How to engage university students in reading H.C. Andersen's fairy tale "The Little Mermaid" through a serious game

6th semester project by group 605

Authors:
Irene Ubieto Alamillo
Delan Kasim Ali
Niklas Fruerlund Jensen
Thevakorn Kjær Lauritsen

Supervisor:
Thomas Bjørner

Medialogy
Aalborg University of Copenhagen
Denmark
June 3, 2020

Aalborg University Copenhagen

Semester:
6th

Title:
How to engage university students in reading H.C. Andersen's fairy tale "The Little Mermaid" through a serious game

Aalborg University Copenhagen
A.C Frederikskaj 12, 2450
Copenhagen SV, Denmark
Semester Coordinator: Thomas Bjørner
Secretary: Judi Stærk Poulsen
Phone: 9940 2468
Judi@create.aau.dk

Project Period:
6th of February to 3rd of June

Semester Theme:
Interactive Systems Design (CPH)

Supervisor:
Thomas Bjørner

Project group no.:
605

Members:
Irene Ubieto Alamillo
Delan Kasim Ali
Niklas Fruerlund Jensen
Thevakorn Kjær Lauritsen

Abstract:
This project uses H.C. Andersen's fairy "The Little Mermaid" in a serious game to engage university students to read more. This project builds on previous work for both serious games and engagement, specifically reading engagement in an interactive media narrative and storytelling. The game itself is done in Unity 3D, has three main levels and multiple transitions levels. A progression system is implemented to make sure the participants are not missing out on any important story element. This project consists of 25 participants within the age range of 18-24, 20 being male and 5 being female. The participants were recruited online through social media. The test itself consists of a questionnaire, a web-based game and a semi-structured interview. The questionnaire consists of four parts ("Introduction", "Mid-game 1", "Mid-game 2" and "After the game"). The results for this project show an increase in engagement within the game by introducing new information but decreases when they finish the game. However, the implications of these results do suggest that more work needs to be done and more areas can be explored.

Pages: 57
Finished: 03/06-2020

Table of Contents

1 Introduction

Over the past few decades, young adults have an decreased reading of novels and have been spending more time reading online than before (Nørtoft 2019; Twenge, Martin, and Spitzberg 2018). This is a huge concern when this age group enters an educational structure, which requires a higher level of reading ability to gain knowledge from e.g. articles and textbooks (Respondek et al. 2019).

The idea for this project is to use serious games to achieve engagement in reading, by involving the university students personally in the story. This project uses H.C. Andersen's "The little Mermaid", a fairy tale which is widely known amongst Danish university students, although it is most often the Disney version that the students knows. While the main parts of the story remain largely the same, the ending is vastly different. In Hans Christian Andersen's version, it ends with the little mermaid giving up her own life instead of sacrificing the prince to get her mermaid form back. Whereas the Disney version, ends with the little mermaid getting married to the prince. Due to many university students knowing the Disney version this project uses the Hans Christian Andersen's version, so the students experience another ending than they know. We will design a serious game following this fairy tale, and use it as a medium for making the students lose consciousness of their surroundings and of what they are doing so they can only be aware of the story itself.

Designing a game where the player is the main character, where they will be able to interact with and proceed with the story at their own pace. This kind of interaction with the environment will influence the player to react and respond with actions focused on the narrative. By letting the player know the story with a video game, it will create a way to engage them to the narrative itself. This will happen because the game will make them change their perception within the story as they live it as a personal life experience, which will give them opportunities to get engaged in the challenges and the narrative (Slota et al. 2015).

This has led to this project's study research question which is: *How to engage university students in reading H.C. Andersen's fairy tale "The Little Mermaid" through a serious game?*

2 Previous Research

2.1 Engagement

Engagement is a term that can have varying degrees of definition based on previous research within the field of user experience (O'Brien and Toms 2008), and is characterized by attributes such as challenge, positive affect, endurability, aesthetic and sensory appeal, attention, feedback, variety/novelty, interactivity and user control (Hassenzahl and Tractinsky 2006; Jacques, Preece, and Carey 1995; Laurel 1993; Chapman, Selvarajah, and Webster 1999). Engagement is also a meta-construct with behavioural, affective, and cognitive components that vary both situationally and dispositionally. Effort and task persistence constitute some of the behavioural components of engagement, while the affective components include valence, arousal, and discrete emotions like interest and curiosity. The cognitive components of engagement include attention, concentration, and the use of learning strategies (Mills et al. 2013). A study (Schoenau-Fog and Bjørner 2012) has divided engagement in video games into 6 different types: Sensory, Physical, Intellectual, Social, Emotional and Narrative engagement. For this project, Narrative, intellectual and reading engagement will the primary engagement types.

The narrative of a game contains multiple aspects that shapes the game. These aspects are: Story world, character, emotion, game world, narrative interface and the narrative arc (Schoenau-Fog, Louchart, et al. 2013; Bizzocchi 2007). The narrative aspect are not the only way for a player to be engaged, they can also experiment with how they can affect the narrative, and see how they are able to change the outcome of a story (Schoenau-Fog, Louchart, et al. 2013).

Intellectual engagement revolves around challenges, activities and creativity where the player's intellect is tested (Schoenau-Fog, Louchart, et al. 2013). This type of engagement is enhanced by activities involving problem based forms of learning and challenges which can increase player's motivation to learn (Schoenau-Fog, Louchart, et al. 2013; Reid 2012).

Reading engagement can be defined as the interest and attitude towards reading, and the time used to read a diversity of material for pleasure (Brozo, Shiel, and Topping 2007). Engaged in reading requires the reader to be motivated for external gains and the text's content, strategic in the way of comprehending the text, knowledge-driven to use prior knowledge and gain new knowledge, and socially interactive to use the knowledge and/or lessons learned from the text in social interactions (Naumann 2015; Guthrie, Wigfield, and You 2012). The frequency of both leisure reading and educational reading also contributes to the engagement in reading (Brozo, Shiel, and Topping 2007; Solheim and Lundetræ 2018; Cox and Guthrie 2001). Both reading engagement and reading practice allows for more and/or better reading achievements and more motivation towards reading. Better readers tend to be more motivated in reading and therefore read more, which leads to improvements in vocabulary and comprehension skills, whereas poor readers experience a decline in skill level (Brozo, Shiel, and Topping 2007; Solheim and Lundetræ 2018). Frequent reading activities also enables readers to discuss an array of topics and comprehend different viewpoints in social groups (Cox and Guthrie 2001). Reading engagement incorporates other forms of engagement in terms of emotional engagement to both positively (interest) and negatively (boredom) affect the engagement of the reader, and cognitive engagement for the reader to exert their mental effort to comprehend the text (Guthrie, Wigfield, and You 2012; Cox and Guthrie 2001). However, by introducing another cognitive activity which distracts the reader can negatively affect the amount of information the reader can recall (Vanco and Christensen 2016). Between the age of 22 and 90 there are no statistically significant difference between the reading achievements of the genders. However, around the age of 15 there is a difference favouring girls. Furthermore, girls are favoured for narrative and/or continuous texts. Although both genders' reading ability is related to reading frequency, boys' reading ability is more affected by their attitude, enjoyment and interest in the text's content than the girls (Solheim and Lundetræ 2018). Reading

engagement after the age of 15 is important to maintain and develop reading skills, which in return will help students overcome some barriers to academic success (Brozo, Shiel, and Topping 2007).

2.2 Transmedia storytelling

One way to engagement a user in reading is through transmedia storytelling. The concept of engagement in transmedia storytelling, in terms of text and video games, is to engage the learner in the storytelling experience and make them focus on the story itself, instead of the text or gameplay alone. Transmedia storytelling allows the learner to involve them personally in the story of the text, by knowledge from it being understood by the learner without the need to decode and understand the text through bottom-up processing (Pasalic et al. 2017). Using transmedia storytelling can make the learner gain experience and knowledge through complex texts without the cognitive load of plain text by using their interactive involvement (Pasalic et al. 2017; Dubbles 2009). This also allow for the learner to be engaged in the text by having the them be the center of the story, able to progress at their own pace and involved in the story (Pasalic et al. 2017). By using a media that the learner is comfortable with, interested in and have a positive attitude to the learner are more likely to engage in the text and therefore more likely to better comprehend the text. Using video games as the media allows the learner to experience, comprehend and be engaged in topics that are complex, provocative and motivating, along with topics that are not often present in text due their complexity (Dubbles 2009).

2.3 Serious Games

Games can cover a variety of activities and its definition strongly depends on the users' perspective of the game (Pourabdollahian, Taisch, and Kerga 2012). Games that are aimed at education and learning can be defined as "Serious Games" (Wouters, Tabbers, and Paas 2007). It is defined by some researchers as an activity whose main purpose is learning serious context through playing while entertainment, enjoyment or fun is not(Charsky 2010; de Freitas 2006; Zyda 2005; Djaouti, Alvarez, and Jessel 2011; Michael and Chen 2006). A game has to be challenging for the player in order for them to be engaged and learn something from it (Hamari et al. 2016; Fullerton 2008).

3 Methods

3.1 Participants

The 25 participants were danish university students attending first - fourth semester, around the age of 18 - 24, as they were the age group reading the least. Student were, for this project, defined as; an education that was at least higher than Gymnasium level, and requires heavy Reading literature, such as; a nurse or a medialogy student. This study included:

ID	Age	Gender	Field of Study	Game-time in hours pr. Week	Novels read within 3/6 months
01	20	Male	Computer Science	1-10	0
02	24	Female	Medialogy	11-21	1
03	22	Male	Humanitarian	over 30	2
04	20	Female	Nurse	0	4
05	19	Male	Computer Science	11-20	0
06	21	Male	Computer Science	21-30	0
07	20	Male	Medialogy	over 30	0
08	23	Female	Medialogy	11-20	0
09	23	Male	Medialogy	1-10	0
10	21	Male	Medialogy	11-20	0
11	24	Female	Medialogy	1-10	1
12	23	Male	Electrical Engineering	11-20	0
13	19	Male	Medialogy	11-20	1
14	21	Male	Medialogy	1-10	0
15	23	Male	Medicine	over 30	0
16	24	Male	Medicine	11-20	0
17	21	Male	Medialogy	11-20	0
18	19	Male	Electrical Engineering	1-10	0
19	20	Male	Medicine	0	2
20	23	Female	Medialogy	11-20	0
21	24	Male	Sustainable Design	0	0
22	22	Male	Sustainable Design	11-20	0
23	18	Male	Techno Anthropology	21-30	0
24	20	Male	Techno Anthropology	11-20	0
25	24	Male	Techno Anthropology	11-20	0

Table 1: Participant profiles

Since the testing was performed on students who, likely had varying levels of reading speed. It became important that we respect each individuals speed, and not hurry them or judge them by said speed. It was also important that we did not judge their intellect based on their understanding of the text, how much they remembered, nor how fast they read. The following was the informed consent used during the experiment:

I consent to completing this questionnaire that the group 605, 6 Semester of Medialogy on behalf of Aalborg University can analyze and store data for two years. Attendance is voluntary and you are free to withdraw or cease participation at any time. You will remain completely anonymous and the data is intended for university and research purposes only.

3.2 Data Collection and Procedure

This project had an Iterative model, where multiple pilot tests took place before the actual test. These tests was used to optimize; the questionnaire, interview questions, and the games quality, as the game should not cause the engagement to fall due to issues the participants could run into.

The participants were be contacted via Discord or Zoom, where they were given a introductory questionnaire, to gather their initial knowledge and information regarding the participants. The Participants were then given a game of the little mermaid, before undergoing a second questionnaire, focusing on what they have learned from their reading, and how engaged they were.

After they played the second level of the game, they returned to a third questionnaire. The third questionnaire looked at the same elements, as the second questionnaire. Afterwards the remaining part of the game was completed, before they received the fourth and final questionnaire. This questionnaire focused on; their engagement, and also their understanding (if they remember anything, as we wanted to make sure it has been read, and to check if they actually engaged in the text). Afterwards an interview was conducted to gain further knowledge into the participants; level of engagement and understanding.

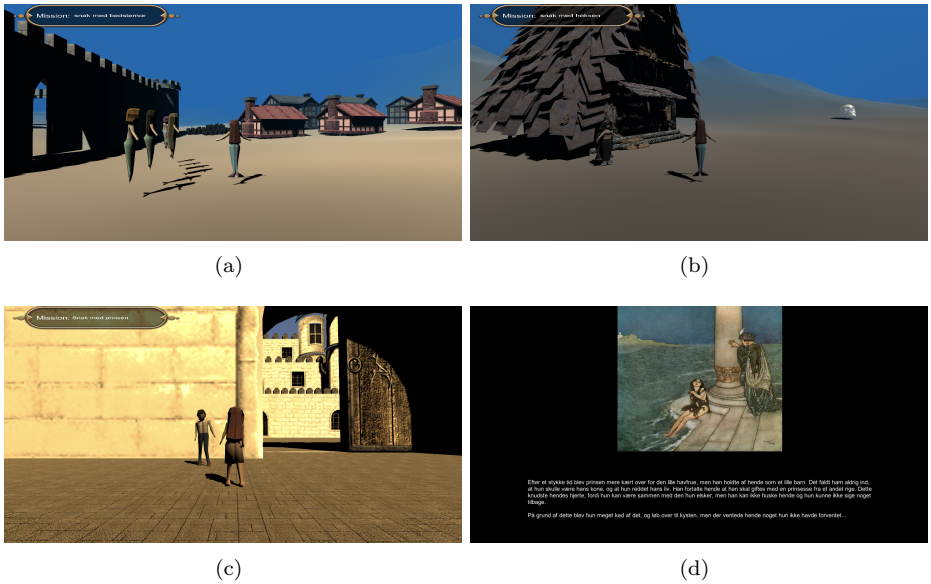
To collect the data, four surveys were used throughout the test as well as an interview at the end. These surveys and the interview, was planned to be answered by participants gathered through convenience sampling where students are from a university, they spoke danish, went to educations with required literature. They were people gathered from Facebook and Discrod primarily. The interview was a semi-structured interview (Appendix section 7.1), and the survey was constructed using Likert scales with points ranging from 0 - 5, and was be treated as ordinal data when data analysis took place.

