THE CASTLE ENGINEER GAME DESIGN DOCUMENT

27.12.2020

Lassi Perälä

TABLE OF CONTENTS

1. OVERVIEW

1.1 High Concept Statement

The Castle Engineer is a simple mechanical 2D game. The object of the game is to build a castle strong enough to survive the enemy attacks and keep the King alive.

1.2 Gameplay

The Castle engineer is a 2D game viewed from the side view (see FIGURE1). The gameplay involves two phases. In the first phase the player observes the map and builds a castle around the King unit. Player's ability to build is limited by budget that varies on the difficulty of the level.

When the player thinks that the castle is good enough to keep the king alive, the player can start the enemy attack. If the King survives the attack, the level is completed. If the king does not survive the attack, the player must start over again.

1.3 Game World

The game world will be kept as simple as possible. The background will be a blue image or an image of the sky and clouds. We will be using simple grass/cliff sprites as the ground of the game. The enemies can be found from the left side of the screen. The player may build the castle to the right side of the screen.

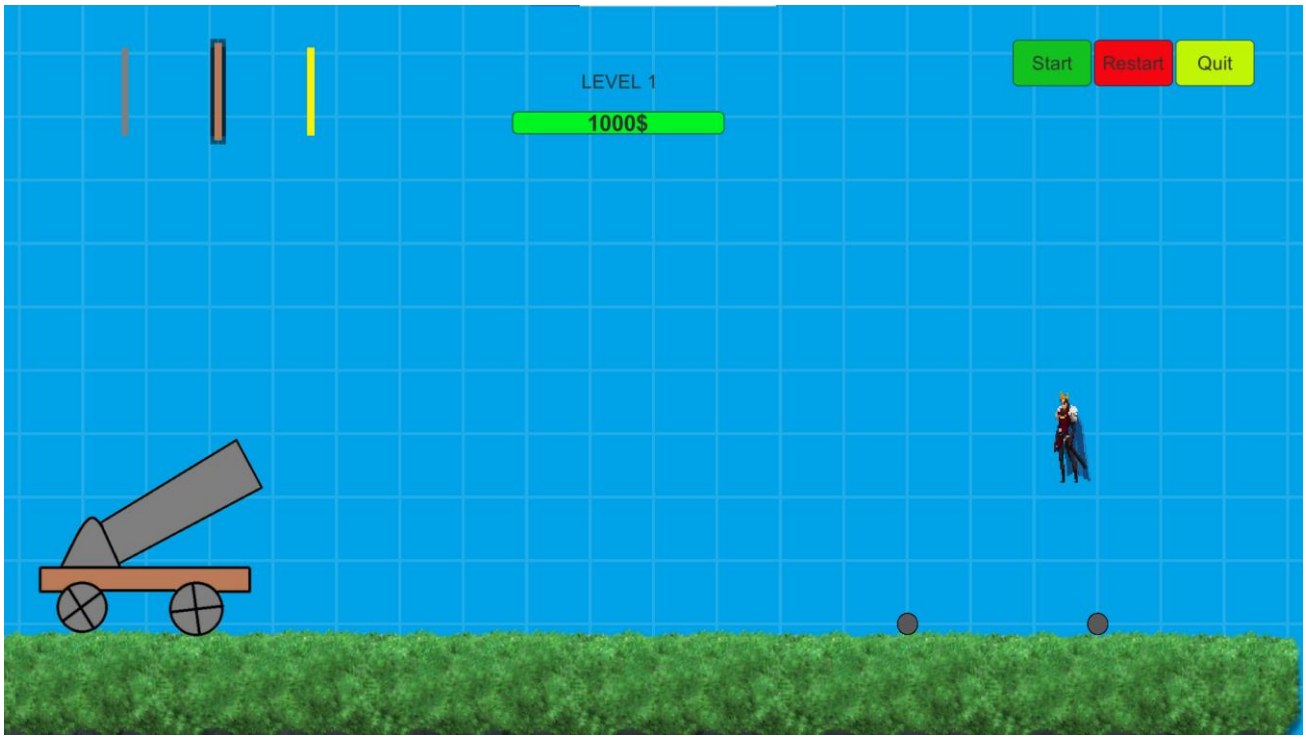


FIGURE1

1.4 Target Audience

The target audience of The Castle Engineer will mostly consist of players who are familiar with games such as Angry Birds. The Castle Engineer follows the same pattern but this time the player builds the defences and the AI is the one attacking. Also, players of games such as Poly Bridge could easily pick up The Castle Defender as the building mechanics are very similar.

