GAME DESIGN & DEVELOPMENT (GDD)

Final Bachelor Project Report

Raúl Trullenque Viudas
gerostv@gmail.com
TABLE OF CONTENTS

I. Introduction ................................................................................................................................................. 3

II. Inspiring Games ........................................................................................................................................... 4
   1. Age of Empires II: The Age of Kings ................................................................................................. 4
      1.1. Gameplay ........................................................................................................................................ 4
   2. Stronghold ............................................................................................................................................... 6
      2.1. Gameplay ........................................................................................................................................ 6

III. Vision statement ....................................................................................................................................... 7
   1. Game Logline ......................................................................................................................................... 7
   2. Gameplay Synopsis .............................................................................................................................. 7

IV. Game Structure ......................................................................................................................................... 9
   1. Interfaces .............................................................................................................................................. 9
      1.1. Mouse ................................................................................................................................ .......... 9
      1.2. Keyboard ....................................................................................................................................... 11
      1.3. Graphical User Interface ............................................................................................................. 13
   2. Scoring/Wining conditions .................................................................................................................. 16
      2.1. Scoring ........................................................................................................................................... 16
      2.2. Wining .......................................................................................................................................... 16
   3. Level .................................................................................................................................................... 18
   4. Game Characters ................................................................................................................................... 19
      4.1. Survivors ......................................................................................................................................... 19
      4.2. Zombies ....................................................................................................................................... 20
   5. Abstract Rules ...................................................................................................................................... 21

V. Technical analysis ..................................................................................................................................... 22
   1. Development platform ........................................................................................................................ 22
   2. Object Diagram .................................................................................................................................. 23

VI. Process of Development ........................................................................................................................ 25
   1. 1st iteration .......................................................................................................................................... 25
      1.1. Initial Planning .............................................................................................................................. 25
      1.2. State of the Game ......................................................................................................................... 26
   2. 2nd iteration ........................................................................................................................................ 26
2.1. State of the Game .................................................................................. 26
2.2. Testing conclusions ............................................................................. 27
3. 3rd iteration ............................................................................................... 27
    3.1. State of the Game ................................................................................ 27
    3.2. Testing Conclusions .......................................................................... 28
VII. Conclusions .......................................................................................... 29
    1. Functionality and Completeness ............................................................ 29
    2. Balance .................................................................................................. 29
    3. Accessibility ......................................................................................... 30
    4. Fun ....................................................................................................... 30
    5. Game Maker ......................................................................................... 30
    6. GDD ..................................................................................................... 31
I. Introduction

This document is the report for the final bachelor project named Game Design & Development made by Raúl Trullenque Viudas during the spring semester of 2013.

The project consists on the design and development of a videogame called GDD using Game Maker, this document will explain the process, vision and other aspects of the creation of the project

GDD and all the other project data can be downloaded following this URL: http://goo.gl/JIECp
II. Inspiring Games

As any honest creator would recognize, nobody creates nothing from scratch, every one of us are always influenced by our past experiences and education and this fact is always more obvious when we try to create something new.

So to be faithful to this way of thinking and honest to the reader I will describe the games that had inspired me the most during the development of the project.

1. Age of Empires II: The Age of Kings

Age of Empires II: The Age of Kings is a real-time strategy (RTS) video game developed by Ensemble Studios and published by Microsoft. Released in 1999 for the Microsoft Windows and Macintosh operating systems, it was the second game in the Age of Empires series.

1.1. Gameplay

The game focuses on building towns, gathering resources, creating armies, and destroying enemy units and buildings. Players conquer rival towns and empires as they advance one of 13 civilizations through four "Ages": the Dark Age, the Feudal Age, the Castle Age and the Imperial Age. Advancing to a new Age unlocks new units, structures, and technologies, but players must pay a sum of resources to advance to the next age (typically food and gold) once certain buildings from their current age are constructed.

