, which I will explore in more detail in the next section.

Goal-oriented

Games have a clear, pre-defined goal which players constantly work towards. Each action brings them a tiny step closer to their main goal. Without goals, there can be no meaning to play.

Artificial

A game has its own boundaries in time and space, specifically created by the game designers. Therefore it is an artificially facilitated form of interaction.

Conflict

In any game, conflict arises naturally from the interaction in a game [Craw 84]. The goal the player is trying to achieve is blocked by obstacles. Passive or static obstacles present a puzzle, active and dynamic obstacles a true game challenge. Active, responsive obstacles always are embodied by an intelligent agent, and conflict between a player and such an agent is inevitable.
Whereas games emphasising cooperative efforts rather than conflicts have been designed, they were never successful commercially. However, as Crawford [Craw 84] states, *"conflict can only be avoided by eliminating the active response to the player's actions. Without active response, there can be no interaction"*.
Therefore the focus in modelling games should be on shifting the conflict so cooperative efforts can be included. Teams can be formed, whose members cooperate with each other in a team conflict (which the creation of the model should present) with another team. The challenge is then to achieve success in this collaborative puzzle. This way, the decision-making character of modelling can be expressed, and the challenge can be guaranteed. Players must at least have the illusion of purposeful reaction to their actions, otherwise the game collapses.

Quantifiable outcome

At the end of the game, the player has either won, lost, or at least has received some kind of measurable score. The quantifiable outcome is what often distinguishes games from less formally organised forms of play.

Rules

Rules provide the structure out of which play emerges by constraining what the player can and cannot do. They are a crucial part of games.

Freedom of action

Even though the player's actions are delimited, a certain degree of freedom is still necessary in order to allow the player to develop feelings of autonomy and competence. These basic psychological needs, among others, are necessary to motivate the player to keep playing the game [Ryan 06]. In the next section, I will elaborate further on this.

2.2 The power of games

An ideal game is an absorbing, captivating, imaginative world in which people can lose themselves completely. Players may create a virtual character, or *avatar*, can make their own decisions and exercise control and influence. Through this autonomous nature of their character, they ultimately become a part of the game world, like a second reality. A comparison can be made with a book so captivating it cannot be put down until all the surprising details have been revealed.

But there is more to a game. Gaming also has a strong social aspect, especially the immensely popular *Massive Multiplayer Online Role Playing Games* (MMORPGs). It is an activity which is experienced together with many players, each in the form of a fantastic, other-worldly character. Such a game cannot be finished and thus also not be put down. There is lots of interaction between players, and collaboration is a natural strategy to achieve one's goals.

But exactly what is it that makes games so captivating? Why are people seemingly indefinitely motivated to play, even to the extend of becoming severely addicted to games? In this part I will explore research on motivation for games and provide an overview of the most important factors.

2.2.1 Key aspects for a motivating game

People engage in gaming voluntarily. No physical rewards are given for playing; in fact, people often have to pay in order for play to be possible. So what is it that motivates people to play a game?

Computer gaming is fast becoming the most widespread form of human entertainment. Participation in gaming is already common across different demographic groups, involving players of all ages. With this increasing participation come the arguments of the negative side effects of gaming, such as excessive violence, lower psychological and physiological well-being, lower productivity and achievement and impaired personal relationships. However, a new school of thought is flourishing, promoting the benefits of gaming for improved learning [John 06, Gee 03] and that a certain degree of heightened psychological well-being can be achieved, if only on the short term [Ryan 06].

Considering the variety and complexity of video games and the enormous differences in user behaviour which can be observed, it is clear that games can be both beneficial and harmful to players as far as psychological well-being is concerned.

Regardless of the debate, games have a huge appeal on players, who in turn are very much motivated to devote a lot of time and energy to the activity of game playing. But in order for players to do any such thing voluntarily, they must be able to reap some type of satisfaction out of gaming, otherwise interest will quickly be lost.

Previous research has tried to create general typologies for player classification [Bart 96, Yee 06]. These typologies are based on the main focus of players, such as *achievement* (seeking mastery and power), *socialisation* (seeking interaction and in-game relationships) and *immersion* (seeking to escape reality and become part of the game story) [Yee 06]. However, these structures mostly consider player behaviour and reflect the structure and content of current games, rather than the fundamental or underlying motives and satisfactions that can keep players engaged regardless of the game type or individual preferences [Rigb 04]. Recent research by Ryan et al. [Ryan 06], on the other hand, aims to address the general factors associated with game enjoyment and sustained motivation across players and game genres. They look at how games differing in controllability, structure and content still appeal to basic human motivational tendencies and psychological needs. For another, they investigate what impact games have on short-term psychological well-being. The latter is presumed to be a function of the basic psychological needs games satisfy.

Game appeal

Players seek to satisfy basic psychological needs in the context of play. Intrinsic motivation is regarded to be the core motivation for play, since effort and money are required before play can actually be started. This implies that people seek only fun and entertainment, as a distraction from their daily activities.

One of the most important aspects of a good game is the *experience* a player goes through [Remo 08]. A certain *freedom of action* is required, which must always be meaningful. For example, standing outside a vehicle with no weapons, being unrestricted in action yet doomed to die in a matter of seconds, is useless freedom. The player must have something to do at all times.

According to a representative from Ubisoft, who lectured at the Ugame-Ulearn symposium in Delft [A], gaming should typically be *accessible* to all: easy to play, difficult to master. A further aspects he stressed was that games need *narrative* qualities, as they are a modern way of story-telling. Intense *emotions* are experienced by letting the player actively participate in the story, set in an inspiring world.

Gaming presents a combination of fun, learning, intuitive behaviour and social activity. It reflects true human nature: there is an environment and there are basic rules, the adventures are directed by the player's own actions. Several important factors of a good game are freedom of action, usable control, display and environment. The player has to have the freedom to perform missions in his own way, and actions should be actively experienced.

However, in primitive games, the true essence of a game's attractive power is the combination of *challenge and reward* [A]. Everyone likes to be challenged and then get something in return. A game is merely a platform which facilitates this desire. Learning in a game happens indirectly by means of active participation, not by reading. It is a Pavlov principle: something is fun to do and this stimulates one to improve one's skills all the time.

The gaming industry greatly emphasises *experience*. However, this is a very broad, widely interpretable term. In order to be more specific about experience, Ryan et al. [Ryan 06] use *Self-Determination Theory* (SDT)[Ryan 00b, Deci 85], a theory of human motivation, to account for player motivation in a gaming context. The theory addresses factors that both stimulate and inhibit motivation, considering intrinsic as well as extrinsic motivation.

A sub-theory of SDT, *Cognitive Evaluation Theory* (CET) [Deci 80, Deci 85, Ryan 00a], specifically concerns "contextual factors that support or thwart intrinsic motivation". Within CET, several basic psychological needs are discussed. The first two are *autonomy* (player's free will to act) and *competence* (optimal challenge for player's abilities). CET states that a sense of autonomy and competence support a player's intrinsic motivation, which translates to enjoyment and motivation for the game. If autonomy and competence are undermined, intrinsic motivation is undermined as a consequence.

Firstly, *autonomy* is typically high when an activity is done out of personal interest. Therefore the score for this need is always expected to be high because a game is voluntarily chosen. When a player feels he is being controlled in his actions and choices, autonomy diminishes. For another, autonomy can vary depending on personal appeal, design and game content. Autonomy is enhanced by flexibility in game mechanics: movements and strategies have to be free choices, and any rewards should provide feedback rather than controlling the player's behaviour [Ryan 06]. The game industry reflects this by programming the game to respond dynamically to player's choices without constraining or anticipating them first.
Secondly, the experience of *competence* can be greatly enhanced by acquiring new skills, getting positive feedback and feeling optimally challenged, which in turn will stimulate intrinsic motivation. Following from this, game controls should be intuitive and tasks should provide sufficient challenge. Perceived competence is of vital importance because it makes people feel capable of accomplishments and in control.
The third important factor is *presence*, the sense of being involved in a game. A player has to have an illusion of 'non-mediation', that is, the content of a medium should be perceived and responded to as though the medium were not there. Presence is experienced when games allow players to focus purely on game play rather than game mechanics. Moreover, presence directly depends on how well game play itself satisfies one's psychological needs (autonomy and competence) [Ryan 06]. Again, the importance of intuitive controls can be stressed here, because they may never interfere with experience of game playing itself.

Intuitive controls can contribute to motivation because they are associated with greater sense of freedom and control and enhance sense of competence [Ryan 06]. However, on their own they are not enough to motivate players. Intuitive controls have been found to contribute to motivation only in combination with autonomy and competence. Therefore many motivational models of game experience distinguish between *game mechanics* and *game play* [Rigb 04], because the two are independent. Accessible game mechanics alone are not enough to motivate players or provide enjoyment.

Salen and Zimmerman [Sale 04] also point out that part of a game's challenge is its inefficiency: the rules of play make achieving the main goal inefficient. The most straightforward way to your goal may not be possible because the rules constrain your behaviour. But this presents the very challenge of play. The challenge is to achieve your goal as efficiently as possible within the constraining rules prohibiting efficiency. However, the delicate balance between guidance and freedom of action may never be violated. The player must always have the feeling of being in control and of having full understanding of the world around him. He must never be forced to act. The game has to provide hints and nudges in the right direction, but it is up to the player to discover and exploit them. Also, players should not be directly punished if a clue is overlooked. Not being able to do anything else will only frustrate the player [Remo 08]. Apart from personal achievements, 'happy coincidences' (such as a gamble paying off) also help to make a game fun [Shef 08]. A sense of exhilaration typically follows such experiences.

Impact on short-term psychological well-being

A second sub-theory of SDT is *Basic Psychological Need Theory* (BPN). Deci & Ryan [Deci 00, Ryan 95] state that "the impact of any activity on well-being is a function of the person's experience of need satisfaction". Furthermore, the following three factors are thought to underlie psychological wellness: autonomy, competence and *relatedness*. The latter refers to the feeling of being connected with others, which enhances motivation and well-being. The psychological pull of games can be said to be due to the player's ability to develop feelings of autonomy, competence and relatedness, both to the extend of motivating further play and enhancing short-term psychological wellness (subjective vitality, self-esteem, positive affect).

As far as competence is concerned, players with a higher level of competence show a higher state of self-esteem, positive mood and vitality both before and after playing. High scores on autonomy, competence and intuitive controls appeared to promote further play of the same game, and lead to better psychological state of mind before and after. Moreover, high autonomy also implies high self-esteem, more positive affect and players see the value of the game.

To summarize these findings, Ryan et al.[Ryan 06] state that feelings of motivation for a game are caused by the satisfaction of psychological needs and enhancement of feelings of well-being a player experiences from playing the game.

A variable such as *relatedness* is found in particular within so-called Massive Multiplayer Online games, or MMOs, because of its collaborative nature. But it can also be explained in the context of single-player games. The game character typically represents a kind of supercharacter with characteristics and competences the player can only dream of having, therefore the player can desire to feel a strong relatedness with this character. The focus, however, will remain on MMOs.

MMOs also allow for the meaningful application of Yee's [Yee 06] measures for player motivation (social, immersion and achievement). The first two have already been covered and will therefore not be further considered, but the latter adds an extra dimension.

In MMOs relatedness is satisfied because of rich content and the possibility of interaction with other players. This promotes a strong sense of presence, enjoyment and motivation for further play. Yee [Yee 04] has even found game loyalty to be largely a side effect of loyalty to social networks in a community.

However, when players are focused on achievement [Yee 06] their post-play mood turned out to be more negative, suggesting that the competitive tendency induces some pressure and stress which is not truly appreciated. Focus on achievement does, however, lead to an increase in hours per week of play, suggesting that players are involuntarily triggered to score as high as possible on this variable.

However, both competence and relatedness also lead to more play hours per

week.
Finally, even though game exposure can be somewhat tiring or draining to a player, need satisfaction makes up for this. People who experience autonomy and competence during play show more positive attitudes after play, explaining why games can be pleasure and maybe even a mental restoration to certain people.

Another factor which may impact psychological well-being is *adaptability*, a strategy every effective game should incorporate. Adaptability means that the ”game adapts to each player’s skills and abilities through highly advanced artificial intelligence programs that sense just how a player is doing” [Grei 07]. The game should change slightly whenever the player is in danger of losing interest in mastery due to mismatches between difficulty expectations and what he is actually getting. Games which adapt exactly to an individual’s ability level can be a very powerful attraction indeed.

2.3 On game design

What makes a good game? How can players be motivated to keep investing their voluntary time in a specific game?
Game designers are experts when it comes to keeping people motivated. Many have written extensively on the topic, and even though there are no universally agreed upon truths about what makes a good game, there are general principles that apply for many a successful game.
Scientific literature on this topic is scant, tending to focus more on the analysis of game elements and rule structures than practical design principles themselves. Therefore I will firstly explain several approaches to successful game design according to the industry, and afterwards describe how formal game analysis can be conducted.

2.3.1 Emotions

In the game industry, succesful game design is largely centered around the core concepts of *emotion* and *motivation*. It is widely accepted among emotion theorists that emotions are phasic: first, there is recognition of an agent, event, or object as significant, which produces plans to cope with the situation. In the next phase, these plans lead to a so-called action readiness, followed by the bodily and expressive effects of emotions, such as facial expressions and actions [Oatl 96]. Emotion theorists have produced competing categorisations of emotion types, e.g., basic emotions and their subcategories, but it is generally accepted that certain emotions have tendencies to lead to a similar kind of action readinesses. This observation is especially important when creating contexts to induce emotions.
Stéphane Bura, in his article on ’Emotion Engineering’ [Bura 08], describes game design as ”experience crafting for the purpose of emotion engineering”. Players play mainly to feel emotions, which makes game design intrinsically hard because its output is an interactive system which requires two steps between the creators and the final users. Firstly, the game designer has to produce rules for interaction. Secondly, the player interacts according to the rules, gen-

erating a game state that induces emotions in the player.
Bura provides the following representation for this:

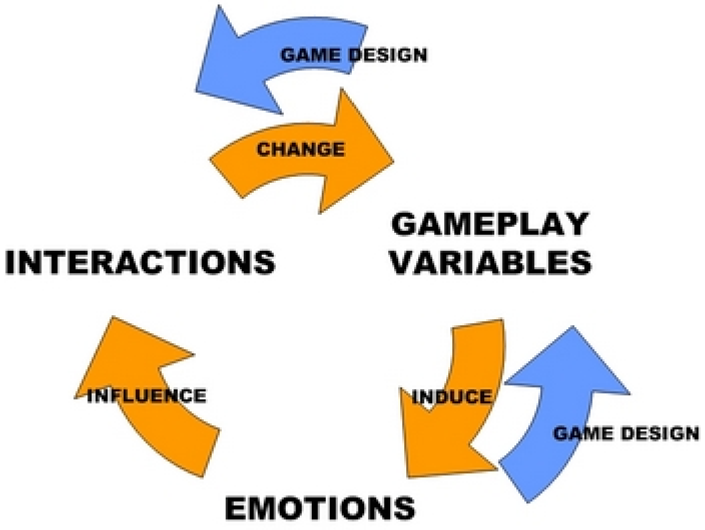


Figure 2.1: How emotions influence game play

Interactions between the player and the game cause changes in the gameplay variables. These in turn affect player emotions. For instance, a bar representing the state of health which is completely full will make the player feel more confident. This influences his interaction with the game. In this case, confidence might result in more risk taking.
Game design can guide emotions by crafting sequences of events in such a way that they elicit the targeted emotion. Emotions can also come forth out of the moment-to-moment interaction with the game. It must be mentioned that players are unique and therefore it can never be truly predicted what emotion a certain game state will elicit. However, knowledge of psychology and cognition enable the creation of the right context for expressing a certain emotion.
Game design itself tries to predict player emotions from changes in the interactive system. Establishing whether this works requires extensive testing, due to sparse knowledge and player individuality.

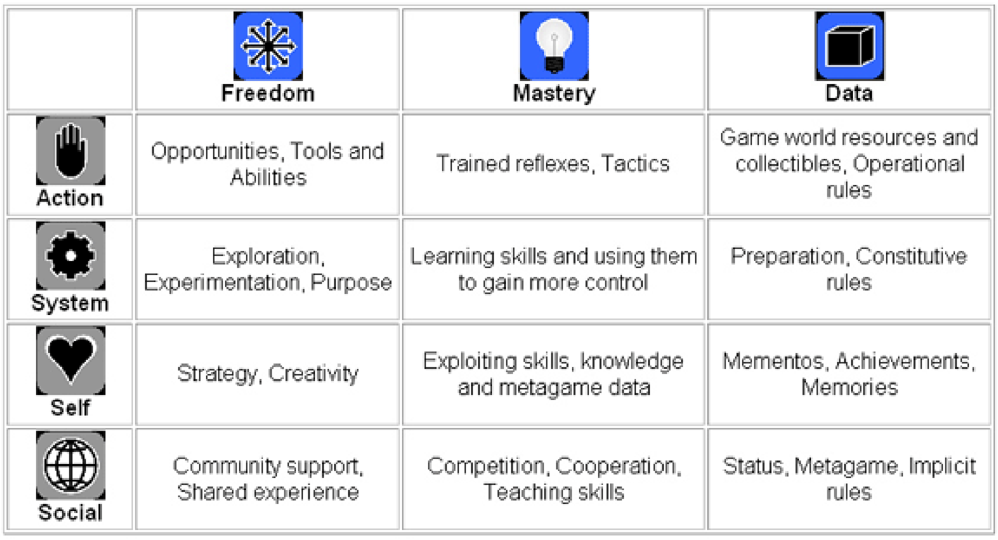
In order to understand game appeal according to the variables important for game design, Bura has organised them into comprehensible categories, which I will proceed to describe.
He distinguishes between 4 different perspectives of looking at player action and explains them in terms of 3 measurable game variables.

Player Action Perspectives
Action is the level of the body, the visceral, immediacy and short feedback loops.
System is the level of the mind, the cognitive, logic and direct action plans.
Self is the level of the soul, reflexive thoughts, desirable goals, private experiences and inner changes.

Social is the level of the community, shared experiences, rituals, culture and relationships.

Measurable Game Variables

Freedom deals with measuring choices and opportunities for choices: how the player is empowered or hindered in choice making during the course of the game.
Mastery deals with measuring skills: what empowers or hinders the player in skill acquisition and skill use, both at the immediate and cognitive level.
Data deals with measuring content, information, rules and real-life objects.



	Freedom	Mastery	Data
Action	Opportunities, Tools and Abilities	Trained reflexes, Tactics	Game world resources and collectibles, Operational rules
System	Exploration, Experimentation, Purpose	Learning skills and using them to gain more control	Preparation, Constitutive rules
Self	Strategy, Creativity	Exploiting skills, knowledge and metagame data	Mementos, Achievements, Memories
Social	Community support, Shared experience	Competition, Cooperation, Teaching skills	Status, Metagame, Implicit rules

Figure 2.2: Organisation of variables

Each game variable can be described from the perspective of each action level, together making up important concepts to keep in mind during game design. A more elaborate explanation of each variable-level description can be found in [Bura 08].

Bura illustrates how changes following from the game's actions [B.1] and from the player's actions take place [B.2]. A change can either be temporary, requiring little effort to cancel, or persistent, meaning it affects the variable value on the longer term.

Emotions can be linked to the above mentioned game variables. In this way, the right context for a certain emotion can be more precisely described. The game variable value can vary between low and high, and depending on the opportunities available to the player at a certain point in time the value can decrease, stay the same, or increase.
An example is the balance between a challenge and the appropriate reward. Such a reward acknowledges the player's skills and successes (high Mastery at the Self level), which causes the player to feel a sense of pride. However, if he is offered an easy way out and above all rewarded disproportionally, a sense of dissatisfaction will take over due to the ill-gotten gains, thereby decreasing the

feeling of Mastery at the Self level.
Being able to reason about such situations does not ensure that the player indeed feels the intended emotion, but knowledge of favourable contexts allows the game designer to provide the player with appropriate feedback and cues without causing the emotional feelings to skew.
An overview of emotions associated with changes in gameplay variables can be found in [B.3]

Using the variable organisation and the values provided in the tables in [B], the process of 'emotion engineering' can be described being able to "move backwards in the 'interactions - game variables - emotions' loop and design game systems that can induce chosen emotions" [Bura 08].
Firstly, the emotions table has to be consulted. The designer can pick the desired emotions and avoid all instances of the emotions he does not want to induce, keeping in mind all the points in the game where they could occur. Then he must find out what game variables are involved and any possible required value ranges and variations.
Next, in the game and player tables can be found which appropriate game systems can create favourable contexts to achieve these values and variations, either from the game's actions or by giving the player opportunities to make these changes.
Finally, the designer instantiates the chosen systems within the context of the desired emotion. This last part depends entirely on the designer's personal experience, goals and constraints.

An example of how to apply this method to a complex emotional state can be given for the sensation of being hunted, taken from [Bura 08]. Each of the instantiations of game systems shown in italics represents one of the possible choices a designer could make.

Being Hunted

$$BeingHunted = Dread + Unpredictability + Caution + AnticipationOfFailure - Hope$$

Dread (Persistent Low Freedom at the Action level)

- Linear path
  - Limited number of significant path choices
  - Limited number of hiding places
- Temporary loss of ability
  - Being slowed down by a wound
- Resource loss
  - High cost for being caught
- Limited opportunities to escape when seen.

Unpredictability (Decrease when High in Mastery at the System level)

- Negative feedback

    - Hunters catch-up
    - No possible escape
  - Inconsistent or random behaviors
    - Hunters cannot be tracked or their positions predicted
  - Involuntary gameplay mode switch
    - Stealth
    - Flight
- Caution (Persistent Low Mastery at the Action level):
  - Taking a risk
    - Risky portions in main path or secondary path, where the player can be spotted more easily
  - Misdirecting affordance
    - Traps
- Anticipation of failure (Persistent Low Mastery at the Self level):
  - Pretend danger
    - Ominous noises
  - Perceived cheating on the part of the game
    - How can they always find me?
  - Opportunities for recurring mistake
    - Hunters can hear the player when he makes noises
- Avoid Hope (No Increase when Low in Mastery at the Action level, Freedom at the System level and Mastery at the System level):
  - Can't exploit advantage opportunity
    - Hunters show no weaknesses
  - Can't pursue concurrent goals
    - Fleeing prevents access to resources or means of escape
  - Can't exploit preparation
    - Hunt can only end by a narrow escape that feels lucky

This example illustrates another continuous emotive factor driving people to continue playing: *freedom from fear*. Players engage in a sorting process to make things clear and predictable so no crazy and unpredictable events that might harm them will emerge [Shef 08]. Achieving this in itself makes people feel happy because stressed emotions disappear. These feelings of stress also explain why focusing on achievement induces negative emotions [Yee 06].

In the game world, *diversity* is important as well. The player does not want to view the same surroundings over and over again. The environment and emotions should, however, remain realistic at all times. The artwork, programming and environmental context greatly contribute to a game's success. The gadgets and other features must appeal to the user and consequently reap satisfaction. With the introduction of new ways of interaction, such as the Nintendo Wii, ever new challenges and variety are provided.

2.3.2 Motivation

A different focus is taken by David Ghozland. In his article 'Designing for motivation', which appeared in *Gamasutra* on June 7th, 2007, Ghozland [Ghoz 07] describes a system to manage internal motivations coming from both the game and the game context. External motivations are not taken into account, since they are highly personal and cannot be verified.

A typical game starts with a 'tutorial'. This guarantees accessibility, essential to guide the player in his development. The tutorial aims to get the player acquainted with the game. It typically constitutes the first few levels in which the basics are taught. However, it should not be obvious to the player that he is 'only' in a learning stage, it must appear as natural and similar to 'real' gameplay as possible. Tutorials listing rules and game controls are not immersive and will not stimulate the need to learn and understand [Ghoz 07].

Motivation is embedded in the gameplay and depends on the player's needs. In the beginning, the needs of the player as he immerses himself in the universe are directly linked to the game. They are artificially created by the game designer. It can be said that there is a ”tacit agreement with the player” [Ghoz 07]. This means that the game design proposes a certain promise of the game universe and the game itself. For example, an RPG promises character growth combined with a measure of empowerment. An FPS (First Person Shooter), on the other hand, promises large weapons and powerful enemies.
In the game mechanics and system the player has to feel challenged in order to stay entertained and be tested along with rewards to motivate him to keep playing. The player is motivated to achieve the promised goals, the game must fulfill its promise if the player displays the right skills.
Therefore it is not only important to know and understand player needs, but also to control the progress of his needs, so they can be increased, varied and modified throughout the entire course of the game. This is where adaptability starts playing an important role.

It is important to know at all times what state the player is in. This information can be used to determine his needs, adequate rewards and the level of challenges to be offered. Therefore player motivation is an outcome of the following functions [Ghoz 07]:

- Player state (P)
  This is the state of game variables of the player's avatar, for example his life, armor and the quality of his equipment. Talent, knowledge of the world and of game mechanisms are also typically included. We can say this function represents the player's strength.

- Needs (N)
  These are the needs at the moment the challenge arises. Needs depend on the player's state and on his progress into the game. Game design also adds needs as the player goes along.

- Reward (R)
  This is the player's expectation of the reward. The value depends on the

type of reward, which is a function of the needs. It depends on the estimated difficulty and also on the player's past experience with the reward system.

- Challenge (C)
  This is the player's expectation regarding the challenge. The value is high if the player believes in his own capacities. However, if he does not feel comfortable or if he has doubts about his skills, the value is low.

This means the following equation applies to describe motivation M for a particular Player State p:

$$M(p) = N(p) * C(p) * R(p)$$

Graphically, this can be presented as follows [Ghoz 07]:

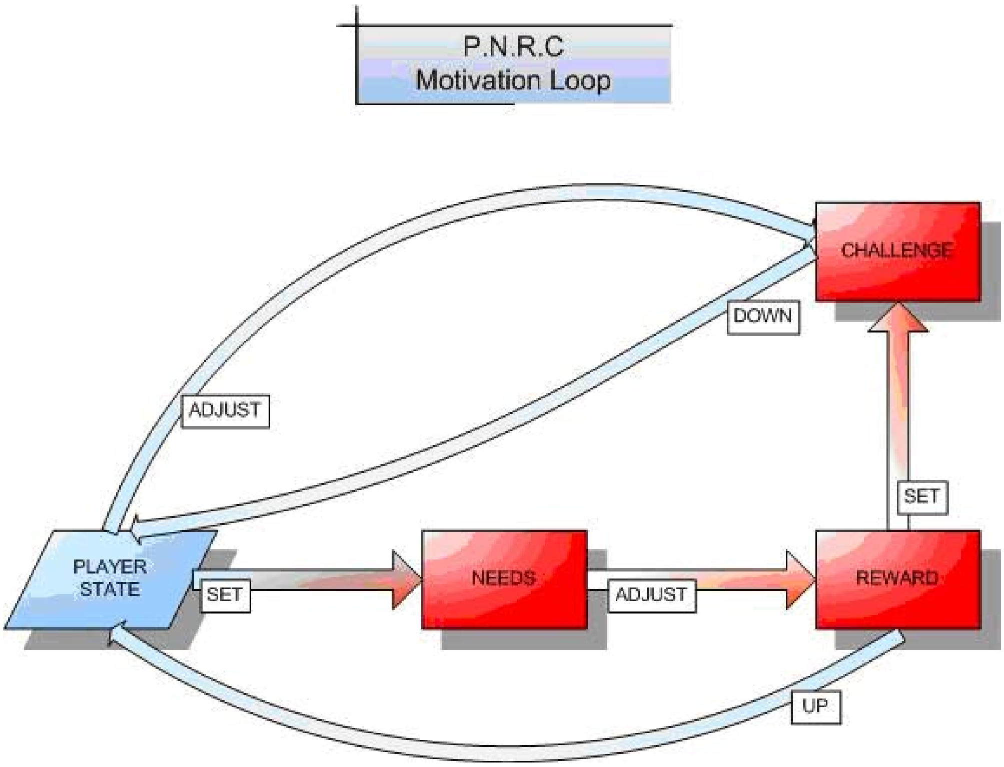


Figure 2.3: Motivation as a function of Player state, Needs, Reward and Challenge

Rewards for challenges must always be in correlation with the universe and the player's expectations. When a player receives a reward that is disproportionate or inferior to the difficulty, the challenge-reward system will not work. But rewards may not come randomly either, as such a system cannot be properly grounded and mastered and can become a source of frustration to the players.

The core structure of a game constitutes a constant cycle of building the player's needs and answering them by a succession of challenges and rewards [Ghoz 07]. The player, now equipped with skills and basic gear to start his adventure, will have to be guided into his new world by the designer. Guidance is done by offering progressive challenges which are rewarded by, for instance, new tools necessary to master new challenges.

A very basic method underlying challenge-reward cycles is the *D&D method*, more commonly known as "door, monster, treasure". However, this is a raw, functional tool which calls for tuning and creativity if a motivating system is to result.

Motivation from rewards

Rewards can take many forms, but their main role is to keep the player motivated. In many RPGs, gameplay is built on the growth of the character's strength. This happens by accumulating points of experience, gained from mastering challenges. When the player has earned enough points, he can pass to the next level. Experience as well as equipment for further challenges will be gained this way. Leveling up is the most important and obvious feedback mechanism for improvement in competence, because each time a level is mastered, something more difficult and complex will be presented as the new challenge [Grei 07].

The system works as follows:

1. The needs (N) are great
2. The expectations on rewards (R) are big
3. Expectations on challenge (C) are proportional to player strength (P)

As P grows throughout the game, N, R and C must grow accordingly.

However, this line of reasoning has weaknesses. Progression cannot help but be limited. If there were no limits, there would be no goals or references, and thus motivation would be weak. But as soon as the limit of progression is reached, the game similarly loses its captive power and motivation disappears. In the action-RPG *Diablo* the designers attempted to solve the problem by randomly assigning categories and reward characteristics, so as to increase players' motivation to re-attempt a challenge in order to achieve the perfect item. The problem in this case is the lack of reference for the challenge-reward ratio, since the reward value is not in any way proportional to the challenge.

Motivation from needs

Motivation can also be driven by needs. For example, in RTS (Real-Time Strategy) games such as *StarCraft*, motivation comes mainly from construction mechanics requiring resources. Players begin with just enough resources to start the development of a base and troops. The whole game is based on resource acquisition and control. N will remain significant throughout the game, with R being directly related to N. P changes as battles with other players are fought and C depends on P.

However, once the resources run out, the game will be decided. The catch here

is that as soon as it becomes impossible to have access to resources, the motivation loop will break and the player will have no reason to continue playing. This leaves the winner frustrated because his victory comes from the opponent's retirement.
Possible solutions are creating other resources which are easier to access, such as food or farms, the possibility to obtain resources by trade or inexhaustible resources. With the latter, though, motivation is based on the number of sites a player can control and on time optimisation.

Motivation from challenge

When motivation is related to challenge, the underlying important factor is P which represents the level of the player's skills. P now follows a learning curve, since the main objective is to improve skills so harder challenges can be mastered. C is the aim of beating the opponent, N is then a need for knowledge and excellence, and R is the victory won by mastering each challenge, proving the player's skill level and bringing him closer to a final victory.
Motivation comes from the player's visible progress in skill acquisition and mastery. However, just victories are not sufficient to keep motivation going. Possible additions are access to more options as performance improves, or accumulation of points. Motivation becomes directly linked to the game mechanics [Ghoz 07].

Motivation from the player state

Games in the style of FPS (First Person Shooter) get their motivation from player state P. Whatever the player possesses in the game is not permanent, artefacts may be lost as the level ends or when he dies. The objective is then to keep the gained strength as high and as long as possible so the final 'boss' can be defeated.
In this case, C is directly related to P, as the challenge will become harder as the player gains strength. N is high and can be increased through getting bonuses and upgrades, which form the R.
Difficulty will continue to rise independently of P, so the player has to see that he keeps up in strength gain. If difficulty rises too high for P, motivation will quickly disappear. This results in a natural selection, leaving only the strongest and most perseverant players to progress.
Possible accessibility options are permanent bonuses, manual adaptation of difficulty level and the possibility of winning without upgrades. However, in these cases the game story and mechanics have to be sufficient to keep the player going.

In most games, motivation comes from a delicate balance between the four functions. Even though the emphasis can be on a main motivation, balance remains essential. An example of how these four functions can be merged, considerations for a score system are discussed:

Score system

A score system is an integral part of a reward system, and therefore a good way to manage motivation. It allows for both the rewarding and the confirmation of the player's successes. Certain mechanics can be built in to encourage the player, but for the score system, the focus remains on score bonuses and experience points.

The reward for the player consists of the points he earns, and his rising in the ranking. Score determines progression, bonuses reward good performance. This is what determines the value for R.

The difficulty of a challenge can be expressed by the number of points to be gained, which pushes the player to harder and more complex challenges C in order to gain more points. In this way, the whole game universe becomes organised according to the point values.

As far as experience points are concerned, they are earned in principally the same way save for one difference: the score is integrated into the game system. This is where the needs N start playing a role. Progression can be measured in the form of levels and associated talents, so the player has a need to acquire skills so he can move up in levels. Another motivating aspect of the experience system is the resulting ranking compared to other players and compared to the world. The problem with this system is the progression limit, or the maximum level. Once this limit is reached it sounds the death knell for the motivation.

Here, the motivation is related to the progression and the performance. The score is thus an efficient tool of the system to recognize player's efforts and to follow his evolution (P).

2.3.3 Game accessibility

Another key point in game design is accessibility. A game can be truly appealing, but if the interface and controls are counter-intuitive and hard to master, motivation will inevitably be lowered.

There are a few considerations to keep in mind while designing games for a wide audience.

To begin with, it is important to ”extract meaningful affordances around the genre you are dealing with, and make sure they match the audience's expectations” [Glin 08]. When someone picks up a game, they have a certain expectation of it, which appeals to them and triggers the motivation to start playing. For instance, in an FPS game, the expectations are a large assortment of weaponry, the ability to kill enemies and possibly a multiplayer element. If such promises are not fulfilled, gamers will undoubtedly be disappointed.

Secondly, simple control schemes that make sense for the game work miracles. The designer should create accessible interfaces, with as few buttons as possible for game actions. Simple, configurable user interfaces and control schemes must be available. If complex interfaces are necessary to facilitate more control over the game, two types of interfaces have to be offered: beginner and expert interfaces. This lowers the threshold for playing the game. Duration and seriousness of the game also have to be taken into account: a game that lasts for only 30 minutes requires control schemes that are really easy to master, otherwise the effort of learning the controls will overpass the game experience.

A game is merely a ”system waiting to be understood” [Rose 08]. A player satisfies his needs N by actively influencing the game, and the only way to accomplish this is through game mechanics. Adding mechanics complicates the player's influence over the game world. However, it also adds complexity to player input, whereas it rarely alters the player output [Rose 08]. Therefore the complexity of the mechanics has to be well thought over, as it may never interfere with the gamer's ability to comprehend the game world and active experiences.

Games distinguish themselves by means of their gameplay. They determine what a player does during the course of play, thereby defining the game's personality. Art, technology and storytelling are just side features when compared to mechanics. How a player felt and what he did will decide his final judgement of the game.

It is of great importance that any unnecessary things are removed. A small set of mechanics will always help make a game easy to understand and hard to master [Rose 08]. A small collection of player choices guarantees that the mechanics are well-tested, used and enjoyed. Overcomplicated games always find their faults in their highly complicated mechanics. This only dilute's the player's experiences, distracting him from the truly appealing features of games. The feel of mechanics has to be tuned to require as little cognitive load as possible. Strong, simple ideas are the ones that explain themselves and reinvent gaming.

Finally, the game has to be extensively tested with all groups the game is designed for, including mainstream and disabled gamers [Glin 08]. Each unique user has different requirements, and even though generalisations can be made, it is important to know the reactions to certain types of input modules, game mechanics and emotion-inducing contexts.

2.3.4 Competition versus cooperation

As mentioned at the start of this chapter, competitive elements are a key aspect in all games. Without this, the ability to judge individual progress would be difficult, making meaningful play impossible.

The most important meaning to be derived from the game lies in its victory conditions. The competitive activity driving a game's system of conflict is the struggle among players during the course of play to achieve the game's ultimate goal. Games are constructed in such a way that achieving the goal is difficult and represents the intrinsic challenge. Conflict arises as soon as players start struggling among each other.

Competition occurs in many forms. A single player can compete against a game system (eg. Tetris), against another single player (eg. Draughts) or against a whole group (eg. Tag). Another possibility is that each player plays only for himself, such as in a footrace, or that they individually compete side by side against a game system (eg. casino Blackjack). Finally, players can compete as a group against another group (eg. a soccer match) or the group of players can cooperate against a game system, such as when performing a quest in a multiplayer game. Most games, however, do not employ a single form of competition, but integrate several forms. There are even games in which the competitive system changes over time, as seen in television show games in which teams initially work together, but as the game progresses and team members fall away, the remaining players become each other's rivals [Sale 04].

Conflict in a game can take on a direct or an indirect form. In Tetris, for example, a ranking of high scores is kept. The competitive goal is then to beat the current high score, which can be a direct competition against yourself or an indirect competition against another player who holds the high score. In a direct competition, the player has the ability to directly influence the outcome of the

game through direct interaction with the opponent. An indirect competition, however, happens by means of a third medium, such as a ranking of high scores or a panel of judges giving scores for the player's individual performance.
Direct and indirect conflict can also be integrated in a game. The form of conflict the game takes depends on the way the game answers certain design questions. Important aspects to consider are the following [Sale 04]:

- How many players can play?
- Do they play simultaneously or do they alternate playing the game?
- Is there a high score list?
- Are players given constant feedback about their relative scores?
- Does the game pause to allow players to directly compare their scores and other game statistics?
- Are there computer-generated opponents and obstacles that players face together or do the players serve as opponents for each other?
- Does the structure of the game allow players to have direct conflict with each other?
- Are there resources for which players can compete?
- Can players spend money to continue the game or enhance their play?

The entire scope of the game is a space for possibility, for possible conflict to take place. A rich space of possibility supporting a wide range of conflict increases the potential variety players can experience and thus lead to a more meaningful game.
There are criticisms, such as by Bernard DeKoven, game designer and author of *The Well-Played Game* [De K 78], who points out that as soon as winning and losing enters the conflict of the game, players become focused on that single goal, blocking out the many other aspects the game has to offer. However, this cannot be stated as plainly as that. Competition offers a meaningful goal players strive for. It gives structure to the game, in that the player's actions are integrated in the larger game context and are dependent on the competitive nature of the game. Without a goal, there is nothing to motivate players to take an action that changes the game state from A to B, since there is no point in the change.

While it is true that all games are *competitive*, it is just as true that all games are *cooperative* [Hopp 09, Sale 04]. The two terms are by no means mutually exclusive, and therefore it is not a contradictory statement. Playing a game means the player immerses himself in the discourse of the game with other players. Each player has to understand and engage in the shared meanings of the game in order to play. All players 'speak the language' of the game, and therefore can play together.
Cooperation can occur on two levels: *systemic* cooperation, referring to the fundamental, discursive cooperation that is intrinsic to all games [Sale 04], and

player cooperation, which means that all players collaborate to achieve a common goal. The latter is typical for multiplayer games.

Players cooperatively form the space of the game, in order to create a competition for their own amusement. So it can be stated that while there always is a form of competition in a game, it occurs only within the larger cooperative frame sustained by the participation of the players.

2.3.5 Goals

Goals are fundamental to games. They are integrated in each form of conflict. When a game finishes, the goal has either been reached or it has not. This quantifiable outcome is part of a game's defining character [Sale 04].

Goals can be subdivided in short-distance, middle-distance, and long-distance goals [Shef 08]. A long-distance goal is something the player has to be constantly reminded of, typically the overall final goal of the game. Middle-distance goals are goals which cannot be directly solved, but should be solved within a short period of time. An example is running to a bridge in a forest for which you have to perform a certain task before you can gain access to it. Short-term goals are things that can be immediately achieved in a simple, linear way.

In a game, the player is confronted with multiple goals. It is the player's responsibility to determine which goal to accomplish first. Important is the game mechanic of how to get at the middle-distance goal. This is the way most games are set up. It is a constant cycle of 'fear' and 'relief'. If you are in an enclosed area, then completing a middle-distance goal to escape it makes you feel relieved [Shef 08].

Goals are defined by the game's rules and are tightly interwoven into the formal game structure as a whole. It is, in fact, the central feature of this structure. It serves to sustain interest, engagement and desire to keep playing. Through players' actions, they get a small step closer to the goal with every step they take, and the distance to the goal allows players to understand the impact of their actions, making meaningful play possible.

A long-term goal typically defines the endpoint of the game. When it has been reached, the game is over. There are games which postpone the endpoint by presenting the same goal to the player over and over again, each time in a slightly more challenging way. This does heighten the sense of inevitable death, though, because the game will end once the player's capacities to fulfill the challenge do not match the difficulty of the challenge anymore.

A game is a journey from the starting point to the final goal, and throughout this journey, the well-designed game should make the journey as efficient as possible without compromising the challenge, and letting each element contribute either directly or indirectly to the whole game experience.

2.3.6 Rules

Rules make up the very heart of a game. They define the inner structure, the possible actions and guide player behaviour. They are the formal identity of the game. Even if environment and aesthetics are stripped away, the rules remain the same.

Rules typically share the following characteristics: they limit player action, are explicit, unambiguous, fixed, binding, repeatable and all players share and ad-

here to them. Rules and experience are two separate elements. Even though rules shape the experience, it is well possible to change the latter without altering the first.
Rules can occur on three different levels [Sale 04]:

- **Operational rules**
  These are the guidelines underlying the game the players need to know in order to play, presented in the written rule-book accompanying the game. When considering digital games, the operational rules are concerned with both the game's internal and external events, such as player input and output, and expression of choices to the player.

- **Constituative rules**
  They are the underlying formal structure existing 'below the surface' of the operational rules. They are formal and mathematical. In digital games, they are typically integrated in the code and handle the game's internal events (how the player's choices are processed).

- **Implicit rules**
  The 'unwritten' rules of the game which are important during play, such as etiquette, fair play and other implications of good player behaviour. Unstated assumptions about the game platform can also be included. Implicit rules are context- and game-dependent. Playing with implicit rules often results in innovative design ideas.

The aforementioned 'formal identity of the game' is made up by the operational and constituative rules. There is a two-way relationship between the concrete, real-world rules on the one hand, and the logical, abstract ones on the other. There is an interpretation between the two rule levels that produces meaning as a translation occurs between the levels [Sale 04]. One form of rules allows for the expression of the others. How players make sense of a game, however, depends on the intertwined interaction between all three rule levels.

When designing, the designer builds the game systems through iterative processes, addressing a certain rule level with every iteration. Understanding how rules interact to shape games makes it easier to design meaningful play.
Good game design becomes apparent when players can focus solely on game experience rather than be concerned with the game rules. So the important aspect here is to create experiences in which elegant rule design allows for maintenance of play focus. For instance, the aesthetics of the game should embody all aspects of game information, both the progress towards the final goal as the possible actions during play itself. Representations of the players (avatars, pawns) have to be in the same space so that their positions, characteristics and properties such as strength and health can be compared. This comparison has to be immediate and intuitive. Finally, the players have to be able to see the consequences of their actions at all times.
The whole process of game design can be seen as "creating a clear ruleset and tying it to the actions and outcomes of genuinely meaningful play" [Sale 04]. If this succeeds, *discernible* and *integrated* outcome should result.

2.3.7 The MDA framework

The MDA framework, or *Mechanics, Dynamics, Aesthetics*, is a formal approach for understanding games [Huni 04]. It attempts to "bridge the gap between game design and development, game criticism and technical game research". Games are broken down into their distinct components, which are then coupled to their design counterparts: Rules to Mechanics, System to Dynamics and Fun to Aesthetics. By using these levels of abstraction, the dynamic behaviour of game systems can be conceptualised, and viewing games as systems helps develop techniques for design and improvement, control outcomes and tune for desired behaviour [Sale 04, Huni 04].

Designers and players typically have different perspectives of a game. For a designer, mechanics give rise to dynamic system behaviour, which then leads to aesthetic experiences. On the contrary, a player typically perceives the aesthetics first, emerging from observable dynamics and eventually also operable mechanics. By taking the player's perspective, experience-driven design is encouraged. Therefore, the aesthetics will be described first, then the dynamics and finally the mechanics.

Aesthetics are the desired emotional responses evoked in the player when interacting with the system. In order to be more direct in describing aesthetic experiences, Hunicke et al. [Huni 04] suggest the following (limited) vocabulary:

- *Sensation*
  Game as sense-pleasure
- *Fantasy*
  Game as make-believe
- *Narrative*
  Game as drama
- *Challenge*
  Game as obstacle course
- *Fellowship*
  Game as social framework
- *Discovery*
  Game as uncharted territory
- *Expression*
  Game as self-discovery
- *Submission*
  Game as pastime

Using these terms to describe the aesthetic aspects the game in question emphasises, it can help in understanding how and why certain games appeal to certain players.

Dynamics describe the run-time behaviour of the mechanics acting on player inputs and their outputs over time. They work to create aesthetic experiences.

For example, *Fantasy* is created by a wonderful, imaginitive game world in which the player immerses himself. *Fellowship* comes from interaction with other players or by creating tasks which are too difficult to achieve alone. *Expression* happens when individual players are enabled to leave their own mark, such as by building or earning certain game items or by creating a personal avatar.

Mechanics are the particular components of the game, described at the level of data representation and algorithms. Together with the game's content, the mechanics support the overall gameplay dynamics. Tuning the mechanics can allow for subtleties in the dynamics which may keep players motivated and involved. Changes on the level of mechanics work their way through dynamics and aesthetics, creating a different experience for the player.

The fundamental idea underlying this framework is that games are more like artifacts than like media, meaning that *behaviour*, rather than the media the player perceives, makes up the content of the game. Since behaviour emerges from interaction according to rules, this is in line with Salen and Zimmerman's reasoning about rules defining games [Sale 04]. This line of thinking helps view games as systems building behaviour through interaction and supports clearer design choices and analysis at all levels of study and development [Huni 04].

2.4 Applied Ludology

To take the analysis of games still one step further, Järvinen [Jarv 07b] introduces methods for what he calls *applied ludology*: a practical hands-on analysis and design methodology, meant to complement theories of games as systems with psychological theories of cognition and emotions.
The work follows from Järvinen's previous work in which he proved two theses:

1. Any kind of game can be identified through a limited number of structural features called game elements.
2. The experience of playing a game can be analysed with a set of 'psycholudological' concepts (i.e. psychological principles adapted for the specific purpose of analysing play in games).

This *Game Design Theory* allows for analysis and design of game rules without including the behaviour of human players. The actual playing is left to humans. However, this separation results on essential qualities from either the systemic side or the human side being ignored, therefore applied ludology aims to produce an analysis tool which integrates both aspects. For a full understanding of game playing and how game mechanics and structures support this, such an analysis is essential.
The interactions between players and games have been conceptualised as *gaming encounter*. This is defined as "a concept that emphasizes the behaviour of players, and the contexts where the game takes place, rather than the inner workings of the system" [Jarv 07b].

Järvinen proposes four key steps for identifying particular gaming encounter aspects:

1. Method for identifying and analysing game elements.
2. Method for identifying game mechanics and the goals they relate to.
3. Method for identifying player ability sets.
4. Method for identifying eliciting conditions for emotions in gaming encounters.

Method for identifying and analysing game elements

In order to understand how a game system works, the different elements comprising the system must first be identified. These elements stem from Game Design Theory [Jarv 07a]. The following is a short overview, in increasing order of complexity:

- **Components:**
  The resources for play, the things the player can possess and modify in the game, between players and the system. Typical examples are tokens, guns, balls, characters, points and vehicles.
- **Environment:**
  The space for play, such as boards, grids, mazes, levels and worlds.
- **Ruleset:**
  The procedures with which the game system constrains and moderates each individual possibility for play. Rules define the relationships between game elements, with goals being the most important.
- **Game mechanics:**
  What actions the player can take in order to interact with game elements to attain goals and influence game states. Placing, shooting, manoeuvring are examples of what players have to perform in many games.
- **Theme:**
  The context of the game which functions as a metaphor for the system and the ruleset to give it a unique meaning.
- **Information:**
  What the players need to know and what the game system stores and presents in game states: Points, clues, time limits, etc.
- **Interface:**
  In case there are no direct, physical means for the player to access game elements, interface provides a tool to do that. Examples are joysticks, steering wheels, guns.
- **Players:**
  The participants in the game who play, in various formations and with various motivations, by performing game mechanics in order to attain goals. Their behaviour, moods and skills are included.
- **Contexts:**
  The physical location where the gaming encounter takes place.

