

The Qbe

Andreas Bjerkeholt
MS Student Interaction Design
+46(0)702-649469
harteex@gmail.com

Christian Broniewicz
MS Student Interaction Design
+46(0)734-028032
c.broniewicz@gmail.com

Disa Faith
MS Student Interaction Design
+46(0)704-398622
disafaith@gmail.com

Sophie Kores
MS Student Interaction Design
+46(0)762-393458
kores.sophie@gmail.com

ABSTRACT

In this paper, we describe the process and different phases of the creation of the game Qbe, a third person 3d puzzle created in the course Interaction Design Project at the master program Interaction Design at Chalmers University of Technology. This paper covers both the design process as well as the implementation.

In the game, a robot is stuck in a cube and has to rearrange blocks to collect items in order to find his way out. The theme is lightness hence the blocks have different weights which can be modified using light sources. Once a block is illuminated it loses one weight.

General Terms

Documentation, Design.

Keywords

3D puzzle, Rubik's Cube, computer game, Xbox, XNA.

1. INTRODUCTION

The Qbe is a puzzle game developed for a project course at the master program Interaction Design at Chalmers. A robot is stuck inside a cube and in order to get to the next level (room) he has to collect the items placed in the cube. To assist him there are boxes and lamps in the room. The boxes has four different weights, from so light that the box is hovering in the air to so heavy that the robot is unable to move it. When a box is put in the light from a lamp, the light will cause the box to get lighter, and because of that also lose one unit in weight.

The game is meant to be a casual game, easy to learn but still providing the player with some challenge and hence forcing the player to think and figure out a smart solution to a the current problem.

2. BACKGROUND AND RELATED WORK

The initial idea was based on a Rubik's Cube and the intention of the authors has, from the very beginning, been to create and develop a well thought-through game idea rather than to create a game that looks good but does not bring any meaning to the player. The idea was very vague, and a great amount of time was spent to develop a working concept, considering more platform like games, different kinds of fun, intended users, puzzles,

mechanisms and other components that builds a game. In the end, a decision to develop a puzzle game was taken and the idea with Rubik's Cube was defined and delimited to the current concept.

The reason for developing a puzzle game was mainly to attract casual players to an everyday game that is easy to learn, to spend a few minutes (or more) to clear the mind, kill some time but essentially of course for the pleasure of playing a good game.

The game is intended for a casual gamer, described as a person who is not willing to spend neither that much money nor time on it. According to E. Adams (Casual vs. Core, 2000) the casual gamers "wants to play games the same way she watches TV or reads a book: sit down, do it for a while, then stop and do something else. She doesn't want games to consume her life; she wants them to entertain her for a while." Therefore, it is very important that the game is fun to play, since that is the only reason for a casual gamer to play the game; "the casual gamer plays for the sheer enjoyment of playing the game." (E. Adams, 2000)

Inspiration was taken from different games, some of them being Sokoban, Braid, Block Out (3D Tetris), Continuity and Cogs. Sokoban¹ and Continuity² are two games that are both simple to play at the same time as they forces the player to think through their moves thoroughly and therefor also brings a challenge to the game, a quality the Qbe strives to obtain as well.

The authors developed many different ideas and concepts, working them through thoroughly before settling on the final idea. One of the hardest problems to handle was to determine how the puzzles should be integrated in the large cube to bring meaning to the use of the large cube as well as the puzzles. Many logical and mechanical problems were encountered which will be further described later on.

Another decision to be made was whether the game should be adapted to the fact that it should be presented in public, in a exhibition in Nordstan³, and therefor adjusted for a larger screen,

¹ <http://www.joriswit.nl/sokoban/flash/>

² <http://continuitygame.com/playcontinuity.html>

³ <http://www.nordstan.se/>

spectators or other social aspects, for instance a meta game, a high score or similar.

A good game is, according to K. Salen and E. Zimmerman a game with a meaning. A good game is a game that brings meaning to the player, regardless if it is a casual game or a more extensive one. A good game is a game where the gamer in some sense know what to do and how to do it, the gamer can take actions that changes the state of the game and therefore bringing it forward and open up for new alternatives and new situations, where other actions can be made.