Civilian units, called "villagers", are used to gather resources. Resources can be used to train units, construct buildings, and research technologies, among other things; for example, players can research better armor for infantry units. The game offers four types of resources: food, wood, gold, and stone. Food is obtained by hunting animals, gathering berries, harvesting livestock, farming, and shore fishing and fishing from boats. Wood is gathered by chopping down trees. Gold is obtained from either gold
mines, trade or collecting relics in a monastery, and stone is collected from stone mines. Villagers require checkpoints, typically depository buildings (town center, mining camp, mill, and lumber yard), where they can store gathered resources.

Besides the five single player campaigns that containing historically-based scenarios such as Genghis Khan's invasion of Eurasia, Barbarossa's Crusade, or Saladin's defense of the Holy Land there are different game modes available to the player in The Age of Kings. One mode, random map, generates a map from one of several randomly chose map generating scripts, with players starting in the Dark Age with a Town Center, three villagers and a scout unit. The game can be won through military conquest, by constructing a special building known as a Wonder and keeping it standing for a certain amount of time, or by obtaining control of all relics on the map for a set amount of time. Deathmatch mode allows players to begin with large amounts of resources, creating a focus on military dominance, while in the regicide mode each player is given a king unit, winning by killing all of the other monarchs.
2. Stronghold

Stronghold is a historic real-time strategy (RTS) game developed by Firefly Studios in 2001. Because of its popularity, the game spawned four sequels: Stronghold: Crusader, Stronghold 2, Stronghold Legends and Stronghold 3 as well as several compilation packages.

2.1. Gameplay

In Stronghold, the player takes the role of a lord in a kingdom. The goal is to create a stable economy and a strong military to defend against invaders, destroy enemy castles and accomplish the mission objectives.

On a very similar way to Age of Empires II, this game also has two different kinds of units, soldiers and civilians, which work in almost the same way, but stronghold has a much more developed economic gameplay, with many different primary resources that have to be stored and processed. For example one of the many ways of obtaining food is creating a wheat farm, the wheat that the farm produces has to be processed on a windmill, and then the flour created, used in a bakery to finally get the bread, but that’s not the end because this bread has to be stored and kept safe from bandits and enemies that would try to stole it.

The main game mode is the military campaign, which is based upon a map of England. The player then has to regain control of the Kingdom, re-gaining control of one county each mission. An economic campaign is set after the main campaign, where the player re-constructs parts of the Kingdom. The player is set goals to complete against a variety of obstacles, such as bandits and fire.
III. Vision statement

1. Game Logline

GDD is a real-time strategy game set during the zombie apocalypse focused on exploration, gathering of resources and extermination of the undead menace.

2. Gameplay Synopsis

The game takes place on the present, but on an alternate universe where a couple of months ago the apocalypse zombie took place.

Like on many histories of the genre it all started with a series of mysterious attacks, many people saying that have been attack by incoherent and crazy people that tried to bite them. Following these, each day more common, events a new disease appeared between the survivors of these attacks but when this two facts where linked it was already too late since the people where already scattered over all the world. These new disease seemed too put the patients through a great agony for days to finally make them very violent, so much that they would start attacking everybody. Beyond this point everything started to happen very fast, the governments declared the state of emergency and gathered their people on their cities, but this only make it easier to the disease to spree among the population, so they started so get sick by thousands. By the time the infected where identified as undead zombies the society was already one step to disappearing, one final step that was took when the remaining military started to slaughter both the zombies and those suspicious of being infected, until, at the end, they were also wiped out.

On the present day a group of the few people that have manage to survive through all the horror and chaos of the last months has decided that they are tired of living on fear and that is time that they retake what is theirs, so they have gathered on an unidentified city with the purpose of exterminating all the zombies that roam on it and start the rebuilding of civilization.

