A Framework for Real-time Collaborative Game Design in Digital Environments

Rakan El-Awwad & Jeppe Tuxen
IT-University of Copenhagen

May 31, 2013

Contents

Introduction

In this thesis we propose a Framework for Collaborative Game Design facilitated by a digital environment that enables a designer to design a game while it is being played by a player. The instantaneous designer creation and simultaneous player interaction with the created content creates a new space for exploration. To show how this framework could be implemented, we have created a digital design tool, which we will discuss and evaluate with experts from the game development industry.

The motivation for this project originated as an attempt to instrumentalize procedural content generation(PCG)[1], in a game design context. The purpose was to examine how such procedures could aid the designer in the design process. We imagined a mixed-initiative approach (Kerssemakers et al., 2012; Smith et al., 2010), where the designer would be able to generate levels or other game content which would instantly be testable by a human player.

What we discovered during our initial research was that there was no existing game design platform that could support this kind of real-time testing. This led us to the conclusion that in order to leverage the computational creativity potential of PCG, as a valuable game design attribute, we would have to develop a tool that could support a real-time design situation, where both the designers and player, would be present, within a digital environment. This thesis is an attempt to frame the design principles of such a tool, by connecting design literature to the practices and processes of game design. The framework we propose does not yet include computational intelligence, but the reason we mention PCG here in the introduction is because it connects to how we imagine a real-time collaborative game design tool to be further developed.

In the first chapter we will frame the grounding theory of design and in particular game design. Furthermore we will look into existing research that shares similarities with our own.

In the second chapter we will present the framework in detail. This chapter will first provide an overview of the framework, followed by a discussion on how it is positioned in relation to game design theories and practices.

[1]The process of generating game content, using algorithms and procedures

In the third chapter we will present our tool Kre8, an implementation of the framework. We will discuss how the different features of the tool relates to the framework, and reflect on our design choices.

The tool was evaluated by four industry experts at the danish game company IO Interactive[2], and the process and result of this will be the focus at the end of chapter three. Lastly we will conclude our research and point towards further inquiry in chapter four and five.

[2] http://www.ioi.dk

1 Existing Game Design Practices

1.1 Design

In this chapter we will first introduce the concept of the design situation, as it is being used throughout this thesis, to make it clear how we see design as a process. After framing design we will outline the design philosophies, on which our framework has been evolved. The purpose is to frame our definition of design, and connect how these philosophies are leveraged in game design practices, and how they influence the problem space of game design.

1.1.1 The Design Situation

Before we can move towards a description of the design philosophies we would like to describe a more general view of what constitutes design. The view on design we present here, has been influential on how we have developed our framework and will be useful to the discussion on what kind of design processes our framework supports.

Lawson draws a picture of the design task, as trying to find a *design fix* to a *multi-dimensional design problem*, defined by internal and external constraints, generated by the different stakeholders, and the context of the design situation (Lawson, 2006). By multi-dimensional Lawson means that most design problems can be divided into a series of sub-problems. What makes a design task difficult is that these are connected in a net of interdependencies and that an ideal solution for one sub-problem, might induce negative influence on another. This makes *fixing* a design problem a very complex task. To circumvent the complexity, and move towards an integrated-solution, the designer needs to work on the sub-problems in parallel (Lawson, 2006).

Shön has described the design process as a conversation between designer and the design situation (Schön, 1983). According to Shön the designer moves towards a design solution through *reflection in action* and *reflection on action*. By experimenting with a temporary solution the designer creates a new situation in which he can extract new knowledge of the design situation. This new knowledge, will provide a basis for new experimentation, which again will lead to new knowledge, thus the design process can be seen as a loop between acting on the design, and observing and thinking on the result of this action, which again will lead to a new action on the design. Shön calls this kind of interaction with the design situation, a *conversation*, between the designer and the design material.

This is backed up by Löwgren and Stolterman who notes that the design process is comparable to a conversation in that "the designer asks question to the situation - through actions..", and then "..listens for replies..", that again will influence the designers further actions (Löwgren and Stolterman, 2007).

In Löwgren and Stoltermans model of the design situation they work with three abstraction layers of the design. These are *Vision, Specification* and *Operative Image*.

The Vision is to be understood as the first idea, of a possible solution, the designer gets when presented with the design proposal. It can be seen as a "first organizing principle", internal to the designers mind.

For Löwgren and Stolterman the early stages of the design task is a process of moving from *Vision to Specification* (Löwgren and Stolterman, 2007). The Vision is an abstract internal image of the design, that has a very elusive nature, and it is the process of concretizing this abstract vision, making it explicit, external and specified in a *Specification* that is the focus of the early stages of the design process.

What is interesting in their model is how the designer moves from Vision to Specification. According to Stolterman and Löwgren the designer creates a temporary image of the design called the *operative image*, which will gradually be transformed into a specification of the final design. By working with the operative image the designer generates new knowledge of the design situation, that can introduce dilemmas between the situation at hand and the original vision. i. e. the designer might need to change the vision if he discovers that the practical constraints of the situation makes it impossible to follow the vision. Because of this the *operative image* functions as a bridge between "the abstract and the elusive vision to the concrete and complex situation." (Löwgren and Stolterman, 2007, p. 19).

For Löwgren and Stolterman the three abstraction layers of the design process, the *Vision*, *Operative Image* and *Specification* is therefore a "fully dynamic dialectical process" (Löwgren and Stolterman, 2007, p. 17), where the designer needs to negotiate compromises between the different layers. They argue that what drives a design forward is "the designers ability to refine the operative image" To produce an *operative image*, the designer can use different design tools, amongst those are sketching and prototyping, which we will return to shortly.

To summarize this section on the concept of design we see that design is the process of trying to find a solution to a complex interdependent multidimensional problem, which require the designer to work simultaneously on multiple aspects of the design problem. To drive the design process forward the designer manipulates a temporary image of a solution, an *Operative Image*. By doing this the designer acquires new knowledge of the design situation which can change how the designer understands the design problem. The design process can be seen as a conversation, where the designer takes an action, and then listens for the result.

Having this view upon design as the process, of moving towards a possible solution, by acting on the design situation through an *operative image*, we can understand the reasons for the practices that we find in game design. But before we get to game design, we would like to visit some general design philosophies influential to our view on design.

1.1.2 User Centered Design

Norman advocates for a "user centered design"(UCD) philosophy, based on "the needs and interest of the users" (Norman, 2002, p. 188), who ultimately should be able to use and understand the designed artifacts. By neglecting the actual use situation of the design and the psychology of the target users, designers run the risk of designing products, that ultimately cannot be understood and used by the intended users.

Unlike methodologies like feature driven development, UCD effectively adds the users to the problem space of design. This implies that the designer will need to acquire knowledge about the user, in order to make informed choices about the design. The design space is not only constrained by what is possible from a technological or practical perspective but also on the users themselves, including their preferences.

For designers, the shift from focus on technology to a focus on users has resulted in a flourish of design and research methodologies on how to empathically understand a given target group, and how to ensure that a designed artifact meets the intended user experience goals. Many of these position the user as a research subject, i.e. the *Think Aloud* method, where a user is asked to verbally express thoughts while interacting with a design (Nørgaard, 2006; Hoonhout, 2008). Other methodologies, like *Participatory Design*, which we will return to in a later paragraph, tries to include the user as co-author in the design process (Muller, 2007; Muller et al., 1994; Lochrie et al., 2011).

The principles of UCD has been adopted by many design domains among those are game design. The game design equivalent of UCD is *Player Centered Design* (PCD).

In PCD it is the players' experiences which is the prime focus of the design task. This is evident in Fullerton's textbook on game design, where she writes that a designer must "be an advocate for the player", and always keep focus on the "player experience" (Fullerton, 2008, p. 2). We will return to how the inclusion of users/players to the problem space of games has resulted in an iterative design process, and how this influences contemporary game design.

1.1.3 Collaborative Design

Collaborative Design covers a design situation where multiple human agents, are forming the design in collaboration. It can be argued that all design situations are in fact collaborative, and several authors have pointed toward the notion of design as a social collaborative process, heavily dependent on the designers ability to communicate and negotiate compromises, between different stakeholders in the design. To draw a few examples of this Löwgren and Stolterman writes: "Design is a social process, which means that communication with other participants is crucial" (Löwgren and Stolterman, 2007, p. 59). Nelson and Stolterman even argues that design is a form of *democracy* (Stolterman and Nelson, 2012, p. 47). For Binder et al. the core of design work is: "..about cooperating with others, and mobilizing one's and other's imaginations" (Binder et al., 2011, p. 16).

The reason that design requires collaboration can be found in the complexity of the previously mentioned multi-dimensional problem space of design. A singular person cannot be an expert in all of the dimensions, and therefore design often requires a multidisciplinary approach to find a solution.

1.1.4 Participatory Design

Participatory design (PD) can be seen as a subset of collaborative design, but is more specific in its requirements. A PD design process always involves the participation of non-designer stakeholders, usually the users of the product. The professional designer has the responsibility of facilitating the design process. This is done by providing methods for collaboration between designers and users, i.e. in the form of activities such as exploratory design games, which we will return to later.