The decision to develop in XNA was taken for a couple of reasons. Prior discussions had listed the benefits and disadvantages with other platforms. In the end however, the reason for choosing XNA was because of a number of reasons. The authors of this report wanted to learn more about the development platform, and XNA had much documentation and samples available to get started. On one hand it might have been better for a causal game like the Qbe to be implemented in Flash and therefor easy accessible on a website. On the other hand, the aim of the game was also to provide an immersive experience, which would be easier to provide through XNA.

Immersion in this context means for the player to get involved in a game to the extent that the player forgets about the rest of the world while playing. There are many different kinds of immersion. This game does not aim for the form of reality immersion, usually found in simulators, but rather logical immersion. A form of immersion that stems from not how realistic the game world is when compared to the real world, but when compared to itself.

Markus Montola calls this type of immersion 360 illusion in the book Pervasive Games [3], but also adds that imagination is one of the key to logic immersion. Often the players can easily agree upon ignoring elements that might break the immersion. According to Montola this usually has to do with the environments not being able to fully able to create the environment wanted. For example a medieval village may be crossed by an electric grid. In a computer gaming world this often translates to objectives given to the player or hints.

In order to achieve the logical immersion in this game, a number of different design decisions have been made. First of all, the game play must be logical and not overly complex. The main game play is centered on pushing blocks and there is nothing more the player can do directly to the blocks. At first, the player might ask why he or she cannot lift the blocks, but citing the 360 illusion given by Montola, it is clear that this miss of a feature will be understood by the players – the puzzle become more fun and better – and within a short amount of time, the players will simply look beyond this as they have come to accept it.

2.1 The philosophical meanings of lightness

Lightness is a word with multiple meanings and for a game it can serve as a foundation of the user feeling trough the game. Apart from being the opposite of heaviness and darkness, the word can also be analyzed from a philosophical perspective.

A list of lightness properties was composed with the purpose to provide a foundation for the aesthetic design as well as the interaction design and the gameplay design. By using these concepts and discussing how the design can reinforce the feelings connected to lightness it is possible to make the game more emotionally involving.

- Lightness is the absence of *burden*

- Lightness is absence of *meaning*
- Lightness is *freedom*
- Lightness is *indifference*
- Lightness is *superficial*
- Lightness is *desirable*

This list consists of different peoples thought and theories. Some words are positive and some are negative, making it more dynamic and interesting to work with. Ideally the game should reflect both the positive and negative aspects of lightness.

3. CONCEPT

3.1 Puzzle

The theme imposed on the game was lightness. While this can be interpreted in a number of ways, the group thought of it in terms of weight and brightness. The theme was discussed and extensively brought up during concept sketches as hence became a natural part of sketching out the concept. While there were some issues fitting the theme and the concept, a concept closely based on the theme was developed.

The main concept is a third person, 3d puzzle game where the player has to move and affect blocks of different characteristics in order to finish each level. To do so the player needs to rearrange the blocks by pushing or pulling them, but also by pressing buttons and pulling switches. Blocks vary in weight and in shades of color. The heavier the block is, the darker it gets, and vice versa. There are four types of blocks, ranging from black to white. The blackest block is too heavy for the player to push or pull, waging four units⁴, while the white block is waging one unit. There are also magnetic blocks that repel or attract each other as well as static blocks which cannot be affected at all. (These blocks are implemented however not shown in any existing game level yet.)

The player may push or pull blocks of a total amount of three units. The consequence is that the player may push up to three white blocks on top of each other or one light grey block waging two units with a white block on top or beneath.

Moreover, light affects blocks. As the brighter the blocks are, the less they weight, the player may use light to make blocks lighter. When struck with light, blocks weight one unit less; turning a black immobile block into a movables dark grey block with a weight of three units. White blocks lose all their weight and will smoothly Hoover towards the roof of the level in the lamplight.

Furthermore, one of the main ideas was the concept of the outer game world. The game world is about to be represented as a giant Rubik's Cube. The different small blocks building up a traditional Rubik's Cube are the different levels of the game, and the player needs to rotate the cube to get from one level to the next. The idea is that each time the player finished a level (from now on called *a room*), the view was zoomed out to the Rubik's cube (from now on called *the cube*). From this view the player is about to be able to turn the cubes sides in order to get to another room.