3.3 Data Analysis

To make sure that our data is normally distributed and the variance between them are roughly equal, we used an Anderson-Darling test, and a histogram. The mean was calculated for each group (Game engagement, and story engagement) at the first stop (Mid-Game 01) and the second stop (Mid-Game 02) to compare the engagement level across the whole game. The histogram was also used to visually compare means. The data was then treated as non-parametric as our sample size is small and had a non-normal distribution. Spearman's correlation coefficient was used for the relationship between the engagement level for the game and for the story.
The gameplay was be recorded and used to analyse the players behavior in the game based on different themes; how long they spent reading, how often they got lost in the game, and how often they needed guidance (due to not reading the text from the game). This data along with the interview was used to further back-up data gathered from the survey, and also be used to explain potential reasons to why the result became what is discovered during evaluation. The interviews were analyzed by traditional coding (Bjørner 2015) in which the interviews are transcribed, then; separated, organize into themes, and then interpreted.

4 Design and Implementation

The game was designed in Unity 3D, using C# and is both playable on Mac and PC. The level design is based on three main scenes. The first one is under the sea in the water castle scene, and is a scene created mainly to trap the player into the atmosphere of the story and get them to know the premise, as we use this scene to tell the introduction of the fairy tale. First, the player will have to go and speak with the little mermaid's grandmother, and she will tell them to speak to the sisters. Once that is done, the grandmother will ask the player to collect oysters and pass through rings. The second scene is under the sea in the witch's place, and it covers the part were the little mermaid goes to talk to the which in order to get her legs. In the scene, the player will need to speak to the which and get the goal of collecting skulls and passing through rings in order to make the potion. The third and final scene is on the surface near the prince's castle. In this scene the player will need to speak to the prince and the sisters. In each of the scenes the player has the main goal of the scenes is to talk to the NPCs (non-player characters) to get to know the story. Besides those objectives, the small challenges were designed in order to get the players used to the movement of the character, to keep them focused and avoid them getting bored. These missions are simple goals (like gathering oysters or swim through hoops). We also designed scene transitions to give the players context about the fairy tale and about what they are doing.



(a) (b)

(c) (d)

Figure 1: The different levels within the game: Underwater Castle Scene (a), Witch's Place (b), Prince's Castle (c) and Scene transition (d)

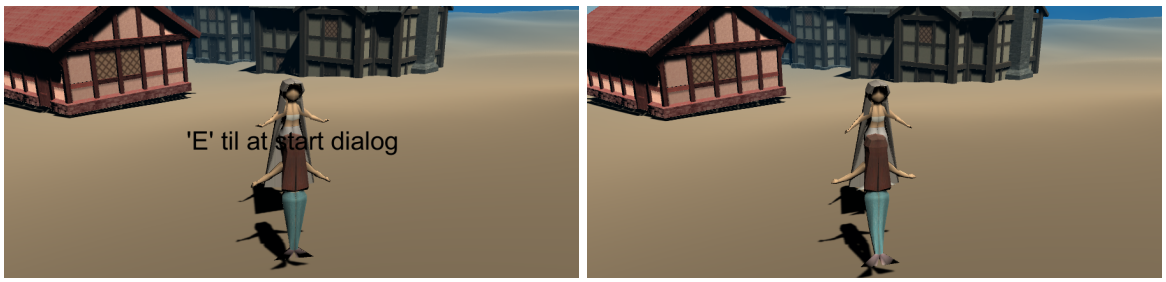
The assets used for the all of the scenes were gathered from turbosquid, free3d, cgtrader and unity asset store (*Turbosquid* 2020; *Free3D* 2020; *Cgtrader* 2020; *Unity Asset Store* 2020). The music and sounds are from a copyright and royalty free youtube channel (*CO.AG Music* 2020). The models and animations of the player and NPCs were designed by ourselves using Blender.

We used assets to follow the low polygon aesthetic, and did the rest of the models and characters ourselves. We did not focus on the engagement of the player by the graphics of the game and designed the scenes in order to fit the fairy tale and set the atmosphere.

(a) (b)

Figure 2: a) Underwater Assest, b) Land Assest

A progression system were implemented to ensure that the player did not miss any of the story and experienced the story chronologically. The system goes through each object in the scene belonging to the progression system to see how many stages were in the scene and how many objects were in each stage. To progress in the system the player must complete/interact with each object within the current stage, where the last stage contained a collider object that transitions the player to the next scene.



(a) (b)

Figure 3: Example of the difference of being on the character's stage and not on its stage. Translated text: 'E' to start dialogue

```
1  void OnStageUpdate()
2  {
3      currentStage += 1;
4      foreach(ProgressionData currentObject in data)
5      {
6          if (currentObject != null)
7          {
8              if (currentStage == currentObject.stage)
9              {
```

```
10                  ... //Collapsed code for activation of the interaction with the object
11              }
12              else
13              {
14                  ... //Collapsed code for deactivation of the interaction with the
                        object
15              }
16          }
17      }
18  }
19
20  public void OnCompletion()
21  {
22      completedObjects +=1;
23      if (completedObjects >= valueArray[currentStage-1])
24      {
25          OnStageUpdate();
26          completedObjects = 0;
27      }
28  }
```

Snippet 1: The code for stage update and stage completion. The full code can be found in appendix in section 7.6.20.

The OnStageUpdate method on line 1 of snippet 1 starts by increasing the current stage by 1 (line 3) to signify the arrival of a new stage. In the lines 4 to 17 the code goes through each gameobject tied to the progression system and checks if the current stage corresponds with the given gameobject's stage. If it is the same the code will enable the interaction with the gameobject, if not it disables the interaction, so previously enabled interactions get disabled. The collapsed code on lines 10 + 14 is a series of if-else statements checking which kind of interaction the current gameobject has and disables/enables them accordingly (The full code can be seen in appendix in section 7.6.20). The OnCompletion method (line 20) is called every time an interaction related to progression is completed. It starts by increasing the counter for completed objects by 1 (line 22), where after it checks in line 23 if the amount of completed objects are higher than or equal to the amount of objects in the current stage (to see if everything in the stage is completed). If it is higher or equal it calls the OnStageUpdate method and sets the completed objects to 0 to reset the counter.

To ensure that the player were able to experience as much of the story outside of interpretation, the player would be able to interact with certain NPCs (Non-playable characters) through dialogue. The dialogue helped the player feel like they were a part of the game's world. This mechanic were encountered early in the game so the player would know that they could engage in dialogue with certain NPCs.

Figure 4: Example of the dialogue window

```
1  public void startDia(Dialogue dialogue)
2  {
3      animate.SetBool("IsOpen", true);
4      name.text = dialogue.name;
5      sentence.Clear();
6
7      foreach(string sentences in dialogue.sentence)
8      {
9          sentence.Enqueue(sentences);
10     }
11
12     DispNextSent();
13 }
14
15 public void DispNextSent()
16 {
17     if(sentence.Count == 0)
18     {
19         EndDialogue();
20
21         if (!completed)
22         {
23             FindObjectOfType<Progression>().OnCompletion();
24             completed = true;
25         }
26         return;
27     }
28     string sen = sentence.Dequeue();
29     StopAllCoroutines();
30     StartCoroutine(typeSent(sen));
31 }
```

Snippet 2: The code for dialogue start and next part of the dialogue

The StartDia method on line 1 of snippet 2 starts by setting the animation Boolean of IsOpen to true (line 3). It then changes the variable name.text to dialogue.name gained from the dialogue object (line 4) and clears the sentence queue (line 5). It then goes through every sentence gained from the dialogue object (line 7) and puts it in the sentence queue (line 9). Lastly, it calls the DispNextSent method (line 12). The DispNextSent on line 15 starts by checking if there are any sentences left in queue (line 17). If there are no sentences left it calls the EndDialogue method (line 19; The method can be seen in appendix in section 7.6.6) and if the current dialogue has not been "completed" before it calls the OnCompletion method from the progression system (line 23) and marks the dialogue as completed (line 24). In line 28 the code sets the sen variable to the first item in the queue and removes it from the queue. Where after the code stops all coroutines (line 29; used to display the text) and starts a new coroutine with the newest text from sen (line 30)

During the game, the player will traverse two of the scenes under water. Instead of only being able to move the character on two axes they will be able to move the character on three axes, using the Space bar (ascend) and shift key (descend). Furthermore, the player character will also be affected by a small amount of gravity to simulate slowly sinking in the water.

```
1  public override void Movement()
2  {
3      float vertical = Input.GetAxis("Vertical");
4      float horizontal = Input.GetAxis("Horizontal");
5      Vector3 newMove = Camera.main.transform.forward.normalized;
6      Vector3 forwardVel = new Vector3(newMove.x, 0, newMove.z) * vertical * speed *
           Time.deltaTime;
7      Vector3 sideVel = Camera.main.transform.right * horizontal * speed *
           Time.deltaTime;
8      cc.Move(sideVel + forwardVel);
9  }
10
11 void ascend()
12 {
13     if (Input.GetKey("space") && transform.position.y < 140)
14     {
```

```
15          Vector3 ascendVel = new Vector3(0, 1, 0) * speed * Time.deltaTime;
16          cc.Move(ascendVel);
17      }
18
19  }
20
21  public void Gravity()
22  {
23      if (gravityForce != null)
24      {
25          Vector3 gravityVel = new Vector3(0, -gravityForce, 0) * Time.deltaTime;
26          cc.Move(gravityVel);
27      }
28  }
```

Snippet 3: Code example for movement in water

The movement method starting at line 1 in snippet 3 works by getting the arrow keys or WASD input from the player (line 3 + 4) and the direction of the camera (line 5). In line 6 + 7 the input and camera direction are used to create vectors for moving forward/backward (line 6) and sideways (line 7). Lastly, the .Move (line 8) is used to move the player character in the direction of the vectors. Both ascend (starting at line 11) and descend works the same way, by moving the character in the y-axis (positively for ascend and negatively for descend). The gravity acts the same way as descend. However, it is constant instead of on input and is not affected by the movement speed of the character.

5 Evaluation

Our data showed that from the 25 participants, within the last 3 months; 21 had read no novels, 2 had read 1 novel, and 2 had read 2 novels. Correlating with our findings in Introduction (See section 1).

5.1 Game engagement

There were a slight increase in engagement from "Mid-Game 01" (The first pause as seen in the questionnaire in section 7.4) (Mean = 3,36) to "Mid-Game 02" (The second pause as seen in the questionnaire in section 7.4) (Mean = 3.52), as well as "Mid-Story 01" (Mean = 3.96) to "Mid-Story 02" (Mean = 4.08). While a slight decrease was seen between "Mid-Story 02", and "Read the rest of the story" (Mean = 3-96). Meaning that the players became more engaged throughout the game, but it then fell when they finished the story (see figure 5).

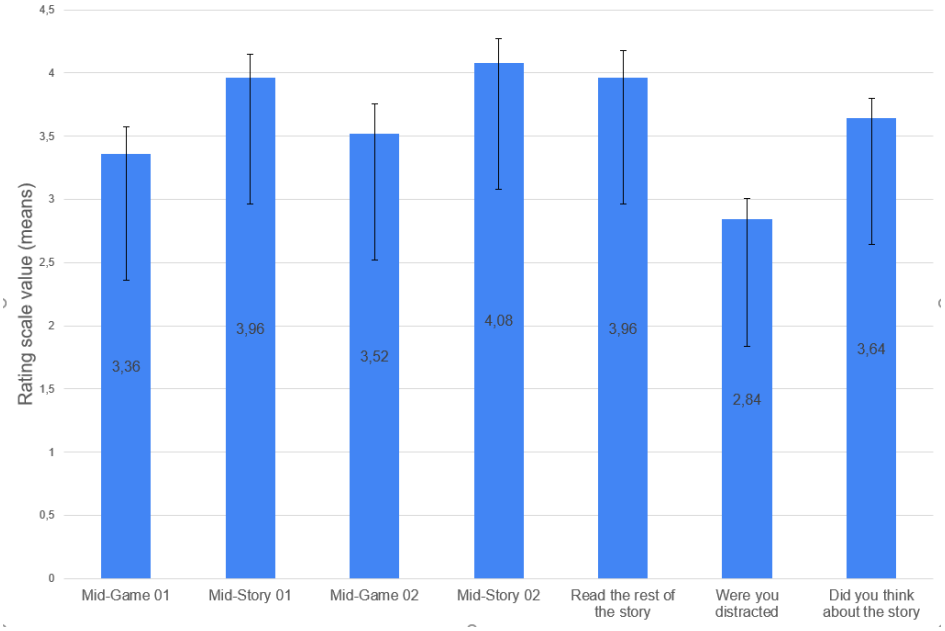


Figure 5: Engagement levels though the differing stages of the game using means, and Standard error as error bars. Scale 0 - 5. n = 25)

In figure 5 we see the average rating (the mean) for each question asked pertaining to their engagement. Majority of the data lies in a neutral area (3 = neutral in our scale), with the question "were you distracted" being primarily negatively skewed (2 = Disagree). From this we can see that they were not severely distracted during the game (Mean: 2,84). Most participants disagreed or simply felt neutral when asked if they were distracted, during the questionnaire and interview. That being said there were some distractions through the game, for the participants.