On the start of GDD the player takes control of this group of survivors and the first that realizes is that the city is crowded with zombies, much more than the ones he can defeat with the small group of humans that has at the moment, he also notices that all the zombies that see his survivors are moving towards them. So the first reaction of the player is to select his men an order them to move to a safer place, probably on another, less crowded, street. At this moment the user realizes that the zombies are still moving, even that now they no longer see the survivors, but they are just going to the last point
they saw the humans, and then wandering off. Besides this, there are too many zombies and there is no safe place on the streets, so the user decides to get their survivors into a building, discovering that the zombies chasing them cannot follow them inside. Upon entering the building the user get the opportunity to explore each one of the different floors to scavenge for food, wood or new survivors, but knowing that it’s also a threat to the survivors on the building, because on that floor might be trapped zombies that could get loose during the exploration and attack them.

After exploring some floors the player realizes that to get more survivors he needs to give them somewhere to live, so with the material gathered, he transforms one of the explored floor on a refugee which allows him to get a larger army. So with all this knowledge he continues exploring more buildings to increase and feed his army, and starts killing the zombies. With this intense combat, his survivors eventually get injured so the player uses another of the explored floors to build a hospital where one of the survivors will take care of all the injured people on the building and slowly get them back to health.

Following these path of gathering new resources, getting new recruits and healing them when they get injured, the player works his way around the town defeating the zombies until there are none left, and so winning the game.

The main difference that makes GDD unique among the other games is its completely new combination of genre and setting. The mechanics follow a traditional path inspired by other real-time strategy games, where the player can select and move his troops to any point of the map, attack other enemies with them or get them inside buildings, and these same building can produce resources and be upgraded; but is the combination with these new setting that creating new mechanics and gameplay. For example, traditionally RTS games are always located on great portions of land with noting on it, where the players can build their base and battle the enemy, and these is this way not only on fantasy and medieval settings, where it could make sense to not have cities, but also in futuristic and contemporary settings; instead, in GDD the player is encouraged to use the building that already exist on the city, explore them, clean them of any threat and reuse them with new purposes, making fresh turn to the genre and achieving a new level of realism.
IV. Game Structure

On this section will be explained all the different parts that form GDD

1. Interfaces

1.1. Mouse

The mouse allows a great variety of different interactions:

- Movement of the view: By approaching the mouse cursor to anyone of the four edges of the screen of the game the player can move the view around the map, of course this movement changes depending on the side of the screen approached, for example is the side approached is the south side, the view will move to the south, and continue until the view gets to the end of the map or the mouse return to a more centered position on the screen.

- Selection: The player can select any of his survivors or any of the buildings of the map by pressing on them with the Left Button on the mouse. When a unit is selected a green ellipse appears at the bottom of his sprite, helping the user to know if he has selected correctly the unit he wanted to select. For the same reason when the players selects a building a green rectangle appears at the base of his sprite.
• Multiple selection: The player can select more than one unit by pressing and maintaining the Left Button of the mouse on any empty point of the map and then moving the mouse, on doing so a green rectangle appears, having one corner on the point where the mouse was pressed and the opposite on the position of the mouse. If the player releases the Left Button, all the survivors that in that moment inside the rectangle will be selected. Since more than one building cannot be selected at once, this rectangle will not select buildings.

• Unselecting: The player can deselect a unit, a group of units or a building by pressing the Left Button on an empty space of the map.

• Movement: If the player has at least one survivor selected, he can press the Right Button on any free point of the map to command the selected units to move to that point. The units will move to that point avoiding obstacles like buildings, but they can still receive damage from nearby zombies. At the point selected a green rectangle would appear to provide feedback to the player on the goal of his units.
• Attacking: If the player has at least one survivor selected, he can press the Right Button on a zombie to command to the selected survivors to attack that zombie. The units will chase the zombie until they reach it and then they’ll attack it until its death. To improve the accessibility of the game a red ellipse would appear at the base of the sprite of the zombie to indicate that the zombie has been successfully targeted.

• Enter into buildings: If the player has at least one survivor selected, he can press the Right Button on any building to command to the selected survivors to enter into that building, the survivors will move to the entrance of the building and the enter into it.

• Using Buttons: The player can press with the Left Button on the buttons of the GUI to use them, the actions that these buttons perform are explained on the GUI section.