The cube in the design is a 2*2*2 cube, meaning there could only be four levels in total. Having a cube larger than that would create a whole in the middle that would not be reachable in the current

⁴ A unit is a weight unit, for instance 1 kg. Since the metrics system is not used in the game it is just called "unit".

design. Therefore there should be several different cubes with their own set of rooms to enable more levels than four.

3.2 Gravity

When starting to develop the idea of how the large Rubik's Cube and the smaller puzzles in the rooms should be connected to each other a new problem emerged. When turning the large cube, how would gravity inside the puzzle rooms be affected?

The initial concept idea was that the large cube itself was subject to gravity. Thus direction of gravity would change inside the rooms depending on the large cube's current rotation. Blocks inside the rooms would fall to the new floor, which may have acted as a wall before. The main idea with this design was that some rooms would be impossible to complete unless the cube was turned a certain way. And some rooms might even need several revisits in different rotations to complete.

Then the question rose what should happen to the player in the room. Should it be possible to rotate the cube while being inside a room that would be affected by the rotation? And if so, where should the player be placed after the cube was rotated? A lot of sketches were made during these discussions to come up with a good solution. The most important question in need of an answer was what the main game play was going to consist of; to twist and turn the cube and solve the rooms primarily by using gravity, or to actually be inside the puzzle rooms and solve puzzles by manually moving blocks around in different ways.

Many long discussions and a couple of experiments led to the decision that the gravity changing would not be implemented. One of the reasons was, in addition to the already mentioned, was that a puzzle could become unsolvable. The main reason however was that a new idea had started to grow, a solution for puzzles that did not need gravity changing, instead using light and different weights to construct puzzles. The goal with each level was decided to get to a door, through that into the next room hence the next level. In the end it was decided that it was better to keep it simple and keep the gravity fixed to every room, as it could cause much confusion to the player.

Another problem related to the questions was also discovered: in what order should the levels be played?

This was also subject to much discussion, with a number of different ideas. One idea was to keep the game mostly linear, where the player follows a pre-determined path between rooms. This had the advantages that the planning of the rooms was easier. Another idea was that the player could rotate the cube freely and build his or her own path. While the advantages of this were more freedom to the player, the disadvantage was a largely different difficulty curve. A player could end up in a room before learning about the concepts, and could cause frustration. This was a major reason why it was decided in the end to go with a linear approach.

The decision made in the end was to use the cube as a progress bar and an easy meta puzzle. Hence the need of a clear and instructive design emerged in order to maintain the main function of the cube; the meta puzzle. One option was to make the cube rotate by itself but indirectly that would remove the function of the cube so that idea was discarded rather quickly.

To use the original colors and concept of sides of the Rubik's Cube was also soon discarded due to the lack of instructive design and the fact that the original puzzle of Rubik's Cube is rather complex and hard to solve for the inexperienced. Instead, with

some inspiration from the movie Tron⁵, another idea emerged. Creating sort of a map with lines on the outside of the different rooms would clearly annotate where the different rooms are connected through doors. Using different colors on the road map to indicate which doors has been connected and hence unlocked would also tell the player how far in the game he or she is.

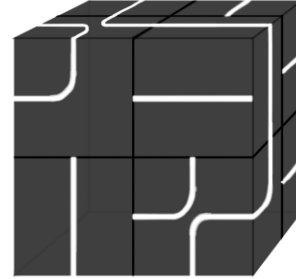


Figure 1. Sketch of the cube



Figure 2. Image of inspiration from the movie Tron

The games visuals are nice and clean trying to achieve an artistic feeling. The idea is to have the game look very much pre-designed, nothing that would appear in the nature or actually be built by humans.

The calmness of the visuals and sound help to detract the player from the world and makes him rather focus on the puzzle, while subconsciously being aware that he or she is in a quite realistic world. This will make it easier to immerse into the game.