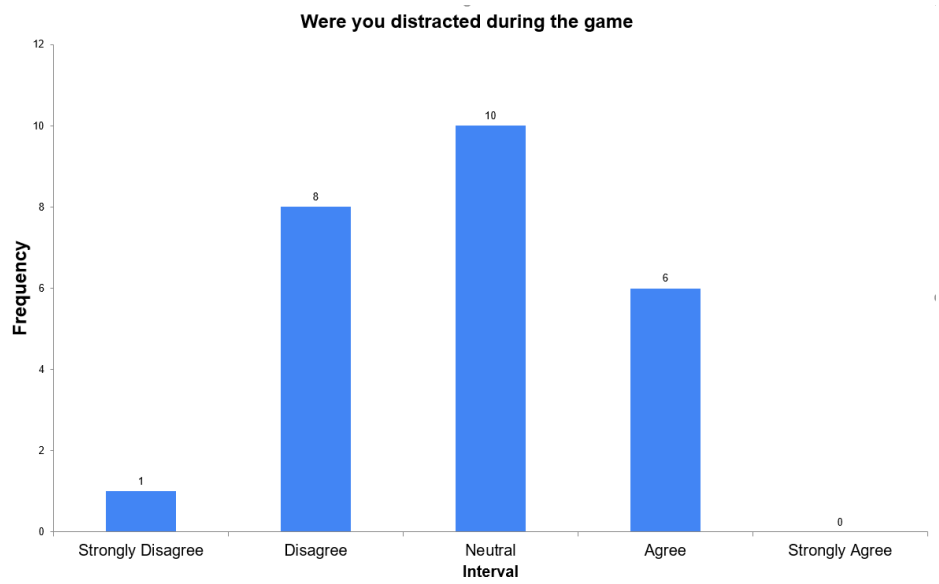


Figure 6: How distracted the participants felt during the test Scale 1 - 5. (n = 25)

Reasons for this seems to mainly have revolved around the game-play, with a few mentioning the way the reading was done. One player having a lot of lag, while others noticed smaller details such as models, animations and more. They would also get confused in terms of direction, most commonly when looking for the seven rings and the sisters in the first scene.

In the interviews we also noted, that most stated when asked, that they did not feel particularly bored.

Participant NR.	Answer
ID: 01	*No, I think that there was a fine balance between the actions you had to take and the amount of text you had to read.*
ID: 02	*No, not at all*
ID: 03	*No, I do not feel like I did. The only time where I felt like I was bored was when the witch had to go through like 12 dialogue boxes to explain the situation. I felt like this part could have been shortened down in one bite instead, and maybe highlighted the most important parts, then I would have been fine with it. The text speed is maybe a bit slower than my regular reading speed so that could have also been adjusted as well.*
ID: 04	*I do not think it was more boredom, but it was more like exited to see what was going on, and what do I need these things for exactly. Maybe towards the ending I felt a bit cheated and was like "oh, well. . . ".*
ID: 05	*No, I felt that there was a nice transition between the different scenes themselves. So, like you play the game, then you read the story and so forth.*
ID: 06	*No, I would not say that. It was a different way to tell the story.*

Table 2: Participant comments taken form the interview (transcript found appendix at section 7.5), within the game

(For full transcript see appendix section 7.2.)

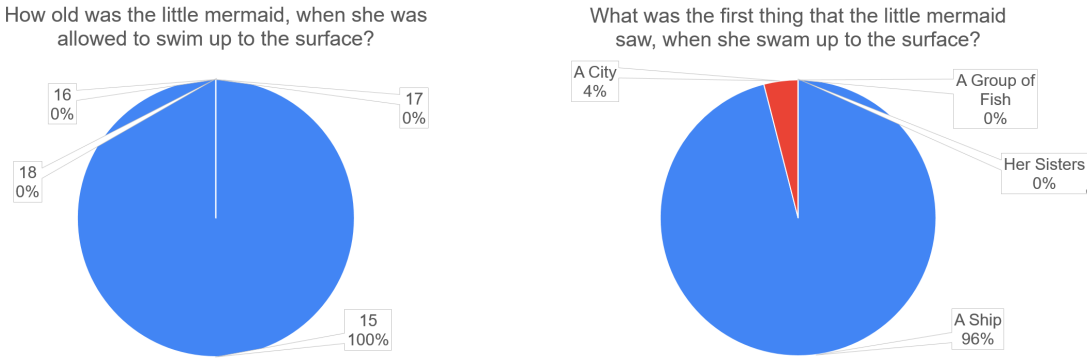
However, this also shows some inconsistency. They all mention elements that caused them to be ripped out of the game, due to a bad design decision form the teams part. This could be due to each participant being recorded, and so not wanting to be to forward. Or perhaps their definition of distracted actually differed from the ours.

5.2 Story engagement

The engagement for the story fell as seen in figure 5. However, this may not have been due to the story itself. In the interview, it was mentioned often, that the story differing from the Disney version made it interesting.

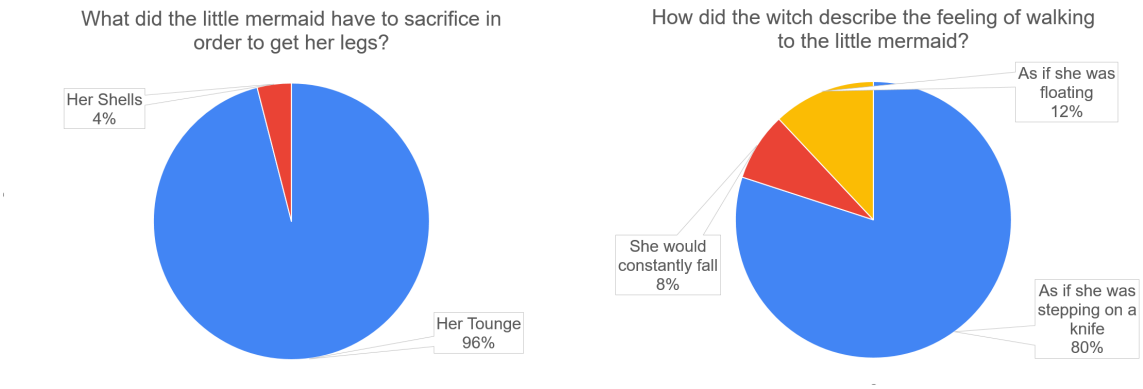
Participant NR.	Answer
ID: 01	*I thought that it was interesting. It is a version that I have not heard before, because I only know the Disney version but not H.C. Anderson's version aka. The original. It was more interesting to see how the original version played out instead of the Disney version.*
ID: 02	*I've always thought it was a cool story, or a nice story, but I know most about Disney's version of it. But I still want to learn the original story because it tells more. It is just as interesting if not more interesting.*
ID: 03	*I like the story... At the end of the game, I got the decision to kill myself or go up and stab the prince in the back and I was ready for stab him. Because that is not how the story goes with the Disney version.*
ID: 05	*No, I have not. The H.C. Andersen that is. It was interesting to get to know that version. I have read some of the others and then seen Disney's version. So, there was a big difference.*

(For full transcript see appendix section 7.2.)
This co-aligns with our discovery while researching reading engagement (see section 2.3, that new information can make a reader more engaged. To further look into their engagement in the story, we looked into their ability to remember story beats. Majority of the participants answered all the questions correct, while a few answered wrong.



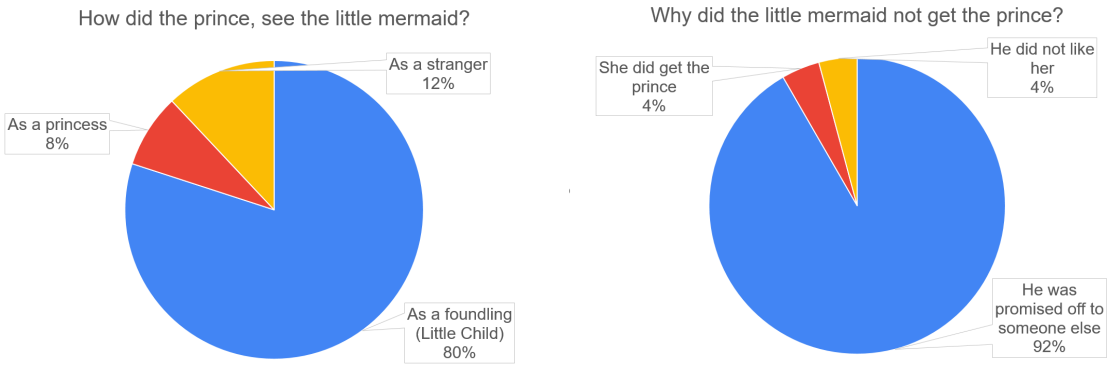
(a) (b)

Figure 7: Questions pertaining to the first level in the game. Scale 1 - 5. (n = 25)



(a) (b)

Figure 8: Questions pertaining to the second level in the game. Scale 1 - 5. (n = 25)



(a) (b)

Figure 9: Questions pertaining to the third level in the game. Scale 1 - 5. (n = 25)

The questions were designed to be increasingly difficult, to test if they really read the story, or simply skimmed parts of it. Here we see that only the age of the little mermaid was answered fully right. Both question following questions ("What was the first thing that the little mermaid saw, when she swam up to the surface" and "What did the little mermaid have to sacrifice in order to get her legs") have a single answer wring. 1 out of 25, means that overall, most seem to have read thoroughly enough. Overall for all the questions we see that a vast majority always answered right, meaning most participants did most likely read the story properly enough. For others the issue may have lied within the amount of text, as seen when ID: 01 in table 2, or simply in the game.

This alongside the interview showed that the vast majority of the participants had an understanding of the story. Which would suggest that they were engaged enough in the story, that they remembered details from what they read. The engagement could have come from the game. However, in the interview and comments from the survey, we saw that the majority of participants found many issues with the game, and talked positively about the story (see table 3 and 4).

Spearman correlation coefficient was then used to compare the relationship between game engagement to story engagement, as it was important to know if their engagement from reading the story, was a

side effect of the engagement derived from the game.

Samples	r
Mid-Game/Story 01	0,64
Mid-Game/Story 02	0.50

It showed a significant enough correlation between the two, meaning that as the engagement for the game grew, so did the engagement for the story. This could indicate that engagement for the story comes from the game. However, it could also mean that the question was not clear enough, as to what was meant specifically (the gameplay and not the game itself). This was a possibility as some comments and responses mention that the game itself was the biggest problems with few talking about the reading aspect. Participants in the interview also mentioned the story being the key reason for completing the game.

ID	Section	Comment
ID: 10	Mid-Game 01	I thought that it was hard to figure out, how to progress. It felt like there was some lag throughout the game, it confused me a bit. Spelling mistakes her and there. Music was tip top and fit in well. I also liked the story, and i am a bit excited as to what will happen next.
ID: 25	Mid-Game 01	I like this version of the little mermaid. Maybe more text for the five sisters, and for the quest counter to work
ID: 03	Mid-Game 01	I think the story is ok, but collecting oysters and rings? I think something more fun could be here instead.
ID: 08	Mid-Game 02	A bit monotonous, maybe more challenging / hidden - darker / blurry.
ID: 04	Mid-Game 02	Why do I have to swim through rings and collect skulls? Seems like a very divisive focus when it comes to having game-play and telling a story. I also swam through 14/7 rings.
ID: 01	Mid-Game 02	Found a bug where the rings end with 14/7 instead of 7/7. The witch may have a little too much text.

Table 3: Participant comments on the quests within the game

Participant	Comment
ID: 01	Probably the story and figuring out what was happening
ID: 02	As I have also written in one of the parts, I think that it is a very engrossing game. Where I actually want to know what happens next. And that made me follow along a bit more. So I hope that I followed along properly, and got the answers correct.
ID: 03	Mainly to see how the story goes really. I wanted to see how this version played out as I only know about the Disney version.
ID: 05	I was interested and curious as to how the story was developing. It was also because I knew that most H.C. Andersen's stories do that have a happy ending and wanted to know the original ending.
ID: 06	Probably the story itself.

Table 4: Participant comments regarding the story

This suggested that although the game itself engaged the player, it may not have engaged them to read more. However, from the comments in table 3 we could see that although the Spearman correlation

coefficient test suggests that as the game engagement went up so did the story. However, it can also be interpreted that the question simply was not clear enough, as the comments did not reflect the results from the treatment. They would suggest that the game-play was the major issue and the story was the main draw. Perhaps the participants assumed that the question "I would like to continue playing the game?" meant the entire game and not only the game-play. Another example of this was how the comments from 3 were from Mid-Game 02, and could therefore also be the reason why the engagement falls after being increased.
For full transcript on both interview and questionnaire, see appendix section 7.2 and 7.5

6 Discussion and Conclusion

This project was carried out during covid-19, and the subsequent lockdown, therefore the project changed a bit. Testing was initially planned to have only 2 levels with the last part being read by the participant. Throughout the test, one of the observers were meant to slightly alter the room, in order to test if the participants were engaged. If the participant did not notice the change, then they were engage enough in the text. The change was never fully planed, it could have been a simple card being held up, every now and then. Or changing light conditions etc. We had not taken into account other students exam-times, so most possible participants reported their own exam being to close for them to find testing a worthwhile activity at the moment. Other altercations also arose; one member worked at a hospital and another was a exchange student who was in their home country at the time. Therefore times to meet up and work distribution became a harder task to complete.