1.5 Target Systems

The most suitable systems for The Castle Engineer are Windows/Linux environments. A version for mobile phones (Android/Ios) is also possible.

1.6 Competition

The competitor games will be pretty much the same as the target audience. In the worst-case scenario players will think that shooting the castle is more fun than building it and they will stick with games such as Angry Birds. The same thing goes with the building part. The players that love the building mechanics might find it more entertaining to watch their buildings in casual use of cars etc. instead of enemies attacking it.

1.7 Unique Selling Points

The key selling point of The Castle Engineer will be the flip of the roles used in popular games. Let us use the Angry Birds as an example again. Everyone knows how the game works; you have your ammunition and you must use it to destroy the enemy buildings. Expecting that the player knows how to play games such as Angry Birds, it is extremely easy to explain how The Castle Engineer works. It is the exact same thing but this time you are the one building the defences. You do not have to be an extreme Angry Birds fan to understand what you are supposed to do in The Castle Engineer. This makes the consumer's decision of picking up the game extremely simple.

2. GAMEPLAY & MECHANICS

2.1 Controls

2.1.1 Basic Controls

The game will be controlled with mouse. Some hotkeys for certain functions (such as the 2.1.2 Navigation Menu) might be added but the gameplay will still be highly mouse centred.

2.1.2 Navigation Keys

The top right corner of the screen will contain the navigation keys for each phase (See FIGURE2). Each level boots with the building phase. The player can build as long as he/she wants (building explained in 2.2.). When the player wants to start the Enemy attack (explained in part 2.3.), the player uses the left mouse key to select the Green "Start" button. This immediately starts the enemy attack and the player can no longer

build anything. If the attack goes horribly and the king dies, the player may use the red “Restart” button. This button stops the attack and puts the player back to the building phase. The player can quit the level at any point by using the yellow “Quit” button.

FIGURE2

2.2 Castle Building

The building of the castle will be done with a line drawing technique. Each level will contain two or more points connected to the ground. These are the points where the building of the castle should be started. As the player draws a line from the ground point, there will be another connection point in the end of the line drew. The new connection point can be used to draw another line. Left mouse button will start the drawing of a line and it can be cancelled with the right mouse button. The drawn lines will be made of materials chosen by the player (Materials explained in the section 3.2). These lines will be used to build the castle around the king. The FIGURE3 represents two drawn lines. The first line is connected to the ground point and the second line is connected to the connection point created by the first line.

