<table>
<thead>
<tr>
<th>Author</th>
<th>Degree title</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Justus Juutilainen</td>
<td>Bachelor of Engineering</td>
<td>May 2021</td>
</tr>
</tbody>
</table>

**Thesis title**

Developing a game to maintain and develop logical thinking

33 pages

1 pages of appendices

**Commissioned by**

"ReissuEllu -wellbeing on wheels – a mobile learning environment“ -project/South Savo Social and Healthcare Authority ESSOTE

**Supervisor**

Timo Mynttinen

**Abstract**

Goal for this thesis was to develop a game for elderly people that could be used to train cognitive skills in a fun way while being easy enough to use. Game developed was a 2D puzzle game created using Unity game engine. The project was done by using iterative development method to ensure that the needs of a target group were filled.

The theory part of the thesis consists of discussing about the relevance of games to cognitive health, explaining game development tools and agile development. While the project done reflects how these factors were considered during the development and explains how different parts of the project were accomplished.

The outcome of the project was a functional puzzle game that was given to the commissioner of the thesis. The game includes a board game-like grid with tile elements, animated instructions, and a scoring system with a target to motivate a player to keep playing.

**Keywords**

Unity, video games, puzzle, scrum
5.6.2 Score .................................................................................................................. 27
5.7 End Scene ............................................................................................................ 28
6 CONCLUSIONS .................................................................................................... 29
REFERENCES ............................................................................................................ 30

LIST OF FIGURES

APPENDICES

   Appendix 1. ReissuEllu assignment
1 INTRODUCTION

This thesis was made to be used as a part of ReissuEllu project. ReissuEllu project was managed by Etelä-Savon Koulutus Oy and co-implemented by South Savo Social and Health Care Authority (Essote) and Mikkeli Development Company Miksei Oy. The project was funded by the European Social Fund (ESF) and European Regional Development Fund (ERDF). The ReissuEllu project built a mobile learning environment around wellbeing car, ReissuEllu, which develops health, well-being, and service guidance in order to reach elderly people in remote areas. (Appendix 1)

The practical part of the thesis, the actual project, will be done in co-operation with a peer student. The goal of the practical part is to develop a game which can be used over a web browser to improve and maintain the ability of logical thinking. The main target group for the project will be elderly people. Even though the practical project was done in collaboration with another person, theoretical perspectives between the two theses differ. This thesis focuses on the development of the game while the other focuses more on the post-production development, based on the user experiences and feedback.

The opportunity to complete such a thesis project occurred to us while participating Games by Saimaa event organized by Game Lab Mikkeli together with Hack Lab Mikkeli between 15th and 17th of November 2019. The purpose of the event was to gather people from various backgrounds in order to develop, play, and watch different kind of games. For the event organizers, ReissuEllu representatives had given a challenge to develop something which could be used in their project. The copy of the given challenge and more information about the ReissuEllu project can be read in the appendices.

During thesis work, we will get to know games relevance to cognitive health, getting familiar with methods already existing to reach the similar goals. After, we are going to go through some components and tools used in game development. Finally part of the thesis includes reporting the methods used to complete the given project.
2 RELEVANCE OF GAMES IN MAINTAINING COGNITIVE SKILLS

This chapter will focus on games among elderly people. Our bodies are not meant to function forever. To maintain better health, we also need to train our brain and mind, in addition to staying physically healthy.

Based on a survey performed in the United States in 2016, 38% of the age group -50+ play video games, with three quarters of the gamers playing weekly. Considering the people over 50, more consistent gamers seem more common in the older age categories, with 37% of the gamers in the 50–59-year-old category reporting daily gaming, whereas 43% in the 60+ category. (Anderson 2016.)

2.1 Benefits of games

Cognitively stimulative games have multiple positive effects. As we use the term games, we are meaning puzzles, board games, cross words and video or mobile games.

It has been evaluated by Cochrane, a British charity organization, that putting jigsaw puzzles together can help seniors, suffering from dementia, get higher scores on memory evaluations compared to those who did not work on puzzles on a weekly basis (Buck, Rob. 2019).