4. REALISATION

A detailed plan was made to ensure the time available was spent wisely. As mentioned earlier, the aim was to create a game as good as possible considering concepts, game play, game mechanics and the visual parts of the game. This section will describe the process from the beginning to the end of the project.

4.1 Method

In order to structure the work Jones Phases [4] was used; starting with the divergence phase, followed by the transformation phase and end with a convergence phase. The divergence phase is mainly used for generating ideas using different methods, such as brainstorming, mock ups, sketches and video sketching. During the transformation phase all the ideas are further defined, evaluated and discussed. This phase allows for the developers to discuss and understand the different ideas and in the end move on to the convergence phase, the phase where the decisions are made. Here the good ideas are chosen, and further examined to

⁵ <http://www.disney.se/tron/index.jsp>

distinguish the possible use, can, for instance, some of the ideas be combined?

One important thing to remember is that no idea should be rejected until the last phase. Even though an idea might not be good in the beginning, with some further discussions and exploration it might be a good idea or generate an even better idea.

From the start, a number of ideas were agreed upon. The game would be a 3D puzzle game involving cubes or blocks. But the idea was still quite unclear and undefined, and as the group members probably had different views of it, it was decided that everyone would think about it on their own and then present the ideas.

After presenting the ideas to each other, and discussing them, brainstorming was used, as it's an excellent way to generate new ideas. The good thing with brainstorming in group is that you can easily help build upon other people's ideas. For keeping track of the ideas, post-it notes were used. These post-it notes became quite many, and they were then sorted into different categories to make it easier to compare ideas against each other.

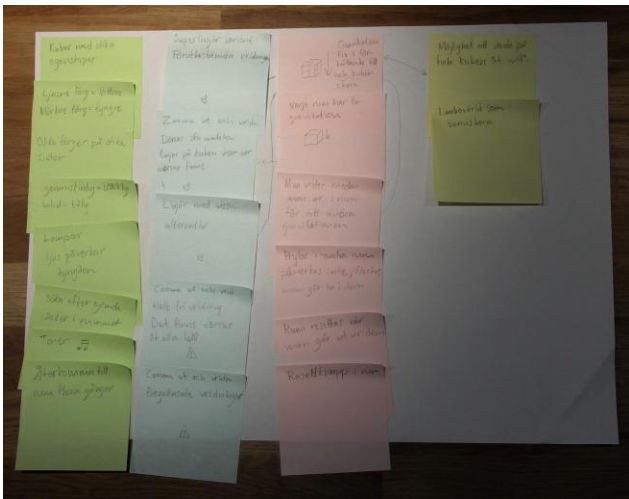


Figure 3. Mock ups

To further let the members of the group push for their ideas, and show what makes them interesting, mockups were used. In those mock ups examples of puzzles were given.

The video sketching⁶ was used to see how a level could look but also to see how the ideas worked out in practice. Most of the concepts and rules were already developed and decided upon, and therefore the video made a good example of the use of different mechanics and ideas. To see, as an observer, the actions taken is a good way to draw conclusions and identify problems and flaws in the design.

4.2 Planning

Jones Phases were used iteratively during the work, both when developing the game play idea as well as the visual design of the game. However, the smaller iterations are not denoted in the

⁶http://www.youtube.com/watch?v=qvtwxB5zYY&feature=mh_1olz&list=LL_V_w1gf4X2k

schedule below since they are taking place during just one week. The work actually met the schedule better than expected although the time for coding and graphics was underestimated.

To keep the work going during the process and to be able to see how the work was going according to the plans and schedule, some internal deadlines were set in addition to the courses' deadlines. In week two a deadline for a physical prototype was met to use for coming up with ideas. The two first weeks were devoted to fully develop a concept. In week three the digital preparations started, a digital prototype was developed. As cannot be seen in this image since this schedule was set up in week one, the divergence phase and the transformation phase exceeded one week each, and therefore the convergence phase was moved one week ahead.

At the time of the midterm presentation the ideas and concepts were settled. The development of the real game started. Until then, no discussions had been held about the visual aspects of the game, hence a new, shorter, round of Jones phases started to develop a design suitable for the game.

Continuously through the whole development of the game small user tests were conducted. First, paper prototypes were used and later on the different levels and elements of the game were tested using the digital prototype.