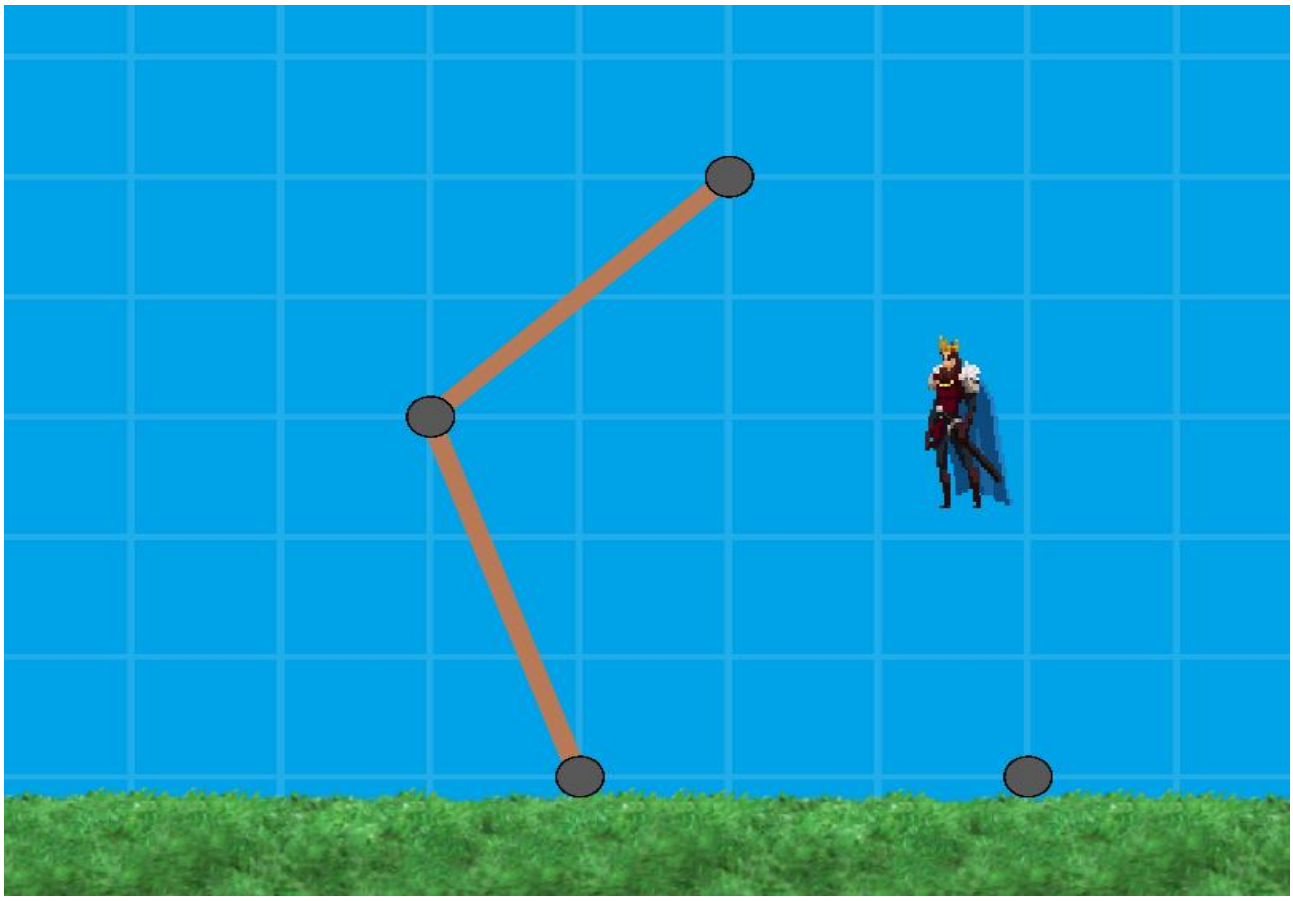


FIGURE3

2.3 Enemy Attacks

The attacks of the enemies can be started with the navigation keys as explained in the section 2.1.2. The attacks will contain varying enemies with different weapons.

2.3.1 Cannon

The Cannon is the main enemy in the current build of the game. The Cannon shoots varying projectiles (see section 3.1) for certain amount of times based on the difficulty of the level.

The Cannon itself consists of three parts:
-The base of the cannon (Left side of the FIGURE 4)
-The pipe of the cannon (Middle of the FIGURE 4)
-The actual cannon game object that does all the shooting (Right side of the FIGURE 4)