While working on puzzles both sides of one's brains are stimulated. When puzzles are being solved neurons in one's brain coordinate visual input and the fine motor of hands. This improves the motor skills such as hand-eye coordination – while also increasing judgement and processing times. (Buck, Rob. 2019)

Dementia commonly damages visual perception and memory in seniors. Visuospatial functioning is integral to a person's ability to process, identify, detail, and analyze a space and visual form. It is also essential to a person's movement, and depth and distance perception. Puzzles and different hands-on problem-
solving methods do help maintaining these important skills. When matching colors, patterns and shapes, long-term memory is stimulated. Also, short-term memory is needed when finding piece which did not have place earlier. As the different areas of brain function together to solve the mystery, brain neurons create new connections to bypass damaged communication routes. (DPS Publishing Pty Ltd 2021)

Cardiovascular system is another part of human body which benefits from this. Completing a puzzle can have a calming effect for a senior with dementia, and it lowers breathing, heart rate and blood pressure. (Buck, Rob. 2019)

In addition to the great benefits above, activities such solving puzzles can fill a day and give a sense of accomplishment. (DPS Publishing Pty Ltd 2021)

2.2 Existing methods

Sometimes one does not even realize that he or she is developing or maintaining brain functionality. Completing sudoku or Rubik’s cube is one way to improve the way we recognize logical paths. Those mentioned are not the only ones, but now-a-days they are rarely used intentionally to improve this kind of brain function - they are mainly played for fun. For example, chess has been used to improve military leaders’ skill of anticipating opponents’ actions. Below are some examples of games that already exist.

**Chess**

As mentioned earlier, Chess is a great mental workout to maintain strategic and problem-solving skills.

Idea of chess is to “eat” the opposite players king pawn. Pawns can be eaten by moving one’s piece to the same square. Each type of pawn has its own way that it can be moved around the chess grid.

Chess encourages to think ahead and to react to ever changing situations. Chess requires two players which also makes it a great way to socialize and to connect with another chess enthusiasts. (DPS Publishing Pty Ltd 2021)
**Sudoku**

Sudoku is a great puzzle to train ability to recognize patterns and to think ahead. Each space needs to be filled with number from one to nine. In grid of nine by nine each row and column must contain all the numbers, and there should not be any collapses – meaning one row/column should not include the same number twice (DPS Publishing Pty Ltd 2021).

**Crossword**

The nature of crosswords makes them a great way for evolving one’s use of critical thinking, and it requires problem solving skills and brainstorming. Crossword is a puzzle where there is a grid which needs to be filled with a word. Each row or column has its own hint, and additionally, words do overlap with other words meaning that the player needs to fill in the correct word to be able to solve another properly. (DPS Publishing Pty Ltd 2021)

**Jigsaw Puzzle**

Jigsaw puzzle is most probably drawn to one’s retina when the word puzzle is mentioned. It is basically a picture of some sort that has been printed onto a cardboard and then cut into different pieces – usually squarish with different edges that fit together with the correct counterpart.

Difficulty of a jigsaw puzzle can vary based on the size of the pieces, amount of the pieces or the desired picture. Some people use jigsaw puzzles to relieve stress, but they are also great way to improve analytical side of one’s brain. Again, like with sudoku, jigsaw puzzles require pattern recognition skills in addition to fine motor and creative skills. (DPS Publishing Pty Ltd 2021)

**Word Search**

In word search one must find words from a grid full of letters. Word search might have hints of words to guide the search. To find the words from the grid visual perception is needed.
From all the games mentioned above, chess is the only game that requires two players, all the other games can be played alone, but they can also be played together with someone. For example, jigsaw puzzles can be a joint activity with one’s family. (DPS Publishing Pty Ltd 2021)

These are by no means the only games that are useful when considering maintaining one’s mental health. For example, classic board games such as Monopoly and Cluedo are great ways to train one’s brain. (DPS Publishing Pty Ltd 2021)

3 GAME DEVELOPMENT TOOLS

It has been a while since video games took over. Between 1990 and 2020 we have seen games evolving from arcade machines into portable machines like Nintendo DS series or Sony’s PSP, Play Station Portable. Nowadays there are different ways to enjoy this virtual amusement. Even one’s mobile device can be used as a platform for playing video games. Playing video games has even got a viral verb: gaming.

So, what does one need to develop a game? Unlike in the early days of video games, there is no need to make everything from scratch. Different kind of tools have been created to ease the process.