The relationships between the elements can be represented as follows:

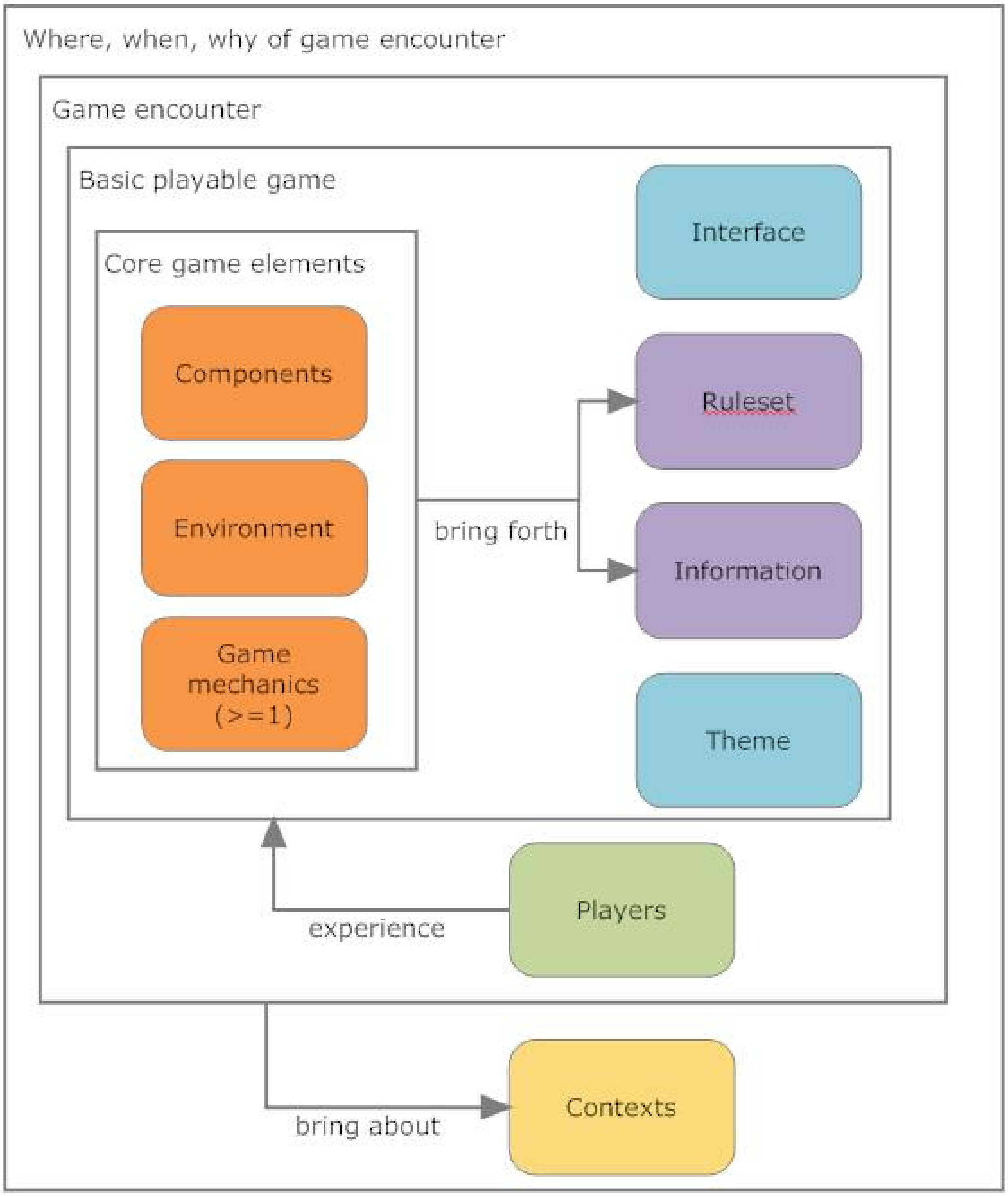


Figure 2.4: Relation between game elements

In order to be meaningful, a game must minimally have Components, Environment and at least one Game Mechanic. From these elements a Ruleset and Information emerges. Players are needed for a gaming encounter, and several Contexts are brought about, possibly varying with each encounter.
The Ruleset, Game Mechanics, Theme, Interface and Information are compound game elements. They seldom exist separately, they are typically embodied in other elements, keeping the dynamic whole together. Rules become embodied in game elements.

Once the game elements have been identified, it is important to attribute ownership to the them, since ownerships can create inherent tensions due to conflicting goals and scarcity of objects. Tension is a prospect for emotion. Ownership can fall into one of three categories: the *self*, *other(s)* or the *system.* When this has been accomplished, the final step of this phase in the game analysis is to analyse whether the elements have other significant attributes relating to things like goals and player roles.

Method for identifying game mechanics and the goals they relate to

Game mechanics are about doing something in a game. In face, from the perspective of the player, the whole game experience might be defined by performing mechanics. How the mechanics are given value to by self, others and the system is decisive not only for the outcome of the game but also for the player's subjective experience.
Besides individual mechanics specific for a certain game, the *primary* mechanics, there are often other supporting mechanics, or *sub*mechanics, which are of a more general nature. For example, in a fighting game, the primary mechanic may be to kick your opponent in the face to weaken him, and the submechanic can be to crawl up to him.
Game mechanics are the means to attain goals, and therefore they are directly related to goals. This implies that they are available on the same levels as goals: short-, middle- and long-term. It appears that primary and submechanics are mostly available on the long term, throughout the whole game, whereas a third type of game mechanic, a *modifier game* mechanic, may be available on the short term. Think of a certain action that has to be performed in order to achieve a specific short term goal, such as a speed boost or a special invincibility shield.
Salen and Zimmerman [Sale 04] mention a 'core mechanic', which is defined as "the actions that players repeat in a game, again and again". In the context of applied ludology, core mechanics consist of the possible combinations of primary game mechanics and submechanics, possibly complemented with modifier mechanics [Jarv 07b]. The method aims to decompose the core mechanics into the three individual mechanics mentioned above.
The last distinction to be made is that the goal of the core mechanics is not necessarily identical to the ultimate goal of the game. The core mechanics might be focused on killing as many monsters as you can, while the overall goal is to kill the grand boss and to free the empire.

The template for the study of core mechanics Järvinen suggests, based on analysis of "over a hundred games of various types" [Jarv 07b], comprises the following steps:

1. Identify the long-term goal.

2. Identify the core mechanic consisting of a primary mechanic and its possible submechanics.

3. Identify the middle-term goal the core mechanics relate to.

4. Identify the possible modifier mechanic(s).

5. Identify the short-term goal the modifier mechanic(s) relate to.

It is possible to apply this method even when games increase in complexity. The approach is to divide the game into analysable sections, for example study the core mechanics and goals of the auction house in World of Warcraft as one entity, and the respective core mechanics of particular quests and 'grinding' on their own, after which these wholes can be analysed in relation to each other, according to the overall goal hierarchy of the game [Jarv 07b].

Method for identifying player ability sets

The need for identifying player abilities originated because digital games call on both cognitive and psychomotor abilities, and increasingly also physical abilities, especially with the rise of exergaming such as Nintendo Wii. The abilities are tested by game mechanics and goals, which means the analysis has to focus on the combination of cognitive, psychomotor, and physical abilities that game mechanics require players to perform. When the abilities for some reason do not match with the goal, and the performance of the game mechanic, this should raise questions about whether there is a flaw in the game design. Therefore this analysis tool also serves to explore and validate design solutions.
Highly relevant is to identify the abilities that can directly affect a successful performance, in other words, which abilities contribute to the margin of error. These can be seen as *uncertainty factors.* This choice in focus enables us to distinguish abilities which are "not high level prerequisite abilities (for example visual and auditory perception) or not rapidly routinised to the degree of triviality" [Jarv 07b].

The basic idea behind the method is, for a certain game, to view player abilities as uncertainty factors and study their relation to the core mechanics and the function they serve on the different goal levels. In more complex games with multiple goals and game mechanics, it is possible to spot inconsistencies in the space of player abilities by identifying abilities through the goal hierarchy and set of game mechanics.

Player abilities are also differentiating factors between players, and thus relate to uncertainty concerning outcomes. Uncertainty is a very important factor in most games. It motivates players to play, so they can improve their skills in order to reduce uncertainty. Uncertainty can thus be said to be a fundamental source of emotions for players, and believing in one's own abilities affects it as an emotional component of gaming encounters.

Method for identifying eliciting conditions for emotions in gaming encounters

The challenge in a game is to analyse game-specific eliciting conditions and the emotions they are likely to trigger. The intensity of emotions can be affected by how close in psychological space the player feels to the situation which potentially elicits emotions. Emotions have already been discussed in detail [2.3.1], therefore I will proceed with the method.
Firstly, the eliciting conditions for hope and fear have to be identified. Together with uncertainty, hope and fear make up a compound emotion essential to player experience: suspense. When these have been identified, the next step is to identify how the eliciting conditions are embodied into the design of the elements. This can be accomplished by applying the theory of game elements.

2.5 Summary

This phase of my research has been about the reason why people play games, the interaction that takes place between players and game elements, and how players respond to certain conditions. It has been important to know how professional game designers tune the different gaming aspects in order to create successful, appealing games, so that I can apply this knowledge when designing the Aquima modelling game. People must be intrigued and stay motivated when playing.

Inducing motivation depends on many factors, many of which are interrelated. An overview will be provided below.
The concepts from the different models for game aspects allow for analysis of player-game interaction and can serve as a base for new game design. Using this, important aspects can be emphasised and inconsistencies eliminated.

Different frameworks for successful game design have also been explored. In many frameworks, game motivation aspects have already been integrated. Methods such as Bura [Bura 08] and Ghozland [Ghoz 07] describe focus directly on game design, whereas approaches such as the MDA framework [Huni 04] and applied ludology [Jarv 07b] aim at creating an understanding of games through post-design analysis.
The methods of applied ludology together facilitate the explanation of the inner workings of both the game mechanics and how their players respond. Applied ludology as Järvinen describes it integrates the most important aspects in a single theoretical framework, which, with the inclusion of the other discussed models, will therefore be used to form a solid theoretical base for well-grounded game design.

Principles of good game design

The key behind good game design is to create a constant cycle of motivation, a balance of Needs, Challenge, Reward and Player state, as captured by Ghozland [Ghoz 07] in his equation:

$$M(p) = N(p) * C(p) * R(p)$$

Furthermore, the following principles seem most important to keep in mind during game design:

- Games should be kept simple, with as few distracting unnecessary features as possible.
- Ensure that controls are intuitive to prevent cognitive overload.
- Make the rules so transparent that player can focus on gameplay only.
- Always keep goals in mind: make sure they are achievable in good time, otherwise the player will feel frustrated.
- Make sure the mechanics are properly related to the goals.
- Make sure the rewards are proportional to the challenges.
- Always keep the player's needs and state in mind.
- Keep in mind how interactions will affect the player's emotions.
- Immerse the player in the game world.
- Ensure the player feels competent and autonomous.

2.5.1 An overview of gaming aspects

Gaming motivation variables

From this exploration of where motivation for gaming comes from, the following list of variables can be extracted:

- Experience
  - Goals
    - Short-term
    - Middle-term
    - Long-term
  - Autonomy
    - Freedom of action
  - Interaction
    - Competition
    - Cooperation
  - Competence
    - Challenge
      - Time pressure
      - Score system
      - Skill improvement
    - Reward
      - Feedback (leveling up)
    - Being in control
    - Specialisms / role playing
  - Emotions
    - Freedom from fear
    - Narrative
      - Presence / immersion
  - Accessibility
    - Intuitive controls
    - Manageable tasks
    - Intuitive interface
    - Guide player behaviour
  - Psychological well-being
    - Relatedness
    - Vitality
    - Self-esteem
    - Positive affect
    - Achievement
    - Need
    - Adaptability

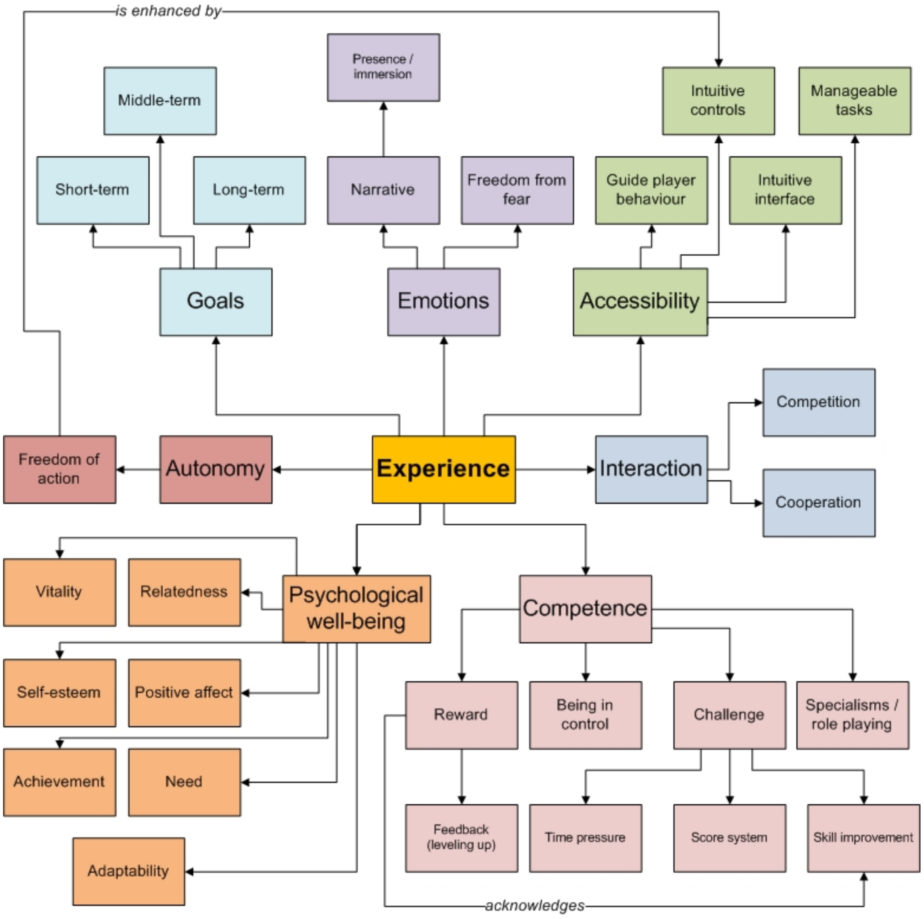


Figure 2.5: Factors contributing to motivation for gaming

Game Design Aspects

From the exploration of good game design, the following aspects have to be included to create a sound game.

- Game Design
  - Components
  - Rules
  - Environment
    - Diversity
    - Realism
  - Theme
  - Information
  - Interface
  - Players
  - Contexts
  - Mechanics
  - Dynamics
  - Aesthetics
    - Graphics
    - Music/sounds
    - Visual effects

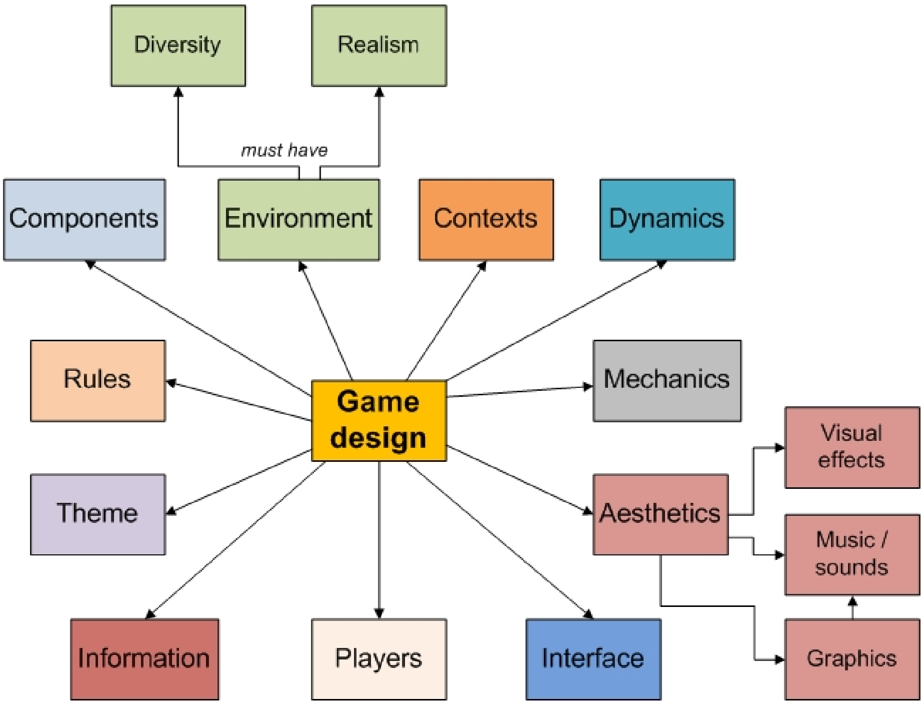


Figure 2.6: Important factors for game design

Chapter 3

Modelling, gaming, and HCI

This chapter first explores a paradigm which describes an interaction-oriented perspective on formal modelling. Considering modelling by looking at the interactions involved captures the essential aspects which constitute the whole activity. This perspective is very useful when analysing the structure of existing modelling tools and modelling processes, as will become clear in Chapter 4. Both modelling and the goals and constraints imposed on the process are analysed using the interaction-oriented perspective, exposing the core activities of modelling.

Secondly, the benefits of serious gaming are briefly analysed. It appears that active interaction is a very powerful tool for knowledge transfer and retention due to its engaging nature. Then the different elements are linked together in an explanation of why gaming, modelling and interaction should be considered a vital part of Method Engineering.

A striking fact is that educational games are not sold on such a large scale as entertainment games. A comparison between the two genres is made to find out what entertainment games have that so massively appeal to the general public.

Finally, in order to evaluate modelling game design, I will explore the necessary HCI aspects for playable modelling. Not only playability has to be accounted for, but also player mood and the success of the collaboration must be measured. Therefore I will provide an integrated approach with measurable indicators for all aspects.

3.1 Embodied Embedded Usability

To begin with, I want to investigate a relatively new perspective within usability research which provides a more intuitive way of looking at modelling. This ultimately fits in with the gaming approach. I will explain this paradigm, *embodied embedded usability*, first described by Van Dijk [Dijk 08], in more detail. Then I will describe how this perspective can be used to explain modelling interaction more intuitively.

Embodied embedded usability can provide us with an **interaction-oriented** perspective on formal modelling processes and their implementations. When looking at modellers at work, they appear to be engaged in an intuitive, collaborative process. However, on the level of modelling languages, elements and relations modelling can sometimes appear to be a blur of thoughts, negotiations and decisions. In order to apply structure to this seemingly chaotic process it can be very clarifying to view it in terms of interaction [Hopp 09]. The focus is then on interactive behaviour, such as discussions between and decisions taken by the modellers.

The paradigm describes an "action-centred, subpersonal account of daily behaviour" [Dijk 08]. People appear to function largely by means of automatism in their daily activities, without conscious thoughts about each individual action. However, there is continuous interaction with the features encountered in the environment: a direct coupling between brain, body and the environment. The latter provides feedback to guide and at the same time constrain a user's actions. In this way users shape their own environments through their actions [Dijk 08].

Interaction is essential to bring modelling to a successful end [Hopp 09]. Modellers must engage in continuous interaction both with each other and the modelling domain. They must agree upon a clear final goal and the scope of the domain. Clear boundaries and rules must be set, which form constraints on the modelling process as a whole. Behavioural factors and jargon may also pose constraints on the process. In short, modellers shape their own environments and scope through their interactions.

Further constraints on the model itself are factors such as the need to express the model in a formal language and syntactical correctness [Bomm 08]. They may also emerge progressively as the modellers decide upon certain things or if environmental constraints in the modelling domain are encountered.

The main idea behind this approach is to let the world be its own representation [Broo 91]. Behaviour-based robots show that this is a true statement: their intelligent behaviour arises from the tight coupling between their bodily actions and the physical environmental constraints they encounter, which simultaneously restrict their behaviour. There is no need for internal representations of knowledge, in fact, representations only seem to get in the way.

This idea essentially holds for people too. During their daily activities, people tend to do many things semi-automatically. They do not consciously think about each individual action. Rather the actions are part of a larger task, or a well-known routine: in terms of skill learning, such tasks have reached an *autonomous level*, in which they require little cognitive attention but become hard to interrupt. This behaviour, even though the basic idea is the same every day, is still guided by environmental elements and arising opportunities which may vary if the tasks are performed in a dynamic environment. Therefore cognitive action emerges from the continuous flow of input and output between the actor and the environment [Dijk 08].

The same holds for interaction between users and technology. It is not simply the user telling the computer what to do, rather it is the user responding to the computer's prompts and the computer acting on the user's actions: a game of responding to each other. Manninen [Mann 01] captures this nicely in figure 3.1.

A problem with automatic behaviour is that it can go wrong if the right en-

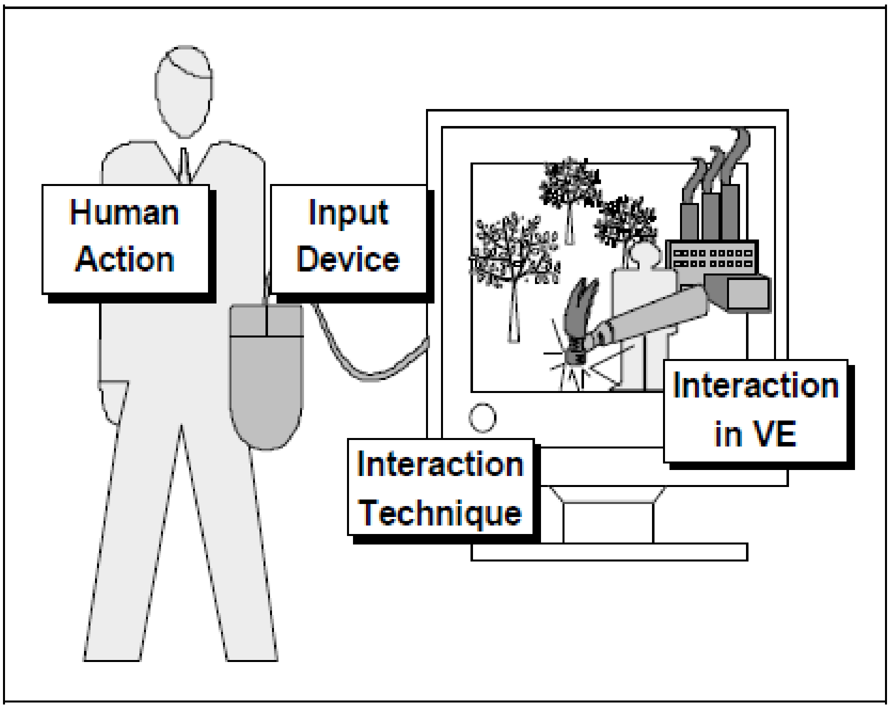


Figure 3.1: How users respond to the interface

vironmental cues are somehow missing. However, in practice it turns out that humans are creative enough to improvise and act appropriately at the required moment.
An additional benefit of taking environmental aspects into account is that it allows for the offloading of part of the cognitive burden onto the environment. This is why tools are of such great value: they facilitate a lower cognitive load, thereby making goal accomplishment easier [Dijk 08].

From an embodied embedded usability perspective, an application should ideally not draw any attention to itself. It should be so seamlessly integrated with the environment that the user is not consciously aware that he is using the application to achieve his goal. His focus should be on the goal itself, and the application is simply an extension of oneself facilitating the achievement of the goal.
If the application's ease of use is compromised, the situation changes and the application itself becomes a goal. The user will be too much occupied with trying to get the application to work appropriately.

3.1.1 An embodied, embedded view on interaction

Currently, modelling requires experienced technical experts to create models which fulfil traditional requirements such as correctness, completeness and validity [Hopp 09]. However, such experts are few and relatively expensive. Most importantly, though, the people who have greatest interests in the final product and the best domain knowledge are the business people themselves. Therefore it is only a logical step to attempt to make modelling available to non-modelling experts. This does require a user-friendly, easy-to-learn and easy-to-use interactive modelling tool and a way to present the process as self-explanatory and easy to understand.

Facilitating such interaction systems and support for them requires a study of real-time interactions both between modellers and between the modellers and the environments / models. This is the point where modelling meets human-computer interaction.
Traditional modelling has mostly ignored the human factors it is now directly confronted with, such as individual perception and expression, motivation, collaboration, decision making, problem solving and communication [Hopp 09].
Moreover, modelling typically can be placed in a context, both the context of use and the context in which it is created. When taking these into account there are more 'products' rather than just the final model. Common understanding, consent and commitment between participants result from the collaborative activity they have engaged in [Hopp 05].
Ideally, these interactions could be studied and described through the interaction-oriented view embodied embedded usability has to offer.

Success in modelling can be achieved through active involvement of all team members. The key aspect to successfully describing what happens between participants during a modelling process is interaction. Each participant has to actively put forth his views so that interpretation and decision-making proceeds on a uniform level. The whole process is in fact a set of negotiations and consequent decisions.
The interactions are embedded in the modelling process and emerge without participants' conscious awareness. Therefore a study of such interactions requires focusing on subsets of the process, for instance asking two modellers to create a simple model and ask them to specifically speak out every thought and action they are about to perform. Such an approach can provide valuable information on the type of interactions used to create a model, which can then be used to optimise the modelling process and interactive modelling tools.

The model has to be validated along the way, at each decision point a short reflection moment should take place. The interaction process is guided by the actions of the modelling procedure, which in turn follow from the procedure's *rule set.* Each participant produces output, and other participants in turn react to that. From these interactions the collaborative process emerges, and viewing modelling as an exchange of inputs and outputs between participants can greatly improve insight in their interactions. Ultimately, insight may also lead to improved quality of interaction.

3.1.2 Goals and constraints for modelling

When a model is created, it is preferable that modellers fully understand and agree with each other from the very beginning. Before starting modelling, the participants should agree upon a rough final goal: what should the model represent and what aspects are important to include.

So far the following seven goals for modelling have been defined [Hopp 08b, Bomm 08]:

1. Creation goals are the deliverables expected to result from the process.
2. Grammar goals mainly concern traditional values like correctness of syntax.
3. Validation goals are about the level of agreement about models, both on personal level and community level.
4. Argumentation goals state why participants agree that a model is good.
5. Interpretation goals ensure that everyone understands the model in a more or less similar way.
6. Abstraction goals are about the level of abstraction to be maintained for the model.
7. Efficiency goals typically entail the cost-benefit ratio and what methodological and process choices to make.

The goal should also be the one and only thing the modellers focus on. The formal method used to create the model may not be the focus of attention, rather it should be the 'application' through which the model is facilitated.

In a gaming environment, the rules of play define both the main goal and how the modelling process should proceed. Rules of play are typically formulated so that they are easily understood by all targeted players, therefore modellers will no longer be distracted by side issues such as how certain constructions work or how a certain fact can be expressed. These things are all embedded in the rule set.

A rule set is essentially made up of constraints in the game environment, goals and game strategies. Even though a rule set can prescribe a certain course of action players have to follow, the players' own strategy emerges from their behaviour and cognitive interaction. Factors contributing to a specific strategy are social behaviour, the unique individual way of responding to others, the perception of the domain, the best way of representing it, and the decisions made along the way. This strategy is often unique to the modelling process taking place. It promises improvement in model quality because it has gradually emerged from the modellers' own unconscious autonomous behaviour, implying that the strategy reflects a way of working the modellers experience as easy and pleasant.

During the process of creating a model, it is necessary to define certain constraints first. These constraints apply to the modelling environment, the modelling process and the model itself. First of all, the modelling environment forms a constraint on the possible actions. The idea is to incorporate the context of environment directly in the modelling tool, to 'let the world be its own representation', as dictated by embodied embedded usability.

The modelling domain is basically a subset of the environment, an isolated part or aspect which is relevant for the model to be created. In the domain environmental factors can be encountered which pose direct constraints on the model. If a cue is not present in the domain, it cannot appear in the model as a fact.

A model itself is also heavily constrained, having to be expressed in a formal language and needing to be syntactically correct. Formalisation must also be clear and rationally supported, that is, by means of analytical thinking.

On the other hand, a good model also requires some freedom of thinking and acting. There must be room for being messy along the way as long as an innovative and resourceful solution results in the end. These characteristics are difficult to combine [Hopp 09].

A gaming environment is an ideal solution to integrate these seemingly contradicting factors by providing a rule set which has to be followed, while at the same time allowing complete freedom of thought. Only the actions are constrained by the rule set [Hopp 08a, Hopp 09].

Even though a formal language is necessary, from the player's point of view this need not be a complicated set of constructions. Rather it should mirror the constructs humans intuitively create. For instance, when people who have never modelled anything before are asked to draw an overview of the structure of their school library they will typically come up with a diagram consisting of blocks with arrows between them, as I found when I asked 5/6VWO pupils to draw such a diagram on a study information day. Such simple constructions should be the basis of a rule set and no additional, confusing semantics should be assigned to the building blocks of the model.

3.2 Interaction: the core of gaming

Interaction is a powerful mechanism used in many different contexts. Learning and engaging people in a workflow are some of the most obvious applications of interaction, in the form of a serious game. Actively using a modeller-centred, interaction-oriented view when modelling, however, can also work to clarify the structure of the development process and the team's activities. The game format is ultimately suited for this.

3.2.1 Vitamine G?

Gaming is good. A phrase that is being uttered more and more often in various contexts. Serious games for training and education are being employed large scale in military, business, and school environments. While the focus of these serious games differs from the collaborative modelling approach, there are a lot of things to learn as far as usability, playability, and motivational factors are concerned.

Serious gaming is an example of how an interaction-oriented approach can help people learn and be immersed in their work environment. A serious game can be defined as "*a mental contest, played with a computer in accordance with specific rules, that uses entertainment to further government or corporate training, education, health, public policy, and strategic communication objectives*" [Grei 07]. In contrast with entertainment games, serious games "consciously employ pedagogy to impart knowledge or skill through activities that educate or instruct" [Grei 07]. Learning through gaming involves an active learning paradigm, providing the student with interactive learning experiences presented in realistic contexts.

A serious game is successful if it is realistic, assesses the player, provides feedback and ensures that the player has learned what was intended. In the case of the modelling game, success is achieved when collaboration succeeds and results in a comprehensible, commonly agreed upon model, compliant to (formal) conceptual demands, and ready for some specific sort of implementation. Examples are analysis, input for knowledge systems, software generation, and constraining design space.

Humans acquire new knowledge more easily when it is stored in well-connected knowledge structures, or a set of relationships among facts and concepts. Recall of facts depends on the degree of automaticity the knowledge structure has acquired. Cognitive load will therefore be low if the latter is the case. If the material has not been structured appropriately, however, cognitive load will be high because all concepts will be considered discrete.
When presenting people with interactive experiences in learning, it forces them to ”think about, organise and use the information in ways that encourage active construction of meaning, help build lasting memories, and deepen the understanding of the material” [Grei 07].
Games support this learning process by providing visual, textual and auditory means for feedback, challenges, goals and other components. Games typically do not entirely rely on instruction manuals in order to engage the player: rules are deduced from playing rather than being told. A collaborative modelling environment should ideally have the same properties. It should provide basic functionality and participants should be able to manipulate objects and create new things in the level.
Another objective of the game is to allow modellers to gain experience and learn new things. A game facilitates this by requiring the player to master particular skills by advancing through many levels, each one slightly harder than the one before.

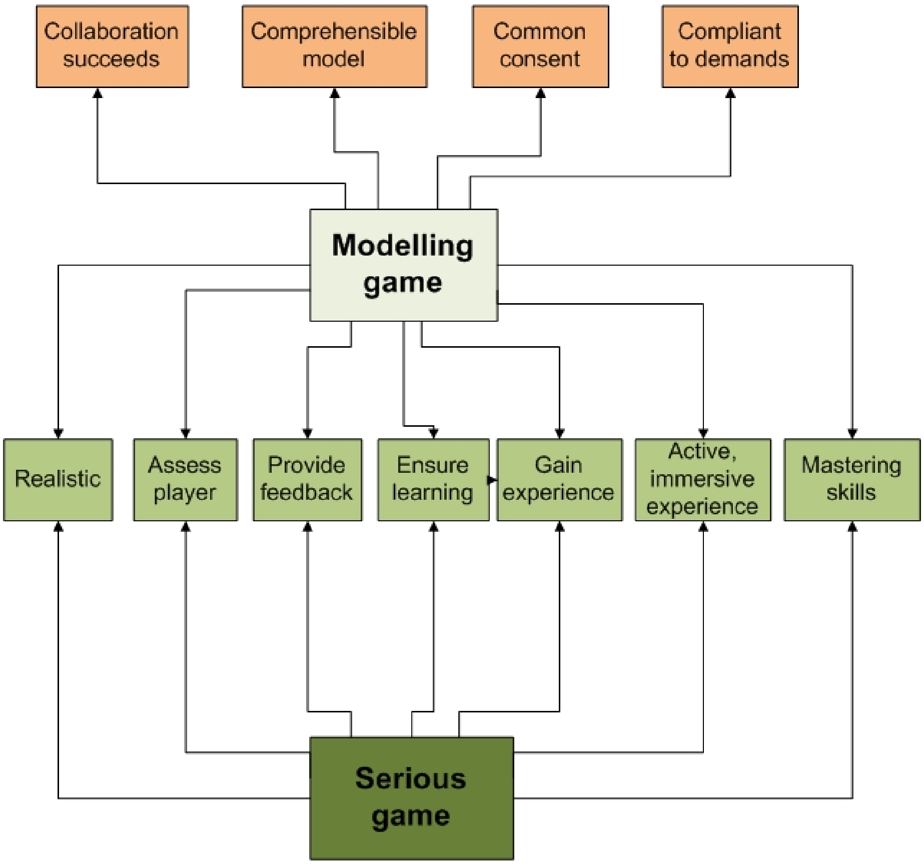


Figure 3.2: The overlap between serious game goals and modelling game goals

3.2.2 What games, modelling, software engineering and ME have in common

When analysing complex situations such as a software engineering design process, many points can be discovered where collaboration and communication can go wrong. The same trend can be seen in Global Software Development (GSD) [Ager 08]. This involves the employment of large, multi-skilled and low-cost workforces from economically developing countries. As GSD is practiced on the international level, it involves difficulties as far as distance is concerned. Geographical distance, cross-cultural distance and temporal distance can all provide problems when it comes to smooth collaboration.

Such problems have a close overlap with the modelling problems as described so far: difficulties in collaboration and communication because of differing interpretations, skills and backgrounds. However, there are also many benefits to GSD which clearly outweigh the difficulties. Agerfalk et al. [Ager 08] have identified several known and unknown benefits of GSD. Especially the more unknown benefits have an interesting overlap with the gaming approach to modelling.

Agerfalk et al. divide them in three categories:

1. Organisational benefits
2. Team benefits
3. Process and task benefits

Examining them in more detail shows just how close the overlap in benefits is to games for modelling.

1. Organisational benefits
   (a) Innovation and shared best practices
   Team members from different backgrounds collaborate, and organisations can take advantage of shared best practices and increased innovation. This is typically one of the objectives a modelling game tries to achieve. When working together online in the same environment, knowledge can easily be shared and stored for others to consult at a later point. All this bundled knowledge can result in innovative thinking.

   (b) Improved resource allocation
   Access to large, multi-skilled work forces allows for isolated expertise to be relocated in more strategic positions. Educating newcomers, or creating more effective skill-broadening tasks and teamwork. When no longer limited to physical space and place, modelling team members can be selected purely on skill, expertise and capacity, resulting in efficiently working teams.

2. Team benefits
   (a) Improved task modularisation
   Splitting the work across feature content into well-defined small modules allows a team to make decisions about each module independently, diminishing interdependencies and coordination costs. It is

useful and clarifying if a large and complex task is done is small, comprehensible steps. This improves product and process quality, increases common understanding and leads to decreased misinterpretations.

   (b) Reduced coordination costs
   Temporal distance allows for workers from different time zones to complement each other's work, the one working when the other is sleeping. This requires less coordination, since fewer people work at the same time. While this may not immediately be relevant to a nationally located modelling team, it could be useful to consider if multiple company establishments in different countries work on a project together.

   (c) Increased team autonomy
   Geographical distribution allows teams to keep their own working culture, necessary to keep the teams working efficiently. Teams will always be more motivated to work if their way of working suits them.

3. Process and task benefits

   (a) Formal record of communication
   Digital communication through channels such as a collaborative chatbox can be stored and kept as possible reference material.

   (b) Improved documentation
   Distributed teams must pay more attention to documentation since it is vital that other teams understand exactly what the modules they now have to work with do and how they function. Collaborative modelling does not happen with distributed teams, but a reward, eg. in the form of points, could be placed on the improvement of documentation.

   (c) Clearly defined process
   Again, when teams are distributed, the need for a formal process definition is much greater than when every team member has direct access to the others. However, when playing a digital modelling game, the modellers play within the formally defined structure of the game, implying a formal process definition.

For a graphical overview, see [3.3].

Collaborative modelling games thus can be seen as frameworks facilitating structured method engineering. It is an engaging collaborative forum in which both project management and development team can work together. As Wodehouse and Bradley have put it:

> ”Computer games provide a structured framework through which players must navigate, but are played each time in a different manner. Structuring the ways in which users navigate and move through the product design process by controlling the allowable inputs and output in a manner similar to a game could allow action and reflection to be configured for optimal decision-making.” [Wode 06].

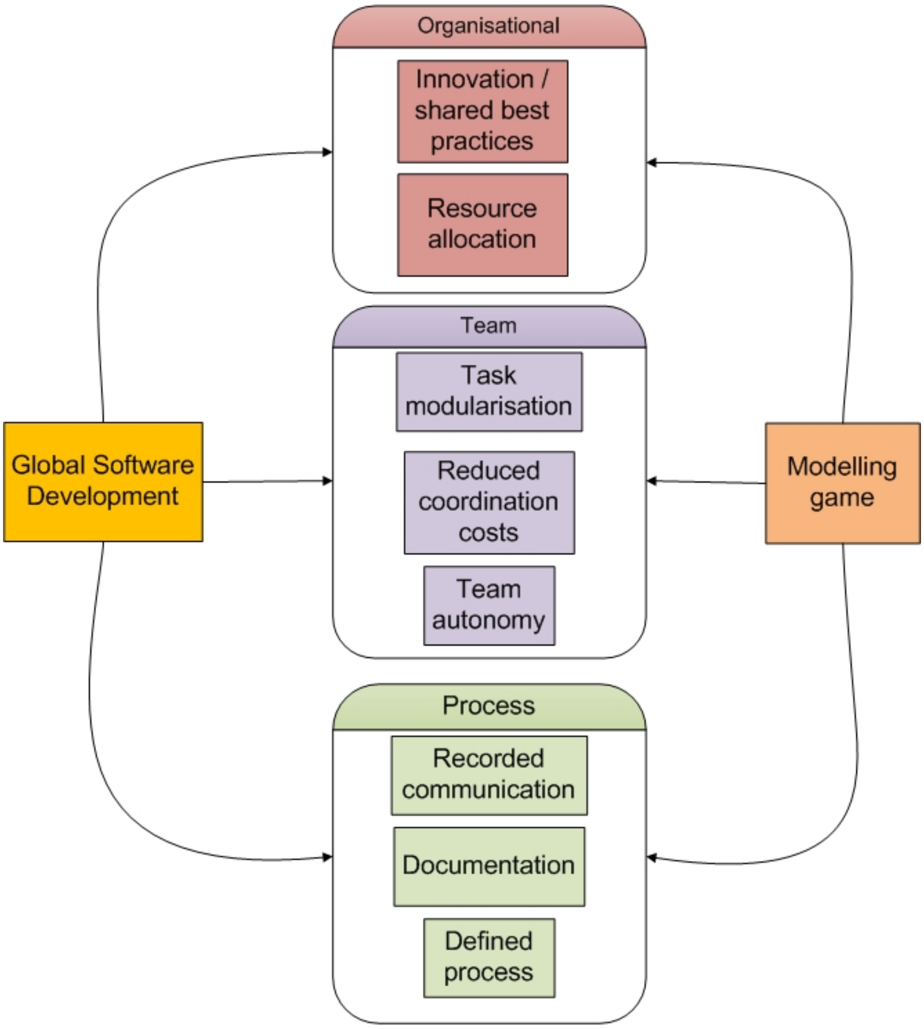


Figure 3.3: The overlapping benefits for GSD are the goals hoping to be achieved by the modelling game

The use of computer-based tools to integrate different aspects of the development process is not a new idea. However, systems to date are mostly document-centric, such as the LIRÉ system by Davis et al. [Davi 01]. The system was developed at Carnegie Mellon University and is based on an information flow analysis in order to figure out the team's workflow. While these systems try to use the power of digital information storage for quick retrieval, they generally do not address the characteristics of team interaction [Wode 06]. Team members from different backgrounds and disciplines working together, having developed expertise in certain areas or tools, do not necessarily work and communicate with the process as a whole in their minds. Interaction again plays a key role in determining the effectivity of the environment, for it is how users interact with the environment that determines how it is used and what output is produced. Shaping this process through the use of gaming techniques can be a very promising contribution indeed.

Companies are often information-rich environments which can have a confusing effect on team members as to what they are doing and where in the process they are. Games typically work by integrating a number of components such as a narrative element, addictive gameplay elements, or challenging puzzles. Literal forms of communication used within each component are a decisive factor in the overall effectivity of involving the player in the activity [Wode 06]. In a modelling game, all different stages and all information generated by members can be integrated and stored in a central repository. At the same time, support for new members can also be offered by more experienced team members. With different stages of the development process, different forms of play and thus different forms of communication and interaction are used.

Manninen [Mann 01] provides an interaction taxonomy of the main types of interaction found in current multimedia games 3.4. Both low-level and high-level interactions are represented, since cognitive reasoning and other psychological processes often require a combination of many types of interaction.

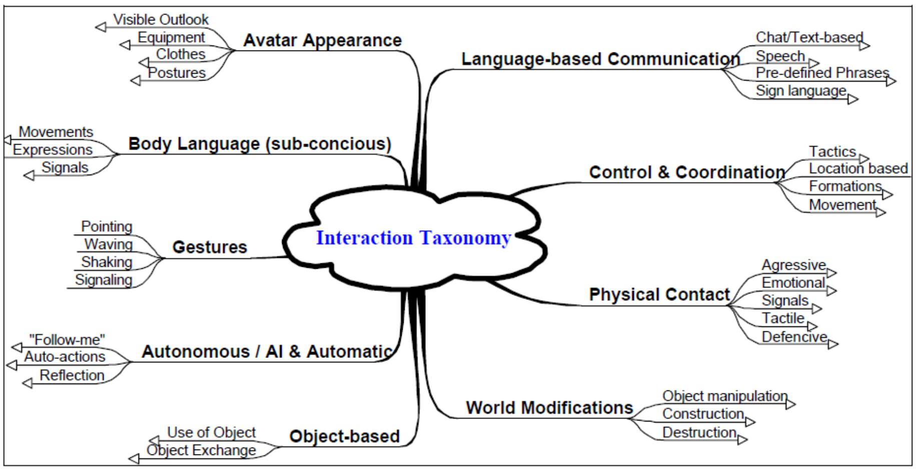


Figure 3.4: Interaction type taxonomy

This taxonomy can be used for interaction task analysis occurring inside the virtual environment. However, notice should be taken of the fact that there is not always a one-to-one mapping of the user's physical objectives and the acts of the virtual avatar. To go into the details of social theory applied to relationships between real-world incentives and virtual representation is unfortunately beyond the scope of this thesis. For my purposes, the taxonomy provides insight in the type of interactions to expect when studying, for instance, a modelling session.

In line with the idea from EEU that the world is its own representation and that environmental constraints provide constant feedback in guiding one's actions, one could view the interface as a continuous feedback loop. Reading body language and other subtle clues and signs plays a key role in direct human communication, even though we do not consciously realise it. These aspects of communication are lost when interacting mainly through a game interface. Yet gameplay and context are still powerful enough to fully engage players.
It is important to realise that, even though games in this context are only a framework for facilitating method engineering, gameplay and content are of equal importance when attempting to design a successful game. A player must not simply be "offered chunks of 'fun' play as a carrot to endure 'tasks' which are perceived as boring" [Wode 06].

3.3 Games: educational versus entertainment

There are many educational games on the market, but they simply do not have the appeal that huge recreational games have. What do 'games' have that educational games don't have?
When playing a typical educational game, a few things immediately come to mind. The goal is very clear: 'complete the sums', or 'read the words'. But apart from that, nothing truly exciting happens. The player solves a problem, scores a point and the next problem is presented. In some games points are disguised as, for instance, gaining home runs or earning cookies. But still, the graphics and gaming environment are not very exciting. There is no underlying story to appeal to the player and trigger his imagination. Players do not truly get absorbed in the games.
Motivation for such games comes purely from the achievement perspective: the will to improve one's score. Apart from that, the emphasis is too clearly on learning, and that is generally not what people like.

More differences can be observed when comparing the ways of marketing the different game genres.
Educational games are typically marketed by emphasising the following traits:

- Challenge
  Provide children with a fun way to improve their math skills. With single and multiplayer modes to enhance competition.

- Nice layout

- Accessibility
  Everyone can play it, so involve the whole family, young and old alike, because the game is good for all.

- Self-improvement
  Scientific research shows that mental health can be improved with the right brain training.

Popular fantasy games aimed at reaching huge audiences emphasise completely different aspects:

- Experience
  There is a fantastic, mythical world, explore every inch of it.
- Immersion
  Become a part of the world and rewrite history by being some great hero.
- Emotions
  Tension-filled, unforgivable battles, dangerous and suspenseful action.
- Mastery
  Make decisions as the industry director, predict outcomes of battles.
- Challenge
  Overcome monsters and puzzles requiring fast reactions with your state-of-the-art gear.

From this small comparison the conclusion can immediately be drawn that entertainment games employ many of the universal concepts mentioned in the previous chapter that make a game *fun* [2.5]. Worlds that appeal to people's imagination, things they do not get to see and do in real life. When designing an entertainment game, the only limit to the game world is the designer's imagination.
Educational games stick much more to reality, they market themselves with realistic, educational concepts which do not immediately trigger a person's imagination. Moreover, educational games often target skills that players are not necessarily good at, which means they have to invest more effort in successful completion of the game. Entertainment games call for skills that naturally appeals to humans, such as problem solving, collaboration and experience of intense emotions. And even an entertainment game becomes frustrating if the player cannot succeed.
Besides these, budget issues can also play a role in how elaborately the game world interface can be developed.

To summarise, games are a good medium for facilitating serious purposes such as education or business processes. Moving away from the 'everyday tasks' is of course impossible. However, if such a game is ever seriously to appeal to players, more emphasis has to be put on the fun aspects such as a good story, a fantastic game world and emotional experiences.

3.4 HCI aspects for playable modelling games

Method engineering in the form of a game calls for usability studies. When desiging a modelling game and evaluating its playability, special heuristics should be used specifically aimed at games. While there are many overlaps with traditional software evaluation heuristics, there are differences that have to be accounted for. For example, a software application is typically easy to use, easy to learn and easy to master. A game, on the contrary, should be easy to play but difficult to master. Moreover, aspects such as a storyline and the ability to experiment with and progressively develop game skills must also be included. It is therefore of no use to consider things such as *number of errors*.