Our framework and resulting tool share some resemblance with PD games. Like PD games our framework is also dependent on player participation. But unlike a PD design situation we do not require the players to actively take part in the design process. What we ask from the players to do is to be players and play games. For this reason our framework can be said to be collaborative instead of participatory.

1.1.5 Playful Design

Playful Design is based on the idea, derived from the work by dutch anthropologist Johan Huizinga that humans are ludic creatures that likes to engage with their surroundings in a playful manner (Huizinga, 1955).

In his essay *Designing for Homo Ludens - Still* Gaver advocates for a design view that stresses playfulness over usability (Gaver, 2008). It can be argued that playful design is in opposition to user centered design, but if we accept that the focus of UCD is the *needs and motivations* of the users, and that these users are playful creatures, we see playful design as a sort user centered design.

The view we have on playful design is derived from the research by Gaver et al. where the interpretation of the design is deliberately made ambiguous, leaving the users to make their own conclusions. In Gavers view, ambiguity "is a property of the interpretative relationship between people and artefacts.", that "require people to participate in making meaning" (Gaver et al., 2003). In this view ambiguity can be seen as a resource for creativity because, it expands the possible space of interpretation, and requires people to make an effort in trying to understand the semantics of the designed artifact.

In this section on design philosophies, we have mentioned four design philosophies that can help us understand the object of design. These are;

User Centered Design the idea that the users are the main focus of the design.

Collaborative Design as a way to circumvent the complexity of "the multidimensional design problem", by introducing a multidisciplinary approach to design.

Participatory Design , where the user is involved in the design process, facilitated by the designer.

Playful Design which accepts the notion that humans are playful creatures, and therefore advocates for playfulness and ambiguity over utility.

All of these design philosophies is relevant to the discussion on why real-time collaborative design is a good idea, and how such a framework should support game design. In the next chapter we will present three views on how to understand games, from a formal or semiformal perspective.

1.2 Game Design Theories

This chapter will look into how game design can be approached from a formal or semiformal point of view. Formalism in games makes it possible to breakdown the inner workings of games into smaller parts that can be analyzed and synthesized into new designs.

1.2.1 Mechanics, Dynamics & Aesthetics

The *Mechanics, Dynamics & Aesthetics* framework(MDA) by Hunicke et al. is a formalized view on games that tries to "bridge the gap between game design and development, game criticism, and technical game research", by providing a model of games that can be used to promote understanding between the designers, researchers and scholars (Hunicke et al., 2004).

The core idea of MDA is that games are behavioral system artifacts designed by designers but consumed by players. The designer creates a game system defined by mechanics and rules. The dynamics layer is the system of mechanics set into motion, and the aesthetic layer is how this system is experienced by the player. The fact that the designer knows how the mechanics and the game system works makes it difficult for him to experience the game as if he was a player, which is why the MDA framework can be used to understand the reason to why the iterative design approach, which we will discus further on, is extensively applied to game design situations.

The MDA framework provides a model of games, that relies on the assumption that the experience of a game (aesthetics) is exclusively dependent on the dynamic behavior and not on the media of the game. This view upon games neglects the power of narratology, the semantics of game content, and the context in which the game is played as well as the players themselves.

1.2.2 Design, Play & Experience

In the *Design, Play and Experienceframework(DPE)*, an extension to MDA by Winn, he argues that play should be seen as a "mediated experience", which is ultimately dependent on the players themselves,"including his or her cognitive, social, cultural, and experiential background that he or she brings to the given play experience." (Winn, 2008). Because players are different Winn stresses the importance of an iterative design process that takes the target audience for the game into account.

What we can use from both MDA and PDE is the different relationship to the game, that the designer and player have. The game can be seen as a way for designers to communicate to the players, through the game system. We also see in PDE, that this kind of communication is dependent on who the players are, and how they interpret the game.

Furthermore we would like to stress that both models depict the communication between designer and player as one-way only, going from the designer through the game system to be interpreted by the player. That means that the designer will have to hardcode all the intended experiences and messages into the game system before letting a player interact with the system. This means that there is no interaction from the designer once the game is in progress.

In our framework we also use the game as a way of facilitating communication between player and designer, only we will have both the player and the designer present in a much more dynamic *conversation*, where both parties will impose influence over the game space, while the game is being played and created.

1.2.3 Game Design Patterns

The *Game Design Patterns* (GDP) by Björk et. al is an attempt to empirically investigate the combinations of mechanics in existing games and transform these into design patterns thus provide a basis for a common language of game research and design (Björk et al., 2003). The authors claim that GDP is a useful tool to support creativity. In theory the patterns should be usable both as a tool for inspiration, ideation, and as a problem solving tool. By synthesizing different patterns the designer should in theory be able to produce novel combinations of game mechanics.

Kuittinen has tried to utilize GDP, by creating a pattern combination software tool. The purpose of the tool is to provide a way for designers to combine patterns and hereby create a digital prototype. The conclusion of Kuittinens research is that the complexity and relatively low abstraction level GDP makes it difficult to leverage in a digital game design situation, and suggest a framework with a higher abstraction level (Kuittinen, 2008).

Kuittinens attempt to utilize GDP exemplifies how difficult it can be to implement formalized game design theories in a digital tool. When working analogue only, we can accept

a certain level of rule bending, that allows us to visualize a design, at a higher abstraction level, without going into low level details. But when entering a digital domain the rules are governed by the digital system, that has no idea how to to bend rules, and therefore interprets them strictly. Because of this, all rules of a digital game has to be defined in detail with no ambiguity, in order to be processed by the digital machine.

It is exactly this problem that Kuittinen has run into when trying to implement the GDP digitally, which gives reason for him to suggest a framework with higher abstraction level.

We have tried to circumvent this problem of computational rigidity in our tool by enabling the designer to communicate through the semantics of visual and audible game content, that can be created live by the designer, but more on this in the section on the framework.

To place our framework and its applicability in the space of game design we will need to understand how games come into being. In the next section we will look at the process of making games, and the phases involved.

1.2.4 Game Design Phases

Before a game is finalized it goes through different phases that end up forming the end product. Each phase has some internal processes that are required to be processed before continuing to the next phase. The phases are often described as *Pre-Production, Production* and *Post-Production*[3].

[3]Some literature position concept development as a separate phase before pre-production

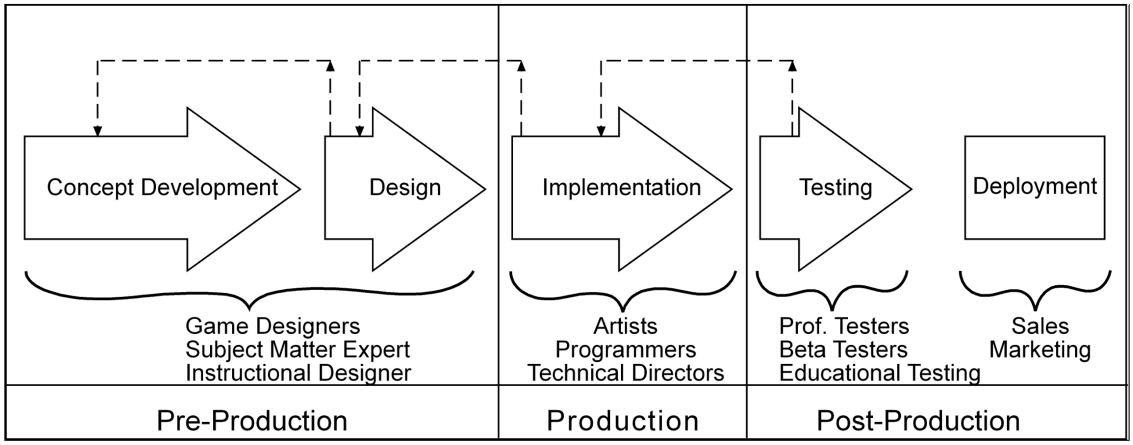

Figure 1: The Game Development Process, E-Games/Purdue University[4]

- Pre-production
  - The concept phase deals with the ideation of a game concept.
  - The design phase is where the idea gets specified into how it should work as a game, in relation to the constraints of the design situation.
- Production
  - The implementation phase is where the specified parts of the game are constructed, this includes all game content such as code, graphics and sounds.
- Post-production
  - The testing phase or quality assurance(QA) is the final step before the game is deployed.

1.2.5 The Iterative Design Loop

In design, iteration is based on a process of prototyping, testing, analyzing and refining the work in progress, thus develop the design incrementally.

According to Fullerton *iterative* means that designers should "design, test, and evaluate..over and over again.... each time improving upon the gameplay... until the player experience meets your criteria" (Fullerton, 2008, p. 15).

[4] `http://www.e-games.tech.purdue.edu/images/Design_2_Big.png`

Figure 2 shows the iterative design process as described by Fullerton. The designer discovers a design problem, generates ideas on how to solve this problem, then formalizes the ideas into a testable version, test the ideas, and from the results acquires new knowledge, that shows either new design problems or no design problems. This process can be seen as a way of working with the *operative image* of the design, we described in the first section.