3.1 Game Engines

When considering a game engine to use on one’s game, there are basically two game engines available – Unity and Unreal Engine. Unity has been a dominant engine for mobile game development since it became popular, while Unreal has been game engine for PC/Console game development. Even though the pattern for game engine use has been as previously mentioned, both game engines have been trying to increase the variety of the games that can be built with the engines. In other words, Unity is expanding towards PC and Console games while Unreal Engine is trying to involve more on mobile games.
Unreal engine supports developing games for mobile platforms like iOS and Android, for consoles like PS4, XBOX ONE, and Nintendo Switch, and for desktop operating systems like Windows, Mac, and Linux. In addition, Unreal also has support for VR platforms such as Oculus, SteamVR, PSVR, Google Daydream, and Samsung Gear VR. (Mahesa, Raka 2018)

Unity does not only support all platforms mentioned above, but it also has support for smart TV platforms such as Android TV and Samsung SmartTV. Another technology Unity supports unlike Unreal is AR (Augmented reality), for example, Apple ARKit and Google ARCore are supported. In other words, if you are looking for more compatible platform for game development choose Unity, Unity is also the platform first to support platform after one is launched. (Mahesa, Raka 2018)

When considering features between the two Unreal might be the winner. Unreal Engine provides more built-in tools for game development, for example cinematic editor, where-as Unity relies much on 3rd party addons which can be used through asset store. (Mahesa, Raka 2018.)

3.2 Modeling

Modeling is crucial component of today’s modern digital media landscape. Since we are working with Unity, we will discuss modeling from Unity point of view. As discussed on earlier chapter, Unity relies on third-party addons with different features. Considering big game development enterprises, having dedicated modeling responsible is not farfetched.

What is modeling? “3D modeling is a technique in computer graphics for producing a 3D digital representation of any object or surface.” (Petty, Josh. 2018). These digital objects are capable of begin fully animated, making it an essential process for character animation and creating special effects. How are these so-called models created? (Petty, Josh. 2018).
Modeling artist manipulates vertices in virtual space to form an object, one object requires multiple vertices called collection of vertices. Mesh of vertices is the core for a model. Each point or vertex in a mesh has its own position in virtual space. By combining points into shapes, one is able to create surface for the desired object. To manipulate the surface, vertices can be adjusted to form different shapes. (Petty, Josh. 2018).

Maya, Cinema4D, Blender and SketchUp are all popular modeling software with each having their own benefits and features. In other words, selecting a platform for modeling can be dependent on the nature of the project. Whether one is looking to create animated characters, game models, or design a house, there is a program for any purpose. (Petty, Josh. 2018).

3.3 Animations

Above we discussed modeling which is a way to create characters and object into a gaming scene. To make characters or object move so that the motion seems realistic, one needs animations. Animations can be created by using drawings, models, or computer images. Animations are created by rapidly swapping images of a same object in a different position. Desired illusion is created with a change so fast that one’s brain interprets the change of position as continuous movement. (Pluralsight 2014)

Originally, drawings were the method of choice for animation. Now a days it is more common that animations are made with CGI (Computer-Generated Imagery). When it comes to drawings, they were applied to transparent sheet and then photographed. With CGI involved, we are able to adjust the position of an object and then record the change – this method is called motion capture whereas documenting every change manually is called stop-motion method. (Pluralsight 2014)

Most of the animation applications used in digital animation are so evolved that they can predict the path for the object by comparing the starting position to the end position. This is useful when considering, for example rotational movement –
we are not required to capture an image of each possible position but with few positions which determine the direction of the rotation and the desired end position. This kind of technology enables us even to create animated movies and video games that look realistic. (Pluralsight 2014)

4 AGILE DEVELOPMENT METHODS

Agile is an umbrella term for many modern development methods widely used, for example scrum. Agile is about getting rapid feedback to make the product and development culture better. Target of agile development is continuous improvement. To accomplish this continuous improvement within a project, project can be divided into smaller segments. This is where iterative development comes in meaning that as a first step, create a prototype and then finetune it to perfection. Sprint is a term one uses for these segments. To visualize movement between sprints, sprints can be thought of as cycles, after the first cycle is done the next one begins. Each cycle includes the same steps to complete before the next cycle begins - planning, development, testing, deployment and reviewing. (Eby, Kate. 2016)

Each sprint consists of four different meetings or “ceremonies”, which are an important part of agile development. Good example of agile development method is scrum. Subsequent chapters will dip into scrum and the four ceremonies from the scrum point of view.

