

# Group 62

Group Software Project

## Design

## Turris

Oliver Legg | Thomas Coupe | Daniel Taylor | Matthew Smith | Alessandro Wang | Kieran Baker

## Summary

Our group intends to make a tower defence game that can entertain users for hours. The overall goal of the game is to:

- Entertain
- Challenge
- Intrigue
- Escape
- Compete

To complete these goals, we would like to give an experience that the user wants from a game. Not everyone would want to compete – therefore we'd like to give them choices and options within the game. To tackle these problems, we have some main objectives to achieve this:

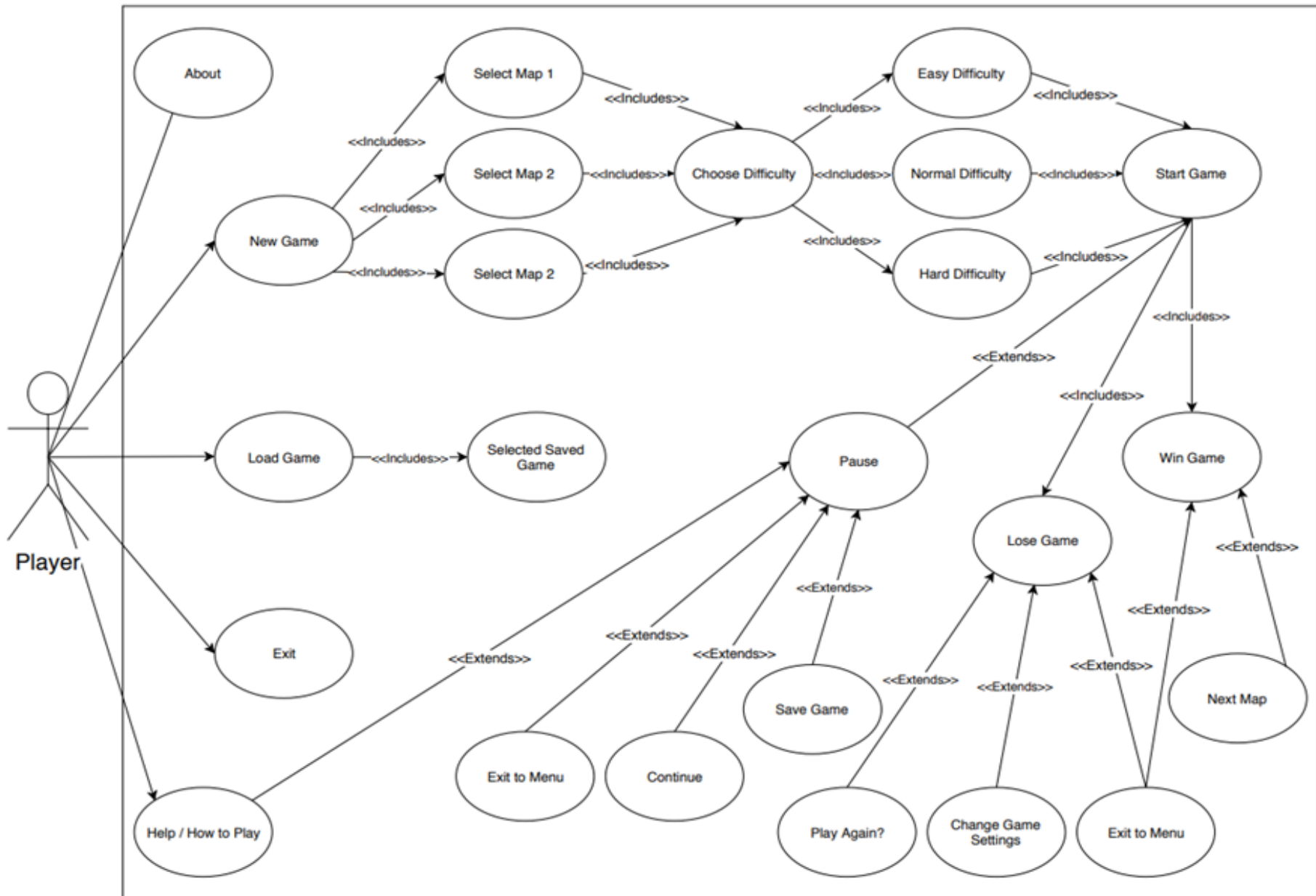
- Create a satisfying experience for the user to interact with the game and impress them with gameplay quality, graphics and sound effects.
- Create an environment in the game that will challenge the user's ability. The user can challenge themselves by settings the difficulty from easy, medium and hard.
- Intrigue the user on what happens when you win the game or play a level with their own strategy.
- Create an environment of the game and overall experience that will captivate the user and help them to escape.
- Create endless mode which allows the user to have a certain time that they survive for. This can be compared with another player's score – this will bring a competitive aspect of the game.