1.2. Keyboard

• Movement of the view: The player can press the movement keys to move the view in that direction until it reaches the end of the map or the player stops pressing the key, on the same way he can also use the W,A,S,D keys.

• Show/Hide the GUI: Sometimes the GUI may hide units behind it or maybe the player prefers having a bigger field of vision of the map, to solve this the user can press the Z key in the keyboard to hide the GUI, and he can press it again to show it back again.
- **Show/Hide the buildings**: Due to the isometric view of the game, sometimes zombies or unit can hide behind tall buildings and become unreachable, for that reason the X key can be used to change the sprite of the buildings to a new one that one shows their base, and pressing another time the player can change it back.

- **Spawn a zombie**: Pressing the B key the user can spawn a new zombie on the position of the mouse. This functionality is only for debugging purposes and it’s not intended to be used by the player during the game.

- **Spawn a survivor**: Pressing the V key the user can spawn a new survivor on the position of the mouse. This functionality is only for debugging purposes and it’s not intended to be used by the player during the game.
1.3. Graphical User Interface

- Resources Panel: This panel shows to the user information about the exact amounts of resources he has, his current number of survivors and limit of population.

- Context Panel: This panel shows information and provides different action buttons related to the entity selected.
  - Selecting a survivor: On the current version of the game there are no information or actions related to units, but the reader can get information about them on the Future Extensions chapter of this document.
  - Selecting a building: When selecting a building the panel shows three different sections:
    - Left Section: This part of the panel has a diagram of the building that shows clearly the different floors. Each one of this floors is represented by a different image that shows what kind of floor is it. This images can also be pressed to change the current selected floor that, to increase the accessibility, will be show in green.
      - Unexplored Floor: This is the state of every floor on the game before it is explored by the player.
• Explored Floor: This is the image shown to represent that a floor has been explored:

![Explored Floor Image]

• Refugee: This is the first basic floor transformation the player can make, it represents that the floor selected has been transformed into a refuge where more survivors can live, thus raising the maximum of population by 5.

![Refugee Image]

• Hospital: This is a more advanced transformation that represents that the floor has been turned into a hospital. This hospital will take a survivor as doctor and will heal 5 points of damage on every survivor inside the building each minute.

![Hospital Image]

- Central Section: This section holds information about the current selected floor. It mainly has two purposes, shows the number of the selected floor and has the buttons that allow the transformations of the floor. This button are shown on grey when to all the requisites to use them are met and on this estate cannot be pressed.

• Exploration Button: This button explores the selected floor, to use it there has to be at least one survivor inside the building. During the exploration the player can find a number between 0 and 20 of wood and food and between 0 and 2 of survivors, but also has a chance of finding zombies inside the building and having to fight them, the result of the fight is calculated in base of the number of survivors on the building in a way that the more survivors the less the chance of having casualties during the struggle.

![Exploration Image]
• Refugee Button: This button transforms the selected floor into a refugee, but to be used the player has to have at least 5 of wood that will be spend on the construction.

• Hospital: This button transforms the selected floor into a refugee, but to be used the player has to have at least 50 of wood that will be spend on the construction and a survivor that would take the role of doctor.

• Right Section: This section shows the number of survivors inside the building and contains buttons that affect the whole building.
  • Exit One: This button takes one survivor outside the building.
  • Exit All: This button empties the building of people.
2. Scoring/Wining conditions

2.1. Scoring

On GDD the score is a number that allows the player to know on a quantitative way how better was this game from his last, this way a very experienced player that wins the game almost every time he plays, can still improve his strategies by comparing his scores.

The score is calculated starting from zero on every new game, and then adding to it whenever one of the next events happen:

- If the player gets 1 of Canned Food add 1 point to the score
- If the player gets 1 of Wood add 1 point to the score
- If the player rescues 1 Survivor add 10 points to the score
- If the player kills 1 Zombie add 5 points to the score

2.2. Wining

The objective of GDD is to regain the city from the dead hands of the zombies, to achieve that the player has to eliminate the zombies until they are no longer a threat to the city, which translated to numbers means that the player wins the game when there are less than ten zombies in the city. When it happens the game end and the player is presented with the victory screen.
On this screen the player can see the score he has reached on the last game and can press the Start button to start a new one.