Figure 4. The first digital prototype

4.3 Multiplayer vs. single player

In the very beginning of the development of this game a vital question rose: whether the game should be single player, multiplayer or cooperational play. Since the main game play was decided to be to solve puzzles, the multiplayer and cooperational options were rejected. There should not be any competition between two or more different parts since that might change the focus of the game. Another aspect is that either an advanced AI (artificial intelligence) would be needed or the possibilities for a single player to start a game would be reduced or eliminated. To create an AI is an advanced and time consuming job and therefore not an option. One possible solution is to create an online setup of the game so that the players does not have to be at the same place for playing but that would still rise some problems; if no-one else is waiting to play the game and no AI exist the game would never be played. The intention of the game to be an everyday, casual game would then be spoiled due to the fact that an arrangement between two players has to be set up in order to play the game. Hence, in the end the choice was to create a single player game. At an early stage we decided to have a third person game and not like Tetris (with no player).

4.4 The robot

Presented below is a brief overview of the creation of the robot. To suit the rest of the theme the robot was designed to be nice and clean as well without any unnecessary decoration.

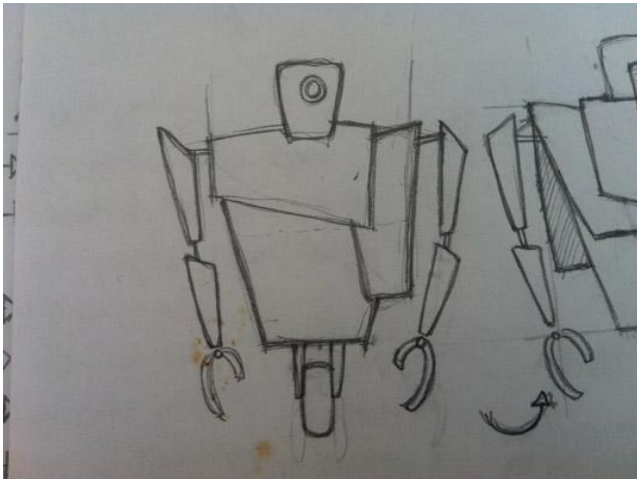


Figure 5. The first sketches of the robot

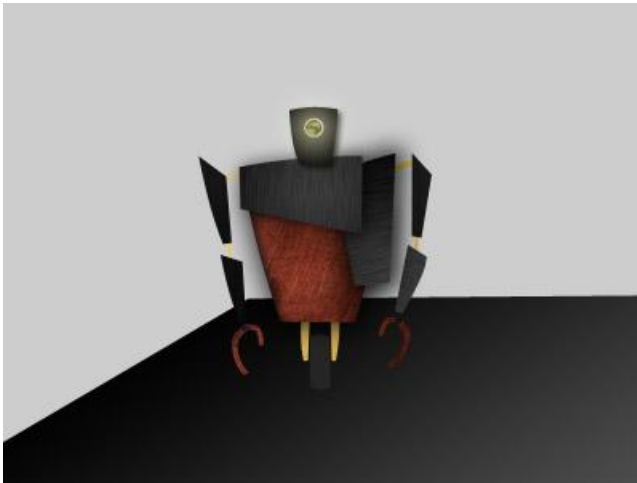


Figure 6. A further developed sketch with textures

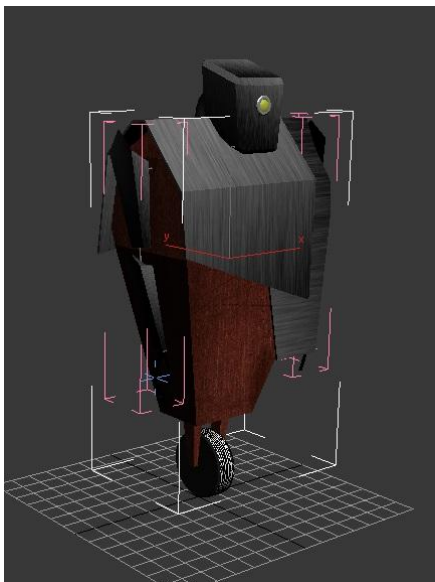


Figure 7. The figure in 3D

4.5 Usability testing

Most tests was conducted in an informal way, however a predefined structure of the testing were used. The reasons and goals for the tests were to examine and develop the interaction with the game as well as mechanics, levels and functionalities. Sauli Laitinen writes in the article *Better Games Through Usability Evaluation and Testing* in *Gamasutra* "Usability is about maximizing effectiveness, efficiency and satisfaction. This definition originates from the traditional software industry, but it translates well to game development. In games, usability is about delivering a better and deeper experience with less unnecessary interruptions or challenges that have not been designed by the developers."

Since games usually and hopefully are played for fun, and game designers tend to like people playing their games, it is very important that the game is well designed, considering the aesthetics as well as it is fluent, understandable, exciting and fun; that it meets the players' expectations. S. Laitinen writes in his article; "The reason why testing is important is that it provides direct and objective information about how real players play the game and what are the exact usability problems that players face when playing the game. This data is irreplaceable when developing the game and making it easy to use."

The test method used was mainly participatory evaluation since cognitive walkthrough performs best when the test conducted is divided into multiple easily distinguished parts and therefore was unnecessary complicated. During a participatory evaluation the test person is given a short description of the game and a playable prototype with the most basic commands and functionality available. The test person is then encouraged to speak his or her thoughts aloud, while the conductors remain silent throughout the entire process. Some different aspects, listed below, are important to keep track on during the tests.