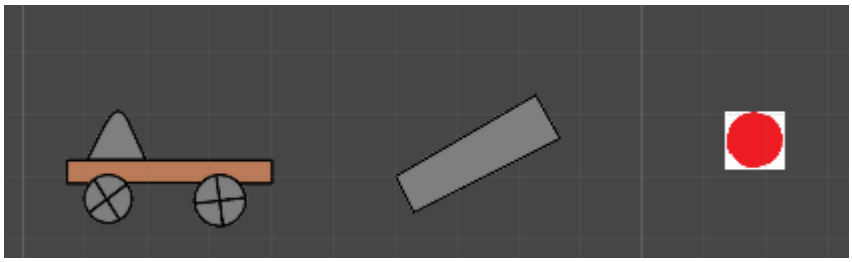


FIGURE4

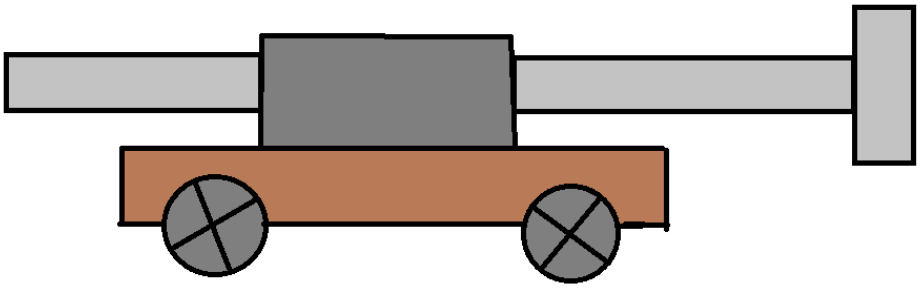
The first two parts of the cannon are completely cosmetic. They are separated into two parts just to give more possibilities for the level design. In some levels the king might be located on top of a hill, for example. In this scenario the cannon has to shoot a bit higher than on a levels that are completely flat. When the parts are separated, we do not need to create a completely new cannon sprite. We can just adjust the angle of the pipe part of the cannon, so that it looks like the cannon is actually aiming higher.

The last part of the cannon (red circle with white background) will be hidden behind the pipe part of the cannon. This is the actual game object that creates instances of projectiles and shoots them. The type and number of projectiles and the shooting angle will be adjustable for each level. Each shot will have a randomly generated angle to make the gameplay more unpredictable. This part of the cannon is separated just to make it useable on other types of enemies in the future.

2.3.2 Ideas

<u>The Wagon</u>

The idea for some sort of rolling wagon has been part of The Castle Engineer for the entire development. The Wagon would roll from the left side of the screen with certain constant torque to the wheels. Upon collision the wheels would lose all of their torque and the Wagon would keep on rolling and destructing as long as it can. A quick sketch of what The Wagon could look like:



The Dropper

The Dropper would be some sort of an object that flies over the castle. The Dropper would shoot a cannonball directly towards the ground when it is on top of the castle. This enemy type would put the roof structures of the castle to the maximum test.

3 GAME OBJECTS

In this section we will talk about rest of the game objects that the player will be interacting with in the current build of the game.

3.1 Projectiles

The objects that collide with the player-built castle will be called projectiles.

3.1.1 Cannonball

The Cannonball will be the most common projectile used by the cannons. The Cannonball looks like a bowling ball and it is shown in the FIGURE5. The Cannonball follows the physics of Unity's Rigidbody2d and has the mass value 3.

FIGURE5

3.1.2 Cannonball big

The Cannonball big is just a copy of the Cannonball. The Rigidbody2d mass of the Cannonball big will be 5.5 and the size will be slightly (around 1.5x) bigger.

3.2 Building materials

The materials are the different types of lines (castle walls) the player can create. The player can choose the materials from the materials menu shown in the top left of the FIGURE1. The main difference between the materials are the maximum length, breaking force, price and the collision properties.

3.2.1 Maximum length

Maximum length is the line's maximum length measured in the grid units. For example, the wood material has the maximum length of 5, so the longest possible wood wall takes up to 5 units as visualized in the figure 6.

FIGURE6

3.2.2 Breaking force

The breaking force property gives each material their unique strength. The breaking force is part of the Unity's Hinge Joint 2d component. As explained in the section 2.2, each drawn line is connected to two connection points. Breaking force equals to the minimum force required to make the connection between the connection point and the drawn line to break. Alternating the breaking force gives us the chance to make different kinds of materials.

For example, a wood wall has the breaking point of 350. If the collision force between a wood wall and a collider exceeds the number of 350, the connection points will break, and the wall falls down. On the other hand, Steel wall has the breaking point of 450. This means that a collision force that would break a wood wall might not have enough breaking force to break down a steel wall.

3.2.3 Price

Price is the value that is reduced from the current budget (the remaining amount of money left for use on a certain level) with each drawn line. The total price of a drawn line is calculated based on the material and the length of the drawn line in units. Let us observe the line drawn in the FIGURE6. The material is wood so the wall has the base price of 30. Because the drawn line is 5 units long, we will multiply the base price with the number 5. The price of the drawn line in the FIGURE6 would be 30x5=150. After this line has been drawn, the current budget would be reduced by 150.

3.2.4 Collision

The collision property declares if the wall material can be collided with. If the collision property of a certain material is false, the material is supposed to be used as a supporting structure. These types of walls will not take hits from cannonballs and the king can stand in front of these types of walls. The only material in the current build with false collision is the Back material.

3.2.5 Materials cheat sheet

This is a table that represents all the properties of each material.