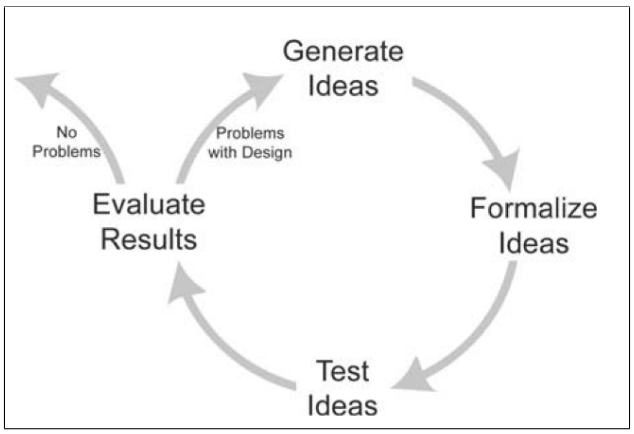

Figure 2: Fullertons illustration of the Iterative Design Process (Fullerton, 2008, p. 15)

The iterative design loop is mostly present in the early phases of the game development process, as can be seen on figure 3.

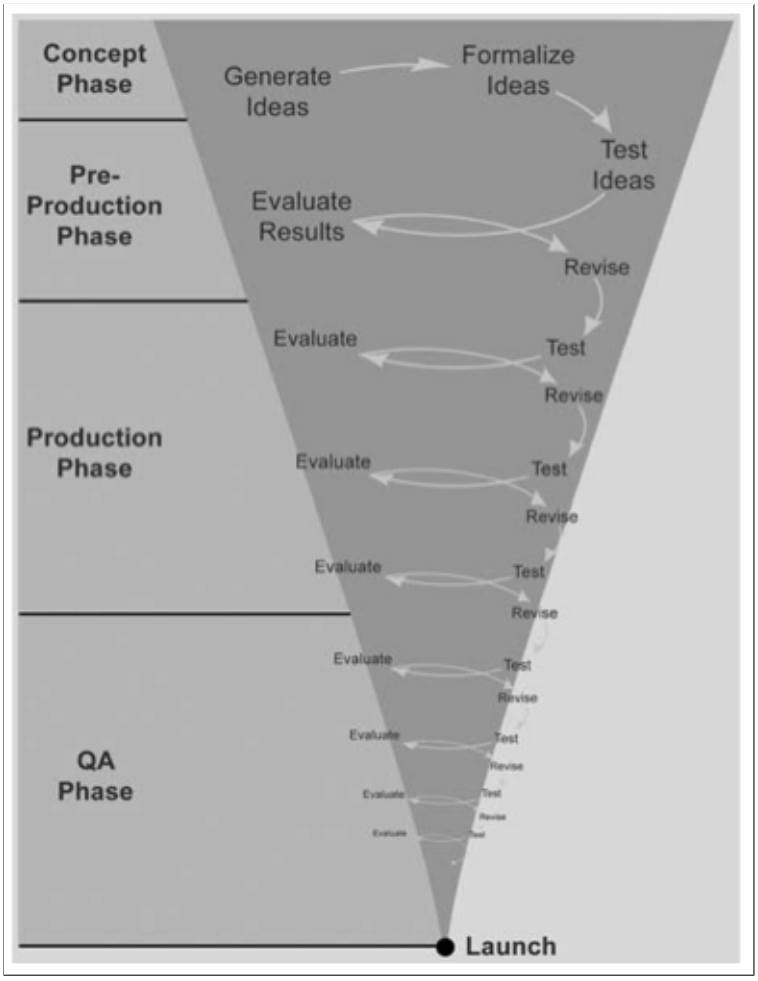

Figure 3: the Iterative Design Process in connection with the Game Development phases (Fullerton, 2008, p. 249)

.

1.3 Design Practices

Because our framework is supposed to be used by designers we will need to understand how they produce their designs, what practices/tools are being used. This chapter will look at the practice of sketching and prototyping, and how they connect with the iterative design process. Furthermore we will mention two explicit methods for prototyping/sketching games.

1.3.1 Sketching

Sketching is the process of quickly externalize design vision, with the purpose of refinement, sharing and ideation. Typically sketches take the form as drawings on paper, but they can

just as easily be made from other materials, just as long as they are cheap to make, and easy to throw out. Of some of the relevant attributes that defines sketching, Bill Buxton mentions that a sketch should be a quick, inexpensive and disposable while affording exploration, rather than confirmation, of a design idea (Buxton, 2010, p. 113).

Another attribute of sketches that should posses is *ambiguity*. We have previously mentioned ambiguity as a an attribute of playful design (Gaver et al., 2003, p. 113). Just as in the case of playful design, ambiguity in sketching provides a space for individual interpretation, and exploration.

Designers can use *design by drawing*, as a way to quickly sketch their ideas, but one of the problems with sketches, especially those made on paper is that "The designer can see from a drawing how the final design will look but, unfortunately, not necessarily how it will work." (Lawson, 2006, p. 27). Because games are ultimately behavioral systems, designers need to understand if a system works as intended. For this purpose designers can create a prototype of the design.

1.3.2 Prototyping

Like a sketch, a prototype is an instantiation of a design concept, but it has a different purpose. Buxton argues that while the process of sketching is mostly utilized in the early phases of the design process, we find the production and evaluation of prototypes concentrated in the later phases (Buxton, 2010, p. 139). This is partly because prototypes are more expensive than sketches, in terms of time used in their creation, but also because the purpose of prototypes is to test and evaluate rather than explore. A prototype is therefore much more specific, than a sketch, and tries to generate answers to design questions rather than raise new ones. We can say that prototypes are better for tweaking design ideas than generating new ones, but because they are more expensive they are also more sparsely represented.

Wizard of Oz The Wizard of Oz methodology is an example of a quick prototyping and testing technique where the user interaction and dynamics of a system can be simulated by acting out the users interactions. Studies have shown that Wizard of Oz techniques can provide valuable design knowledge at an early stage of the design (Höysniemi et al., 2004).

We have included this method to our background section because our framework can facilitates such the use of Wizard of Oz techniques because of the Real-Time manipulation

of game objects, and user feedback.

Both prototyping and sketching exists in the domain of the "operative image", in that they both describe methods for working with a temporary image of the design.

The concepts of sketching and prototyping is important to understanding the purpose of our framework. We try to combine the concept of sketching and prototyping by providing a method for sketching ideas, while simultaneously making a testable digital prototype.

1.3.3 Design Games

Another method of conducting design is via design games. As mentioned in the section on design philosophies, design games are especially used in participatory design situations as means to facilitate communication between designer and non-designers (Brandt and Messeter, 2004). We will mention two design games that has relevance to our framework. Especially in terms of the creative activities they facilitate and how they support communication between participators.

The Silent Game The 'Silent Game', (Brandt, 2006) is a game where the active players are not allowed to talk together as they are playing. The idea is to have the players implicitly understanding each other through their interaction with the design they are working with. The first player will be the one who 'invents' a pattern, where the second player interprets and expands on the pattern, allowing the first player to follow. Having a wide array of game elements and even several of each type is important to make it possible for the players to copy each others moves in regards of defining and interpreting patterns.

The concept of the silent game is relevant because it can be implemented as a practice within our framework, as it involves a certain type of collaboration, it is session based where two players actively interact in Real-Time. Exquisite Corpse Is a game created by the 1920'ies surrealist art movement. In this game players draw on a sheet of paper one at a time in a sequence. The paper is folded in such a way that each player has his own drawing space, while being unable to see the other players space (Johansson and Linde, 2005).

Before the drawing commences, the players are given a topic, word or something to have in mind to inspire the drawing, an example could be somewhere they would like to be.

The first player starts drawing at the top of the paper, and then folds it making the drawing unseen by the other players, on the fold, the player leaves the end of the drawing lines visible for the next player to interpret and draw from, and so on, until all players have drawn, the paper is opened from its fold, and the full drawing is revealed. The combined result is thus an abstract image.[5]

1.3.4 Computer Aided Design

The framework we here describe is in "digital environments", which implies the use of computers in the design process. In this chapter we will outline how we can benefit from computer aided design and where the limitations lie. This knowledge can be used in the discussion of the applicability and scope of our framework.

Bryan Lawson has argued that many designers find the current use of CAD frustrating because it does not enable a bidirectional conversion with the designer (Lawson, 2006). The conversion is on the terms of the computer system, and much knowledge has to be leveraged in order for a designer to meaningfully interact with contemporary CAD programs.

As lawson writes CAD is often used as means to perform neat tricks, that is supposed to make it easier for the designer to perform specific tasks. By being procedurally in their nature computers will inevitably produce further constraints on the design situation which might not be preferable to the designer. We wish to take a different approach to the use of computers in design, and in particular game design.

Because computers have the ability of mass communication through network it is astonishing how little they have been applied as facilitators for communication between participants of a design situation. We would like to examine the possibility space of computers as real-time facilitators of communication between participators of a game design situation.

1.3.5 Similar Research

This research presents a strive towards explaining a framework that implements CAD as a facilitator for Game design. While computer game technology has been used as a means to

[5]Video demonstration of Exquisite Corpse can be seen at `http://www.rakan.dk/kre8/exquisite.html`

afford collaboration between stakeholders of visual and architectural design (Doughty and O'Coill, 2005; O'Coill and Doughty, 2004), it is not often used as a design tool of digital games themselves. In this section we will look at some of the existing research and tools on this topic.

