



Alternative Energy

Development of an Android Serious Videogame to
change the mindset about the use of renewable energy sources
through a video game



Andrea Benito Fuentes - al341488

Final Degree Work
Bachelor's Degree in
Video Game Design and Development
Universitat Jaume I, Castellón, Spain
Hanzehogeschool University of Applied Science, Groningen, Netherlands
July 3, 2019

Advisors:
Antonio Morales

ABSTRACT

This Final Degree Project is based on the development of a serious videogame to show the advantages and disadvantages of the use of renewable energies. This work was done in the Game Lab program at Hanzehogeschool, being part of the Erasmus program 2018/19. The purpose of the serious game is to raise awareness on global warming, pollution and other problems that affect the Earth.

Because my client works with selling solar panels, he wants a videogame to convince people to invest in them.

Table of Contents

1. Introduction	5
1.1. Work motivation	5
1.2. Team	6
1.3. Client	6
1.4. Objectives	6
1.5. Related subjects	6
1.6. Planning	7
1.7. Expected results	8
1.8. Tools	9
2. Design	10
2.1. Inspiration	10
2.2. Summary	11
2.3. Concept	11
2.3.1. Concept for the Minimum Viable Product	12
2.3.2. Concept for the final product	12
2.4. Mechanics	12
2.4.1. Tap to build and navigate into the game	13
2.4.2. Zoom in and zoom out	13
2.5. Graphics	13
2.5.1. Video game environment	13
2.5.2. Video game UI	13
2.6. Gameplay	15
2.7. Game Flow	16
2.8. The Islands	16
2.8.1. Coal Island	16
2.8.2. Gas Island	16
2.8.3. Wind Island	16
2.8.4. Solar Island	16
2.9. The energy buildings	17
2.9.1. Types of energy buildings	17
2.9.2. Investing in energies / Unlock energy buildings	18
2.9.3. Upgrade cost	19
2.9.4. Money per tap	19
2.9.5. Destroying buildings	20
2.10. Other buildings	20
2.10.1. Parks and theme parks	20
2.10.2. Monuments	21

2.10.3. Police Station and Fire Station	21
2.10.4. Diner and shop	21
2.11. Monetization	22
3. Work development	23
3.1. Minimum viable product	23
3.1.1. MVP planning:	23
3.1.2. MVP implementation	23
3.1.2.1. Touch system	23
3.1.2.2. Structure Blueprint class	24
3.1.2.3. Build system	24
3.1.2.4. CreateEnvironment class	25
3.1.2.5. Camera setup	26
3.1.2.6. City stats / player	26
3.1.3. MVP results	26
3.1.4. MVP testing	26
3.2. Prototype	27
3.2.1. Prototype planning	27
3.2.2. Prototype implementation	28
3.2.2.1. Save and load system	28
3.2.2.2. Tutorial	31
3.2.2.3. Toon water shader	31
3.2.2.4. Day/night and weather system	31
3.2.2.5. Environment collider	32
3.2.2.6. UI Manager	32
3.2.2.7. Sound system	32
3.2.3. Prototype results	32
3.2.4. Prototype validation	33
3.3. Final version	33
3.3.1. Final version planning	33
3.3.2. Final version implementation	33
3.3.2.1. Quest system	34
3.3.2.2. Share button	35
3.3.2.3. Better explained tutorial	36
3.3.3. Final version results	37
4. Results	38
5. Deviations	40
6. Conclusions and future work	44
7. References	45

1. Introduction

1.1. Work motivation

Global warming is growing exponentially and this is notably affecting the environment. Many countries do not use renewable energies because of many different factors which include the pricing. Clean energy development is vital for combating climate change and limiting its most devastating effects. The Earth's temperature has risen by an average of 0.85°C since the end of the 19th Century. Renewable energies are sources of clean, inexhaustible and increasingly competitive energy. In fact, the use of renewable energies will have very positive economic consequences. According to the International Renewable Energy Agency (IRENA), doubling the renewable energy share in the world energy mix will result in additional global growth of 11% by that year (equivalent to 1.3 trillion dollars), an increase in wellbeing of 3.7% and in employment in the sector of up to more than 24 million people, compared to 9.2 million today.

The main advantages of clean energies are:

- Environmental benefits: Vital for combating climate change
- Inexhaustible
- Jobs and the economy
- More countries with energy: Tanzania, Niger, Sierra Leone, Burkina Faso, Central African Republic, Liberia, Malawi, Burundi, Chad and South Sudan are countries that have an average of 10% of people without energy.

This project was born from our concern on these problems: we see that our countries do not do anything to avoid global warming. After an exhaustive search about why we do not use renewable energies we figured out that people are not concerned about the environment and also because they think that new technology is expensive. The question that everybody asks and considers is whether it is worth the cost. However, it's not just the money. We need to make it part of the culture but most people are not ready for a change. The government is the best at facilitating change faster than it can naturally happen. Knowing that money is the best incentive for people to do anything, the government could potentially give those who use renewable energy a tax break or lower tax. And that is why our intention is to change the municipality council's mindset, giving them the power to build different types of energy sources, to try to avoid polluting the city.

The game will put the player in the main role of the story: the player is a proud mayor of a new city that he needs to create. The mayor can build different types of buildings and the player decides if he/she wants to be green or not. The mayor can unlock buildings and expand the city by doing quests or tasks and make the most beautiful island.

1.2. Team

Due to my study contract, this bachelor's thesis is done in a group in two subjects called *Game Lab 1* and *Game Lab 2*. With me, the team counts with 6 members: two artists, two designers and two programmers. I am the leader programmer in the group.

The group dynamic has been quite hard because the other group members are second-year students and they do not have the same experience as me. This point has made the development a little bit more complicated and slowed down the progress of the game: they still do not have the idea of using any game engine, only a little bit about Unity. This means that every small change in the project (as art or design) needs to be done by me, apart from the whole programming part of the game. Because of this, the other programmer could not program anything, so his job was to make the menu and make some particles.

I had a fluent communication with the rest of the team and I worked with the designers to decide what we need to do in every phase of the project.

1.3. Client

As a team, we had to choose two cases from among those proposed by the university of Hanze, one the preferred project we want to pitch for and the other one as a backup project. The concept of the game must be related to the chosen case. We decided to choose *Case 3 - Soleila* as the preferred project. Soleila is a company who develops solar parks. All the information that Hanze gave us about our client can be found in this link: <https://bit.ly/2Y3K0D4>.

In addition, Soleila recorded an explanatory video about what they want. The video is available in the following link: <https://www.youtube.com/watch?v=654LuzMvyyE>

Our backup project was *Case X* which allows us to choose our own assignment or implement our own ideas in the project. This case was discarded by the teachers when we presented the ideas.

1.4. Objectives

- To make a playable demo which faithfully represents the final game.
- To raise awareness of consequences that come with using certain energy resources, showing it in a fun way.
- To do the game accessible to people who usually do not play games with simple mechanics and UI.
- To include different types of energy resources: windmills, solar energy, gas extractors...

1.5. Related subjects

- Algorithms and Data Structures (VJ1215)
- Game Engines (VJ1227)
- Video Game Conceptual Design (VJ1222)
- Software Engineering (VJ1224)
- 3D Design (VJ1216)

1.6. Planning

Due to my role as programmer in a team, I divided the task in 4 blocks: *Research*, *Programming*, *Documentation* and *Ideas/Group meetings*. I think that it is not going to be necessary to do art because it is not my role.

In *Research* I put everything related to the previous research that I did for create the first concept: research about the situation of the renewable energies in different countries and why they did not use it. Also, I put here the playtesting sessions and Unity tutorials.

In *Programming* I will include everything related to Unity: create prefabs, modify scene, create a build, create scripts, implement the UI, etc. In other words, scripting and putting everything together (visual assets, mechanics and design choices). The Game Lab program in Hanze is divided in 2 phases, Minimum Viable Product (MVP) and Prototype. Because of that, I divided the programming task in 3 phases: Minimum Viable Product (MVP), Prototype and Final Game. The last one is when the classes finished, I will add changes to the project.

In *Documentation* I will put everything related to the thesis and other documents that I need. Memory task includes all the feedback that I will get from my advisors and create some diagrams or graphs which explain better the game or the idea.

In *Ideas/Group meetings* I will put everything related to the teamwork: meetings, feedback with the teachers in Hanze, presentations, meetings with our client (Soleila), etc.

TOTAL HOURS: 300

Research	
Task	Expected hours
Renewable energies and types of energies	4
Video Games about energy	4
Unity tutorials	20
Playtesting	10
TOTAL	38

Programming	
Task	Expected hours
Minimum Viable Product (MVP)	50
Prototype	50

Final game	15
TOTAL	115

Documentation	
Task	Expected hours
Technical proposal	4
Memory	60
Final presentation + preparation	5
TOTAL	69

Ideas / Groups meetings	
Task	Expected hours
First Brainstorming	3
First Concepts	5
Pitching the concept to Hanze's coaches to get greenlit	10
Questions and problems with Hanze's coaches	2
Meeting with Soleila's boss	3
Final Concept	20
Mechanics	15
Game loop	20
TOTAL	78

1.7. Expected results

The aim of the game is to change the mindset about renewable energies, making them interesting and also to show what will happen if we don't do something. We also want to teach about all the different types of energy and how can make them sustainable in long term.

We have two expected results: Minimum Viable Product and Final Product. The expected result for the Minimum Viable Product must be to finish the main mechanics (drag and drop, build, recollect...), to have the main scenario, to have a basic UI that allows the player to build and see the task that the player needs to do and finally to have both types of controls, touch and mouse-keyboard.

The expected result for the final products must be to have different stages, to have different task, the UI must be clear (for that we are going to do a testing session for getting feedback), and trying to publish the game in some digital store.

1.8. Tools

- Unity3D. The game engine to develop the game.
- Visual Studio. Code editor used alongside the game engine.
- Adobe Photoshop and Adobe Illustrator. Image editor for create and modify images and texture.
- Adobe Premiere. Video editor for create the footage of the game.
- 3DS Max. Software used to create some 3D assets.
- Google Drive. Used to compose documents and share files and media.
- Discord. Used to communicate with the team and scrum daily stand-up platform.
- Trello. Scrum weekly stand-up platform.
- GitHub. Version control, to backup the project
- Microsoft Word / Google Docs. Used to compose documents.
- Microsoft Excel / Google Docs. Used to create the breakdown of hours spent in the project.
- Draw.io. Class diagrams creation for Google Drive.