For the evaluation of the modelling game, I will base the evaluation heuristics on three different aspects. Firstly, the basic indicators for the expression of positive and negative affect and learnability, as I have personally found them to be most useful during my own previously conducted usability studies. Secondly, I will use the *Heuristics for Evaluating Playability (HEP)* as described by [Desu 04]. Desurvire et al. indicate that the heuristics are best used during the initial stages of game design. Finally, the collaborative aspects have to be considered. Since collaboration during modelling is a primary goal of the gaming approach but has never been used in this way before, a set of criteria to assess collaboration has to be constructed.

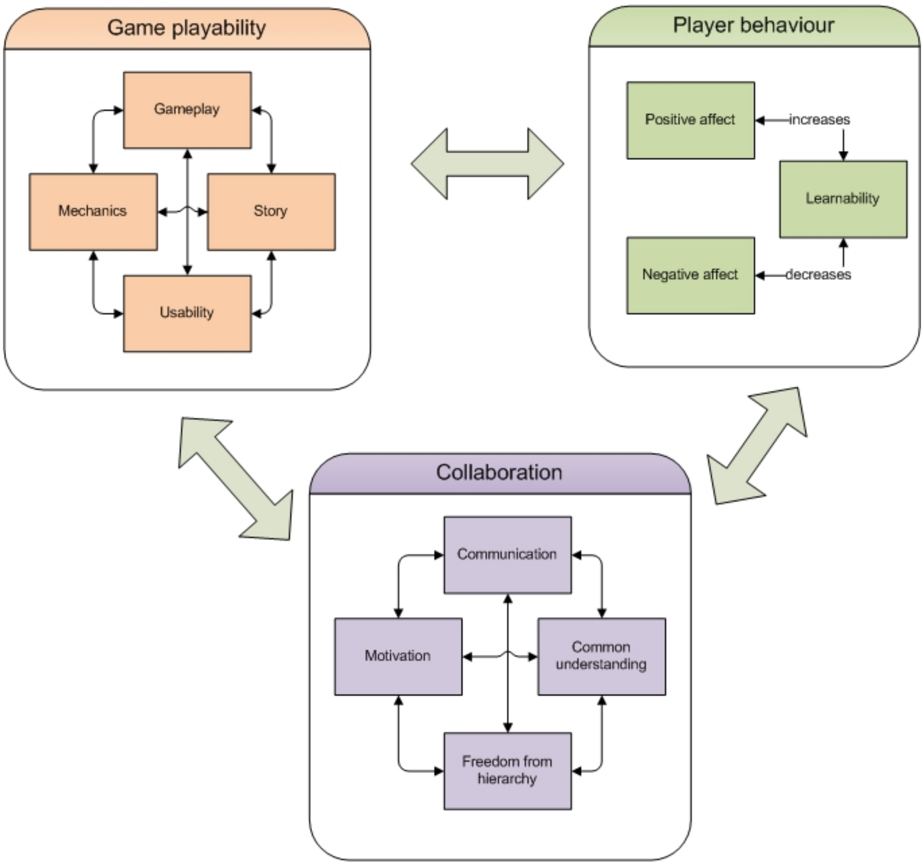


Figure 3.5: Modelling game success is centred around three main concepts.

3.4.1 Basic Player Mood Indicators

The following is a list of indicators for assessing the player's mood while playing the modelling game.

1. Positive affect:
    (a) Player's physical expressions:
        i. Verbal mentions of satisfaction, fascination or security
        ii. Facial/bodily expressions of satisfaction, fascination or security
        iii. Behavioural expressions of ease, fascination or security
    (b) Expressions from player's actions:
        i. Fast and easy progress
        ii. Smooth task completion
        iii. Immersion in game world
2. Negative affect:
    (a) Player's physical expressions:
        i. Verbal mentions of explicit frustration, boredom or insecurity
        ii. Facial/bodily expressions of frustration, boredom or insecurity
        iii. Behavioural expressions of difficulty, boredom or insecurity
    (b) Expressions from player's actions:
        i. Slow progress
        ii. Difficult task completion
        iii. No immersion in game world

As far as learnability is concerned, the following aspects concerning clues and specific instructions have to be considered. In the ideal situation, all clues are hidden in the gaming environment. Any clues or instructions asked for and given indicate a shortcoming in the game's balance between guidance and freedom of action.

1. Learnability:
    (a) Number of explicit instructions given
    (b) Kind of instructions given
        - Help with basic actions
        - Clarification of task
        - Correction after completion
        - Reassurance
        - Intervention
    (c) Number of clues given
    (d) Kind of clues given
        - Speed up task
        - Help with task completion
    (e) Number of questions asked

(f) Kind of questions asked

- Ask confirmation
- Request clarification of task
- Request help with basic action

3.4.2 Heuristics for Evaluating Playability

The observations made by considering the previous indicators will be given meaning and context by integrating them with the HEP by [Desu 04]. The following tables are literally taken from [Desu 04].

Heuristic nr	Description
1	Player's fatigue is minimized by varying activities and pacing during game play.
2	Provide consistency between the game elements and the overarching setting and story to suspend disbelief.
3	Provide clear goals, present overriding goal early as well as short-term goals throughout play.
4	There is an interesting and absorbing tutorial that mimics game play.
5	The game is enjoyable to replay.
6	Game play should be balanced with multiple ways to win.
7	Player is taught skills early that you expect the players to use later, or right before the new skill is needed.
8	Players discover the story as part of game play.
9	Even if the game cannot be modeless, it should be perceived as modeless.
10	The game is fun for the Player first, the designer second and the computer third. That is, if the non-expert player's experience isn't put first, excellent game mechanics and graphics programming triumphs are meaningless.
11	Player should not experience being penalized repetitively for the same failure.
12	Player's should perceive a sense of control and impact onto the game world. The game world reacts to the player and remembers their passage through it. Changes the player makes in the game world are persistent and noticeable if they back-track to where they have been before.
13	The first player action is painfully obvious and should result in immediate positive feedback.
14	The game should give rewards that immerse the player more deeply in the game by increasing their capabilities (power-up), and expanding their ability to customize.
15	Pace the game to apply pressure but not frustrate the player. Vary the difficulty level so that the player has greater challenge as they develop mastery. Easy to learn, hard to master.
16	Challenges are positive game experiences, rather than a negative experience (results in their wanting to play more, rather than quitting).

Table 3.1: Game Play HEP

Heuristic nr	Description
1	Player understands the story line as a single consistent vision.
2	Player is interested in the story line. The story experience relates to their real life and grabs their interest.
3	The Player spends time thinking about possible story outcomes.
4	The Player feels as though the world is going on whether their character is there or not.
5	The Player has a sense of control over their character and is able to use tactics and strategies.
6	Player experiences fairness of outcomes.
7	The game transports the player into a level of personal involvement emotionally (e.g., scare, threat, thrill, reward, punishment) and viscerally (e.g., sounds of environment).
8	Player is interested in the characters because (1) they are like me; (2) they are interesting to me, (3) the characters develop as action occurs.

Table 3.2: Game Story HEP

Heuristic nr	Description
1	Game should react in a consistent, challenging, and exciting way to the player's actions (e.g., appropriate music with the action).
2	Make effects of the Artificial Intelligence (AI) clearly visible to the player by ensuring they are consistent with the player's reasonable expectations of the AI actor.
3	A player should always be able to identify their score/status and goal in the game.
4	Mechanics/controller actions have consistently mapped and learnable responses.
5	Shorten the learning curve by following the trends set by the gaming industry to meet user's expectations.
6	Controls should be intuitive, and mapped in a natural way; they should be customizable and default to industry standard settings.
7	Player should be given controls that are basic enough to learn quickly yet expandable for advanced options.

Table 3.3: Game Mechanics HEP

Heuristic nr	**Description**
1	Provide immediate feedback for user actions.
2	The Player can easily turn the game off and on, and be able to save games in different states.
3	The Player experiences the user interface as consistent (in control, color, typography, and dialog design) but the game play is varied.
4	The Player should experience the menu as a part of the game.
5	Upon initially turning the game on the Player has enough information to get started to play.
6	Players should be given context sensitive help while playing so that they do not get stuck or have to rely on a manual.
7	Sounds from the game provide meaningful feedback or stir a particular emotion.
8	Players do not need to use a manual to play game.
9	The interface should be as non-intrusive to the Player as possible.
10	Make the menu layers well-organized and minimalist to the extent the menu options are intuitive.
11	Get the player involved quickly and easily with tutorials and/or progressive or adjustable difficulty levels.
12	Art should be recognizable to player, and speak to its function.

Table 3.4: Game Usability HEP

3.4.3 Assessing the collaborative aspects

Collaboration added to the modelling process creates a wealth of by-products other than the final model. Through collaboration model quality improvement has to take place, so it is of vital importance to be able to assess if the collaboration process runs smoothly.
To begin this analysis, I will first examine important factors influencing or resulting from collaboration, as described in [Hopp 09, Bord 99].

- Common understanding
  - Consent
  - Commitment created among participants
  - Equality of contribution (relatively speaking)
- Communication
  - Real-time and offline interaction
    - participant-participant
    - partipant-model
    - participant-modelling environment
  - Negotiation
  - Decision making
  - Active putting forth of views / spontaneous initiative-taking
  - Problem solving
  - Knowledge sharing
- Motivation
- Hierarchy-free organisation
  - Freedom of acting and thinking
  - Goal-oriented behaviour
  - Competition
  - Individual perception and expression
  - Role distinction according to specialisation

Common understanding, communication and motivation are all factors which, if the collaboration is successful, should improve. A hierarchy-free organisation allows for more autonomous input and creative thinking, thereby greatly contributing to successful collaboration.
The following tool is a collaboration assessment tool, taken from [Bord 99] and adapted to include the above mentioned factors.

Assessment factor	**Description**
Communication	The collaboration has open and clear communication. There is an established process for communication between meetings, both real-time and offline.
Common understanding	Members understand and are open towards each other's perspectives, they are committed to the project and contribute equally according to level.
Sustainability	The collaboration has a plan for sustaining membership and resources for the whole project duration. This involves membership guidelines relating to terms of office and replacement of members
Reviewing	The collaboration has obtained information to establish its goals and the collaboration continues to collect data to measure goal achievement.
Working culture	The working environment stimulates autonomous decision making, negotiation, active participation, problem solving and knowledge sharing.
Organisation structure	Members must take on a role determined by their specialities rather than their position within the company. Competition must be allowed for.
Motivation	The collaboration provides sufficient challenge and members help and motivate one another.
Resources	The collaboration has access to needed resources. Resources refer to four types of capital: environmental, inkind, financial, and human.
Catalysts	The collaboration was started because of existing problem(s) or the reason(s) for collaboration to exist required a comprehensive approach.
Policies/Laws/Regulations	The collaboration has changed policies, laws, and/or regulations that allow the collaboration to function effectively.
Connectedness	Members of this collaboration are connected and have established informal and formal communication networks at all levels.
Leadership	The leadership facilitates and supports team building, and capitalizes upon diversity and individual, group and organizational strengths.
Community Development	This community was mobilized to address important issues. There is a communication system and formal information channels that permit the exploration of issues, goals and objectives.
Understanding Community	The collaboration understands the community, including its people, cultures, values and habits.

Table 3.5: Collaboration Assessment Tool

3.5 Summary

In this chapter, I have explained the viability of viewing modelling in terms of interactions between participants, to allow for more insight in and understanding of modelling actions.
It is also important to chart the modelling goals before attempting to draw up a ruleset for a modelling game. Charting the goals provides insight in the relations between different constraints (on the model, the process, or the environment), on which the rules can in turn be based.

The development process and team activities can be viewed in an interaction-oriented way by means of a game, to create insight and understanding through an interactive experience which has proven to work extremely well in learning [Grei 07]. While the use of computer-based tools to integrate the whole development process is not a new idea, doing this with the focus on interaction and gameplay is new. It integrates all the benefits of serious gaming into modelling and product development.
Gaming provides many benefits for the development team, which show a curious overlap with GSD [Ager 08].
Most current educational games are of good intention, but the learning aspect is still too prominent. Educational games would do well to move away from the 'carrot of fun', and provide a much more immersive environment with focus on both gameplay *and* content.

Finally, evaluation heuristics from several disciplines have been integrated to form a coherent approach for evaluating the success of a modelling game. It focuses both on product outcome, process outcome and participant behaviour.

Chapter 4

Game aspects and Aquima Studio

In this chapter I will describe the Everest way of development and how a modelling game can support and improve this process.
There have already been some initiatives towards innovation by means of game aspects within Everest. The first step has been work on a collaborative chat client in Aquima Studio by previous graduate students. Furthermore, new gadgets in the latest Aquima version also tend towards enabling a more collaborative and easy way of working, such as the drawing tool for entity-relation diagrams.
In the second part I will explore tried and tested business games, how they have enhanced the work process and what aspects have been found to be essential for success. These aspects will be integrated in the eventual Aquima game design.

4.1 The Aquima approach

Aquima Studio represents business knowledge in a structured, visually appealing way, and does not require knowledge of programming in order to perform maintenance operations on business knowledge. Flexibility is possible through localisation, tailored authorisation and context-sensitive support.
The core of Aquima's strength is in the fact that they see the functional model as the solution, instead of using it as a specification to base program code on. This means that the application is directly specified and no lines of code are programmed.

Aquima Studio in essence models page flows. All entities, attributes, relations and processes are stored in a central repository. These are loaded into memory and are then executed. This surpasses the need for code generation, which allows for the separation of program logic and business logic. This is a key aspect of Aquima. When the business logic is changed, the application does not have to be structurally changed. The initial model is turned into an XML format, which is then interpreted and executed by separate applications: Aquima Interactions or Aquima Documents. These can be used as a simulation and testing environment before applications are launched. If changes are necessary, the model in Studio can be easily adapted. If satisfactory, it can be published in the real

production environment.
Interpretation happens step by step, and runs through the entire process. This is why the process has to be modelled in its entirity in advance.
Core Aquima functionality differs with the role the individual takes. For business engineers, core functionality will constitute of producing the specified application. However, for people on the maintenance side and project leaders, the core functionality is typically to gain more insight in the development process and also in the application itself. How do different components communicate, what is their relationship, what are the dependencies? Stef Demeester states that insight is a point in which Aquima falls short [D].

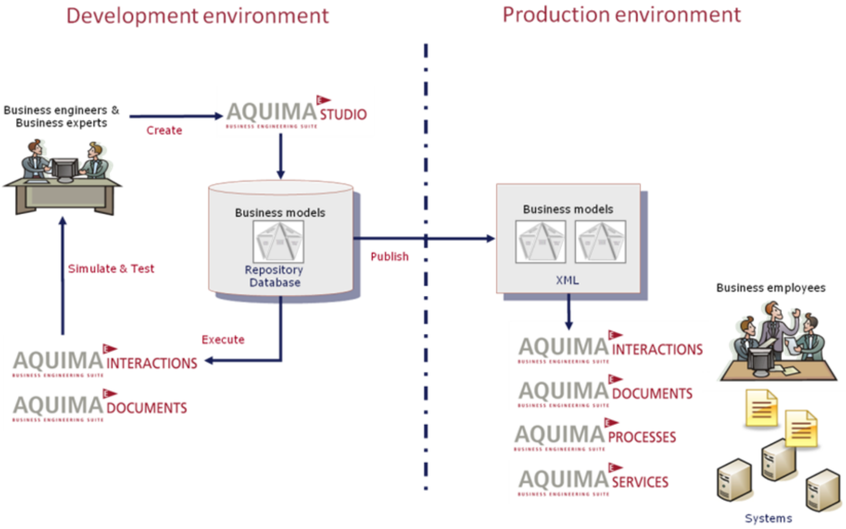


Figure 4.1: Overview of the Aquima workflow

4.1.1 Aquima Studio in terms of Method Engineering

In this component, the actual modelling of the application takes place, so this is also where the implementation of gaming techniques for collaboration is most useful.

In order to create a full understanding of the working of Aquima Studio, it is necessary to analyse the process by using the interaction-oriented, modeller-centred perspective described in chapter 3. I will use the modelling goals as defined by Van Bommel et al. [Bomm 08].

- *Creation goals*
  - Model
  - Application
  - Collaboration
  - Personal knowledge and experience gain

  - Insight in development process
  - Improvement of process quality
  - Improvement of product quality

- *Grammar goals*
  - Syntax should be correct
  - Application should understand and execute commands given by modeller

- *Validation goals*
  - Common understanding and interpretation of case
  - Consent on solution

    Communication with client after models and application have been completed is largely lacking on client-modeller level. Business engineers indicate that there is too little communication with the client. The same holds for communication with other team members while working on model. Business engineers indicate that modellers work on a model alone, and are somehow reluctant to call for help when things get complicated.

- *Argumentation goals*
  - There are no objective criteria for when a model is finished or when everyone agrees that the model is 'good'. A possible solution to this problem has been proposed by Stef Demeester, one of the business engineers I spoke to. There should be reviewing after every module is considered 'finished', and a comparison should be made with an ideal scenario test case, written in advance.

- *Interpretation goals*
  - At present there are no specific mechanisms to ensure that everyone understands the model in a similar way. Since the client is never directly confronted with the model, this is only an issue for the modellers, who typically work on it alone or in a small team. In the latter case team members collaborate daily and are likely to speak the same language.

    However, as Rob Gense and Stef Demeester [D] indicate, there should be more focus on insight. A small step in that direction can be made on the level of model interpretation. Continuous reviewing of finished models and having model creators explain them to other team members can be of help for unambiguous interpretation.

- *Abstraction goals*
  - There are no currently known regulations for establishing the level of abstraction to be maintained for the model. This level is implicitly established as the model develops.

- *Efficiency goals*

  - There are no specified efficiency goals. Models are created based on what the client wants. As for the methodology, there is no standard methodology everyone follows. The modelling language used depends on the modellers' knowledge and whether they show their client things or not. Most of the times blocks and arrows are used, either on paper or by means of Visio diagrams.
    However, an effort is currently being made to structure this process more tightly. A very simple modelling tool is already available in Aquima Studio, allowing for simple entities, attributes and relationships to be visualised. Building forth on that, an advanced modelling tool is currently being developed which is expected to be delivered in late 2009. This tool aims to make Visio and paper drawings superfluous, by providing an integrated modelling tool with an obvious added value being that the drawing is directly executable.

4.2 Development within Everest

4.2.1 Project roles

There are several roles to be distinguished within the project development team. The roles are taken by two parties: Everest and the client.

Everest

- Project/teamleader
  This is the person who is ultimately responsible for the entire project, for what will be delivered.

- Technical architect
  Checks if the to be built application will fit in technically with the client's architecture and infrastructure, he is ultimately responsible for the technical side of the project.

- Technical engineer
  This is the programmer, who has to program things.

- Business architect
  The business side equivalent of the technical architect. He decides what processes will be used and what models are to be created, and is ultimately responsible for the business part.

- Business engineer
  This is the person who models and communicates with the client and the business. Even though it is not an official distinction, business engineers can be viewed in two categories: the requirements business engineer *is the one who interacts with the domain expert to gather and prioritise the requirements, and the* modeller business engineer*, who works on the actual implementation in Aquima Studio.*

Client

- Project leader/sponsor
  This is usually the project leader from the client side who plays the part of sponsor part-time.

- Enterprise architect
  Checks if the application will fit within the currently existing architecture.

- Business analyst
  Decides which processes and services the client would like to have modelled.

- Business expert
  The person who knows the business inside out.

- System administrator/infrastructure
  Responsible for proper maintenance of the system.

- Testers
  Test if the functional specification matches the delivered product. They are often supplied by the client.

There is one last relevant role to be mentioned, as brought up by Stef Demeester and Rob Gense. This role, however, is not yet being executed in practice. It is the role of the *reviewer*, who comments on the quality other team members produce. Each team member can take on the role of reviewer, or even the team as a whole can review their own work.

4.2.2 Everest way of working

The development process starts with a requirements engineering phase. Once the business engineers have a clear vision of what the client wants, they create a conceptual model. Sometimes this can be a previously created model, which can be adjusted to suit the situation. When no such model is available, a new model is created from scratch. Modelling is done on paper by hand, or in tools such as Microsoft Visio. When the model is finished, the implementation phase in Aquima Studio starts.

In a typical Everest development team the role involvement is as follows. The requirements engineering phase involves the domain expert and the requirements business engineers. The creation of a conceptual model is the work of the requirements business engineer, which is largely a solitary process. Questions are not often asked, and collaboration with more experienced people is rare. After the model has been created, the modeller business engineers start the implementation of the model in Aquima Studio. Both the domain expert and the requirements business engineers have nothing to do with Aquima Studio.
The team does strive to involve the domain expert more actively in the entire process. For this they use the Agile Development methodology, which comes down to working in short iterations and showing progress in the form of a working prototype at the end of each iteration. Communication is direct and preferably physical rather than by means of telephones or chat clients. The domain expert, however, is not a part of the development team.

Despite these efforts, communication with the domain expert is still largely asynchronous. There are peaks, but then periods can follow in which there is no communication at all. Everest business engineers also indicate that there should be more, frequent communication. The problem is often limited time, both from the domain expert as the business engineers.

Very roughly, the modelling process can be described as follows:
The requirements engineering phase is the line of communication between the client and the requirements business engineers. After that, the requirements business engineers collaborate with the modeller business engineers, who implement everything in Aquima Studio.
The client has nothing to do with Studio, which should probably stay that way since Studio is a much too technical environment for a client. The client should, however, be made a more active participant of the project team.

SCRUM

Within Everest there are efforts towards the employment of a new way of working, which is currently largely being used for internal projects. It concerns an agile, cyclical, flexible way of working, with project phases being development cycles typically lasting 2 - 4 weeks after which booked results are visible and can be evaluated. During each cycle, also knows as a *Sprint*, there are daily Scrum meetings and regular workshops [Jord 08]. Before each sprint, all tasks for the coming cycle are discussed and divided. A workshop allows stakeholders to see and evaluate functionality, and the Scrum meetings are meant to provide insight in the team's progress. In this way automation happens in a comprehensible, step-by-step way which also allows for the creation of a solid basis for the resulting application.

4.2.3 Business Engineering

Business engineering efforts tend towards putting the business in control of the definition of their own processes, concepts and business rules. The keywords are iterative and incremental working, allowing for simulation and testing after each development cycle. This should lead to higher quality of deliverables and a shorter time-to-market, since the application's integration with the implementation environment is gradual. The aim is also to include more business logic in commercial applications.
Everest defines business engineering in an internal company document [Jord 08] as follows:

> "Business Engineering is a combination of smart project management, tooling and people, where the customer and the business of the customer play the centre role. It makes business knowledge and processes explicit, transparent and easily adjustable. Customers can thus rapidly adapt and improve their business processes and improve efficiencies and customer services."

Everest business engineering takes the middle road between top-down design (Model Driven Architecturen, or MDA) and bottom-up design (Application

Driven Modelling, or ADM), using the best of both worlds. Aquima Studio currently supports bottom-up design by allowing for the application to be directly specifiable and executable. Support for top-down design is being worked on by means of the integrated modelling tool. Also for workshops Aquima Studio provides sufficient support, since stakeholders can immediately view what has been done and changes can be made on the fly.

It should be noted, though, that this form of business engineering is still mostly used in internal projects. It is difficult to apply this method when working with large business clients, because Everest cannot specify precisely in advance what the client will eventually get. Without the promise on paper, companies are very reluctant to spend money on the project, even though the traditional waterfall way of working cannot guarantee success either. It merely provides a foothold for the business.

If the business has a leading role in the project and they have also issued the order, they will typically have a decisive role in how the development process will take shape. Apart from that, Everest is also keen on cooperating with the client and adhering to their ways, rather than imposing their own way of working, especially because the main clients are financial institutions who have strict sets of rules to adhere to themselves [D].

However, the people I spoke to strongly believe in the cyclical, flexible nature of business engineering, and state that in a world of rapidly developing technologies, desires and changing specifications, waterfall development is near impossible. Clients often sabotage their own project successes by trying to state in advance what they will need in three years' time.

An overview of the work process with its entire role involvement is given in figure [4.2].

4.2.4 Work process issues

There are some issues for improvement in every work process. To begin with, there is not a lot of knowledge sharing. The process of documenting processes, achievements and best practices and making them available to everyone within the company was being stimulated by means of the Everest community website. However, this attempt was not successful due to a few people being highly motivated and most others not using it at all.

Knowledge sharing may be stimulated if there is a reward for it. For example, when considering the MMORPG *Lineage*, which was a failure in the West but a great success in Korea, an examination by Huhh and Park [Huhh 05] reveals that the game's motivation was largely based on trading markets, which became an economic motive not only inside the game but also outside it, trading game items for money and becoming a way of making a living for many people.

The idea of a trading market parallel for knowledge sharing, with rewards based on experience points, status gain or both, could work as a stimulating factor for people to share their best practices and thereby benefit the whole organisation. However, knowledge sharing should not happen in the working interface of Studio itself. For many people who do not deal with Studio daily the interface may appear to technical. This was a concern voiced by most of the business engineers I spoke to. People should not feel hampered or scared to access a knowledge

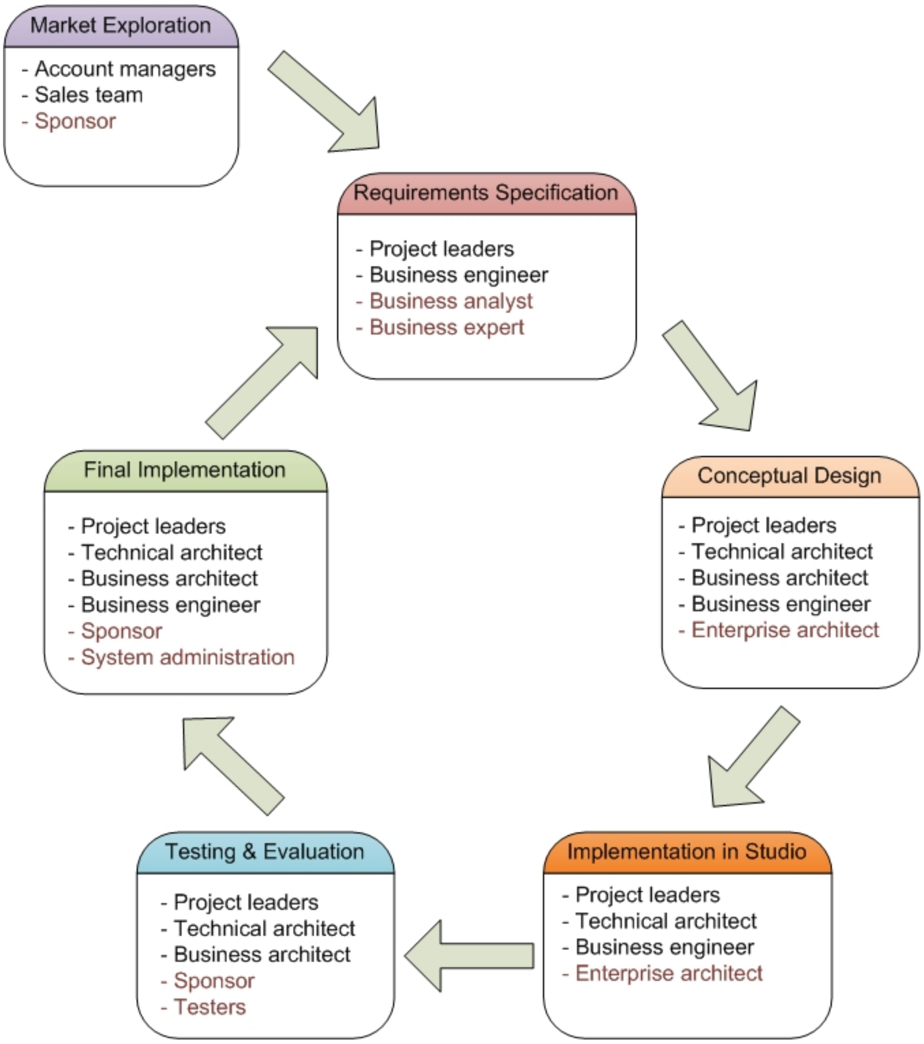


Figure 4.2: Everest product development process model with all involved roles.

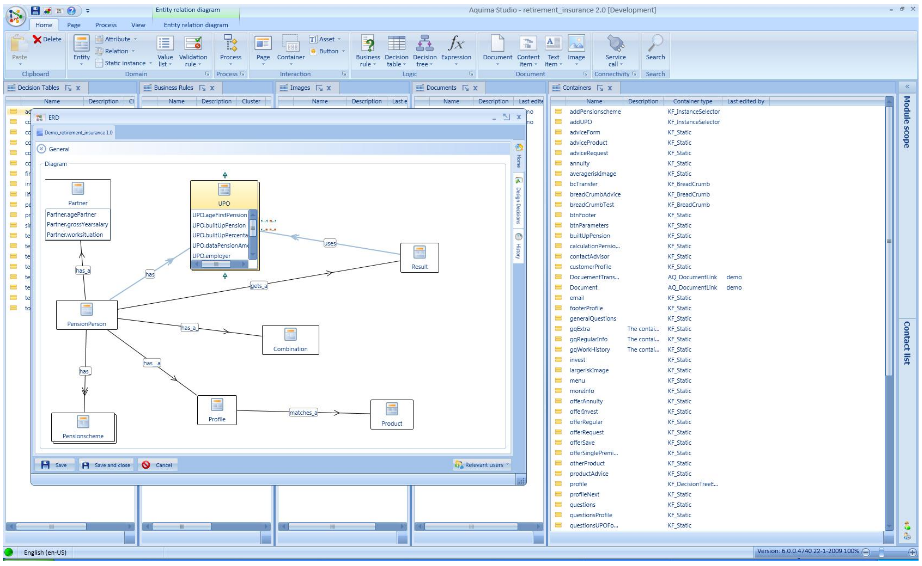
source.
Therefore a possible solution could be to build a more abstract interface around Studio, which can be viewed separately from the implementation interface. Clients and other stakeholders could go in there, together with the requirements business engineer, and things could get documented in that interface.
The idea is a lot like the Business View idea already covered by former graduates. The main idea was that business people would work in a separate interface, which would have a direct link to Aquima Studio for the modeller business engineers. The Business View as described by them allows for people to work on each process phase in a familiar-looking, Microsoft Office-like interface with a text editor or modelling tool depending on the task.

Everest usually scores well as far as client satisfaction is concerned. Clients are satisfied with what they have, but there are always more demands in today's ever-changing business environment. However, satisfaction differs with each individual client. It is very typical that only the manager is consulted about the degree of satisfaction for the application. Managers are also the only ones who decide whether there is a demand for the application, and Everest has a very convincing sales team. But people on the workfloor generally do not like change if their current product works well enough.
This illustrates the importance of product quality improvement, so that even if there is no very urgent need on the client's side, the people on their workfloor will still be happy to work with a new application. It must be obvious to them that Everest's application is of additional value.

4.3 Aquima: current status

There has been a lot of effort to improve collaboration in Aquima Studio already. Graduate students have worked on a chat client, which is currently being released for the first time, and is also currently being expanded to include a group chat function.
To make modelling easier, the beginnings of a drawing tool for modelling have been included. This drawing tool allows for a directly executable graphical model (UML, entity-relation diagram) to be created online, which can be stored for later use 4.3.
The syntax is rather simple. A single block represents an entity. A double block is a multivalued entity, meaning it can have several instances. An arrow denotes a relation. Dragging and dropping for drawing has very recently been implemented, the development of the modelling tool progresses very rapidly.

Furthermore, other graduate students have worked on the Business View layer on top of Aquima Studio as mentioned above.
All these aspects together are very promising if they can be integrated in the entire gaming approach.



Name	Description	Container type	Last edited by
addPensionscheme		KF_InstanceSelector	
addUPO		KF_InstanceSelector	
adviceForm		KF_Static	
adviceProduct		KF_Static	
adviceRequest		KF_Static	
annuity		KF_Static	
averageriskImage		KF_Static	
bcTransfer		KF_BreadCrumb	
breadCrumbAdvice		KF_BreadCrumb	
breadCrumbTest		KF_BreadCrumb	
btnFooter		KF_Static	
btnParameters		KF_Static	
builtUpPension		KF_Static	
calculationPensio...		KF_Static	
contactAdvisor		KF_Static	
customerProfile		KF_Static	
DocuementTrans...		AQ_DocumentLink	demo
Document		AQ_DocumentLink	demo
email		KF_Static	
footerProfile		KF_Static	
generalQuestions		KF_Static	
gqExtra	The contai...	KF_Static	
gqRegularInfo	The contai...	KF_Static	
gqWorkHistory	The contai...	KF_Static	
invest		KF_Static	
largeriskImage		KF_Static	
menu		KF_Static	
moreInfo		KF_Static	
offerAnnuity		KF_Static	
offerInvest		KF_Static	
offerRegular		KF_Static	
offerRequest		KF_Static	
offerSave		KF_Static	
offerSinglePremi...		KF_Static	
otherProduct		KF_Static	
productAdvice		KF_Static	
profile		KF_DecisionTreeE...	
profileNext		KF_Static	
questions		KF_Static	
questionsProfile		KF_Static	
questionsUPOFo...		KF_Static	

Figure 4.3: Example of an ER diagram being created in Aquima Studio.

How game aspects are already integrated in Aquima Studio

The chat tool meets several gaming demands. First of all it realises *interaction*. Players can actively converse, collaborate and discuss. This communication should improve common understanding and lower the threshold to speak to a person, for instance if he is in a different geographical location or simply a floor below. Moreover, the ability to post a message and view it whenever the recipient is available also makes it easier for people to make a comment and get a reply in a short time.
Chatting with colleagues about the game issues also contributes to *immersion* in the game. When working on something together and striving to finish it as well as possible creates a feeling of unity and works to stimulate people. *Emotions* play a role here as well, since they are unavoidably linked to immersion.
For another, *feedback* can also be given through the chat client. A player can voice his satisfaction or dissatisfaction about things, so that they can be immediately solved or stimulate other players.

The drawing tool forms part of the *game mechanics* and can certainly contribute a greater usability and *accessibility*. When making model creation more intuitive for people and if it saves them the trouble of drawing by hand first and then copying everything into Studio by hand, people will certainly embrace the added value of the tool. And as we have seen in chapter 2, greater accessibility and intuitiveness contribute to a good gaming experience since frustration decreases and a greater sense of *freedom* and *control* are felt [2.2.1]. Again we see *emotions* being called on.

The business view layer provides the *interface* through which players will be interacting with Aquima Studio. Though the idea of the business view layer is useful, I will design an interface myself to suit the needs of the Aquima game descibed in the next chapter 5.

4.4 What aspects are useful in Aquima?

During my interviews with business engineers Rob Gense and Stef Demeester, and with game design lecturer dr. Penny de Byl, some interesting suggestions and important points came up. This section summarises the most important and useful ideas for an Aquima Studio game.

4.4.1 General emphasis points

One of the key points to make work fun is to enhance the emotions involved, to make people feel positive affect while doing their daily chores. Emotions can certainly be enhanced through setting and environment, but in my opinion the focus should on the short term be mostly on enhancing emotions through interactions with team members, because a fantastic game environment for Aquima seems only realistic on the longer term.
The same goes for stimulating knowledge sharing. The previously mentioned mechanism of *trading markets* as in Lineage could serve to exchange knowledge, skill, and possibly also game objects such as tools or game-specific best practices in the Aquima game. The motive would not be economical benefits for the individual player directly, but status and experience gain in the game. The economical benefit will be visible on a higher level, by increasing production, revenues and profits for Everest as a whole. However, for all players participating in the trading market, the competitive element should be obvious.

The preference for a game perspective is a point of discussion. [Yee 04] mentions that goal oriented users like to see their avatar as a pawn with which to achieve their goal, thereby preferring a third-person perspective (3PP). On the other hand, relationship oriented users like to identify themselves with the avatar and therefore like a first-person perspective (1PP). Motivational differences are thus linked with a preference for perspective.
For achieving the objectives in the Aquima game, however, a 3PP sounds most promising, since it is the best way to provide overview and insight in the entire situation in the game world.

A successful game should include basic storylines that players must experience, since the story is vital to experience. During play, the system must constantly reflect on the player by means of providing game clues or system feedback, so that he can judge the progress he is making from his in-game achievements. This enables the player to relate the experience to the final goal.
There should be a strong focus on challenge, allowing people to manipulate and exercise control in the virtual world. The game should in the first place provide basic functionality, and based on what people do with it adaptations can be made. It is important to use Agile methods of development in the game, providing constant feedback, and allowing people to demonstrate what they have learned in the environment. The gaming set-up also allows Agile to be more easily integrated in the design process, since it is highly beneficial for game success.

When the system provides essential game clues and the player overlooks them, he should not be directly punished for his carelessness. However, he will have to notice some way or other, since game clues are often vital for successful progress. The system should provide further hints, or the player should find out from other players, for instance if he does not possess a certain resource or a specific bit of knowledge he should by now have obtained.
Furthermore, it is important to create responsibility in the game. This makes people play compulsively, since they feel responsibility towards their team mates and sense the game cannot progress without their individual participation.

Player involvement is another highly important factor to create a successful game. This goes for both the client and the IT developers. It is not a wise idea to allow the client into Aquima Studio, since the application is much too technical and the client will probably get scared off by it.
A possible solution, proposed by Stef Demeester, is to build parallel worlds on top of Studio and let people collaborate there, with other people of their 'own kind', who speak the same language. The world for domain experts, for example, should not be linked to Aquima Studio, whereas the worlds for technical experts and analysts should facilitate access to Studio so that decisions can be immediately implemented.
In these worlds, requirement 'evolution' and reviewing should take place. A wish was expressed by Stef Demeester to integrate requirements management in Aquima, for instance by means of the Issue Management System (IMS), a system already in use to comment on and keep track of project issues.
Communication between the different worlds has to occur, but specialised people are available for that, such as business engineers. Communication should be possible both synchronously and asynchronously. The use of an integrated chat client when all avatars are in the room together is useful, and is already being worked on. For offline communication, a solution is message posting to specific avatars. Communication has to be aimed specifically at the targeted person and feedback should be available all the time. This is a continuous process.
The whole team has to be involved in the whole development process. For instance, a message has to be sent around if something in the world has changed, and should also allow people to react to this change. Version control is important here, so that changes can be undone if the team cannot reach consensus over a decision.

The reviewing mechanism should provide an objective way to create a ranking of projects, performance-wise. Each review session should contribute to an overall project rating. For project components, an objective way of judging their quality has to be available. A way to do this is to write a test case in advance, representing the ideal situation which every team member has agreed upon, and use this as a measure of 'goodness'. Currently, test cases are already being used on the technical side, but not in Aquima Studio.

All project roles as mentioned above [4.2.1] have to be integrated in the final game. It is a good idea to allow for specialisation within the roles. Each specialisation must be rewarded accordingly, and a good balance between specialisations has to be ensured within the team by the team leader. It is not a good idea to use a single specialist for a certain tasks, but to have some newcomers on the team as well so that they can work with and learn from the specialist. This way they become backup experts, which are useful resources for the company.

The main task is often very badly defined, for instance clients say they ”want a mortgage management system”. This task has to be split into comprehensible quests and be formulated as such. Also the reward for completing it as far as experience points, level gain, status gain etc. are concerned should be mentioned. A list of these tasks can then be drawn up, and divided according to skill and preference within the team. A personal goal can also play a role: does someone want to learn from it, or do it to become more specialised? This can create optimal role and task division.

Finally, game playability must not be underestimated. The game should be usable and a pleasure to play. If usability and achievement of good collaboration result in better final products, success is within reach.

4.4.2 Options for score systems

A score system is necessary for tracking performance, both for the personal achievement record and in order to compare one's own performance with others. However, a score system in a business environment raises a few side issues. First of all, it may be the case that colleagues, who should be collaborating in harmony, start feeling the wrong kind of competitive feelings towards one another. This may result in them trying to inhibit each other's work in order to enhance their own performance. It should also not be possible that employees become so much focused on their individual score that they spend more time trying to increase it than on doing their tasks.
It is also possible that they figure out how scores are increased, resulting in them doing only those actions that increase scores.
Such actions would be detrimental to the work process and the entire team's performance. The score system should by no means be kept secret, but it should be carefully designed in such a way that it is not possible for players to increase their scores by means of easily repeatable actions.

A suggestion from Everest employee Rick Fleuren, technical engineer, is the use of an *achievement system* - based solution. An achievement system is used in for example Assassin's Creed, and works as follows: the player needs a minimum of eg. 1000 points to complete a level. So during the level, points have to be gathered, for instance if you perform a certain action extremely well. An example: if you kill all the guards, not just enough to get through, you gain 10 extra points. This forces the player to give some aspects of the game extra attention and to achieve optimal results, for otherwise progress is impossible.
Indirectly forcing people to do certain things extra well could work wonderfully in a business setting. The only lurking danger is that people figure out how points are gained. That is why the design has to enforce that it is not interesting for people to do such things, they have to be able to gain more points from just doing what they should do. Another solution is that a certain variety of tasks has to be performed in order to level up, not just that one thing that gains points. Team members' reviews can play a role too.

However, the score should only serve to compare project team performances, and work as a stimulating factor. The scores should not be considered in functional reviews of the employee's skills, because that can lead to skewed judgements.

Another important factor to consider for the score system are the many different roles included in the game. It may not be so that people who simply work with Aquima more often get high scores than people who have other functions, such as requirements gathering or serve as communicative line with the client.
A possible viable solution is to employ a rating/peer assessment system. The idea is to let all team members rate each other, and the validity and importance of this rating has to be weighed as objectively as possible according to the individual's level. People who have a higher level can assign weights to ratings from people who have lower levels. The weight then indicates how seriously people should take the given advice. Also, the higher your level, the more your opinion is worth, unless, of course, an even higher level person indicates otherwise.
However, in such a system true objectivity can never be achieved, so each individual will still have to employ a healthy dose of common sense when interpreting the advice.

4.4.3 Team formation

A typical phenomenon in multiplayer games is the formation of *guilds*, or groups of people coming together to form an organised whole to achieve a certain goal. Guilds and project teams have several aspects in common. They both have to be well-organised; members must know of each other what they are doing to avoid redundancy, leadership must be clearly defined, active participation and cooperation are required and skill levels within the team must be balanced.
Duchenaut et al. [Duch 07] have examined what essential factors allow a guild the best chances of survival. He came to the following results:

- Simple coordination and interaction tools are required to show who is logged on and must include a private chat channel.
- A formal coordination mechanism.
- Admission procedure.
- Guilds are fragile, so leadership is important. The game design has to keep the player motivated and allow for the resolving of internal conflict.
- Guild membership should be fluid to prevent people from becoming bored with their responsibilities.
- Guilds provide a stable social backdrop, and as members collaborate more they play longer.
- Bigger guilds are likely to survive longer, but the internal structure has to be sound. Skill classes have to be balanced, and it was found to be beneficial if smaller subgroups form within the guild, since coordination of smaller groups is easier.

- Levels have to be widespread, a concentration of knowledge was not as beneficial as was hypothesised. Fresh players serve as backup to replenish high attrition rates and members who transfer to a different guild.

- Guilds who take on more complex problems show better health, since it proves that their organisation is sound.

- There must be good connections between peers.

- If individual progress is the goal, then a smaller guild is better.

- For smaller guilds, larger subgroups turned out to work better since it allowed for a more diverse choice of quest partners.

- The time spent online should overlap with other players'.

- A guild allows for people to team up with a few other members for each task. They form an organic, team-based organisation, which lets people complete a task with whomever they think will be most capable in assisting them.

What can be learned from guild formation that is useful for team formation is that even though there should be lots of room for members to be creative in solving their tasks, leadership has to be clearly coordinated and respected.

While a formal admission procedure is undesirable, the team leader should go so far as to screen potential members on their abilities to ensure a good balance of skill.

Interactive cooperation should be encouraged, since it will make people feel more useful, appreciated and responsible. Again this calls on emotions for motivation. If cooperation proceeds successfully, it is also very much likely that connections between team members is good.

Finally, fluid team membership (occasionally changing team structure) will depend on the situation of the development team, but if possible it seems like a good idea. If people do not adjust too much to each other, and change often, they will become more flexible and adapt to many situations. Moreover, knowledge gained from previous situations can be used in different contexts, contributing to experience gain, and be shared with other team members.

4.5 Business games and their successes

To date, gaming in business applications is mainly used for training and simulation, such as flight simulators, military training applications and management simulation games. However, examples of virtual online communities are available.

To illustrate the added value of a collaborative online community for business enhancement, I have studied some case studies on current experimental business games and their lessons learned. The first is the case of the Internet Chess Club (ICC), as described by Ginsburg & Weisband [Gins 04]. The ICC is a popular subscription-based *Virtual Community Business*, which is no more than a virtual community serving as a means for promotion or for generating a main source of revenue. The study identifies some factors contributing to the ICC's success. A different study by Lainema [T 01] claims that business games enhance participation in business processes.

First of all, Ginsburg & Weisband [Gins 04] mention that any virtual business community has three basic requirements:

1. Inimitable information assets
2. Persistent money flow to gain trust
3. Economic infrastructure in which transactions can take place, facilitated by trust

On top of the basic requirements, there are more factors making the Internet Chess Club community to a success. When closely considering this list, it becomes clear that these findings can be mapped exactly back to the list of variables as described in Chapter 2 [2.5.1]. The following is a list of those findings with the corresponding game variables.

- Tools should be designed with the user's needs in mind.
  Accessibility, Positive affect, Need, Interface, Mechanics, Dynamics, Aesthetics
- The user should have a critical amount of functionality at his disposal.
  Autonomy, Interaction, Freedom of action, Mechanics
- Managers should delegate authority to members lower down in the ranks so that they will have power to shape the rules of the game.
  Autonomy, Competence, Psychological well-being, Rules, Players
- User feedback channels are important.
  Feedback, Interaction, Emotions, Goals, Challenge
- The acknowledgement of equity among members, so each one will feel appreciated.
  Autonomy, Emotions, Psychological well-being
- The availability of timely help and guidance.
  Feedback, Skill improvement, Goals, Accessibility
- The segmentation of communication channels to avoid overloading the user with unnecessary information.
  Accessibility, Feedback, Interaction, Psychological wellbeing
- Persistent IDs to enhance the sense of belonging and aid member retention.
  Presence/immersion, Emotions, Autonomy, Players
- The allowance of a real online market.
  Interaction, Competence, Dynamics, Mechanics, Rules
- Useful features in the game interface and easy navigation to those features.
  Accessibility, Emotions

The study by Lainema [T 01] focuses more on business processes as the key component of the business game, and hypothesises that the game format can help people learn and gain insight about the structure and flow of the processes. He defines a business process to have the following characteristics:

- A set of interrelated work activities characterized by specific inputs and value-added tasks that produce specific customer-focused outputs.
- It is recurrent.
- It affects some aspects of organizational capabilities.
- It can be accomplished in different ways that make a difference to the contribution it generates in terms of cost, value, service, or quality.
- It involves coordination.

When businesses intend to make a change, business process innovation typically has to happen. If more insight is desired, one has to ”fully understand the information needs of autonomous business units, and processes across the enterprise” [Dave 93].

Lainema argues that managing change involves employee empowerment, communication, process vision development, and process-based team formation. Employees should be able to establish self-management and rather than being directly guided as individuals, they should be intrinsically motivated by collaborative teamwork. The game should therefore support ”a change of mentality through training and education to create an environment for later change” [T 01].

In order to be able to properly describe the present business environment and business processes, the business game should directly embed the influence and importance of time. Moreover, participants should be able to view a holistic picture of the business, that is, the most important business functions and stakeholders. Also, the game should be configurable for each different business environment.

The basic lesson learned from Lainema is that business games are promising for enhancing participation, motivation and learning, but that it is still in the experimental stage. However, he does describe some useful game requirements that correspond mostly to the game design variables for *Interface*, *Environment* and *Context*. However, he mentions self-management and intrinsic motivation thought collaboration, which are supported by *Autonomy*, *Competence*, *Interaction*, *Presence/immersion* and *Feedback*.