Machinations The Machinations Framework is an example of a computer aided design tools for game design. It uses diagrams to represent the flow of tangible and abstract resources through a game. Dormans describes how these diagrams can be used to simulate and balance games before they are built (Dormans, 2011). The Machinations framework formalizes a particular view on games as rule-based, dynamic systems, and is mostly concerned with structural features of game mechanics that are for a large part seen as responsible for the dynamic gameplay of the game as a whole. This is largely similar to how games are described in the MDA framework, that we have mentioned earlier. There is however no involvement or real players, in the Machinations framework, and while the Machinations framework allows designers to model and simulate certain kind of games in an early stage of development it focuses on a game's internal economy, and does not represent level design or tactical maneuvering, or different player types. We find it more relevant for games, where the internal economy plays a central role, and the object is to tweak this economy through simulation, rather than to generate design ideas.

Game Orchestration Graham et al. has defined Game Orchestration as "the activity of creating experiences for game players at run-time" (Graham et al., 2012). The authors show examples of how game orchestration is being facilitated in existing digital games, by providing the player with *Game Master* abilities (BioWare, 2002), the ability to *Command* other players (Entertainment, 2002) or as a method for Game Sketching, exemplified in the tool Raptor (Graham et al., 2012).

Raptor Raptor is a digital game sketching tool that enables *game orchestration* through the use of a tabletop computer. Designers are able to form a game world by using gestures and mixed-reality interaction involving physical props and digital artifacts. If the designer for an example places a toy car on top of the tabletop computer, the system will be able to detect this and insert a pre-programmed digital car model, complete with physics and interaction behavior into the game world. By placing a game controller on the tabletop computer the designer will be able to connect the outlets of the controller to the interactions

inlets of the digital car. The player will then be able to control this car, using a similar game controller.

The research of raptor is focussed on how the interface of the tabletop computer "supports collaborative game sketching better than a more traditional PC-based tool" (Smith and Graham, 2010). Our project is different in this respect in that we focus on how the real-time collaborative digital environment changes the way we do design. Another difference between raptor and our implementation of our framework (Kre8) is that we do not have any pre-programmed game content. While raptor uses a combinatorial approach that requires a library of pre-programmed game content, we enable the designer to draw objects directly into the game world, making it a more open process.

On the critique of existing CAD tools in game design Neil writes that these tools often are developed by the academic world, from which they never leave, and therefor almost never used by game professionals. She points towards the fact that the tools generally lacks evaluation of their applicability (Neil, 2012).

1.4 Summary

In this chapter we have explained how we understand the design process, as described by relevant literature. Furthermore we have connected this view on design to the practices of game design, exemplified in existing digital and analogue methods. With this in mind we move towards to how we think design should be approached, which will be the topic of the next chapter.

2 A Framework for Real-Time Collaborative Game Design in Digital Environments

In this chapter we will describe the *Framework for Real-Time Collaborative Game Design*. We will show how an environment of digitally connected devices, can support Real-Time creation and communication between designer and player. This chapter will be about defining the framework and comparing it with the view of design we have presented in the previous chapter.

We define A Real-Time Collaborative Game Design situation as a situation in where one or multiple designers can design a game while the game is being played by one or multiple players. Even though there can be multiple designers and multiple players in a Real-Time collaborative game design situation, we will, for the purpose of simplicity, present the framework as if there is one designer and one player, thus the plurative has to be imagined. The designer will have direct control over the game design, through a game editor, while the player will have indirect control, over the creational process, through interaction.

The Real-Time Collaborative Game Design situation is facilitated by a digital environment, supporting Real-Time synchronized update of the game world, so that both the designer and the player can participate in the flow of creation simultaneously.

The design evolves around the following steps:

1. Designer creates game content
2. Player interprets game content
3. Player interacts on game content
4. The designer interprets on interaction
5. Repeat

The program flow of a real-time collaborative game design situation is shown in figure 4.

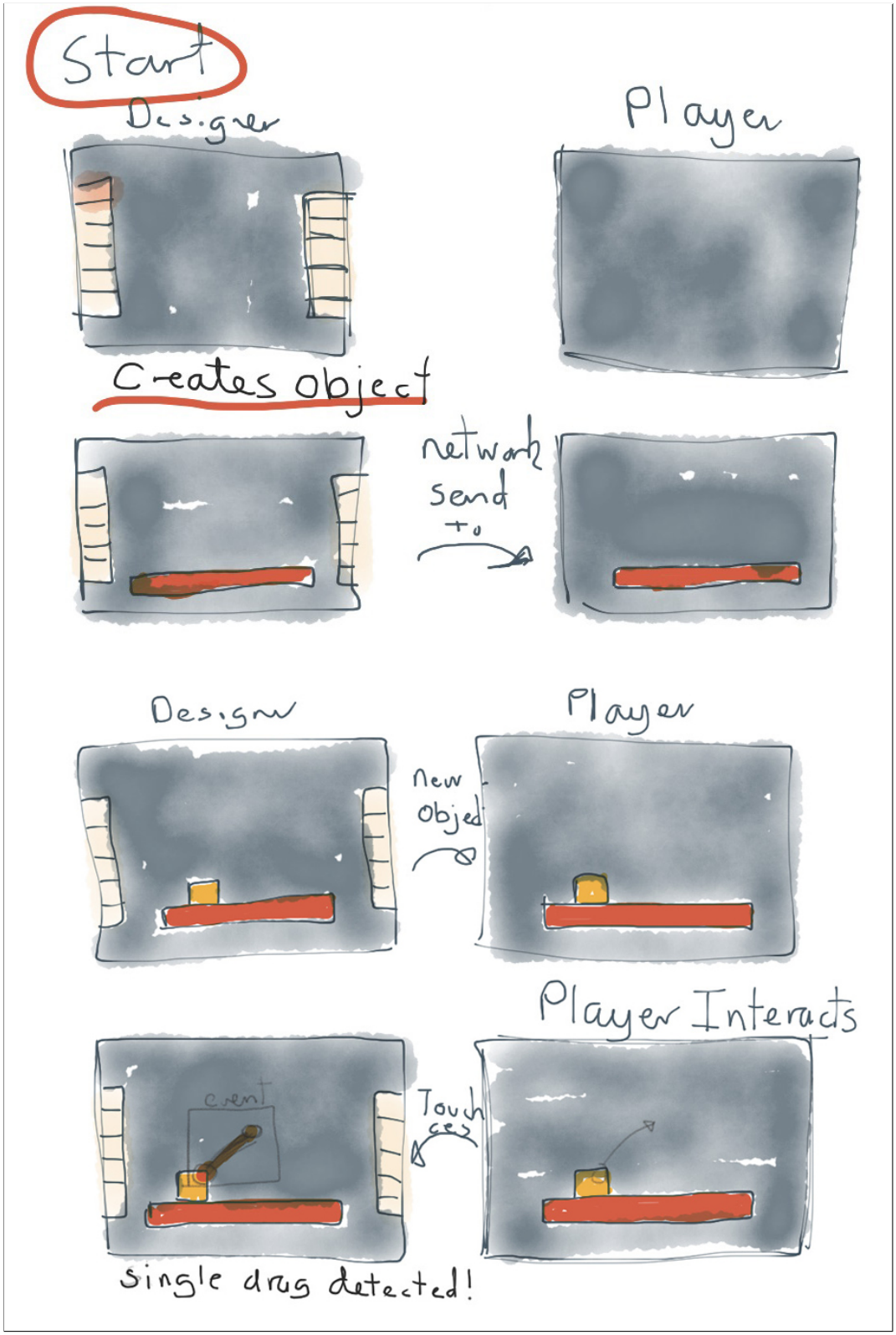

Figure 4: Program flow of a real-time collaborative game design situation

2.1 Design Space

The design space of a collaborative game design situation facilitated by a digital environment is defined by three requirements:

1. It is Real-Time, meaning a synchronized state, supported by a network connection between clients
2. It is Collaborative, meaning multiple users, that can interact in the process
3. It is in a Digital Environment - meaning it is a program run on multiple digital devices

These requirements, coupled with the game design theory and design practices presented in the previous chapter, results in a constrained design space, from which we have extracted a number of design principles.

2.1.1 Real-Time

The Real-Time element in the framework means that all interaction and creation in the process is done live in an always synchronized state. All participants will be able to see changes immediately as they occur.

2.1.2 Collaborative Game Design

The participants in the Real-Time collaborative design situation is at a minimum the end-user, which we henceforth will denote as the player, and the content creator, which we will call the designer. It is important that both the designer and the player roles are represented in the design situation, because it is the conversation between designer and player that drives the design forward. Regardless if the roles can be switched during a session.

The minimum requirements of participants for a collaborative situation will be one designer and one player, but there are no limits to how many participants representing each role. A design situation could consist of 15 designers working on a game while only one player generates feedback, or the opposite situation with 15 players and one designer. The only limit is the network capabilities of the tool facilitating the collaborative design situation.