2. Design

2.1. Inspiration

The game is inspired in city builder/simulation games like *Cities Skylines*, *Clash Of Clans* and *The Simpsons: Tapped Out*. In *Cities Skylines* you can build a city with roads, police stations, monuments, parks, energy, water, etc, and it is really realistic because you can modify the taxes for the citizens, ask for a loan if you do not have enough money, create routes for the public transport, etc, so it is more like a simulation of a real city and the goal is to make your city successful. *Clash Of Clans* is a tower defense but it was an inspiration for two reasons: it is a game with microtransactions (but you do not need to pay for playing the game) and there are some elements that do not add anything, is only visual. In *The Simpsons: Tapped Out* there are some quest and activities that the citizens of Springfield need to do and you can build different types of houses and other emblematic buildings but some of them do not add anything new to the city (only visual as *Clash Of Clans*).

The other research that I did was to see how the art and the HUD of city builders on mobile are and I discovered that many of them looks crowded and is not easy for children and/or people who usually do not play games (see Figure 1).

Moreover, I investigated about games on Google Play which teach something about renewable energies but I did not find any game like our concept of a city builder game focused on green energies, I only found informative applications with text.



Figure 1: Comparison between different city builder mobile games. They look crowded, with too many elements on the screen and they are difficult to understand in one sight.

2.2. Summary

- **Game Title:** Alternative Energy
- **Game Genre:** Simulation, City Builder, Serious Game
- **Platform:** Android devices, PC with touchable screens
- **Intended Target Audience:** Kids between 8 and 12 years old
- **Game Summary:** Alternative Energy is a serious video game for Android smartphones and tablets aimed at teaching about renewable energies to its players and showing them the advantages and disadvantages of this.

The starting point of this project was Soleila, a dutch company that sells solar panels, and they wanted a tool that helped them to discuss about how they desire to generate the energy needed in the municipality meetings.

The video game puts the player as mayor of a new island where he / she has to decide what type of energy is going to apply to his/her island. In this island there are other factors like pollution, happiness, money, etc.

- **Unique Selling Points:** Freemium, ads, microtransactions, fun, cartoony, customizable.

2.3. Concept

The concept has changed a lot during the different phases. At the beginning, we presented two ideas to the Hanze's coaches for the first part in the Game Lab program in which we showed them our ideas for the game and gave us the greenlit to continue with that concept.

- **First idea (Case X):** A cooperative game. The concept was based on Overcooked game. We had four characters: a coal miner, a solar panel engineer, a nuclear scientists and windmill engineer. Each character had different abilities and the objective was to build more energy sources than the others. There are two different endings, the good one that is a world without pollution (when the solar panel engineer or the windmill engineer wins) and the bad one that is a world all polluted (when the nuclear scientist or the coal miner wins). At different levels there were different complications but also different powers or events that makes the game funnier.
- **Second idea (Case 3 - Soleila):** A city builder game. The player is the last hope of the city. It is his/her job to power the city and get people to move back in. Use both solar panels, wind turbines, coal factories and gas extraction sites to supply energy to the houses.

As I said at the beginning of this document, the first case/idea was discarded by the teachers when we presented the project ideas, so we kept the second one. During the Minimum Viable Product, the idea changed, I will explain this changes later.

Once we had the concept of a city builder video game, the concept had little changes between the Minimum Viable Product and the Prototype.

2.3.1. Concept for the Minimum Viable Product

In the first phase of the project, the game is a simple city builder game with only one city. The user will play as the mayor of an island. At the beginning of this phase, the concept of the game was a small polluted city and the objective was to replace all the coal factories with renewable energies to reduce the pollution. When the user reduce the pollution, the city expands because there is more space to live without polluted air, something like the circle storm of Fortnite (is a gameplay mechanic in the game *Fortnite Battle Royale* which makes people to stay inside the eye of the storm) but instead of reduce the circle, expands the circle when the pollution is reduced. People can only live inside the eye of the storm because is the only unpolluted zone. The player has limited resources and he/she has to use them wisely, you also have to keep people happy You play the game in a time span of 10 years (according to experts, it's the time we have now in real life to do something to reduce pollution) and in the end you will see how good you did it.

Later on we discarded this idea and started with an empty island instead of a polluted island. In this island, the objective is to have the best city and make the citizens happier. The user need to build different types of energy (instead of replacing), choosing between renewable energies and non-renewable energies. The user can see the consequences of his/her actions, learning the differences between them.

The goal of the game is to change the mindset about the use of renewable energy sources through a video game giving information and seeing the consequences

Furthermore, there are a button which says facts about renewable and non renewable energies, in order to raise awareness.

We changed the concept because we realised that it is not funny for a video game and because of the feedback we got in the playtesting sessions.

2.3.2. Concept for the final product

In this part of the process, the concept changes a little bit. In this phase we created different islands and the game looks more like a common city builder game. The concept of the islands are explained in further detail in the following sections.

Moreover, we add a new functionality, the research tab. Instead of having all energy buildings unlocked, the user must invest in investigate energy sources to unlock it. When the player unlocks the new energy building, he/she can build it in your island. The user needs to increase the level of the island to expand the city. When he/she reaches a new level, the user gets new buildings as rewards.

The game also incorporates a share button to encourage to have the most beautiful island and to share it with your friends.

2.4. Mechanics

The target devices for the game are devices with touchable screens. This can be an Android mobile phone, an Android tablet or a Windows PC with touchable screen. The game can be also expanded to run in PCs with mouse, but that is not the main goal.

Due to the nature of the target devices, the mechanics of the game are focused on the normal use of a touchable devices: tap and pinch mechanics.

2.4.1. Tap to build and navigate into the game

One of the core mechanics of our game is placing buildings. The player can increase the energy of the city by placing coal factories, gas extraction sites, solar panels and wind turbines but also can increase the happiness of the city with other buildings and decrease the pollution. The tap mechanic is also the way to navigate into the different menus and interact with the game for example collecting bolts (our game's currency) when it spawns..

2.4.2. Zoom in and zoom out

Another mechanic is zoom in and out. The player is able to zoom in and view each building's details or zoom out and view the entire island, showing the progress of the provided energy. The player will also be able to see what they have created in its entirety.

2.5. Graphics

2.5.1. Video game environment

The game takes place in an island. The graphics remain the same throughout the different islands, with slight changes in the UI and the environment. In the different stages of the game development this has changed a lot (see Figure 2).



Figure 2: Comparison between the MVP final result of the game at the left and the prototype final result of the game at the right. In the first one the city was a plane object with green colour, but in the second one the city was a 3D model of an island, with some elements like rocks and grass.

2.5.2. Video game UI

The user interface shows information about your city (pollution, energy, happiness, money, etc) at a simple glance. The UI keeps static throughout the game and most of its elements can be interacted with at any given time. As you can see in Figure 3, this is the UI elements that there are always visible. Starting from left to right and from top to the bottom:

- **Level:** The level of the current island. The circle bar and the number indicates how many energy your city has.
- **Stats:** Stats include the pollution bar and the happiness bar. The smiley face next to the happiness bar changes depending on the value of happiness.
- **Calendar:** A timer that indicates which day is in the game.
- **Money:** The total money that the player has.
- **Pause button:** Opens the pause menu with three options, which are resume the game, go to the menu and quit the game.
- **New quest:** Gives to the player new quest when is tapped until there is not more.
- **Active quest:** Opens a window with all the quest and the player can claim the rewards there.
- **Demolish button:** Enable the demolition of the buildings.
- **Build menu:** Opens an emergent menu to build energy sources or other buildings.
- **Travel to islands:** Saves the current state of the island and goes to the travel scene, which allows you to go to other island.



Figure 3: A brief explanation of the static UI in the game.

Furthermore, the game allows you to build energy sources in two different ways: one when you press the build menu button once (see Figure 4) and another when you tap in all the buildings menu button. The last one is found in the build menu and then in the "currency" button. A new window appears and show different tabs for the different buildings (see Figure 5).



Figure 4: Simple build menu. This appears when you press once the build menu button. To see all the buildings and the research window, you need to click on the last button in this menu.



Figure 5: Expanded build menu. This shows all the options in different tabs. The first one refers only to the four types of energy, the second one all the monuments, the third one the rest of the buildings and the last one is the research window.

2.6. Gameplay

The essential game loop is pretty straightforward. There is a main island which connects every island. Travelling from one island to another requires to unlock the island. Islands are explained in further detail in section 2.8 of this document . You can get money and buy different types of buildings which has different consequences in the stats of the city.

When the player reach a new level of the city, are unlocked different buildings as reward. This buildings can help to reduce the pollution or to increase the happiness of the city.

2.7. Game Flow

The game takes place into a new island, so the island does not have any buildings, only a few neighbourhoods of people that need energy to life there. The player can place the town hall which gives 100 000 bolts and can invest in research the first type of energy which is shown with a small tutorial. To emphasize on the consequence of each type of energy, the player can unlock first the pollutant energies and later on the renewable energies. The first invest the player can make is on the coal factories. Once the player invests in the research of coal factory, the player can build coal factories, invest in the gas extractors and unlock the first island. There is one island for each energy source, which the player unlocks through investment. The user needs the energy buildings to provide the city with power. It is up to he/she which energy sources he/she uses and which energy source he /she upgrade.

By upgrading and leveling up your main island you unlock new buildings, styles and colors for his/her current buildings and rewards that include bolts (our game currency). This will helps him/her to be able to customize his/her island to however he/she wants, and to make it his/her own.

2.8. The Islands

Player starts with one island and he/she can unlock up to 5 islands: main island, wind island, solar island, coal island and gas island.

2.8.1. Coal Island

The island has the coal town hall. Player must reach certain amount of energy to level up city (which means expand the city). Player increases energy by placing coal factories. When player levels up city, player can upgrade the main coal building. When main coal building is upgraded, coal factories give more money. Player can make people happier by placing a build for workers to relax, in this case, a diner/cafeteria.

2.8.2. Gas Island

Same idea as the coal island, but this island is used to make more money with the gas buildings. Here, the relax place is a food truck.

2.8.3. Wind Island

Same idea as the coal island, but this island is used to make more money with the windmills. Here, the relax place is a cafe.

2.8.4. Solar Island

Same idea as the coal island, but this island is used to make more money with the solar panels. Here, the relax place is a restaurant.

2.9. The energy buildings

As I said before, the game has four different types of energy sources. Depending on which the user wants to build, it has different consequences.