So in conclusion, I can state that experimental business games and online communities which have been tested largely adhere to the variables previously set from a psychological exploration of games. This contributes to the justification of using the list of game variables as a base for the Aquima game design as will be described in the following chapter.

Chapter 5

Aquimatization: Game Concept

This chapter proposes a conceptual design for a possible new Aquima-based game. It introduces the concept of the game, and explains the basic ideas and storyline behind the game.
The game involves the entire product development cycle. The choice to include the entire process in the game was made to allow for more insight in the many different subprocesses and events which typically run in parallel and, if not properly monitored, may introduce many misunderstandings and errors. It became very clear during the workshops that this was, at least at the moment of writing, much more desirable than the original idea of introducing gaming aspects only to the way of modelling in Aquima Studio [C.1].

A detailed design description can be found in Appendix E. Some elements may be implemented on the short term, although the greater part is long-term thinking.

5.1 Introduction

Aquimatization is a multiplayer online business process progress-monitoring and development game that involves elements of role-playing, simulation, real-time strategy and puzzle solving. Explore and take on challenges in a fantastic, other-worldly game environment. Create hordes of minions to help you create settlements, and face the ultimate challenge of building and governing the most successful, thriving settlement on the planet. Collaborate closely with teammates and use a smart strategy to achieve your goals. Construction of stable buildings, careful resource and people management are key skills required to bring the challenge to a successful end.

5.2 Genre

The game genre for Aquimatization is a complicated issue. The game encompasses all aspects of the project development process, and since different process

areas have parallels with different game genres [Wode 06], Aquimatization contains and combines elements of many different genres. For a mapping of the development phases to the corresponding game genres, see figure [5.1].
To begin with, a description of each different genre involved:

- **Real-Time Strategy**
While the game is merely a metaphor to provide insight in and structure the development process (simulation), the project still takes place in real time. Therefore, decisions and implementations are likely to take effect either immediately or on short term. Tasks will run in parallel, and team members will therefore work concurrently, whether working on the same or a different task. Players will always have to make sure necessary information is in the right place at the right time, otherwise consequent tasks will not be able to kick off.
Strategic decisions play a part in constructing and maintaining whatever is built during the process.
Wodehouse & Bradley [Wode 06] found avatars to play a very important role in an RTS environment. Communication, control and coordination happened through avatars, they allow for the game situation to be quickly recaptured if the player has been offline for a while.
Typical RTS games involve fast-changing environments and are usually played by larger teams. These playing conditions of a geographically diverse, autonomous and asynchronous group successfully working together in a shared game world, show strong parallels to multi-disciplinary, global design team management [Wode 06, Ager 08].

- **Simulation**
Simulation games use a combination of skill, chance and strategy in order to simulate an aspect of reality, in this case project development. They typically involve complex data management, with the player using a multi-layered interface to support the organisation of textual and visual information. The balance between guidance and freedom is effectively solved using advisers or wizards. Wodehouse & Bradley [Wode 06] have conducted a study of how gaming techniques can impact the product development process. They found simulation games to have the most obvious comparison to the part of the design process in which design constraints are set and adjusted, and when using the information-rich interface in order to familiarise someone with a certain problem. This is because it provides rich visual support, and consequences of the player's actions can first be previewed in a 'safe' game environment before being implemented in the real world.
Simulation games have proven to be highly successful in engaging players for hours on end as well as including useful elements such as instructivism, constructivism, action-based learning, problem-based learning and situated learning [De B 08].

- **Role Playing Game**
The game will contain role playing elements. There are several roles in the project development process, as described in chapter 4 [4.2.1], which will have to be taken on during the game. Each player is part of a team which progresses through the game, experiencing a certain set storyline

and taking up quests.
Essential in a role playing game are stimulators, which can take the form of AI. Stimulators may be environmental elements providing the player with clues, or simply be the quest-giver who states the player's goal to him.
Spontaneous rather than strategy-based interactions happened through both language and symbols, such as icons. Creativity thus plays a large role here, but even though creativity is increasingly being valued in business environments, it is still hard for a business to organise creativity [Wode 06]. It thus seems that RPGs provide the greatest added value for creativity management, by means of a low-pressure, low-consequence environment in which people feel free to contribute and which could even stimulate them to think of innovative new ideas.
Typical RPG games take place in a 3D world through which players can navigate and engage in structured dialogues with other players in order to solve problems. To retain valuable ideas originating in these dialogues, however, a method to store and display the ideas would be required, together with access to an area in which creative thinking can take place.

- **Puzzle Solving**
This genre is most apparant when the players try to figure out the best way of how to set up the application, how it will fit in with the client's architecture, what modules to build and how to furnish those modules.

5.2.1 Interface genre

The player will view the world through a combination of perspectives. When navigating through the world, in search for new places to build towns, or when looking for homes to repair, or seeking a certain person to negotiate with, the world will be seen through an *isometric* perspective. This means viewing the world at a fixed, elevated angle, and when navigating it appears as though the player is looking down at the environment from an airplane. This perspective is best suited for providing overview of what is going on in the world.
On the other hand, when working alone, or when collaborating with a small number of team members, the *first-person* perspective will be used to signify direct interaction. In short, it can be said that this perspective will take over whenever working or communicating directly with an object or another person.
The visual, textual and spatial information will have to be organised in a multilayered interface, so that the player can switch between them and always have rapid access to whatever information he needs, whether it be the results from last week's workshop meeting, or a small map showing him where in the game world he is currently navigating.

5.2.2 Procedure genre

The procedure followed during game play will be a combination of strategy, adventure, puzzle and playing God. The game's focus is on building application modules, represented as houses or other buildings. This means a strategy of "careful creation, management and skillful positioning of resources on a game map" [De B 08] is required. Whereas there is no opponent intending to wipe

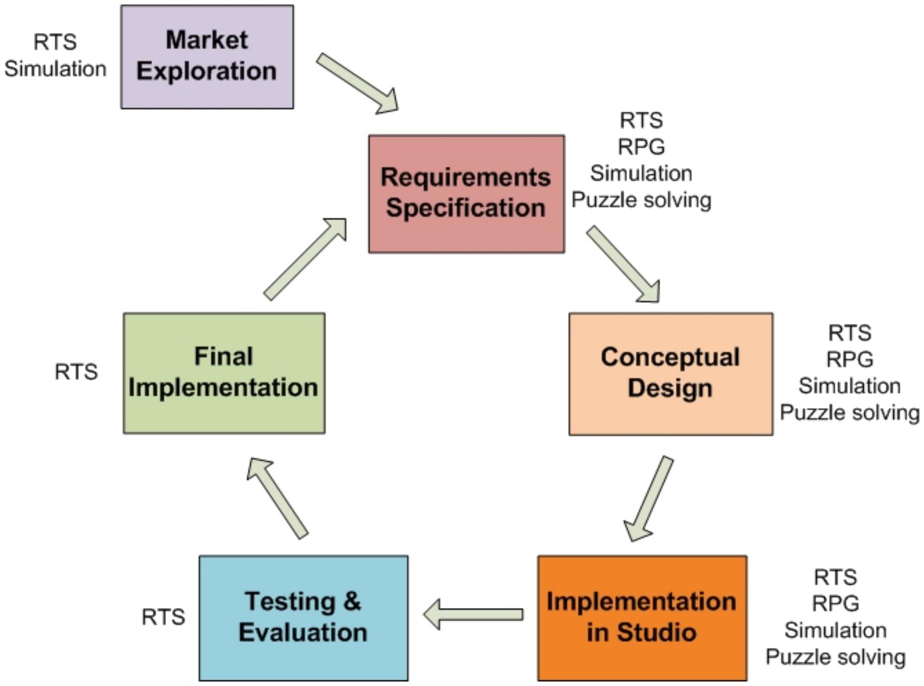


Figure 5.1: The product development process with the applicable game genres for each phase.

out whatever your team has created, there will be other teams to criticise the team's work, and possibly attack parts they don't like.

Adventure comes in when players set out to discover the unexplored world. They should seek out good places to build houses and towns.

Puzzle solving is a procedure that can come up at any time, during any activity. It merely signifies that the world is not straightforward, and if certain buildings don't match, or if storeys cannot be built on the foundations, something is clearly wrong and the puzzle is then to find out what.

Finally, *playing God* indicates that the team has complete control over the world, and that they can create and govern it exactly as they think is viable.

Besides the different types of focus in the procedure, the flow has to be of a short and cyclical nature, following the daily Scrum, weekly Workshop and monthly Sprint cycles Everest currently employs in the project development process.

5.3 Goals and Objectives

5.3.1 Game goals

The game goals are divided into long term, middle term and short term goals, as described in Chapter 2 [4.1.1].

Long term

- The main objective is to create a settlement as efficiently and robustly as possible so that the nation will be able to flourish.

The player is constantly reminded of this goal because he is continuously working on the settlement, improving it all the time.

Middle term

- The mayor must be kept happy at all times. Disasters, strikes and attacks should be avoided.
- Keep the game activity going at all times. Inactivity is a sign that players are waiting for new input which has been delayed.
- Gain experience points and get positive reviews for personal score and level improvement, and a high position in the project-ranking.
- Create viable and sturdy requirements. They must form the sound basis for the settlement.

The cycle of 'fear and relief' characterising middle term goals is integrated because players need to be continuously aware of the game world's state. Fear of deviations from normal functioning, relief when everything proceeds as it should.

Short term

- Create requirement minions.
- Set the minions to work to create a building.
- Review team members' work.
- Contribute to the library's wealth of knowledge.
- Have meetings and collaborate.

These goals can all be achieved quite simply, in a straightforward, linear way completely under the player's control. The results of these goals contribute to the achievement of the middle term goals.

5.3.2 Higher-level organisational goals

- Improve profits. If people like this way of working and more revenue can be gained this way, ultimately profits will improve too. Try to deliver something faster than the competitor.

- To create a constant cycle of feedback and improvement. This serves to improve process and product quality, and to allow people to learn and gain experience in an interactive fashion.

- Create insight and overview in project process. Particularly interesting for project leader, but each player has to have a complete overview of the specific domain and level he is working in.

- To actively involve the client in the process, which enhances the effectivity of the cyclical nature of the work process. It is easier for a client to review a house that has been constructed during a development cycle than a partly finished application with lots of mock-up effectivity. Judging a house will appeal more to the client than a prototype that seems to work but in reality does not. He is less likely to say he likes it, and can give more targeted criticisms.

- Improve requirement quality. Closer collaboration with the stakeholders and stimulating control mechanisms ensure that requirement quality is checked and reviewed multiple times by several knowledgeable experts, creating a sound, solid basis for the application. This suits the cyclical nature of the Everest development process well.

- Make modelling fun! If the task is fun to do, it will be done thoroughly and this will ultimately lead to improved product quality and improved process quality.

5.4 Theme

There are two prominent themes in the game procedure. The first is ’Rags to riches’: you start small and end big. When you start, you will find yourself in an unexplored, empty world which is at your team’s disposal to develop. You build settlements in a sustainable way, which gradually expand and develop as the project progresses. The game is set in a typical spirit of pioneering and colonisation. If your strategy has worked well, your team ends with a wealthy, thriving town.
The second theme is ’Playing God’: having complete control over your stretch of the world and being able to set it to your own hand. The game will be a type of ’Conquer and build the world’ game, which means unexplored land is at the your disposal and you can build and direct it as desired. Of course, you will have to share this right with your team members, but together you will be in control of how the settlement behaves.

5.5 Story

The game is set on the planet Aquimars, which has just been discovered, and the news goes that the atmosphere and environment are very similar to ours. Excitement all over about this possible New World. A team of brave explorers have been send there to colonise and civilise it. The pioneering spirit has been rekindled anew! The world is fresh, lush, rich with resources and waiting to be explored.

A state-of-the-art spacecraft has just landed on the fertile soils of Aquimars after an exhausting journey from Earth. The explorers are eager to set foot on the land for the first exploratory trip. All they are equipped with is a map of the unexplored world, which initially only shows the contours of the land. It is up to them to find the best places to settle and build new towns, so that the map will gradually be filled in. What they will meet in the land is unknown. The only thing they know is that satellite images have been taken showing the most fantastic beasts walking the land.

A suitable spot for settling will be a spot with lots of resources available at hand. The pioneers will have to use their advanced knowledge of genetic modification to create a host of minions who will get to work for them, cultivating land and building towns. However, the minions must be taken good care of if they are to evolve into sturdy, hard-working forces.

Once the basics have been taken care of, the building process can start. Resources for houses must be managed, because they must last until the whole town is complete. The more resources are left at the end, the greater the profits. Houses are expected to be solid and robust, and each house must be well-connected to other houses. Roads between them must be built and maintained, and the more actively the roads are used, the healthier the work process that is going on. The town will have a mayor (client, who at first is just part of the colonisation team), who comes and checks on the building process every month. He will then state what he likes, what he does not like and what he would like to see changed.

The construction process goes on, and as the work progresses the results become more and more visible. Failing work process issues become visible if there is a lack of activity from the minion forces.

In the town there is a hierarchical rule, and the higher up in the hierarchy, the more of the project you can see. If you are very low in the hierarchy, you can also see the entire process but you do not have as much access to project components as people higher up.

As the completion of the town approaches, and the mayor becomes more and more satisfied, it looks as though the challenges have been overcome. That is one small part of Aquimars conquered! News of their success is sent back to Earth. A big, non-virtual party will celebrate the end of the project (and immediately serve as an informal evaluation mechanism in which clients often end up saying more than during formal evaluation).

Part of the group of colonisers will be left behind to ensure that the established town will keep running properly. After all, even the mayor likes change every once in a while, and this has to be properly managed. He may even desire to let the town evolve into a full-fledged city.

A group of adventurous explorers will leave the comforts of town to set off in search for new places to civilise.

5.6 Description of the Player's Experience

When you start the game, after log-on, you arrive at a menu screen (Start game, Options, Exit). If you are a first-time player, you begin by creating your avatar, which embodies your online game personality as long as you play. When the avatar is all to your liking, you go back to the standard menu screen, possibly to change some settings or options, or to proceed to the game from there. The game automatically saves as you progress, so at every start you carry on from where you have finished off the night before.
When you start for the first time, all you see is an empty map representing the unexplored game world. You will be able to navigate the map, viewing it from a bird's eye perspective. Indicated are the different project sites *(which have already been filled in by the sales team - as project orders come in, sales people are required to fill in the details (client details, available budget, deadline etc) in the game application, which the game will translate into a suitable project site, with available resources based on the budget the client has made available)* which gives an overview of all activity going on in the world. A large red arrow will indicate the project site where you are required to work. Clicking on it will zoom in on that specific site, and the perspective changes to an isometric one.

You will meet your fellow explorers and colonisers on site in the virtual world. The site will be no more than a vast stretch of grassland, trees, hills, distant mountains, and occasional caves. All Everest team members will be present, and plans will have to be made for how to construct a flourishing settlement in this place. The first priority is to create a meeting place for the whole team including the client. For now, a cave and a campfire will have to do for a town hall. Later on, as the town evolves, so will the town hall. The town hall must have facilities for online collaboration and file sharing, so a whiteboard supporting real-time modelling, file-sharing and real-time application access must be available. Players have access to the town hall's facilities at all times, it serves as a common place to meet for organised collaboration. A real-time chatbox must also be available, but this will remain visible and usable at all times.

After the first kick-off meeting, you take on the role just assigned to you. Firstly, you start creating your minions (requirements). The mayor is invited to come and visit the site, and using the chatbox facilities the requirements must start being drawn up. After having spoken to the mayor, minion creation can begin. You can genetically modify a single body so that it will grow arms, legs, wings, a head, horns, whatever features you like. There will be a button allowing for a new minion to be created, and clicking this will make a small limbless body appear on your screen. This shape can then be defined with the desired features. As you give the minion features, you are prompted to enter information about the requirement in a small typing window. The minion evolves as the requirement evolves during each development cycle. Elaborating existing information makes the minion grow, adding new information creates new features. There is no limit to the features you can give your minion.
There is one demand, though, and that is that the minion must be helpful. He must, together with all other minions, get to work on building the town. As soon as the requirements are reviewed by team members, even in the initial stages, and have been agreed upon as being viable, the minion will become part

of the workforce. Related requirements can be grouped by giving such minions similar features, be it in limbs, colours, patterns etc.
The activity of the minions is a sign that the requirements are sound, and therefore the building of the town can start. Meetings in the town hall determine the plan and the mayor's wishes, and as soon as maps and models are drawn, the houses (project modules) can be built.
When the development cycle has been repeated so many times that the minions have reached full strength and maturity (the requirements are fully complete and correct), the minions will be able to take care of themselves, and will just continue working for you.

As you issue orders for buildings to be created, you will see the minions going to work. You can zoom in on a house, and enter it. At this point the perspective will change to a first-person perspective. Inside the house you will see empty rooms, depending on the modules and entities you have created. You will see that there is a link to the Aquima Studio graphical modelling editor in your interface, which can then be used to fine-tune your model. As a reward for creating a good model with correct attributes and valid relations, your room will automatically be furnished as you model. Later on, you will have the opportunity to edit the furnishings as a reward for your good work.
Then roads and paths must be constructed connecting different houses. As long as the process goes smoothly, the roads will stay in good condition. If bugs are detected, road quality will deteriorate.
The town evolves during each development cycle, just like the minions. Each development cycle is a new level. You as a player, however, can only level up by gaining points as you work. So take care to construct your houses and tend your minions well, you will be rewarded for it!

As the mayor pays frequent visits to the town under construction, he will want to see hard-working minions and beautiful houses. If the mayor is happy, the situation is peaceful and everybody lives in harmony. The mayor also decides about the working culture. He can decide to work democratically, or to just dictate exactly what he wants the town to become. The mayor reflects the client's culture, to which the work forces usually adapt.
However, if the process is not going well, your houses are not stable, your roads are poor and the minion forces are going on strike, the mayor will not be happy. Neither will the lead supervisor be. This is a situation to watch out for, since disaster can strike at this point. It is your challenge to get your forces back on track before the mayor becomes too angry.

As the work progresses through each evolving level, you also evolve as you gain experience with the process, learn about best practices and mistakes and guide your minions through successful building challenges. Reviewing is another essential mechanism for your personal development. As you review another, and others review you, you learn to judge and rate your peers, and others will give you feedback. Positive feedback will increase your personal score, just as gaining experience points will do.
Weekly meetings in the town hall with all relevant stakeholders present allow for the discussing and reviewing of the delivered products at that time, and for maintaining the motivated spirit and staying on track.

As you gain or seek experience, the town library is a good place to visit. This store of knowledge starts of simple too, but will evolve depending on the project's progress. If all goes well, shelves upon shelves of books will be stored there, each one of them containing precious knowledge, tips and best practices waiting to be revealed to you. You must rate whatever you read, and if you feel you can make a worthy contribution to the treasure chest of knowledge, you can simply place another book on the shelves. Knowledge from other projects they wish to share is also available, so there is unlimited access to whatever you want to know.

As the project proceeds and nears completion, preparations for the final feast must be started. You will meet face to face with all team members and stakeholders for the celebration of project completion, where the final reviews can be done informally.

5.7 Game Variables in Aquimatization

Seeing the creations, both minions and buildings, successfully evolve and having the feeling of being in control (playing God) induces a sense of *autonomy* and *competence*, the first two prerequisites for a successful game [Ryan 06]. Emotions being drawn on are happiness, the instinct to care and create, affinity for your creations, and these feelings enhance immersion in the game.
The mayor presents a different kind of emotion. He plays an important part in the project, but he represents the typical input for a 'freedom from fear' cycle. It is the player's job to keep the mayor happy, and he responds to the mayor's behaviour in such a way that a good balance is always kept. Mayor satisfaction after all efforts to please him is a reward that boosts *competence, self-esteem, positive affect, relatedness* and *achievement.*
Also, watching progress is intrinsically rewarding, just as receiving feedback from reviewers. Apart from rewards, there is also an element of *skill improvement* and interactive learning experience through *competition* (trying to produce the best work of the team) and *cooperation* (peer reviewing and possibly helping each other) involved.
The player is assumed to have several *needs* during the entire course of play. These needs are for: intellectual challenge, resources, feedback, knowledge gain, and skill improvement. Intellectual challenge is present at all times, in the way you set up the requirement and how to group them, how to set up your modules and other development issues. Resources are present in sufficient quantities in the game world, depending on how much money the mayor has provided. Careful management of resources is needed, though, and *adaptability* will be important here. Feedback, knowledge gain and skill improvement are all facilitated through collaboration, reviewing and active knowledge sharing. Each of these mechanisms are rewarded either through experience points, or expert feedback.

Chapter 6

Conclusions

6.1 Project results

In this thesis, I have achieved several immediate results. I will describe them as answers to my research questions.

What aspects of gaming are essential for a successful and motivating game?

First of all, I have distilled from the literature important principles of good game design, and charted the most important variables for game motivation and game design [2.5.1] in an integrated overview. This was done in such a way that any future (modelling) game design can be directly based on these variables. The focus during this exploration has been mostly on the psychological mechanisms underlying motivation and emotions.
Secondly, the role of interaction in a modelling game has been elaborately explored, together with the benefits gained when effective collaboration takes place in a virtual environment [3]. The overlap with serious games has been described [3.2.1], providing background information on the benefits of collaborative modelling games and how they provide frameworks for facilitating structured method engineering.

What HCI aspects have to be taken into consideration to make modelling more accessible?

I have deviated slightly from this question. Modelling is an activity that can be made more accessible by emphasising the collaborative, interactive aspects and providing an interface which appeals visually and emotionally, and presents the model components in a logically structured way. Instead of just modelling, I have focused on how to assess a modelling *game* for playability.
I have investigated several methods for assessing game usability 3.4.2 and integrated one of these methods with conventional usability research methods and collaboration assessment tools, to come to a collection of usability assessment tools for a collaborative modelling game 3.5.

Where in Aquima Studio is the implementation of gaming aspects most desirable?

After the theoretical foundations had been laid, and the situation at Everest and their development methodology explored, I organised several workshop sessions in which ideas for the new Aquima modelling game were to be generated. However, it was found to be much more desirable, at the time of writing, to create a game which would monitor the team's progress, keep the process flowing smoothly, allow more insight in the structure of requirements and related models, and which would greatly facilitate collaboration and online meeting [C.1]. I thereby deviated again from the original question.

How may such an implementation be realised?

Finally, based on the outcomes of the workshops, I have developed a concept [5] and a detailed design [E] for a *business process progress-monitoring and development game* encompassing all aspects of the development process. I have specifically chosen that name because the game encompasses all aspects of the business operational process, it monitors development progress by visualising every player action in the game world, and development is what players do in it.

The practical use of the game cannot yet be determined, since the game has so far only been reviewed by selected Everest employees. It has been met with enthusiastic reactions, but has not been tested in context because the creation of a working prototype was not part of the project scope.
However, the expectation is that the game will contribute to more insight in applications, the development process itself, and product quality, due to integrated collaboration and peer reviewing. A new aspect is that people are rewarded for engaging in collaboration, reviewing and knowledge sharing, so that motivation to participate in such activities will significantly increase. Rewarding people playfully enforces that they do certain things. The playful character, however, ensures that people never consciously *feel* forced to do things, which is detrimental to motivation.

An overview of the most obvious benefits the game promises:

- Ensures requirement correctness, because application building can only start after the requirements have been reviewed and approved by all team members, including the client.
- Reviewing is playfully enforced by rewarding the player for it, or constraining actions until reviewing has taken place.
- Documentation and knowledge sharing is playfully enforced, again by directly rewarding the player for it.
- Provides insight in applications for maintenance activities due to the graphical representation and available documentation.
- Collaboration is made so obvious and easy that players cannot avoid it. Players can see each other's work, and also meet in the game world.

- Project progress is easy to track, since all activity can be immediately seen in the game world and performance statistics are tracked.
- The online virtual world saves travelling time and costs, especially for the client, or other participants who do not play an everyday role in the project.
- The game may stimulate the client to become a more active participant in the development progress, since he is given a high status and has the right to act immediately if something is not to his liking.

Game aspects for a successful game

Computer gaming has become one of the most popular forms of human entertainment. With revenues surpassing those of Hollywood [Yi 04], the question of what factors are responsible for this enormous engaging power of games has been extensively researched. Ryan et al. [Ryan 06] state that gaming is all about satisfying basic psychological needs. Satisfaction has to be achieved in relatively short time, otherwise the player will become frustrated and lose interest.

Highly important factors in finding satisfaction are the direct experiences during play, immersion in the game world, an appealing interface, and most importantly, finding sufficiently challenging missions from which appropriate rewards can be gained. Secondary factors, which the player experiences subconsciously, are the induced emotions, narrative qualities, a feeling of autonomy and freedom of action, and the game's accessibility. Poorly operable game controls result in player frustrations, which are clearly undesirable. Players may also play for the collaborative aspect, to feel relatedness to game characters and other players.

A good game should have a positive impact in short-term psychological wellbeing. Three factors are thought to underly psychological wellness: autonomy, competence and relatedness [Ryan 95, Deci 00]. That is, the game should cleverly adapt to the player's situation so that the player will feel like he has mastered significant game skills, the player must experience a feeling of happiness and vitality after playing, and he should feel immersed in the game environment. If a game can induce such feelings, and players are willing to dedicate voluntary time and energy to it, a successful game has been created.

Aquima Studio and game aspects

The component Aquima Studio currently serves for model implementation and interpretation at Everest. The goals at the moment are mainly to create a model, do this in a syntactically and sematically correct way so the model makes sense, and from this generate a working application module.

An attempt has been made in the game design to map each Studio action to a game action, so that modelling may ultimately take place in the game world in the future. However, for the time being, a temporary solution has been sought making use of the graphical modelling editor, which has recently been implemented in Aquima Studio. It can be accessed from the game world to do detailed, complicated editing of models. There is a link between the game world and Studio, so that model editing will become directly visible in the game representation of the model (the building), by means of additional furniture and other decorative items appearing in rooms. The reason modelling directly in the game has not been elaborately investigated is because the focus in the workshops [C.1] has been geared towards an all-encompassing development game rather than a solely modelling game.

However, the game does aim to achieve the Aquima Studio goals I have analysed and identified in [4.1.1], that are still largely lacking. The most important are: common understanding and intepretation of the case, consent on the solution, collaboration, improvement of process and product quality, improvement of personal knowledge gain and sharing, and enhanced communication with client and team members.

Game aspects in the Aquimatization game design

In my game design I have carefully integrated all necessary game design variables. Whereas the variables *autonomy* and *competence* are induced throughout the entire game, each phase in the development process induces its own emotions.

The game's goals have been defined as *long-term, middle-term* and *short-term*, which serve as guidelines for what the player has to achieve.

During the requirements engineering phase, when requirements have to be created and developed in the embodiment of fantasy minions, basic instincts such as *caring, happiness* and *affinity* are called on. Rewards for your efforts are seeing your minion grow bigger and stronger, and finally having it go to work for you when the requirement is mature.

The client, in his role of mayor, presents a challenge in that he has to be kept happy. This is the typical cycle of achieving *freedom from fear*. Every time he threatens to become unhappy, and this can be prevented, rewarding success is achieved.

A challenge is presented in reviewing and knowledge sharing. The aim is to get positive peer reviews and high ratings for your work, so there is a *need* to do the work as well as possible.

Another challenge is careful resource management. Resources are limited by the money the client provides, so these have to be managed so that they last for the entire duration of the project. This challenge is of an intellectual kind, requiring long-term planning. Rewards come in the form of success at the end of the project, and mayor satisfaction.

Concerning the scoring system, I have specifically chosen for a system in which the player can score points based on the reviewed results of his work, rather than his direct actions. When rewarding the player for his results, it ensures that he will do his work properly to get the highest possible appreciation. If, for instance, scores are granted for the first person to have created 10 entities, players are likely to create unnecessary entities just for the sake of gaining points. In this case, he will get punished for that, because other players reviewing him will not appreciate useless entities. Also, points are awarded if your work processes run smoothly. Again, they can only run smoothly if the underlying model functions properly.

A scoring mechanism, however, is a necessary component in a game that enhances *competitiveness*, motivates the player's basic need to improve himself, and last but not least acknowledges the player's capacities to make him feel *competent.*

During game play, each player is assigned a role. This serves mainly to enhance the game's *narrative content*, and the player's sense of *presence* and *immersion.* Finally, the game is of a massive multiplayer online (MMO) type. This parallel is used because collaboration is a second nature in MMO games, just as it should be in the Aquimatization game.

In this thesis, I have focused greatly on the HCI aspects of project development, especially on how to create insight in the complex whole and how make it *fun* for the participants. So far, human aspects and specifically fun have always been of minor importance as compared to aspects like *ease of use* or *learnability.* If fun was taken into consideration, it has always been largely in the form of offering a piece of fun as a reward for enduring an otherwise boring task. However, I believe that making the entire application fun to work with in combination with good usability design immediately lowers the learnability threshold because people will be more eager to work with the application.
Efforts to create integrated collaboration systems in which the entire work process is concentrated are not new, but they have always been document-centric (the document is the foundation; as it loads, the appropriate software loads with it) [Davi 01] and not application-centric (the application is central, and loads so that editing can take place). My approach allows a game world to load, in which several activities can take place, such as real-time collaboration and model editing. Possibly, in the future, parallel game worlds may be built so that collaboration can ultimately be integrated in all of Everest's activities.

6.2 Achievements for method engineering

I have provided a psychological foundation for the use of games for method engineering. Psychological research shows that people are indeed naturally attracted to games because it satisfies their basic psychological needs. Therefore it can be safely assumed that this power of attraction will still work when game design is applied to method engineering to make it more fun.
Moreover, I have closely looked at the process of modelling from an interaction-oriented perspective, and have confirmed that interactions indeed play a significant role in modelling, as already suggested in [Hopp 09].
Interaction is also a key condition for successful collaboration, as has been the goal throughout the entire project. Collaborative work has many benefits, as can be seen when examining different forms of (virtual) collaborative work. The most illustrative example has been Global Software Engineering [Ager 08]. The 'unknown benefits of GSD' Agerfalk describes have strong parallels with the indirect benefits for the team and work process that will hopefully be achieved by using a gaming approach for project development, such as innovation, shared knowledge, better resource allocation, reduced coordination costs, team autonomy, better documentation, recorded communication and a structured work process.
Serious gaming is a field in which games are used to create interactive learning experiences. They have long recognised the potential of interaction, and the benefits of serious gaming, as outlined in [3.2], may be achieved in a modelling game as well. Especially benefits such as mastering skills, gaining experience and having an active learning experience will be very useful for the rather com-

plicated and abstract process of formal modelling. It has been proven that active learning experiences have a deeper impact on humans compared to passive learning by being told or reading from a book [Grei 07].
Moreover, some additional benefits may be gained from a modelling game which are not directly aimed for in a serious game. Such benefits are successful collaboration, a comprehensible model, common consent and compliance to customer demands [3.2], originating from the interactive nature of the game and the active participation of both modellers and client.

The business process progress-monitoring and development game as described in this thesis can be applied to project development in general, since it follows the standard phases of project development and is only linked to Aquima Studio by the fact that it has a built-in link to Studio for model editing.
Modelling can be done in any graphical model editor that interprets models directly, or in a possible future modelling game.
In essence, it can be said that the mechanisms underlying the game are the key factors for success. Collaborative chat facilities, online real-time modelling tools, virtual meeting possibilities and online file-sharing are just a few of the game mechanisms meant for the promotion of interaction. While any of these mechanisms do not necessarily need a game interface to function, the game metaphor has been chosen because of its psychologically appealing effect and motivational pull. Implementation of competitive game mechanics such as a score and reviewing system serve as an extra motivational challenge for modellers to do their work thoroughly. These findings can be applied in any context.

6.3 Future research

Because the focus in this thesis has shifted from applying gaming aspects to modelling to creating an all-encompassing development game, future research should focus on how to create a game specifically geared for the process of modelling itself. This could become one of the parallel worlds accessible from the general project's game world, so that project development will eventually become a game in its entirety.
There certainly seems to be a desire to turn modelling itself into a game, since my concept of furnishing a room as a parallel for creating entities and related attributes was enthusiastically met. However, it needs to be researched and described in more detail before the concept can become functional.
Furthermore, the basic framework I have set up for evaluating modelling game playability can be further developed, and from this refinement a set of objective criteria for judging model and collaboration quality may result. This may ultimately be useful to judge the quality of the designed games for modelling and model validation.

Important steps to take now are to develop a prototype of the described game and to test its value in practice. It will be very beneficial if the largely theoretical exploration of gaming concepts in business environments can be backed by actual test results. If it should turn out to be successful, the game goals can be taken one step further, from mainly serving to track progress and guard process and product quality to a true experience of a model. For that, the embodi-

ment of the models should become even more interactive, reacting to the player rather than being just editable. A reaction could take the form of presenting contents to the player, or prompting the player with questions resulting from a (mis)match between the model embodiment and the objective test case (which has been written in advance). Games should ultimately be designed to experience a model in the same way a player currently experiences a game story.
Developing such games will require interdisciplinary research from fields such as Cognitive Engineering, Human-Computer Interaction (HCI), Method Engineering, Collaborative Modelling, but also Game Design Theory and Game Psychology. I will be starting a PhD in this research direction in April 2009.
The aim is to do research in collaboration with the industry, so that concepts may be tested in realistic contexts. In this way, I hope to gain more insight in how interaction works in context, if gaming concepts work similarly in industrial work contexts as in voluntary play contexts, and what the best ways are to apply these game concepts in a modelling game.

Afterword

One of my great personal interests is nature conservation and sustainability. During my literature research I happened to come across a paper explaining a method for assessing how ICT solutions can contribute to a reduction in CO_2 emissions [Hash 05]. It caught my attention and I began reading.
It so turns out that the greatest savings in environmental burdens are achieved in *office space*, a collective name for ”operating efficiency, documentation space and equipment floor space due to ICT introductions” [Hash 05]. This was due to a reduction in person-hours and thereby in energy consumption.
Hashitani et al. [Hash 05] mention eight factors which are used for assessing environmental burden:

1. Resource consumption
2. Human transportation
3. Goods transportation
4. Office space
5. Warehouse space
6. Electricity for ICT/NW equipment
7. Network data communication
8. Waste generation (total weight of the above factors)

It appeared to me that the use of Aquimatization will contribute to many of these goals, not just office space. First of all, resource consumption can be significantly reduced. The online nature of the game will make CD media superfluous. Since the interface should be kept simple and online help and documentation is always available (town library and online experienced peers) elaborate user manuals, guides and hardcopy shared knowledge are also no longer necessary.
The game can account for a reduction in human transportation, since meetings can be done online, saving the client and other off-location team members the need to travel to do good work. This is directly related to a reduction in office space, for the reason already mentioned by Hashitani et al.
I find it very encouraging that ICT has the potential to contribute to an efficient, environmentally conscious way of working, and that my game may be a small step in that direction.

Bibliography

[Ager 08] P. Agerfalk, B. Fitzgerald, H. Olsson, and E. Conchuir. “Benefits of Global Software Development: The Known and Unknown”. *Lecture Notes in Computer Science*, Vol. 5007, p. 1, 2008.

[Bart 96] R. Bartle. “Hearts, clubs, diamonds, spades: Players who suit MUDs”. *Journal of MUD Research*, Vol. 1, No. 1, 1996.

[Berg 08] J. van den Berg, J. Jonkergouw, and M. Olislagers. “Business View”. Graduation report, 2008.

[Bomm 08] P. v. Bommel, S. Hoppenbrouwers, H. Proper, and J. Roelofs. *T. Halpin et al.: Innovations in Information Systems Modeling: Methods and Best Practices*, Chap. Concepts and Strategies for Quality of Modeling. *Advances in Database Research series*, IGI Global Publishing, 2008.

[Bord 99] L. Borden and D. Perkins. “Assessing your collaboration: A self evaluation tool”. *Journal of Extension*, Vol. 37, No. 2, pp. 67–72, 1999.

[Brig 03] R. Briggs, G. De Vreede, and J. Nunamaker. “Collaboration Engineering with ThinkLets to Pursue Sustained Success with Group Support Systems”. *Journal of Management Information Systems*, Vol. 19, No. 4, pp. 31–64, 2003.

[Brin 96] S. Brinkkemper. “Method engineering: engineering of information systems development methods and tools”. *Information and Software Technology*, Vol. 38, No. 4, pp. 275–280, 1996.

[Brin 99] S. Brinkkemper, M. Saeki, and F. Harmsen. “Meta-modelling based assembly techniques for situational method engineering”. *Information Systems*, Vol. 24, No. 3, pp. 209–228, May 1999.

[Broo 91] R. A. Brooks. “Intelligence without representation”. *Artificial Intelligence*, Vol. 47, pp. 139–159, 1991.

[Bura 08] S. Bura. “Emotion Engineering: A Scientific Approach for Understanding Game Appeal”. http://www.gamasutra.com/view/feature/3738/emotion_engineering_a_scientific_.php, July 29th 2008.

[Craw 84] C. Crawford. *The Art of Computer Game Design.* Osborne/McGraw-Hill, 1984.

[Dave 93] T. Davenport. *Process Innovation: Reengineering Work Through Information Technology.* Harvard Business School Press, 1993.

[Davi 01] J. Davis, E. Subrahmanian, S. Konda, H. Granger, M. Collins, and A. Westerberg. “Creating Shared Information Spaces to Support Collaborative Design Work”. *Information Systems Frontiers*, Vol. 3, No. 3, pp. 377–392, 2001.

[De B 08] P. De Byl. *Handbook of research on effective electronic gaming in education*, Chap. Designing games based embedded authentic learning experiences, pp. 1068–1087. IGI Global Publishing, Hershey, 2008.

[De K 78] B. De Koven. *The Well-Played Game.* Anchor Press, 1978.

[Deci 00] E. Deci and R. Ryan. “The ”What” and ”Why” of Goal Pursuits: Human Needs and the Self-Determination of Behavior”. *Psychological Inquiry*, Vol. 11, No. 4, pp. 227–268, 2000.

[Deci 80] E. Deci and R. Ryan. “The empirical exploration of intrinsic motivational processes”. *Advances in Experimental Social Psychology*, Vol. 13, pp. 39–80, 1980.

[Deci 85] E. Deci and R. Ryan. *Intrinsic Motivation and Self-Determination in Human Behavior.* Springer, 1985.

[Desu 04] H. Desurvire, M. Caplan, and J. A. Toth. “Using heuristics to evaluate the playability of games”. In: *CHI ’04: CHI ’04 extended abstracts on Human factors in computing systems*, pp. 1509–1512, ACM, New York, NY, USA, 2004.

[Dijk 08] J. v. Dijk. “Cognition is not what it used to be: reconsidering usability from an embodied embedded perspective”. *In press*, Vol. , p. , 2008.

[Dour 01] P. Dourish. *Where the Action Is: The Foundations of Embodied Interaction.* Cambridge, MA: MIT Press, 2001.

[Duch 07] N. Ducheneaut, N. Yee, E. Nickell, and R. J. Moore. “The life and death of online gaming communities: a look at guilds in world of warcraft”. In: *CHI ’07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 839–848, ACM, New York, NY, USA, 2007.

[Ever 09] Everest. “Everest BV - Dynamic Business Process Management - Aquima Business Engineering Suite”. http://www.everest.nl/everest/NL/aquima, 2009.

[Game 07] E. H. C. Gámez. “The role of input devices in the gaming experience”. In: *ECCE ’07: Proceedings of the 14th European conference on Cognitive ergonomics*, pp. 289–291, ACM, New York, NY, USA, 2007.

[Gee 03] J. Gee. “What video games have to teach us about learning”. 2003.

[Ghoz 07] D. Ghozland. “Designing for motivation”. http://www.gamasutra.com/view/feature/1419/designing_for_motivation.php, June 7th 2007.

[Gins 04] M. Ginsburg and S. Weisband. “A Framework for Virtual Community Business Success: The Case of the Internet Chess Club”. *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS’04)*, Vol. 07, p. 70196b, 2004.

[Glin 08] E. Glinert. “Designing games that are accessible to everyone”. http://www.gamasutra.com/view/feature/3538/designing_games_that_are_.php, February 13th 2008.

[Grei 07] F. L. Greitzer, O. A. Kuchar, and K. Huston. “Cognitive science implications for enhancing training effectiveness in a serious gaming context”. *J. Educ. Resour. Comput.*, Vol. 7, No. 3, p. 2, 2007.

[Hash 05] T. Hashitani, S. Suzuki, and Y. Horikoshi. “Method for Approving ICT Solutions as Environmentally Conscious”. *Fujitsu scientific and technical journal*, Vol. 41, No. 2, pp. 153–159, 2005.

[Hopp 05] S. Hoppenbrouwers, H. Proper, and T. P. van der Weide. “Formal Modelling as a Grounded Conversation”. In: *Proceedings of the 10th International Working Conference on the Language Action Perspective on Communication Modelling*, pp. 139–155, June 2005.

[Hopp 08a] S. Hoppenbrouwers, P. van Bommel, and A. Jarvinen. “Method Engineering as Game Design: an Emerging HCI Perspective on Methods and CASE Tools”. In: *Workshop proceedings of EMMSAD ’08: Exploring Modeling Methods for Systems Analysis and Design, affiliated to CAiSE ’08*, June 2008.

[Hopp 08b] S. Hoppenbrouwers. “Community-based ICT development as a multi-player game”. In: *Proceedings of conference ”What is an Organization? Materiality, Agency and Discourse”*, May 2008.

[Hopp 09] S. Hoppenbrouwers. “Setting rules of play for collaborative modelling”. *International Journal of e-Collaboration, Special Issue on Collaborative Business Information System Development*, Vol. , p. , 2009.

[Huhh 05] J. Huhh and S. Park. “Game design, trading markets, and playing practices”. *Unpublished manuscript*, 2005.

[Huni 04] R. Hunicke, M. LeBlanc, and R. Zubek. “MDA: A Formal Approach to Game Design and Game Research”. In: *Proceedings of the AAAI-04 Workshop on Challenges in Game AI*, pp. 1–5, 2004.

[Jarv 07a] A. Järvinen. *Games without Frontiers, Theories and Methods for Game Studies and Design.* PhD thesis, University of Tampere, Finland, 2007.

[Jarv 07b] A. Järvinen. "Introducing Applied Ludology: Hands-on Methods for Game Studies". In: *International Conference of the Digital Games Research Association*, DiGRA 2007 Situated Play, Tokyo, Japan, September 24th - 28th, 2007.

[Jarv 08] A. Järvinen. "GameGame". http://gamegame.blogs.com/, 2008.

[John 06] S. Johnson. *Everything Bad Is Good for You: How Today's Popular Culture Is Actually Making Us Smarter*. Riverhead Books, 2006.

[Jord 08] E. R. Jordens). "Business Engineering in five minutes". Internal company document, 2008.

[Karl 08] F. Karlsen. "Quests in context: A Comparative Analysis of Discworld and World of Warcraft". *The International Journal of Computer Game Research*, Vol. 8, No. 1, 2008.

[Mann 01] T. Manninen. "Rich Interaction in the Context of Networked Virtual Environments-Experiences Gained from the Multi-player Games Domain". In: *People and Computers XV: Interactions Without Frontiers: Joint Proceedings of HCI 2001 and IHM 2001*, Springer, 2001.

[Mirb 06] I. Mirbel and J. Ralyt'e. "Situational method engineering: combining assembly-based and roadmap-driven approaches". *Requirements Engineering*, Vol. 11, pp. 58–78, March 2006.

[Oatl 96] K. Oatley and J. Jenkins. *Understanding Emotions*. Blackwell Publishers, 1996.

[Remo 08] C. Remo. "Learning From Crysis: The Making of Crysis Warhead". http://www.gamasutra.com/view/feature/3785/learning_from_crysis_the_making_.php, September 15th 2008.

[Rigb 04] S. Rigby. "Player Motivational Analysis: A model for applied research into the motivational dynamics of virtual worlds". 2004.

[Rose 08] J. Rose. "Fewer Mechanics, Better Game". http://www.gamasutra.com/view/feature/3621/fewer_mechanics_better_game.php, April 15th 2008.

[Ryan 00a] R. Ryan and E. Deci. "Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions". *Contemporary Educational Psychology*, Vol. 25, No. 1, pp. 54–67, 2000.

[Ryan 00b] R. Ryan and E. Deci. "Self-Determination Theory and the Facilitation of Intrinsic Motivation, Social Development, and Well-Being". *American Psychologist*, Vol. 55, No. 1, pp. 68–78, 2000.

[Ryan 06] R. Ryan, C. Scott Rigby, and A. Przybylski. "The Motivational Pull of Video Games: A Self-Determination Theory Approach". *Motivation and Emotion*, Vol. 30, No. 4, pp. 344–360, 2006.

[Ryan 95] R. Ryan. “Psychological Needs and the Facilitation of Integrative Processes”. *Journal of Personality*, Vol. 63, No. 3, pp. 397–427, 1995.

[Sale 04] K. Salen and E. Zimmerman. *Rules of Play, Game Design Fundamentals.* Cambridge, MA: MIT Press, 2004.

[Shef 08] B. Sheffield. “Game Design Psychology: The Full Hirokazu Yasuhara Interview”. http://www.gamasutra.com/view/feature/3769/game_design_psychology_the_full_.php August 25th 2008.

[Swar 03] W. Swartout and M. van Lent. “Making a game of system design”. *Commun. ACM*, Vol. 46, No. 7, pp. 32–39, 2003.

[T 01] L. T. “Enhancing Participant Business Process Perception through Business Gaming”. *34th Annual Hawaii International Conference on System Sciences (HICSS-34)*, Vol. 01, p. 1014, 2001.

[Wode 06] A. Wodehouse and D. Bradley. “Gaming techniques and the product development process: commonalities and cross-applications”. *Journal of Design Research*, Vol. 5, No. 2, pp. 155–171, 2006.

[Yee 04] N. Yee. “Unmasking the Avatar: The Demographics of MMO Player Motivations, In-Game Preferences, and Attrition”. http://www.gamasutra.com/resource_guide/20040920/yee_pfv.htm, September 21st 2004.

[Yee 06] N. Yee. “Motivations for Play in Online Games”. *CyberPsychology & Behavior*, Vol. 9, No. 6, pp. 772–775, 2006.

[Yi 04] M. Yi. “They got game: Stacks of new releases for hungry video enthusiasts mean its boom time for an industry now even bigger than Hollywood.”. San Francisco Chronicle, December 18th 2004.

List of Figures

List of Tables

Appendix A

Ubisoft on gaming

On March 5th, 2008, a large symposium on learning by gaming was held at the Technical University of Delft. The first speaker was a representative from Ubisoft, who lectured on the essential aspects of gaming. Here follows an overview of what was said.

First of all, games have to be accessible. Everyone has to be able to play. However, the main rule of thumb to be kept in mind is *easy to play, difficult to master*. The game always has to present a challenge. Being challenged is one of the most essential factors for triggering motivation. A game present a puzzle challenging us, the excitement it induces in players is of vital importance.
A game can be seen as a modern way of story-telling. That is why games are often a combination of a movie and a book. In fact, the movie industry is often used as a rich source of inspiration for new plots and settings. The main aim is to let the player experience intense emotions by actively participating in a story, set in a fascinating world. Emotions literally have to be brought across through the screen, making it seem as realistic as possible for the player. It is important to consider carefully what the player likes, and what he seeks in a game. An element that follows from this is diversity: the player does not want to view the same environment, buildings and people all the time. The difficulty in this is that while each environmental element has to have its own unique personality, it also has to remain realistic at all times, no matter what the context.