The minimum requirement is one designer and one player. We do not consider a situation with only one designer a collaborative situation even though one designer can change between

designing and playing. All participants will have access to an instance of the unfolding design running on their own digital device, networked with other participants devices.

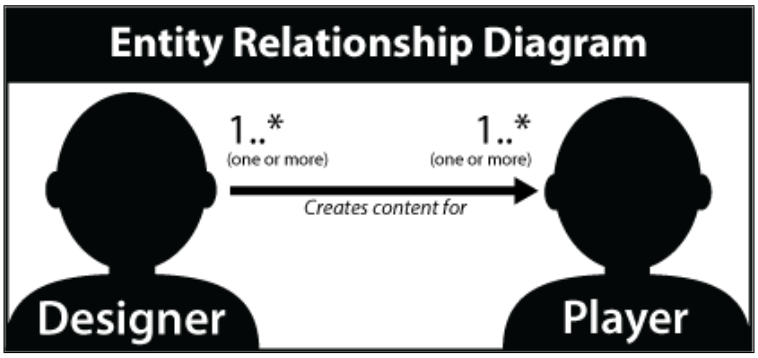

Figure 5: Entity relationship between participants

2.1.3 Roles of Participants

The role of the designer is to create and manipulate a game world, thus giving the player something to play with. This is done within an game editor, with a UI tweaked towards quick and open creation. The designer have access to a number of tool, the designer toolbox, that enables the designer to make changes and additions to the game, thus controlling the player experience. All changes are instantly updated on the players screen, through a network protocol. This simultaneous synchronization of the game state on both the designer and the player device makes the Real-Time component of the frame work.

In most game design tools that facilitate collaboration, changes are created and tested locally before being distributed to a shared server, preferably under some sort of version control. But in a Real-Time collaborative design situation, the design changes, is instantly distributed to all participant, thus affording a simultaneous flow of creation. It should always be visible to all participants what the other participants are doing.

The role of the player is to play around with whatever is created by the designer, thus providing live feedback on how she interprets the design. The feedback is generated through the players attempt to interact with the design. The interaction is redistributed to the designers device, where it will be converted into a visual representation of the interaction. From this representation, the designer will be able to assign game mechanics and interaction possibilities, so that the player gets constrained manipulative access to the world and agency in the game.

2.1.4 Feedback Loop

The simultaneous communication between the designer and the player is at the absolute core of the framework, and is named the "feedback loop". Figure 6, shows a model of the feedback loop, at a high abstraction level.

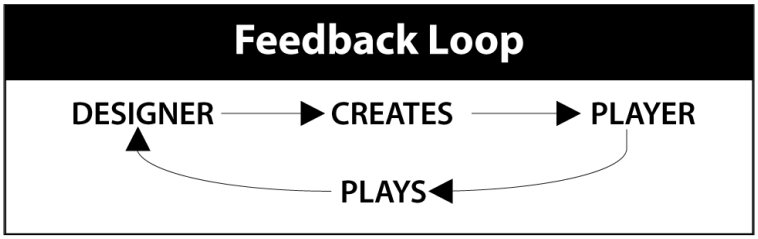

Figure 6: The Feedback loop of a real-time game design environment

The loop is initiated by the designer creating something, in the game world, which is instantly updated to the screen of the other participant, this can then be interpreted by a player who can try to interact with the created content. This interaction results in visual feedback that again can be interpreted by the designer, and thus the constant moving back and forth between designer and player by interaction and reaction, resembles a conversation, where the communication line is the game itself.

By enabling feedback loop, through the game content and player interaction, both the designer and the player has control over the design. The designer will have direct control over the game content, through the designer toolbox, while the player will have indirect control by the generation of feedback through interaction with the game content.

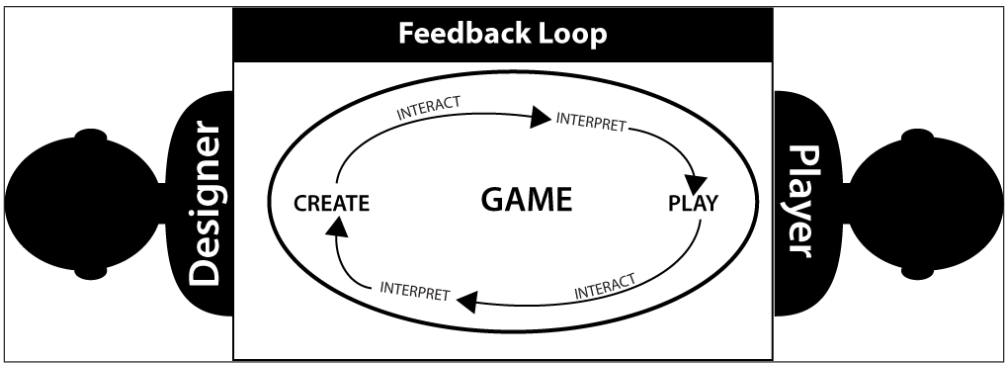

Figure 7: Designer and player communicates through the game

2.1.5 The digital environment

The communication between player and designer, is facilitated by digital devices connected in a network grid with other devices. All devices are running a software game editor on top of a game engine. The digital device automatically updates the state of the game according to rules, defined partly by the game engine, ensuring world coherency, and partly by the designer in the mechanics and events, that has been defined throughout the session. This automatic update of the game state, is possible by implementing the framework in a digital domain and differentiate the framework from analogue design methodologies in which the game state has to be updated manually. This also implies that this type of framework is especially suited for prototyping games in which the game state is updated very often, which is the case for action oriented games, opposed to games that resembles board games. Figure 8 shows the complete framework with one player and one designer:

A tool should be focussed on quick creation opposed to detailed manipulation, to enable the designer to maintain focus on the Real-Time evolvement of the game, and not go into programming details. It should on the other hand also be detailed enough to provide the right amount of openness.

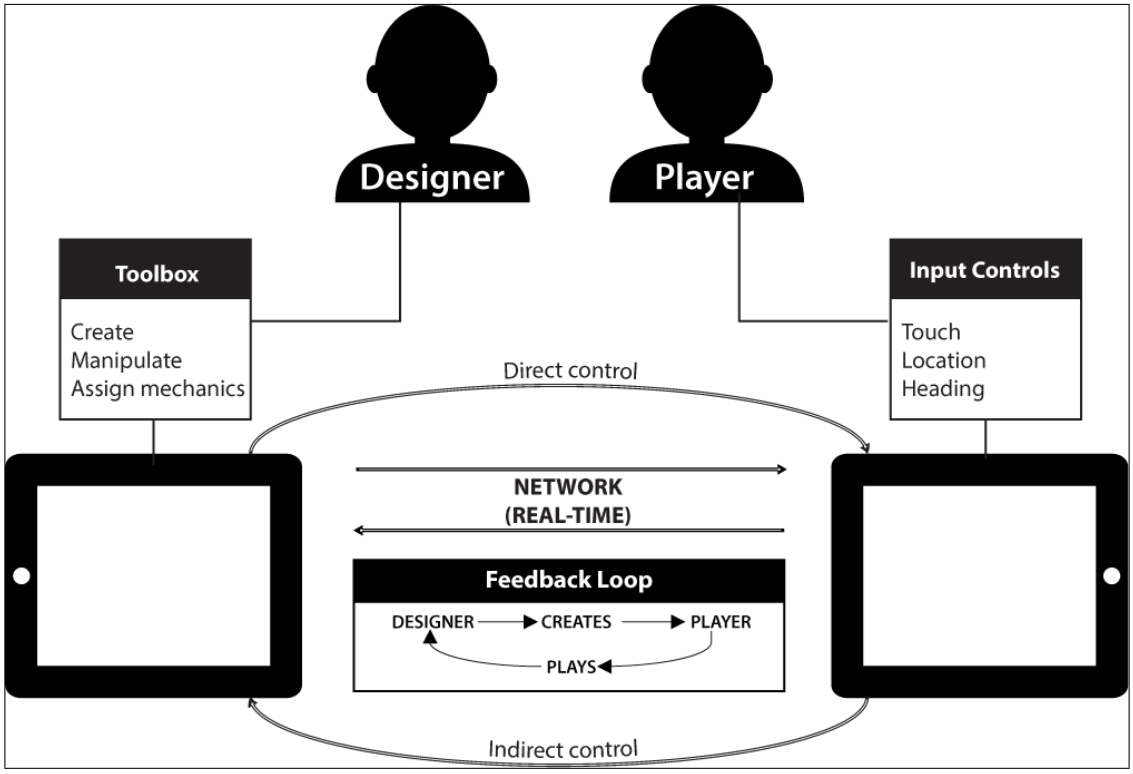

Figure 8: A Framework for real-time collaborative game design in a digital environment

2.2 Discussion

2.2.1 Why Real-Time?

By utilizing Real-Time we are able to maintain a flow where the designer can produce and moderate content freely within a session. In Real-Time the game state is always synchronized to the input from the designer and the player making the designer able to make changes to a game 'on the fly' instead of the usual iterative design approach:

- Design the concept
- Produce a prototype
- Test it
- Elaborate on results (expectations are met, finalize)
- Design changes
- Repeat

In Real-Time, the same steps are involved but the steps are happening at once in a single process, instead of being different processes. We believe this not only optimizes time, but also gives less risk of spending too much effort in changes and rebuilding, calling in for testers again and again, being Real-Time is beneficial on quick testing and improvisation as all processes can be run through in the same session, and the designer is able to instantly see the result of his design actions.