We want the user to feel good when playing the game and want them to feel rewarded for playing well. To do this, we have we do have to difficulty aspect and loss aspect, if you can only win at the game, it will feel undeserved and meaningless.

## Business rules

- To reduce storage space, each user will be limited to one save file which will store data for a single game. If the user wishes to save a different game, the old file must be deleted.
- The save file must only contain the relevant data that is required for the save such as the remaining lives, wave number and turret locations, any extra information should be omitted to reduce file size on the disk.
- In the event of an error occurring while the user is attempting to save or load their game any data for the current save file should not be modified or deleted.
- The name that the player enters must be at least three characters long and no longer than 15 characters. The name entered must only contain letters and numbers and must not have any special characters in it.
- Only 5000 enemies can be spawned at a time as this would be too computationally intensive to process for our users. This would also consume a lot of memory in the computer, eventually slowing it down.
- As there will be a grid-based placement system, there will be a certain number of towers to be placed on the map. If the screen is 800x600 and each tile is 100x100, there will be 48 tiles ( $8 \times 6 = 48$ ) and a lot of these will be taken up by paths. Therefore, there is a constraint on how many towers you can place on a map at a time.

## Use Cases



## Use Case Descriptions

<b>ID</b>	<b><u>UC 1</u></b>
<b>Name</b>	New Game
<b>Description</b>	Allow the player to start a new game
<b>Actor</b>	Player
<b>Pre-condition</b>	Game is launched and player ready to play
<b>Post-condition</b>	Map options displayed
<b>Main flow</b>	1. System displays available maps
<b>Includes</b>	UC 5 (Select Map)
<b>Extensions</b>	UC 17 (Play Again?)

<b>ID</b>	<b><u>UC 2</u></b>
<b>Name</b>	Load Game
<b>Description</b>	Allow the player to start a saved game
<b>Actor</b>	Player
<b>Pre-condition</b>	The player Saved previously a started game
<b>Post-condition</b>	Map options displayed
<b>Main flow</b>	1. Player select desired saved game 2. Player loads in the desired saved game
<b>Includes</b>	Selected Saved Game
<b>Extensions</b>	None

<b>ID</b>	<b><u>UC 3</u></b>
<b>Name</b>	Help/How to play
<b>Description</b>	The system will display a rules and mechanics of the game It can be also selected when you pause during a game
<b>Actor</b>	Player
<b>Pre-condition</b>	Game is launched
<b>Post-condition</b>	Help and guides displayed
<b>Main flow</b>	1.Player get initial idea of how the game work and how to play it
<b>Includes</b>	None

## Group 62

<b>Extensions</b>	None
-------------------	------

<b>ID</b>	<b><u>UC 4</u></b>
<b>Name</b>	Exit
<b>Description</b>	Allow the player to shut down the game and back to the desktop
<b>Actor</b>	Player
<b>Pre-condition</b>	Game is launched and player not willing to play
<b>Post-condition</b>	Game is exited
<b>Main flow</b>	<ol style="list-style-type: none"> <li>1. Player stops desiring to play</li> <li>2. Game exited</li> </ol>
<b>Includes</b>	None
<b>Extensions</b>	None

<b>ID</b>	<b><u>UC 5</u></b>
<b>Name</b>	Select Map
<b>Description</b>	Allow the player to choose a desired map
<b>Actor</b>	Player
<b>Pre-condition</b>	Game is launched and player ready to play
<b>Post-condition</b>	Map options displayed
<b>Main flow</b>	<ol style="list-style-type: none"> <li>1. Player given option to select map</li> </ol>
<b>Includes</b>	UC 6 (Choose difficulty)
<b>Extensions</b>	None

<b>ID</b>	<b><u>UC 6</u></b>
<b>Name</b>	Choose Difficulty
<b>Description</b>	Allow the player to pick the desired difficulty in a new game
<b>Actor</b>	Player
<b>Pre-condition</b>	Player has already chosen the map
<b>Post-condition</b>	Map difficulties displayed
<b>Main flow</b>	<ol style="list-style-type: none"> <li>1. Player given option to select map difficulty</li> </ol>
<b>Includes</b>	UC 7, UC8, UC 9

## Group 62

<b>Extensions</b>	None
-------------------	------

<b>ID</b>	<b><u>UC 7</u></b>
<b>Name</b>	Easy Difficulty
<b>Description</b>	Player is playing game on easy
<b>Actor</b>	Player
<b>Pre-condition</b>	Player has chosen the difficulty
<b>Post-condition</b>	Map ready to start
<b>Main flow</b>	<ol style="list-style-type: none"> <li>1. Player selected difficulty</li> <li>2. Map ready to start</li> </ol>
<b>Includes</b>	UC 10 (Start Game)
<b>Extensions</b>	None