2.9.1. Types of energy buildings

- **Coal factory:** The first energy building that you can unlock.
 - Happiness: -5
 - Pollution: 10
 - Energy: 10
 - Description: Pollution raises the most when placing coal factories due to the amount of CO₂ that is released

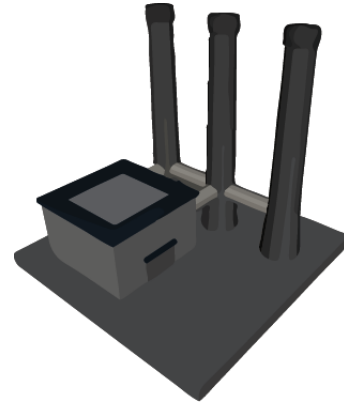


Figure 6: Coal factory 3D model

- **Gas extractor:** The second energy building that you can unlock.
 - Happiness: -7
 - Pollution: 8
 - Energy: 10
 - Description: Happiness decreases the most when gas extraction sites are placed due to the problems linked to fracking

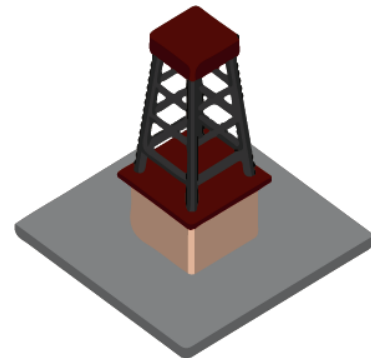


Figure 7: Gas extractor 3D model

- **Windmill:** The third energy building that you can unlock.
 - Happiness: 3
 - Pollution: 1
 - Energy: 10
 - Description: By placing wind turbines you are barely adding to the pollution due to the low amounts of CO₂ they release

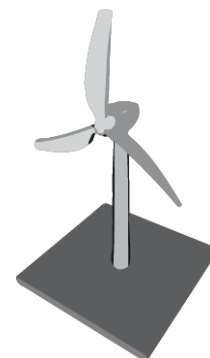


Figure 8: Windmill 3D model

- **Solar panel:** The fourth energy building that you can unlock
 - Happiness: 5
 - Pollution: 2
 - Energy: 10
 - Description: Placing solar panels gives you the most energy, money per tap and happiness due to their efficiency and low maintenance

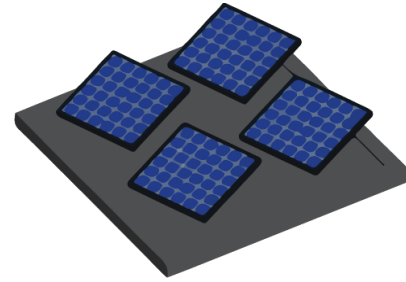


Figure 9: Solar panel 3D model

2.9.2. Investing in energies / Unlock energy buildings

To be able to unlock the different islands, the player must invest in the four different energy sources (see Figure 10). The player will start with coal. Once he/she has invested in coal, the coal island is unlocked. This also unlocks the ability to start investing in gas, which would lead to unlocking the gas island, which unlocks the ability to invest in wind, which unlocks the wind island and then unlocking the final energy source, solar (see Table 1).



Figure 10: Research window with the coal factory unlocked. The user can invest in gas extractor and when is unlocked, they also unlock the invest in windmills.

Building	1st invest cost	2nd invest cost	3rd invest cost	4th invest cost	Unlock
Coal	20 000	20 000	25 000	27 000	Gas island Invest in gas
Gas	25 000	27 000	30 000	32 000	Wind island Invest in wind
Wind	30 000	32 000	35 000	37 000	Solar island Invest in solar
Solar	35 000	37 000	40 000	42 000	-

Table 1. Cost per invest in a energy source

2.9.3. Upgrade cost

All the energy sources can be upgraded with a cost (see Table 2). This upgrade gives more money when it spawns. To upgrade a building the user needs to tap in the desired building and a pop-up menu will appear with two options, information and upgrade.

Building	Level 1	Level 2	Level 3	Level 4
Coal	-	5 000	7 000	10 000
Gas	-	7 000	10 000	12 000
Wind	-	10 000	12 000	15 000
Solar	-	12 000	15 000	17 000

Table 2. Upgrade cost for the different energy sources and levels

2.9.4. Money per tap

Depends on the level of the building, the player will get a different amount of money when it spawns (see Table 3).

Building	Level 1	Level 2	Level 3	Level 4
Coal	1 000	1 500	2 000	2 500
Gas	2 000	2 500	3 000	3 500
Wind	3 000	3 500	4 000	4 500
Solar	4 000	4 500	5 000	5 500

Table 3. Amount of money per tap, depends on the level of the building,

2.9.5. Destroying buildings

Destroying buildings costs half of the price that was paid to build it, so for example, to destroy a coal factory it would cost 10 000 bolts. When a building is destroyed, all the stats go back as they were before the building was placed, except for the money. The energy, the pollution and the happiness go back as before. The user can demolish everything except the town hall and the houses.

Destroying cost:

- Coal factory: 10.000
- Gas extractors: 12.500
- Windmills: 15.000
- Solar panels: 17.500
- Parks: 500
- Monuments and other buildings: 20.000

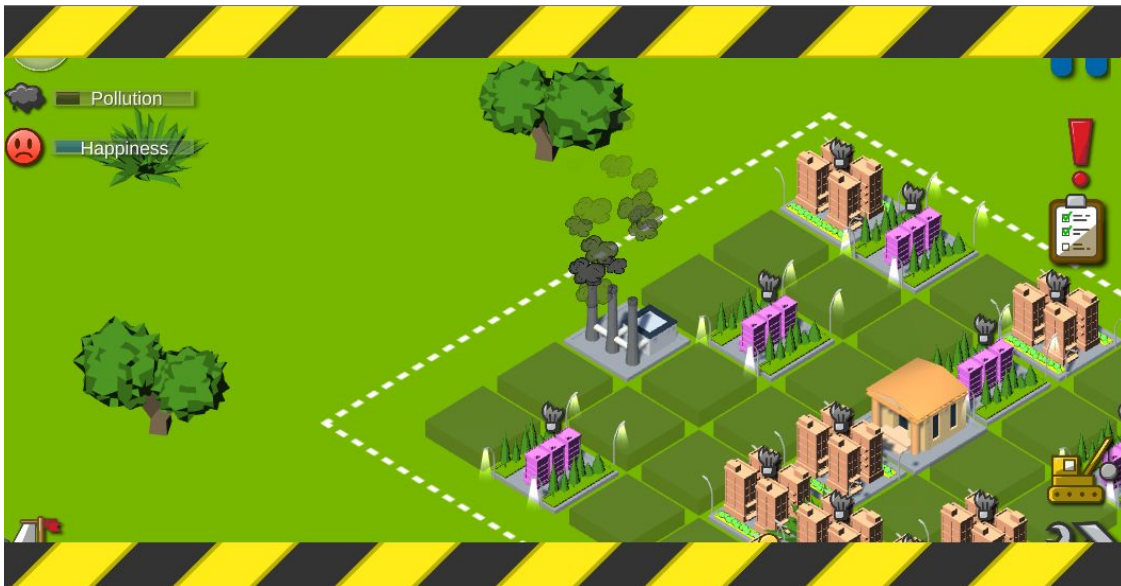


Figure 11: Destroy screen when the user press the demolish button in the bottom right corner.

2.10. Other buildings

For making the game more funny, I add some different buildings. Due to the way I implemented the build system, it is easy to add extra buildings.

2.10.1. Parks and theme parks

The theme parks only increase the happiness of the city but parks increase the happiness and reduce the pollution.

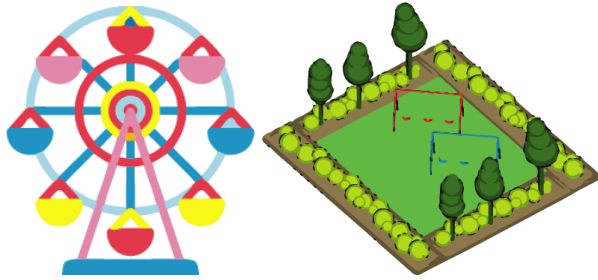


Figure 12: Icons for the wheel fair and the park.

2.10.2. Monuments

This buildings did not add anything to the game, they are only visual. For now, the monuments in the game are the Eiffel Tower, the Statue of Liberty and the Pisa Tower.

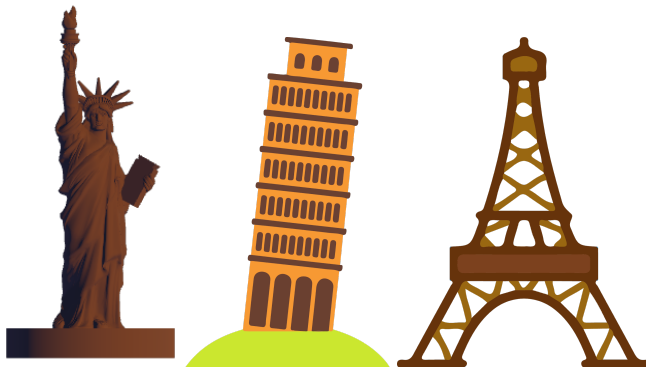


Figure 13: Icons for all the monuments in the game.

2.10.3. Police Station and Fire Station

I did these buildings for add something more to the game and show my 3D art skills. These buildings only increase the happiness of the citizens because they are more protected but need a little bit of energy.

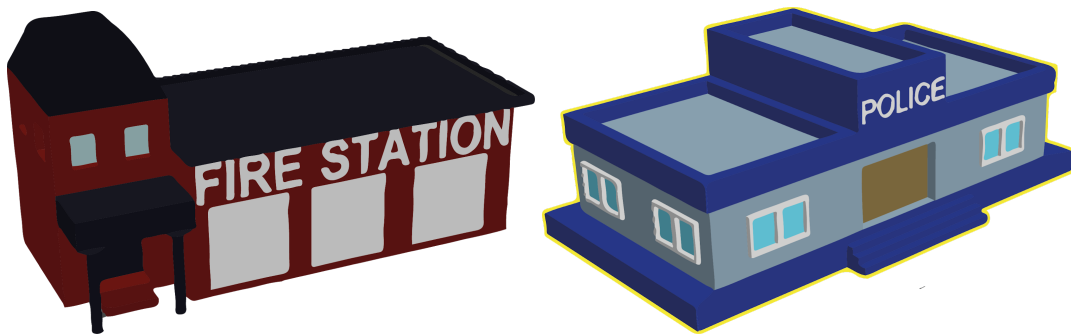


Figure 14: Icons for the Fire Station building and Police Station building.

2.10.4. Diner and shop

These buildings reduce the time of spawn the currency in the energy buildings because workers are happier and work better. The diner and the shop are the only buildings available in the other islands (not in the main island).