Figure 1 Agile development cycles (Tartila. 2020)
4.1 Scrum

Scrum divides the project into smaller segments called sprints. Point of each sprint is to get part of the total project done. There should always be a useable product after each sprint - meaning that after the first sprint there should already be a raw version of the desired outcome ready for launch. Latter sprints are used to add features and to finetune the previously completed parts. (Drumond, Claire)

4.2 Scrum team

In scrum, project group is considered to be a team. Team consist of product owner, scrum master and the development team.

**Development team**
Responsibility of a development team is to deliver work to get desired outcome for the project. Unlike one might think, development team is not just a group of programmers. Development team consists of everyone whose effort is required to get the product done. Considering game development, a development team could, for example, include modeling and animation experts, writers, designers, and programmers. While each member of a development team might have a different expertise and area of responsibility, it is crucial for the team to work transparently. To ensure transparency, development team meets at a daily stand-up. (West, Dave)

![Figure 2 Development team (West, Dave)]
**Product owner**

Product owner is a member working between the development team and stakeholders. Product owner’s responsibilities include understanding the vision of stakeholders and then, based on the vision, setting clear directions for the development team to optimize the value created. To be able to set clear directions, it is product owner’s responsibility to keep track of the scrum board and prioritize tasks included in it. Scrum board is a visual representation of tasks’ state at the time. Scrum board can also be referred to as a backlog. (West, Dave)

![Scrum board example](Rodriguez, Guerra 2016)

**Scrum master**

Scrum master’s role is to keep it all together. Scrum master is responsible for making sure scrum process is properly harnessed in a project. Key word for a scrum master is transparency. Scrum master ensures that every opportunity to improve is used, for example by inspecting daily stand-ups. (West, Dave)

Scrum master should also be able to adapt. For scrum to work as a process, scrum master should help development team and product owner if necessary, by solving blocks that may decrease efficiency in workflow. For example, by helping product owner to understand the stakeholders’ vision and to prioritize the backlog. (West, Dave)
A common misbelief is that scrum master is responsible for holding ceremonies but that is not the case. Scrum master is responsible of making sure the ceremonies are held and everything is done properly scrum wise. (West, Dave)

4.3 Ceremonies

As mentioned earlier, meetings, also known as ceremonies, are crucial part of the nature of agile development methods. The idea of ceremonies is to simplify communication across the scrum team. Each ceremony has its own role in the process. The four ceremonies are sprint planning, daily stand-up, iteration review and retrospective. Scrum board can be used as a tool in these ceremonies to go through tasks on hand. (Radigan, Dan)

**Sprint Planning**

Sprint planning ceremony is held at the beginning of a sprint. The goal is to determine sprint duration and goals. Product owner describes the goals for the sprint, in other words comes up with the prioritized backlog. Development team then reviews the product backlog, evaluates the necessary effort required to fulfill the goals. This consists of determine priority for desired features and adjustments marked to the product backlog. Development team then negotiates with the product owner resulting sprint plan based on value and effort. (Radigan, Dan)

**Daily Stand-up**

Daily stand-up is held once a day. The norm is to have it at the start of the day. Daily stand-ups should be kept short – no more than 15minutes. Point of this Ceremony is to go through what was done yesterday, what is done today and if there are any blocks or problems. Going through these prevents two team members of doing the same task, in addition to keeping track of the progress. (Radigan, Dan)
**Iteration review**

Iteration review is held at the end of a sprint. The idea of an iteration review is to go through the work done during a sprint and get quick feedback from product owner or stakeholders. Like mentioned earlier, there should be some features or parts complete and ready to be demonstrated – meaning that none of the unfinished work will be gone through in this ceremony.

Iteration reviews do not always have to be formal, they can be used to give development team a time to celebrate the work done during a sprint. (Radigan, Dan)

**Retrospective**

Retrospectives are held once an iteration. Agile development is all about continual improvement, what a better way to accomplish that than getting rapid feedback – that is the idea of having retrospective ceremonies. In a retrospective ceremony, development team, product owner and scrum master take a look at the iteration and go through the process from the point of view what worked well and what did not. (Radigan, Dan)

5 **REISSUELLU PROJECT**

This project started as a challenge from ReissuEllu in an event: games by Saimaa. The goal was to develop a game which could be used to train cognitive skills. Form of the game was up to the developers to decide, and the best idea was to be recognized. Our team won the best idea and prototype challenge. Our team consisted of Justus Juutilainen, Jani Saari and Juuso Kolehmainen. After the recognition, ReissuEllu representative Sointu Pajunen asked whether we would like to continue developing the game as a thesis project.