Gaming presents a combination of fun, learning, intuitive behaviour and social activity. It reflects true human nature: there is an environment and there are basic rules, the adventures are directed by the player's own actions. Several important factors of a good game are freedom of action, usable control, display and environment. The player has to have the freedom to perform missions in his own way.
However, actions have to be actively experienced. For example, when leaping across a gorge, the player has to feel a jolt of excitement, but must never be scared away. The player is guided in his actions, but the right degree of tension has to be stimulated.
The game should also actively involve the player. In Ubisoft's 2007 release *Assassin's Creed*, the story is set in such a way that the player assumes the role of the assassin's descendant. A clear link between the past and the present is

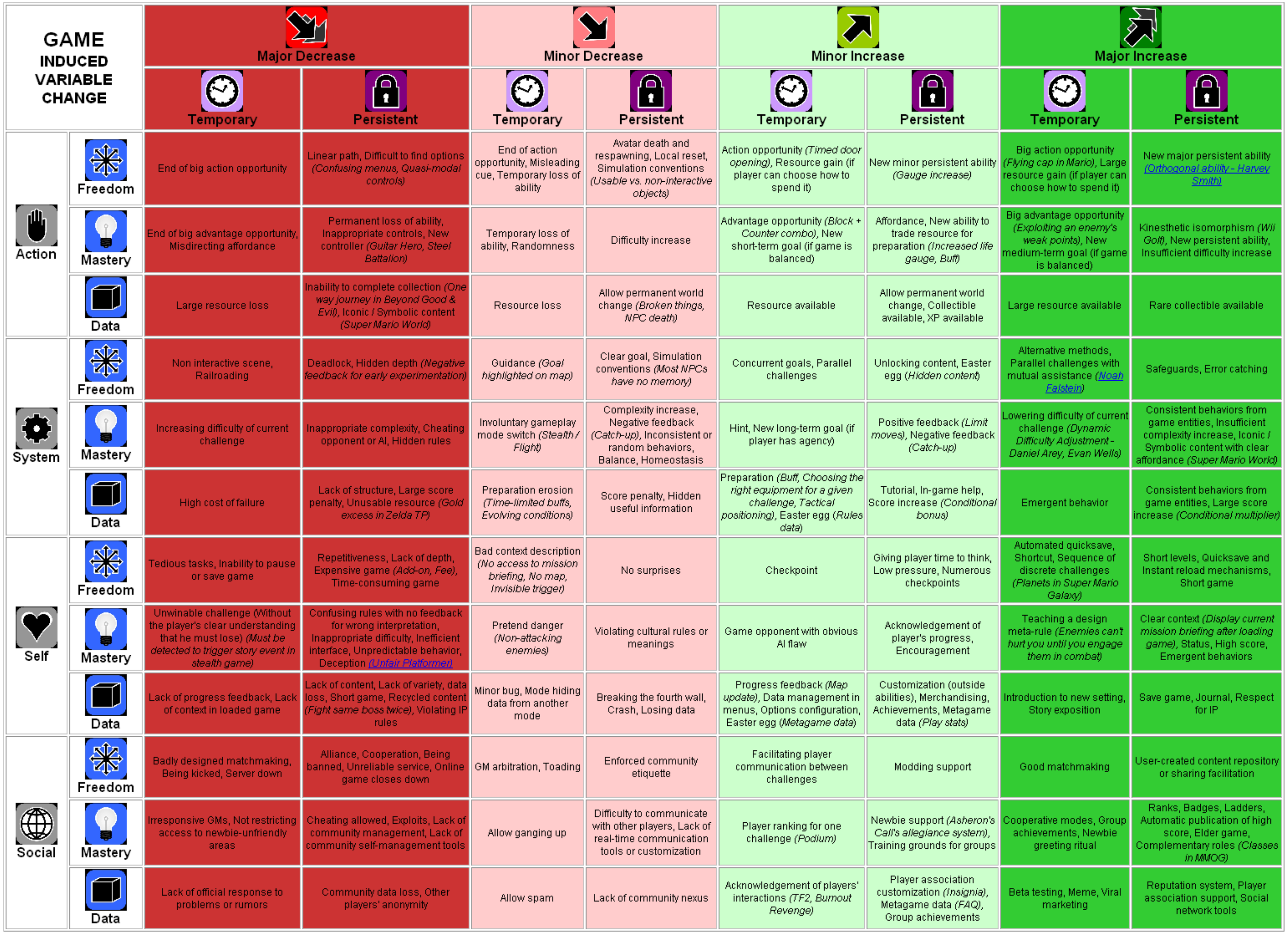
drawn, making the player truly feel part of the story.

The ultimate goal of gaming is to tell a story, to let the player experience emotion and to transfer knowledge whereever the player may be: online, on a mobile phone, at home or at school. The artwork, programming and environmental context greatly contribute to a game's success. The gadgets and other features must appeal to the user and consequently reap satisfaction.
The future vision of games is, according to Ubisoft, focused around so-called *light entertainment*: gamers who want to play for a short period of time. Adventure for all is essential, the games have to be easily playable and yet provide the same challenge and experience as complex multiplayer games. The online location of games will become more important. Moreover, new ways of interaction are introduced, such as the Nintendo WII, providing ever new challenges and variety.

However, in primitive games, the true essence of a game's attractive power is the combination of *challenge and reward.* Everyone likes to be challenged and then get something in return. A game is merely a platform which facilitates this desire. Learning in a game happens indirectly by means of active participation, not by reading. It is a Pavlov principle: something is fun to do and this stimulates one to improve one's skills all the time.

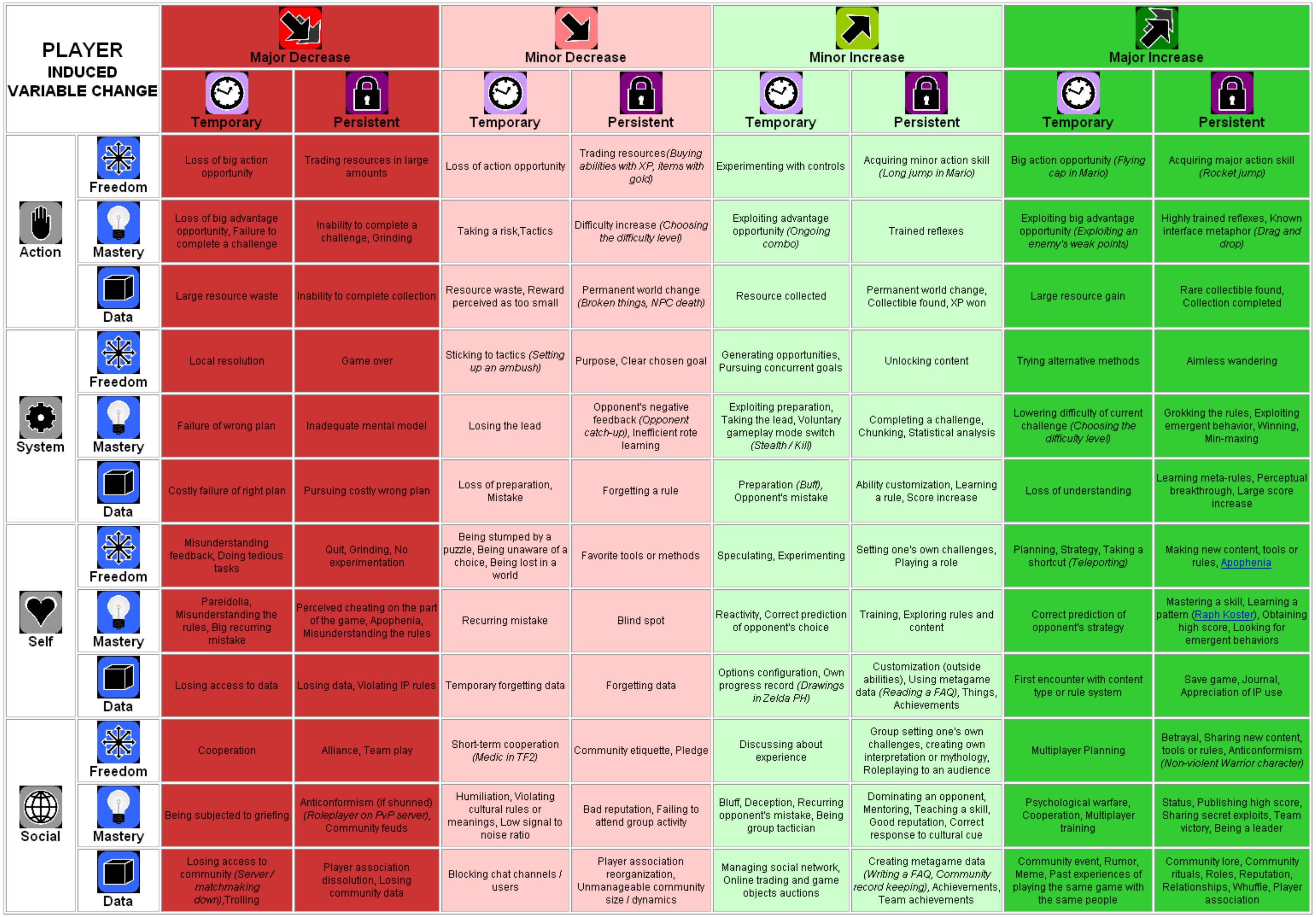
Appendix B

Game variable tables Bura



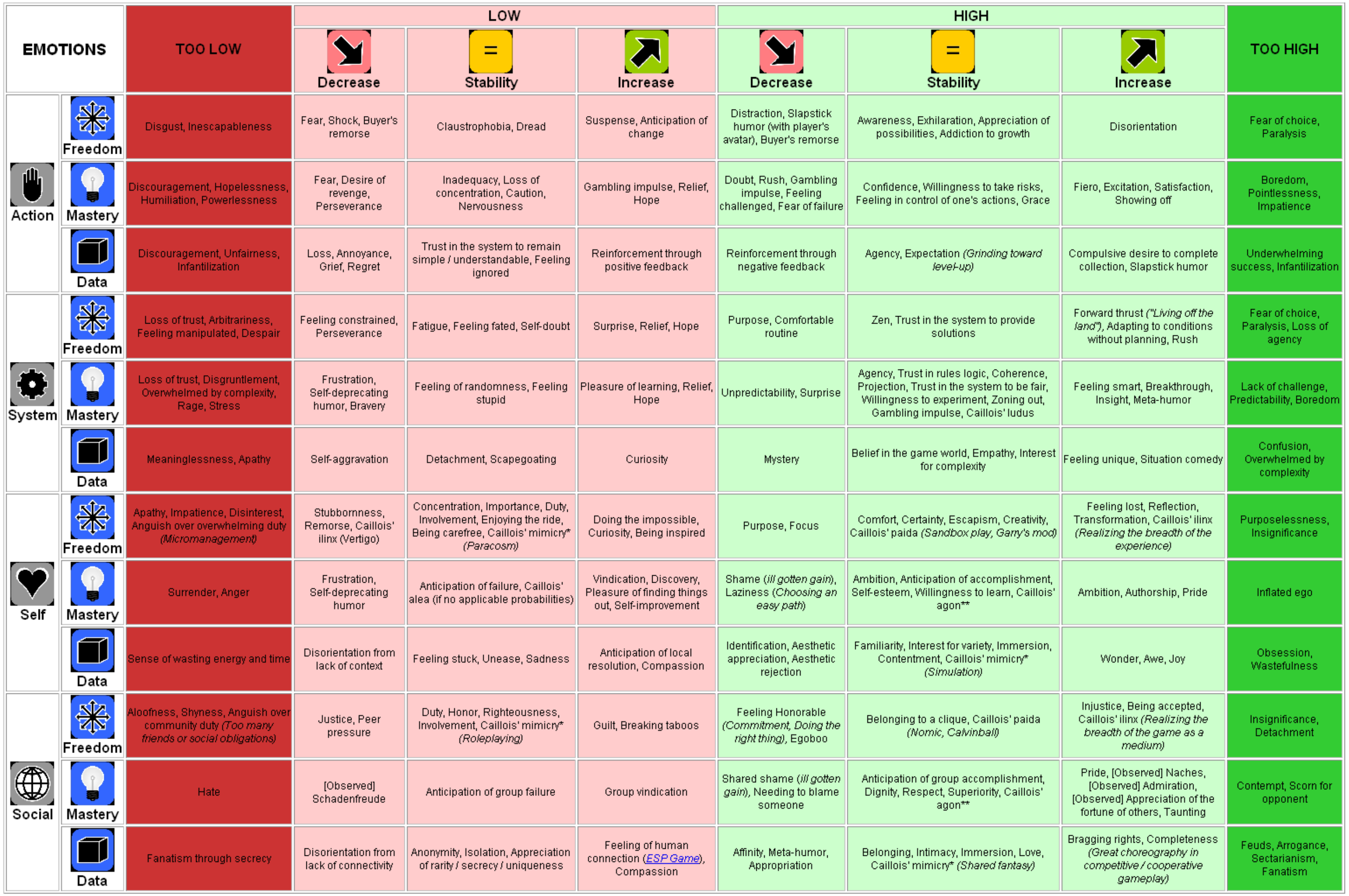
GAME INDUCED VARIABLE CHANGE		Major Decrease		Minor Decrease		Minor Increase		Major Increase	
		Temporary	Persistent	Temporary	Persistent	Temporary	Persistent	Temporary	Persistent
Action	Freedom	End of big action opportunity	Linear path, Difficult to find options *(Confusing menus, Quasi-modal controls)*	End of action opportunity, Misleading cue, Temporary loss of ability	Avatar death and respawning, Local reset, Simulation conventions *(Usable vs. non-interactive objects)*	Action opportunity *(Timed door opening)*, Resource gain (if player can choose how to spend it)	New minor persistent ability *(Gauge increase)*	Big action opportunity *(Flying cap in Mario)*, Large resource gain (if player can choose how to spend it)	New major persistent ability *(Orthogonal ability - Harvey Smith)*
	Mastery	End of big advantage opportunity, Misdirecting affordance	Permanent loss of ability, Inappropriate controls, New controller *(Guitar Hero, Steel Battalion)*	Temporary loss of ability, Randomness	Difficulty increase	Advantage opportunity *(Block + Counter combo)*, New short-term goal (if game is balanced)	Affordance, New ability to trade resource for preparation *(Increased life gauge, Buff)*	Big advantage opportunity *(Exploiting an enemy's weak points)*, New medium-term goal (if game is balanced)	Kinesthetic isomorphism *(Wii Golf)*, New persistent ability, Insufficient difficulty increase
	Data	Large resource loss	Inability to complete collection *(One way journey in Beyond Good & Evil)*, Iconic / Symbolic content *(Super Mario World)*	Resource loss	Allow permanent world change *(Broken things, NPC death)*	Resource available	Allow permanent world change, Collectible available, XP available	Large resource available	Rare collectible available
System	Freedom	Non interactive scene, Railroading	Deadlock, Hidden depth *(Negative feedback for early experimentation)*	Guidance *(Goal highlighted on map)*	Clear goal, Simulation conventions *(Most NPCs have no memory)*	Concurrent goals, Parallel challenges	Unlocking content, Easter egg (*Hidden content*)	Alternative methods, Parallel challenges with mutual assistance (*Noah Falstein*)	Safeguards, Error catching
	Mastery	Increasing difficulty of current challenge	Inappropriate complexity, Cheating opponent or AI, Hidden rules	Involuntary gameplay mode switch *(Stealth / Flight)*	Complexity increase, Negative feedback *(Catch-up)*, Inconsistent or random behaviors, Balance, Homeostasis	Hint, New long-term goal (if player has agency)	Positive feedback *(Limit moves)*, Negative feedback *(Catch-up)*	Lowering difficulty of current challenge *(Dynamic Difficulty Adjustment - Daniel Arey, Evan Wells)*	Consistent behaviors from game entities, Insufficient complexity increase, Iconic / Symbolic content with clear affordance *(Super Mario World)*
	Data	High cost of failure	Lack of structure, Large score penalty, Unusable resource *(Gold excess in Zelda TP)*	Preparation erosion *(Time-limited buffs, Evolving conditions)*	Score penalty, Hidden useful information	Preparation *(Buff, Choosing the right equipment for a given challenge, Tactical positioning)*, Easter egg (*Rules data*)	Tutorial, In-game help, Score increase *(Conditional bonus)*	Emergent behavior	Consistent behaviors from game entities, Large score increase *(Conditional multiplier)*
Self	Freedom	Tedious tasks, Inability to pause or save game	Repetitiveness, Lack of depth, Expensive game *(Add-on, Fee)*, Time-consuming game	Bad context description *(No access to mission briefing, No map, Invisible trigger)*	No surprises	Checkpoint	Giving player time to think, Low pressure, Numerous checkpoints	Automated quicksave, Shortcut, Sequence of discrete challenges *(Planets in Super Mario Galaxy)*	Short levels, Quicksave and Instant reload mechanisms, Short game
	Mastery	Unwinable challenge (Without the player's clear understanding that he must lose) *(Must be detected to trigger story event in stealth game)*	Confusing rules with no feedback for wrong interpretation, Inappropriate difficulty, Inefficient interface, Unpredictable behavior, Deception *(Unfair Platformer)*	Pretend danger *(Non-attacking enemies)*	Violating cultural rules or meanings	Game opponent with obvious AI flaw	Acknowledgement of player's progress, Encouragement	Teaching a design meta-rule *(Enemies can't hurt you until you engage them in combat)*	Clear context *(Display current mission briefing after loading game)*, Status, High score, Emergent behaviors
	Data	Lack of progress feedback, Lack of context in loaded game	Lack of content, Lack of variety, data loss, Short game, Recycled content *(Fight same boss twice)*, Violating IP rules	Minor bug, Mode hiding data from another mode	Breaking the fourth wall, Crash, Losing data	Progress feedback *(Map update)*, Data management in menus, Options configuration, Easter egg (*Metagame data*)	Customization (outside abilities), Merchandising, Achievements, Metagame data *(Play stats)*	Introduction to new setting, Story exposition	Save game, Journal, Respect for IP
Social	Freedom	Badly designed matchmaking, Being kicked, Server down	Alliance, Cooperation, Being banned, Unreliable service, Online game closes down	GM arbitration, Toading	Enforced community etiquette	Facilitating player communication between challenges	Modding support	Good matchmaking	User-created content repository or sharing facilitation
	Mastery	Irresponsive GMs, Not restricting access to newbie-unfriendly areas	Cheating allowed, Exploits, Lack of community management, Lack of community self-management tools	Allow ganging up	Difficulty to communicate with other players, Lack of real-time communication tools or customization	Player ranking for one challenge *(Podium)*	Newbie support *(Asheron's Call's allegiance system)*, Training grounds for groups	Cooperative modes, Group achievements, Newbie greeting ritual	Ranks, Badges, Ladders, Automatic publication of high score, Elder game, Complementary roles *(Classes in MMOG)*
	Data	Lack of official response to problems or rumors	Community data loss, Other players' anonymity	Allow spam	Lack of community nexus	Acknowledgement of players' interactions *(TF2, Burnout Revenge)*	Player association customization *(Insignia)*, Metagame data *(FAQ)*, Group achievements	Beta testing, Meme, Viral marketing	Reputation system, Player association support, Social network tools

Figure B.1: Game induced variable change



PLAYER INDUCED VARIABLE CHANGE		Major Decrease		Minor Decrease		Minor Increase		Major Increase	
		Temporary	Persistent	Temporary	Persistent	Temporary	Persistent	Temporary	Persistent
Action	Freedom	Loss of big action opportunity	Trading resources in large amounts	Loss of action opportunity	Trading resources*(Buying abilities with XP, items with gold)*	Experimenting with controls	Acquiring minor action skill *(Long jump in Mario)*	Big action opportunity *(Flying cap in Mario)*	Acquiring major action skill *(Rocket jump)*
	Mastery	Loss of big advantage opportunity, Failure to complete a challenge	Inability to complete a challenge, Grinding	Taking a risk,Tactics	Difficulty increase *(Choosing the difficulty level)*	Exploiting advantage opportunity *(Ongoing combo)*	Trained reflexes	Exploiting big advantage opportunity *(Exploiting an enemy's weak points)*	Highly trained reflexes, Known interface metaphor *(Drag and drop)*
	Data	Large resource waste	Inability to complete collection	Resource waste, Reward perceived as too small	Permanent world change *(Broken things, NPC death)*	Resource collected	Permanent world change, Collectible found, XP won	Large resource gain	Rare collectible found, Collection completed
System	Freedom	Local resolution	Game over	Sticking to tactics *(Setting up an ambush)*	Purpose, Clear chosen goal	Generating opportunities, Pursuing concurrent goals	Unlocking content	Trying alternative methods	Aimless wandering
	Mastery	Failure of wrong plan	Inadequate mental model	Losing the lead	Opponent's negative feedback *(Opponent catch-up)*, Inefficient rote learning	Exploiting preparation, Taking the lead, Voluntary gameplay mode switch *(Stealth / Kill)*	Completing a challenge, Chunking, Statistical analysis	Lowering difficulty of current challenge *(Choosing the difficulty level)*	Grokking the rules, Exploiting emergent behavior, Winning, Min-maxing
	Data	Costly failure of right plan	Pursuing costly wrong plan	Loss of preparation, Mistake	Forgetting a rule	Preparation *(Buff)*, Opponent's mistake	Ability customization, Learning a rule, Score increase	Loss of understanding	Learning meta-rules, Perceptual breakthrough, Large score increase
Self	Freedom	Misunderstanding feedback, Doing tedious tasks	Quit, Grinding, No experimentation	Being stumped by a puzzle, Being unaware of a choice, Being lost in a world	Favorite tools or methods	Speculating, Experimenting	Setting one's own challenges, Playing a role	Planning, Strategy, Taking a shortcut *(Teleporting)*	Making new content, tools or rules, Apophenia
	Mastery	Pareidolia, Misunderstanding the rules, Big recurring mistake	Perceived cheating on the part of the game, Apophenia, Misunderstanding the rules	Recurring mistake	Blind spot	Reactivity, Correct prediction of opponent's choice	Training, Exploring rules and content	Correct prediction of opponent's strategy	Mastering a skill, Learning a pattern (Raph Koster), Obtaining high score, Looking for emergent behaviors
	Data	Losing access to data	Losing data, Violating IP rules	Temporary forgetting data	Forgetting data	Options configuration, Own progress record *(Drawings in Zelda PH)*	Customization (outside abilities), Using metagame data *(Reading a FAQ)*, Things, Achievements	First encounter with content type or rule system	Save game, Journal, Appreciation of IP use
Social	Freedom	Cooperation	Alliance, Team play	Short-term cooperation *(Medic in TF2)*	Community etiquette, Pledge	Discussing about experience	Group setting one's own challenges, creating own interpretation or mythology, Roleplaying to an audience	Multiplayer Planning	Betrayal, Sharing new content, tools or rules, Anticonformism *(Non-violent Warrior character)*
	Mastery	Being subjected to griefing	Anticonformism (if shunned) *(Roleplayer on PvP server)*, Community feuds	Humiliation, Violating cultural rules or meanings, Low signal to noise ratio	Bad reputation, Failing to attend group activity	Bluff, Deception, Recurring opponent's mistake, Being group tactician	Dominating an opponent, Mentoring, Teaching a skill, Good reputation, Correct response to cultural cue	Psychological warfare, Cooperation, Multiplayer training	Status, Publishing high score, Sharing secret exploits, Team victory, Being a leader
	Data	Losing access to community *(Server / matchmaking down)*,Trolling	Player association dissolution, Losing community data	Blocking chat channels / users	Player association reorganization, Unmanageable community size / dynamics	Managing social network, Online trading and game objects auctions	Creating metagame data *(Writing a FAQ, Community record keeping)*, Achievements, Team achievements	Community event, Rumor, Meme, Past experiences of playing the same game with the same people	Community lore, Community rituals, Roles, Reputation, Whuffie, Player association

Figure B.2: Player induced variable change



EMOTIONS		TOO LOW	LOW			HIGH			TOO HIGH
			↘ Decrease	= Stability	↗ Increase	↘ Decrease	= Stability	↗ Increase	
Action	Freedom	Disgust, Inescapableness	Fear, Shock, Buyer's remorse	Claustrophobia, Dread	Suspense, Anticipation of change	Distraction, Slapstick humor (with player's avatar), Buyer's remorse	Awareness, Exhilaration, Appreciation of possibilities, Addiction to growth	Disorientation	Fear of choice, Paralysis
	Mastery	Discouragement, Hopelessness, Humiliation, Powerlessness	Fear, Desire of revenge, Perseverance	Inadequacy, Loss of concentration, Caution, Nervousness	Gambling impulse, Relief, Hope	Doubt, Rush, Gambling impulse, Feeling challenged, Fear of failure	Confidence, Willingness to take risks, Feeling in control of one's actions, Grace	Fiero, Excitation, Satisfaction, Showing off	Boredom, Pointlessness, Impatience
	Data	Discouragement, Unfairness, Infantilization	Loss, Annoyance, Grief, Regret	Trust in the system to remain simple / understandable, Feeling ignored	Reinforcement through positive feedback	Reinforcement through negative feedback	Agency, Expectation *(Grinding toward level-up)*	Compulsive desire to complete collection, Slapstick humor	Underwhelming success, Infantilization
System	Freedom	Loss of trust, Arbitrariness, Feeling manipulated, Despair	Feeling constrained, Perseverance	Fatigue, Feeling fated, Self-doubt	Surprise, Relief, Hope	Purpose, Comfortable routine	Zen, Trust in the system to provide solutions	Forward thrust *("Living off the land")*, Adapting to conditions without planning, Rush	Fear of choice, Paralysis, Loss of agency
	Mastery	Loss of trust, Disgruntlement, Overwhelmed by complexity, Rage, Stress	Frustration, Self-deprecating humor, Bravery	Feeling of randomness, Feeling stupid	Pleasure of learning, Relief, Hope	Unpredictability, Surprise	Agency, Trust in rules logic, Coherence, Projection, Trust in the system to be fair, Willingness to experiment, Zoning out, Gambling impulse, Caillois' ludus	Feeling smart, Breakthrough, Insight, Meta-humor	Lack of challenge, Predictability, Boredom
	Data	Meaninglessness, Apathy	Self-aggravation	Detachment, Scapegoating	Curiosity	Mystery	Belief in the game world, Empathy, Interest for complexity	Feeling unique, Situation comedy	Confusion, Overwhelmed by complexity
Self	Freedom	Apathy, Impatience, Disinterest, Anguish over overwhelming duty *(Micromanagement)*	Stubbornness, Remorse, Caillois' ilinx (Vertigo)	Concentration, Importance, Duty, Involvement, Enjoying the ride, Being carefree, Caillois' mimicry* *(Paracosm)*	Doing the impossible, Curiosity, Being inspired	Purpose, Focus	Comfort, Certainty, Escapism, Creativity, Caillois' paida *(Sandbox play, Garry's mod)*	Feeling lost, Reflection, Transformation, Caillois' ilinx *(Realizing the breadth of the experience)*	Purposelessness, Insignificance
	Mastery	Surrender, Anger	Frustration, Self-deprecating humor	Anticipation of failure, Caillois' alea (if no applicable probabilities)	Vindication, Discovery, Pleasure of finding things out, Self-improvement	Shame (*ill gotten gain*), Laziness (*Choosing an easy path*)	Ambition, Anticipation of accomplishment, Self-esteem, Willingness to learn, Caillois' agon**	Ambition, Authorship, Pride	Inflated ego
	Data	Sense of wasting energy and time	Disorientation from lack of context	Feeling stuck, Unease, Sadness	Anticipation of local resolution, Compassion	Identification, Aesthetic appreciation, Aesthetic rejection	Familiarity, Interest for variety, Immersion, Contentment, Caillois' mimicry* *(Simulation)*	Wonder, Awe, Joy	Obsession, Wastefulness
Social	Freedom	Aloofness, Shyness, Anguish over community duty *(Too many friends or social obligations)*	Justice, Peer pressure	Duty, Honor, Righteousness, Involvement, Caillois' mimicry* *(Roleplaying)*	Guilt, Breaking taboos	Feeling Honorable *(Commitment, Doing the right thing)*, Egoboo	Belonging to a clique, Caillois' paida *(Nomic, Calvinball)*	Injustice, Being accepted, Caillois' ilinx *(Realizing the breadth of the game as a medium)*	Insignificance, Detachment
	Mastery	Hate	[Observed] Schadenfreude	Anticipation of group failure	Group vindication	Shared shame (*ill gotten gain*), Needing to blame someone	Anticipation of group accomplishment, Dignity, Respect, Superiority, Caillois' agon**	Pride, [Observed] Naches, [Observed] Admiration, [Observed] Appreciation of the fortune of others, Taunting	Contempt, Scorn for opponent
	Data	Fanatism through secrecy	Disorientation from lack of connectivity	Anonymity, Isolation, Appreciation of rarity / secrecy / uniqueness	Feeling of human connection (*ESP Game*), Compassion	Affinity, Meta-humor, Appropriation	Belonging, Intimacy, Immersion, Love, Caillois' mimicry* *(Shared fantasy)*	Bragging rights, Completeness *(Great choreography in competitive / cooperative gameplay)*	Feuds, Arrogance, Sectarianism, Fanatism

Figure B.3: Emotions associated with changes in gameplay variables

Appendix C

Aquima Game Design Workshops

C.1 Workshop session 1

On the 25th of November, 2008, the first session of a Game Design Workshop was held at Everest, in which several experts came together to work out a first conceptual design of the Aquima Modelling Game. In order to logically organise and provide meaning to the gaming concepts, I considered several game design frameworks and finally merged several to create a meaningful whole. My framework is based on the framework provided at the Serious Game Design workshop which I followed at the Technical University of Eindhoven on October 22nd, 2008, with a few modifications from the GameGame approach by Aki Järvinen [Jarv 08].

Serious Game Design Workshop

Ludus worksheet

Title

What is the title of your game?

Type of game

What type of game did you design? Use terms as “digital” or “ analog”, puzzle, simulation or strategy game, etc. Also, refer to other games, like “Pacman meets Unreal Tournament” or “Super Mario vs. SimCity”.

Players

Is it a single or a multiplayer game? If a multiplayer game, how many players could participate in one session?

Game world

Where and when does the game take place (time period, geography, etc.)? What does the world look like?

Perspective

In what type of perspective is the game played? How is it shown to the players? Computer-related terms such as 3D can be used or analog terms, like a board can be used as well.

Interaction model

How do players interact with the game? Mention in case it is digital the hardware (mouse, wii mote etc.) an in case it is analog what paraphernalia is used (dice, clock etc.). If it is a multiplayer game, describe here how players interact with each other. What is the interface going to look like?

Components

What objects are there in the game world for the player to manipulate?

Story

What is the background story? What happens in the game and how does it end? NB. Every game has a story, even Tetris. If it does not have a traditional story, describe what happens throughout the game.

Goal

What is or are the goals of the game? How does the player know when this goal or these goals are achieved (so, how does the player get **feedback** through the game)? Specify the end-condition (when the game is finished) and the victory-conditions (what makes the game competitive).

Challenges

What challenges does the player need to take to achieve the goal(s)?

Actions

What possible actions can a player perform and what game mechanics does he have for this? What does the player use to perform these actions (what are his or her **resources**)? If resources are needed to perform actions, how does player get these resources?

Rewards

What type of rewards does the player in the game (points, money, items etc.)? What can the player do with these rewards?

Rule-based system

Give a short description of the system of the game and its rules. It is recommended to make a diagram (with blocks and arrows).

Other remarks

Serious Game Design Workshop

Semiosis worksheet

Goal

What is or are the goal(s) IN your game?

Semiosis goals

What message do you want to integrate into your game?

Integration

Describe how you want to integrate your semiosis goals into your game. If significant changes are made in the concept of the game (challenges, actions, story, etc.), you have to mention these here as well.

Target group & setting

For whom is the game intentioned? Students or professionals? And *where, when* and *how* will the target group play the game?

Why

Gaming has become a bit of a hype. Due to this, it is important to think of WHY a game needs to be used for providing a message and not some other medium.

Context

Like any other medium a game has advantages and disadvantages. For serious gaming it is especially important to embed the game within a certain context. Indicate how this context would look.

Reflection

Reflection is next to practice an important aspect for providing a message. *Where* and *how* does reflection take place in your game?

Feedback

Within learning processes feedback is another important aspect. How will feedback be given in your game? Also indicate how this feedback is related to the semiosis goals (so, what is the use of the feedback to achieve the goals?).

Diversity

Some players know more about A and less of B, while the opposite is true for others. How will your game handle this “diversity”?

Other remarks

Serious Game Design Workshop

Ontology worksheet

Theme

What theme did you pick?

Focus

On which aspect of the reality of this theme do you concentrate?

Problem

Think of a problem (or make up one) related to your theme of why a game needs to be developed for it.

Actors

What actors are involved? What is their role and what are the relationships between the different actors?

System

Describe the system of your problem. Where does it start and where does it end? What aspects are involved and what aspects are not?

Process

Describe the process of what occurs in your system. What happens? When, how and why?

Other remarks

C.1.1 Transcription

Workshop Aquima Game Design, November 25th, 2008
Den Bosch, Everest, zaal Norgay.
9:30 - 12:15
Participants: Mark Mastop (Manager Product Development & Innovation), Ingrid van Baast (Consultant), Rick Fleuren (Technical Engineer), Dick van Soest (Business Engineer), Vincent Knoop Pathuis (Business Engineer)
Facilitator: Ilona Wilmont

Assignment 1: Prior to the design rounds, think of a storyline which the player has to experience while playing the game. Keep this in mind during the workshop rounds to come.
Corresponding audio file: IC_A0001.wav

The participants were divided into two groups, one of three (Mark, Ingrid, Rick) and one of two (Vincent, Dick). Each group came up with their own storyline and presented this to each other.
The following is an account of the presentations (in Dutch):

Rick:
Ons verhaal gaat erom de wereld te veroveren, of te ontdekken. De hoofdinspiratie komt van Civilizations. Je begint klein, het einddoel is dan wel om de wereld te veroveren, maar het is onduidelijk hoe je daar komt. Je hebt meerdere manieren om je doel te bereiken, je kunt bijvoorbeeld gaan samenwerken met andere volken die je tegenkomt, maar je kunt ze ook een kopje kleiner maken of dat soort dingen.

Ingrid:
Uiteindelijk moet er ook iets van competitie in zitten. Competitie met collega's vonden wij niet zo sympathiek, die zagen we meer voor de samenwerking, maar tevens proberen we ook de concurrentie te verslaan.

Mark:
Proberen iets beter en sneller op te leveren dan de concurrent zou kunnen door samenwerken of door andere dingen, en het snelle zit hem in bijvoorbeeld de deadline. Je hebt op een gegeven moment een deadline, een datum waarop iets klaar moet zijn. Je weet alleen niet van tevoren wat er klaar moet zijn, dat is namelijk de grap. En de weg er naartoe is ook niet helemaal bekend.

Rick:
Dus binnen de tijd iets opleveren waar de klant tevreden mee is.

Dick:
Bij ons kwamen er wel een beetje dezelfde ideeën op, zoiets als Spore. Dat is een spel waarbij je creatures ontwerpt, je begint heel eenvoudig en die worden in de verschillende levels steeds complexer. Op het hoofdniveau heb je zelfs echt een samenleving. Wij kwamen op het idee om in eerste instantie je requirements als creatures te nemen. In het begin praat je met business mensen, dus de creatures worden gestuurd of gemaakt door business mensen, als developer kun

je ermee interacteren. We hadden ook het idee dat je ze kunt beoordelen, een goede requirement is helder en testbaar, dus dat is zeg maar het talentenjacht aspect van creatures, hoeveel punten geef je hem? Verschillende deelnemers aan de ontwikkeling van je product hebben verschillende rollen, ieder heeft z'n eigen manier en z'n eigen redenen om creatures te beoordelen. Dat is ongeveer waar we waren. Je kunt de architectuur als een omgeving ziet waar die creatures zich in gaan ontwikkelen, maar daar hebben we nog niet echt over nagedacht, dat kwam net bij me op.

Vincent:
Wat ik daar misschien nog aan toe kan voegen is dat zo'n creature verschillende evolutiefases doormaakt. In het begin kan hij nog niet zoveel, alleen maar requirements laten definiëren, maar later moeten er natuurlijk services gebouwd worden, en interfaces en toestanden, dan gaat hij groeien. De interactie met de creature veranderd ook, misschien gaat hij er ook wel anders uitzien, dat weet ik niet.

Dick:
Het leuke van creatures is ook dat je ermee kunt interacteren, dus als je hem vraagt "wat ben je nou eigenlijk?" dan gaat hij over zichzelf vertellen. Je kunt ook vragen "wat doe je nou precies", het is eigenlijk een soort levensverhaal dat hij over zichzelf verteld. Dat is ongeveer waar we waren.

Design round 1: Ludus elements
Corresponding audio file: IC_A0002.wav

Vincent:
Zal ik een poging doen dit keer? Nou, de titel die is er al, dat is Aquima. Of Aquima Online, World of Aquima, noem maar wat. De type of game, we hebben er drie termen aan toegekent: evolutie, life en diplomacy.
Evolutie, dat bleek al een beetje uit dat verhaal, het gaat verder. Er zit een aspect in dat volgende generaties meer kunnen dan vorige. Life omdat het een heel leven van zo'n creature beschrijft, ze hebben een begin en een eind, een geboorte en een dood, wellicht. Diplomacy vanwege ook de interactie met elkaar, het onderhandelen over kan dit nu wel, kan het niet simpeler, dat soort dingen.
Het aantal spelers die actief binnen het project meedoen zullen er waarschijnlijk, nouja, hoeveel mensen zitten er op een project, zeg een man of 10, 20, maar er is wel een modus daarbuiten waarbij je kan communiceren over projecten heen. Die andere projecten kunnen natuurlijk niet zomaar binnen jouw project kijken maar je kunt wel communiceren. Dan kun je zelfs bepaalde dingen openzetten, je kunt bepaalde requirements eens bekijken, of je kunt bepaalde beschrijvingen voor naar buiten toe beschikbaar stellen, en een aantal kun je natuurlijk met elkaar communiceren.
Over de game wereld waren we het samen nog niet helemaal eens, maar in elk geval een geabstraheerde, werkelijke wereld. Niet te lollig, wel een beetje gerelateerd aan de werkelijke wereld, wellicht zelfs de zakelijke wereld. Met abstractie kun je zorgen dat het iets minder lollig wordt, maar aan de andere kant moet het wel iets werkelijks hebben, iets herkenbaars, gekoppeld aan het perspectief. Als je tot in het extreme abstraheert zou een creature zelfs nog wel een vierkan-

tje kunnen zijn, maar dat is wel erg saai, dus het gaat om een 2D omgeving, of 3D of 4D als je de tijd er ook bij betrekt. Het meest waarschijnlijke is toch 3D, omdat dat het leukst oogt.
Het interactie model, nou, Wii mote vonden we wat ver gaan, maar de muis toch zeker wel. Je zal verschillende objecten die er aan bod komen moeten kunnen bedienen, aanklikken en dan vervolgens in natuurlijke taal de requirement beschrijven. Bijvoorbeeld, je klikt aan 'New Requirement', of 'New Creature', en dan moet je iets intypen en hem aldus beschrijven. De interactie vind in ieder geval primair plaats via die objecten, dus je vult iets in, en iemand anders die bekijkt dat en doet daar iets mee. Daarnaast kun je natuurlijk ook nog rechtstreeks met elkaar communiceren, maar dat is meer om iets te overleggen.
De componenten zijn dus de creatures, daarnaast hadden we nog bedacht dat je waarschijnlijk iets van muren en hekken nodig hebt om dingen af te scheiden, of dat je de muren juist gebruikt om dingen te koppelen, creatures kunnen verband hebben met elkaar, kamers om te categoriseren, levels om aan te geven dat een creature een requirements creature is die in de fase requirement is, later wordt er een service creature gemaakt, die komt een een latere fase en die kan afgerond worden op een hele andere manier.
Het verhaal hebben we natuurlijk al een beetje toegelicht. Het eindigt als je alle soorten creatures hebt die nodig zijn, ze zijn volwassen en ze hebben een stabiele relatie met elkaar. Het enige wat je nog mist is: "waar komt dan de requirement voor de creature vandaan?", "waarom heb je creatures nodig?", maar dat is waarschijnlijk een stap voorafgaand aan het creëren van die creatures, dan heb je iemand die gaat zeggen "ik heb die en die creatures nodig", je allereerste project initialisatie eigenlijk, op een hoog niveau, vergelijkbaar met je level editor.
Het doel op een hoog niveau is natuurlijk gewoon al die creatures maken en zorgen dat ze een stabiele relatie hebben. Per component is er een progress bar waarin hun vooruitgang wordt getoond binnen het level. De requirements zijn in eerste instantie basaal beschreven, en dan later zijn ze helemaal beschreven, op een gegeven moment zijn ze gereviewed.
Uitdagingen zijn er voornamelijk in de vorm van achievements. Je kunt je voorstellen dat jij op een gegeven moment tien creatures gecreëerd hebt, dan krijg je de achievement 'je hebt tien creatures gecreëerd, daarmee ben je een Moderate Creature Creator, later wordt je dan Master Creature Creator.

Dick:
Je kunt je ook voorstellen dat de creatures ook echt beoordeeld worden, en als jouw requirements creatures vaak goed beoordeeld worden, dan ben je dus kennelijk een hele goede, daar krijg je weer een reward voor.

Vincent:
Dat beperkt zich natuurlijk niet alleen maar die dingen, je kunt de gekste dingen bedenken, je kunt een hele leuk insteek nemen, bijvoorbeeld "jij bent degene die nu vijf uur lang zonder pauze heeft zitten werken", dat is dan een achievement, een beetje absurde dingen, en daar worden dan ook weer titels aan gekoppeld. Die titels kun je terugzien in je chatwindow, je kunt kiezen, vandaag ben ik Master Creature Creator, dan ziet iedereen dat je dat bent.
Challenges is een beetje wat we net zeiden.
Actions, dat was een moeilijke. In elk geval hebben we natuurlijk het creëren

van een creature, het verder door zijn progress heen halen door acties uit te voeren, dat verschilt een beetje met wat je moet doen. Het definiëren van een requirement, het maken van een service, het erin hangen van een service, weet ik wat.
Resources, ik denk niet dat we echt resources hebben in de vorm van graan of dat soort categorieën. Dan hadden we hier genoemd, een creature is een resource, een relatie, ja in zekere zin wel, en dan hebben we ook nog precondities, dat er wel aan een preconditie voldaan moet worden. Dus je moet de requirement reviewen voordat hij af is.
Rules is nog wel heel summier. Je creature moet levensfases door en ze moeten relaties krijgen met anderen. Ik denk dat dit een concept is dat je flink uit moet werken als je er echt mee aan de gang gaat.

Mark:
Leuk spel, maar jammer! Want... De wereld ziet er namelijk helemaal anders uit. Wij hebben ook een spel bedacht, ons spel heet Aquimatization, het is een massive digital strategy simulation 3D puzzle waarbij de groepgrootte minimaal 3 en maximaal 7 is. Eigenlijk moet je alle teams groter dan 7 opsplitsen in kleinere teams, anders kunnen mensen niet op een goede manier samenwerken, daarom hebben wij gekozen voor een maximum van 7 personen.
Als je start zit je wat ons betreft op een onbewoond eiland, relaxed, of op een andere plek die jou een hoop rust geeft want voordat je allerlei stappen kunt zetten moet je op een bepaalde manier even rustig na kunnen denken. Dat kan zijn... wij vonden een onbewoond eiland leuk, maar het zou ook in een ruimte kunnen zijn waarin je gaat bedenken waar je vandaag allemaal heen gaat. Als team bepaal je eigenlijk een beetje de setting waarin je je lekker gaat voelen.
Uiteindelijk, als je gaat kijken en je gaat met elkaar werken, dan kijk je van bovenaf op het bord, en naar mate de tijd verstrekt zal het ook allemaal een beetje vager worden. Je ziet dus ongeveer de contouren van waar je heen wilt, want je weet wel grofweg wat je wilt gaan doen, maar hoe zich dat voor zal doen zal zich naderhand voordoen. We hebben daar ook een aparte manier van werken mee, misschien kennen jullie dat Microsoft Service ding, zo'n grote tafel waar je met z'n allen omheen kunt staan en boven kunt hangen. Vandaar dus ook een iets kleiner team, je gaat echt gezamenlijk daaraan werken en je schuift ook gewoon afhankelijk van wat je gaat doen. Zo heb je daar een koffiecorner en een bar, waar je even kunt relaxen, à la scrum, ik heb nu even behoefte aan koffie maar ook een stukje entertainment, à la YouTube, dingen die wij doen kunnen we gelijk weer op YouTube neerzetten, dat in andere werelden weer bekeken kan worden, een soort broadcasting mechanisme. De bedoeling is dat mensen continue weten waar we mee bezig zijn en welke stappen we gaan zetten, zodat het los van een stuk uitdaging ook een stuk entertainment biedt. Wat je eigenlijk gaat doen is bouwen, van kleine dingen tot complete nederzettingen neerzetten, het hoofddoel. Een nederzetting kan een project zijn, of een oplevering, en uiteindelijk is dat ook grofweg waar je het mee moet doen. Daar heb je een voertuig voor nodig, een soort vehikel waarmee je het eiland gaat verlaten, je hebt gereedschap nodig om die gebouwen te gaan bouwen. Belangrijk is dat we geld nodig hebben, we hebben een economie. De burgemeester of de opdrachtgever, die uiteindelijk blij is of niet blij is met de nederzetting, is ook actief betrokken in onze wereld, die zit erin, doet ook mee, kijkt of hij het mooi of niet mooi vind, of fijn. Vandaar ook dat naar mate de tijd verstrekt de hele

wereld ook grijzer en schimmiger wordt, dat weten we gewoon nog niet. We hebben wel een idee, maar we weten niet waar we uiteindelijk heen willen. Belangrijk is ook dat we aan de kenniseconomie gaan werken. Innoveren, stappen zetten en daarmee gaan werken.

Ingrid:
Het is iteratief, we gaan ook continue onderhandelen met die contractgever.

Mark:
Heel belangrijk is dat wij niet weten waar het eindpunt is. Dus we gaan niet van tevoren met die bouwer helemaal bepalen wat wij precies voor hem gaan maken. Het enige dat we weten is dat we een bepaalde tijd hebben, en daarmee een bepaalde capaciteit en geld, en dat we grofweg weten wat we willen hebben: een nederzetting. Of die groen of geel wordt, of dat rode stenen of dat het beton wordt, of dat het stenen worden, dat weten we van tevoren niet, het enige wat we weten is wat we gaan doen. Het belangrijkste wat we gaan doen is dat we een plek gaan bouwen om een feest te vieren. Op het moment dat de nederzetting klaar is, is er altijd een heel groot Asterix en Obelix feest. Namelijk, na een tijdje goed toeven moet er een groot feest zijn. Het doel is dus de bouw van een nederzetting, dat kan later uitgroeien tot meerdere nederzettingen of een metropool, dat is een beetje afhankelijk van wat de burgemeester voor ideëen heeft voor zijn stad, dat is de analogie met een project. Als je gaat kijken als klant, die hebben allemaal kleine nederzettinkjes, maar je hebt ook klanten die echt gaan bouwen aan een metropool, de grootte weten we van tevoren niet. Dat kan de klant bepalen, ik wil nu daar een nederzetting gaan bouwen, en deze nederzetting gaan uitbouwen tot een metropool. De klant, burgemeester of de keizer, die bedenkt de events, die kan zeggen 'er komt een storm', of 'we gaan dingen doen' of die komt in één keer inbreken. Dat kunnen we niet weten. We gaan ook niet van tevoren proberen dat helemaal af te dichten, maar dat is ook het proces waarmee we de klant op dat ogenblik meten.
De organisatievorm bepaald heel sterk het verloop van het spel. Heb ik een dictatuur, dan bepaald de burgemeester gewoon 'zo zit het', of een democratie, dan gaat het gewoon onder elkaar in een gezellige harmonie.

Ingrid:
Dat is ook een beetje de cultuur van je klant die je daarin weerspiegelt. En je vergeet nog iets, die events, daar heb je zelf ook invloed op. Naar mate je meer met die klant communiceert zul je voor minder vervelende verrassingen komen te staan. Naar mate je dat vergeet, en minder aandacht besteed aan je klant, zal die met meer vervelende events komen gedurende het spel. Als het ware wordt je daarvoor beboet, en als je veel met de klant communiceert wordt je ervoor beloond in de vorm dat alles soepel verloopt.