<b>ID</b>	<b><u>UC 8</u></b>
<b>Name</b>	Normal Difficulty
<b>Description</b>	Player is playing game on normal
<b>Actor</b>	Player
<b>Pre-condition</b>	Player has chosen the difficulty
<b>Post-condition</b>	Map ready to start
<b>Main flow</b>	<ol style="list-style-type: none"> <li>1. Player selected difficulty</li> <li>2. Map ready to start</li> </ol>
<b>Includes</b>	UC 10 (Start Game)
<b>Extensions</b>	None

<b>ID</b>	<b><u>UC 9</u></b>
<b>Name</b>	Hard Difficulty
<b>Description</b>	Player is playing game on hard
<b>Actor</b>	Player
<b>Pre-condition</b>	Player has chosen the difficulty
<b>Post-condition</b>	Map ready to start
<b>Main flow</b>	<ol style="list-style-type: none"> <li>1. Player selected difficulty</li> </ol>

## Group 62

	2. Map ready to start
<b>Includes</b>	UC 10 (Start Game)
<b>Extensions</b>	None

<b>ID</b>	<b><u>UC 10</u></b>
<b>Name</b>	Start Game
<b>Description</b>	After player settled desired options the game will finally start
<b>Actor</b>	Player
<b>Pre-condition</b>	Player decided map and difficulty
<b>Post-condition</b>	Map started
<b>Main flow</b>	1. Player selected options 2. Map started
<b>Includes</b>	UC 12, UC 13 (Lose Game, Win Game)
<b>Extensions</b>	UC 11 (Pause)

<b>ID</b>	<b><u>UC 11</u></b>
<b>Name</b>	Pause
<b>Description</b>	When the player is playing the game and want a break or want to exit or save the game
<b>Actor</b>	Player
<b>Pre-condition</b>	Player is currently playing the game
<b>Post-condition</b>	Game paused
<b>Main flow</b>	1. Player decides to take a break 2. Game is paused
<b>Includes</b>	None
<b>Extensions</b>	UC3, UC 14, UC 15, UC 16 (Help/How to play, Exit to menu, Continue, Save game)

<b>ID</b>	<b><u>UC 12</u></b>
<b>Name</b>	Lose Game
<b>Description</b>	Player lose the game after reaching a limit
<b>Actor</b>	Player
<b>Pre-condition</b>	Player is playing the game

## Group 62

<b>Post-condition</b>	Player is not able to defend a certain amount of enemy
<b>Main flow</b>	<ol style="list-style-type: none"> <li>1. Player is playing</li> <li>2. Player cannot defend a certain amount of enemy</li> <li>3. Player loses the game</li> </ol>
<b>Includes</b>	None
<b>Extensions</b>	UC 17, UC 14 (Play Again?, Exit to menu)

<b>ID</b>	<b><u>UC 13</u></b>
<b>Name</b>	Win Game
<b>Description</b>	Player win the game after defending a certain number of rounds
<b>Actor</b>	Player
<b>Pre-condition</b>	Player is playing the game
<b>Post-condition</b>	After several rounds if player health is more than zero
<b>Main flow</b>	<ol style="list-style-type: none"> <li>1. Player is playing</li> <li>2. Player kills a needed amount of enemies</li> <li>3. Player win the game</li> </ol>
<b>Includes</b>	None
<b>Extensions</b>	UC 14, UC 18 (Exit to menu, Next Map)

<b>ID</b>	<b><u>UC 14</u></b>
<b>Name</b>	Exit to menu
<b>Description</b>	A option for the player to exit from the current map and back to the menu, game datas will be lost if player decides or forgets to save, it will also display in situation where the player lose the game or wins it
<b>Actor</b>	Player
<b>Pre-condition</b>	Player is currently playing in a map
<b>Post-condition</b>	Player wants to go back the menu
<b>Main flow</b>	<ol style="list-style-type: none"> <li>1. Player currently in a map</li> <li>2. Player pause and want to save the game or the player won or lost the game</li> <li>3. Exit to main menu option taken</li> </ol>
<b>Includes</b>	None
<b>Extensions</b>	None



## Group 62

<b>ID</b>	<b><u>UC 15</u></b>
<b>Name</b>	Continue
<b>Description</b>	Continue is an option when the game is paused by the player
<b>Actor</b>	Player
<b>Pre-condition</b>	The game is paused
<b>Post-condition</b>	Player decides to continue
<b>Main flow</b>	<ol style="list-style-type: none"> <li>1. Game paused</li> <li>2. Game unpaused</li> </ol>
<b>Includes</b>	None
<b>Extensions</b>	None