As mentioned in Introduction, project was done using Unity and Visual Studio. C# was the programming language used and all of the visual outputs were done by the development team. The game is designed to be used with a tablet computer or by PC.

In this chapter we are going to go through the actual project done for ReissuEllu.
I will divide the project into few sub-subjects based on developed scenes.

5.1 Challenges

As all projects, we also faced some difficulties related to development. Some of the challenges we knew we needed to consider while some were challenges that we were not prepared for beforehand.

5.1.1 Team Coordination

One of the challenges we knew we had to face was the fact that our team was not going to develop the game inside the same premises together. So, to be able to keep track of where we were going, we needed to adapt the methods from agile development meaning having a scrum board of a sort and ceremonies to keep track of the process. In addition to methods mentioned in the previous sentence, we needed to have a platform which could be used to push and pull our features to the shared project.

For version control, we ended up using unity hub which had a feature called unity collaborate for shared projects. With unity collaborate we were able to do version control and to share new features with the rest of the development team.

Considering our agile development method, we used scrum-like method, meaning that we did not have all the ceremonies held as in scrum, but we held sprints, including planning, and reviewing the progress. Daily standups would not have worked for us since we were not developing the project on a daily basis. We agreed that it would be best to develop when ever we were capable as long as the goals for the sprint were met.

5.1.2 Target group

Another challenge we knew was coming for us was the fact that the target group for our project was not digital native. Digital native as a term means a person who has grown up in digital era, is familiar with digital equipment and is using digital devices in daily basis.
As we had recognized our target group early in the project, we decided to keep the UI (User Interface) as simple as possible. Meaning that we could prune all the fancy special effects and animations off from the output, just to keep it simple enough and not to create any un-necessary disturbance that could misplace the focus while playing the game.

We did not recognize all the challenges created by the target group before it was time to test the project with the target group. For example, objects in the game should be large enough so that they are clearly visible and the contrast between colors should be large enough to help players to separate objects from each other. As we noticed during our testing phase of the project, if previously mentioned factors are not considered playing the game can lead to frustration instead of the target – having fun whilst training cognitive skills.

5.2 Development based on results

At the beginning of the project, we had decided that we will use iterative development as our development method. Meaning holding few testing sessions with the target group between versions to recognize possible problems within the game.

This ended up not being as easy as we intended first, mainly because of the COVID-19 outbreak in Finland during spring of 2020. The pandemic caused us to delay our testing sessions and to rethink how to organize them. Initial idea was to meet with the target group and introduce our project for them but due to the COVID situation, we had to arrange one-to-one testing sessions with elderly as a means to avoid endangering anyone’s health.

Iterative method gave us insight on problems, such as visual impairments. For example, while recognizing different shades of a color can be easy for some it might not be the same for everyone. Especially for elderly in which by the age of 65 approximately one third of people have a condition that impairs their vision
(Ganley & Roberts, 1983). This problem manifested during our first testing session, especially the problem was present with road tiles in our Main Scene.

![Figure 4 Road tile](image)

The contrast between the shades was improved after the first session, which was then proven to be helpful during the next session.

### 5.3 Scenes

Scenes contain the objects of one’s game. There can be multiple scenes in one project. Scenes can be used to create different levels to a game, and they can be used to move between different parts of a game – in our case we had just one level of gameplay and three other scenes which will be explained later.

![Figure 5 Scenes](image)
5.4 Introduction Scene

Point of the introduction scene is to display three buttons and logos of the organizations involved in ReissuEllu project – and in the game development.

![Introduction scene with buttons](image)

Figure 6 Introduction scene

```csharp
public class AloitusruutuKoodi : MonoBehaviour
{
    public GameObject OhjeTaulu = null;
    public void Aloita_Painettu()
    {
        SceneManager.LoadScene("SampleScene");
    }

    public void Lopeta_Painettu()
    {
        Application.Quit();
    }

    public void Ohjeet_Painettu()
    {
        SceneManager.LoadScene("OhjesceneNew");
    }
}
```

Figure 7 Code to move between scenes

Buttons on the scene direct player to desired scene. Possibilities are start, instructions and end. To move between scenes, we needed to create public class which included public voids for each button in Visual Studio. To be able to attach
the code to the actual button, we needed to include the code to the button object in Unity.