- Performance - How much time and how many steps are required for people to complete basic tasks?
- Accuracy - How many and what kind of mistakes did people make?
- Emotional response - How does the test person feel about the tasks completed? How do the test person act during play?

During the tests some things were noted, one thing being the controls, playing with a hand control is easier than using the keyboard. Another thing was that the main game play worked rather well, with the different prototypes of levels used during the test the game was often considered to have "potential" when the levels had been further developed. It was often clear to the player what he or she should do, when not, using the method "trial-and-error" the right path was soon discovered. Another thing noted was that the camera view is very important to orient in the 3D world. Therefore possibility for the player to operate the camera view was implemented.

4.6 Prototype

Before coding the game engine, it was decided to first build a digital prototype, to be able to try out gameplay ideas. Since the time was limited it would not be possible to implement everything from the concept. Therefore the main ideas and features were prioritized. This was done by listing all the wanted features to implement and provide them with a priority number and implemented them according to the list.

As the prototype was simply a prototype, it was decided that the code should be thrown away after the first phase. Therefore, many shortcuts were taken during the development, which caused the code to be badly structured. On the other hand it boosted the speed of development and the prototype was running immediately.

One thing that became apparent during the development of the prototype was that there were a lot of features and things the concept did not define properly. There were many special cases how things, such as blocks, should behave that were never thought about. This forced further development of concepts to decide about.

4.7 Game Engine

When the work started on the real game engine, there was once again the need to prioritize. The prototype served well as a way to limit the work, which will be mentioned in more detail in the evaluation. In the process of implementing the game engine, both the prioritized features list and the features implemented in the prototype were targeted in the first hand.

The first task was to organize future code into a good structure, as opposed to the prototype. Almost all code written in the game engine was new, only smaller parts were re-used from the prototype. The same format was used for storing the levels, with just some smaller tweaks, which meant that levels created could be re-used and easily built upon.

5. EVALUATION

5.1 Exhibition

The game was shown in an exhibition, where many people were able to try the game out.

This was good from at least two perspectives. The rough ideas were forced to be polished; for instance creating menus and fixing a number of smaller details. In one sense it forced a second iteration of the game, now with a solid base to build upon, and less work the next iteration.

From another perspective, the exhibition worked great as a play test much larger than the play tests we conducted earlier, both in the number of people trying out the game, and the range of different people in age and experience.

It was also a better opportunity to observe and see how different people thought about how to approach a level. The on beforehand intended solutions were shown not to be the only ones, people who are new to the game tend to see other possibilities and solutions. To see how people expect things to work can help in developing the game further, especially in areas of what features need more focus in the tutorials.