Mark:
Ja, want die klant die zit gewoon in het spel, die ziet dat dagelijks, net zoals toen ik mijn huis ging bouwen, toen liep ik dagelijks over de bouw, als ik het niet eens was zei ik 'nou je moet dat toch even anders doen, dat vind ik niet fijn', waardoor je uiteindelijk grofweg weet hoe het huis gaat worden, maar ik wil exact krijgen wat ik wilde hebben.
Doel, er is maar één doel, en dat is winst. En op het moment dat we winst

hebben, hebben we geld, of hebben we Aqurijnen om een feest te geven. Winst kan zijn als we sneller klaar zijn dan we hadden gezegd, het kan zijn dat de klant veel tevredener is omdat hij veel meer heeft dan hij had. Uiteindelijk is het doel dat de burgemeester winst maakt op zijn bouwproject, en dat hij dat aan ons geeft zodat wij daar een enorm feest van kunnen gaan bouwen. Uitdagingen op dat ogenblik zijn of eerder aan de verwachtingen voldoen, namelijk zorgen dat je eerder klaar bent...

Dick:
Heb je dan een koper voor die gebouwen? Die winst, waar komt die vandaan?

Mark:
Van de opdrachtgever, die betaald gewoon. En je hebt meer winst als het eerder klaar is. Ik zei dat het 1 december klaar is, maar het is al half november klaar, en dan hebben we twee weken winst, dat is gewoon keihard geld, dan kan ik een groter feest organiseren. Kijk, iets op tijd af hebben, dat is iets wat je met de klant afspreekt, dat kan niet de uitdaging zijn. Dan krijg je een normaal, regulier feestje, maar als het er eerder is, of mooier is, of fijner is, nou, dan is het feest wat groter want de klant is ook meer gelukkig.
De speler die uiteindelijk in het spel zit, die evolueert. Elke keer kan het zijn dat je andere werelden gaat ontdekken, het is de bedoeling dat de speler groeit in zijn mogelijkheden. De speler begint gewoon altijd als metselaar, of als schilder. Op een gegeven moment kan hij gaan opperen, kan hij drie schilders aansturen, kan hij ook een beetje werk verdelen. Op een gegeven moment gaat hij richting aannemer, of misschien wel architect. Dus de grap is uiteindelijk, in de analogie van de bouwwereld, kun je gewoon stappen zetten binnen de organisatie, en naar mate je meer stappen hebt gezet en een hogere verantwoordelijkheid hebt kun je meer geld vergaren en kun je grotere dingen doen en dan kun je uiteindelijk mee gaan beslissen.
Ben ik nog iets vergeten?

Ingrid:
Nee, niet echt.... Nouja, je hebt nog niet verteld wanneer de wereld nou veroverd is..

Mark:
Ja, dat is nooit, dat kan niet. Het is namelijk Pinky and the Brain... kennen jullie die tekenfilm? Nee? Nou, Pinky and the Brain zijn twee muizen, en die hebben één doel en dat is de wereld veroveren. En elke ochtend vraagt de kleine aan de grote, Pinky vraagt aan The Brain ”What are we going to do today?” ”Today we're gonna conquer the world”. En dan krijg je een heel verhaaltje, en aan het eind van de dag is het natuurlijk mislukt, want ja, je kunt de wereld niet zomaar veroveren. En dan vraagt hij ”Wat gaan we morgen doen?” ”Morgen gaan wij de wereld veroveren”. En dat gaat zo alsmaar door.

Rick:
De initiële resources, dat is misschien de tijd die je hebt tot aan een bepaalde deadline, en als je met de klant gaat onderhandelen, zeg maar, we kunnen nu wel een gebouw neerzetten, maar we kunnen ook bijbouwen aan een ander gebouw, en dat kost dan minder geld, en dan kan de klant zelf bepalen welke richting we

op gaan.

Ingrid:
Hij kan ook besluiten om op een gegeven moment meer resources te geven, als hij dingen heel anders wil zien.

Dick:
Hoe weet je dat je de goede richting op gaat? Is daar een reward voor?

Mark:
De reward is de klant, de klant is tevreden.

Dick:
Dus de burgemeester is de tevredenheidsmeter.

Mark:
Ja.

Ingrid:
Je bent niet het enige project dat daar bezig is in die wereld. Met een beetje mazzel zitten er daar nog meer die...

Mark:
Vandaar dat we ook gekozen hebben voor wat kleinere teams die zelf aan het bouwen zijn.

Ingrid:
En je streeft gewoon naar de meest tevreden klant.

Design round 2: Semiosis elements
Corresponding audio file: IC_A0003.wav

Ingrid:
Ik had als belangrijkste doel van ons spel dat je leert om te blijven onderhandelen met je klant, in contact te blijven met je klant zodat je uiteindelijk kan opleveren wat die klant echt bedoeld had en niet zozeer wat hij initieel specificeert. Als tweede doel hadden we nog innovatie eigenlijk op je project en bijvoorbeeld ook je projectaanpak. Naast economie kennen we ook zoiets als politiek in het spel en verschillende politieke stromingen die representeren eigenlijk verschillende culturen van de klant en als jij dezelfde aanpak hanteert in een monarchie of in een democratie dan kan dat een andere uitwerking hebben in je spel. Daar zit dus een bepaald lerend effect in, als ik bepaalde handelingen doen en ik ga op een bepaalde manier met die klant om en ik neem bepaalde beslissingen, dan kan het in de ene cultuur helemaal mis gaan en in een andere cultuur kan het hartstikke goed gaan. Dat waren eigenlijk de belangrijkste doelen, dat je daarvan leert en dat je dat ook met elkaar kan delen.
Hoe dat dan geïntegreerd zit in het spel heb ik al een beetje aangegeven. Eigenlijk, wat voor een cultuur die klant heeft, of wat voor politieke stroming je kiest in je spel dat beïnvloed de reactie van het spel op de acties die jij doet. Een actie

kan verschillende uitwerkingen hebben afhankelijk van je omgeving. Daarnaast wordt je beloond, op het moment dat je je gewenste resultaat gaat halen dan zul je meer winst hebben, of als je beter presteert dan eigenlijk van je verwacht werd zul je meer winst hebben dan wat je anders zou krijgen, dus daar zit eigenlijk het onderhandelingsaspect in. Je bent alleen maar in staat om te maken wat die klant graag zou willen als je continue met hem blijft onderhandelen. Doe je dat niet dan wordt je daar eigenlijk voor bestraft en doe je dat wel dan wordt je daarvoor beloond.
Voor wie is het eigenlijk bedoeld? Het spel is voor de klant bedoeld, en voor het team, en team eigenlijk in de breedste zin van het woord. Van accountmanagers tot projectleiders, technical engineers, business engineers, eigenlijk heel Everest zeg maar. Die accountmanagers, hadden we bedacht, die gaan de wereld ontdekken. Die sturen we erop uit en die gaan allerlei nieuwe plaatsen ontdekken voor ons waar wij weer nieuwe nederzettingen kunnen bouwen. Dus die zijn er ook bij betrokken, en die komen dan terug met een rapport waarin staat 'dit is de cultuur en dit hebben we daar aangetroffen en het is voor jullie een mogelijkheid om daar een mooie nieuwe nederzetting neer te gaan zetten'.
Waarom zou je dit eigenlijk in een spelvorm gieten? Dat is eigenlijk vanwege het leereffect. Doordat je problemen uit de context kan halen, uit z'n dagelijkse context, want dit spel representeerd eigenlijk wat we toch al doen bij klanten, maar door het uit die vertrouwde context te halen wordt het probleem veel helderder. Dan wordt het veel makkelijker om daarover te praten. Dat is net als wanneer je zelf midden in een probleem zit dan helpt het vaak als je het tegen een ander verteld, die ziet gelijk waar nou precies het probleem zit. Die kan je daar veel sneller mee helpen. Dus door het uit de context te halen, daarom zouden we het in een spel willen doen.
De context zelf is eigenlijk het hele proces, vanaf het binnenhalen tot het opleveren van het project en het hele spel wat daartussen zit.
Reflectie en feedback... Nou, feedback hebben we geïnterpreteerd als wat het systeem je teruggeeft, daar zie je eigenlijk het verloop van je proces. Je merkt op een gegeven moment dat je proces niet lekker loopt doordat je klant ontevredener wordt. Je klant kan ook tevredener worden, als het goed loopt. Dat is eigenlijk de feedback die je krijgt vanuit het systeem.
Reflectie, daar gaat het spel je op zich niet bij ondersteunen, maar wat wel zo is is dat door andere keuzes te maken kun je de stemming van je klant eigenlijk beïnvloeden waardoor je dan met behulp van die feedback weer aan zelfreflectie zou kunnen doen. Als ik dit doe, dan wordt hij wel gelukkig, dus dan moeten we misschien meer van dat gaan doen.
Nou, diversiteit heb ik ook al heel even genoemd. Eigenlijk zitten er een heleboel rollen in, de accountmanager zit erin, een architect, iemand die de initiële opzet gaat maken, bouwers zitten erin die uiteindelijk die steden en die dorpen gaan bouwen, je klant zelf zit erin. Diversiteit komt niet alleen in je verschillende rollen terug, ook die verschillende culturen zitten erin. Nog meer aan diversiteit? Nee, dat was hem wel.

Dick:
Zal ik dan een poging doen? Nou, zingeving, waarom doe je dit eigenlijk? De goals in your game, wij hebben dat opgevat als subdoelen. De subdoelen worden weergegeven door de levenslust, de progress bar die je hebt per creature. Een creature kan verschillende levensfases doormaken, maar ook binnen een levens-

fase heeft het verschillende doelen die bereikt moeten worden, dat zijn eigenlijk de subdoelen die je moet nastreven, per creature, of misschien zelfs per kamer of per kudde, vervulling van het levensdoel van het creature verder zien te krijgen. Waarom zou je dit als een spel doen, waarom deze vorm? Het idee is dat je je principes van software ontwikkelen beter na gaat leven. Bijvoorbeeld, als je requirements hebt, heel vaak is het gewoon een zinnetje, okay, dit het is requirement. Maar er zit veel meer achter. En door zo'n creature ook verschillende dingen te laten doen, hij moet bijvoorbeeld zijn requirement kunnen uitleggen, als die dat niet kan is het geen goed creature, is die niet levensvatbaar. Zo dwing je eigenlijk de speler de principes van software ontwikkeling goed na te leven, de requirement moet helder zijn. Andere spelers kunnen ook oordelen over je creature. Bijvoorbeeld, het is wel een leuk creature dat je hier maakt, een leuk software component, maar de interface loopt niet lekker, dus dan geef ik een lager puntje voor dat aspect. Dus je maakt de dingen die je toch moet doen binnen software ontwikkeling expliciet.
Waarom zou je het dan in deze vorm gaan doen, nou die creatures die 'leven', tussen aanhalingstekens, dus naast het feit dat je weet dat je voor software ontwikkeling moet leven, moet je ook nog eens een keer je eigen creature kunnen onderhouden. Dus jouw idee van Tamagotchi, wat je eerst zei, ja, persoonlijk snap ik niet waarom mensen zoiets zouden gaan doen, ik zie er eigenlijk helemaal niets in, maar het gaat toch wel iets beter voor je leven, je doet het niet alleen maar voor je software ontwikkeling, je doet het ook voor je creature. Het geeft je ook een mogelijkheid om overzicht over je spel te krijgen, dat je niet alleen van je creature de levenslust ziet, maar ook van de hele kudde, van het hele spel. alle creatures in je spel. Er moet ook een zeker fun element zijn, zeker voor mensen die software ontwikkeling toch wel een beetje lastig vinden, een beetje moeilijk, er wordt iets aan toegevoegd wat het een beetje fun maakt. Dan heb je heel duidelijk waar een spel naartoe kan. Een game is voor IT'ers sowieso wel interessant, maakt het wel iets leuker. En het is er elke dag, de bedoeling is ook dat je er elke dag iets aan doet. Dat is anders dan iemand op een cursus sturen, cursus software ontwikkeling, want dat doe je één keer, of een maand lang iedere avond, of één keer per week, en dan zakt het weer weg. Nee, er moet iedere dag iets gedaan worden.
Context.. ook al komt de context hierbij voor, software ontwikkeling, ook de context bij de klant waarbij je niet de mogelijkheid hebt iedereen elke dag te zien, maar ze kunnen wel zelf interactie met dat spel opstarten en op die manier bijdragen op een moment dat het hun zelf goed uitkomt. Bedenk ook dat je door te visualiseren, bijvoorbeeld een lastige requirement of een lastige software component, die zou je een ander uiterlijk kunnen geven door dat te visualiseren kom je dus sneller in je context, dan zie je 'o ja, dat is die, dat was ik weer vergeten, die context, daar wordt die lastig, dat is de context waar ik in moet zitten'. Die twee vragen over reflection en feedback, eigenlijk heb je die door het in deze vorm te gieten gratis meegekregen, door die life bars zie je vanzelf, dat gaat dan meer in op diversiteit, maar door het beloningssysteem dat je inbouwt zie je meteen wat voor diversiteit je allemaal hebt. Eén persoon kan misschien meerdere rollen spelen, nou, dat zie je ook. Dan krijg je meerdere versies.
Nog stukjes gemist? Feedback... Dat is eigenlijk min of meer ingebouwd.

Vincent:
Misschien nog even over die groepen, of die targets... Kijk, wij vinden het werk

voor het grootste deel toch wel vrij leuk, want we werken hier al een tijdje. Maar vooral voor partnering is het maar de vraag of die Aquima zoals het nu opgezet is ook zo leuk en uitdagend vinden, dus daar kun je ook zoeken naar wat extra uitdaging om te bieden.

Dick:
En we hebben een naam bedacht: Aquima Evolution.

Design round 3: Ontology elements
Corresponding audio file: IC_A0004.wav

Mark:
Wij hebben een visgraatdiagrammetje getekend, opgehangen aan mens, machine en middelen, want dat zijn uiteindelijk de actoren die je nodig hebt om het spel tot een geheel te kunnen brengen. Ik heb er eentje toegevoegd, en dat is zeg maar de wereld, waarin we ons op dat ogenblik begeven, dat is wat lastiger, zeg maar, normaal bij de aarde zijn wij dat zelf, maar nu moet je dat creëren. De wereld, en de dingen als eilanden, wegen, ruimte er omheen, dus eigenlijk het totale plaatje of het bord waarin je je moet begeven heb ik maar eventjes 'de wereld' genoemd.
Dan krijgen we 'mens', of de actoren, daar hebben wij bouwers, oppers, architecten, verkopers, de cultuur, de politiek, en de burgemeesters, eigenlijk gewoon de rollen en de dingen die van invloed zijn op de wijze waarop wij ons onderling gaan gedragen.
Middelen... geld, een stukje beleving willen wij als middel aanreiken, namelijk de beleving tijdens het spel, en dingen als bouwmaterialen en andere zaken die je dan nodig hebt om überhaupt in onze de wereld iets voor elkaar te kunnen krijgen. Gereedschap, machines, natuurlijk het vehikel waarmee wij gaan... wat dan ook, vliegen, varen. Tools, hamer, spijker en weet ik wat allemaal. Dat zijn eigenlijk een beetje de rollen en actoren die eraan vast zitten. Dan geef ik het woord graag aan mijn compaan, Rick...

Rick:
Nou, dan ga ik iets vertellen over het proces. In den beginne is er niks, dan heb je een paar.. hoe heten die mensen... ontdekkers, of explorers, dat zijn de accountmanagers, die zoeken een opdracht, dan zodra er een opdracht gevonden is dan komt er een team, die gaan dan de opdracht definiëren, samen met de klant, dan wordt er onderhandeld over de resources, en als er besloten is over de resources dan wordt er gebouwd, dan is er een stukje innovatie, dat wordt ook gedeeld door het project. Dit is over Everest heen, dit blokje. Dan specifiek aan het project, daar wordt gebouwd, natuurlijk. Als er al iets gebouwd is, dan wordt het geëvalueerd met de klant, dan hebben we twee verschillende mogelijkheden, alles wordt bijgesteld, en dan wordt de opdracht weer gedefiniëerd en dan gaan we deze sequence weer in, of het is klaar en het is feest. Na het feest is er ook weer een evaluatie, met drank, dit ook weer globaal over Everest heen.

Mark:
En waarom is die 'met drank' belangrijk? Je moet nog even een toevoeging doen.

Rick:
Nou, dat is omdat je met drank de waarheid spreekt, dus dan durven mensen ook te zeggen waar het op staat.

Dick:
Aha, als de waarheid is aan de man, dan is de wijsheid in de kan. Ja, dronken mensen en kinderen spreken altijd de waarheid.

Rick:
Ja, klopt. Nou, dat is het proces. En dan, in principe, volgende keer dit rondje van de buitenkant, totdat het een keer een punt bereikt dat het klaar is. Nou, dan heb je hier nog een stippellijntje. Als het feest is, dan kan de klant in principe twee dingen doen. Of hij kan zijn nederzetting uitbuiten tot een metropool, dus dat betekent dat de opdracht weer gedefiniëerd gaat worden, of hij kan teruggaan naar nog een stapje hoger, en dat betekend dat er eventueel een andere nederzetting op verzekeringen wordt gezet, of hypotheken, en dan gaat dit proces weer lopen.

Ingrid:
Misschien kunnen we hier beter een event van maken in plaats van een feest, want het kan natuurlijk ook tegenvallen, dan wordt het meer een begrafenis.

Ilona:
En hoe ga je die virtuele klant dan dronken voeren?

Ingrid:
Ja, bij ons is niet alles virtueel, want wij hebben natuurlijk die tafel, dus we kunnen ook de communicatie tussen projectleden onderling ook gewoon direct doen.

Dick:
Ik zou Microsoft eens bellen, want volgens mij heb jij eindelijk een keer een doel voor dat ding bedacht.

Ingrid:
Ja.. volgens mij hebben ze zelf zoiets ook wel voor ogen.

Vincent:
Volgens mij ben ik nu aan de beurt... Even kijken, hoe gaan we dit beschrijven. Het thema van ons spel is zeg maar de levensloop, of de mens in relatie tot de ander, een mens wordt gerepresenteerd in die creatures, nou levensloop dus. De focus daarbij ligt op het expliciet maken van die relaties, verwachtingen van actoren, dus het beter maken, het verder helpen, het afmaken, met specifiek ook de focus op fasering, groei, meer mogelijkheden krijgen. In eerste instantie kun je nog niet zoveel, je kunt alleen maar requirements beschrijven en later kun je dingen gaan invullen, dan komen er meer mogelijkheden bij. Dus die groei zit daar heel duidelijk in.
Actoren vonden we een bijzonder moeilijke, en je komt toch een beetje terug op de mensen die deelnemen aan je project, en dat verander je ook niet, het is ook goed om daar dichtbij te blijven. We hebben dat toch een beetje expliciet gemaakt. Rondom requirements heb je mensen die het gaan beschrijven.

Het initiëren, dan zit er een review slag achter, dus mensen gaan reviewen, en uiteindelijk moeten de requirements vervuld worden. Maar dat gebeurd voor een groot deel ook in latere levensfases of in de vorm van andere creatures, dat zijn nog even twee varianten. Het is duidelijk dat het evolueert, maar het is nog de vraag of je oude creatures blijven bestaan. Als je dat doet kun je ook relaties tussen creatures leggen, zoals 'deze realiseert die'.
Nou, een ontwerp creature zou in eerste instantie als actor kunnen hebben dat iemand het ontwerp bedenkt, het vertaalt vanuit de requirements. Het ontwerp moet gemaakt worden, beschreven, getekend, noem maar op wat je met zo'n ontwerp kunt doen, en ook weer gereviewed worden en uiteindelijk soms geaccordeerd, bij sommige ontwerpen hoeft dat niet.
Met de realisatie krijg je ook zo'n soort stappenplannetje, de realisatie moet bedacht worden, gebouwd worden, ook iets technisch. Een service kent een realisatiefase, ondanks dat je dat niet in Aquima doet voor een groot deel, maar je vind het wel terug, dus een bijzaak eigenlijk. Die moeten er namelijk ook ingehangen worden, dat is dus ook een actor die dat doet. Ook dit wordt weer getest, in eerste instantie een unit test, in de volgende fase gaan we ook werkelijk systeem testen, misschien dat ook acceptatie testen daarin een rol kunnen spelen. Ook de opzet van testen kan in een actor terugkomen. Dat is een beetje het idee dat we daarover hadden.

Dick:
Verschillende fysieke mensen kunnen verschillende rollen spelen. Het kan ook dat een review gedaan wordt door een externe partij.

Vincent:
Dat is wat we zo'n beetje hadden, dat laatste zijn we niet helemaal op teruggekomen, maar dat is impliciet al wel zo'n beetje besproken

C.2 Workshop session 2

The second workshop session was held on the 15th of December, 2008, in Den Bosch. The same set-up and worksheets as in the first session were used.

C.2.1 Transcription

Workshop Aquima Game Design, December 15th, 2008
Den Bosch, Everest, zaal Norgay.
13:00 - 16:00
Participants: Rob Gense (Business Engineer), Stef Demeester (Business Engineer), Vincent Jansen (Technical Engineer), Friso van der Meer (Technical Engineer)
Facilitator: Ilona Wilmont

Assignment 1: Prior to the design rounds, think of a storyline which the player has to experience while playing the game. Keep this in mind during the workshop rounds to come.
Corresponding audio file: IC_A0005.wav

The participants were divided into two groups of two (Rob and Vincent, Stef and Friso). Each group came up with their own storyline and presented this to each other.
The following is an account of the presentations (in Dutch):

Friso
Ja, ons concept was eigenlijk dat je à la Settlers een wereldkaart hebt, en daar heb je eigenlijk allemaal factie's op die dan beginnen, dus je kan denken aan functionele mensen, mensen die met de klant interacteren...

Stef
Eigenlijk alle rollen die er in een project spelen, zeg maar...

Ilona
Die staan op die kaart?

Stef
Ja, die staan op die kaart, de Sales, de uitvoerders ...

Friso
Je ziet eigenlijk een factie met een stukje land...

Ilona
Een factie?

Stef
Een groep, ja een groep met een stukje land op de kaart.

Friso
Nou, wat zegt de klant, wij willen iets hebben en dat moet er ongeveer zo uitzien.

Dus die zetten een kasteel neer, dat moet gebouwd worden, en daar hebben ze allemaal requirements aan, en dat slaat aan, en dat wordt dan behandeld door een aantal mensen die dus op die kaart staan, dus voor het functionele zijn er een aantal mensen bezig, die specificeren dus: ”nou, daar hebben we dan dit voor nodig”, want die bouwen dan ook een huisje, en dat moet dan aansluiten op wat er technisch gerealiseerd kan worden, dus technisch moeten ze dan ook weer een huisje gaan neerzetten. Dan heb je dus allemaal huisjes van dingen die gebouwd zijn, die heb je echt fysiek gerepresenteerd op die kaart. Nou, zo heb je dus allerlei onderdelen, en dan beginnen er paden tussen die huisjes te lopen over de landkaart, dus ook naar de andere factie toe, en die aansluitingen representeren dus eigenlijk gewoon de communicatielijnen dan. En als mensen dan veel bugs hebben op een bepaald punt, worden die daarop ingeschoten, dan komt er eigenlijk gewoon afval of rotzooi bij dat huisje te liggen en dan zie je dus op het laatst van ”hé, dat zit hier”, en dat kun je dan fysiek zien, waar het probleem zit. En als mensen het dan niet mooi vinden, elke factie die heeft zijn eigen kanon, en die schiet dan op een huisje, en als er maar genoeg mensen dat doen vanaf meerdere plekken, dan krijg je een zwartgeblakerd huisje en dan zie je dus ”hé, er zit iets hier wat niet mooi is”.

Stef
Je kunt het ook anders representeren, hè, maar...

Friso
Ja... Het uiteindelijke doel is dan dat je een mooi gevulde landkaart hebt met elke factie, die kan natuurlijk ook een beetje groeien of krimpen, afhankelijk van hoeveel ruimte er nodig is, waarin er dus veel paden lopen en er niet dingen dus heel afzonderlijk los staan, wat eigenlijk een niet gebruikt iets representeerd, en dat er niet allerlei zwarte plekken zitten van die kanonskogels. Dat is eigenlijk het idee, die kun je ook opruimen door dingen dus aan te passen. Dat is een beetje het principe waar wij aan zaten te denken.

Rob
En die iglo dan?

Stef
Die iglo, in de winter zijn de huisjes door iglo’s voorgesteld, en in de zomer door een strand in de Bahama’s... Je zou kunnen zeggen, dat wil de klant hebben, dat is dan zijn huis... Je zou ook kunnen zeggen dat er opeens een huis staat, bij de techneuten bijvoorbeeld, waar geen mannetje tussen loopt, dat betekent dat zij iets gebouwd hebben dat helemaal niet overeenkomt met wat de klant gevraagd heeft. Dus uiteindelijk heb je een reviewsessie of weet ik veel wat, en dan moet het huisje door iedereen goedgekeurd worden. De klant bijvoorbeeld, die moet het echt gevolgd hebben, stel dat het zijn huisje is, een beschrijving van wat er moet gebeuren. Als het niet goedgekeurd wordt, dan lopen er geen mannetjes, dan, net als in Sim City, verpaupert je huis. Of de techneuten hebben iets gebouwd dat helemaal niet gevraagd is, of omgekeerd.

Friso
Ja, en als het dan niet gewild is, dan kun je ook zeggen ”ik vind dat mannetje niet zo leuk, ik geef daar niet zoveel support aan”. Ook het weggetje daar naar-

toe, dat wordt het in plaats van een mooi stenen paadje een mooi modderpaadje, uiteindelijk, en daar kun je dan een aantal gradaties in aanleggen, dan zie je dus hoe een aantal dingen aan elkaar zitten. Het resultaat is dus dat je een mooie kaart krijgt waarop aan het eind van het project ook heel erg goed te zien is hoe een hele projectstructuur in elkaar zat, wie wat heeft beoordeeld, wat heeft bekeken, en dan kun je toch wel... het is toch een representatie van hoe werkte dit project nou eigenlijk, dat is dan het idee.

Ilona
Dus zeg maar, alle projectcomponenten dat zijn dan die huisjes...

Stef
Ja, ja.... Wij hebben naar het proces gekeken, voornamelijk. En de rollen binnen het project. Ik had eerst het idee, met die iglo, zeg maar, dat was eigenlijk meer de applicatie, maar Friso kwam op dat moment met een heel ander idee...

Friso
Ja, ik dacht, je kan wel het wel toe laten passen op de bouw zelf, maar dat is zo direct... Je wil het ook eens een keer van een andere kant bekijken, want je bent elke keer toch bezig met "hoe knoop ik alle onderdelen aan elkaar?" Maar je kan ook eens een keer kijken naar het proces, van "hé, wie heeft er nou eigenlijk allerlei links naar elkaar toe gelegd?" Dan begrijp je de setting van je project misschien wat beter. Een manier om te visualiseren zonder dat je met je dagelijkse ding bezig bent, maar je kan het er wel aan hangen, dus wat jij gemaakt hebt wordt wel fysiek beloond door aanwezig te zijn op die kaart.

Rob
Misschien valt er nog wat te combineren, zeg maar, want wij hebben inderdaad wel veel meer gekeken naar de bouw van de applicatie. Het project is gewoon een te ontdekken landkaart. Je zult ooit globale requirements nodig hebben, dus er is een land, met ergens een heuvelrug die je BKR toets is, je hebt wat meer de beoordeling, of weet ik veel wat allemaal, en je bent als gebruiker eigenlijk een ontdekkingsreiziger, zeg maar, die dat land gaat ontdekken. Het onontdekte land is natuurlijk te visualiseren met een kaart met wel ontdekte onderdelen en welke nog niet. Een ontdekt onderdeel, dat is een onderdeel dat af is, dat is getest, doet het, is goed, enzovoorts. En op een gegeven moment moet die landkaart helemaal gevuld zijn, anders is je project niet af. Je gaat dus het land ontdekken, de Balkenende-VOC mentaliteit ... En je hebt tools nodig om dat te ontdekken, en tools, dat kunnen services of beslisregels dingen zijn, zeg maar, om een bepaald stukje te kunnen ontdekken. Het kan zijn dat je inboorlingen tegenkomt, dat zijn je schermen, eigenlijk, dat is communicatie, waar je mee praat, zeg maar, met elkaar kunnen kletsen. En je zou nog iets met spiegeltjes en kraaltjes, handel kunnen doen, over je requirements onderhandelen... Want je inboorlingen dat zijn feitelijk je klanten, dat is wat ze willen... Je inboorlingen, dat is wat je gaat ontdekken, wat willen zij en dan ga je onderhandelen en ruilen over wat je wil hebben zeg maar, en dan zou je nog kunnen kijken van hoe je... Ja, wat is testen dan? Daar zaten wij nog naar te kijken... Als je inboorlingen kwaad worden, je zou nog iets moeten doen om je test er nog in te bouwen. Je zou oorlog kunnen krijgen met je inboorlingen, dat zou kunnen gebeuren omdat je niet voldoet aan de test case... Je hebt onderhandeld over

iets, je voldoet niet aan de test case dus er is oorlog, dan moet je gaan meppen, met tools, om het weer op orde te krijgen. En dan zou je nog met kolonisten kunnen gaan werken, dat zou je meta-model kunnen zijn, daar waren we nog niet helemaal... Maar zo zou je zeg maar je wereld kunnen ontdekken, bouwen, verder uitbreiden, maar uiteindelijk zal je het hele land moeten bestrijken, anders heb je niet alles ontdekt.

Friso
Maar wie zorgt ervoor dat zeg maar de representatie van de oorlog, dat er flink wat fout gaat, ook gevisualiseerd wordt op je interface?

Rob
Dat zou je moeten doen met een soort van test cases die je ingebouwd hebt op het moment dat je je onderhandeling ingaat, een soort van unit test-achtige dingen waarmee je kunt kijken of iets voldoet. Ik zit nu te denken, een regressietest zou zijn als een vreemde horde die in een keer het hele land binnentrekt en dan kijkt of alles het nog goed doet, zeg maar. Of er komt iemand nieuw aan, een nieuwe kolonist, die gewoon alles gaat bekijken. Barbaren... of barbaar...

Stef
Je zou ook nog een soort Romeinse setting kunnen gebruiken. Vanuit Rome haal je het Romeinse emperium... dat is dan een project... Van je zeven heuvelen ga je dan af, naar de gesloten buitenwereld....

Rob
Ja, dat soort aspecten eigenlijk. Je project is een te ontdekken land...

Stef
Je zou het goed kunnen combineren, zeker het functionele en het technische stuk zou je helemaal zo kunnen doen, dan heb je, in je land of je representatie, dan heb je daar nog alles wat hij moet doen, al die huisjes en heuvels en whatever...

Rob
Ja, en het voordeel hiervan is dat je een stukje alleen kunt ontdekken, maar je kunt het ook met meerderen doen, zeg maar, dat je met een paar mensen een stuk land gaat ontdekken, en met die paar mensen ga je ook al je resources toewijzen.

Friso
Zo hadden wij ook al, van wat je zei met die heuvelrug, dat je bepaalde dingen tussen factie's kan leggen. Het representeerd ook een barrière, van ”hier zit een barrière...

Stef
Inderdaad, als je een presterend systeem heb, er kan een meer tussen liggen, of de specs zijn nog niet opgeleverd door de analisten...

Rob
Je kunt ook het einde van de wereld in pleuren, dat het zeg maar een platte aarde is...

En we hadden nog een idee bedacht, van een spookhuis. En de test cases zijn dan je spoken, en die moet je bestrijden.

Friso
De test cases zijn je spoken, of de resultaten van je test cases?

Rob
Nou, je test case is een spook, en je test case, je spook, moet je dus bezweren...

Friso
Op de goede plek houden, zeg maar...

Rob
Door te voldoen aan je test case hou je het spook buiten...

Friso
Ik dacht aan kamers, dat je dan entiteiten en attributen kan maken, en dat dat dan sleutels zijn om de kamer binnen te kunnen kijken of daar de uitgang is...

Stef
Zo zou je inderdaad je testen heel goed kunnen visualiseren, door inderdaad zo'n spookslot, en de spoken zijn dan je bugs, of je potentiële bugs, en die moet je in een kot houden...

Friso
Ja, het zijn testen, wij dachten ook, misschien kun je monsters hebben, en die dan vasthouden...

Gezamelijk
Lijken die uit de kast vallen... Dungeons onder de grond... Armageddon, het laatste oordeel....

Design round 1: Ludus elements
Corresponding audio file: IC_A0006.wav

Vincent:
Wij hadden als Title *Conquer...* uitroepteken!

Rob
Met als strijdkreet "Aquima!"

Vincent
Het type was een beetje een digital puzzle simulation, om even alle woorden te gebruiken die in die zin op het werkblad stonden. Een beetje Kolonisten van Catan meets Colonisation-achtig spel dat tevens als multiplayer of als single player te spelen is. Dus op zich kun je zelf weten of je zelf de hele wereld gaat ontdekken of dat je het met meerdere mensen wil doen. De world, wij hadden eigenlijk zoiets van, je kunt het zo maken dat je zelf kunt selecteren waar je wil zijn, 'waar' ligt er vooral aan wat de klant wil...

Rob
Je kunt ook nog een tijdvak kiezen, je kunt er de Spaanse Conquestadors van maken, of de VOC die ergens naartoe gaat, of Star Trek die outer space gaat ontdekken, natuurlijk...

Vincent
Op zich kan het natuurlijk van alles zijn, dus je zou zeggen, omdat verschillende mensen het moeten spelen, kun je het zo maken dat je het configureerbaar maakt, want de één vind het bijvoorbeeld leuk om te Star Trek-ken, en de ander vind het leuk om de achttiende eeuw....

Rob
... jungle te ontdekken in Afrika of Amerika, of de Noordpool...

Vincent
Dus op zich leek ons dat wel leuk. Als perspective leek het ons wel leuk als je een beetje 3D kan rondkijken. Interaction model was gewoon met de muis en toetsenbord, voor als je wil chatten en notes... als je met elkaar wil praten, communiceren.

Rob
En kaarten aan elkaar kunnen laten zien, schatkaarten. Een bug is een schatkaart... Daar zit iets fout!

Vincent
Een uitdaging is op zich wel leuk, mensen houden van uitdagingen. De verschillende onderdelen, daar hebben we het in de vorige fase ook al over gehad, de kraaltjes, die dan de requirements zijn, en de inboorlingen, de klant, de tools, de services, decision tables en zo.
Als verhaallijn hadden we dan dat je het ding moet ontdekken, dus de hele kaart ontdekken. Je moet alles niet alleen zien maar ook echt bezetten met kolonisten. En je moet zorgen dat je test cases allemaal goed zijn, anders heb je ruzie met de inboorlingen en dan zit het fout.

Rob
Dus gedurende de loop der tijd moet je alle inboorlingen wel tevreden houden. Je zou het ook Peace and Harmony kunnen noemen...

Vincent
Je kunt ze ook met andere dingen tevreden houden, minder vriendelijke dingen. Je goals en je challenges zijn dus om op zoek te gaan naar onontdekte territoria en te communiceren met de inboorlingen dus om te zorgen dat er geen oorlog is en te onderhandelen over wat voor kraaltjes ze willen en hoe die er allemaal uitzien, dus je requirements vaststellen. Op zich zijn dat ook ongeveer je actions, het zoeken, het communiceren met de inboorlingen. En als reward hadden wij gedacht aan XP's die je krijgt, en hoe meer XP hoe meer je van de globale kaart ziet. Dus je moet nog wel op detail ontdekken, maar dat je al wel een beetje globaal ziet...

Stef
Ja ja, dan zijn de voorwaarden wat minder.

Rob
En ook als je dat meeneemt naar andere projecten kun je je XP's meenemen, en hoe meer XP's je dan hebt, hoe meer ervaren je bent en hoe meer je dan ook van de globale kaart ziet, dat je dan ook mee kunt denken, dat heeft natuurlijk alleen maar waarde als je ook de grote lijnen kunt zien en daar iets mee te doen.

Vincent
En dat was het in het kort wel zo'n beetje.

Rob
Graphics hebben we nog niet.

Friso
Wij? Nou, er kwamen twee titels voorbij. Eentje was *As The World Turns*, een beetje cheesy, maar ook wel weer leuk, en de andere was, maar dat was meer de working title waarmee we begonnen, *Settlers meets Collective Mindmapping.* Dat mindmapping, dat knoopt natuurlijk allemaal dingen aan elkaar, en je ziet zeg maar wat er dan is, en hoe dat in elkaar steekt, maar dat is voor jouw mind. Maar als jij als team met allerlei andere belangen bezig bent, dan zie je ook allemaal lange lijnen lopen, dus dan krijg je ook een beetje collective mindmapping. Wat is nu eigenlijk allemaal gebeurd en hoe zijn de verhoudingen tot elkaar, hoe zitten die factie's en hoe werkt dat allemaal? Dus vandaar dat Collective Mindmapping even voorbij kwam zetten.
Type of game was een beetje Settlers, Sim City-achtig, een soort simulation met een strategie-component. Je hebt dus een wereldkaart, en die ziet er zo uit, het is dus niet dat je een bepaald doel bereikt, het is meer de aesthetica van 'het moet er mooi uit zien'. De bedoeling is dat je de kaart vult met van alles, en dat begint dan je project te representeren. Dat is een beetje de typering van het spel, er zit wel wat strategie is, je kunt onder elkaar dingen verhandelen. En als ik toestemming heb, dan mag ik een kasteel bouwen, en als 3 factie's dit van mij willen, dan ziet het eruit als een kasteel, en als niemand heeft gezegd dat ze dat van je willen, dan staat er een hutje. Meer kun je er dan ook niet van maken. Maar je kan ook je mening in je object verwerken, dan maak je er zelf een beetje van, zo'n toilethuisje buiten in het veld als noodzakelijk kwaad, zo zit het allemaal een beetje in elkaar.
Dan komen we op het volgende punt van de game world uit. Op elke kaart, van elke factie, staat er van 'wij hebben iets nodig en dat moet dit doen', nou, dan zetten we er een object neer, en als dat object gevraagd is door iemand, dan kunnen ze dus kiezen, van 'oh, we zetten er een mooi badhuis neer, of een kasteeltje' en dat zetten zij neer. Maar zij hebben dus weer dingen nodig van anderen om het echt af te kunnen maken dus dan ligt er een link naar een andere economie, van een andere factie, van 'hé, wij willen dat van jou', de andere factie zegt 'prima, maar dat zijn dan 6 huisjes, 6 onderdelen, die moeten ook gebouwd worden, en dan komt er eigenlijk automatisch een paadje tussen die twee dingen te zitten. Dat paadje representeerd dus communicatie, daarom lopen er ook poppetjes overheen die een soort van status representeren. Dus van nou, er wordt gevraagd dat er gebouwd wordt, dus dan moeten er poppetjes heen

gestuurd worden, of dat gaat eigenlijk soort van automatisch, dan wordt eraan gewerkt. Dus die zitten dan te timmeren aan die huisjes, op het moment dat diegene in die factie zegt 'nou is het klaar' dan gaan de poppetjes weer terug en dan kunnen de anderen eens komen kijken, van 'hé, is dit eigenlijk wel mooi?' Is het mooi, zijn we het ermee eens, dan kunnen ze dat aanklikken, dat is dan wel af, dan zie je ook dat het huis compleet is, dat het daardoor ook bloemetjes buiten gezet worden, dat idee, anders is het huis gewoon kaal, zit er niks in. En die link daartussen, dat representeerd wie, welke partijen eraan mee hebben gedaan om het tot stand te brengen, dat is eigenlijk het idee erachter.

Rob
Maar ook dat je met meerdere mensen elkaar's code even reviewt, dat is goed om daarover...

Friso
Ja, het is echt je mening, en het is mening buiten je factie, dus je bent met een setje mensen bezig en een onderdeel van een team, dus je kan zeggen 'ik ben met twee man bezig om een interface hoek te maken', dat is dan een interface team, dat is dan de factie die je representeerd op de kaart. Nou, het leuke is, van elk punt dat aangedragen is door de klant of door een functionele rol gezien, daarvan kan je zeggen, 'nou, dit hebben we, of dit hebben we nodig', en dat linkt dan aan wat een andere factie verzorgt of ook heeft gevraagd. Dus de klant vraagt het, functioneel team denkt erover na, zo willen we het maken, en dan is er technisch weer een realisatie nodig. En elk deel heeft ook een mening over wat er gemaakt wordt. En dat is ook wel weer leuk. De klant zegt 'ik wil dit hebben' en ja, wij hebben toch een klant nodig als noodzakelijk kwaad, dus die zetten een puntje. Functioneel team zegt 'als we dat een klein beetje zo doen, dan vinden we dat toch eigenlijk best wel mooi', dus dan wordt het een badhuis, kom je bij technisch, dan is het 'leuk principe, hebben wij wat van geleerd', ook een mooi huis! Dus dat zijn allemaal dezelfde dingen waarover je spreekt, maar het ziet er anders uit in een andere economie. En dat maakt zeg maar een soort van diversiteit op je kaart. Dat is het idee erachter.
Nou, dan hadden we dus net ook al dat je met een kanonnetje kunt schieten als mening, naar een andere factie toe. Want je eigen mening over een object, een deliverable die je hebt gemaakt, kun je al stellen door een type huis zeg maar te verhogen of te verlagen. Dus dat is het interactie-idee. Nou, er is een beetje een risico-handeling erin, want er is maar een beperkte tijd, beperkte hoeveelheid, dus die handel je eigenlijk, soort van... Hoe we die precies erin moesten passen, daar hadden we nog niet over nagedacht, maar daar zou je iets mee kunnen doen...

Rob
Maar zie jij nou techneuten als een factie, analisten als een factie, klant als een factie...

Friso
Nee, dat is een groep die het team... maakt niet uit, een groep die iets doet.

Rob
Dus die kan bestaan uit technici...

Friso
Je kan heus zeggen in een project, dan heb je twee factie's, dat is lekker saai, hup, technisch, en hup, analisten..

Rob
Nee, maar het lijkt me, je wil ze ook gebruiken om dan samen te werken.

Friso
Ja, je kunt ook mixen, want je hebt een factie op interfaces, dat bedoelde ik als voorbeeld, die met interfaces bezig is, dus dan heb je een analist, en een techneut, iemand die met de klant praat, of iemand van de klant zelf... en dan heb je dus drie personen die daarmee bezig zijn, en die hebben ook allemaal een mening over hoe dat in elkaar steekt, en die zijn ook met z'n drieën echt bezig. Maar je kan ook hebben dat inderdaad een project vrij klein is en het bestaat uit 2 mensen, ja, hoe deel je dat dan in? Je kan zelf kiezen wie tot welke factie je behoort en hoe dat eruit moet zien. Elke factie heeft dan een kleur op de kaart en de grens die zie je dan door de kleur.

Stef
En wat ze zelf verder aan kunnen geven is bijvoorbeeld 'wat is de status van dat ding?', 'heb ik iets nodig van iemand anders?' en dat wordt dan gerepresenteerd als poppetjes die staan te wachten, ik heb iets nodig van iemand anders, of m'n huis staat nog in de steigers, of de fundamenten zijn er net, of zijn bijna klaar...

Rob
Nee, maar ik ga het spel spelen, dus ik heb een login, en dan kom ik in een soort wereld waar ik allemaal huisjes zie, en mannetjes...

Friso
Je werk begint, en je ziet niks. Je hebt alleen een kale kaart. Zo, nou er zitten wat heuvels in...

Rob
En in de leegte komen allemaal poppetjes bij elkaar...

Stef
En ze weten nog helemaal niks. En al je objecten zijn deliverables. Als er een deliverable opgeleverd wordt, dan worden die gemapt, zeg maar, op de kaart, en die deliverable zal hoogstwaarschijnlijk leiden tot iets anders. Bijvoorbeeld, requirements, dat moet ook nog mee in een ontwerp, heb je gelijk al twee deliverables. Misschien, hoogstwaarschijnlijk, gaan die requirements in 3, 4, 5, of 6 ontwerpen resulteren. Dan zie je al in die verschillende factie's verschillende huisjes ontstaan. En de spelers die geven dan onder andere status aan, of ik wacht nog op een test, of het moet gereviewed worden, en dat zie je ook welke partijen dat moeten gaan doen, dat zijn namelijk de lijntjes ertussen, de afhankelijkheden, en aan de status van de poppetjes die over en weer lopen, of niet over en weer lopen, die staan te wachten, zie je ook, zeg maar, wat er gebeurt. En de wereld moet doorgaan, As The World Turns, het moet draaien, en niemand mag staan wachten. Of het moet klaar zijn.

Friso
Ja, en als je dat inderdaad ziet, want dat is leuk als visualisatie, iedereen staat op elkaar te wachten, dan staan er allemaal poppetjes zo op hun hoofd te krabben, van 'wij kunnen niets, wij zitten te wachten ergens op'. En als overview, of projectleider, die kan dan gaan kijken, van 'hé, hoe loopt het hier?' Ah, er wordt lekker getimmerd, en andere dingen lopen lekker heen en weer, dan betekent van, het is goed. Maar als alles stilstaat, of als er veel punten zijn waar de aandacht gevraagd wordt, misschien moet er dan geschoven worden met de factie's, misschien moet er dan wat minder communicatie gaan plaatsvinden, kom je eigenlijk op het volgende punt uit, dan kan een projectleider gaan zeggen van 'klik, hier moet maar eens even een berg gaan komen, of een heuvelrug, of een muur als verdediging'. Dat zou een projectleider allemaal kunnen doen, dus die heeft een andere set van tools, hier even wat minder en focus daar eens wat op, en zo krijg je dan een beetje het effect van een uitziende wereld die ook de elementen bevat zoals wij de wereld kennen: meertjes, bergen, dat soort dingen, daar zou je dan ook iets mee kunnen doen.

Design round 2: Semiosis elements
Corresponding audio file: IC_A0007.wav

Stef
Als doel zagen wij het inzicht verschaffen in het proces, van hoe verloopt het? Inzichten in de verschillende belangen die de factie's hebben, en ook het inzicht verschaffen in de mening die de verschillende groepen hebben over waarmee ze bezig zijn, en vooral dan.. het zou het interessantst zijn als die dan afwijken. Los daarvan, je moet het wel professioneel houden, zodat je ook kan zeggen dat je vind dat je met een flutding bezig bent. De vraag is dan of de klant dat ook moet zien, maar je kan wel je mening erin kwijt...

Friso
Dat is nog een stukje emotie, zeg maar, alleen je kunt het op een leuke manier doen.

Stef
En redelijk anoniem, waardoor je misschien eerder geneigd bent te zeggen 'dit is klotewerk' of zoiets.

Friso
Het hoeft nog niet direct iets te zeggen over de persoon, of de factie, maar dat maakt het spel niet duidelijk, het gaat er maar om om vanaf een hoger niveau inzicht te krijgen. Het doel van het spel zelf, ik weet niet precies wat nou de vraag was, wat je met het spel wil bereiken of wat het spel zelf als doel heeft, maar het spel zelf heeft als doel dat alles vloeiend blijft verlopen, dus dat je poppetjes blijven lopen, dat er niets staat te wachten, geen zwartgeblakerde velden door die kanonskogels, dat is een beetje het idee.

Rob
Ben je altijd maar één poppetje?

Friso
Nee, je bent niet echt een poppetje, je plaats huisjes en daartussen lopen poppetjes, een poppetje representeerd eigenlijk een proces wat gaande is, 'ik moet nu daarheen, ik moet die informatie hebben en dan kan ik daar timmeren, of bouwen'.

Stef
En als speler plaats je je deliverables in een faction, bijvoorbeeld, ik maak nu een F.O., of een stuk code, of requirements, ik stel een plan voor de requirements op.

Rob
Maar een poppetje is in principe wel iemand die iets moet uitvoeren.

Stef
Ja... je deliverable heeft een status, en die druk je dan uit of er is tussen 2 groepen iets wat moet gebeuren, en dat druk je dan uit door bijvoorbeeld poppetjes heen en weer te laten lopen of te laten wachten... 'hij staat te wachten op jouw deliverable, hij kan niet meer verder' druk je dan uit. Je druk de status uit van je deliverable.