<b>ID</b>	<b><u>UC 16</u></b>
<b>Name</b>	Save Game
<b>Description</b>	Save game is an option displayed when the game is paused, and it is a very important feature to preserve the player's progress
<b>Actor</b>	Player
<b>Pre-condition</b>	Game paused
<b>Post-condition</b>	Player decides to save his progress
<b>Main flow</b>	<ol style="list-style-type: none"> <li>1. Game paused</li> <li>2. Progress saved</li> </ol>
<b>Includes</b>	None
<b>Extensions</b>	UC 19 (selected saved game)

<b>ID</b>	<b><u>UC 17</u></b>
<b>Name</b>	Play Again?
<b>Description</b>	Option displayed when the player lose the game
<b>Actor</b>	Player
<b>Pre-condition</b>	Player lost the game
<b>Post-condition</b>	Player decides to play or no to play again
<b>Main flow</b>	<ol style="list-style-type: none"> <li>1. Player lose the game</li> <li>2. System displays play again message</li> <li>3. Player play again</li> </ol>

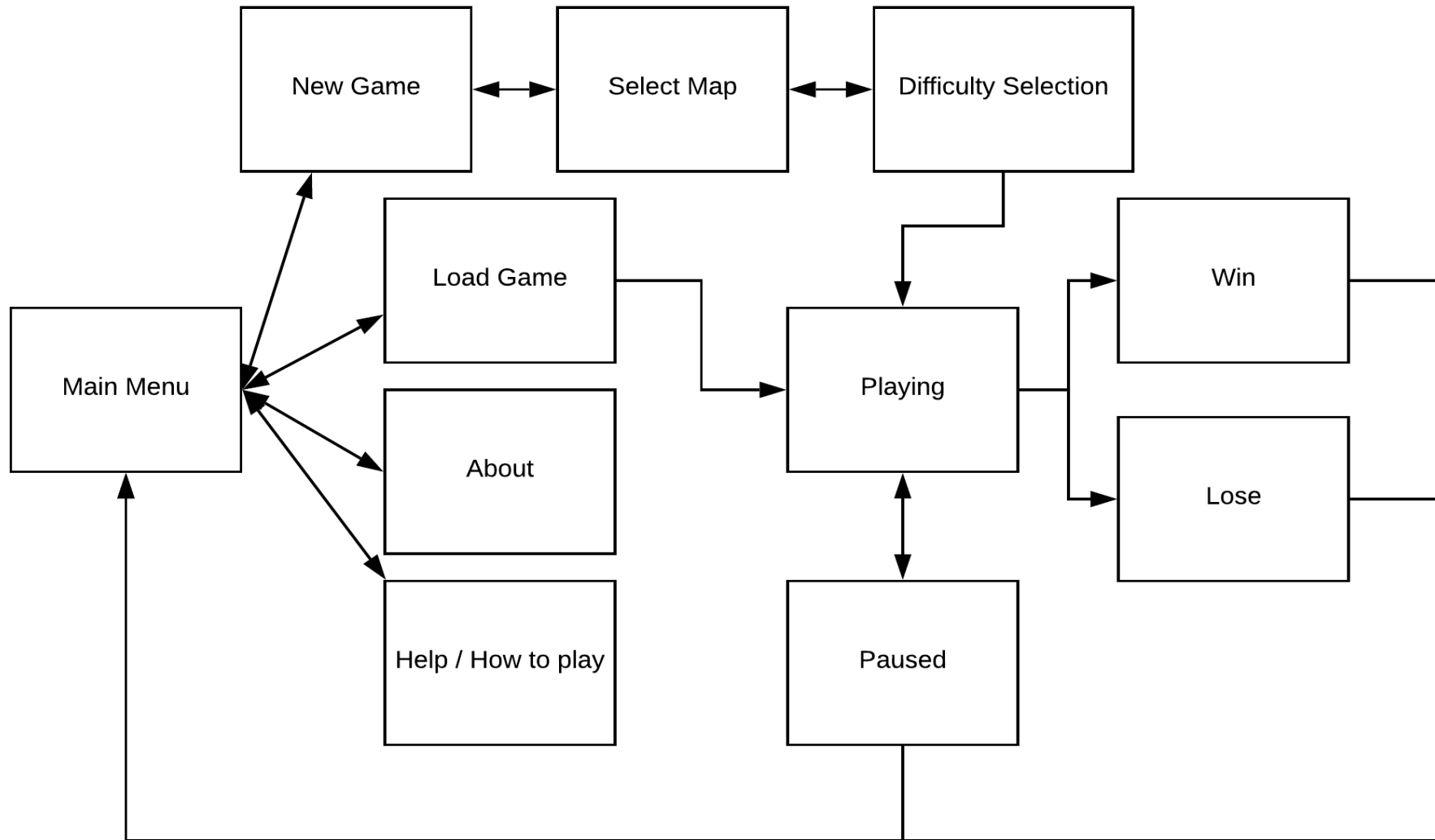
## Group 62

<b>Includes</b>	None
<b>Extensions</b>	None

<b>ID</b>	UC 18
<b>Name</b>	Next Map
<b>Description</b>	Option displayed when the player wins the game
<b>Actor</b>	Player
<b>Pre-condition</b>	Player wins in current map
<b>Post-condition</b>	Player decides to play the next map
<b>Main flow</b>	<ol style="list-style-type: none"> <li>1. Player wins in current map</li> <li>2. Next map option displayed</li> <li>3. Player enters in the next map</li> </ol>
<b>Includes</b>	None
<b>Extensions</b>	None

<b>ID</b>	<u>UC 19</u>