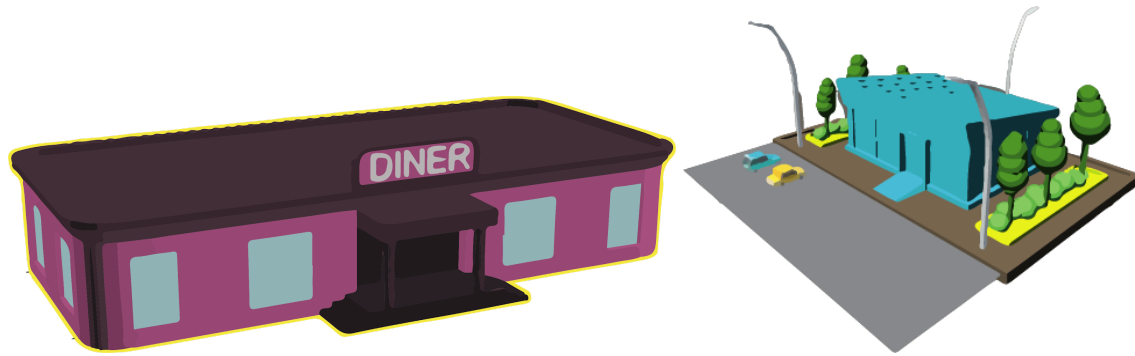


Figure 15: Icons for the Diner building and the Shop building. Both reduce the time of spawn money.

2.11. Monetization

The monetization part of the game is not implemented at this time, but it will be. We were thinking in something like *Clash Of Clans* or *The Simpsons: Tapped Out*. These games has an extra currency which is gems for the first one and bagels for the second one . In our game we want to implement a new currency call batteries. With this new currency, the user can get customizable items for the game, such as special buildings in special events like Christmas, Halloween, etc. Moreover, the user can get free batteries watching adverts in the game.

3. Work development

The development of the video game was entirely carried out by me. The development was divided in three phases, each one incrementally adding and improving the experience. Each phase consisted on three parts: planning, implementation and validation/verification through play testing.

The phases are:

1. Minimum Viable Product (MVP)
2. Prototype
3. Final videogame

The two first phases was developed in Hanze with my team, but the final part corresponds to the things that I wanted to implement in my own bachelor's thesis with the feedback that I received from my teachers in the Netherlands and the classmates in the demonstration day.

3.1. Minimum viable product

The minimum viable product is the smallest unit of a videogame that represents the full extent of its mechanics and possibilities. In this part we are not focus on making the video game beautiful or finished, the focus of this first implementation was to test if the users liked the idea. Also in this part, we had a different target audience than the one we end up with because our client said something different that they really want. In the documents and videos that they gave us (explained in the section 1.3 of this document) we understood that Soleila wanted a tool to present their ideas to the council members: we decided to make a tool for the meetings with the government to change the mindset and to convince them to use renewable energies.

3.1.1. MVP planning:

The Minimum Viable Product phase of the project consisted of:

- A functional city in which you can build different buildings
- Zoom in and zoom out and navigate into the scene
- Different stages for the different levels of the city
- A basic task system
- First UI to reflect the consequences of each decision

3.1.2. MVP implementation

In this phase, the most important part was to have the core mechanics implemented. Also I was focused on creating code which was easily to expand and/or to add new things.

3.1.2.1. Touch system

For touching in the different places in the city, each node has a *NodeTouch* script, which check if it is being touched and if it is, check another parameters in that node. For doing the touch system I needed to create a method call *IsPointerOverUIObject* that returns

true or false, depending on if the user is touching an UI element or not. This is necessary because otherwise the user would touch a GameObject or a building and a button in the UI at the same time.

Inside the method *OnMouseDown()* we check if it is an UI element and if not what will happen with that node. This script calls the *BuildManager* script.

3.1.2.2. *Structure Blueprint class*

It is the base class of the buildings. It contains all the information that we need for each building:

```
public class StructureBlueprint
{
    public string title, description;
    public Sprite icon;
    public idBuildings id;
    public GameObject prefab;
    public int levelBuilding = 1;
    public int levelResearch = 1;
    public int[] costResearches = new int[4];
    public int[] costUpgrades = new int[4];
    public int cost;
    public float timeMoney;
    public int moneyPerTap;
    public int amountOfEnergy;
    public int amountOfHappiness;
    public int amountOfPollution;
    public int costOfDemolition;
}
```

Figure 16: The *StructureBlueprint* class. Here you can see all the information that is necessary to place a building in the city.

There is another auxiliar script, called *InventoryBuilding*, which stores each type of building (which is a *StructureBlueprint*) and creates an ID to identify it. This class never gets destroyed because the information is the same for all the islands.

3.1.2.3. *Build system*

When the user clicks the buy building button, the information of that building is stored in the *structureToBuild* variable in the *BuildManager* script. After that, the user needs to select the desired location for the building, checking if the node is available or not in the *NodeTouch* script. If the user taps in one node occupied by another building, this shows the options for that building which are information and the option to upgrade the building.

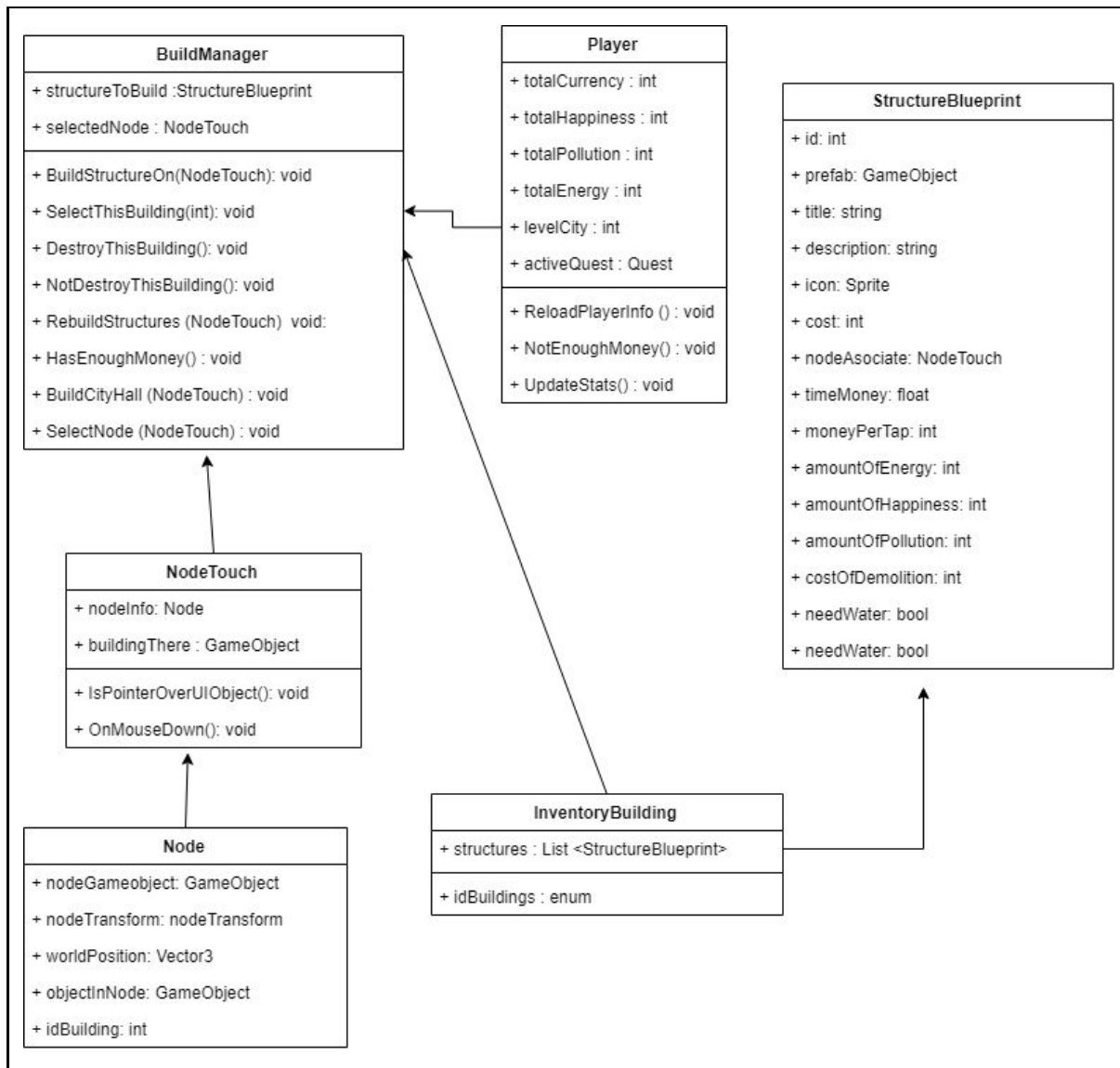


Figure 17: UML class diagram of the build system.

3.1.2.4. CreateEnvironment class

This part of the code is the part that creates all the necessary components of the island. It spawns the houses/neighborhoods, creates all the nodes in the island, creates the stage depending on the level of the city, etc.

At first, the nodes are created and stored in a matrix. Once created, the stage (the amount of nodes which are unlocked depending on the level of the city) is shown and the rest are disable. Then, the houses are spawned and the game starts.

In this script there is also methods to expand the amount of nodes when the city reaches a new level.

3.1.2.5. Camera setup

This type of games usually has an orthographic camera because the user wants to see everything at a glance and we do not need that our objects have depth. This camera can only move in the X and Y axis and can zoom in and zoom out depending on the ortographicSize of the camera. I decided to use this type of camera because it is the most common type for city builders and strategy games.

3.1.2.6. City stats / player

There is a script called Player which stores all the information related to the island: money, happiness, pollution, etc. Also here is stored all the active quest in the game.

3.1.3. MVP results

This phase took around 2 months and a half to complete. Every part of the code was carefully thinking because I wanted to have a good base that allows to implement more functions, scripts and buildings easily for the next block.

In the link you can see the first apk and the first build of the game: <https://drive.google.com/drive/folders/1kJHIVME2EzBTLydlrZ3euh0s93im1DII?usp=sharing>

Note: This PC build only works with touchable screens.

3.1.4. MVP testing

We did two playtesting sessions: The first one (demonstration day) with our classmates and teachers and the second one (thinking aloud testing) with random people that they don't usually play video games.