The data does show a potential to such, as their engagement went up primarily due to the story. The fall in engagement could have been the game and not so much the reading itself, but could also be due to the game telling most of the story, and the participants largely finding that they know the main parts of it, perhaps causing them to find reading the remaining text unnecessary. More substantial work would have to be done to get a more concise answer as to whether it can engage them to read books more. The serious game design should be further iterated, so the game does not cause a fall in engagement Furthermore, the experimentation would have to involve actually reading part of the text, in order to measure their engagement in reading after having played the game, otherwise only assumptions can be made. The test procedure was not optimal for the purpose, however, the chosen method was suitable. Recording the users, and being in a call with them over an app such as Discord or Zoom, could have made them more nervous or held-back. However, it was a way to observe their behaviour in game after the test, to see if anything peculiar would be discovered. Looking over the comments, it also seems most were honest and direct with their feedback for the most part, but the questions regarding distractions have some contradicting elements (5.1). Measuring the participants engagement in reading the remaining text was a challenge, as reading is becoming less and less prevalent (see introduction 1). As well as having to measure weather the engagement is connected to the game or only the story, in our case, it does seem to be somewhat connected to the game. There are also different types of reading engagement (2.3), so differing methods should be tested to find the most optimal way to increase engagement. Another important aspect is how long they are engaged for, if their engagement simply last for the test, then the participant is only engaged in reading the text provided. Since we made a game to give new information to the reader, we also have the problem of controlling participants expectations, as they are likely to see the game, and expect certain things, like more game-play than reading. Overall increasing reading engagement was a difficult task, and our method is a primitive version which can still be developed upon allot.

Our data suggests that it could be possible to engage people more in reading, by introducing new information (see reading engagement 2.3)., to make the participants more engaged in the story of the little mermaid. The little mermaid was used as it is a story, where mostly iterations of it is known, and the original is more obscure, but that more work needed to be done and more areas can be explored.

References

Bizzocchi, Jim (2007). "Games and narrative: An analytical framework". In: *Loading-The Journal of the Canadian Games Studies Association* 1.1, pp. 5–10.

Bjørner, Thomas (2015). *Qualitative methods for Consumer Research*. Henning Persson. ISBN: 9788741258539. URL: https://hansreitzel.dk/products/qualitative-methods-for-consumer-research-bog-36789-9788741258539.

Brozo, William G, Gerry Shiel, and Keith Topping (2007). "Engagement in reading: Lessons learned from three PISA countries". In: *Journal of Adolescent & Adult Literacy* 51.4, pp. 304–315.

Cgtrader (2020). URL: https://www.cgtrader.com.

Chapman, P., S. Selvarajah, and Jane Webster (Feb. 1999). "Engagement in multimedia training systems". In: 9 pp. ISBN: 0-7695-0001-3. DOI: 10.1109/HICSS.1999.772808.

Charsky, Dennis (Mar. 2010). "From Edutainment to Serious Games: A Change in the Use of Game Characteristics". In: *Games and Culture - Game Cult* 5. DOI: 10.1177/1555412009354727.

CO.AG Music (2020). URL: https://www.youtube.com/channel/UCcavSftXHgxLBWwLDm_bNvA/about.

Cox, Kathleen E and John T Guthrie (2001). "Motivational and cognitive contributions to students' amount of reading". In: *Contemporary educational psychology* 26.1, pp. 116–131.

de Freitas, Sara (2006). *Learning in immersive worlds: a review of game-based learning*. English. WorkingPaper.

Djaouti, Damien, Julian Alvarez, and Jean-Pierre Jessel (Jan. 2011). "Classifying Serious Games: the G/P/S model". In: *Handbook of Research on Improving Learning and Motivation through Educational Games: Multidisciplinary Approaches*. DOI: 10.4018/978-1-60960-495-0.ch006.

Dubbles, Brock (2009). "Video games, reading, and transmedial comprehension". In: *Handbook of research on effective electronic gaming in education*. IGI Global, pp. 251–276.

Free3D (2020). URL: https://free3d.com.

Fullerton, Tracy (Jan. 2008). *Game Design Workshop. A Playcentric Approach to Creating Innovative Games*. ISBN: 9780240809748. DOI: 10.1201/b22309.

Guthrie, John T, Allan Wigfield, and Wei You (2012). "Instructional contexts for engagement and achievement in reading". In: *Handbook of research on student engagement*. Springer, pp. 601–634.

Hamari, Juho et al. (Aug. 2016). "Challenging games help students learn: An empirical study on engagement, flow and immersion in game-based learning". In: *Computers in Human Behavior*. DOI: 10.1016/j.chb.2015.07.045.

Hassenzahl, Marc and Noam Tractinsky (Mar. 2006). "User experience - A research agenda". In: *Behaviour and Information Technology* 25, pp. 91–97. DOI: 10.1080/01449290500330331.

Jacques, Richard, Jennifer Preece, and Tom Carey (Jan. 1995). "Engagement as a Design Concept for Multimedia". In: *Canadian Journal of Educational Communication* 24, pp. 49–59. DOI: 10.21432/T2VG77.

Laurel, Brenda (Jan. 1993). "Computers as Theatre". In: vol. 37. DOI: 10.2307/1146303.

Michael, David and Sandra Chen (Jan. 2006). "Serious Games: Games That Educate, Train, and Inform". In:

Mills, Caitlin et al. (July 2013). "What Makes Learning Fun? Exploring the Influence of Choice and Difficulty on Mind Wandering and Engagement during Learning". In: vol. 7926, pp. 71–80. DOI: 10.1007/978-3-642-39112-5_8.

Naumann, Johannes (2015). "A model of online reading engagement: Linking engagement, navigation, and performance in digital reading". In: *Computers in Human Behavior* 53, pp. 263–277.

Nørtoft, Magnus (2019). *Vi elsker krimier - uagtet alder, uddannelser og køn*. Statistics of danish people reading novels, last visited: 24-02-2020. URL: https://www.dst.dk/da/Statistik/bagtal/2019/2019-07-23-Vi-elsker-krimier-uagtet-alder-uddannelser-og-koen#.

O'Brien, Heather and Elaine Toms (Apr. 2008). "What is User Engagement? A Conceptual Framework for Defining User Engagement with Technology". In: *JASIST* 59, pp. 938–955. DOI: 10.1002/asi.20801.

Pasalic, Ademir et al. (2017). "How to Increase Boys' Engagement in Reading Mandatory Poems in the Gymnasium: Homer's "The Odyssey" as Transmedia Storytelling with the Cyclopeia Narrative as a Computer Game". In: *International Conference on Smart Objects and Technologies for Social Good*. Springer, pp. 216–225.

Pourabdollahian, Borzoo, Marco Taisch, and Endris Kerga (Oct. 2012). "Serious Games in Manufacturing Education: Evaluation of Learners' Engagement". In: *Procedia Computer Science* 15, pp. 256–265. DOI: 10.1016/j.procs.2012.10.077.

Reid, Leslie (Dec. 2012). "Redesigning a Large Lecture Course for Student Engagement: Process and Outcomes". In: *The Canadian Journal for the Scholarship of Teaching and Learning* 3, pp. 1–39. DOI: 10.5206/cjsotl-rcacea.2012.2.5.

Respondek, Lisa et al. (July 2019). "Linking changes in perceived academic control to university dropout and university grades: A longitudinal approach." In: *Journal of Educational Psychology*. DOI: 10.1037/edu0000388.

Schoenau-Fog, Henrik and Thomas Bjørner (Oct. 2012). ""Sure, I Would Like to Continue" A Method for Mapping the Experience of Engagement in Video Games". In: *Bulletin of Science, Technology & Society* 32, pp. 405–412. DOI: 10.1177/0270467612469068.

Schoenau-Fog, Henrik, Sandy Louchart, et al. (2013). "Narrative engagement in games-A continuation desire perspective." In: *FDG*, pp. 384–387.

Slota, Stephen et al. (Aug. 2015). "Game Narrative, Interactive Fiction, and Storytelling: Creating a "Time for Telling" in the Classroom". In: pp. 199–222. ISBN: 0128014628. DOI: 10.1016/B978-0-12-801462-2.00010-2.

Solheim, Oddny Judith and Kjersti Lundetræ (2018). "Can test construction account for varying gender differences in international reading achievement tests of children, adolescents and young adults?–A study based on Nordic results in PIRLS, PISA and PIAAC". In: *Assessment in Education: Principles, Policy & Practice* 25.1, pp. 107–126.

Turbosquid (2020). URL: https://www.turbosquid.com.

Twenge, Jean, Gabrielle Martin, and Brian Spitzberg (Aug. 2018). "Trends in U.S. Adolescents' Media Use, 1976-2016: The Rise of Digital Media, the Decline of TV, and the (Near) Demise of Print". In: *Psychology of Popular Media Culture*. DOI: 10.1037/ppm0000203.

Unity Asset Store (2020). URL: https://assetstore.unity.com.

Vanco, Bryan M and John L Christensen (2016). "Ego depletion increases regulatory success in educational digital media environments". In: *Computers in Human Behavior* 62, pp. 602–612.

Wouters, Pieter, Huib Tabbers, and Fred Paas (Sept. 2007). "Interactivity in Video-based Models". In: *Educational Psychology Review* 19, pp. 327–342. DOI: 10.1007/s10648-007-9045-4.

Zyda, Michael (Oct. 2005). "From Visual Simulation to Virtual Reality to Games". In: *Computer* 38, pp. 25–32. DOI: 10.1109/MC.2005.297.

List of Figures

7 Appendices

7.1 Interview questions

What did you think of the little mermaid story you just played?
- Ask more if needed

What do you remember about the game's story?
- Ask more if needed

Did you enjoy reading the story?
- Ask more if needed

Did you enjoy playing the game?
- Ask more if needed

Did you want to keep going with the story?
- Ask more if needed

Did you feel bored at any point during the story/game?
- Ask more if needed

Why made you complete the game?
- Ask more if needed

7.2 Transcripts from interviews

These are translated from Danish to English and some phrases may sound weird due to some expression being originally Danish.

Interview with participant ID: 01

Interviewer: So, I have a couple of questions for you

Participant: Yea

Interviewer: What did you think of the little mermaid story you just played?

Participant: I thought that it was interesting. It is a version that I have not heard before, because I only know the Disney version but not H.C. Anderson's version aka. The original. It was more interesting to see how the original version played out instead of the Disney version. Interviewer: Did you like it?

Participant: Yes. To that extent.

Interviewer: Okay. What do you remember about the game's story?

Participant: There was a little mermaid, who had 5 sisters, and was a princess. She had a father who was the king. She wishes to swim to the surface when she turns 15 years old. She sees a ship with the prince on it. She saves his life, swims down and starts developing feelings for him. Against other's advice, she went over and started talking with the witch. She must sacrifice something to gain legs. Once she has legs and meets the legs, she is not satisfied with their relationship since he is promised

to another princess. We conclude with a dilemma or an ultimatum where she must choose either the little mermaid or the prince dies.

Interviewer: What did the little mermaid choose?

Participant: I do not know.

Interviewer: Alright, let us continue with the next question then. You said that you like the story, what did you think about the game itself?

Participant: Uhm, you can say that it is a prototype, but I think that it was working fine. There was not much focus on the details on the visuals probably because it must function and not look pretty. Some tones in the music that was chosen was a bit too high. That was probably what weighed in the must in my evaluation was the sound itself. Otherwise I think it was a fine unity project that was created. Interviewer: Did you feel bored at any point during the story/game?

Participant: No, I think that there was a fine balance between the actions you had to take and the amount of text you had to read.

Interviewer: What made you complete the game?

Participant: Probably the story and figuring out what was happening.

Interview with participant ID: 02

Interviewer: What did you think of the little mermaid story you just played?

Participant: I've always thought it was a cool story, or a nice story, but I have mostly known about Disney's version of it. But I still want to learn the original story because it tells the more. It is just as interesting though not more interesting.

Interviewer: Do you remember parts of the story, or; What elements do you remember most from this part?

Participant: I mean, it was more violent, more realistic in a way, and it is a complete different set of emotions, the witch describes for the little mermaid.

Interviewer: So you liked the story in it of itself, what did you think of the game, in it of itself?

Participant: I feel that the game was well made. It was a nice way of going through the story, and learn more about it, at least I think so.

Interviewer: And would you be interested in continuing to read the story, cause the game itself is only telling fragments of the story, to make sure it stays fluid?

Participant: Yes, I would not mind. I would be fine with, changeably, read and play to get through.

Interviewer: So you did not feel bored throughout the game or the story?

Participant: No, not at all.

Interviewer: What made you complete the game, excluding that this is a test?

Participant: As I have also written in one of the parts, I think that it is a very engrossing game. Where I actually want to know what happens next. And that made me follow along a bit more. So I hope that I followed along properly, and got the answers correct.

Interview with participant ID: 03

Interviewer: What did you think of the little mermaid story you just played?

Participant: I like the story. It is a classic. I like how it was presented. In a game, I play a character and I feel that I constantly have decisions to make, and I am a spoiled computer gamer who can decide on these decisions when I play a game. At the end of the game, I got the decision to kill myself or go up and stab the prince in the back and I was ready for stab him. Because that is not how the story goes with the Disney version. I really wanted to play that part out. There was at some point where I was taken out of the story. Like collecting rings, skulls and oysters. It did not really fit the story itself. Like if I were collecting 4 different parts to make the potion then I would really like that those items were part of the story as well. Overall, I really liked how the story was presented.

Interviewer: What do you remember about the game's story?