<b>Name</b>	Selected saved game
<b>Description</b>	Player in the menu can choose a saved progress through the load game option
<b>Actor</b>	Player
<b>Pre-condition</b>	Player has saved at least one progress previously
<b>Post-condition</b>	Map ready to start
<b>Main flow</b>	<ol style="list-style-type: none"> <li>1. Player select saved game</li> <li>2. Player loads in saved game</li> <li>3. Map starts</li> </ol>
<b>Includes</b>	None
<b>Extensions</b>	None

## Process and states of the game



The above diagram shows the states in which the games will be in. You can see that when you access the main menu, you can create a new game, load the game, go to the about screen and final the help screen. Once you are on all these states, you can go back to the main menu. However, if you proceed onto another state, for example, the playing state, you are not able to navigate directly back to the main menu without pausing the game, winning the game or losing the game.

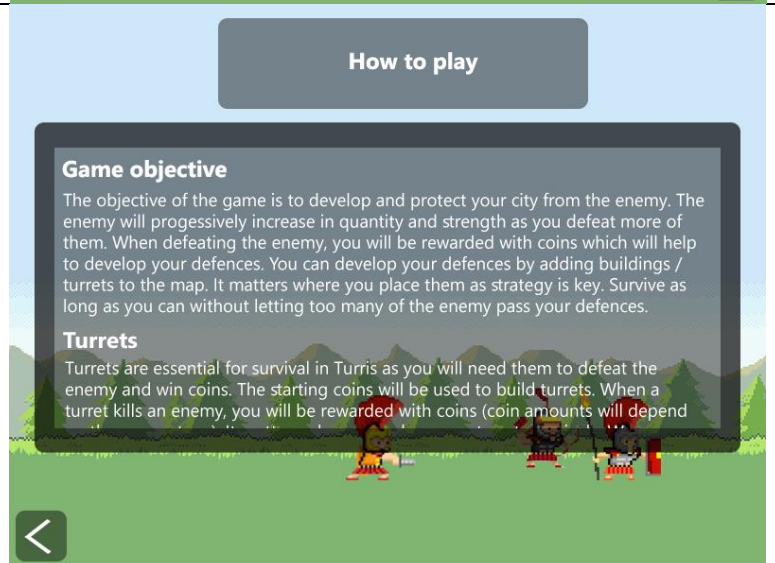
## User interface

This is the main menu state. When you start the game. This is the first thing you'll see. As you can see you will have access to:

- New Games
- Loading old games
- Document on how to play the game
- About screen
- Settings menu



The how to play menu will go into detail about not only how to play the game, but it will also show you how to be much better. You can revert to the main menu by pressing the bottom left button.