- **Demonstration:** The 9th of April we did a playtesting session in the university with our classmates and teachers (see Figure 18). Here, we could see other Gamelab's project.
- **Focus group and thinking aloud testing:** The 15th of April we did a testing session with eight people that they do not usually play video games. In this session, we tested it with two different methods, the focus group and the thinking aloud. In the thinking aloud session we gave them an introduction of the game and we wanted to let the testers play the game individually with a list of task to be completed in the session, and watched them testing to see if the game works properly and if there were something missing. As developer, I was focused on the system but I also got feedback about the design of the game and new ideas that they gaves me.

After the thinking aloud session, we did the focus group. In this part, all the testers were in the same room with pizza and drinks and we asked them questions about the game. It is here when we got many ideas and they helped us to planning the further phases. Here is some of the feedback that we got is the following:

- Too much clicking when trying to build something

- Money too fast
- Confused about the different buttons (quest, info, pause, exit, etc)
- Was not able to finish the “expanding the city” task without help
- Did not see the purpose of the different statistics
- Overlapping menus
- Suggestion about hide menus when you click elsewhere
- Suggestion about animations and sound when you place a building
- More statistics about pollution and happiness (put numbers)

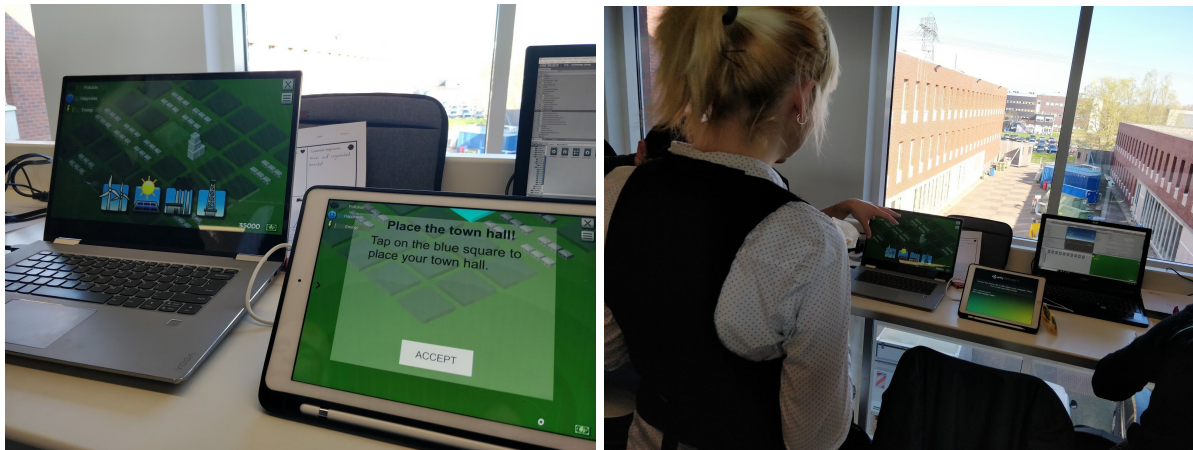


Figure 18: The game during the demonstration day.

3.2. Prototype

After the development of the MVP the next step was the refinement of the systems in place, addition of better graphics, UI, objects and implement some particles in the game. This new additions and improvements were based on the feedback obtained through the focus group playtesting session and by following the feature set originally envisioned.

Now, we focus on the fun part of the game, so the player starts in the main island and can unlock upto 5 islands: main island, wind island, solar island, coal island and gas island. In the main island, the player can place the town hall and can buy the cheapest type of energy. Once the city has expanded, the player unlocks new island, the coal island.

3.2.1. Prototype planning

For this new phase, I was focus on the following task:

- Fix the errors of the playtesting session
- Improve the art
- Improve the task and rewards system
- Change the build system without too many clicks
- Save and load system
- Tutorial
- Make it funnier

- Day and night system
- A day timer
- Weather system
- Publish on Google Play
- Music and sounds
- VFX

3.2.2. Prototype implementation

In this phase I improved and updated scripts when it was necessary and I implemented new features such as the save and load system and the tutorial.

3.2.2.1. Save and load system

To allow the players to leave the game and play again when they want, I created a save and load system (see Figure 19) that creates a .dat file in the device, saving the required values (see Figure 20) and loading when the user press play or press the travel button to travel to another island (see Figure 21). There are different methods for the different elements the game needs to store.

These values are the current state of the city, the total money, the stats of the city (pollution, happiness, level, etc...), the achieve quest, the day and the hour.

When the player goes back to the menu or travel to another island, the *Save()* function is called and the .dat file is created. In this file we store the matrix of nodes with the id of the building.

When the game starts or exists a save file for that island, the *Load()* function is called and set the corresponding parameters, instantiate the buildings, reload the date in the game, etc.

The .dat file is stored in different parts depends on the device. If the game is running in a Windows PC, the *Application.persistentDataPath* points to %userprofile%\AppData\Local\Packages\AlternativeEnergy\LocalState. If the game is running in an Android device, the *Application.persistentDataPath* points to /storage/emulated/0/Android/data/com.GasCan.AlternativeEnergy/files. On my mobile, I can see those files in the following path: Storage>Android>data> com.GasCan.AlternativeEnergy > files

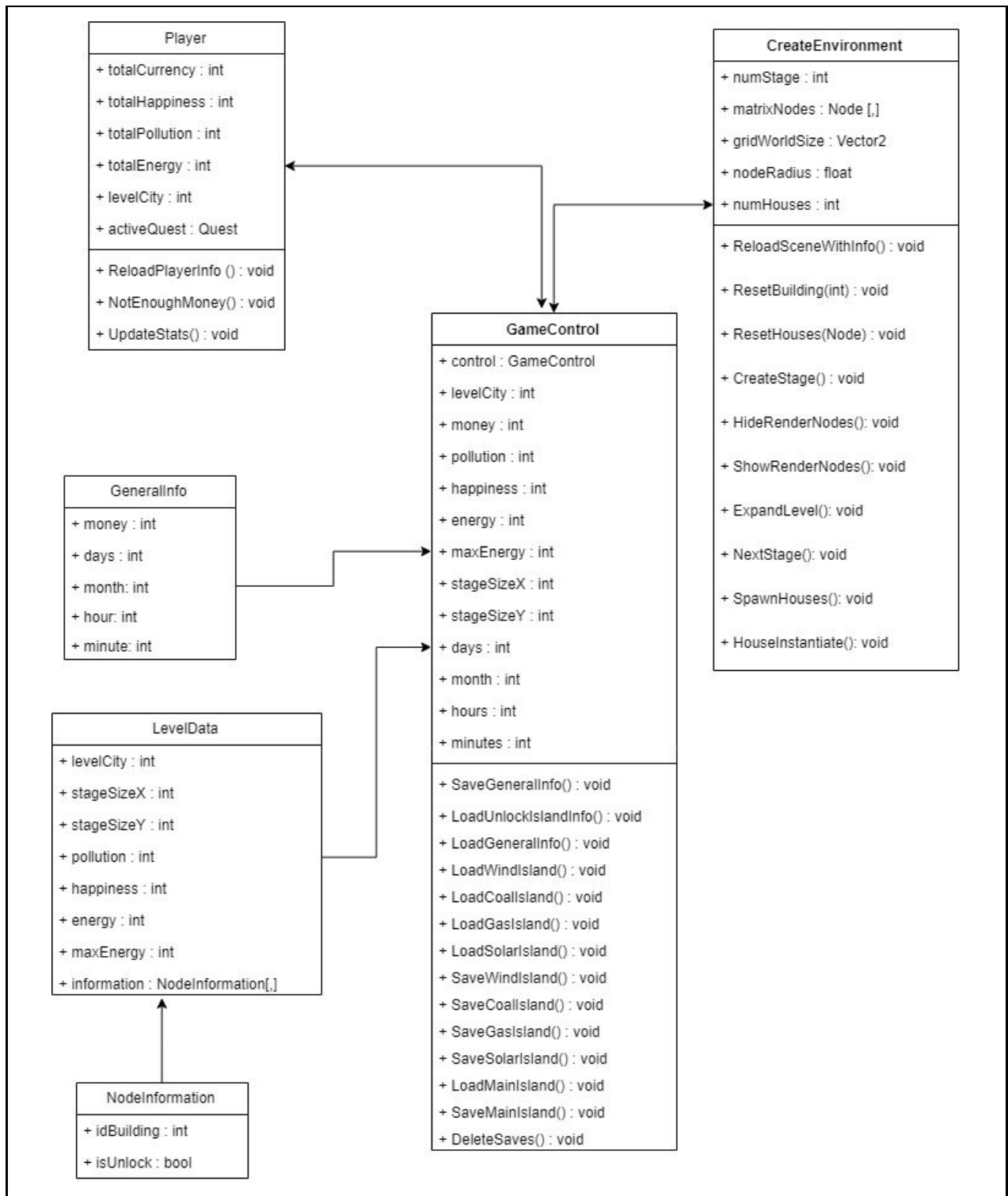


Figure 19: UML class diagram for the save and load system

```
public void SaveGeneralInfo()
{
    BinaryFormatter bf = new BinaryFormatter();
    FileStream file;
    file = File.Create(Application.persistentDataPath + "/generalInfo.dat");
    GeneralInfo info = new GeneralInfo();
    CreateEnvironment ce =
GameObject.Find("GameManager").GetComponent<CreateEnvironment>();
    WeatherController weather =
GameObject.Find("GameManager").GetComponent<WeatherController>();
    Player player = GameObject.Find("Player").GetComponent<Player>();
    info.money = player.totalCurrency;
    info.days = weather.fakeDays;
    info.hour = weather.fakeHour;
    info.minute = weather.fakeMinutes;
    info.month = weather.actualMonthNumber;
    info.night = weather.isNight;
    info.tutorial = tutorial;
    info.unlockIslandCoal = player.unlockCoal;
    info.unlockIslandGas = player.unlockGas;
    info.unlockIslandSolar = player.unlockSolar;
    info.unlockIslandWind = player.unlockWind;

    bf.Serialize(file, info);
    file.Close();
}
```

Figure 20: Save general info of the scene.

```
public void LoadGeneralInfo()
{
    if (File.Exists(Application.persistentDataPath + "/generalInfo.dat")) {
        BinaryFormatter bf = new BinaryFormatter();
        FileStream file = File.Open(Application.persistentDataPath + "/generalInfo.dat",
        FileMode.Open);
        GeneralInfo info = (GeneralInfo)bf.Deserialize(file);
        file.Close();
        money = info.money;
        days = info.days;
        minute = info.minute;
        hour = info.hour;
        month = info.month;
        night = info.night;
        tutorial = info.tutorial;
        unlockIslandCoal = info.unlockIslandCoal;
        unlockIslandGas = info.unlockIslandGas;
        unlockIslandSolar = info.unlockIslandSolar;
        unlockIslandWind = info.unlockIslandWind;
    }
}
```

Figure 21: Load general info when the file exists

3.2.2.2. *Tutorial*

When the game starts it checks for a file and load it. If the file doesn't exist, it won't load anything and it means that the player is starting the game for the first time, in which case a tutorial would be displayed.