5.5 Main Scene

This scene was designed for the gameplay. This scene consists of a grid in which different tiles can be placed, customers that can be collected, score, menu button, and arrows used to control our vehicle. Customers are human-like objects that are displayed in the scene. Driving through these objects improves the overall score.

![Main scene](image)

Figure 8 Main scene

Idea of the game is to construct a road from the car to the village house while collecting as many customers as possible on the way. Road can be constructed by using two road tiles displayed left on the screen from which the player can chose which one to place on the grid.

5.5.1 Driving

Originally in our demo for Sointu at the Games by Saimaa event, we did program car to be driven by using keyboard characters “W”, “A”, “S” and “D”. To
accomplish the movement, we needed to modify the car object’s positional vectors while specific key was pressed or held down.

```csharp
if (instantiatetepalikka.existingPositions.contains(pos))
{
    if (Input.GetKey(keyCode.W))
    {
        this.GetComponent<Transform>().Translate(0f, Time.deltaTime * 1.6f, 0f);
    } //if

    if (Input.GetKey(keyCode.S))
    {
        this.GetComponent<Transform>().Translate(0f, Time.deltaTime * -0.8f, 0f);
    } //if

    if (Input.GetKeyDown(KeyCode.A))
    {
        this.GetComponent<Transform>().Rotate(0f, 0f, 90f);
    } //if

    if (Input.GetKeyDown(KeyCode.D))
    {
        this.GetComponent<Transform>().Rotate(0f, 0f, -90f);
    }
} //if
```

Figure 9 Driving using W A S D

This made sense to us since we had been used to use these characters in other games we play. For elderly we decided to use different approach – drive the car by using visual objects to control the movement. To accomplish this, we needed to add arrows to our main scene in which we then harnessed the same logic as used with keyboard characters in earlier version.

Figure 10 Arrows used to drive the car
To avoid players from driving around without the road, we added a code to check if there was a vector (a road tile) underneath the car, if there was not, car would not move.

5.5.2 Tile Selection

There are two road tiles visible at the time from which the player can use a tile to be placed on a grid. These two road tiles are randomly selected from seven different options, including different 90-degree angles, crossroad, and straight road options. To identify which tile was selected out of the two, we highlighted borders of the tile selected with yellow.

![Figure 11 Two options for construction](image)

For tile selection we created a new public class TileSpawn. Logic for the tile selection includes checking whether the pointer is on the grid or not, considering the tile selection, pointer should not be on the grid, instead it should be over 2 possible options. Reason for checking whether the pointer is on the grid is that considering placement of the tiles, tile will be placed if the pointer is on the grid and mouse1 is pressed.

Also, logic includes the selection of a tile. To be able to select a tile, we needed to spawn two tiles which we could use as options.
To spawn to random tiles, we needed to create two random integers which would reset each time the selected tile was placed to the grid. We stored these values to the integers “rdm1” and “rdm2”. We then used these integers together with a switch to determine tiles to spawn for selection.

```java
// checks if there is already a tile
vector3 pos1 = new vector3(-1.9f, 1.3f, 0f);
vector3 pos2 = new vector3(1.9f, -1.3f, 0f);

// also if the mouse is not on the grid
if (!existingposition.contains(pos1) && (xhiri1 > 0 || xhiri1 < -9 || yhiri1 > 0 || yhiri1 < -9) && Instantiaappi.tilePlaced)

// upper tile randomization
int rdm2 = Random.Range(1, 0);
// lower tile randomization
int rdm1 = Random.Range(0, 1);

switch (rdm)
{
    case 1:
        GameObject aputett1 = Instantiate(this.tile, new Vector3(-9.9f, 1f, 0f), Quaternion.identity);
        aputett1.name = "tile1";
        spawn = aputett1;
        tapaus = 1;
        break;
    case 2:
        GameObject aputett2 = Instantiate(this.tile, new Vector3(9.9f, 1f, 0f), Quaternion.identity);
        aputett2.name = "tile2";
        spawn = aputett2;
        tapaus2 = 1;
        break;
    case 3:
        // code for case 3
}
```

Figure 12 Logic for tile spawn

Based on, whether the upper or lower tile was selected we set Boolean value true for upper or lower tile. Then we gave integer “tapaus” and “tapaus2” values which were earlier randomly selected for “rdm1” and “rdm2. The reason for these, setting Boolean value and storing the tiles shown for selection, is to be able to use these values while placing the tile on the grid.