Rob
Maar het kan best zijn... aan de ene kant is het wel mooi om te zien waar je bottlenecks zitten, aan de andere kant... iemand kan maar een bepaalde hoeveelheid doen op een gegeven moment, dus het kan best zijn dat hij op drie punten staat te wachten terwijl die op vijf andere punten aan het rondrennen is...

Friso
Ja, dat ook, maar je kan best hebben dat er heel veel staan te wachten, dat zal je in het begin van je project natuurlijk heel veel hebben, maar het idee is om er daar wel een aantal van weg te kunnen krijgen, die niet meer staan te wachten en dat het af is, en je ziet ook, hoe meer er staan te wachten ten opzichte van het aantal deliverables, weet je dat het proces enigszins vastgelopen is.

Rob
Wat dan mooi zou zijn is als je inzicht kunt krijgen in... het kan bijvoorbeeld zijn dat er ergens een poppetje staat te timmeren, maar dat hij nog best veel tijd over heeft om een poppetje wat staat te wachten af te lossen.

Friso
We hadden het op meerdere manieren kunnen doen, van een poppetje wat staat te wachten, na de eerste tijd, als dat gepland is, zeg maar, ja, dan verveeld die zich nog niet, dan staat die daar wat.

Stef
Maar dan is een poppetje wel een speler... Want nu zijn we eigenlijk uitgegaan van de deliverable, en niet van het poppetje. Degene die de dingen uitvoert, dat zou je ook kunnen doen, maar dat is een andere insteek. Dan ga je kijken naar je... je zou het zelfs tegelijkertijd kunnen doen, dat je dus echt poppetjes aan deliverables hebt, maar dan... het kan dus best zijn dat iemand dan meerdere

rollen heeft binnen een project, en dan gaat het wel allemaal heel erg door elkaar lopen. Je zou bijvoorbeeld iemand kunnen hebben die gedeeltelijk analyse doet, al is het maar technische analyse over hoe iets technisch gerealiseerd moet worden, waar zit die dan?

Rob
Ja, okay, ik kan me voorstellen als technisch iemand ben je misschien met de ene service bezig, die container bezig of dat taakje bezig, maar daar ben je heel druk mee bezig dus kom je ergens anders niet aan toe, dus op die plek staat er dan nog een poppetje te wachten.

Friso
Ja, dat hoeft nog niet direct zijn te wachten, alle vragen zijn dan wel beantwoord, dat is dan het verschil. Wachten bedoel ik meer van, alles kan nog niet gestart worden, want alles is nog niet helder, of de dependency is nog niet af, compleet gemaakt voordat het echt kan beginnen. Je krijgt dus ook een beetje je dependencies op een rij.

Stef
Ja, we werken op deliverables, en niet op resources..

Friso
Je ziet alleen, er zijn zoveel man om te vragen, elke discussie die begint, en daar komen twee vragen uit, en die vragen moeten beantwoord worden en dan kan dat afgemaakt worden. Daar zie je dat meer aan, de dependencies en de deliverables, en niet zozeer of een project persoon druk bezet is.

Stef
Ja... dat zien wij ook niet. Waar zitten we nu... Target group and setting... Ja, het bedoelde publiek is eigenlijk alle rollen binnen het project waarbij we wel hadden dat het waarschijnlijk pas interessant wordt bij de grotere projecten, als er maar drie mensen in zitten zal het waarschijnlijk ook niet nodig zijn, zeg maar, minder nodig zijn om dit spel te spelen om inzicht te krijgen in hoe het loopt, of het allemaal loopt.
Why... nou, het levert toch op een leuke manier inzicht, en je kunt je uiten, in een virtuele wereld, zeg maar, waardoor je misschien wel minder terughoudend bent, of je minder politiek correct opstelt dan dat je anders misschien zou doen.

Friso
Je uit het niet zozeer met tekst, maar meer met acties. Je zou ook best iets kunnen doen met legertjes, ja, als jij eens een keertje.. als je een keer wat meer moeite hebt met een bepaalde groep, dan kun je met z'n allen achter de PC even die legertjes erin gooien. Dat bevordert op zich ook alweer communicatie omdat je wel met z'n allen hebt zitten lachen over die acties. Dat soort effecten zou je ook kunnen bereiken. Daar zou je denk ik wat meer op moeten uitbreiden, het was maar een los idee.

Stef
Nou, de context zijn dus de deliverables, en hoe komen die tot stand en waarom? Of waarom niet? En de communicatie daar omheen, en de belangen van alle

factions die meespelen.
Reflection, dat is eigenlijk over dat uiten van 'wat vind hij er nou zelf van', plus dat de eenvoud van het reviewen van elkaars dingen reflection is, dus eigenlijk is dat de core van het ding, de lijntjes van de verschillende groepen die constant reflecteren op wat de ander opgeleverd heeft. Feedback zit in hetzelfde straatje, daar kan ik dezelfde uitleg op geven. Eigenlijk het mooiste is, op een bepaald stuk, is er feedback? Staan er mensen op jou te wachten? Of niet? Loopt het of loopt het niet? En Diversity, ja... die zagen wij even niet. Daar hebben wij not applicable achter gezet.

Rob
Ons doel... nouja, ons doel is meer, zeg maar, inzicht in het project, het complete van je project, je hebt een overzicht, je kunt zien hoeveel land er al ontdekt is, en ook wat er nog niet ontdekt is. Ons doel is ook, zeg maar, zolang er vrede is voldoe je aan je test cases.
Dus is je kwaliteit gewaarborgd, je werkt samen, je kunt het alleen doen, je kunt het samen doen doordat je samen een bepaald stukje land gaat ontdekken. Het is een leuke manier om je project af te handelen, en je klant zou je mee kunnen laten doen door hem inboorling te laten spelen, zeg maar.... Het zou kunnen.
De onderliggende message, de boodschap, dat is eigenlijk klanttevredenheid en kwaliteit. Zolang je kwaliteit levert is de klant tevreden, is er vrede, voldoe je aan je klanten specs.
Even zien... integratie... doelen in je game... nouja, op een gegeven moment als er oorlog komt met je inboorlingen, als ze je kolonisten gaan aanvallen of iets....

Friso
Speelt de klant echt effectief mee?

Rob
Nou... dat zou ook fictief kunnen zijn, want je inboorlingen zijn feitelijk je test cases. Als je test cases niet tevreden zijn, ja, dan onstaat er oorlog in dat gebied. Op die manier kun je je kwaliteit monitoren.

Friso
Het zou natuurlijk ook kunnen dat ze iets waar ze eerst wel tevreden over waren... je moet dus je kwaliteit wel blijven bewaken.

Rob
Het geldt ook niet, wat je gehad hebt dat is geweest. Nee, dat blijft natuurlijk gewoon voor het hele land gelden.
Voor wie is het bedoeld? Gewoon voor de mensen hier. Je kunt de klant mee laten doen als je dat zou willen... Tijdens een project doe je dat... Waarom? Voor de lol, het moet gewoon leuk zijn, zeg maar.
Context, dat is gewoon je ontwikkeltraject, het werk met klanten. Reflectie krijg je doordat je reflectie op je eigen handel doet. Doordat het land te koloniseren is weet je of het goed gaat, en feedback komt feitelijk door de inboorlingen, als je voldoet aan die test cases. Feedback kan ook ontstaan tijdens het onderhandelen over je spiegeltjes en kraaltjes. Diversiteit ontstaat doordat je mensen met meer of minder Experience Points hebt, dan kunnen sommigen ook verder kijken of niet. Maar doordat je ook samenwerkt onder elkaar... Je kunt alleen

een gebied ingaan en het proberen op te lossen en als dat niet lukt haal je er andere mensen bij. Wat voor ons ook geldt is dat je natuurlijk op meerdere plekken gelijktijdig bezig kunt zijn. Je zou jezelf moeten kunnen clonen dan.... Dat waren zo'n beetje de puntjes... Maar het maps nogal aardig, zeg maar... Dat je binnen een project zo de puntjes kunt pakken...

Design round 3: Ontology elements
Corresponding audio file: IC_A0008.wav up to and including 13:22

Friso
Thema, setting, dat is fantasy, en dan met name het bouwen. De focus ligt op communicatie over deliverables. Een deliverable wordt gevisualiseerd door een object... een huisje... en kasteeltje. De communicatielijnen zijn eigenlijk de paadjes tussen twee hutjes. De andere focus is het monitoren van het proces, dus dat je ziet 'dit hangt allemaal aan elkaar, het is een heel groot netwerk geworden van allemaal paadjes en huizen, en waar zitten er dan mogelijke knelpunten?' Misschien zijn er teveel lege naar een bepaalde groep toe, of er staan teveel poppetjes op te wachten, dus daar kun je iets uit lezen, en mensen kunnen hun meningen erin kwijt, dat is ook een focus daarin. Dat zijn er drie.
De problem... strandende teams, dat was een beetje het idee in het begin. Een team loopt niet echt soepel, waarin zit hem dat eigenlijk? Ja, dat kan op een heleboel vlakken liggen. Bijvoorbeeld, wat we net ook al noemden, gedrag van 'ik kan niet verder want ik moet daarop wachten, deze requirements heb ik niet, zo kan ik niet werken dus waar zijn die dingen', terwijl de andere kant kan zeggen 'ja, ik heb dat allang gegeven'. Dus waar zit dan het probleem? Dat je dat eerder helder kan maken terwijl dat pas weken later vaak duidelijk wordt. Het andere is bijvoorbeeld, hoe ga je om met professionaliteitsdingen? Soms wil je je gewoon even lekker uitleven op het feit dat je iets niet leuk vind, dus dan pak je je kanon en dan ga je daar even lekker mee aan de slag. Dat zijn een aantal probleemstellingen daarin.
De actors, dus wie spelen erin, dat is iedereen, en ik denk dat dat vrij regelmatig is zonder de projectleider omdat die eigenlijk degene is die de monitoring doet, van 'nou, loopt het lekker?' Want ja, die managed het team en moet zorgen dat iedereen lekker bezig kan, dus als er knelpunten zijn wil hij dat zo snel mogelijk weten.

Stef
En hij kan ook, als het goed is, nou de uitwerkingen van z'n maatregelen zien. Als er ergens een proces stopt en hij neemt maatregelen dan moet het wel handig gaan verlopen daarna.

Friso
Ja... verbetert het dan ook? En de relaties tussen de mensen die spelen, zeg maar, je deelt ze wel in in een groep, en in die groep blijven ze eigenlijk ook, en van daaruit werken ze en zie je de relaties ontstaan. Als dat enigszins constant is moet je een patroon kunnen herkennen van 'zo werkt het', en bepaalde dingen die niet werken die zou je dan ook kunnen herkennen. Dat is het principe. Dus het visualisatie thema is de relaties tussen...
Ja, de system of the problem... een beetje moeilijk om dat samen te vatten...

Het begint met een lege kaart en die eindigt vol.. Dat is wel een hele korte samenvatting. Je zet daar dingen op die de wereld gaan representeren, en dat wordt steeds groter en je wilt juist de knelpunten en de zwartgeblakerde velden van de kanonskogels weghalen. Dan krijg je een soort van werkende geheel waar van alles in beweegt en dat ziet er leuk uit, dat is het doel. Hoe je dat dan precies aan elkaar knoopt....
Dan is het building en visualisation van deliverables en afhankelijkheden tot elkaar, en dat kun je dan weer zien door al die communicatielijnen tussen al die onderdelen.
Het process... ja, wat je wil zien in dat systeem als dus bepaalde dingen vastlopen, bijvoorbeeld als iemand heeft gezegd 'ik heb de requirements opgestuurd' en die communicatielijn loopt dan iets naar de andere kant van 'nou, ik ben klaar', zegt hij van 'ik ben niet compleet', dan is er een verschil in de mening van twee plaatsen, dan gaat er een poppetje stilstaan die niet weet wat hij moet doen. Op die manier kun je problemen en knelpunten echt vrij letterlijk visualiseren. Dat is het idee erachter.

Stef
En het proces van 'je maakt je huisje', iedereen weet toch wel waar je mee bezig bent, meestal gaan er dan afhankelijkheden ontstaan. Die geven je status aan, en gaan om je deliverable heen hangen.. 'ik sta te wachten', of 'ik ben bezig', 'ik ben voor 50 % klaar', dat zijn dingen die gebeuren in het spel, en het spel zelf visualiseert dan hoe het in zijn geheel dan loopt, waar knelpunten en dat soort dingen zitten, wordt er afgebouwd, niet afgebouwd, dat soort dingen.

Friso
Oh ja.. en dan zie je ook echt, bij bezig staan er bijvoorbeeld poppetjes te timmeren, eerst staat het skeletje van het huis er, en als het dan wat verder is dan ligt de fundering erbij, uiteindelijk het dak erop en dan is het ding helemaal compleet. Je ziet dan ook echt verandering, een groeiproces.

Rob
Even een gemene vraag: wat heeft dit allemaal met Aquima te maken?

Stef
Niks. Dit heeft met projecten te maken...

Rob
En dan mag je elke tooling gebruiken die je gebruikt...

Stef
Waarschijnlijk wel. Het is eigenlijk het monitoren van deliverables en als er dan een heleboel communicatie...

Friso
Dus het heeft niks te maken met Aquima en alles met wat we doen.

Rob
Ja, dat klopt, maar ik dacht ik stel even de vraag... Dus het is gewoon een spel dat je speelt naast dat je met Aquima bezig bent.

Stef
Aquima is een tool om onder andere bepaalde deliverables te maken.

Rob
Aquima is een tool om huisjes te maken...

Friso
Maar het is wel een tool dat in ieder project dat Everest doet ongeveer gebruikt wordt, dus...

Rob
Dus daar zou je nog wel een link kunnen leggen natuurlijk, maar niet dat als iets het doet, dat dan het huisje gebouwd wordt...

Friso
Goede vraag, ik heb daar nog niet over nagedacht... niet bij stilgestaan.

Rob
Ja, wij? Thema... wij hadden Explore. Focus... kwaliteit en klanttevredenheid binnen het werken met Aquima. Het probleem is het project dat uitgevoerd moet worden... Je wilt een applicatie maken, dat is wat je met de game wilt doen, en dan probeer je de hele applicatie in beeld te houden... wie doet er mee? Je hebt inboorlingen, dat zouden klanten kunnen zijn, iemand die de klant speelt... En je hebt ontdekkingsreizigers, en dat zijn zeg maar de mensen die met Aquima werken. Systeem en proces... systeem, daar hadden we een beetje een probleem mee. Proces... ja, je bent of alleen of een groepje ontdekkingsreizigers die een stukje land ontdekken en die komen dan ergens waar ze op een dorp met inboorlingen stuiten...

Stef
En dat land is een functioneel iets? Een functie of zo?

Rob
Bijvoorbeeld. Er moet wel iemand zijn, dat zou bijvoorbeeld de architect kunnen zijn die in ieder geval globaal het land definiëerd, hoe het eruit ziet, zeg maar.

Friso
Dus die speelt de Dungeon Master, zeg maar...

Rob
Eigenlijk zoiets, zeg maar, dus vooraf gaat hij in het onontdekte land de grote lijnen definiëren. Nouja, dus je komt een dorpje tegen, en je wilt een kolonie gaan stichten in een gebied van dat dorp, en daar ga je mee onderhandelen, en dat zijn de requirements opstellen voor dit stukje, zeg maar, en vervolgens probeer je te voldoen aan een soort onderhandelingscontract dat je hebt met die inboorlingen.

Stef
Maar eigenlijk staat of valt het systeem met de kwaliteit van die contracten,

niet met het vervullen ervan, dat is het spel, maar vooral met 'wat zit er in die contracten', dat is denk ik het belangrijkste, want als die flut zijn, dan is je kwaliteit ook flut.

Friso
Maar dat is toch al... als je je requirements niet goed...

Rob
Daar hadden we nog niet in detail over nagedacht, hoe ga je het definiëren, hoe ga je het koppelen dat je inderdaad oorlog krijgt als het niet voldoet...

Stef
Want dat vind ik wel een mooi concept, omdat je dat visualiseert, zeg maar...

Rob
Ja, en dat je dan eigenlijk of weer opnieuw moet gaan onderhandelen, of dat je weer rebelleert in een gebied...

Stef
Hoe zit het met dat onderhandelen ook weer, wat doe je dan?

Rob
Nou, dan ga je dus afspreken... Het is eigenlijk je test case opstellen... of je test cases opstellen. En dan moet je in dat gebied je tools bij elkaar halen om te voldoen aan dat contract.

Stef
En dat gebied is eigenlijk pas klaar als aan alle voorwaarden van het contract voldaan zijn?

Rob
Ja, want dan mag er een kolonist komen, zeg maar, en dan mag je daar zelf ook een huisje bouwen.

Stef
Zeg maar, te verstaan, na de acceptatietest? Ik noem maar wat...

Rob
Bijvoorbeeld. Definition not done.
En dan ga je door naar het volgende gebiedje. En dat is dan op micro-niveau, en daarnaast heb je, hoe meer experience je hebt, hoe verder je kunt kijken wat er allemaal nog in het vooruitzicht ligt.

Stef
Als je functioneel indeelt, je gaat toch niet indelen in strategisch of tactisch... het zijn toch allemaal deurtjes die ...

Rob
Nee! Nee!

Stef
Waarom zou iemand met meer ervaring wel kunnen zien dat je nog bijvoorbeeld een koppeling moet maken en iemand met minder ervaring niet?

Rob
Omdat iemand met meer ervaring daar waarschijnlijk wat meer dwarsverbanden ziet en bijvoorbeeld ook kan meedenken bij de tooling die je gaat gebruiken in een bepaald gebied, van 'OK, als we het zo doen kun je dat ook nog gebruiken'. Iemand die weinig ervaring heeft die zal daar niet over nadenken, die moet je daar ook nog helemaal niet mee vermoeien, die moet gewoon zijn eigen stukje zien. Dus daar kun je iemand ook een beetje mee beschermen, zeg maar. Aan de andere kant, hoe meer ervaring hij opbouwt, hoe meer hij ook kan zien en dan moet hij leren dat hij ook nog verder kan kijken.

Friso
Wat wel lastig is, denk ik, is hoe deel je een proces wat van nature meerdimensionaal is toch op een tweedimensionaal ding? Daar hadden wij ook zoiets van 'hoe doe je dat eigenlijk?'

Rob
Ja, daar zitten wat uitdagingen, ook van 'hoe kun je op de ene plek ontdekkingsreiziger zijn, maar op de andere plek ook nog? Hoe hou je dat weer in de gaten? En stel dat iemand op de ene plek bezig is, en op de andere weer niet... hoe houd je de onderlinge afhankelijkheden eventueel bij? Dan zou je misschien weer met onderaardse turfjes... Geen idee! Als ik dat zou weten zou ik game designer worden.

Appendix D

The Everest Work Process

After the second game design workshop, Rob Gense (Business Engineer), Stef Demeester (Business Engineer), Friso van der Meer (Technical Engineer), Vincent Jansen (Technical Engineer) and myself spent some time discussing the work process used at Everest. The following is an account of the discussion, provided in Dutch.
Corresponding audio file: IC_A0008 from 13:23

Ilona
Hoe sequentieel loopt dat proces nou binnen Everest, dat projectproces?

Rob
Het is niet sequentieel, dat is nou juist het probleem. Je moet het vooral niet sequentieel in willen steken, volgens mij.

Ilona
Nee, dat dacht ik ook, maar...

Friso
Je hebt wel een proces van dat Scrum, je hebt gewoon een maand, cycles zijn het, gewoon development cycles. Elke maand heb je gewoon weer een nieuwe versie van het systeem staan en dat moet altijd weer getest zijn, en opgeleverd en bij de klant geïnstalleerd. Daar kunnen ze dan iets mee, en in het begin zal het bijna niks doen, maar je automatiseerd telkens die cycle waarmee je toch telkens een stap verder komt. Elke keer doe je weer meer, dan zit er weer nieuwe functionaliteit bij, en zo maak je elke keer een solide basis waar telkens nieuwe elementen bijkomen, dat is het idee erachter.

Rob
Dat is ook heel vaak toegepast.

Stef
Dat is iets, als we het zelf in de hand hebben, wij vanuit Everest proberen het zo te doen, maar je hebt ook klanten die helemaal anders willen werken.

Rob

Die willen echt waterval werken, die willen het eerst op papier...

Stef
Dus die willen alles op papier uitgekookt hebben wat er op het bord staat, dan gaan de handtekeningen eronder en dan pas gaan we bouwen. Dus dan heb je niet cycles, maar dan heb je elke keer van 'doe het iets?' ja, dan klaar en dan het volgende en dan ga je verder. En terwijl bij dat Scrum verhaal doe je eigenlijk korte cycli waarin je eigenlijk alles doet, wat anders is uitgesplitst in activiteiten.

Ilona
Dus de klant heeft eigenlijk wel een vrij bepalende rol in hoe het werkproces gaat verlopen?

Stef
Het hangt ervan af wie daarin... kijk, als Everest gevraagd wordt van 'bouw dit', en als de klant daar een leidende rol in heeft, kan hij ook bepalen hoe dat gebeurt.

Friso
Ja. En wat dat betreft is Everest ook wat meer zo ingestoken dat als projectorganisatie, om dus wat meer mee te gaan in 'hoe wil de klant eigenlijk dat Everest opereert binnen dat project?' Dat dat ook een doel is. Je hebt ook veel organisaties die zeggen 'als jullie dat willen dat is prima, maar wij werken zo.' Punt.

Rob
Ja... Dat laatste willen we misschien wel meer, maar je hebt sommige klanten, die zeggen 'ja, dat is mooi en aardig, maar wij moeten voldoen aan... veelal financiële instellingen... deze set regels, daar hebben wij mee te leven, en als jullie bij ons een project willen doen dan betekent dat dat jullie op deze manier moeten werken, want anders voldoen wij niet aan die regels, en dus...

Friso
En je hebt ook nog zo'n Document Level, ofzo, zo'n speciaal level...

Rob
Wat, CMM bedoel je?

Friso
Ja CMM level.

Shudder from group...

Rob
Maar het probleem is eigenlijk dat je... en dat vond ik wel mooi van die Agile cursus, want ik heb die Agile cursus gedaan, en Scrum is een onderdeel van de Agile manier van programmeren... als je soep maakt, dan is een waterval, een sequentieel proces, goed te doen... lopende band werk, echt lopende band werk, dan kun je dat doen. Maar zo gauw je te maken hebt met veel stakeholders, met wijzigende specs, met weet ik het allemaal, dan kun je eigenlijk niet sequentieel

te werk gaan, dan moet je eigenlijk al iteratief te werk gaan.

Friso
Want iedereen moet het gezien hebben, en dan kun je je state weer een keer veranderen. Dat is eigenlijk het idee.

Rob
Ja, precies. En het probleem is eigenlijk dat op het moment dat je je state verandert en je gaat de volgende fase in, is datgene wat in de vorige fase is bepaald is allang niet meer geldig. Want dat is ingehaald door de actualiteit. Misschien...

Ilona
Maar dat Scrum, impliceert dat dan ook direct meer interactie met de klant?

Stef
Nouja, als het goed is, het liefst wel. Het liefst, sowieso in iedere projectvorm, heb je het liefst een klant die mandaat heeft om een sessie in te nemen, want dan gaat het proces veel sneller, en of dat nu iteratief is of waterval maakt niet zoveel uit maar in principe wil je dat de klant een grote rol heeft. En, zeg maar, het mandaat dat hij iets kan beslissen.

Friso
Plus, het doel is dan op effectiviteit gericht, dus je hebt een moment van tijd waarop de klant weet 'hé, nu mag ik er weer wat over gaan zeggen' en ze hebben dan demo's en die kunnen ze gaan bekijken, dan kan hij zeggen 'dat is mooi, dat is goed, en dat ook, en daar wil ik dat jullie wat aan gaan doen', en dan kan de klant één keer per maand zeggen 'ik wil nu dat jullie de focus op dat onderdeel gaan leggen want dat wil ik hebben'. En dan gaat het project weer een maand bezig, en dan hebben ze ook gewoon een maand lang de focus zonder dat ze elke dag weer de rapportage en het bellen en dat contact nodig hebben. Wij zijn dan ook een maand verder, en dan zijn er wel resultaten. Je ziet dan, de klant krijgt resultaten, en kan dan ook zeggen van 'nou, ik vond dat fijn, ik wil daar meer focus zien aankomende maand'.

Ilona
En dit gebeurt ook echt nu?

Friso
Ja, sommige teams wel, die werken op die manier.

Ilona
Maar sommige teams zijn ook nog anders bezig?

Rob
Ja, je ziet op dit moment bijvoorbeeld teams waar Friso in zit, dat is eigenlijk een intern team met een interne klant... Het Aquima team zelf ook, dat is ook een intern team met een interne klant, die kunnen dat makkelijker doen dan de projecten die bijvoorbeeld die nu al te maken hebben met een lopende situatie bij een bepaalde klant die zegt van 'ja dit is leuk en aardig'... Het probleem bij een klant is dat je met de Agile methodiek van tevoren niet precies weet wat

je krijgt. En ook van tevoren niet precies helemaal weet wat het gaat kosten. Nou, er zijn natuurlijk heel veel klanten die die stap niet durven te wagen, die zeggen niet van 'weet je wat, Everest, hier heb je een bak geld...' met andere woorden, 'we kopen gewoon duizend uur van jullie en wat we dan aan het eind opgeleverd hebben, ja, dat zullen we weleens een keertje zien, dat gaan we dan samen wel bepalen.' Ja, zo werken de meeste klanten niet.

Friso
Maar dat is juist ook precies de paradox van de manier van werken. Omdat ze zo defensief inkrimpen, van 'wij willen echt zeker weten wat het kost' willen ze alles vooraf gespecialiseerd hebben terwijl je dan een explosie krijgt aan complexiteit, en vergaderingen en communicatie... terwijl het ene feit van 'nou, elke maand krijg je weer iets', misschien voldoet het niet helemaal, maar dan heb je wel iets tastbaars om van te zeggen 'nou, iets meer naar links, en dan ben ik tevreden'. Terwijl ze dat dus helemaal vooraf willen doen, en dat is misschien tegenovergesteld aan...

Rob
Nou, het probleem als je vooraf requirements gaat opstellen is dat iedereen denkt 'weet je wat? laat ik nu maar vast roepen dat ik dat wil want anders krijg ik het straks niet meer'. Met als gevolg dat je enorme systemen krijgt, enorm veel requirements wat dus ook gigantisch wordt ingeschat, als het goed is, dat is nog maar de vraag, of het goed wordt ingeschat, en uiteindelijk wordt die functionaliteit.. er is een onderzoek ooit gedaan naar functionaliteit binnen systemen, zeg maar, hoeveel procent daarvan veel wordt gebruikt, en bijna, of iets meer dan de helft wordt zelden of nooit gebruikt, van de functionaliteit die uiteindelijk gebouwd wordt. En er is geloof ik maar iets van 20% dat altijd of vaak gebruikt wordt. En wat je met Agile probeert te doen is dat je alleen maar die functionaliteit oplevert die ook daadwerkelijk van belang is en dat de klant op een gegeven moment denkt van 'ja, ik kan altijd nog meer verzinnen van wat ik hier allemaal in dit systeem zou willen hebben maar met dit systeem kan ik m'n werk doen. Dan kun je veel beter bepalen wanneer en waaraan je je geld uitgeeft.

Stef
Maar alsnog is de methodiek daar geen antwoord op, want als er begonnen wordt met de fancy things kun je dat nog steeds in een mooi Agile traject nog steeds de 80% inbouwen die nooit gebruikt wordt. Dus eigenlijk zouden de leverancier en de klant een gezond verstand moeten hebben dat je eerst met je core bezig gaat, en dat kan je heel goed doen in een Agile omgeving, en dat je pas daarna, nadat je core klaar is, gaat kijken 'wat kan er nog bij', en dan ziet de klant vanzelf van 'nou, het geld is op, OK, maar we hebben wel de core, de rest doen we misschien volgend jaar als we weer geld hebben, ofzo'. Dat is eigenlijk het ideale plaatje. Alleen zie je inderdaad heel erg wat Rob schetst, de klant denkt heel snel in een Big Bang theory, we hebben nu niks, en over drie jaar hebben we een compleet nieuw systeem dat alles kan. En dan moeten ze drie jaar van tevoren bedenken wat ze drie jaar later nodig gaan hebben, dat is heel erg moeilijk, zeg maar.

Rob
In een wereld waarin de wetgeving veranderd, waarin de kredietcrisis voorbij komt, waar...

Ilona
En in die waterval projecten bij die klanten die dat graag willen, zeg maar, is er dan wel veel interactie met die klanten? Want die mensen zitten daar gewoon op locatie?

Rob
Het gevaar van een waterval traject is dat je in het begin veel interactie hebt, en dat er daarna een soort radiostilte ontstaat, waarna je bijvoorbeeld na drie, vier, vijf of zes maanden op een gegeven moment een systeem moet opleveren, waarbij hij de afspraken die hij ooit heeft gemaakt alweer helemaal vergeten is.

Ilona
Maar er is dan niet in die tussentijd even zoiets van 'klant, zover zijn we en vind je het goed?'

Rob
Ja, vaak op procedure-niveau, highlight reports, van 'we zijn op 50% van de applicatie'. Maar zo hebben wij trouwens bijna nooit gewerkt, altijd met status.

Stef
We hebben wel veel waterval gedaan, want zoals Rob net ook al zei, dat interactieve, of dat cyclisch werken, dat is, omdat de klant niet weet wat hij uiteindelijk gaat krijgen, is heel moeilijk te verkopen.

Friso
Alleen, ja, wat je echt hebt, van 'die twee projecten, daar rolt dat systeem uit', dat is gewoon ultra-complex, terwijl ze met veel minder in het begin hadden gewild. Tenminste, dat bleek daar dus ook, van 'hallo, zit dit er allemaal in, we zochten eigenlijk alleen maar dat'. Met een paar uitzonderingen nagelaten, maar het is helemaal opgeblazen, zeg maar.

Rob
Het vreemde is inderdaad, veel klanten willen dat, want dan denken ze dat ze veel controle hebben, maar feitelijk zijn ze op een gegeven moment...

Stef
... de controle kwijt, en zo'n groot project krijgt een eigen dynamiek, dan zijn ze sowieso de controle kwijt, want het moet af, anders hebben ze niks. En dan hoor je van die monster dingen die inderdaad weer drie jaar uitlopen. Alleen zie je nu ook dat de projecten, de dingen waar we het wel toepassen, Everest zelf de opdrachtgever is, omdat Everest dat zelf ook wil, dus de opdrachtgever moet het eigenlijk willen.

Rob
Nouja, of je moet het goed verkopen... En het verschil van Everest is, er is een groeiperiode geweest en we zijn helemaal aan het afwijken hiervan... omdat we vroeger natuurlijk veel kleine projecten deden op niet-core van de business, van bedrijven, daar hadden we veel meer ruimte voor interactiviteit en veel meer ruimte ook voor cyclisch ontwikkelen, dan nu, waar we veel meer op de core

business van bedrijven... waar we echt op bedrijfskritische applicaties aan het ontwikkelen zijn, terwijl we vroeger vaak de nice-to-haves hadden, die wel leuk waren en daar was veel meer ruimte om 'ga maar eens kijken waar je uitkomt'.

Stef
In voor de bedrijven veel meer innovatieve hoek, we gaan eens proberen om aan fuzzy logic te werken, of met dit of met dat...

Ilona
Dan had ik nog één vraagje. Ik moet straks natuurlijk een ontwerp gaan maken met al deze input... Wat zouden jullie zeggen dat de core functionaliteit van Aquima is, die er zeker in moet zitten... Wat is de core functionaliteit nu, of is er geen core, zit er niets overbodigs in?

Stef
Nou, wij zitten nu, wij met z'n drieën, in een niet-project spel, waar we eigenlijk relatief weinig nieuws maken maar voortbouwen op bestaande dingen, en wat wij heel erg hebben, althans ik, is dat Aquima heel erg gericht is op dingen in elkaar sleutelen, maar veel minder op onderzoeken van 'waar zit nu een fout', of impactanalyses. Wij hoeven niet zoveel nieuws in elkaar te klikken, of hele kleine stukjes, maar Aquima is vooral gericht op het maken van dingen en niet op het onderhouden. Dus voor ons zal de core anders zijn dan...

Friso
Maar dat is niet wat je zoekt, is het wel?

Stef
Nee, maar wat ik hiermee wil aangeven is dat de core functionaliteit die ik zal roepen hoogstwaarschijnlijk heel anders zou zijn dan iemand die een project maakt.

Rob
Ik denk dat voor ons de core functionaliteit inzicht zou zijn.

Stef
Ja, inzicht en impact bepaling. Dat zou voor ons de core zijn.

Rob
Ik ga iets aanpassen, ik ga iets wijzigen... een attribuut... maar wat valt er allemaal onder? Wat voor effecten heeft die?

Stef
En in een project zal dat veel minder als core beschouwd worden.

Friso
Dat is wat jij zegt, dat is gewoon net zoals met een programmeertaal. Eerst heb je writability: kan ik het intikken en zou het dan iets gaan doen? Daarna heb je readability: als ik het dan lees, begrijp ik dan wat er gebeurt? En daarna het je reliability, dus eigenlijk maintenance: er gaat iets fout, waar gaat het fout? En als je het moet aanpassen, hoe kan ik dan zien, inperken dat er bepaalde

fouten ontstaan? Dat soort dingen.

Rob
Maar ik denk dat je je vergist, ik denk dat ook in die grotere projecten dit nog steeds van toepassing is.

Stef
Natuurlijk. Want op een bepaald moment gaan zij ook bugs oplossen.

Rob
Want ook nieuwbouw projecten, die worden bij ons zo groot. Maar je ziet inzicht wel steeds belangrijker worden.

Stef
Maar de focus van degenen die Studio bouwen ligt vooral op het creëren van modellen, van...

Friso
Vroeger was het schrijven, gooi over de muur en klaar.

Stef
Dus ik vind, ik stem op Rob, inzicht vind ik qua functionaliteit van Aquima core, inzicht van wat je erin gestopt hebt.

Friso
Maar jij zoekt een minimale basis voor je game?

Ilona
Ongeveer ja, van 'wat moet er nou per se in dat spel zitten?'

Friso
Dan horen deze dingen er niet per se bij... oh, wat er in het *spel* moet zitten...

Ilona
Nouja, het Aquima spel... Het spel moet eigenlijk een beetje gezien worden als een die er bovenop zou draaien en als er geïmplementeerd moet worden dan moeten er natuurlijk nog steeds Aquima schermen naar voren komen...

Stef
Dan vind ik nog steeds inzicht een core iets, zeker in de versie die wij bedacht hadden gaat het alleen maar om inzicht eigenlijk, maar ook vanuit mijn werk is het iets wat ik mis in Aquima, en als er een spel voor zou zijn, of een andere expliciete representatie is inzicht in wat er onder de hoed zit, inzicht in kwaliteit ook heel belangrijk.

Rob
Het hangt daar iets mee samen, kwaliteit in de zin van 'hoe zou ik in mijn game een zodanig iets kunnen creëren wat kwaliteit afdwingt?'

Stef

Maar het is wel een heel vaag iets, kwaliteit is een heel vaag woord...

Rob
Hoe kun je op voorhand definiëren wanneer iets goed is en hoe kan ik in de loop van mijn traject waarborgen dat alles wat ik gemaakt heb nog steeds goed is? En dan zit je tegen het inzicht aan. Ik denk dat dat...

Stef
Ja, dat zou ook, als ik ergens veel punten aan moest geven, zou ik daar veel punten aan geven, meer dan aan het creëren van een metamodel.

Rob
En wat denk jij, Vincent?

Friso
Nou, wat ik het belangrijkste zou vinden is, want je zag net, het ziet er niet uit, die voorbeeld professional serious gaming, dat is foeilelijk gewoon, krijg je dat voor je neus, dan denk je 'nou, moet ik daar de hele dag in klikken', dat nodigt echt totaal niet uit.

Rob
Vind je?

Friso
Die twee voorbeelden? Nee, dat...

Rob
Er zijn sommige games die... Ik ken 2D games die ik leuker vind dan menig 3D game...

Friso
De enige die ik leuk vond waren die poppetjes in dat bos. Maar die andere, met dat whiteboard, het moet er echt vrolijk en uitdagend en uitnodigend zijn.

Rob
Nouja, ik weet niet, ik kan me voorstellen dat je ouderwetse Donkey Kong, 2D, dat dat een leuker spel oplevert dan sommige fancy gelikte lijkende 3D games.

Friso
Ja precies, dus jouw conclusie is dan eigenlijk, even geen 3D, want 3D gaat ook relatief meer resources in qua geld dan met 2D, en met 2D kun je met heel veel basis dingen al toe, dat klopt, dat is dan niet zo'n punt, maar...

Stef
Het moet uitnodigend smoelen... Dus simpel Donkey Kong of heel realistisch... het moet wel goed smoelen, je moet je er lekker bij voelen, feel good oproepen, en niet iets saais, grijs... vrolijkheid oproepen.

Ilona
Goed, dank jullie wel.

Appendix E

Aquimatization: Game Design

The detailed design of the Aquimatization game. The format of the design is largely based on Järvinen's game analysis framework [Jarv 07a] as described in chapter 2 [2.4]. However, some slight modifications have been made. Useful aspects from other game design formats [De B 08], [C.1] have been included to make the design as complete and all-encompassing as possible.

E.1 Components

In this section, an overview of the main game objects the player can interact with during play is given. Also, a point-wise description of how interaction proceeds is included.

- **World map**
  - The world map represents all ongoing projects using the game.
  - The map shows how far project sites have evolved.
  - Beginning projects are shown with wooden huts or tents, whereas evolving projects are represented by stone houses or castles, if in their final stages.
  - Clicking on a project site will take the player to that site by zooming in to a level of sufficient detail.
  - If the project selected is not the one the player is working on, he will meet a guard at the site.
  - The guard will have to verify why the player wants access to the site that he is not working on.
  - The project leader will be notified of such requests, and can decide whether to allow it or not through the guard.

  *A world map seems the most **intuitive** way to provide insight in the different ongoing projects. By being able to see the projects' progress, project leaders and managers will have a quick way of judging progress. This contributes to the game's **accessibility**. Besides* *that, an ancient map looks exotic and appeals to many people's imagination, inducing* ***excitement****.*

- **Minions**
  - Minions embody the requirements.
  - Minions must develop relationships with each other.
  - Minions go through phases of evolution, which correspond to the development cycles.
  - Minions must be fed and taken good care of if they are to evolve.
  - Minions are created from scratch as information about requirements is entered.
  - Minions start small and grow as information is added.
  - Elaborating existing information feeds the minion.
  - Adding a new information category gives the minion an extra body part.
  - Neglected minions will start whining to make themselves heard.
  - If neglect continues, the minion will experience physical deterioration.
  - Minions present their information if they are clicked on.
  - Minions must be reviewed by other team members.
  - Minions must be ready for labour before work on the actual town can begin.
  - Minions carry out the actual building work.
  - Minions only work on the buildings they embody the requirements of.
  - Minions can only become ready for work after...
    - They have been reviewed by the team.
    - They contain all minimally required parts of the requirements description.
  - Related minions must be grouped as such.
  - Relationships are shown by matching physical features (colour, shape, horns etc.).
  - There are no restrictions on what and how many body parts the minion can have.
  - Minions are coupled to the buildings they embody the requirements of.
  - Minions are commanded by the build supervisors (business engineers).
  - Minions can be seen walking around and helping with other buildings, but will always return to their base house.
  - Minions on strike are a sign that something is wrong.
- **Town Hall**

- The town hall represents a review facility.
- Virtual meetings are always held in the town hall.
- The town hall is meant for *organised* collaboration.
- The minimum number of people to attend a meeting is two.
- The town hall contains facilities for:
  - Real-time collaborative modelling
  - Real-time file sharing
  - Real-time meetings
  - Real-time text editing
- All creations made in the town hall can immediately be stored in the town library.
- The town hall starts small (as a tent or cave) and evolves along with the project.

- **Library**
  - The library is a facility for knowledge sharing and storage of documentation.
  - The library stores:
    - Shared knowledge and best practices
    - Documentation from the town hall
    - Test cases constructed in advance
    - Bugs and solutions, with a link to the corresponding building
    - Design decisions
    - Plans (weekly plans, monthly plans, whatever is created to guide the way the players spend their time)
    - Time-tracking (the amount of time spend on each module)
    - Reviews
    - Demos
    - Project issues
  - To document project issues, the library has a direct link to the Issue Management System (IMS).
  - Documented entries of all kinds are represented as books on shelves.
  - Books can contain text, audio and video material.
  - The above mentioned categories are grouped in different sections of the library.
  - Clicking on a book visualises its contents on the screen.
  - Contents have to be rated.
  - Library books are available to everyone participating in the project.
  - Library books can be made available across projects if the members choose to do so.
  - A special section will be dedicated to cross-project shared knowledge.

  - The library starts small (as a tent or cave) and evolves along with the project.

    This specific method for knowledge sharing has been chosen because it ***rewards*** *players by gaining points and respect from fellow team members. Players who share their best practices and receive positive feedback will feel* ***competent****, their specialisms are being acknowledged. Moreover, players will be* ***challenged*** *to learn from other people's experiences. Having a central depot for knowledge also relieves experienced players because many answers to simple questions can be found in the library. Finally,* ***aesthetics*** *also play a role here. A library full of ancient-looking books is again a visually exotic environment to be in.*

- **Buildings**
  - Buildings embody the application's modules.
  - Buildings can be grouped into 'quarters' to represent the entire module.
  - The whole town represents the entire application being created.
  - The town hall and library are separate features and are not part of the application.
  - Buildings come in different types (in order of importance, starting with least important):
    Importance in this section is defined as: how frequently is the object used by the user or by other objects, as in the eventual operational system? The measure to which it supports other objects also plays a role.
    1. Hut
    2. House
    3. Shop
    4. Church
    5. Castle
  - The type of building represents the importance of the module.
  - Clicking on a building will take the player inside, where he can do more detailed modification.
  - Positive or negative reviews can gradually up- or downgrade the building's type, ie. from house to shop if positive, or from house to hut if negative.

- **Features**
  - Features embody the entities and attributes making up the separate modules.
  - A building has many editable features (rooms, roofs, doors, windows, interior etc.).
  - By editing features, the underlying model is changed as well.

  - Feature editing can only be used for creating the basic model structures.
  - Detailed editing must be done through the Aquima Studio graphical modelling interface.
  - Features evolve and become more visually sophisticated as the module develops.

- **Road**
  - Roads show how buildings (modules) are connected to each other.
  - Roads can come in different types (in order of importance, starting with least important):
    * Mud
    * Gravel
    * Paved
  - The type of road represents the importance and soundness of the connection.
  - Clicking on the road shows the connection's status and properties.
  - Positive or negative reviews can gradually up- or downgrade the road's type, ie. from gravel to paved if positive, or from gravel to mud if negative.
  - Roads support *project* activity, such as minions walking on it, carts being driven along it.
  - Activity shows that the project is still functioning as it should.
  - No activity on the road is a sign that something is wrong. The connection may be faulty, or someone is waiting for input.

    The structure of minions and buildings with features grouped in rooms and connections with other buildings has specifically been chosen to allow more insight in the structure of the application and the flow of the development process, to reduce ***frustrations*** *for people burdened with application maintenance and project leaders. All the necessary information can be accessed from one, visually appealing interface.*

- **Money and Resources**
  - Money and resources are directly related.
  - Resources are available as long as there is money from the client.
  - The amount of money left also shows how much of the resources is left.
  - Using resources automatically diminishes the amount of money left.
  - Using resources is visualised by minions going off to chop wood and set up the buildings.
  - Resources must last until the project is finished, so the challenge is to act sustainably.
  - The only way to obtain more resources is if the client is willing to invest more money in the project.

- The total amount of available money is transformed into physical resources and game currency.
- The total amount of money (and resources) is available to the team as a whole.
- Resources can be found plentifully in the environment (trees, water, rocks, grain, gold, etc.).
- Recycling obsolete buildings saves money and resources.
- The more money and resources are left at the end of the project, the greater the profits.
- If there are no more resources and the project is not finished, it will have failed.

E.2 Environment/Game world

The game environment is the fictitious planet Aquimars, a world not unlike the Earth except for the fantastic creatures that can live there. The setting is rather idealistic, ranging from tropical island to rough wilderness in which civilised colonies have to be built. Such graphical aesthetics make the player feel *comfortable* and encourages him to further explore the world. There are no civilisations living there yet, the world only contains natural features such as woodlands, mountains and rivers.

The player can oversee the entire world by means of a map. On this map, different project locations are marked. The game world supports all ongoing projects. This allows for easy communication with other project teams. However, only their outlines and state of evolution can be clearly seen, employees have no access to other projects' modules and features.

The game world provides all the resources needed for the completion of the project. The amount of available resources is determined by the amount of money the client invests in the project.

The game world has a link to the conventional, graphical Aquima Studio modelling interface so that detailed model modification can be done in familiar ways.

E.3 Gameplay Mechanics

- **Navigation**
  Navigating through the world, through the towns and cities, walking the paths between houses.

- **Selecting**
  Selecting buildings for road construction between them, minion body parts, books in the library, other avatars for collaboration, files to share and upload.

- **Contract**
  Two or more players collaborating on a task without making an official agreement, but instead it happens through an operation acknowledged by the game system, such as cooperation on a model, on a house, assembling teams, reviewing each other's work.

- **Building**
  Creating houses, libraries, town halls, castles etc. by pressing the 'Create building' button. This action is mainly carried out by the player's workforces, but he issues the command himself.

- **Placing**
  Placing modules in the game environment, such as placing houses, creating minions. Placement can be carefully planned, or quite random, if the player just wants to create all the modules required at first and postpone filling them in.

- **Arranging**
  Arranging order, location or composition of town modules, if initial placement has been random (ie. pressing the 'Create building' button several times) or if it does not work out as desired.

- **Choosing**
  The player is presented with informed or uninformed choice between a number of options. Choosing what type of building to build, where to build it, how to set up a requirement, how to draw the road map etc.

- **Substituting**
  Changing modules or replacing them with different ones if the client changes his mind or if things do not work properly. Editing buildings when specifications change.

- **Dialogue**
  Communicating with team members, with client and other stakeholders by means of the chat tool or making posts which people can read at a later time if they are not online at that specific moment or if the aim is to reach a large group.

- **Allocating**
  Managing quantifiable resources in a game, such as modules or time and effort in the hope of a reward.

- **Discarding**
  If something is obsolete or has gone wrong.

- **Submitting**
  When a deliverable is finished and up for review.

- **Reconnaissance**
  Gathering information or making inquiries about surroundings, challenges or other players. It is what account managers do in the pre-project phase, but during building (project development) inquiries have to be made continuously in the form of communication with the mayor, consulting the town library or collaborating with team members.

- **Role-assigning**
  Assigning and taking on a certain role after a meeting for a certain period of time. The role may change as suits the needs of the project.

- **Discharging**
  Cancellation of contracts. If team member does not suffice or work properly.

- **Collecting**
  Collecting information during the course of the game from the client, from the shared library, from written books or the Internet.

- **Timing**
  Enact an action in a specific time frame in order to achieve best results. Making the deadline, having sufficient information available at the right time, to keep others happy and working!

- **Manipulating**
  Possessing, controlling and changing a module you have created, or convincing other people to do things differently.

- **Negotiation**
  The work process is iterative, therefore there will be continuous negotiation with relevant stakeholders about the process, the product and the progress being made. The value of input from both parties weighs more heavily during negotiation than during manipulation.

- **Storytelling**
  The entire game's narrative aspect, the story of the explorers on Aquimars. An important mechanic to induce *immersion*.

- **Trading**
  Players do not individually have money at their disposal, there is just a virtual indicator of how much money there is available. There are only resources, every time something is taken the indicator goes down. Leftover resources can be traded with team mates. This seems the most realistic way to build an economy, because money only flows out of the system, it cannot be earned back since the money supply is as finite as the client decides it to be. This relates back to the player needs as described by Ghozland [Ghoz 07] and how carefully needs such as resources have to be managed.