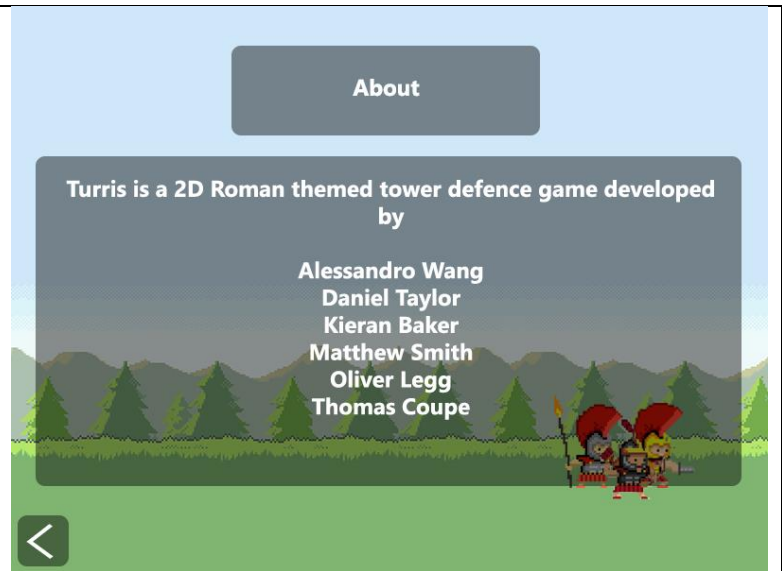


The settings menu will have technical settings and options that you might want to select. A darker coloured button will mean that setting is active, whereas a light button will mean that setting is off or inactive. For example, in the picture you will see that fullscreen is on and vsync is off. The music is set to 33% as you can see from the slider and the sfx is set to 78%. This is clarified by the label on the button too.

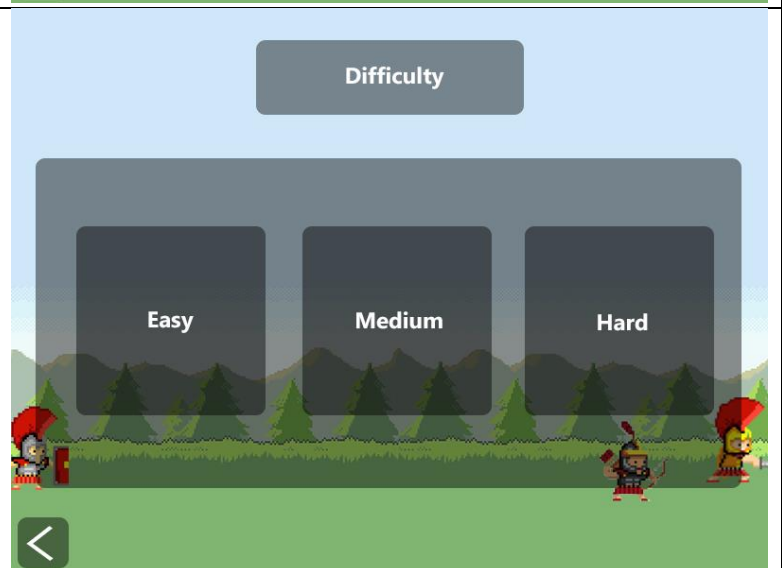


## Group 62

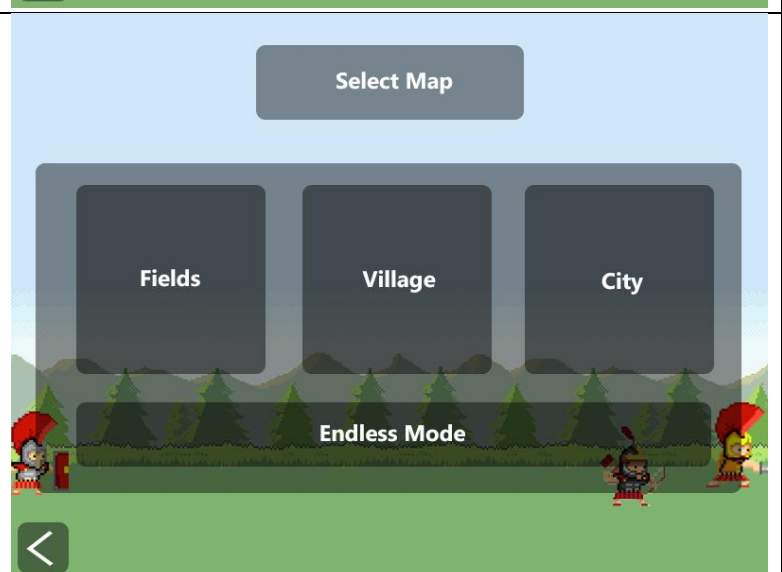
This is the about menu which shows what the game is about and who developed the game.



The player selects the difficulty of the game here. They're also able to revert to the main menu.