### 5.5.3 Tile Placement

Selecting a tile would be useless if we have no means to place it into our playing grid. To create logic for this we created another public class – this was done to help us work on this separately from other areas.

To place the tile to the grid we needed to use the integer values stored in “tapaus” and “tapaus2” variables and the values on Boolean variables “ylempi” and “alempi” to determine which road tile the player had selected. In addition, to
avoid errors, our logic needed to perform a check whether the position on the grid was already taken.

```java
Vector3 pos = new Vector3(x, y, z);
if (positions.contains(pos) & (x > 8 || x < 8 || y > 5 || y < 5))
{
    if (Valinta.ylempi == true)
    {
        switch (tilesFromSpawn.tapaus)
        {
            case 1:
                GameObject spuitel = Instantiate(this.tile1, new Vector3(x, y, 0), Quaternion.identity);
                spuitel.name = "tile1";
                spuitel.gameObject.SetActive(true);
                pista.pistett = 2.5f;
                break;
            case 2:
                GameObject spuitel2 = Instantiate(this.tile2, new Vector3(x, y, 0), Quaternion.identity);
                spuitel2.name = "tile2";
                spuitel2.gameObject.SetActive(true);
                pista.pistett = 2.5f;
                break;
            case 3:
        }
    }
}
```

Figure 13 Logic for placing a tile

To accomplish the desired outcome, we used two IF statements, and two switches (one for each option “ylempi” and “alempi”). The first IF checks the grid, while the second one checks Boolean values to determine which tile was chosen, then based on the Boolean check a switch uses “tapaus” or “tapaus2” to select a correct tile to be placed to the grid.

5.6 Instructions Scene

Instruction scene’s idea is to help the player understand logic of the game. Instruction scene can be accessed from the start scene or through the menu in the main scene.

Our first version of the instruction scene was plain text explaining the idea of the game and how to play it. Then after one or two testing sessions with the target group, we received feedback that instructions were unclear. We discussed about the matter with our product owner - ReissuEllu representative and came to an agreement that we needed to develop animated instructions which would clearly demonstrate how the game was supposed to be played.
In the final version of the instruction scene, we have a combination of animation and text describing each phase of the game. Animations include a hand which demonstrates actions player is supposed to complete to play the game. Instruction scene consists of few scenes in order to demonstrate the gameplay in smaller segments. This way the required actions can be easily explained.

### 5.6.1 Animations

To create animated instructions, we used Unity’s own build-in animator. With the Unity animations, we were able to drag object, in this case a hand, from point A to point B and record the movement. To demonstrate pressing an object in a game, we needed to modify object’s vectors in order to rotate the hand.
After recording the movements, for the animation to be complete, we needed to synchronize rotational and two-dimensional movement to fulfill our desired outcome into a single timeline.

5.6.2 Score

Score can be used to compete with fellow players. In our project, displaying the score is used to motivate the player to complete the game with as few moves as possible while collecting as many customers as possible on the way. Placing a tile degreases score while collecting a customer increases the score.

To display the score while the game is running, we used canvas in Unity. Canvas is a tool to add User Interface (UI) elements, such as button or, for example, to display current score. In Unity canvas is created within the scene. Canvas and game objects are on a different scale to avoid confusing developer.

Positions of canvas elements are predetermined and attached to the camera showing the playing scene – meaning that even though the background in a scene would move, these objects on canvas would still be visible on the screen.
In this project, we used canvas in our main scene to display score, and menu buttons, and in our instruction scene to display text while in addition to the animations.

5.7 End Scene

End scene is the scene player sees when he finishes the game. Our end scene was simple as planned.

End scene displays the final score and congratulates the player for finishing the game.
6 CONCLUSIONS

Goal set at the beginning of the project was to develop a game that could be used to maintain or to improve mental health among target group – elderly people. To achieve the goal, we created a 2D puzzle game with randomized road tiles that needed to be connected to construct a road from point A to point B, and then drive a car through it. To add cognitive stimulus, mechanism that displays only two out of seven possible road tile options was created. Point of this was to challenge the player to adapt to the situation instead of just constructing road that they already had planned in their mind beforehand.