- **Exploring**
  Exploring land and finding suitable sites.

- **Enclosing**
  Enclosing part of the game environment and or modules in order to gain control or possession over them, such as with building sites.

- **Herding**
  A means to guide a game module to a certain area or in its development, such as with the commanding of the forces of minions and keeping related ones in their groups.

E.4 Game Flow

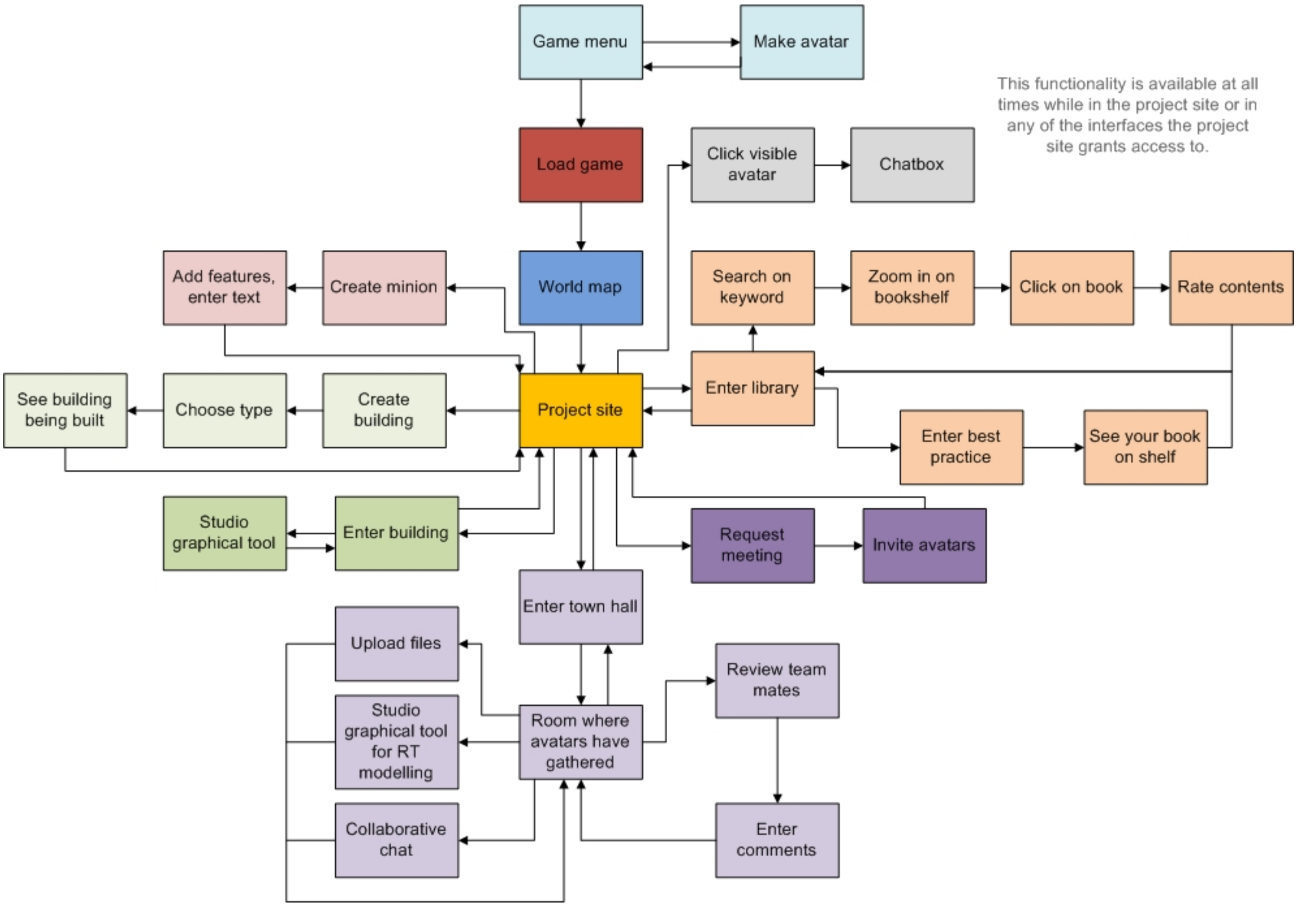


Figure E.1: An overview of the game's basic flow.

E.5 Game Rules

- **Strikes**
  - Strikes occur during automated testing, when constructed buildings...
    - Are syntactically incorrect.
    - Do not match previously constructed test cases.
  - The partly system-driven minions go on strike, not the players.
  - Strikes can be dismantled by solving the problem with the building.
  - Strikes negatively affect the offending player's score.
- **Disaster**
  - Disaster can strike if the mayor is not happy with what is being created.

  - Disaster comes in the form of a storm.
  - Disaster announces itself by dark, threatening clouds packing together.
  - Storms can blow over if immediate contact with the mayor is sought and the problem is solved.
  - If the problem persists, the storm will still strike.
  - Storm does visible damage to the buildings, but will not harm the underlying models.
  - Damage is restored by the minions in a short time.
  - Storm negatively affects all players' scores.

- **Attacks**
  - Attacks happen when a team member is not happy with another team member's work.
  - Attacks are carried out by the team member's minions.
  - The attacking team member gives his minions a message.
  - Attacks can be counteracted by the player's own minions.
  - The attacked team member's minions defeat the other's minions.
  - The message is carried across.
  - The attacked team member must solve the problem in collaboration with the attacking member.
  - The number of attacks in a project are taken into account in the final project review done by the project leader.
  - Attacks negatively affect the group's bonus score (to avoid individuals hampering each other).

    *Strikes, disaster and attacks are another example of the constant **'fear and relief'** emotion cycle. It is the player's **challenge** to play in such a way that these negative events do not happen. If he succeeds, he will once again feel victorious (**autonomy and competence**) because of his achievement. Firstly, motivation comes from fear, after succeeding, motivation comes from a desire for more victory.*

- **Virtual meetings**
  - Daily meetings
    - All players meet in the town hall at the start of each day.
    - This meeting does not involve the client.
    - Current status and the day's work are discussed.
  - Weekly meetings
    - All players meet in the town hall once a week for a review session.
    - The client is present at this session.
    - Results, progress and further steps are discussed.
    - Peer reviews are done during this meeting.

- Monthly meetings
    - All players and client stakeholders are present at this meeting.
    - Deliverables are presented and reviewed.
    - The past month's work process is evaluated.
    - A new plan for the coming month is drawn up.

- **Leveling up**
  - A player can reach a higher level when...
    - He has gathered enough experience points.
    - His peers have given him positive ratings.
  - The two resulting scores are combined.

- **Scoring**
  - Experience points are gained if:
    - There is lots of activity on the roads connected to the modules you have created.
      50 points per hour of road activity. The choice to take time as a measure rather than to reward per individual road was chosen so that players will not feel inclined to create as many roads as possible, but rather to create viable roads quickly. The challenge here is that with each hour of lost activity time, the player misses experience points.
    - Your minions are working actively on your houses.
      50 points per hour of minion activity.
    - You have few strikes in your town.
      1000 bonus points are gained at the end of each cycle. For each strike, 50 points will be deducted from the bonus.
    - There are few bad events initiated by the mayor.
      A bad event with the mayor deducts 100 points from the bonus score.
    - The house is 'valid' (ie. adheres to requirements and matches the test case).
      For each valid construction, 100 points are gained. The danger of players just creating many houses for this is much smaller, since unnecessary houses are unlikely to be described in the requirements. If that should be the case, something has gone wrong during the requirements review sessions.
    - Your contributions to the shared knowledge library are positively rated.
      The maximum rating is 5 stars. For each star your contribution receives, you receive 50 points.
    - The ratings you receive during the review sessions in the town hall are positive.
      The maximum rating is 5 stars. For each star you receive, 100 points are added to your score. This high amount of points is to emphasise the importance of good reviewing. Competitiveness

    ensures that players will not simply give good ratings because they happen to like the person they are reviewing. Yet the prospect of being able to gain so many points needs *will motivate the player to do the best he can.*

  - The minions (requirements) are productive.
    For each productive requirement under your control, you receive an extra 300 points. Again a high number of points to be gained, to ensure the players take their requirements seriously.

- Mission points are awarded if:
  - The mayor is happy after every delivery cycle.
    A happy mayor results in 500 points for each team member. The mayor can decide to reduce the number of points if he is generally happy, but has some points about which he was not happy.

- Peer reviewing is compulsory during every weekly meeting.
  - All team members should rate and review each other.
  - The validity of this rating is weighed against the rater's achieved level. If you have lots of experience (and thereby a high level), your rating will receive a heavier weight than if you are new to the game. This is an automated process.
  - Players with high levels review the comments and can adjust the automatically assigned weights. If they consider a newcomer's criteria to be extremely useful, they should assign a heavier weight to it. Contrarily, if a review from a member with a high level is considered useless by another with a high(er) level, the weighting should be accordingly adjusted.

- Object reviewing is compulsory during every weekly meeting.
  - All team members should review each other's objects.
  - Buildings must be reviewed in order to maintain solid and up-to-date.
  - Roads must be reviewed to check the validity of the connection.
  - Positive reviews contribute to the upgrading (in status) of buildings and roads.
  - Negative reviews contribute to the downgrading (in status) of buildings and roads.

- Scores also count for a total group score.
  - At the end of each cycle, the total of the individual scores is added up to create an overall group score.
  - The lead supervisor reviews the overall project progress and adds a bonus score to the group score.
  - This overall total score is compared to other groups' scores to make a group comparison.
  - The best groups are mentioned in Everest newsletters / during company gatherings.

- The score marks an individual's progress in the game.

- The score serves to contribute to the team's entire performance.

  - Score is NOT a mechanism used for individual (monetary) bonuses.
  - If desired, such rewards should be based on team performance to avoid undesired angry competition between colleagues.
  - Score is only visible to oneself and the project leader, so that he can track overall progress and possibly offer help if things are not going well for someone.

  Assigning a score is merely a mechanism to enhance ***competitiveness*** *and to motivate by calling on* ***player needs****: the need to improve themselves.*

- **(Automated) Testing**
  - Buildings must constantly be matched against previously constructed test cases.
  - This happens automatically, so that the player will only notice errors if the forces go on strike.
  - Players can also run test scripts manually.
  - If the test succeeds, the player can continue working.
  - If the test fails, the part of the building containing the bug will collapse.
  - The collapsed part is only visualisation and does not harm the underlying model.
  - The collapsed part is repaired by the building workforces as soon as the player fixes the bug.

E.6 Game Actions

This overview of actions is a suggestion for the complete replacement of conventional modelling by game actions. However, more detailed research into how details such as attributes can be clearly represented by game objects is needed before gaming can responsibly take over conventional modelling. I have specifically chosen to concentrate more on a project development game in this thesis, because the workshop participants unanimously presented such game ideas during the workshops.
For now, conventional modelling will not be entirely replaced by game actions. After creating buildings and storeys, and maybe entities, detailed model modification will still be done with the graphical modelling editor in Aquima Studio. The room will be dressed up with features automatically as the model changes. The player can, however, decorate the room, or even the house, depending on the kind of modifications done, with the available features as a reward for creating a good model.

Aquima Studio action	**Aquimatization action**
Create module	Create suburb (selection of related buildings).
Create page	Create building (only the outline is created. The inner space is empty).
Create container	Create storey (for every container on page, a storey is added to the building).
Create entity	Create rooms (only the partitioning walls are created. The space inside the room is empty. Related rooms are adjacent).
Create attribute	Create features (dress up the room).
Create relation/page flow	Create road (between linked structures, they can be buildings or suburbs).
Execute model	Click on an icon in the interface that takes the player straight to the conventional Portal interface.
Set business rule/decision table/precondition	Click on a room/room feature and enter the business rule/decision table in the conventional way. The business rule/decision table will be stored with the corresponding room/feature and a marking will appear.
Put entity in container	Happens automatically, as soon as a room is created inside certain building on certain floor.
Put container on page	Happens automatically, as soon as a floor is created in a certain building.

Table E.1: Mapping of Studio actions to game actions

E.7 Information

The game interface has several mechanisms for showing game statistics.

Always available to all

- **Mayor satisfaction meter**
  - A progress bar is constantly in sight depicting the status of the mayor's satisfaction.
- **Deadlines**
  - A button is available showing deadline statistics at all times.
  - There is a deadline representing the time left until overall project completion.
  - There is a deadline showing the time left for the current development cycle.
  - There are deadlines set until certain reviews have to be in.
  - The deadline gradually counts down as the project progresses.
  - Time is counted in days.
  - The deadline can only be stretched if the client should decide to give the project more time.
- **Resource statistics**
  - A button is available which shows the amount of money left at all times.
  - The statistics also show the rate at which resources are being used up.
- **Player statistics**
  - There is a button for viewing personal player statistics at all times, tracking...
    - Number of houses built.
    - Number of minions created.
    - Minion activity.
    - Minion disturbances.
    - Positive and negative reviews received.
    - Number of library contributions.
    - Meetings attended.
- **Score**
  - A banner showing the player's personal score is visible at all times.
  - The banner also shows the player's achievement level.

Only available to the project leader

The project leader has additional facilities for monitoring the team's progress at all times. Apart from being able to see the whole town and its activity on the map, like the business engineers can, he can also view and compare all his team members' statistics. That way he can see who is doing well, who might need some help and will help the project leader discover individual members' expertise, so that task division can be done more precisely.
This will help enhance feelings of *autonomy* and *competence*, since the approach recognises the players' abilities and talents. As a consequence, the players will feel motivated to prove their value to their superiors.

E.8 Interface

E.8.1 Input and output devices

Providing input will happen by means of keyboard and mouse. A lot of text will have to be entered, for instance for:

- Describing requirements
- Communicating with other players
- Posting messages
- Adding descriptions to house attributes
- Adding descriptions to design documents
- Knowledge sharing in the library
- Writing test cases

The mouse will be used for selecting objects by means of clicking, dragging certain objects, changing views, directing navigation. Interaction happens mainly through the game objects.
During virtual meetings, input devices such as a microphone and speakers are desirable, so that communication can happen vocally and does not require typing. An advantage of typing, though, is that everything that has been said can immediately be stored as documentation. Vocal communication can be recorded and stored, but this decision is still open.

Output will be presented to the player via the monitor, and possibly through the speakers during virtual meetings.

These interaction devices have been chosen because Everest employees are already familiar with mouse and keyboard and the associated actions of clicking, selecting and dragging. Moreover, they seem the most intuitive way to facilitate the required game mechanics.

E.8.2 Screens and icons

General interface objects available to all players

The general interface [E.7] has been designed with *intuitiveness* as a main goal [2.5]. The interface has been kept relatively simple, with the main controller interface staying the same between the different modes, to prevent players having to learn a different interface every time they switch between modes. The number of buttons has been kept minimal, with intuitive icons representing a picture of the action they represent rather than a textual description. Each object in the interface is meant to be self-explanatory as far as its goal is concerned.
The rules are largely integrated in the possible game actions. When dealing with issues such as storms or strikes, the game system timely provides the information.
For a visual impression of the interfaces, see [E.11].

- The *chatbox* visualises all conversation between players. Chatting can be done one-to-one or with multiple participants.
- The *message board* shows all messages that have been posted by players. Messages can be entered and posted at all times, and will remain visible when the player goes offline.
- *Buttons* are available for the following functions:
  - Show resource statistics.
  - Show player statistics.
  - Show deadlines.
  - Show who is currently online.
  - Create minion (will take the player to the minion creator interface).
  - Create building (will let the player choose the type of building to be constructed).
  - Create road (will create a road between selected buildings, these can be more than two).
  - Share knowledge (will take the player to the library).
  - Request a meeting (allows the player to send a meeting request to selected players by selecting their avatars).
- The *in-game map* shows the overall state of the project, how far along it has evolved.
- The *mayor satisfaction meter* showing the state of the mayor's feelings about the town.
- The *score* banner showing the player's personal score.

Apart from the general interface, with each task the player arrives in a different mode, with different functionality. The following is an overview of the features in each mode.

Minion creator mode [E.6]

- Feed minion (adds information on the requirement, makes the minion grow bigger and stronger)
- Body (allows for addition / change of body)
- Paws (allows for addition / change of paws)
- Wings (allows for addition / change of wings)
- Feet (allows for addition / change of feet)
- Head (allows for addition / change of head)
- Coat (allows for addition / change of coat)

Build mode [E.7]

- Build house (upon click, list of different types of houses appears. House can be selected and dragged to the site the user indicates).
- Add attribute (upon click, list of different attributes appears (such as window-sill, hearth, type of roof, type of door etc., which can be clicked on and the house will be changed).
- Enter modelling interface [E.8](the business engineer / build supervisor has a possibility to edit the model in a traditional Visio-like interface. The client, however, only has access to the game world)

Town hall mode [E.4, E.5]

- Upload (for slide show/presentation, mockups, model diagrams).
- Access to modelling interface for real-time modification and real-time modelling.
- Ability to click on team members' avatars for peer reviewing.

Library mode [E.9, E.10]

- Enter knowledge to share.
- Search for information.
- See shelves of clickable books.
- Rate contents.
- Provide additional information to existing books.

Project leader interface

The project leader has some additional features in his interface. He can do everything a business engineer can, and on top of that he has a button which will take him into *Team progress monitoring* mode.
Besides that, the project leader will have access to a special kind of *Peer reviewing mode.* He does the actual reviewing in the same way as the other team members. However, after the review session, he will receive all the reviews (including the names of the reviewer and the one being reviewed). He can then check them, assign weights of importance to them, and send them on to their recipients, removing the names of the reviewers in the process. This is to prevent tensions between team members.

Team progress monitoring mode

- A list of team members' avatars is available at all times.
- Clicking on an avatar allows the project leader to see all above mentioned statistics for that player.
- Different graphs are available for the above mentioned categories of player statistics.

Peer reviewing mode

- Open a review.
- Assign a weight of importance to the review.
- Remove reviewer names.
- Send them to recipients.

Client interface

The client can see all activity going on in the town. He can see the status and size of buildings, can see whether minions are at work or on strike or fighting. All of this represents the status of the project. He has access to the town hall and can lead meetings. He can chat with team members and provide reviews and comments. He is not entitled to edit buildings or minions.
The client does have one additional feature, the *Satisfaction mode.* After each check-up on the project's status, he can enter his degree of satisfaction.

Satisfaction mode

- Rate the project progress.
- Rate the project status.
- Rate the team's attitude.

All these ratings combined make up a certain satisfaction score, which will be directly visible for the project team. If the score drops below a certain threshold, a warning in the form of a storm will automatically be generated by the system, warning the team to take action and resatisfy the mayor.

Visitor interface

A visitor basically enters the game in a read-only mode. He can view the buildings and minions, can chat with the team members and can attend meetings, but he cannot edit anything. The only way a visitor can participate in the process is by providing reviews and comments on delivered or ongoing work.

E.9 Players

E.9.1 Characters

- Player characters are represented by an avatar.
- Characters take on a role.
- Roles can change during the project, if certain tasks are completed or if the lead supervisor decides so.
- Game roles represent real project roles.

E.9.2 Roles

Roles serve to enhance *player immersion* in the game, and to keep the player focused on what his duty within the game is.

- **Everest**
  - Project/teamleader = Lead supervisor
    Checks whether the project progresses as desired, if all workforces remain active, that there are no strikes or attacks and that the mayor stays happy. Ensures the meetings take place, invites necessary stakeholders and checks all reviews, comments and final products.
  - Technical architect = Craft supervisor
    Checks if the to be built application will fit in technically with the client's architecture and infrastructure, he is ultimately responsible for the technical side of the project. Ensures the foundations are in order before the buildings are constructed.
  - Technical engineer = Craftsman
    Responsible for the construction of foundations, and the game as a whole.
  - Business architect = Architect
    Ensures the models and road maps are drawn up correctly.
  - Requirements business engineer = Shepherd
    Ultimately responsible for the herds of minions, that they are grouped properly and which ones are most important. This role can be taken by a business engineer, an external information analyst or the client himself, depending on who provides the requirements.
  - Modeller business engineer = Build supervisor
    Sets the workforces to work building houses. He is the one who issues the order for a house to be build and can adapt it to his liking. Must

also make sure his forces keep working, that the connection between his houses and others are properly maintained and active and that no one is waiting for information.

- **Client**

  - Project leader/sponsor = Mayor
    This is usually the project leader from the client side who plays the part of sponsor part-time.

  - Other client roles = Visitors
    There is no need to assign the other client roles active game roles, since they will only be addressed when the Everest people need information from them. As stakeholders, they can log on to the game and pay a visit to the virtual game world, in which case they would meet in the town hall. This will save them travelling time and costs.

E.9.3 Multiplayer interaction

Chatting happens by clicking on player avatars walking around. A chatbox window will pop up, and the conversation can start.
However, if a chat is desired with a player who is not currently in sight, an icon is available, which upon clicking shows a list of all online players. A chat window will open, displaying the avatar's face next to it. If it is a multiple chat session, more avatar faces will be displayed. This way you have an overview of who participates in the conversation.
Players can choose to chat personally or 'broadcast' messages to the whole team. In the latter case, the message will appear in every team member's chatbox. Modelling sessions are done in a collaborative fashion, with communication happening through the chatbox.
Players can work on buildings collaboratively, if they choose to seek help, or if the building is simply too complex to create on one's own. Changes can be made simultaneously, but also in turn-based fashion, depending entirely on the collaborators.
Players work in small teams, even though each member works on his own buildings. Reviewing is a team-based happening, in the town hall where everybody will meet. The lead supervisor will guide the session. Each player can enter his comments in a text window, and send it to the lead supervisor. The latter will check all reviews and send them to the team afterwards.

Players who desire personal guidance can ask for it by posting a message on the message board. These calls for help will be marked as high priority messages, so that a player will never have to wait for a long time until help becomes available. A player who thinks he can help can contact the asking player directly.

E.10 Context

The context of the game is the entire product development process, from getting the order to delivering the final product.
The game has multiple physical locations. The largest part will take place at

Everest BV., Den Bosch. The game will run on an internal server, and internal projects will make use of the game's facilities to complete their application.
Their client will run the game wherever he is located, and log on to Everest's server when he needs to be present in the game environment for meetings, collaboration or simply to check up on the progress.
When the game has been played and the application is complete, the virtual world will make place for a face-to-face informal meeting, where project participants including the client can speak freely of their experiences and results.

E.11 An impression of Aquimatization

Disclaimer:
For the creation of the following game impressions I have used image material from The Settlers 6, Spore, World of Warcraft, Age of Empires and Shadowkey. These impressions are for illustrative means only!

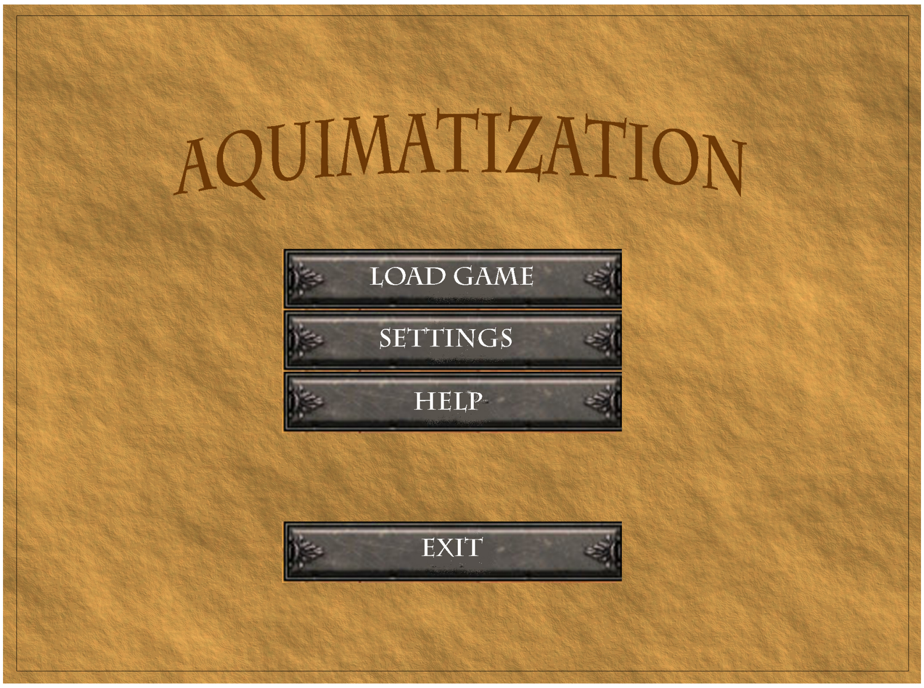
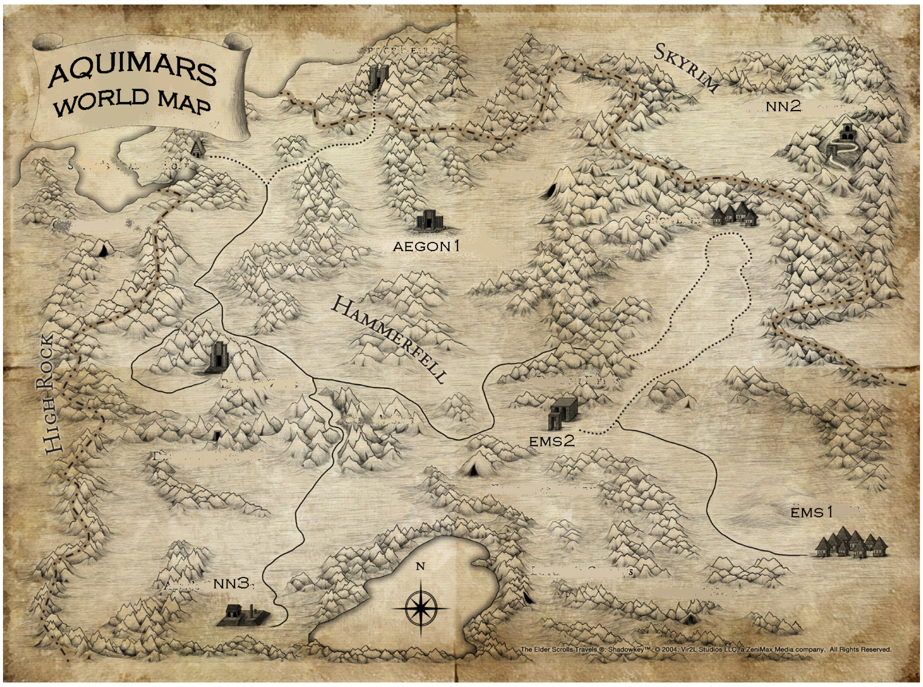


Figure E.2: The game menu appearing at when the game is started. From here the player can choose to change his settings (including a place where he can edit his avatar) or to load the game immediately.



The Elder Scrolls Travels ®: Shadowkey™ © 2004. Vir2L Studios LLC, a ZeniMax Media company. All Rights Reserved.

Figure E.3: After loading the game, the player sees the world map representing all ongoing projects. He can click on a project to visit the site.

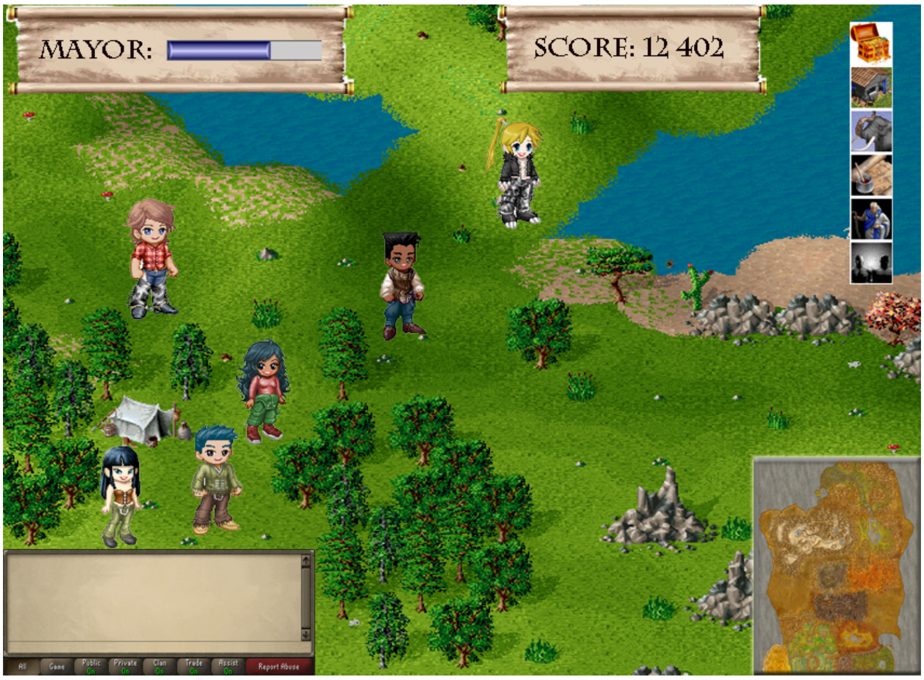
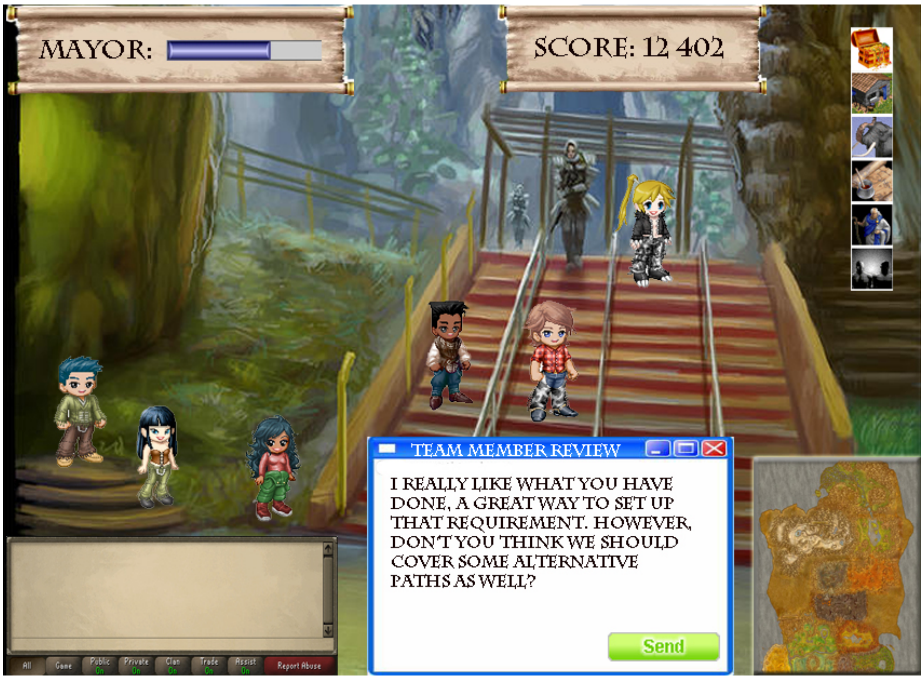
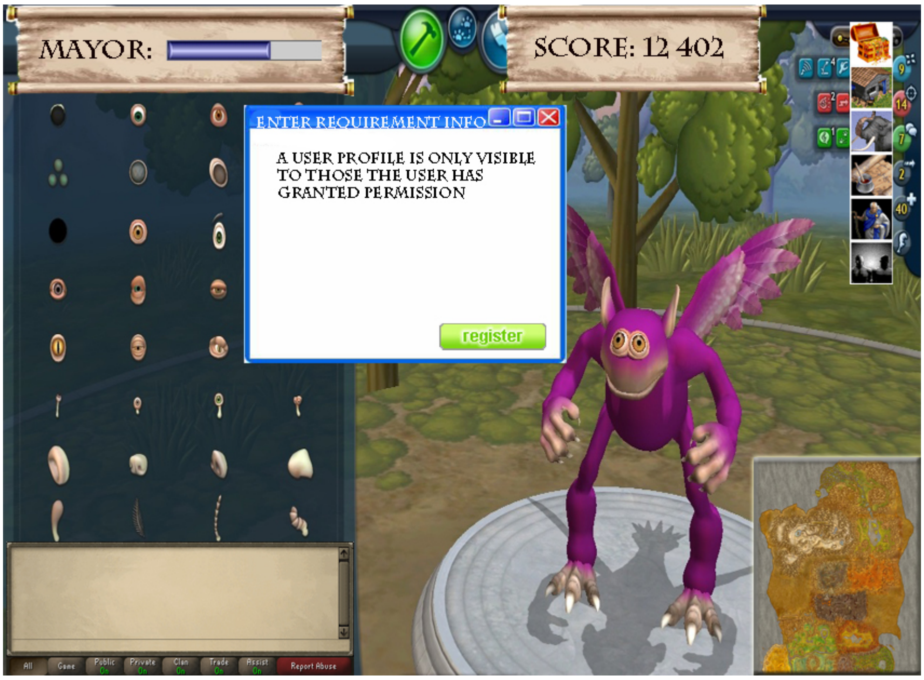


Figure E.4: At project kickoff, there is nothing yet on the site. A simple tent will have to do as a meeting place, from where future plans can be discussed. The tent will evolve into a building as the project progresses.



I REALLY LIKE WHAT YOU HAVE DONE, A GREAT WAY TO SET UP THAT REQUIREMENT. HOWEVER, DON'T YOU THINK WE SHOULD COVER SOME ALTERNATIVE PATHS AS WELL?

Figure E.5: Should a tent not be available, a cave will surely be. Inside the cave, the team members can have meetings, review each other, collaborate in real time and decide upon a project plan.



A USER PROFILE IS ONLY VISIBLE
TO THOSE THE USER HAS
GRANTED PERMISSION

Figure E.6: After the plans have been decided upon, the team members can get started on requirements / minion creation. Adding features adds information to the requirement.

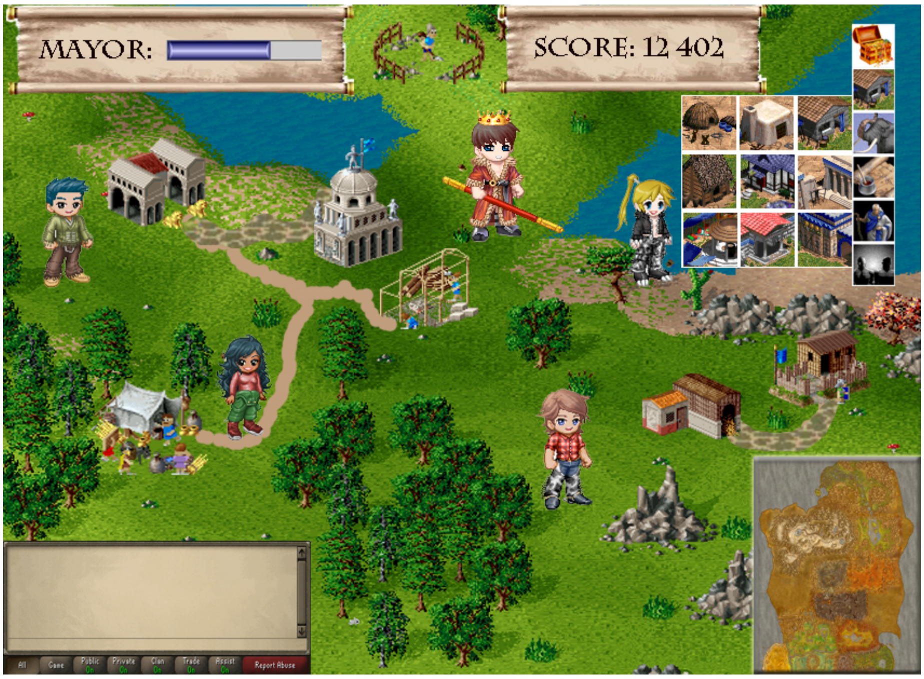


Figure E.7: When sufficient viable requirements have been created, the associated buildings can be set up. This is done by the minions themselves, after they have received their orders from the build supervisor. Build supervisors can choose the type of building to be constructed when they click the 'build' icon. Every now and then, the mayor can choose to visit, to see the progress directly.

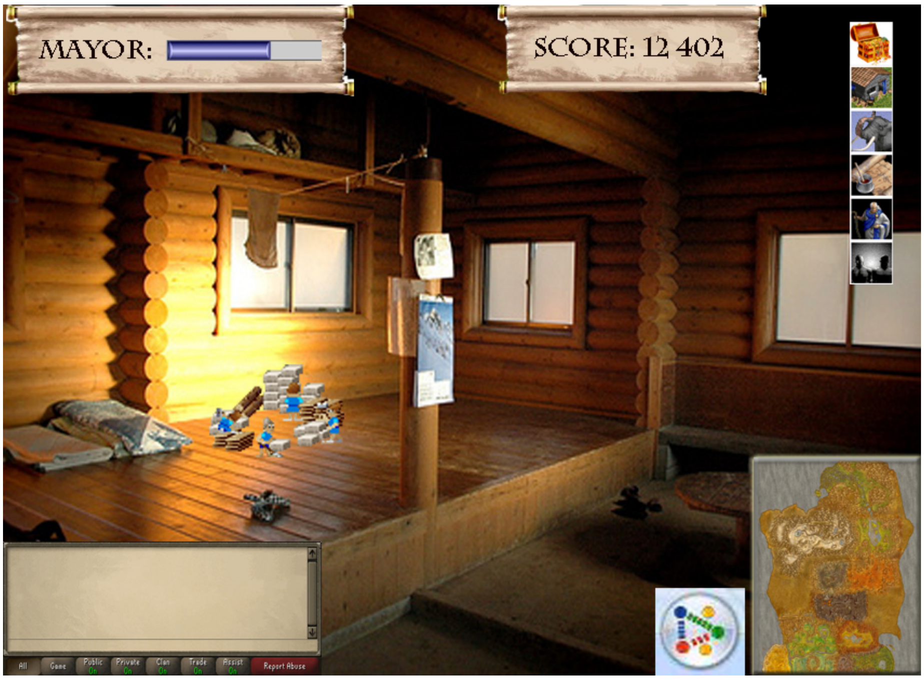


Figure E.8: All buildings can be entered to do more detailed model modification. Access to the graphical editor in Aquima Studio is available from here.

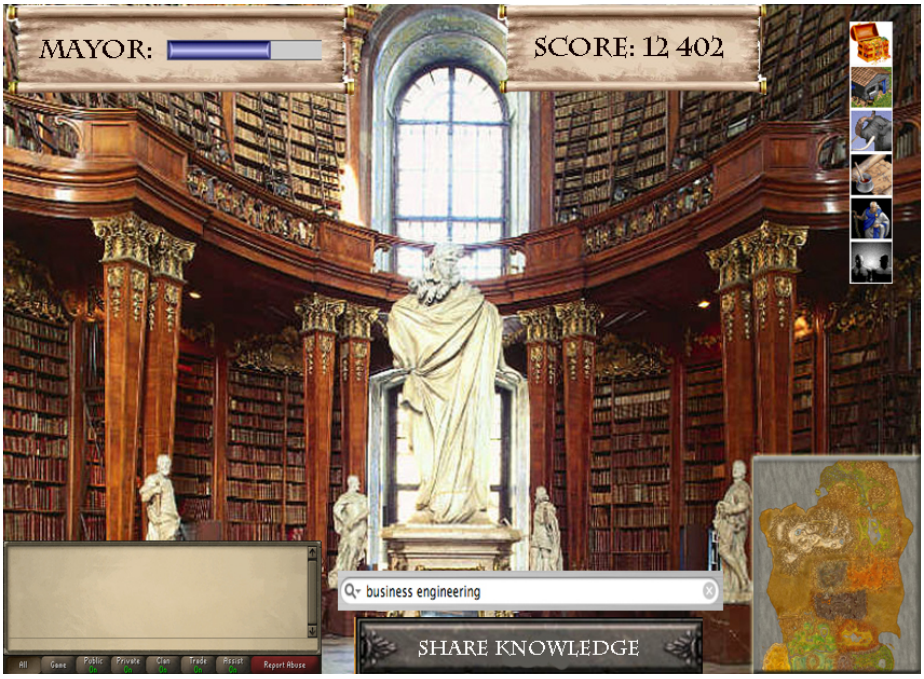
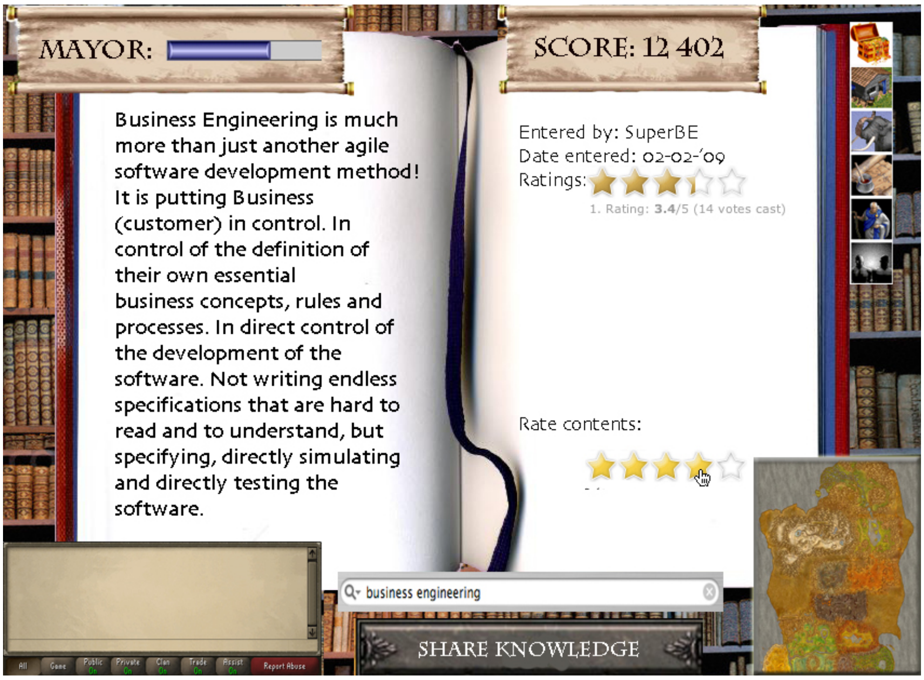


Figure E.9: The project has gone on for some time, and suddenly there are some difficulties. Team members can seek answers to all their questions in the shared knowledge library.



Business Engineering is much more than just another agile software development method! It is putting Business (customer) in control. In control of the definition of their own essential business concepts, rules and processes. In direct control of the development of the software. Not writing endless specifications that are hard to read and to understand, but specifying, directly simulating and directly testing the software.

Entered by: SuperBE
Date entered: 02-02-'09
Ratings:
1. Rating: 3.4/5 (14 votes cast)

Rate contents:

Figure E.10: Searching for desired titles takes the seeker to the relevant section, and from there, the sought book can be clicked. It will open, presenting its contents, which must be rated after use.

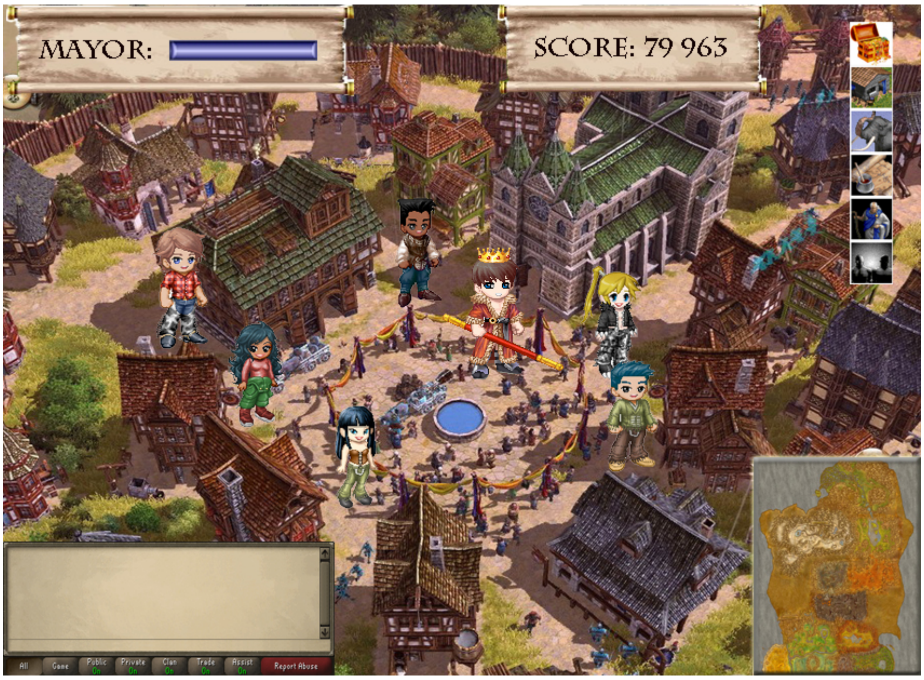


Figure E.11: And finally, after lots of hard labour, the town is finished, the mayor is happy and the final feast can start!

Appendix F

Recommendations for Everest

F.1 Game implementation plan

F.1.1 Phase 1: Core game functionality

The enormous advantage of implementing the necessary changes for Aquimatization, is that the basic functionality underlying Aquima Studio will not change. Core game play actions such as creating entities, attributes, relations, business rules, decision tables etc. will remain the same, only the interface will change. It is possible to build the interface and simply link the existing functionality to it.

The game should be set up incrementally. The first step is to ensure the key mechanisms are functioning appropriately. Basic requirements for collaboration such as the collaborative chat client (still in prototype) and the real-time graphical modelling tool are already available. Furthermore, possibilities for virtual meetings, online storage of requirements, online real-time file sharing and peer reviewing should be created. For this first phase, the interface used can be the conventional Studio interface for real-time modelling and collaborative chatting. Online requirements storage and peer reviewing could be done using Everest's existing Issue Management System (IMS), a forum-like tool used for posting, tracking and discussing certain project issues.
For online file-sharing and virtual meetings, a simple interface with a slide viewer and a chatbox will do.

In terms of the game, the underlying mechanisms for the town hall, the minions and the buildings will be created first. The necessary components for each object are named below.

Town hall

- Chat client
- Real-time graphical modelling tool
- Mouse and keyboard

- Microphone and speakers
  Should real-time vocal interaction be desired.

- Review mechanism
  Possibly a personalised IMS. Reviews go to the project leader first, who can then post the results in the reviewed member's IMS. General project criteria should be visible for all, personal criteria are only visible for the individual.

Minions

- Mouse and keyboard

- Chat client

- Storage for input
  Possibly using the IMS. A separate category for 'Requirements' should exist, in which requirements can be created, described and stored in a categorised fashion. This should be visible for all team members.

- Review mechanism
  Comments should appear in a separate box, visible when the related requirement has been opened. These comments should be visible for all team members.

Buildings

- Mouse and keyboard

- Chat client

- Online model storage
  This can be done in Aquima Studio as it happens now, but visible for all team members.

- Review mechanism
  Comments should be visible when the model has been opened.

With these mechanisms implemented, the game already covers much of the development process: requirements analysis, application building, reviewing and collaboration with the client and the team.
An important point is to keep the interface and the core functionality loosely coupled, so that when the game interface is completed in the second phase, coupling it to the functionality will not pose any significant problems.

Scoring as a challenge reward at this time only comprises ratings given by peers, superiors and the client. The viewing rights for the client have to be defined, and an account for him to access the collaborative facilities and view the immediate results will have to be set up.

F.1.2 Phase 2: Creating the game interface

In this phase the necessary game interface may be built. A suitable game engine and good graphic design will be necessary for this phase. However, the interface can be created separately from the game functionality. After completion of the interface, it can just be linked to the functionality.

Apart from the interfaces for the core functionality, the interface for the initial project phases, such as the world map indicating project sites and in which account managers can enter detailed information about the to be initiated projects. Log-in screens and a client role interface should also be added.

Finally, the detailed scoring algorithm will have to be worked out and implemented.

F.1.3 Phase 3: Knowledge sharing

After successful implementation of the essential gameplay, a next step might be the knowledge sharing project embodied by the library.

An interface will be necessary, a means to add and store information and a review mechanism. The greatest work will be the interface. Storing information on a server is a well-known mechanism, and the review mechanism has already been created in the previous phase.