But if instead of exterminating the zombies, are the zombie the ones that manage to kill all the survivors, the game is lost and the player goes to the Game Over screen, where he can also see his score and start again.
3. Level

The only level on the game takes place on the unnamed city that the player has to regain from the zombie infestation. The city is created by the position of the buildings in lines that resemble streets, alleys and avenues.
4. Game Characters

On this section will be explained the behavior of the different game characters.

4.1. Survivors

The survivors are the characters controlled by the user, they are explored and warriors with a basic AI. They main feature is the way the move from one point to another, they create a path using the A* path taking algorithm, which means that when they are ordered to move to one point on the map, they first calculate the more efficient path and then they follow it. Thanks to this algorithm they are even capable of solving mazes without error, but the downside is that the calculation of this path needs quite a lot of processing resources, which means that if too many survivors are moved at the same time the Frames per second of the game could drop for a moment.

The survivors are described in code by the objPlayer object and the main parts of his behavior are the next ones:

- On the moment of their creation they define their main variables
- Whenever the user presses the right button of the mouse the selected survivor reacts by cancelling the current attack and doing another action depending on the object that was at the point pressed.
  - If the point had a zombie they start chasing him
  - If the point had a building they go inside
  - If the point was empty they move to the point
- Every cycle of the game or step, as Game Maker names it, the survivors check different things.
  - If they are chasing a zombie but not yet attacking him they check if his zombie target is in range to be attacked, if so, they start the attack; if the zombie target has moved they update the path they are following to the zombie.
  - If they are not chasing a zombie, nor attacking one, but there is a zombie inside his range of attack, they start attacking him
  - If they have less than 1 points of life they decrease the population count in 1 and destroy themselves
- The survivors’ attack is an alarm event that triggers themself 10 steps after has been called. On this method the survivor checks if the zombie he is trying to hit still exists and is still on his range of attack, if all of this is true he takes 10 points of life from the zombie and calls the attack method again, if something is not true he cancels the attack.
• On the drawing event the survivor calculates the orientation he is facing and changes his current sprite to represent it correctly, he also check if he is selected, and if he is, draws a ellipse to give feedback to the player.

4.2. Zombies

The zombies are the enemies, they are the plague that has conquered the city and the ones that the survivor have to exterminate to archive victory. The zombies can’t be controlled by the user, they behave only directed by their own AI.

To move from one point to another they don’t use the same algorithm than the survivors, instead they use a method that only allows them to move in straight lines, but since they are dumb zombies the player doesn’t expect smart moves from them and this feature also makes easier to the player to lose a group of zombies following his survivors by turning a few tight corners and unlike the A* algorithm thanks to this method the game can thousands of zombies moving around without losing frames per second.

The zombies are described in code by the obj_zombie object and the main parts of his behavior are the next ones:

• The zombies are always slowly moving from one place to the next, so the first that they do after defining his variables is find a random and accessible point inside his field of vision as goal.

• On the step event the zombie checks several things:
  o If his health is lower than 1 adds 5 points to the score, he marks himself as dying and calls the alarm 1 event that when triggered will destroy him.
  o If he is not already attacking a human searches for the nearest one inside his field of vision.
    ▪ If there is a human at sight and inside his range of attack, he starts attacking
    ▪ If there is a human at sight but too far away he starts moving on that direction
    ▪ If there are not humans at sight he continues to his goal and if he has already reached the goal point, he find a new random point and continues moving

• The zombies’ attack is very similar to the one of the survivors, but is a bit slower, which means that a good strategy of attack to a survivor is to approach a zombie and attack him and then quickly move outside the zombies attack range so the zombie fails his attack.