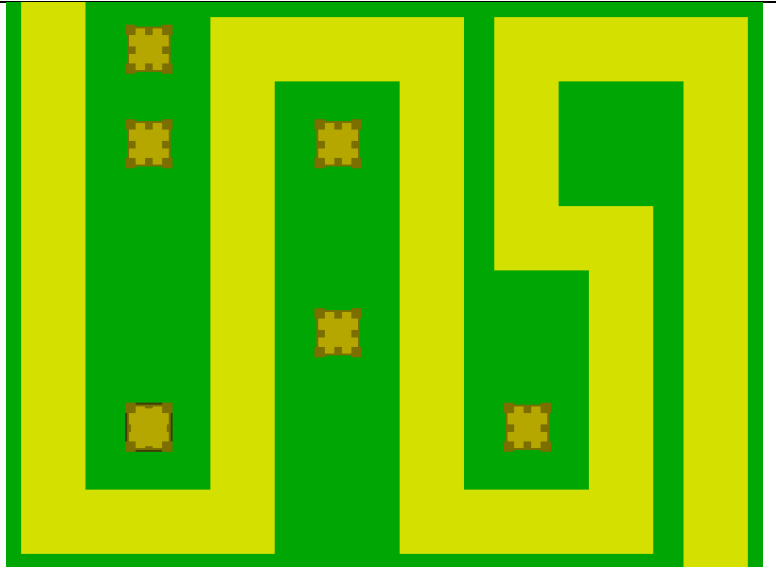


The selection of the map is here, they have 3 choices (and might be given more in the future), there is also an endless mode which selects a special map.

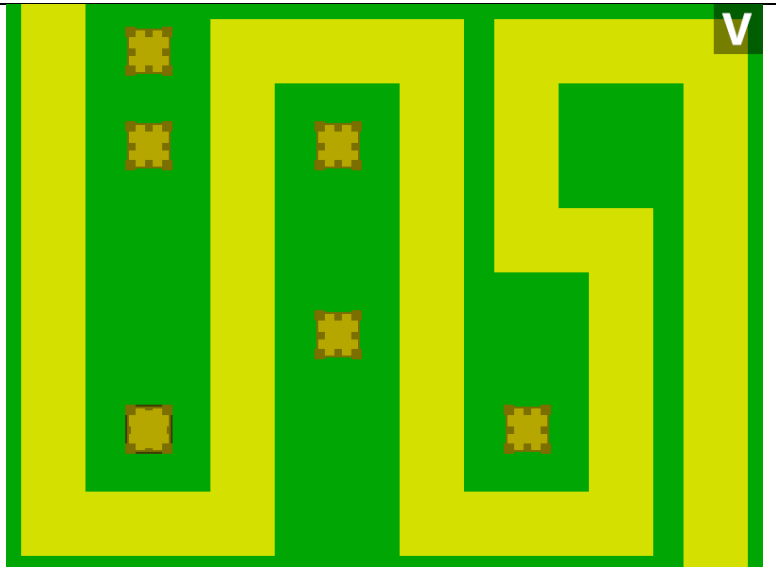


Group 62

This is the playing state of the game. This shows the path that the enemies will go down in on the field. The enemies will start from the top left and finish at the bottom left. Anywhere where the grass is on the grid, you will be able to place turrets. As you can see, turrets are on the field.

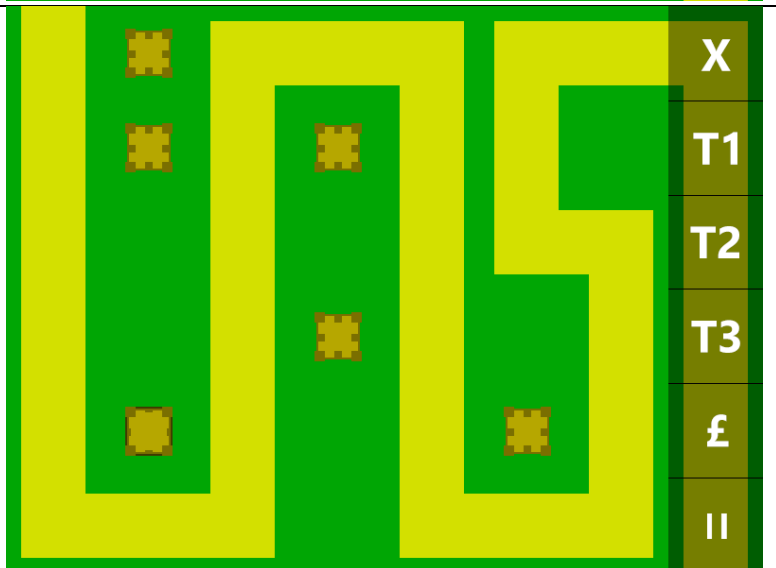


This is the playing state of the game again, but this is taken with the GUI minimised. When you press on the button, it will create the menu shown on the next page.