Participant: I remember most of the things. When you think about the little mermaid, then you would think about Disney's version of the story and this version were more brutal and realistic. There were some elements that were touched upon, but it was not bad elements as it fit the story itself even more. I got more out of playing this version instead of Disney's version. So, it was nice to be fooled a bit.

Interviewer: Did you enjoy playing the game?

Participant: I did like it, as it was nice to get another angle or perspective on the story through a game instead of just being pure text based. But once again, you were presented with some decisions while playing the main character, which I would really have liked to do. Like going around collecting rings and so forth, I would really have liked more "in universe" elements in the actual game and why I need to do this part. I felt like it was a bit separated from the main story.

Interviewer: Did you feel bored at any point during the story/game?

Participant: No, I do not feel like I did. The only time where I felt like I was bored was when the witch had to go through like 12 dialogue boxes to explain the situation. I felt like this part could have been shortened down in one bite instead, and maybe highlighted the most important parts, then I would have been fine with it. The text speed is maybe a bit slower than my regular reading speed so that could have also been adjusted as well.

Interviewer: Why made you complete the game?

Participant: Mainly to see how the story goes really. I wanted to see how this version played out as I only know about the Disney version.

Interview with participant ID: 04

Interviewer: What did you think of the little mermaid story you just played?

Participant: I would say that I was captured at the beginning by the story, but I quickly lost interest and did not know what I had to do after a while. Not what I had to do but what kind of information was missing e.g. the dialogue from the 5 sisters might have contained something for me to watch out for. I wanted to know more to put it in simple terms.

Interviewer: What do you remember about the game's story?

Participant: I thought it was a good story, but I felt that there was something missing to make me feel connected. Like what could I have used this information and why should I know to continue with the story. So, I felt a bit of disconnect with the game and story itself e.g. that the sisters all had information which I could have potentially used while I was going off somewhere to collect rings and oysters. So more of the why are the dialogues in the game and how can I use it.

Interviewer: Did you feel bored at any point during the story/game?

Participant: I do not think it was more boredom, but it was more like exited to see what was going on, and what do I need these things for exactly. Maybe towards the ending I felt a bit cheated and was like "oh, well...".

Interviewer: Looking at your answers, it seems that you were more visually oriented towards the story than the dialogue itself. Is this true?

Participant: Yeah, there might be some truth to that. Maybe because there was a lot of text and was very streamlined. I think it was more that I was missing a focus to read the text, maybe a bit visual clues to the story itself.

Interview with participant ID: 05

Interviewer: What did you think of the little mermaid story you just played?

Participant: Yeah, it was very interesting. I liked it.

Interviewer: Have you read the original?

Participant: No, I have not. The H.C. Andersen that is. It was interesting to get to know that version. I have read some of the others and then seen Disney's version. So, there was a big difference.

Interviewer: Did you enjoy reading and playing the story/game?

Participant: Yeah, there was a bit of lag in the beginning but might have been from my side. It was nice, and it would have been nice of the counters worked when I was collecting the rings etc. and I did not feel that it was intuitive that the prince was by the castle. Otherwise, it was nice.

Interviewer: Did you feel bored at any point during the story/game?

Participant: No, I felt that there was a nice transition between the different scenes themselves. So, like you play the game, then you read the story and so forth.

Interviewer: Well, have you seen the original Disney's version of the story? What did you think about this version itself?

Participant: I liked this version better where the little mermaid was mute, and the prince had to marry another person. H.C. Andersen could his shit.

Interviewer: Why made you complete the game?

Participant: I was interested and curious as to how the story was developing. It was also because I knew that most H.C. Andersen's stories do that have a happy ending and wanted to know the original ending.

Interviewer: And were you surprised by the ending? Participant: A little, I guess. I was also very interested as I saw that the sisters wanted the prince dead.

Interview with participant ID: 06

Interviewer: What did you think of the little mermaid story you just played?

Participant: Well, it is a classical tale. It is probably not something I would read in my spare time. Almost everyone knows about it.

Interviewer: The original or?

Participant: Okay, not everyone knows about the original but probably Disney's version of it. I would probably think that there are a lot of Danish people who knows the original.

Interviewer: Did you enjoy reading and playing the story/game?

Participant: It was fine, but there were some small issues like the how far you have progressed in collecting the rings etc.

Interviewer: Did you feel bored at any point during the story/game?

Participant: No, I would not say that. It was a different way to tell the story.

Interviewer: Why made you complete the game?

Participant: Probably the story itself.

7.3 Guide

Hey, vi er så taknemlige for at du tager tid ud fra din dag, til at hjælpe os med vores bachelor projekt.

Herunder finder du links til vores spil og Spørgeskema, Samt en 7-step guide til hvad du skal gøre.

Spil:

https://theintroverts.itch.io/lille-havfrue

Spøgeskema:
https://docs.google.com/forms/d/e/1FAIpQLSdZE88ORzulAAHKumDCLk31_7nkvePrrYlN7PWZF_tai786og/viewform

Guide:

- 1:
  - Start med at svare på spøgeskemaet indtil du når ” Mid-Game 01”. Uden at svare på denne sektion (endnu), skal du herefter gå ind og starte spillet.
- 2:
  - Spil indtil spillet fortæller dig at du skal skifte til ” Mid-Game 01”.
- 3:
  - Skift og svar på ” Mid-Game 01” indtil du når ” Mid-Game 02”. uden at svare på denne sektion (endnu) skift tilbage og fortsæt med spillet.
- 4:
  - Spil indtil spillet fortæller dig at du skal skifte til ” Mid-Game 02”.
- 5:
  - Skift og svar på ” Mid-Game 02” indtil du når ” Efter spillet”. uden at svare på denne sektion (endnu) skift tilbage og fortsæt med spillet.
- 6:

  Spil resten af spillet færdigt. Skift derefter tilbage til spøgeskemaet.
- 7:

  Svar på resten af spøgeskemaet (”Efter spillet”).

7.4 Questionaire

Den Lille Havfrue

Du skal til at spille et spil, som er en del af en undersøgelse omhandlende spil design. Testen involvere en prototype, lavet af en gruppe af studerende, fra det sjette semester hos Ålborg universitet i København. Spillet handler om den lille havfrue.
*Skal udfyldes

1. Jeg giver samtykke ved at udfylde dette spørgeskema, at gruppen 605, 6 Semester Medialogi på vegne af Aalborg Universitet kan analysere og lagre data i to år. Deltagelse er frivillig, og du er fri til at trække dig tilbage til enhver tid eller ophøre deltagelse på ethvert tidspunkt. Du vil forblive komplet anonym, og dataene er kun beregnet til universitets- og forskningsformål. *

   Markér kun ét felt.

   - ◯ Ja
   - ◯ Nej

2. Hvor mange timer per uge bruger du på at spille video spil? *

   Markér kun ét felt.

   - ◯ Overhovdet ikke
   - ◯ 1 - 10 timer
   - ◯ 11 - 20 timer
   - ◯ 21 - 30 timer
   - ◯ over 30 timer

3. Hvor mange noveller(romaner) har du læst inden for de sidste 3 måneder? *

Markér kun ét felt.

- Ingen
- 1
- 2
- 3
- 4
- 5

Mid-Game 01

Vær så venlig, og kun svar på disse spørgsmål når en fra vores personale siger til.

4. Jeg vil gerne fortsætte med at spille spillet? *

Markér kun ét felt pr. række.

	Meget Uenig	Uenig	Neutral	Enig	Meget Enig
Er du enig?					

5. Jeg er interesseret I at vide hvad der sker i historien herefter? *

Markér kun ét felt pr. række.

	Meget Uenig	Uenig	Neutral	Enig	Meget Enig
Er du enig?					

6. Har du nogle kommentere, omhandlende den oplevelse du lige er gået igennem? *

Mid-Game 02

Vær så venlig, og kun svar på disse spørgsmål når en fra vores personale siger til.

7. Jeg vil gerne fortsætte med at spille spillet? *

Markér kun ét felt pr. række.

	Meget Uenig	Uenig	Neutral	Enig	Meget Enig
Er du enig?	◯	◯	◯	◯	◯

8. Jeg er interesseret I at vide hvad der sker i historien herefter? *

Markér kun ét felt pr. række.

	Meget Uenig	Uenig	Neutral	Enig	Meget Enig
Er du enig?	◯	◯	◯	◯	◯

9. Har du nogle kommentere, omhandlende den oplevelse du lige er gået igennem? *

Efter spillet

Vær så venlig, og kun svar på disse spørgsmål når en fra vores personale siger til.

10. Hvor gammel var den lille havfrue, da hun havde tilladelse til at svømme op til overfladen? *

Markér kun ét felt.

- 15
- 16
- 17
- 18

11. Hvad var det første den lille havfrue så, da hun svømmede op over overfladen? *

Markér kun ét felt.

- Et Skib
- En by
- En gruppe af fisk
- Hendes søskende

12. Hvad skulle den lille havfrue give op, for at kunne få hendes ben? *

Markér kun ét felt.

- Hendes tunge
- Hendes hår
- Hendes skaller
- Hendes søstre

13. Hvordan beskrev heksen følelsen af at gå til den lille havfrue? *

Markér kun ét felt.

- Som at gå på en skarp kniv
- Som om hun var svævende
- Der ville ikke være nogen forskel
- Hun ville helle tiden falde

14. Hvordan så prinsen den lille havfrue? *

Markér kun ét felt.

- Som et hittebarn (lille barn)
- Som en princesse
- Som en fremmede
- Som en fisk

15. Hvorfor kunne den lille havfrue ikke få prinsen? *

Markér kun ét felt.

- Han var forlovet til en anden
- Han kunne ikke lide hende
- Hun fik prinsen
- Hun elskede ham ikke længere

16. Jeg vil gerne læse resten af teksten, for at finde ud af hvad der sker derefter? *

Markér kun ét felt pr. række.

	Meget Uenig	Uenig	Neutral	Enig	Meget Enig
Er du enig?					

17. Jeg blev let distraheret, mens jeg fulgte historien? *

Markér kun ét felt pr. række.

	Meget Uenig	Uenig	Neutral	Enig	Meget Enig
Er du enig?					

18. Jeg tænkte på indholdet, som jeg læste fra spillet? *

Markér kun ét felt pr. række.

	Meget Uenig	Uenig	Neutral	Enig	Meget Enig
Er du enig?	◯	◯	◯	◯	◯

7.5 Questionaire Comments

All the following text, is transcribed in Danish.

Mid-Game 01

ID 01: Vil gerne kunne komme ind i husene og se hvad hvem der bor der. Grafikken er okay for en "prototype".

ID 02: Føler der mangler feedback i forhold til progress af samtale med søstrene og missionen hvor man skal samle østers og svømme igennem ringe. Mere specifikt, hvor mange mangler jeg?

ID 03: Jeg synes at historien er okay, men at samle østers og ringe? Der kunne godt være noget mere sjovt.

ID 04: Animations reagerer meget langsomt til input, base rotation på animationerne er off så man ikke visuelt svømmer i den retning man vil, det er uklart hvor man skal hen eller hvor langt man er med sine objectives.

ID 05: Det er et godt spil indtil videre.

ID 06: Synes at grafikken kunne være bedre. Der kunne godt være en counter når man samler ringe og østers, ellers var gameplay fint nok.

ID 07: Virkelig lidt information fra søstrene. Kunne godt bruge lidt mere tekst.

ID 08: Meget intuitivt, simpelt - brugervenligt. Sød grafik, meditativ stemning. Fængende storyline

ID 09: Mangler lidt indhold

ID 10: Jeg syntes det var svært at finde ud af hvordan jeg kom videre. Det virkede også som om at der var lidt lag undervejs, det forvirrede mig lidt. Stavefejl hist og her. Musikken er i top og passer godt. Jeg kan også godt lide historien og er en smule spændt på hvad der sker nu.

ID 11: Kunne ha' bedre grafik

ID 12: Nice GUI

ID 13: Super flot lavet, dog følger din Mission box ikke helt med nogen gange (out of sync) fine mekanismer med svømning osv. den havfrue man selv er svømmer skævt xD, men igen flot lavet

ID 14: Det ville være en god ide at have de knapper man kan bruge til at styre, inden i spillet istedet for uden for spillet især når i anbefaler at det skal være fullskærm. Kameraret går igennem diverse objecter også jorden, kunne se en grøn have eller noget i den stil under jorden

ID 15: Stavefejl og lidt svært at finde ud af hvor langt man er/hvad man skal i delen med ringene og østers. Ellers godt

ID 16: Kender kun til disneys version of historien, så jeg er interesseret i at se hvad forskellen er.

ID 17: Synes at det var okay

ID 18: Counts på ringe :) imens jeg samler dem

ID 19: Synes godt at quest counteren kunne ha' virket for denne her version, men gennemførte det stadigvæk.

ID 20: Kan godt se hvorfor nogle mennesker vil kunne lide det her men det er ikke noget for mig

ID 21: Der var ikke nogen counter for ringene eller østers. Farten på teksten er lidt langsom, og musik var okay.

ID 22: Det hakker lidt - det er svært at følge med i hvor langt jeg er i en given opgave

ID 23: Jeg syntes det er godt lavet. Jeg hakker en smule, jeg er ikke sikker på om det er på grund af mit netværk.

ID 24: Det er åbenlys bare hc andersens history derfor er jeg netural med historien da jeg kender den. selve spillet har nogle meget under lige controls.

ID 25: Kan godt lide denne version af den lille havfrue. Måske mere tekst fra de fem søstre og at quest counter virker

Mid-Game 02

ID 01: Fandt en bug hvor ringene slutter med 14/7 i stedet for 7/7. Heksen har måske lidt for meget tekst.