On this method the zombie checks if the survivor he is trying to hit still exists and is still on his range of attack, if all of this is true he takes 10 points of life from the
survivor and calls the attack method again, if something is not true he cancels the attack.

- On the drawing event the zombie calculates the orientation he is facing and changes his current sprite to represent it correctly, he also check if he is being targeted, and if he is, draws a red ellipse to give feedback to the player. Besides this the zombie also changes his sprite to represent his current state, moving, attacking or dying

5. Abstract Rules

The last rules that define the gameplay of GDD are small set that control the usage of resources and difficulty of the game.

- The first rule states that every minute each survivor consumes one unit of canned food. This rule represents the use of food on the real world and encourages the player to keep searching for resources, and if he doesn’t have enough food, each minute every survivor will take 10 points of damage.
- Also minute, 4 new zombies will appear on different positions of the map, this rules also encourages the player to search for resources and new survivors since he is going to need them to face the ever growing horde of zombies.
V. Technical analysis

1. Development platform

This project has been made using a tool called Game Maker. Game Maker is a game development software application written by Mark Overmars in the Delphi programming language. It is designed to allow its users to easily develop computer games without having to learn a complex programming language such as C++ or Pascal. GameMaker’s development interface uses a drag-and-drop system, allowing users unfamiliar with traditional programming to intuitively create games by visually organizing icons on the screen. These icons represent actions that would occur in a game, such as movement, basic drawing, and simple control structures. For experienced users, Game Maker contains a built-in scripting programming language called "GML", or "Game Maker Language". Games can be distributed under any license (including no license), as non-editable executable ".exe" files or as the source code file (.gmk, .gm6, .gmd, .gmf) itself.

The most important parts on the structure of a game maker game are the Rooms and the Objects.

- The Rooms represent the different levels or screens that the game can display, for example, in GDD the rooms are the main menu, the game level and the winning and game over screens.
- The Objects are the equivalent to the Classes of a normal software application, they contain the most part of the code, but they are a little different. GameMaker uses what is called an event driven approach. This works as follows. Whenever something happens in the game the instances of the objects get events (kind of messages telling that something has happened). The instances can then react to these messages by executing certain actions. For each object you must indicate to which events it responds and what actions it must perform when the event occurs. The events used on GDD are the next ones:
  - Create event: This event happens when an instance of the object is created.
  - Alarm events: Each instance has 12 alarm clocks. You have to set these alarm clocks and then the alarm clock ticks down until it reaches 0 at which moment the alarm event is generated.
  - Step events: The step event happens every cycle or “step” of the game
- Collision events: Whenever two instances collide (that is, their sprites overlap) a collision event appears.
- Keyboard events: When the player presses a key, a keyboard event happens for all instances of all objects. There is a different event for each key.
- Mouse events: A mouse event happens for an instance whenever the mouse cursor lies inside the sprite representing the instance. Depending on which mouse buttons are pressed you get the no button, left button, right button, or middle button event.
- Draw event: Instances, when visible, draw their sprite in each step on the screen. When you specify actions in the drawing event, the sprite is not drawn, but these actions are executed instead. This can be used to draw something other than the sprite, or first make some changes to sprite parameters.

2. Object Diagram

This diagram illustrates the relationships between the different objects of the game in a similar way than a UML class diagram would in a normal software project.

Inside the object are listed the events of that object, that could be similar to the methods of a class. The Scripts are represented by a dotted lined because they are not forming an object and are instead a group of methods that can be used from any object.
VI. Process of Development

1.1. Initial Planning

The image on the left is a Gantt diagram representing the initial planning for the project made on the first iteration of development. The objective for each week are listed below.