2.2.2 Why Collaborative?

The player can be argued for at the biggest stakeholder when it comes to games. Since they are the end-users, it is essential that they enjoy the game for it to be a success, in regards to success criterias such as sales or active players. This is why we advocate for a collaboration between designer and player in the ideation phase. By collaborating with the intended end-users the opportunity of being able to see what works and what does not, Real-Time as content is being created, gives the designer the possibility to quickly change the direction of the game design by either trashing or tweaking elements of the game to make it more enjoyable for the user. This approach enables the designer to test ideas they might come up with during a session, thus improvising on the design situation.

2.2.3 Why in Digital Environments?

In a digital environment the designer can simulate objects in a variety of ways, where the digital tool will be able to calculate and process the properties of an object, an example could be floating objects, which in a digital environment could be done by setting parameters of particular variables, whereas in an analogue environment it would prove difficult to simulate, as a second demonstration is highly prone to be different from the first since it is up to the designer to control the objects physically.

By going digital, some opportunities for different types of feedback are possible. The feedback loop between the designer and the player, can go through speech, or it can go through interpreting the behavioral input from the player within the game.

When feedback goes through the game, it makes the communication ambiguous, compared to if the player was constantly 'thinking aloud' during the session, because both the player and the designer are forced to speculate on each other's intention. While participants that are located in proximity of each other will have the possibility to communicate verbally, about the design, we have decided not to implement direct voice or text communication as a part of the framework, this is because we want participants to be immersed in the Real-Time process.

We want the designer to create and the player to play. From methodologies like 'Think Aloud' we know that some players have difficulties in maintaining the flow of the play session by having to verbally express their thoughts and experiences while playing. The immersion is continuously lost, and the resulting play is different from a normal play session where the focus of the player is not disturbed by other tasks. Ambiguity also provides a space for improvisation, and thus expands the design space. There is no right or wrong action for the participants, but each have to make decisions that moves the design process forward.

2.3 Design Principles of a Real-Time Collaborative Design Tool

- All participants should have access to their own device.
- The tools available to the designer should map to the designers repertoire.
- The designer should be able to get instant feedback from how the player tries to interact with the game.

- The designer should be able to convert player interactions into mechanics.
- The tool should support rapid creation and manipulation over detailed inspection and tweaking
- The tool should feel like a sketching tool but produce a playing prototype
- The tool should afford exploration through ambiguity.
- The object of the tool is as a generator of design ideas rather than as a testbed.
- The player should be able to interact with the tool as if he was playing a game

3 A Digital Collaborative Game Design Tool

For this thesis we have developed a digital game design tool which we have called Kre8. The tool is meant to show the implementation of the principles behind our framework. Video demonstration of the tool is available at `http://rakan.dk/kre8`.

The focus of this chapter is to explain the features of the tool in connection with the framework described in the previous chapter. Furthermore we have evaluated the tool with industry experts from the danish game company IO Interactive, as to understand how both the tool and the framework could be applied to their everyday workflow and design practises. The findings from this evaluation will be discussed and summarized, at the end of this chapter.

3.1 Kre8

Kre8 is a game editor application running on touch devices, in where one or more designers are able to create and prototype a game experience in Real-Time in collaboration with a player. The domain of Kre8 is primarily 2d - physics based games, but the tool can also be used to sketching out new mechanics and UI interaction elements.

3.1.1 Workflow

The player hosts a session which can be joined by designers. Designers can create game objects, by using conventional drawing tools designed to replicate traditional pen and paper methodologies. Once the designer has finished drawing an object and confirming it (by a check-sign icon) it is instantly an in-game object complete with physical properties and assigned to the object pool from which it can be selected and duplicated. The new object is available for manipulation by all collaborators with access to the designer toolbox. The player can try to interact with the objects in the game, resulting in feedback available to the designer, who can use this interaction feedback to create mechanics for the player. Once a mechanic is created the player will have agency in the game world, and can freely play around with the interaction possibilities. This again provides feedback that further drives the designers design decisions.

3.1.2 The designer toolbox

The designer available a set of tools that can be categorized into 5 different areas:

- Object creation - A drawing tool with an outline layer and color layer
- Object manipulation - moving, edit properties, create object animations
- Player - assign mechanics, control player view
- Events - make rules for dynamic events like collision.
- World - Editing properties, physics

The tools have been designed to be as tangible as possible while still providing a certain level of detail. We have balanced the tools to enable quick creation, and manipulation over the ability to go into details with the design. The main argument is that the designer should always be present in the Real-Time flow of the experience, instead of using time on tweaking a specific parameter. One of the key element in the framework is, that it is the player generated feedback that should drive the design choices, and therefore the designer has to be available to interpret the players interactions when they happen, and make design choices on the fly.

Because we want to create a general game design platform, it is important that the possible content space is as big and unconstrained as possible, so that we do not enforce a specific kind of game genre. For this purpose we have have tried to make the content space as unconstrained as possible by letting the designer draw the game objects instead of choosing between readymade objects.

In the theory chapter we mentioned how designers use "design by drawing" as one of the key tools to externalize their ideas. It is exactly for this reason we have chosen it to be the main method for creating new objects. By taking an already acknowledged and well documented design practice and implementing it in a collaborative tool, we hope that the designer can learn the tool with little effort.

In the description of the framework we mentioned that the designer communicates to the player through the game content. The fact that anything can be drawn, or event written provides a semantic link between designer and player. By searching through a library of pre-fabs the designer would have to leave the session. But by making the designer draw the

objects Real-Time, even if they are of a lower resolution, than pre made objects, we assure his presence in the game.

On the audio side, the choice to focus on quick creation means that we have chosen that the designer should record his own vocal imitation of a sound instead of searching through a library of sounds, which again would require the designer to temporarily lose focus on the unfolding game. I.e. the designer can imitate an explosion sound and assign it to a collision event, instead of going through a library of sounds. Again the quality of the sounds created might be considered lower, of course depending of the vocal abilities of the particular designer, but the quick creation, freedom of expression, and the semantic possibilities embedded in this workflow, makes it better suited for Real-Time sketching purposes.

3.1.3 The feedback pane

The feedback pane gives the designer control over what feedback is being visualized from the player. The tool currently supports two kinds of player input being touch and heading(compass). The data from the headings feedback can be directed towards any object, just by dragging from the compass icon to the object. This tangible way of assigning player interactions to objects, is an example of how we wish the tool to be very easy to use, without the necessity to program the game logic, however on more advanced mechanics and interactions, it is simply not enough to map player interactions data directly to game objects, like in the case of the headings. Which is why we have created two ways of assigning mechanics to the player.

3.1.4 Assigning mechanics

There are two ways of assigning mechanics to players, either by dragging directly from a feedback visualization or by manually choosing the mechanic under the mechanics icon in the player menu, the later type of mechanics are hard coded, but parameterizable while the first are direct mapping of player input to object parameters. That we have chosen both ways is due to the complexity that some mechanics require. It would be very difficult if not impossible to create a regular 2d platformer player control scheme just by mapping player interactions directly to object parameters. Thus we have implemented an easier interface for implementing complex mechanics, but still maintained the direct mapping feature because of its more expressive and open nature possible.

3.1.5 Changing roles

In Kre8, participants will automatically be assigned their roles when they join or create a session, typically the creator will be assigned to the Player role, where the following participants who join the session will be assigned the role of designer. As the tool is not solely focused on creating experiences for a single player (on ingame focus, not sketching phase) but as many players imaginable, the possibility to switch between roles are provided. This is done by pressing the icon for designer. This is to prevent the tool from limiting the type of games that can be created as multiplayer opens up more possibilities for interesting types of games to be prototyped.

3.1.6 Relationship between roles (participants)

A session with Kre8 can be held in different variants, the theory in the early chapter gives examples of ways that the participants can use the tool. Each role should at least have one member assigned to it.

3.1.7 Use Situations

Kre8 tool is meant to encourage creativity, where it is not meant to be used in just one way, but allowing different design games to be applied, making sure it does not follow a strict usage style.

One way to use the tool to create games would be in the manner of each participant staying within their roles in a session where the designer creates content and the player plays it. If we imagine that the participants are in the same room, there are different ways uncover creative ideas and play on ambiguity, facilitated by the tool.

When the designer and player use Kre8, we explained earlier how there is different feedback available by either interpreting the input of the player, and the player interpreting on the content generated within the game screen. Another type of feedback can be generated if the participants decide to discuss orally while going through a session. This variant would be considered to be the general way to use Kre8. In our vision of different usage proposals of Kre8, some suggestions to work by implementing ambiguity and creativity to be a dominant factor, by applying play methods, such as applying design games..