ID 02: Der er overlap i mission progress. Ringene fra tidligere mission bliver en del af den nye mission, så det ender med 12/7 ringe eller sådan. Desuden er det svært at se om man faktisk har ramt igennem ringene. Nogle gange skulle jeg vende mig om for at se om den stadig var der. Når man svømmer vibrerer skærmen underligt. Det "flicker" lidt i bevægelsen.

ID 03: Samme som før. Samle ringe og noget andet objekt. Vil gerne have noget mere sjovt. Synes også at heksen havde lidt for meget tekst at læse igennem

ID 04: Hvorfor skal jeg svømme igennem ringe og samle kranier? Virker som et meget splittet fokus i forhold til at have gameplay og fortælle en historie. Jeg svømmede også igennem 14/7 ringe

ID 05: Meget medrivende.

ID 06: Oh, der var en counter for når man samler objekter, men det viser dog at jeg samlede 14/7. Jeg synes dog at heksen har lidt for meget tekst at læse igennem.

ID 07: Synes at heksen havde lidt for meget tekst at læse igennem.

ID 08: Lidt ensformigt, måske mere udfordrende/gemt - mørkere/sløret.

ID 09: Samme

ID 10: Musikken passede godt, den gjorde mig lettere urolig indeni, som om jeg laver en quest jeg "ikke må". Jeg oplevede i bug da jeg svømmede op til den sidste ring, så blev min position reset til der hvor scenen startede. Den lille havfrues svømme animation er også lidt til venstre, når jeg svømmer ligeud.

ID 11: Der var en bug med at samle ringene.

ID 12: Høj lyd. Ret høj hastighed ved ringene, måske lidt for højt.

ID 13: det er nogle oversize kranier xD ellers ganske fint, måske lidt for meget historie, som man ikke magter læse TBH. noget video ville have være lidt federe, men kræver også meget mere iknow ;)

ID 14: Jeg fandt en bug under fetch questen, den sagde jeg allerede havde 7/7 ringe, det er nok fra det tideligere level, men var alligevel nød til at samle 7 nye, for at kunne forsætte, problemet var dog at efter jeg havde samlet dødninge hovederne forsvandt quest objecterne GUI'en så vidste ikke lige hvornår jeg havde samlet dem alle

ID 15: Igen, lidt problemer med at se hvor langt man er med "missionen". Der stod jeg fra start havde 7/7 ringe. Ellers fint igen

ID 16: Måske lidt for meget tekst for heksen, men ellers jeg interesseret i hvad der sker bagefter.

ID 17: For meget tekst.

ID 18: count på hvor mange ringe :)

ID 19: Ultra meget tekst for heksen! Kunne godt tænke mig at det blev kortere end det

ID 20: Virkelig meget tekst for Heksen

ID 21: Fandt en bug da jeg skulle samle ringene, hvor det sagde at jeg samlede 14/7. Lidt for meget tekst at læse igennem for heksen.

ID 22: Jeg kunne godt tænke mig at man kunne skrue op for tekstens hastighed!

ID 23: Når man svømmer (Ligger ned), så er det ikke altid at den opdager at man bevæger sig gennem en ring. Ellers var det meget godt.

ID 24: spilelt er buggy hvis jeg smale et kranie op så er alle ringe markeret som taget men den vil ikke lade mig fortsætte så tager jeg ringe og bliver tleeporteret når jeg går til bage til heksen hvorefter jeg kan sankke med hende og fortsætte spillet. stadig fortsætter det som hc andersen histore så den er jeg ikke interessert i da jeg allrede kender den

ID 25: Alt for meget tekst. Det var nærmest 2-3 gange hver tekst end den forrige scene.

7.6 Code

7.6.1 Boost

```
public class Boost : MonoBehaviour
{
    public GameObject Play;
    public float boostSpeed;
    public float boostDuration;
    bool pass = false;
    private bool completed = false;
    public MissionController missioncontroller;

    // Start is called before the first frame update
    void Start()
    {
        if (Play == null)
            Play = GameObject.FindWithTag("Player");
    }

    void OnTriggerEnter(Collider other)
    {
        IMoveSpeedBoost<float, float> moveBoost =
            other.GetComponent<IMoveSpeedBoost<float, float>>(); //Get interface for
            movement speed boost
        if (moveBoost != null && this.enabled == true) //Check if the interface is
            not null
        {
            Rings.RingCount += 1; //Update the ring counter
            updateCounter(); //Call the update counter method
            Destroy(gameObject); //Destroy the object
            moveBoost.Speed(boostSpeed, boostDuration); //Run the movement speed
                boost method
            if (this.GetComponent<ProgressionData>()) //Checks if the object has the
                ProgressionData script
            {
                if (!completed) //Check if the ring is not completed
                {
                    FindObjectOfType<Progression>().OnCompletion(); //Set the object
                        as complete in the progression script
                    completed = true; //Sets the ring as completed
                }
            }
        }
    }

    void updateCounter()
    {
        Scene thisScene = SceneManager.GetActiveScene(); //Check the sceen name
        if (thisScene.name == "UnderwaterCastle2")
        {
            string message = "oesters: " + Clams.ClamCount + "/3 Ringe: " +
                Rings.RingCount + "/7"; //Create the string for the ui
            missioncontroller.DoMission(message); //change the string in the ui
            if (Clams.ClamCount == 3 && Rings.RingCount == 7) //Check if all the
                objects have been completed
            {
                missioncontroller.hasFinished(); //Set the mission as finished
            }
        }
        else if (thisScene.name == "WitchsPlace1")
        {
            string message = "Kranier: " + Skulls.SkullCount + "/7 Ringe: " +
                Rings.RingCount + "/7"; //Create the string for the ui
            missioncontroller.DoMission(message); //change the string in the ui
            if (Skulls.SkullCount == 7 && Rings.RingCount == 7) //Check if all the
                objects have been completed
            {
                missioncontroller.hasFinished(); //Set the mission as finished
            }
        }
    }
}
```

7.6.2 Bubble

```
public class Bubble : MonoBehaviour
{
    public float boostSpeed;
    public float boostDuration;

    // Update is called once per frame
    void Update()
    {

    }

    private void OnTriggerEnter(Collider other)
    {
        IMoveSpeedBoost<float, float> moveBoost =
            other.GetComponent<IMoveSpeedBoost<float, float>>(); //Get interface for
            movement speed boost
        if (moveBoost != null) //Check if the interface is not null
        {
            moveBoost.Speed(boostSpeed, boostDuration); //Run the movement speed
                boost method
            Destroy(gameObject); //Destroy the bubble
        }
    }
}
```

7.6.3 CameraLock

```
public class CameraLock : MonoBehaviour
{
    public GameObject player;
    float rotX, rotY;

    private void Start()
    {
        Cursor.lockState = CursorLockMode.Locked;
    }

    // Update is called once per frame
    void LateUpdate ()
    {
        rotX += Input.GetAxis("Mouse X");
        rotY -= Input.GetAxis("Mouse Y");

        player.transform.eulerAngles = new Vector3(0, rotX, 0);
        transform.eulerAngles = new Vector3(rotY, rotX, 0);
    }
}
```

7.6.4 Clams

```
public class Clams : MonoBehaviour
{
    public static int ClamCount = 0;

    // Start is called before the first frame update
    void Start()
    {

    }
}
```

7.6.5 Dialogue

```
[System.Serializable]
public class Dialogue
{
    public string name;
    [TextArea(3, 10)]
    public string[] sentence;
}
```

7.6.6 DialogueM

```
public class DialogueM : MonoBehaviour
{
    private Queue<string> sentence;

    public Text name;
    public Text dialogue;
    public GameObject canvasObject;
    public GameObject playerObject;
    private LandMovement anotherScript;

    public Animator animate;
    public bool completed = true;

    void Start()
    {
        sentence = new Queue<string>();
    }

    public void startDia(Dialogue dialogue)
    {
        animate.SetBool("IsOpen", true);
        name.text = dialogue.name;

        sentence.Clear(); //Clear the sentence queue

        foreach(string sentences in dialogue.sentence) //Go through all the sentences
            in the dialogue
        {
            sentence.Enqueue(sentences); //Add the new sentences to the queue
        }

        DispNextSent(); //Call the DispNextSent method
    }

    public void DispNextSent()
    {
        if(sentence.Count == 0) //Check if no sentences are left
        {
            EndDialogue(); //End the dialogue

            if (!completed) //Check if the dialogue is not completed
            {
                FindObjectOfType<Progression>().OnCompletion();
                completed = true; //Sets the dialogue as completed
            }
            return;
        }

        string sen = sentence.Dequeue(); //Remove the last sentence
        StopAllCoroutines();
        StartCoroutine(typeSent(sen)); //Go to the next sentence
    }

    IEnumerator typeSent (string sentanceDia)
    {
        dialogue.text = "";
        foreach(char letter in sentanceDia.ToCharArray()) //Go through the letters in
            the current sentence
        {
            dialogue.text += letter; //Display the text
            yield return null;
```

```
        }
    }

    public void EndDialogue()
    {
        Activation(); //Call the Activation method
        animate.SetBool("IsOpen", false);
    }

    public void Activation()
    {
        playerObject.GetComponent<LandMovement>().enabled = true; //Enable the
            LandMovement script
        canvasObject.SetActive(false); //Disable the dialogue canvas
        playerObject.GetComponentInChildren<TPCamControl>().enabled = true; //Enable
            the TPCamControl script
        Cursor.lockState = CursorLockMode.Locked; //Make the cursor mode locked to
            the middle of the screen
        Cursor.visible = false; //Make the cursor invisible
    }

    public void GivenDialogueCompletion()
    {
        completed = false;
    }
}
```

7.6.7 DialogueT

```
public class DialogueT : MonoBehaviour, INpcDialogue
{
    public Dialogue dialogue;
    private bool completed = false;
    public MissionController missionController;

    void Start()
    {

    }

    public void triggerDia()
    {
        FindObjectOfType<DialogueM>().startDia(dialogue); //Find the object with the
            DialogueM script and call the startDia method
        if (!completed) //Check if the dialogue is not completed
        {
            FindObjectOfType<DialogueM>().GivenDialogueCompletion(); //Find the
                object with the DialogueM script and call the GivenDialogueCompletion
                method
            completed = true; //Sets the dialogue as completed

            if (missionController != null && this.tag != "Sister") //Set the mission
            {
                missionController.nextMission(); //Call the nextMission method of the
                    missionController
            }
            else if (missionController != null && this.tag == "Sister")
            {
                Sisters.SisterCount += 1; //Update the SisterCounter
                string message = "Soestrer: " + Sisters.SisterCount + "/5"; //Create
                    the string for the ui
                missionController.DoMission(message); //change the string in the ui
                if (Sisters.SisterCount == 5) //Check if all the sisters have been
                    interacted with
                {
                    missionController.hasFinished(); //Set the mission as finished
                }
            }

        }
    }
}
```

7.6.8 EMenu

```
public class EMenu : MonoBehaviour
{
    public void Restart()
    {
        //SceneManager.LoadScene(SceneManager.GetActiveScene().name);
        // Change scene to the game
        SceneManager.LoadScene("StartScreen");
    }

    public void Quit()
    {
        // Exit the game
        Application.Quit();;
    }
}
```

7.6.9 Func

```
public class Func : MonoBehaviour
{
    public void Exit()
    {
        // Exit the game
        Application.Quit();
    }
}
```

7.6.10 IMoveSpeedBoost

```
public interface IMoveSpeedBoost<T, X>
{
    void Speed(T speed, X duration);
}
```

7.6.11 INpcDialogue

```
public interface INpcDialogue
{
    void triggerDia();
}
```

7.6.12 Interact

```
public class Interact : MonoBehaviour
{
    public Interactive focus;

    //This function is going to send triffer events
    private void OnTriggerEnter(Collider col)
    {
        //If the character collides, will interact
        Interactive interactive = col.GetComponent<Interactive>();
        SetFocus(interactive);
    }

    public void SetFocus(Interactive newFocus)
    {
        if (newFocus != focus)
        {
            if (focus != null) focus.OnDefocused();
```

```
            focus = newFocus;
            focus.OnFocused(transform);
        }
    }

    public void RemoveFocus()
    {
        focus.OnDefocused();
        focus = null;
    }
}
```