- Multiple Pathfinding
  - Initial Planning
  - Zombie behavior
    - Random wandering by default
    - Alert when humans on sight
  - Combat
    - Human and zombie HP
    - Melee damage
    - Human characteristics
  - Basic resources
  - Human abilities
- Multiple units selection and movement
  - Select multiple humans and move them at once
  - Avoid odd behaviors
- Construction
  - Barricades
  - Fences
- Animations/Sounds
  - Solve Isometric hiding

Below are the objectives for each iteration:

1. **1st iteration**
   - Initial Planning
     - Multiple Pathfinding:
       - Zombie behavior
         - Random wandering by default
         - Alert when humans on sight
     - Combat:
       - Human and zombie HP
       - Melee damage
       - Human characteristics
     - Basic resources
     - Human abilities
   - Multiple units selection and movement:
     - Select multiple humans and move them at once
     - Avoid odd behaviors
   - Construction:
     - Barricades
     - Fences
   - Animations/Sounds:
     - Solve Isometric hiding

The objectives for each week are listed below:

<table>
<thead>
<tr>
<th>Week</th>
<th>Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Initial Planning</td>
</tr>
<tr>
<td>2</td>
<td>Multiple Pathfinding</td>
</tr>
<tr>
<td>3</td>
<td>Combat</td>
</tr>
<tr>
<td>4</td>
<td>Basic resources</td>
</tr>
<tr>
<td>5</td>
<td>Human abilities</td>
</tr>
<tr>
<td>6</td>
<td>Multiple units selection and movement</td>
</tr>
<tr>
<td>7</td>
<td>Construction</td>
</tr>
<tr>
<td>8</td>
<td>Animations/Sounds</td>
</tr>
</tbody>
</table>

The timeline for each iteration is represented in the Gantt chart.
1.2. State of the Game
At the end of the first iteration the game is still on a very early state, and the plan has been followed quite well but since almost all the work was put on the pathfinding the game is not very funny. It is technically a game since there is an objective; there is a zombie that tries to kill you but that’s all, there is no way to win, just lose.

2. 2nd iteration
2.1. State of the Game
During this iteration development is where becomes mandatory to make changes to the plan since new non-planned features become necessary and problems appear that take priority from another features. Almost half of the work of this iteration goes to the development of buildings as planned, but the other half goes to unexpected sources. For example at this point of the game both, zombies and survivors, used the same algorithm to calculate the pathfinding, and during this iteration it proved itself unsustainable to manage more than four zombies at the same time, so a new system was needed to move the zombies. That and other small bug fixes where the reason because the creation of the GUI was moved to the next iteration and the construction was completely removed.
2.2. Testing conclusions
Due to the almost inexistent content the main reaction of the testers where that the game wasn’t funny to play, this reaction was completely understandable since there were no way of winning or resources to gather, but the testing was instead, very useful on the search for bugs and raising suggestions.

3. 3rd iteration
3.1. State of the Game
During the final iteration the main featured created was the GUI, which turned to be a lot more challenging than expected, since Game Maker resulted of no help for an RTS-
type user interface all the buttons and panel, and his behaviors were made from scratch. This and the bug fixes from the current and last iteration formed the last iteration.

3.2. Testing Conclusions

The last round of testing also showed lots of bugs and gave suggestions, but this time the actual gameplay was tested.

During testing the first important issue that raises is that the lack of a proper tutorial that explains the less obvious mechanics of the game was a problem. All testers easily deduced the controls that move the camera and the most part of the testers whom had already played to other strategy games had no problem with selecting, moving units, attacking and entering al leaving buildings; but the mechanic of exploring and transforming floors was a bit too abstract for some of the testers.

Other surprising result extracted during the testing was the discovery of a strategy that could make the player win always with little effort. The strategy I used to use to play the game and the one that was used during development consisted on trying to create safe zones on the city by blocking streets with survivors, explore the buildings of the street, and killing the zombies that came to near; and then use more survivors to take another street until gradually all the zombies were dead. Instead of this a lot of the testers followed a much simpler strategy that consisted on moving all the survivors always together everywhere, they would explore some building in a row with all the survivors available until they get a big number of them, and they would just go to kill all the zombies one by one, since the zombies are always slower attacking, if you attack a zombie with five or more survivors at once the zombie would die without having a chance to attack.
VII. Conclusions

1. Functionality and Completeness

As proved by the testing sessions, the game is functional, it can be picked up by a non-experienced user and played without getting into an impasse but that by no means it’s complete. As any player that has played any strategy games before could tell, GDD lacks many of the features that are thought to be essential on this type of games, as could be the construction of buildings, civilian automatized workers, a more varied, and even more important, interesting group of resources to be collected, multiplayer capabilities, etc. of course, all this features could make a change for the better, but as is often said, no game is ever complete, there is always room for improvement.