The Silent game applied to Kre8 With the silent game applied it will be similar to as if the participants were not in each others presence, making it impossible to communicate outside the tool. As mentioned earlier, the only feedback to be gathered and act upon, will be what happens on screen. The designer creates objects the player interprets and interact with and while the player is interacting the designer interprets the actions by the player and generates content to form a play experience. This loop continues until the session ends.

The interesting aspect about this is that the designer may have an intention with the content they create, but not being able to be entirely sure about what the player will do with it, forcing him to improvise if the player acts in an unplanned manner.

The ambiguity in this communication affords creativity as it will not be easy to successfully predicting how the other participant may understand and use the design being made.

The Exquisite Corpse applied to Kre8 The rules "The Exquisite Corpse" could be applied but modified a bit to fit the digital environment in terms of the changing roles between who is designing and who is playing, but instead of waiting for each "section" of the paper to be filled with "content", one could have a timing of a fixed period of time each participant will have to generate and play with the content. The changing roles principle is the important aspect of applying this game.

As each participant will have their own device, they are always able to see what goes on while the current designer is drawing, one could compensate by not talking, so in fact adding the silent game to make the situation ambiguous like the analogue version.

This allows for exciting sessions where the designer would have a good chance to learn how much the player understands the ideas the designer is trying to sketch out, as the player will have time to build on to the current project when the designers time runs out and they have to change between roles.

The interpretation of each participants actions while being the designer (in Kre8) will dominate the direction the current designer turns the project to based on their interpretations of the previous actions of the designer.

3.2 Expert Evaluation

To position our ideas in a broader context around the activities in game design, we presented the tool to four industry professionals at the danish game studio IO interactive.

IO interactive is an established company, perhaps best known for their Hitman series, that includes five AAA releases of which the first "Hitman: Codename 47" (Interactive, 2000) was released in 2000 while their latest game "Hitman Absolution" (Interactive, 2012) was released in november 2012.

Even though the kind of visual impressive 3D game experiences that are produced by IO interactive is very far from the ones that can be created with our sketching tool we wanted to get some insights into how they, as industry experts would interpret the applicability of not only the tool itself, but also the principles behind the tool, and discuss the potential of incorporating the Real-Time collaborative game design element into their everyday workflow and design practices.

Of the four people we interviewed, three was from IO's research and development department "Incubation Lab", in charge of the incubation of new game concepts focussing on "core" mechanics opposed to visual design.

They were assigned the roles of:

- Producer (Hakan)
- Engine tech (Jesper)
- Game design (Jonas)

Aside the different roles, the three members were all actively involved in day-to-day game design processes. Furthermore, they all had some degree of programming skills. This means that they were capable of discussing the tool both from a designer's perspective, but also on a more technical level, which spawned some interesting discussions about the scalability of our Real-Time concept.

The fourth person (Viola) represented the user testing and research department of IO, with the focus on how the intended users interact with the designs, and if the player experience goals are met.

The meeting was held at at one of their OI departments within the IO headquarters called the "Incubation Lab". The Incubation Lab is where the crew at IO explore ideas and produce products from small projects that run over a span of around two weeks. After an introduction to their department, and the ideas behind the Incubation Lab, we presented our tool on three network-connected iPads for them to let them play around with. This experience spawned an open discussion about the applicability of the tool, and the idea of Real-Time collaborative game design, which we will use in this chapter as a base for further discussion about Kre8 and the framework. It was our intention to create a casual vibe and let them express freely what they thought of the tool to keep the discussion as open as possible. Instead of scripting a pre-made question sheet we hoped that this more casual approach would lead them to express their understanding of the tool.

3.2.1 Results

Here is an overview of the feedback we got, we will go into detail with all the findings in the next paragraph:

Hakan (Producer):

- The tool could be a fun social experience, a "drinking game"
- IO is out of the target group in regards to the tool
- Scalability of the Real-Time component could be problematic
- Real-Time feedback of users actions could be very usable

Jesper (engine tech, designer):

- Would rather have prefabs than sketching
- The tool is limited to 2D physics based games therefore not implementable in their workflow.
- The Real-Time ideas is intriguing
- Real-Time user feedback a huge plus

Jonas (Design):

- If the scope is sketching the tool works great
- The ability to quickly animate objects by tangible manipulation is great
- The tool should inspire playfulness over seriousness, even though it might seem a bad design choice.
- Being Real-Time designers can simulate AI.

Viola (User Testing)

- Could be a potential tool for learning about users, by letting them play with the tool
- Would like to log interactions, for later analysis
- The concept could be transferred to a UI creation tool, with simultaneously user feedback

3.2.2 Discussion

In the following, we will discuss the results of the interview. The producer was very keen on knowing who the intended users of the tool was, and saw the tool as a great "drinking" game opposed to a serious tool for game design professionals. He also pointed to the fact that the kind of games being produced at IO had very little resemblance with the "games" that our tool would facilitate, positioning him and his colleagues outside the target group of the tool as a professional asset.

On the other hand he was very interested in our thoughts on the scalability of the tool in which he saw some problems with the Real-Time component in a professional production environment, especially when it comes to prioritising simultaneous editing of the same game element, in regards to ownership of changes, and how to control the process so that it would be possible to go back if the game was changed in an unfortunate way.

What is interesting here is that he naturally evaluated the tool as if it was designed as a production tool, where securing data is very important, in order not to lose important work. The idea of a Real-Time production suite in where everything can be edited in the game world while the game is being played, by its intended users, is indeed intriguing, and a revelation that our tool inspired such thought.

However, because of the importance data in a production environment the realization of such a tool would require a deeper technical requirements analysis of the network protocol facilitating the real-time collaboration without losing data, which is completely out of the scope of this thesis, but may be relevant for further inquiry. We have deliberately avoided this complication of real-time collaboration in digital environments by working exclusively on a design-sketching tool, where dataloss is of less importance than in a production tool.

The tech designer also had some difficulties seeing the tool as something he would use in his workflow. He missed the ability to import "prefabs", which is ready made parameterized

game content that can be further tweaked to fit the intended game play. A feature he really appreciated in our tool was the ability to change game parameters, while the game was being played.

Kre8 was generally evaluated in two ways; as a game, where the social aspects of the tool was key to the ludic experience, and as a serious tool for game design. The discussion seemed to wander between these two notions of the usability of Kre8. I.e. the producer instantly saw a potential in the tool as a sort of "drinking game" or fun experience between friends, and asked when we planned to release it.

While creating the tool we might have underestimated the general skill level of industry people, in using production tools like Unity as a resource for design, to quickly make prototypes of game ideas, however, our main intention has not been focused on having a tool which would be able to make finished games, rather having it being able to sketch ideas and encourage exploration and test different mechanics in a quick way without having too many departments within the production team having to send requests back and forth to each other in relation to editing game variables and elements.

Jesper the designer with most technical focus saw a huge advantage in such approach, as compared to the delay that often occurs in the communication between different departments of the game production workflow, exemplified in a typical situation from his own working life, where a variable in the jump properties is worked on. In order to test if the change is indeed an improvement in the game the designer will distribute his build to the testing team who will conduct a user test resulting in a report on the feasibility of the change.

This report will have to be read and understood by the designer who will then make further changes and redistribute a new build, starting the loop all over again. As can be imagined a workflow as described here is both tedious and time-consuming, which is one of the main reasons our principle of Real-Time modification of variables was a considered as great asset and a perfect way to work when letting the people who test the functions being able to actively modify them without the need of consulting other departments.

The focus on applicability of our tool in IO's work ethics was not our aim of displaying the prototype, it was rather to challenge the concept against experienced people within

game development and gather criticism and compliments to focus on the limitations and possibilities of our thoughts behind this new way of working within the field of games.

The user research expert saw a potential in Kre8 as a tool for better understanding a target group, where the collaborative Real-Time features would facilitate a play session where test subjects would use Kre8 on each of their device giving immediate feedback to the researchers. She suggested a log feature, enabling the researchers to replay the session for later analysis of the players. She imagined a similar tool, which would be targeted towards the design of user interfaces.

Such a tool would contain the same Real-Time feedback system from the users, but have predefined UI elements that could be dragged into the screen of the user. We would like to elaborate and extend upon her idea. It is of the utmost importance that the player experience a result of interacting with the UI, in order to fully leverage the Real-Time capabilities of the framework.

In preliminary research, we imagine a combined version of the tool, in where UI designers can quickly create a very simple game experience, as a suitable testbed for their UI design. From here, they could have their testers using the UI to interact with the game, and try variations of the placement of UI elements. In situations where a UI is being designed for an existing game prototype, the system should be able to plug into the different actions available in the prototype. At the core of our framework is the ability for the player to interact a game, thus producing interpretable feedback to the designers.

The interaction has to have influence on the game world, and therefore the game world has to be dynamic. In a UI test without this influence (agency) through interaction from the player side, there would be little incentive for the player to actually interact with the UI and verbal feedback would be of more value than the Real-Time feedback systems in our framework.

It is important to note that the version of Kre8 that was presented to IO was based solely on touch as the main input method available to the players. This singular focus on touch might have influenced how the industry experts understood the applicability of our tool, and thus to place the tool in a broader perspective we would like to extend input features of

the tool to include all available input methods that can be found on mobile devices, which includes gyroscope, accelerometer, location etc.