To show this tutorial, a script called *Tutorial* works only when the game is starting for the first time. Some arrows show to the player where to tap in order to learn how to invest in the first energy resource, the coal factory.

The flow of the tutorial is as follows:

1. Game starts in day 0 without money
2. The user needs to tap in the blue square to place the town hall
3. An arrow shows how to open the simple build menu
4. Another arrow shows how to open the expanded build menu
5. Another arrow shows how to open the research window
6. The last arrow shows how to invest in the coal factory
7. Now the user can play the game as usual.

3.2.2.3. *Toon water shader*

The game takes place in an island

The game takes place in an island and for being an island must have water around it. To implement this water I used a tutorial that I found on the internet. This tutorial creates a shader for create materials inside Unity. But this tutorial use the perspective camera, so I needed to change the code of the shader because I used an orthographic camera and the shader needs the depth. I needed to figure out how this shader works and which was the part of the code I need to change. I add the last line of the Figure 22 in the *ToonWater* shader.

```
...
float4 frag(v2f i) : SV_Target {
    // Retrieve the current depth value of the surface behind the
    // pixel we are currently rendering.
    //float existingDepth01 = tex2Dproj(_CameraDepthTexture,
    UNITY_PROJ_COORD(i.screenPosition)).r;
    float existingDepth01 = tex2D(_CameraDepthTexture, (i.screenPosition)).r;
    ...
}
```

Figure 22: Change the way to get the depth of the shader because of the configuration of the camera.

3.2.2.4. *Day/night and weather system*

I added a timer where the player can see what day is in the game and what time is. For now I put fixed parameters: the day starts at 07:00 and ends at 22:00, rains every 5 days and each month has 30 days. When it is night, the light in the game is less intense than when is day (see Figure 23). In future implementations, after every month, the player will receive a

statistic report showing what he/she has done in the last month. In the second month, it will show your statistics and compare them to the month before. This also helps with the weather changes and seasons. I also would like to add an event to the game (Halloween, Christmas, ...) and in these events he/she can unlock special buildings and/or skins for the buildings in the city. The players will then be able to see when the next event will happen.



Figure 23: The city when it is night.

3.2.2.5. Environment collider

For making the islands more realistic, I created a script that destroys the environments models if collides with a new building. Each environment has a collider and detects the placement of an object.

3.2.2.6. UI Manager

With the intention to regroup everything related for the UI, is added a new script, the *UIManager*. Here I add all the methods that I need for updating the UI.

3.2.2.7. Sound system

To avoid the overlapping of sounds, I created a script called *SoundManager* which stores all the sounds and clips that I want to play, like sound effects when the user taps a button or a demolish effect when he/she destroys something.

3.2.3. Prototype results

The prototype phase took around a month and a half to complete. In this phase, the game was closer to being a finished game and it was uploaded and released in beta on the Google Play Store. I created a Privacy Policy of the video game because it is a requirement to upload the game in the store. The Privacy Policy can be found here: <https://bit.ly/2JhpXb4>

3.2.4. Prototype validation

The final testing session took place on the 18th of June to show the prototype to our teachers and possible investors. We get the following feedback from the coaches in Hanze and we passed the game lab demo with a 9.1:

- The city felt alive now, compared to the MVP phase.
- The game has certain simplicity so that potentially would work well for your target audience.
- The 2D (UI) and 3D art is more coherent art style. Visually, the game looked good.
- The extra islands added a lot more fun to the game.
- All the building mechanics worked very well.
- Hard to determine what I needed to do with the separate islands and how this would benefit my main island. Tutorial for that?
- End goal? Is there something for the user at the end of the game? Or is there maybe more of a background story to why he/she wants to keep building? Who am I helping?
- Monuments are a fun feature but since they do not give the player any bonus, I would communicate this more clearly.
- Benefits per island, UI element that shows the amount of bonus the player get for my main island.
- Monetize the game.

3.3. Final version

This phase started outside the project in the University, doing the rest of the changes on my own. Taking into account all the feedback that I got from my teachers, I decided to make some changes.

3.3.1. Final version planning

The final version of the game included the following improvements and additions:

- Show a feedback screen when the user tap on the statistics button
- Different types of quest with different goals: money, place a specific number of solar panels/windmills, increase happiness, etc.
- Loading screen with different tips/facts.
- Improve the tutorial.
- Change the menu.
- Share on Facebook, Twitter, etc, button
- Implement ads

3.3.2. Final version implementation

First of all, I fix some errors that I noticed in the demonstration day of the prototype phase. Then, I improved other functions in the game like the quest system.

3.3.2.1. Quest system

In the prototype phase, the quest system was not finished yet, it only worked with quest that its goal was to earn money and the reward was also money. In this phase, I created a new window to show all the active quests (see Figure 24), I added different goals (see Figure 25) and each island has different quest.

For this system, I created different scripts: *Quest*, *ActiveQuest*, *QuestGiver* and *QuestGoal* (see Figure 26). The *Quest* script is like the *StructureBlueprint* script I explained before: it contains all the information that we need for each quest. The *QuestGiver* stores all the quest for that island. When the user clicks in the exclamation button, he/she will receive a new quest. The user will click on the accept button and this quest appears on the all the active quest window. Then, when the player finishes the quest, he/she can press in the claim reward button, get the reward of that quest and this quest disappears.

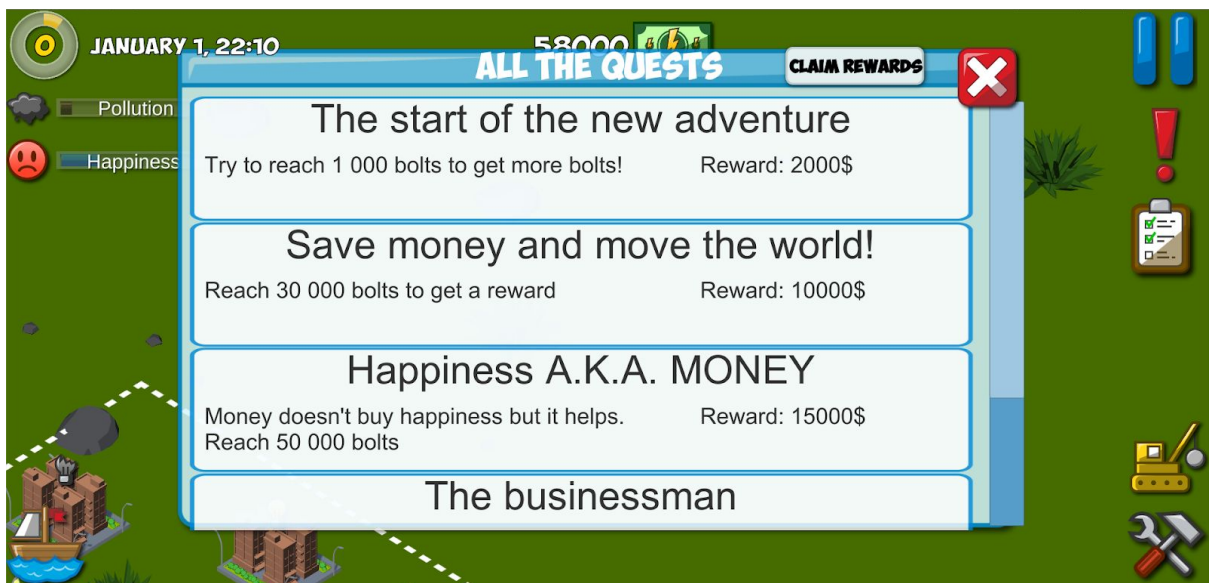


Figure 24: All the active quest in that moment. When the user press the clipboard button he/she will see this window.

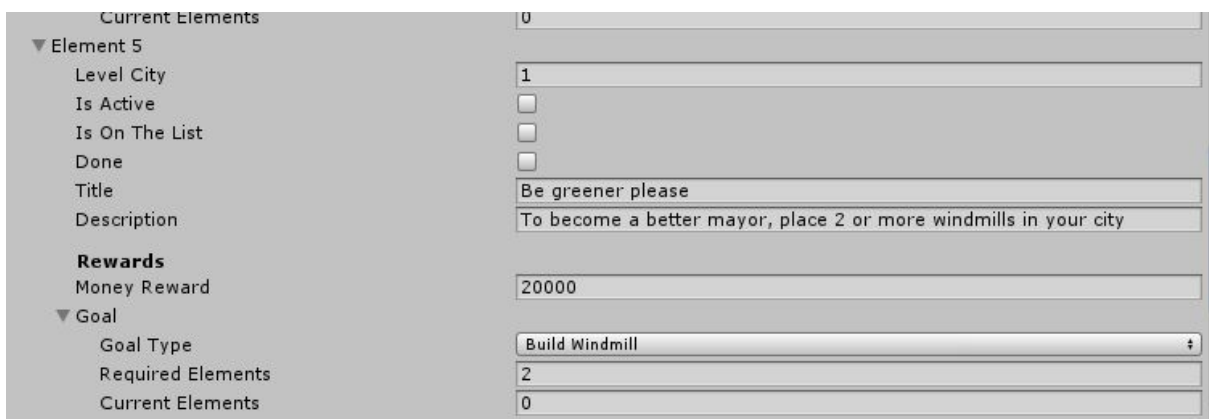


Figure 25: A screenshot of the *Inspector* in Unity3D of one of the *Quest* elements in the *QuestGiver* object to show how easy is to change every quest. The *Level City* variable

indicates which level of the city the user needs to unlock this quest. In the *Goal Type* variable you can select between different options: *Build Windmill*, *Build Solarpanel*, *Money* or *Happiness*. The *Required Elements* variable indicates how much money/happiness the user needs or how many windmills or solar panels the user needs to build.