During our project it became clear that there were a couple of factors we needed to consider in development. These factors were illustrations of age-related conditions. These conditions were related to the way people detect objects. In addition to age-related conditions, another challenge was the intuitive gap between digital native developers and the target group, who were digital immigrants. To adapt to the needs of the target group, iterative development method was harnessed, and tests performed together with the target group.

Overall, the project can be considered a success as it achieved the goals set – being stimulative while also being fun. If a similar project was done for wider distribution, let’s say nationwide, paid team of professionals should be hired to accomplish more polished outcome with, for example, a database that could be used to store players’ score as a means of analyzing development among players. This data could then be used to study if cognitive development was made in the long run.

Games could be used more like this in the future as the digital native generations start to age meaning the gap between developers and elderly people is getting thinner which eases the development process.
REFERENCES


Drumond, Claire. Scrum. Available at: https://www.atlassian.com/agile/scrum [Accessed at 3 April 2021]


Radigan, Dan. Four agile ceremonies, demystified. Available at: https://www.atlassian.com/agile/scrum/ceremonies [Accessed 24 April 2021]


West, Dave. Scrum roles and the truth about job titles in scrum. Available at: https://www.atlassian.com/agile/scrum/roles [Accessed 17 April 2021]
LIST OF FIGURES

Figure 1 Agile development cycles ......................................................... 12
Figure 2 Development team ................................................................ 13
Figure 3 Scrum board example ............................................................... 14
Figure 4 Road tile ............................................................................... 19
Figure 5 Scenes .................................................................................. 19
Figure 6 Introduction scene ................................................................. 20
Figure 7 Code to move between scenes ............................................... 20
Figure 8 Main scene ........................................................................... 21
Figure 9 Driving using W A S D ........................................................... 22
Figure 10 Arrows used to drive the car ................................................ 22
Figure 11 Two options for construction ............................................. 23
Figure 12 Logic for tile spawn ............................................................... 24
Figure 13 Logic for placing a tile .......................................................... 25
Figure 14 Instruction scene ................................................................. 26
Figure 15 Animating hand movement ................................................ 27
Figure 16 End scene ........................................................................... 28
Figure 17 Using canvas ...................................................................... 28
TEHTÄVÄKSIANTAJA: “ReissuEllu huuvinvointia kehitettä liikkuvaa oppimisympäristö” - Essoten osahanke

TEHTÄVÄKSIANTO: ReissuEllu-pelin kehittäminen
ReissuEllu pelistä toivotaan valinnetta tukea huuvinvointia, lisätä liikunnallista aktiivisuutta ja edistää aivoterveyttä. Pelin muoto on vaapuasti vaikevassa Paikoina ilmainen lounas, Essoten tuotepalkinto sekä mahdollisuus jatkokehittää ja testata pelaajaa ReissuEllu-auton asiakastapautumaisuutta.

CUSTOMER: “ReissuEllu - wellbeing on wheels - a mobile learning environment” - South Savo Social and Health Care Authority Essote sub project
The duration of the Reissu-Ello - wellbeing on wheels- project is 1.6.2019 - 30.11.2021. The project is managed by Etelä-Savon Koulutus Oy and co-implemented by South Savo Social and Health Care Authority (Essote) and Mikkelin Development Company Miksei Oy. The project is funded by the ESF and ERDF.
The ReissuEllu project will build a mobile learning environment around wellbeing car, ReissuEllu, which will develop health, well-being and service guidance. ReissuEllu will travel in the Hirvensalmi, Juva, Kangasniemi, Mikkelin, Mäntyharju, Pertunmaa and Puumala areas and provide wellbeing services such as lifestyle counseling, service counseling, social work as well as pharmacist, physiotherapist and nursing counselor. The preventive aspect is emphasized in services.
ReissuEllu is designed to travel on standard routes in sparsely populated areas and participate in various events. Local actors such as village associations, other organizations and representatives of the municipality are invited to plan the routes. ReissuEllu is expected to reach older people in remote areas in particular, where other services are usually far from home.

ASSIGNMENT: Developing ReissuEllu game
The ReissuEllu game is hoped to be a tool to support wellbeing, increase physical activity and promote brain health. The game format is freely selectable.
The prize includes a free lunch, an Essote product award, and the opportunity to further develop and test the game at ReissuEllu customer events.