7.6.13 Interactive

```
public class Interactive : MonoBehaviour
{
    public float radius = 3f;

    bool isFocus = false;
    bool hasInteracted = false;
    Transform player;
    public float distance;
    private GUIStyle guiStyle = new GUIStyle();
    private bool completed = false;

    public MissionController missioncontroller;

    private Component[] children;

    void Start()
    {
        children = GetComponentsInChildren<Renderer>(); //Get the object's childrens' renderer
    }

    void Update()
    {
        if (isFocus && !hasInteracted) //Check if the object is within the focus and has not been interacted with
        {
            distance = Vector3.Distance(player.position, transform.position); //Get the distance to the player

            if (distance <= radius) //Check if the player is within the radius
            {
                if (Input.GetKeyDown(KeyCode.E))
                {
                    hasInteracted = true;
                    Collecting(this.tag); //Call the Collecting method
                    hasCollected(); //Call the hasCollected method
                    OnDefocused(); //Call the OnDefocused method
                    if (this.GetComponent<ProgressionData>()) //Checks if the object has the ProgressionData script
                    {
                        if (!completed) //Check if the object is not completed
                        {
                            FindObjectOfType<Progression>().OnCompletion();
                            completed = true; //Sets the object as completed
                        }
                    }
                }
            }
        }
    }

    public void Collecting(string tag)
    {
        if (tag == "Clam") //Check if the tag of the object is clam
        {
            Clams.ClamCount += 1; //Update the clam counter
            string message = "oesters: " + Clams.ClamCount + "/3 Ringe: " + Rings.RingCount + "/7"; //Create the string for the ui
```

```
54              missioncontroller.DoMission(message); //change the string in the ui
55              if (Clams.ClamCount == 3 && Rings.RingCount == 7) //Check if all the
                    objects have been completed
56              {
57                  missioncontroller.hasFinished(); //Set the mission as finished
58              }
59          }
60          else if (tag == "Skull") //Check if the tag of the object is skull
61          {
62              Skulls.SkullCount += 1; //Update the skull counter
63              string message = "Kranier: " + Skulls.SkullCount + "/7 Ringe: " +
                    Rings.RingCount + "/7"; //Create the string for the ui
64              missioncontroller.DoMission(message); //change the string in the ui
65              if (Skulls.SkullCount == 7 && Rings.RingCount == 7) //Check if all the
                    objects have been completed
66              {
67                  missioncontroller.hasFinished(); //Set the mission as finished
68              }
69          }
70      }
71
72      public void OnFocused(Transform playerTransform)
73      {
74          isFocus = true;
75          player = playerTransform;
76      }
77
78      public void OnDefocused()
79      {
80          isFocus = false;
81          player = null;
82      }
83
84      public void hasCollected()
85      {
86          transform.GetComponent<Renderer>().enabled = false; //diable the renderer of
                the object
87          foreach (Renderer renderTarget in children) //Go through all the children of
                the object
88              renderTarget.enabled = false; //diable the renderer of the object's
                    children
89          OnDefocused();
90      }
91
92      private void OnGUI()
93      {
94          int fontSize = 40;
95          guiStyle.fontSize = fontSize;
96          guiStyle.alignment = TextAnchor.LowerCenter;
97          if (isFocus == true && distance <= radius)
98          {
99              var centeredStyle = GUI.skin.GetStyle("Label");
100             centeredStyle.alignment = TextAnchor.UpperCenter;
101             GUI.Label(new Rect(Screen.width / 2 - fontSize / 2, Screen.height / 2 -
                    fontSize / 2, fontSize, fontSize), "'E' for at samle", guiStyle);
                    //Set the text to a tooltip in the middle of the screen
102         }
103         else
104         {
105             GUI.Label(new Rect(Screen.width / 2 - fontSize / 2, Screen.height / 2 -
                    fontSize / 4, fontSize, fontSize), "", guiStyle); //Set the text to
                    nothing
106         }
107     }
108 }
```

7.6.14 LandMovement

```
1  public class LandMovement : MonoBehaviour
2  {
3      public float speed;
4      public PainfulMovement PainfulScript;
```

```
5       public float gravityForce;
6
7       [HideInInspector]
8       public CharacterController cc;
9       public bool inputAllowed = true;
10      public float duration;
11      public bool boosted = false;
12      public float defaultSpeed;
13
14      public Animator anim;
15
16      void Start()
17      {
18          defaultSpeed = speed; //Set the default speed, used for boosting of the
                players speed
19          Cursor.lockState = CursorLockMode.Locked; //Change the cursor mode to be
                locked to the middle of the scene
20          cc = GetComponent<CharacterController>(); //Get the CharacterController
21
22          anim = GetComponentInChildren<Animator>(); //Get the Animator
23      }
24
25      // Update is called once per frame
26      void Update()
27      {
28          if (!inputAllowed)
29              return;
30
31          //Update the current direction of movement
32          Movement();
33
34          //Update the Gravity
35          Gravity();
36
37          //Painful Movement
38          if (Input.GetKey(KeyCode.W) || Input.GetKey(KeyCode.A) ||
                Input.GetKey(KeyCode.S) || Input.GetKey(KeyCode.D)) //Check if the player
                is moving
39          {
40              if (PainfulScript != null) //Check if a painfulscript is attached
41              {
42                  PainfulScript.PainfulMove(); //Call the PainfulMove method
43                  anim.SetInteger("AnimParam", 1); //Set the integer for the animation
                        to 1
44              }
45
46          }
47          else
48          {
49              anim.SetInteger("AnimParam", 0); //Set the integer for the animation to 0
50          }
51      }
52
53      //Move the character
54      public virtual void Movement()
55      {
56          //Input of movement
57          float vertical = Input.GetAxis("Vertical");
58          float horizontal = Input.GetAxis("Horizontal");
59
60          //Direction of camera
61          Vector3 newMove = Camera.main.transform.forward.normalized;
62
63          //movement vector in the x and z axises
64          Vector3 forwardVector = new Vector3(newMove.x, 0, newMove.z) * vertical *
                speed * Time.deltaTime;
65
66          //Movement vector for sideways
67          Vector3 sideVector = Camera.main.transform.right * horizontal * speed *
                Time.deltaTime;
68
69          //Move the character
70          cc.Move(sideVector + forwardVector);
71      }
```

```
72
73      //Gravity
74      public void Gravity()
75      {
76          if (gravityForce != null)
77          {
78              Vector3 gravityVel = new Vector3(0, -gravityForce, 0) * Time.deltaTime;
                    //Gravity descend
79              cc.Move(gravityVel); //Move the character
80          }
81      }
82  }
```

7.6.15 MissionController

```
1   public class MissionController : MonoBehaviour
2   {
3       public int counter = 0;
4       public bool isDone = true;
5       public GameObject CountCanvas;
6       public GameObject Mission0;
7       public GameObject Mission1;
8       public Text CounterMessage;
9
10
11      // Update is called once per frame
12      void Update()
13      {
14          if (counter == 0 || isDone) //Check if the missions is done
15          {
16              NoMission(); //Call the NoMission method
17          }
18      }
19
20      public void nextMission()
21      {
22          isDone = false;
23          counter += 1;
24          StartMission(); //Call the StartMission method
25      }
26
27      void StartMission()
28      {
29          if (counter == 0) //Check if the counter is 0
30          {
31              isDone = true;
32          }
33          else if (counter == 1) //Check if the counter is 1
34          {
35              if (Mission1 != null)
36              {
37                  if (Mission0 != null)
38                  {
39                      Mission0.SetActive(false); //Disable Mission0
40                  }
41                  Mission1.SetActive(true); //Enable Mission1
42                  if (CountCanvas != null)
43                  {
44                      CountCanvas.SetActive(true); //Enable CountCanvas
45                      CounterMessage.gameObject.SetActive(true); //Enable CounterMessage
46                  }
47
48                  Scene thisScene = SceneManager.GetActiveScene(); //Get the current
                        scene name
49                  if (thisScene.name == "CastleScene2")
50                  {
51                      Sisters.SisterCount = 0;
52                  }
53              }
54          }
55      }
56
```

```
57      public void DoMission(string message)
58      {
59          CounterMessage.text = message;
60      }
61
62      void NoMission()
63      {
64          if (Mission0 != null)
65          {
66              isDone = true;
67              Mission0.SetActive(true); //Enable Mission0
68              CountCanvas.SetActive(false); //Disable CountCanvas
69          }
70          else
71          {
72              nextMission(); //Call the nextMission method
73          }
74      }
75
76      public void hasFinished()
77      {
78          isDone = true;
79          Mission1.SetActive(false); //Disable Mission1
80          CountCanvas.SetActive(false); //Disable CountCanvas
81          CounterMessage.gameObject.SetActive(false); //Disable CounterMessage
82          Mission0.SetActive(true); //Enable Mission0
83      }
84  }
```

7.6.16 MouseManager

```
1   public class MouseManager : MonoBehaviour
2   {
3       void OnEnable()
4       {
5           Cursor.lockState = CursorLockMode.None;
6           Cursor.visible = true;
7       }
8   }
```

7.6.17 PainfulMovement

```
1   public class PainfulMovement : MonoBehaviour
2   {
3       public Image MovementImage;
4       public float cooldown;
5       public float fadeTime;
6       private float defaultDuration;
7       private float fadeTime2;
8
9       // Start is called before the first frame update
10      void Start()
11      {
12          defaultDuration = cooldown; //Set the default cooldown
13          fadeTime2 = cooldown - fadeTime; //Create a secondary fadeTimer for the fade
                out effect
14          MovementImage.CrossFadeAlpha(0f, 0f, false); //Set the movementimage's alpha
                to 0 (invisible)
15      }
16
17      // Update is called once per frame
18      void Update()
19      {
20          if (cooldown > 0) //Check if any time is left for the cooldown
21          {
22              cooldown -= Time.deltaTime; //Reduce the cooldown timer
23          }
24
25          if (cooldown < fadeTime2) //Check if it is the time to begin the fadeout
```

```
26          {
27              MovementImage.CrossFadeAlpha(0f, fadeTime, false); //Fadeout the image
28          }
29      }
30
31      public void PainfulMove()
32      {
33          if(cooldown <= 0) //Check if the cooldown is done
34          {
35              MovementImage.CrossFadeAlpha(1f, fadeTime, false); //Fadein the image
36              cooldown = defaultDuration; //Reset the cooldown
37          }
38      }
39  }
```

7.6.18 PauseMenu

```
1   public class PauseMenu : MonoBehaviour
2   {
3       public static bool IsPaused = false;
4       public GameObject PUI;
5
6       // Start is called before the first frame update
7       void Start()
8       {
9           Resume(); //Resume the game
10      }
11
12      // Update is called once per frame
13      void Update()
14      {
15          if(Input.GetKeyDown(KeyCode.P))
16          {
17              if(IsPaused) //Check if the game is paused
18              {
19                  Resume(); //Resume the game
20              }
21              else
22              {
23                  Pause(); //Pause the game
24              }
25          }
26      }
27
28      public void Resume()
29      {
30          Cursor.lockState = CursorLockMode.Locked; //Change the cursor mode to be
                locked to the middle of the scene
31          PUI.SetActive(false); //Disable the PUI
32          Time.timeScale = 1f; //Set the time scale to normal
33          Cursor.visible = false; //Disable the visibility of the cursor
34          IsPaused = false;
35      }
36
37      void Pause()
38      {
39          PUI.SetActive(true); //Enable the PUI
40          Time.timeScale = 0f; //Set the time scale to 0
41          Cursor.lockState = CursorLockMode.None; //Change the cursor mode to be free
42          Cursor.visible = true; //Enable the visibility of the cursor
43          IsPaused = true;
44      }
45
46      public void LoadMenu()
47      {
48          Time.timeScale = 1f; //Set the time scale to normal
49          SceneManager.LoadScene("StartScreen"); //Load the StartScreen
50      }
51
52      public void Exit()
53      {
54          Application.Quit(); // Exit the game
```

```
    }
}
```

7.6.19 PlayerDialogue

```
public class PlayerDialogue : MonoBehaviour
{
    private Transform target = null;
    public int dialogueRange;
    public GameObject canvasObject;
    private bool dialogueBool = false;
    private GUIStyle guiStyle = new GUIStyle();
    private LandMovement anotherScript;
    private RaycastHit hit;
    private Ray ray;
    LayerMask mask;


    void Update()
    {
        mask = LayerMask.GetMask("Default"); //Mask to make sure that the raycast
            does not hit the player model

        ray = new Ray(Camera.main.transform.position, Camera.main.transform.forward);
            //Create a ray from the main camera

        dialogueBool = false;
        if(Physics.Raycast(ray, out hit, dialogueRange, mask)) //Check if the ray
            hits anything within the dialogue range
        {
            if(hit.collider.gameObject.GetComponent<DialogueT>() &&
                canvasObject.activeSelf == false &&
                hit.collider.gameObject.GetComponent<DialogueT>().isActiveAndEnabled)
                //Check if it hits a npc with the DialogueT script and if the player
                is not in a dialogue
            {
                target = hit.transform;
                dialogueBool = true;
                if(Input.GetKeyDown(KeyCode.E))
                {
                    DialogueHit(Camera.main.transform.forward);

                }
            }
        }
    }

    public void DialogueHit(Vector3 dir)
    {
        RaycastHit hit;

        if (Physics.Raycast(Camera.main.transform.position, dir, out hit,
            dialogueRange, mask))
        {
            INpcDialogue id = hit.collider.gameObject.GetComponent<INpcDialogue>();
                //Get the npc dialogue interface
            if (id != null)
            {
                this.GetComponent<LandMovement>().enabled = false; //Disable the
                    player's movement
                canvasObject.SetActive(true); //Enable the dialogue canvas
                this.GetComponentInChildren<TPCamControl>().enabled = false;
                    //Disable the player's camera movement
                Cursor.lockState = CursorLockMode.None; //Make the cursor mode free
                Cursor.visible = true; //Make the cursor visible
                id.triggerDia(); //Trigger the dialogue
            }
        }
    }

    private void OnGUI()
    {
```

```
57          int fontSize = 40;
58          guiStyle.fontSize = fontSize;
59          guiStyle.alignment = TextAnchor.LowerCenter;
60          if (dialogueBool == true)
61          {
62              var centeredStyle = GUI.skin.GetStyle("Label");
63              centeredStyle.alignment = TextAnchor.UpperCenter;
64              GUI.Label(new Rect(Screen.width / 2 - fontSize / 2, Screen.height / 2 -
                    fontSize / 2, fontSize, fontSize), "'E' til at start dialog",
                    guiStyle); //Set the text to a tooltip in the middle of the screen
65          }
66          else
67          {
68              GUI.Label(new Rect(Screen.width / 2 - fontSize / 2, Screen.height / 2 -
                    fontSize / 4, fontSize, fontSize), "", guiStyle); //Set the text to
                    nothing
69          }
70      }
71
72      private void OnDrawGizmos()
73      {
74          Ray ray = new Ray(Camera.main.transform.position,
                Camera.main.transform.forward * 100f);
75          Debug.DrawRay(Camera.main.transform.position, Camera.main.transform.forward *
                dialogueRange, Color.white); //Create a white line in the editor
76      }
77  }
```