5.2 Programming and Design

When creating Qbe as a team of only four, the authors all needed to partially act as both developers and designers. The main focus was to make a great game, and in order to achieve this, the development process started in a game designer perspective. In this phase the user experience was designed as well as the game rules and the interactive aspects.

When the time had come to create the first digital prototype, the group was divided in two, one half started to focus on the programming, while the other half continued to fine tune the design. Since the group consists of software engineers some insight was shared in the coding process as well. This resulted in a transition of roles from being pure designers to being both designers and programmers.

The digital prototype phase acted as the transition phase. Many design decisions still needed to be made, and since good code was not prioritized in this phase most focus could still be put on the game design. To avoid getting carried away by implementation level questions different shortcuts were frequently used.

The programming actually made it easier to look at the game's rules in a more structured way. To code the rules a good understanding of the rules are necessary. During this phase, several areas were identified where the rules were not explicit enough, areas in need of improvement and other things that did not work as expected. Each time one such situation arose the new input was discussed and a solution could be investigated. Being both a programmer and a designer in this process was a great benefit. To be a pure programmer in a situation like this would result in either needing to contact the designer for clarification or making a decision without them.

Simultaneously, the programming of the prototype was a good lesson of how the implementation of the real game would work. The same programming environment (C#/XNA) was used for both the prototype and the final game, so the prototype worked as a fast tutorial in the skills needed later on. One of the programmers had never used C# before, while the other only had limited prior experience.

When the work began on the final version of the game, more focus went to the technical aspects of coding, but by then, on the other hand, most of the important game design decisions had already been made.

5.3 Thoughts of improvements

An improvement of the process used in the Qbe project would be to make the stages and transitions between roles even more visible. Using the vocabulary from Rules of Play, there is a need to explicitly talk about "the rules" and how to modify them to better understand if a particular discussion is concerned with programming solutions or game design.

Even though it is natural that the rules of the game change during the programming, this cannot be allowed to happen unknowingly. Designers need to maintain control over the design, and by actively talking about the rules and how they are refined throughout the process this can be achieved (K. Salen, E. Zimmerman, 2004s, (p. 142-148)).

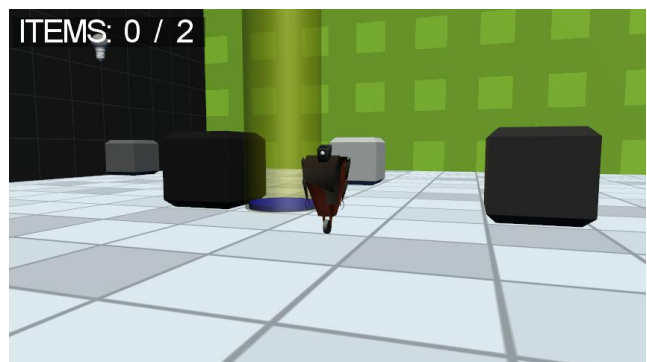


Figure 8. The game as it looks in the end of the course

6. DISCUSSION AND FUTURE WORK

There are several areas of improvement in the current version of the game. The game provides a stable basis for further development, and supports additions of several game concepts,

that were thought of during the design phase but that have not been fully implemented yet.

The main areas of improvement are listed and briefly discussed below.

6.1 Game concepts

Magnetic blocks are implemented in the program code, but not used in any of the current levels.

6.2 Aspects of lightness

Adding some extra functions to the use of the concept of lightness might provide the game with several extra interesting mechanics.

For instance, hovering blocks could be added where the player can stand on a white block while it rises, making it stop mid-air. This will bring the weight of the robot into the game.

The falling speed and the controller vibrations when blocks hit the ground after falling already differ between blocks of different weight. To further enhance the experience of different weights animations of the robot could be implemented. By making the robot look different when pushing a light block compared to when pushing a heavy the impression of weight will be more apparent. Even small signs of struggle could make a big difference, such as irregular blinking with the eye/lamp in the robot's face when something is heavy.

Sounds could be an effective means of communicating weight. Adding sound effects when a block lands on the floor after falling would be suggested.