2. Balance

As the multiple changes through the different iterations had confirmed to me, balance is a complex and difficult thing to accomplish on a game. During this project the balancing efforts where focused on the spawning rate of the zombies. The objective was to get a rate that would not eliminate the player from the game while he is learning how to play, but that after some time transform into a challenge that needed a little ability to achieve, this kind of balance is implemented, but as the latest testing sessions proved, to have a balanced featured does not make a balanced game.

On the latest sessions of testing, the testers had already played to the game sometimes and were already familiar with the mechanics, so they used the time at the beginning of the game that was supposed to be used to learn the mechanics, to quickly gather resources and troops while the number of zombies is still low, and then exterminate them with ease before they numbers get to grow.

With more time, features could be added to the game to balance this strategy, for example the resources needed to maintain a great army could be greater, the attack speed of the zombies could be raised or even develop and algorithm that keeps and number of zombies on the map depending on the size of the survivors army, this way the challenge could be maintained during the game. But at the end, the fact that the experienced player could find a strategy that guaranties their victory is a proof of the usefulness of the testing sessions and a proof of how much GDD can be improved.
3. Accessibility

The testing sessions showed that the game is accessible but only to users that had played before to other similar real-time strategy games, this type of testers would find the movement and attacking mechanics very similar to the ones they have already experienced, they would need a little time and try and error to understand the floor transformation mechanics but they would eventually discoverer them by themselves and keep playing. On the other side, testers with no previous experience on this kind of game or games in general would find themselves completely lost and eventually stop playing if they don’t receive external help. We could still say that the game is accessible if we consider our target public experienced players of RTS games, but is no game should target to a so reduces public, so to improve the accessibility a proper tutorial should be added to explain both the common and original mechanics.

4. Fun

Partially derived from the accessibility of the game comes the fun of it. On the testing session was observed that the users were having fun during the process of discovery of the mechanics and exploration of the map and buildings, but to the non-experienced players that got stuck the fun was quickly killed. This problem could be solved by adding a tutorial as explained on the previous section.

Other point when the fun was diminished is when the enemies stop being a challenge. As was explained on the Balance section, some player developed an always-winning strategy to exterminate the zombies, even with his strategy, to this players the game is fun while they are building his army and gathering resources, but when they are done and start killing the zombies one by one without opposition, the fun is lost, the game changes from a challenge to a tedious work with no surprises. Of course this could also be repaired and improved by adding content and rules that increase the challenge.

5. Game Maker

Game maker a great tool it really has helped a lot to the development of the project, but it also has brought some problems. Game maker is a software that works very well when the user is creating a small 2D game, for example the prototype of a platform game, but when the user tries to do something bigger or more complex it is not so useful. For example, the game maker built-in scripting language allows very quickly coding since
you don’t have to define variables and can access any object from any other, but when the project starts growing this lack of rigid structure makes coding more chaotic.

In conclusion, I think that Game Maker is a very good prototyping tool, but a no so good game development software. If I would had to start the project over again and keep working with Game Maker I would change the genre of the game, but I keep working with GDD I would change Game Maker for a more advanced tool.

6. GDD

The development of this project has been a great experience, it has taught me lots of things about game design and the process of development. At the beginning of the project I was too ambitious and I planned a game that was impossible to create all by myself. I underestimated both the time it would take me to learn how to use properly Game Maker and the time unexpected bugs and needed changes would need. For this reasons GDD is more similar to a strategy engine than to an actual game, it may be more useful if used by developers to create his own strategy games on Game Maker since the its more important feature is its structure and capabilities.