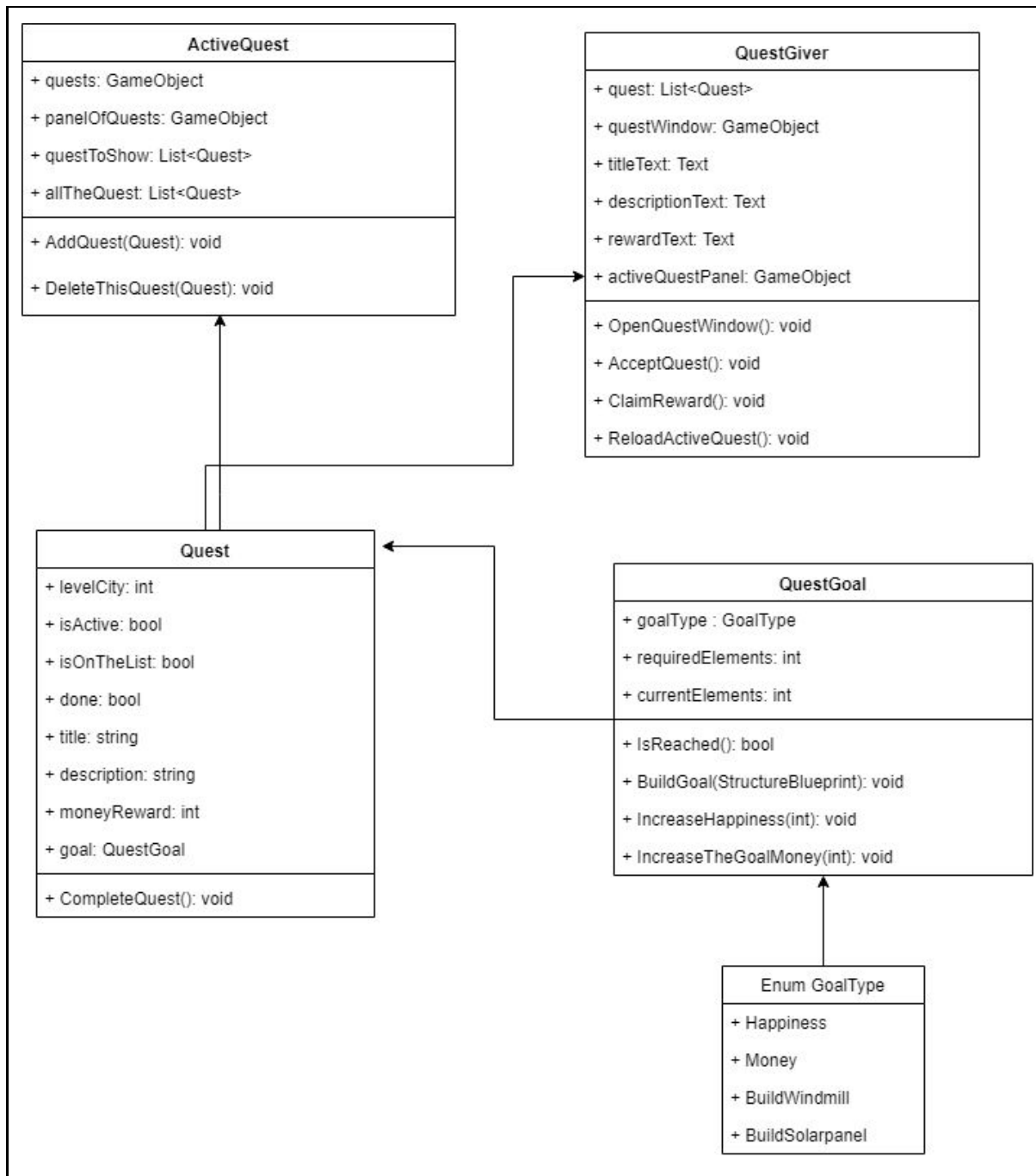


Figure 26: UML class diagram for the quest system in the game.

3.2.2.2. Share button

In order to allow the player to share their progress, I added a share button which make a screenshot of the city and open an emergent Android menu which allows to share it in any

application with a prefixed text. To do this, I needed to modify the Android Manifest of the project because we need special permission to do it. This feature allows to Android devices to share the screenshot everywhere: WhatsApp, Twitter, Drive, Instagram, Facebook, Gmail, LinkedIn, Discord, Telegram, Outlook, Bluetooth, Snapchat, etc. (See Figure 27)

Like the save and load system, the screenshots are stored in the same path, so you can see in that folder the .png file. The name of the screenshot has the following format: *Screenshot_YEARMONTHDAY_HOURMINUTESECOND.png*



Figure 27: Example of how it looks when you share the progress of the game in WhatsApp.

3.2.2.3. Better explained tutorial

I added some instructions at the beginning of the game because I saw that many people have problems understanding what they need to do (see Figure 28). Moreover, I disable some buttons in the tutorial, to avoid the user press in the incorrect button.



Figure 28: Example of the window that appears to explain what the user needs to do.

3.3.3. Final version results

This final version phase started after the 18th of June, the last demonstration day. As I said before, this part is done outside the team, so each new functionality has been performed by me including new buildings (diner, police station and fire station) and new UI elements for the tutorial.

The game can be released now but due to the contract I signed with my team it has been decided to keep it in beta so that when it is really finished we release it.

4. Results

This is the final game and it has been released on the Google Play Store for Android devices to download it and play it for free (for now). Up to 2000 people can access to the early access of the game in Google Play with an Android device using the following link:

<https://play.google.com/apps/testing/com.GasCan.AlternativeEnergy>

Or you can play in PC using the following link:

<https://drive.google.com/file/d/12UdTUncDLngbxQocWJgjiDeblnvCYiuY/view?usp=sharing>

You can access to the whole project in this GitHub repository:

<https://github.com/andreabefuen/AlternativeEnergy-Project.git>

Alternatively, I did a gameplay video of the final version of the game:

<https://youtu.be/jngokgJGm2w>

And finally, here you can see the evolution of the project during the different phases and the playtesting session: https://youtu.be/qfju6XW9_tU

I tested the game in my mobile phone, an OnePlus 5T device, in a friend's phone, a Xiaomi Mi A1 device, in a team member's phone, a Samsung S8+ device, in two different Windows PC with touchable screen (see Figure 29) and in a tablet, a Samsung Galaxy Tab S2. Because the tablet is older than my phone and because of the different specifications of these two devices, I realised that the game runs better in my mobile (see Table 4).

The apk is prepared for Android versions equal or superior to Android 4.1 JellyBean but I could not try it in previous versions to Android 6.0 Marshmallow because nobody has an older version of Android.

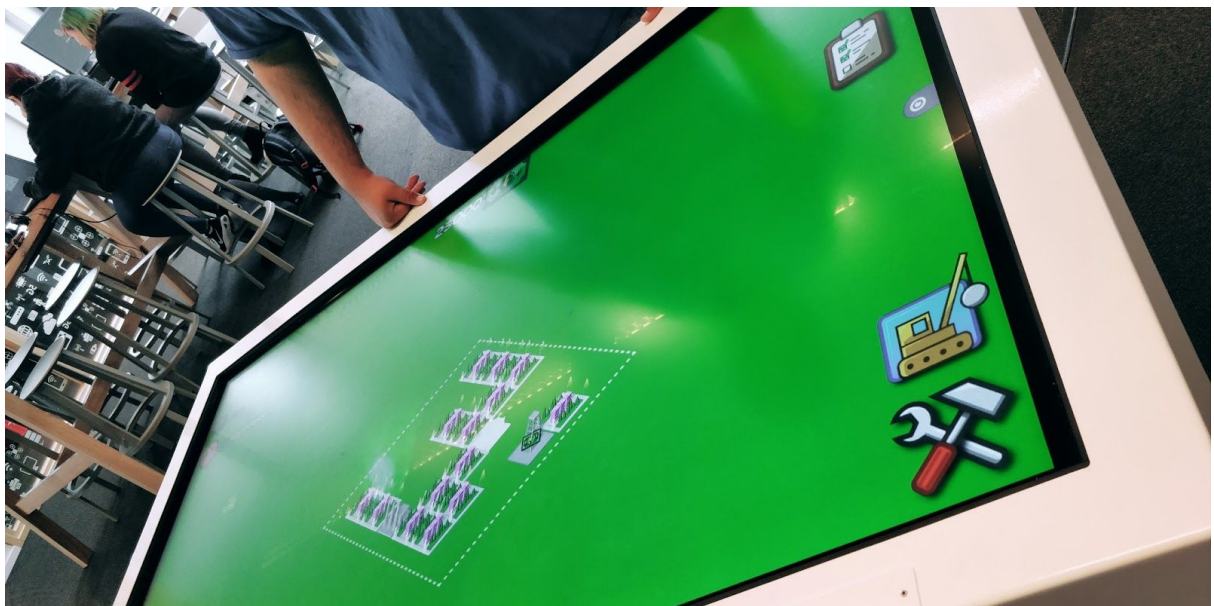


Figure 29: A table PC with touch screen in the University of Hanze. We tested the game there and it was working excellent .

Device	RAM	Processor	Android Version	Performance
Xiaomi Mi A1	4GB	Snapdragon 625	8.0 Oreo	Excellent: Animations and particles are fluid
OnePlus 5T	8GB	Snapdragon 835	9.0 Pie	Excellent: Animations and particles are fluid
Samsung S8+	4GB	Snapdragon 835/Exynos 8895	8.0 Oreo	Excellent: Animations and particles are fluid
Samsung Galaxy Tab S2	3GB	Snapdragon 652	6.0 Marshmallow	Normal: Animations and particles appear fluid but not too much

Table 4: All the Android devices and their specifications where the game was tested.

As of now one of the limitations that need to be addressed is that the game has not any online infrastructure, only the share functionality to make a screenshot of the current state of the player's city, but the expected results of the development mentioned in 1.7 section of this document were met. The video game:

- Is a playable demo which represents a beta version of the final game.
- Can raise awareness of consequence that come with using certain energy sources, showing it in a funny way.
- Is accessible and simple for people who usually do not play games.
- Includes different types of energy sources.

5. Deviations

In the first section of this document (see *Planning* section) an initial assumption about the possible scope of the project has been made. In the development of the project the different feedback we get, the new requirements and some problems showed up and in consequence, the initial planning needed to be changed. In order to keep track of the hours invested in the project, I made an Excel file to record every time I put myself to work as well as to know what tasks I was doing. The breakdown of all the hours dedicated to the project can be found in this link:

https://docs.google.com/spreadsheets/d/1TMOJH2ZxBwu9GYNKzshP_YLSVbUcqjYBc-XNCzF8Ve0/edit?usp=sharing

The amount of hours that I planned in the *Planning* section had changed and some of the tasks that I first imagine for the game. Here are the reasons of why these deviations:

- MVP phase: During the coaching sessions with our teachers, they pushed us to change a lot the concept of the game. Due to this, I started the game more than one month later that I expected and the game in the first phase was not as developed as I wanted.
- Prototype phase: I did not expect to do everything in the project, only the programming part, I thought that the rest of the team knew something about Unity and they could touch small things in the project, so I also spent time in details such as change colors, add textures, move UI elements, etc.
- Final version phase: I tested the game more times and I found errors so I tried to fix it. Also I created more art and some elements like the loading screen to make the game more appealing for the user.