Material Name	Max length	Breaking force	Price	Collision
Wood	5	350	30	true
Steel	3	450	50	true
Back	5	350	30	false

3.3 The King

The King is a game object that can be found on every single level of the game. The King might be located high in the air or low on the ground level. The Castle Engineer is all about building a castle around the location of the King and keeping him alive.

The King follows the basic physics as every other object in the game. The King dies when a certain object hits him with a high enough velocity. The main threat for the king is the cannonballs. The cannonballs only need a collision with velocity of 2 to kill the King. The King can also be killed by the falling walls of the castle or by dropping to the ground too fast. Dying in these two scenarios requires a collision velocity of 5.

3.4 Game Object Ideas

3.4.1 Projectile Ideas

The Cannonball Super

The size and mass properties of the Cannonball can be altered unlimitedly. One more very big and heavy copy of the Cannonball could be suitable for the game.

The Drill

The Drill would be an object that ignores certain building materials. The Drill would for example penetrate wooden materials but get directly stopped by steel. This type of projectile would force the player to enforce certain parts of the castle.

The Water Bucket

The Water Bucket would be a simple bucket that spills its' content on collision. The Water Bucket might not damage the castle heavily, but a good hit could fill the King's room with water resulting with the King drowning.

The Fireball

The Fireball would light wooden materials on fire which would make them weaker or even break them. The fire would also be lethal for The King.

3.4.2 Material Ideas

Stronger versions of the existing materials

The wall materials are very simple. Each material has its' unique properties. The properties can be altered without limits. For example, Titanium walls could be added with the highest breaking point and the highest price.

4 Front End

In this section we will go through things that might not be directly connected to the actual gameplay. Things such as menus, popups and tutorials.

4.1 Tutorial

The Castle Engineer will include a simple tutorial when the game is started for the first time. The tutorial will be kept as short as possible because we want the player to experiment and learn by playing the actual game. As the game is very easy to play, a tutorial level is not needed. A series of pictures, where each of the main features of the game will be explained, should be enough.

4.2 Menus

There will be couple of guiding menus in The Castle Engineer. The most necessary ones will be the level selection/main menu (FIGURE7) and the level completed menu (FIGURE8). Also, some sort of “Try Again” pop up might be added for situations where the King dies.

FIGURE7

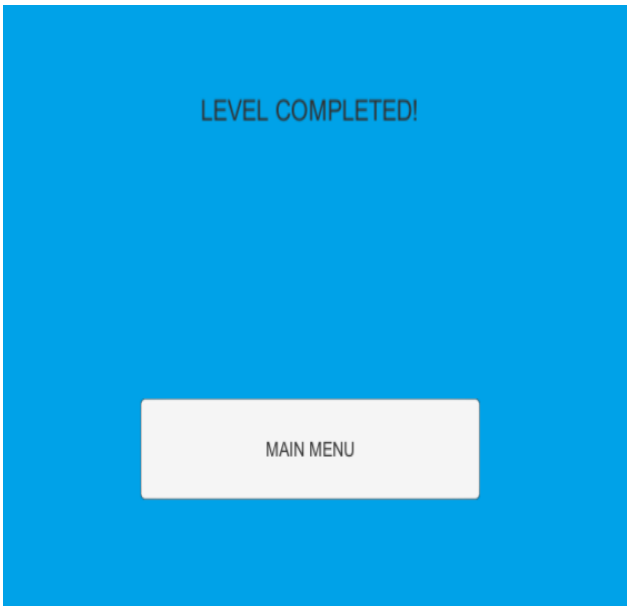


FIGURE8

5 Development Tools

At the current stage of development, free-to-use tools and software should have all the features needed for the game.

5.1 Unity

Unity will be the main platform for the designing of the game. As we have not received revenue/funding of over $100 000 in the last 12 months, we do not need to stress about the usage of Unity at all. The Castle Engineer can be published, even commercially, without any limitations.

5.2 Artwork

The spirtes, environments, backgrounds etc. will be done with Paint or similar free-to-use drawing software. The looks of the game are not the top priority in the current stage. Hiring artists/designers might be possible in the future, if needed.

6 Team

Project Manager	-	Lassi Perälä
Programming	-	Lassi Perälä
Art	-	Lassi Perälä
Design	-	Lassi Perälä
Producer	-	Lassi Perälä