The most positive feedback on the tool was given by the lesser technical focused designer. In his view the tool, was very useful, especially if the scope was limited to sketching out quick game experiences. The tactility of the tool, and the ability to record object animations, by moving objects manually seemed very interesting to him. He added that if it would be possible to record player mechanics in a similar tactile manner, he would buy our product without hesitation. In general he saw the tool more as it was originally intended.

While the Real-Time elements was on his plus list the tactility was just as important to him. He advocated for even more playfulness i.e. by making particle effects available as an action to an event, even though it did not have any influence to the game itself other than adding to the feel of play.

3.3 Summary

From the discussion, we can see that our prototype is not directly implementable in the workflow of the industry experts. The reason for this is mainly because the tool is limited to the domain of 2d physic based games which renders the industry experts we interviewed outside of the tools target group. The kind of games that are produced at IO are simply too different from what can be generated on our tool. There was on the other hand a consensus that Kre8 would be a great asset as a quick tangible sketching tool, enabling designers to directly test their ideas on players.

Of the features that was most important was the direct player feedback, and the tangible manipulation and creation of game content. The prototype was also seen as a game experience in itself, and we were encouraged to put weight on the playfulness in the tool. We also found a new potential usage of the Kre8, as a user research tool, designed to help the researcher acquire knowledge about players.

While there were limitations to the applicability of our prototype, it seemed that the concept of collaborative Real-Time game design, enabling designers to create and edit a game on the fly, involving immediate feedback from the player(s), would be very usable in their workflow. There was however a concern about the complexity involved in scaling the concept, so it would fit their needs. Especially if the resulting tool should also support a

production workflow. This implies that our framework is easier to implement if the goal is to create quick sketches than if the goal is production value.

4 Conclusion

Throughout this work we have presented a Framework for Real-Time Collaborative Game Design in Digital Environments. We have theorized on how a design situation within this framework would enable a much more fluid creational process between Designer and Player, than possible with traditional design methodologies.

Furthermore we have implemented the framework in a digital environment, which we have evaluated with experts from the industry of game design, with the intention to understand how they understood the tool and framework to be supportive to their everyday design processes.

From this we have learned that, while the current implementation of the tool, was unable to fulfill their expectations of a game design tool, they were very interested in the idea behind the tool and the framework itself. They saw a potential for using the realtime environment as a way to avoid the time spent with going from the different stages of the iterative design loop. Another potential domain for the tool, that was discovered during our evaluation with the experts, was as a tool for player research, where the players attempt to interact with the tool could be used to empathically understand the players themselves. In general it seemed that the people form the industry experts group, that were all from different design domains within game design, saw a potential for the use of a real-time collaborative design environment, within their respective domain.

5 Future Work

We hope this work presents a beginning towards a new attitude toward how to leverage digitality in game design. There is still, however, much work that can be done to further develop the tool Kre8, in terms of usability and the types games that can be created, and we plan to release the tool when it gets mature enough. By making the tool available for the masses we could it could be used as a testbed for understanding design collaboration in general, by implementing data mining, and analysis protocols.

Another area in which this work could be influential is found in the concept of Games 2.0, the idea that players themselves should be authors of games (Tavares and Roque, 2007). For this purpose a well crafted Real-Time collaborative game design tool, could function as

a social platform, where designers and players alike could engage in a global social process of making and playing games. For as Tavares et al. writes in their paper on Games 2.0 "why not allow the creation acts to happen online at the same time that players are actually experimenting with the game?" (Tavares and Roque, 2007).

As we mentioned in the introduction we originally envisioned a tool that would include PCG as a valuable asset in a design situation. Within the framework we have presented here including PCG algorithms, would indeed enable an even quicker generation of game content, and improve on the flow of the player experience. We also imagine that simple randomized parameterization of game object properties or world properties, could be used as a method for invoking constraints on the design space, thus forcing the designer to be creative, within that constrained space.

References

Thomas Binder, Pelle Ehn, Giorgio De Michelis, and Giulio Jacucci. *Design Things*. MIT Press, September 2011.

BioWare. *Neverwinter Nights*. Infogrames/Atari, 2002.

Staffan Björk, Sus Lundgren, and Jussi Holopainen. Game Design Patterns. 2003.

Eva Brandt. Designing exploratory design games: a framework for participation in participatory design? *Proceedings of the ninth conference on Participatory design: Expanding boundaries in design-Volume 1*, pages 57–66, 2006.

Eva Brandt and Jörn Messeter. Facilitating collaboration through design games. In *the eighth conference*, page 121, New York, New York, USA, 2004. ACM Press.

B Buxton. *Sketching User Experiences: Getting the Design Right and the Right Design: Getting the Design Right and the Right Design*. Interactive Technologies. Elsevier Science, 2010.

J Dormans. Simulating mechanics to study emergence in games. *Workshop on Artificial Intelligence in the Game Design ...*, 2011.

M A Doughty and Carl O'Coill. Computer game technology, collaborative software environments and participatory design. 2005.

Unknown Worlds Entertainment. *Natural Selection*. 2002.

Tracy Fullerton. *Game Design Workshop, Second Edition: A Playcentric Approach to Creating Innovative Games (Gama Network Series)*. Morgan Kaufmann, 2 edition, February 2008.

Bill Gaver. DESIGNING FOR HOMO LUDENS, STILL1. 2008.

William W Gaver, Jacob Beaver, and Steve Benford. Ambiguity as a resource for design. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 233–240, 2003.

T C Graham, Q Bellay, I Schumann, and A Sepasi. Toward game orchestration: tangible manipulation of in-game experiences. *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, pages 187–188, 2012.

HCM Hoonhout. *Let the game tester do the talking: think aloud and interviewing to learn about the game experience*. Advancing the Player Experience. Game Usability: Advice from the Experts for Advancing . . . , 2008.

Johanna Höysniemi, Perttu Hämäläinen, and Laura Turkki. Wizard of Oz prototyping of computer vision based action games for children. pages 27–34, 2004.

J Huizinga. *Homo ludens: a study of the play-element in culture*. Beacon paperbacks. Beacon Press, 1955.

R Hunicke, M LeBlanc, and R Zubek. MDA: A formal approach to game design and game research. *. . . AAAI Workshop on Challenges in Game . . .* , 2004.

IO Interactive. *Hitman: Codename 47*. Eidos Interactive, 2000.

IO Interactive. *Hitman: Absolution*. Square Enix, 2012.

Martin Johansson and Per Linde. Playful Collaborative Exploration: New Research Practice in Participatory Design. *Journal of Research Practice*, 1(1):Article M5, 2005.

M Kerssemakers, J Tuxen, J Togelius, and G N Yannakakis. A procedural procedural level generator generator. *Proceedings of Computational Intelligence and Games (CIG), 2012 IEEE Conference*, pages 335–341, 2012.

Jussi Kuittinen. Computer-Aided Game Design, January 2008. URL `https://jyx.jyu.fi/dspace/bitstream/handle/123456789/18458/URN_NBN_fi_jyu-200803181271.pdf`.

B Lawson. How Designers Think: The Design Process Demystified - Bryan Lawson. 2006.

Mark Ian Lochrie, Paul Coulton, and Andrew Wilson. Participatory Game Design to Engage a Digitally Excluded Community. 2011.

J. Löwgren and E. Stolterman. *Thoughtful Interaction Design*. University Press Group Limited, 2007.

Michael J Muller, Daniel M Wildman, and Ellen A White. Participatory design through games and other group exercises. In *Conference companion*, pages 411–412, New York, New York, USA, 1994. ACM Press.

MJ Muller. *Participatory design: The third space in HCI (revised).* the concepts, methods and practices of experience-based design. Handbook of HCI 2nd Edition. Mahway NJ USA: . . . , 2007.

Kathatine Neil. Game design tools:Time to evaluate. *Proceedings of DiGRA Nordic 2012 Conference: Local and Global – Games in Culture and Society*, 2012.

M Nørgaard. What do usability evaluators do in practice?: an explorative study of think-aloud testing. In *Proceedings of the 6th conference on . . .*, 2006.

D A Norman. *The design of everyday things.* Basic Books (AZ), 2002.

Carl O'Coill and Mark Doughty. Computer game technology as a tool for participatory design. 2004.

Donal A Schön. *The Reflective Practitioner.* How Professionals Think in Action. Basic Books, 1983.

G Smith, J Whitehead, and M. Mateas. Tanagra: A mixed-initiative level design tool. *Proceedings of the Fifth International Conference on the Foundations of Digital Games*, pages 209–216, 2010.

J David Smith and T C Graham. Raptor: sketching games with a tabletop computer. *Proceedings of the International Academic Conference on the Future of Game Design and Technology*, pages 191–198, 2010.

E Stolterman and H G Nelson. *Design Way.* University Press Group Limited, 2012.

José Pedro Tavares and Licínio Roque. Games 2.0: Participatory Game Creation. *Proceedings of the 6th Symposium on Computer Games and Digital Entertainment*, 2007.

B.M. Winn. The design, play, and experience framework. *Handbook of research on effective electronic gaming in education*, 3:1010–1024, 2008.