Comparing the planned hours with the hours actually used, we can see that the *Design/Meeting* part remains almost without changes, while the *Programming* part has more hours than expected (see Figure 30). In general I have invested more time in the programming (almost half of the total hours) than in the other parts of the project, as I expected (see Figure 31).

In the following chart we can see every task with the expected hours, the final hours and deviations:

Research			
Task	Expected hours	Final hours	Deviations
Renewable energies and types of energies	4	1'5	-2'5
Video games about energy	4	2	-2

Unity tutorials / Assets	20	8'5	-11'5
Playtesting sessions and demonstrations	10	20	+10
TOTAL	38	32	-6

Programming			
Task	Expected hours	Final hours	Deviations
Minimum Viable Product	50	30	-20
Prototype	50	96	+46
Final game	15		
TOTAL	115		

Documentation			
Task	Expected hours	Final hours	Deviations
Technical proposal	4	5	+1
Memory	60		
Final pitch + preparation	5		
Videos	-	4	+4
TOTAL	69		

Ideas / Group meetings			
Task	Expected hours	Final hours	Deviations
First Brainstorming / Meeting	3	2'5	-0'5
First Concepts	5	6	+1

Pitching the concept to Hanze's coaches to get greenlit (GATE 1)	10	4	-6
Questions and problems with Hanze's coaches (Coach meetings)	2	4'5	+2,5
Meeting with Soleila's boss	3	3	0
Final Concept/ Mechanics / Game loop / Design / Task	55	56'5	+1'5
TOTAL	78	76'5	-1.5

Art			
Task	Expected hours	Final hours	Deviations
Create new buildings	-	4'5	+4'5
TOTAL	-	4'5	+4'5

Deviations



Figure 30: Comparison between the hours planned and the hours actually invested in the project.

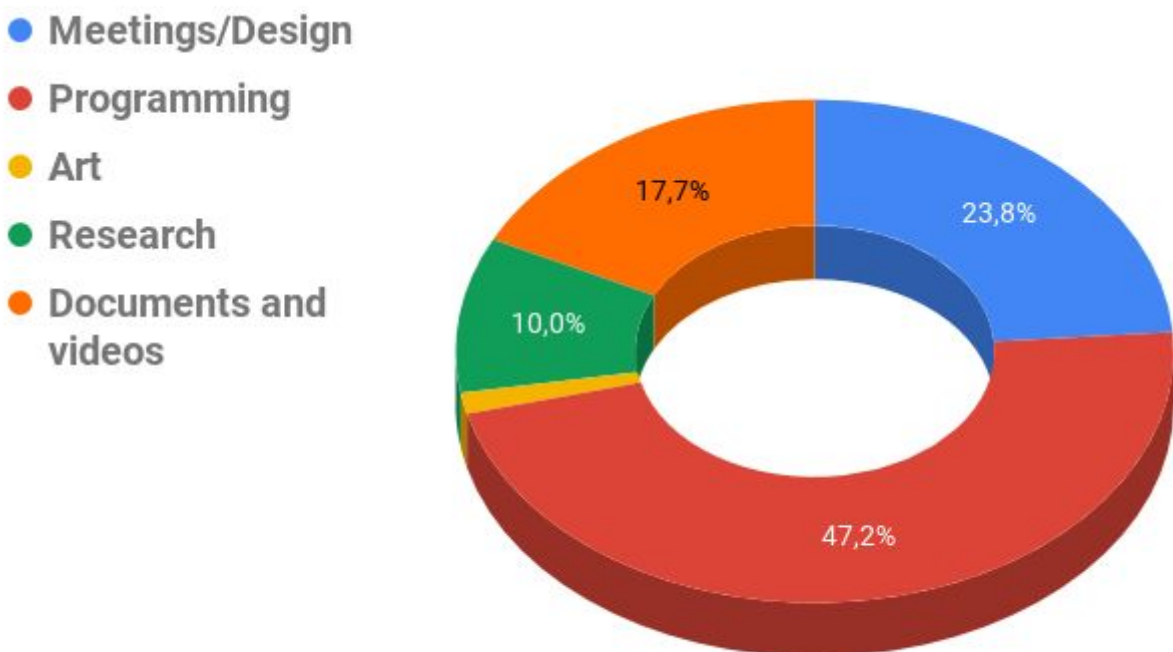


Figure 31: Percentage of time spent on the project according to the different tasks. Almost half the time spent on the project was devoted to programming the game.

6. Conclusions and future work

Working in a group allowed me to develop a bigger project but it was much more difficult due to the necessity of sharing and put in common all the decision we made and it also was complicated to work with second-year students. Moreover, I spent too much time in different discussions with the Hanze's teachers for the concept of the game and our client has not been helpful at all to fix this problem. Due to this, I implement features in the project that I had to discard because of the feedback of the teachers and/or because of the opinion of my team. With all these conditions, even so, the project is done and the basic elements of the game work well.

This project has been a really good opportunity to learn, to experiment with a new way to develop a game and to improve my skills as a programmer. It was thanks to the knowledge acquired throughout the Bachelor's degree in Design and Development of Video Games that the development of the project could be carried out. I think that the project shows everything that I have learned in the last four years.

Nevertheless, the game is not as developed as I would have liked at the beginning: more background of the story is missing (more game design is missing) and it is not complicated enough for people to play it every day, but that is due to the nature of the game: a normal city building game has many elements and buildings to create it in only 300 hours. I also realized that when you create a videogame the prototype phase is quick and easy to implement but there comes a point when you get stuck because you do not what to add more there or how to do it.

Even so, I am proud of my work, I have accomplished almost everything that I expected and I have learned along the way, which is the most important part of this project. Moreover and considering that this type of games is always adding content, the project is ready for expand it.

For the future of the game I want to try to get revenue. We made a business plan for our game and company (called GasCan Productions) and we thought that our game can be a freemium game with ads and microtransactions. So the next thing that I want to implement is the micro transactions that I said in the section 2.11 of this document.

As I said, the game is prepared to add more buildings in an easy way, so it will be easy to expand the game and update it. To sum up, here are some of the most important future updates the project will incorporate:

- Achievements in Google Play
- More buildings
- Leaderboard
- More energy buildings like nuclear plants, hydroelectric plants, etc
- Microtransactions and/or ads in game
- Visit the city of his/her friends for getting rewards

7. References

- ACCIONA. (2019, April 23). Necesitamos una #LeyDeCambioClimáticoYA. Retrieved July 12, 2019, from <https://www.youtube.com/watch?v=SGQjv2KBOlg>
- Agrawal, S. (2019, May 28). Native Android screenshot/image sharing in Unity using FileProvider. Retrieved July 3, 2019, from <https://medium.com/@agrawalsuneet/native-android-screenshot-image-sharing-in-unity-using-fileprovider-4e1617db59c2>
- Agrawal, S.. (2019, May 28). Native Android share to a particular app in Unity - Suneet Agrawal. Retrieved July 3, 2019, from <https://medium.com/@agrawalsuneet/native-android-share-to-a-particular-app-in-unity-d73c955fe6f0>
- Android Developers. (n.d.). FileProvider. Retrieved July 12, 2019, from <https://developer.android.com/reference/android/support/v4/content/FileProvider>
- Cities: Skylines. (n.d.). Retrieved July 3, 2019, from https://store.steampowered.com/app/255710/Cities_Skylines/
- Depth buffer with orthographic camera? (n.d.). Retrieved July 3, 2019, from <https://forum.unity.com/threads/depth-buffer-with-orthographic-camera.355878/>
- Electronic Arts. (2018, July 03). Los Simpsons: Tapped Out. Retrieved July 3, 2019, from <https://www.ea.com/es-es/games/the-simpsons/the-simpsons-tapped-out>

Electronic Arts. (2019, June 19). SimCity BuildIt. Retrieved July 3, 2019, from <https://www.ea.com/es-es/games/simcity/simcity-buildit>

IRENA. (2019, July). Renewable energy statistics 2019. Retrieved July 3, 2019, from <https://www.irena.org/publications/2019/Jul/Renewable-energy-statistics-2019>

IsPointerOverGameObject not working with touch input. (n.d.). Retrieved July 3, 2019, from <https://answers.unity.com/questions/1115464/ispointerovergameobject-not-working-with-touch-inp.html>

Leary, K. (2018, January 17). Renewable energy will be cheaper than fossil fuels by 2020, according to a new report. Retrieved July 3, 2019, from <https://www.businessinsider.com/renewable-energy-will-be-cheaper-than-fossil-fuels-by-2020-2018-1?international=true&r=US&IR=T>

Microsoft .NET. (n.d.). DateTime Struct (System). Retrieved July 12, 2019, from <https://docs.microsoft.com/en-us/dotnet/api/system.datetime?view=netframework-4.8#properties>

Micu, A. (2019, April 19). The price of solar keeps falling, Dubai received the lowest ever asking bid for energy. Retrieved July 3, 2019, from <https://www.zmescience.com/science/dubai-solar-cheap-55116/>

Milman, O. (2019, March 25). 'Coal is on the way out': Study finds fossil fuel now pricier than solar or wind. Retrieved July 3, 2019, from

<https://www.theguardian.com/environment/2019/mar/25/coal-more-expensive-wind-solar-us-energy-study>

Montoya, Francisco & Aguilera, María & Manzano-Agugliaro, Francisco. (2014). Renewable energy production in Spain: A review. *Renewable and Sustainable Energy Reviews*. 33. 509–531. 10.1016/j.rser.2014.01.091.

Nicolas Ortiz. (2015, December 01). Dealing with Permissions when sharing files (Android M). Retrieved July 12, 2019, from <https://medium.com/@benexus/dealing-with-permissions-when-sharing-files-android-m-cee9ecc287bf>

ProBuilder. (n.d.). Retrieved July 3, 2019, from <https://unity3d.com/unity/features/worldbuilding/probuilder>

Roystan. (n.d.). Unity Toon Water Shader Tutorial at Roystan. Retrieved July 3, 2019, from <https://roystan.net/articles/toon-water.html>

Unity Technologies. (n.d.). Application.persistentDataPath. Retrieved July 12, 2019, from <https://docs.unity3d.com/ScriptReference/Application-persistentDataPath.html>