7.6.20 Progression

```
1   public class Progression : MonoBehaviour
2   {
3       public List<ProgressionData> data;
4
5       private List<int> values = new List<int>();
6       private int currentStage = 0;
7       private int maxValue = 0;
8       private int[] valueArray;
9       private int completedObjects = 0;
10
11      // Start is called before the first frame update
12      void Start()
13      {
14          OnStageUpdate(); //Call the OnStageUpdate method
15          HighestStage(); //Call the HighestStage method
16      }
17
18      void OnStageUpdate()
19      {
20          currentStage += 1; //Update the currentStage
21          foreach(ProgressionData currentObject in data) //Go through every object in
                the data List
22          {
23              if (currentObject != null)
24              {
25                  if (currentStage == currentObject.stage) //Check if the currentStage
                        is the same stage as the current object's stage
26                  {
27                      if (currentObject.GetComponent<DialogueT>()) //Check if the
                            current object has the DialogueT script
28                      {
29                          currentObject.GetComponent<DialogueT>().enabled = true;
                                //Enable the DialogueT script
30                      } else if (currentObject.GetComponent<Interactive>()) //Check if
                            the current object has the Interactive script
31                      {
32                          currentObject.GetComponent<Interactive>().enabled = true;
                                //Enable the Interactive script
33                      } else if (currentObject.GetComponent<TransitionCollider>())
                            //Check if the current object has the TransitionCollider
                            script
34                      {
```

```
35                          currentObject.GetComponent<TransitionCollider>().enabled =
                                true; //Enable the TransitionCollider script
36                      }else if (currentObject.GetComponent<Boost>()) //Check if the
                            current object has the Boost script
37                      {
38                          currentObject.GetComponent<Boost>().enabled = true; //Enable
                                the Boost script
39                      }
40                  }
41                  else
42                  {
43                      if (currentObject.GetComponent<DialogueT>()) //Check if the
                            current object has the DialogueT script
44                      {
45                          currentObject.GetComponent<DialogueT>().enabled = false;
                                //Disable the DialogueT script
46                      } else if (currentObject.GetComponent<Interactive>()) //Check if
                            the current object has the Interactive script
47                      {
48                          currentObject.GetComponent<Interactive>().enabled = false;
                                //Disable the Interactive script
49                      } else if (currentObject.GetComponent<TransitionCollider>())
                            //Check if the current object has the TransitionCollider
                            script
50                      {
51                          currentObject.GetComponent<TransitionCollider>().enabled =
                                false; //Disable the TransitionCollider script
52                      }else if (currentObject.GetComponent<Boost>()) //Check if the
                            current object has the Boost script
53                      {
54                          currentObject.GetComponent<Boost>().enabled = false;
                                //Disable the Boost script
55                      }
56                  }
57              }
58          }
59      }
60
61      void HighestStage()
62      {
63          foreach(ProgressionData currentObject in data) //Go through every object in
                the the data List
64          {
65              if (currentObject.stage > maxValue) { //Check if the current object's
                    stage is higher than the maxValue
66                  maxValue = currentObject.stage; //Set the maxValue as the object's
                        stage
67              }
68              values.Add(currentObject.stage); //Add the current object's stage to the
                    values list
69          }
70          valueArray = new int[maxValue]; //Create an array the size of the maxValue
                value
71
72          Dictionary<int,int> stageDictionary = new Dictionary<int, int>(); //Create an
                Dictionary
73          foreach(int item in values) //Go through each item in the values list
74          {
75              if (!stageDictionary.ContainsKey(item)) //Check if the current item is
                    not present in the list
76              {
77                  stageDictionary.Add(item,1); //Add the item to the Dictionary
78              }
79              else //If the item is present
80              {
81                  int count = 0;
82                  stageDictionary.TryGetValue(item, out count); //Get the item's entry
                        and count
83                  stageDictionary.Remove(item); //Remove the item
84                  stageDictionary.Add(item, count+1); //Add the same item with one
                        count higher
85              }
86          }
87
```

```
88          foreach(KeyValuePair<int,int> entry in stageDictionary)
89          {
90              valueArray[entry.Key-1] = entry.Value; //But the count from
                    stageDictionary into the valueArray
91          }
92      }
93
94      public void OnCompletion()
95      {
96          completedObjects +=1; //Update the completedObjects
97          if (completedObjects >= valueArray[currentStage-1]) //Check if
                completedObjects is the same or higher than the value in the valueArray
98          {
99              OnStageUpdate(); //Call the OnStageUpdate
100             completedObjects = 0; //Set the completedObjects to 0
101         }
102     }
103 }
```

7.6.21 ProgressionData

```
1   public class ProgressionData : MonoBehaviour
2   {
3       public GameObject objectName;
4       public int stage;
5
6       public ProgressionData(GameObject newObjectName, int newStage)
7       {
8           objectName = newObjectName;
9           stage = newStage;
10      }
11  }
```

7.6.22 Rings

```
1   public class Rings : MonoBehaviour
2   {
3       public static int RingCount = 0;
4
5       void Start()
6       {
7       }
8   }
```

7.6.23 SceneTransition

```
1   public class SceneTransition : MonoBehaviour
2   {
3       public string sceneName;
4
5       void Start()
6       {
7       }
8
9       public void TransitionScene()
10      {
11          if (this.enabled) //Check if this script is enabled
12          {
13              SceneManager.LoadScene(sceneName); //Load the sceneName scene
14          }
15      }
16  }
```

7.6.24 Sisters

```
public class Sisters : MonoBehaviour
{
    public static int SisterCount = 0;

    void Start()
    {
    }
}
```

7.6.25 Skulls

```
public class Skulls : MonoBehaviour
{
    public static int SkullCount = 0;

    void Start()
    {

    }
}
```

7.6.26 SMenu

```
public class SMenu : MonoBehaviour
{
    public void StartGame()
    {
        // Change scene to the game
        SceneManager.LoadScene("InBetween1");
    }

    public void Quit()
    {
        // Exit the game
        Application.Quit();;
    }
}
```

7.6.27 TPCamControl

```
public class TPCamControl : MonoBehaviour
{
    public float rotationS = 2;
    public Transform target, player;
    float mX, mY;
    public float FOV;
    public float x;
    public float y;
    public Transform obstruct;
    public float zoomS = 2f;

    void Start()
    {
        transform.Translate(Vector3.forward * FOV);
        transform.Translate(Vector3.up * y);
        transform.Translate(Vector3.right * x);
        obstruct = target;
        Cursor.visible = false;
        Cursor.lockState = CursorLockMode.Locked;
    }

    // Update is called once per frame
    void LateUpdate()
```

```
24      {
25          camControl(); //Call the camControl method
26          viewObstr(); //Call the viewObstr method
27      }
28
29      void camControl()
30      {
31          mX += Input.GetAxis("Mouse X") * rotationS; //Get the mouse's x coordinate
32          mY += Input.GetAxis("Mouse Y") * rotationS; //Get the mouse's y coordinate
33
34          mY = Mathf.Clamp(mY, -35, 60);
35          transform.LookAt(target);
36
37          if(Input.GetKey(KeyCode.LeftShift))
38          {
39              target.rotation = Quaternion.Euler(-mY, mX, 0);
40          }
41          else
42          {
43              target.rotation = Quaternion.Euler(-mY, mX, 0);
44              player.rotation = Quaternion.Euler(0, mX, 0);
45          }
46      }
47
48      void viewObstr()
49      {
50          RaycastHit hit;
51          if(Physics.Raycast(transform.position, target.position - transform.position,
                out hit, FOV))
52          {
53              if(hit.collider.gameObject.tag != "Player")
54              {
55                  obstruct = hit.transform;
56                  obstruct.gameObject.GetComponent<MeshRenderer>().shadowCastingMode =
                        UnityEngine.Rendering.ShadowCastingMode.ShadowsOnly;
57
58                  if(Vector3.Distance(obstruct.position, transform.position) >= 3f &&
                        Vector3.Distance(transform.position, target.position) >= 1.5f)
59                  {
60                      transform.Translate(Vector3.forward * zoomS * Time.deltaTime);
61                  }
62              }
63              else
64              {
65                  obstruct.gameObject.GetComponent<MeshRenderer>().shadowCastingMode =
                        UnityEngine.Rendering.ShadowCastingMode.On;
66                  if(Vector3.Distance(transform.position, target.position) < FOV)
67                      transform.Translate(Vector3.back * zoomS * Time.deltaTime);
68              }
69          }
70      }
71  }
```

7.6.28 TransitionCollider

```
1   public class TransitionCollider : MonoBehaviour
2   {
3       public string sceneName;
4
5       void Start()
6       {
7       }
8
9       private void OnTriggerStay(Collider other)
10      {
11          if (this.enabled) //Check if this script is enabled
12          {
13              if (other.tag == "Player") { //Check if the collider object is the player
14                  SceneManager.LoadScene(sceneName); //Load the sceneName scene
15              }
16          }
17      }
```

```
}
```

7.6.29 UnderWater

```
public class UnderWater : MonoBehaviour
{

    //This script enables underwater effects. Attach to main camera.
    //Define variable
    public int underwaterLevel = 7;
    //The scene's default fog settings
    private bool defaultFog;
    private Color defaultFogColor;
    private float defaultFogDensity;
    private Material defaultSkybox;
    private Material noSkybox;

    void Start()
    {
        //Set the background color
        GetComponent<Camera>().backgroundColor = new Color(0, 0.4f, 0.7f, 1);
        defaultFog = RenderSettings.fog; //Set the defaultFog
        defaultFogColor = RenderSettings.fogColor; //Set the defaultFogColor
        defaultFogDensity = RenderSettings.fogDensity; //Set the defaultFogDensity
        defaultSkybox = RenderSettings.skybox; //Set the defaultSkybox
    }

    void Update()
    {
        if (transform.position.y < underwaterLevel) //Check if the player is under
            the underwater level
        {
            RenderSettings.fog = true; //Enable the fog
            RenderSettings.fogColor = new Color(0, 0.4f, 0.7f, 0.6f); //Set the color
                of the fog
            RenderSettings.fogDensity = 0.01f; //Set the density of the fog
            RenderSettings.skybox = noSkybox; //Set the skybox as noSkybox
        }
        else
        {
            RenderSettings.fog = defaultFog; //Set the fog as defaultFog
            RenderSettings.fogColor = defaultFogColor; //Set the fog color as
                defaultFogColor
            RenderSettings.fogDensity = defaultFogDensity; //Set the fog density
                defaultFogDensity
            RenderSettings.skybox = defaultSkybox; //Set the skybox as defaultSkybox
        }
    }
}
```

7.6.30 WaterPlayerMovement2

```
public class WaterPlayerMovement2 : LandMovement, IMoveSpeedBoost<float, float>
{
    // Update is called once per frame
    void Update()
    {
        if (boosted) //Check if the player's movement is boosted
        {
            duration -= Time.deltaTime; //Counts down the knockdown time
            if(duration <= 0) //Checks if the knockdown timer is done
            {
                speed = defaultSpeed; //Change the speed to the default speed
                boosted = false;
            }
        }

        if (!inputAllowed)
            return;
```

```
18
19
20          //Update the current direction of movement
21          Movement();
22
23          Gravity();
24
25          //Update if ascending
26          ascend();
27
28          //Update if descending
29          descend();
30
31          if (Input.GetKey(KeyCode.W) || Input.GetKey(KeyCode.A) ||
                Input.GetKey(KeyCode.S) || Input.GetKey(KeyCode.D)) //Check if the player
                is moving
32          {
33              anim.SetInteger("AnimParam", 1);
34          }
35          else
36          {
37              anim.SetInteger("AnimParam", 0);
38          }
39      }
40
41      public void Speed(float boostSpeed, float boostDuration) //Change speed method
42      {
43          speed = boostSpeed;
44          duration = boostDuration;
45          boosted = true;
46      }
47
48      //Move the character
49      public override void Movement()
50      {
51
52          //Input of movement
53          float vertical = Input.GetAxis("Vertical");
54          float horizontal = Input.GetAxis("Horizontal");
55
56          //Direction of camera
57          Vector3 newMove = Camera.main.transform.forward.normalized;
58
59          //movement vector in the x and z axises
60          Vector3 forwardVel = new Vector3(newMove.x, 0, newMove.z) * vertical * speed
                * Time.deltaTime;
61
62          //Movement vector for sideways
63          Vector3 sideVel = Camera.main.transform.right * horizontal * speed *
                Time.deltaTime;
64
65          //Move the character
66          cc.Move(sideVel + forwardVel);
67      }
68
69      //Ascend the character
70      void ascend()
71      {
72          //Check if the ascend key is being pressed
73          if (Input.GetKey("space") && transform.position.y < 140)
74          {
75
76              //movement vector for ascend
77              Vector3 ascendVel = new Vector3(0, 1, 0) * speed * Time.deltaTime;
78
79              //Ascend the character
80              cc.Move(ascendVel);
81          }
82      }
83
84      //Descend the character
85      void descend()
86      {
87          //Check if the descend key is being pressed
```

```
88          if (Input.GetKey(KeyCode.LeftShift))
89          {
90
91              //movement vector for descend
92              Vector3 descendVel = new Vector3(0, -1, 0) * speed * Time.deltaTime;
93
94              //Descend the character
95              cc.Move(descendVel);
96          }
97      }
98  }
```