By subtly animating the boxes the weight can be further enhanced. For example when the player jumps on top of a white flying block a certain amount of sway before stopping midair would make the scenario a lot more realistic, and a very slight elastic deformation of a block that lands after falling could make it look much better. The possibilities of animations could be further investigated to find more uses.

6.3 Interaction design

Visual clues to interactive possibilities were planned in the design phase, the most important one being textures for the blocks. These textures would help the player distinguish the different colors of the blocks from each other, and also connect an actual number to each weight. This would allow the user to count the total weight of stacked blocks to determine whether they can be pushed by the robot or not.



The control of the camera needs to be slightly adjusted, as many users found it a bit difficult to control in the play tests. To succeed with this it will be necessary to look more closely on the camera properties of some similar games. Fine tuning of controls is important to maintain game flow.

A tutorial will be necessary if the game is to be played by people who are not already familiar with it, and in order to further improve the understanding of the game many interactions are in need of clarification. Examples are pushing and pulling blocks,

which should be made clearer by adding animations of the robot arms.

More feedback from the game based on player action is also needed. For example, when a button is pushed, it should be shown visually that it's actually pushed down, but it can also be enhanced with other forms as feedback, such as sound. Another example is when a block is too heavy to push or pull. This information need to be communicated to the user in an understandable way. It can be done either by adding text information, or by the use of sound and graphics. One suggestion is also to use a sort of thinking bubble for the robot, where the robot's thoughts let the player know what is happening.

6.4 Theme, environments and emotions

The graphical theme of Qbe could be further developed. By creating much more concept art, color charts and character sketches more ideas could be generated. Further discussion about what feeling to convey to the player would create a more interesting and appealing atmosphere. This should be done with focus on the target audience. Focusing on creating an holistic atmosphere would improve immersion.

Textures and colors could also be different in different levels, to increase the variation. Improving the lighting quality and adding shadows would get rid of some of the flatness of the current game.

Background music would improve the emotional aspects of the game. A sound track was created, but has not been added to the game yet. The music was designed to give a very calm and atmospheric feeling. There will be no harsh sounds or sounds made to attract attention, but rather to fit into the scenery.

7. CONCLUSIONS

The long and iterative design process, preferably using Jones phases, has contributed a lot to the outcome of this project. It is estimated that the project would not be as successful if game design had not been as developed in the early stages of development. Furthermore, the software prototype proved very helpful in designing game mechanics and proving the overall concept.

All in all the developers are very happy with the results, although a game with more levels, game mechanics, sound and music and more textures and better models is desirable and also planned to implement.

It is unlikely that the initial design phase could have been any shorter. The phase took the time needed, to develop a solid foundation. From the start, there were many very different ideas amongst the different developers, and if all concepts had not been examined decided upon on a together, it could have caused serious trouble later on. When starting to implement the game it was very clear what to do, which in turn made the long design phase worthwhile.

It is the strong understanding of the authors, that Qbe, when further developed according to the future work headline above, has a very strong potential of being a fun and solid, well thought trough game, which was what the developers wanted to achieve with this project.

8. ACKNOWLEDGMENTS

We would like to thank docent Staffan Björk at the Institution of applied IT at Chalmers University of Technology for his help during the development of the game concepts.

9. REFERENCES

- [1] Laitinen, S. (2005) *Gamasutra: Better Games Through Usability Evaluation and Testing*, [Online], Available: http://www.gamasutra.com/view/feature/2333/better_games_through_usability_.php [26 May 2011]
- [2] Salen, K., Zimmerman, E. (2004) *Rules of Play - Game Design Fundamentals*, MIT Press
- [3] Montola, M., Stenros, J., Waern, A. (2009) *Pervasive Games: Theory and Design*, Morgan Kaufmann Publishers
- [4] Hileman, R. (1998) *Design Methods: seeds of human futures - By John Chris Jones 1970*, John Wiley and Sons, New York and Chichester, [Online], Available: <http://www.smsys.com/pub/dsgnmeth.pdf> [26 May 2011]
- [5] Adams, E. (2000) *Casual Versus Core*, [Online], Available: http://www.gamasutra.com/view/feature/3120/casual_versus_core.php [26 May 2011]