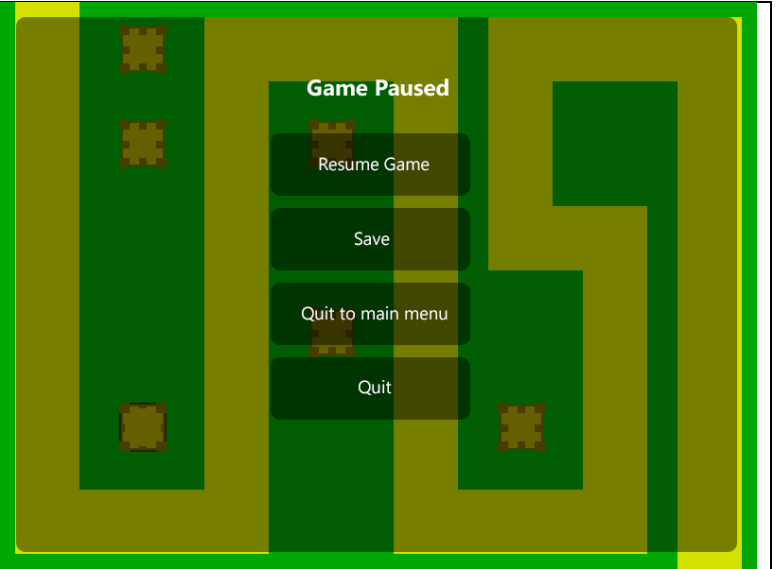
This is the GUI maximised. The button functionality works like so:

- X (closes the GUI)
- T1 (places turret tier 1)
- T2 (places turret tier 2)
- T3 (places turret tier 3)
- £ (sells) turrets
- Pause (pauses the game)



## Group 62

The game is paused here, and you can choose to resume the game, save the game and quit to the main menu. Movements of the enemies and bullets will stop as well as the shooting of turrets. Time will also not increase in endless mode when paused. The pause state can be initiated by pressing the pause button on the previous state, or by pressing the 'Esc' key.



## **Functional Description / Specification**

### **Introduction**

A Functional Description allows us to explain all the details of what we are creating to someone who has no idea or prior information regarding the project. This means that we must be very accurate when writing the specific details for our game. I have considered the feedback that we received from Sebastian Coope which was to go into a lot more detail when we are talking about our game, this makes sense as we want to be able to explain to the outside audience as clearly as possible.

### **Gameplay**

Turris is the name of the game we have come up with, we have decided that it is going to be a tower defense game with turrets / towers, different ammo types, multiple enemy types, multiple maps and 3 difficulties.

The game will include waves of enemies that approach from one side of the map and attempt to make it to the other, if a certain number of enemies reach the other side then they will defeat the player. To stop these waves of enemies, the player can place turrets that are going to be firing / attacking the enemies to stop them from making it to the end. Depending on the difficulty and or map, this can be made harder for the player.

### **Turrets / Towers**

We do not want to go over the top with the details for the game, in fact we want to be able to make it relatively simple for us to create and then make changes as we go along to improve the game. Therefore, we are setting the default number of different towers to 3, this will be useful to us when we are first trying to understand how our game is going to work.

The towers will each have a different cost, the more powerful the tower the higher the cost. It is simple to understand, however saving money to survive the higher rounds is going to be a player-based decision that could allow them to pass certain waves, it is a game of skill after all.

The towers will have a certain range, we will use grid references within a list, to calculate the distance from the towers and the enemies, again the higher cost of the tower the larger range it will have, this will give the player a large advantage as they proceed through the waves.

There are going to be multiple enemies, so placing multiple towers around the map will be a common strategy for the player due to the fact they will have to defeat many enemies at different areas of the map, obviously a single tower will not be able to fire across the whole map, so it is down to the player to decide where these towers are placed (this will again be done with grid references so that we can snap the towers into certain places, we might have harder levels where placements in some parts of the map are out of bounds).

### **Different Ammo Types**

Within the game there are also going to be different ammo types, this means that the towers can have different types of ammo that the player can either purchase or select if they have already purchased some. Obviously, this is going to be a strategic move by the player, as they are going to be the ones to decide whether they will need to purchase the ammo from the shop within the next wave of enemies. They will be able to select the different ammo types for each individual tower, they will be limited to how many times they can fire that specific ammo by how much of it they purchased. There are going to be 3 different ammo types, these will have different properties such as increased damage or can hit multiple targets.

### **Enemies**

We have to have different enemies within our game to make it a challenge and more interesting for the player, I have also stated that we do not want to go over the top and make our game too complicated as we are first building it, so again we are going to have 3 main enemies that are going to spawn in the game. Usually, the game will start off as easy on any difficulty, with the easiest enemies being the in the first few waves, however as the game progresses, we are going to introduce the other enemies that are going to be a concern to the player. The enemies will have different amount of health points that require you to hit over time to defeat them, the further the player gets in the game, the more enemies are spawned, as well as more of the higher tier / harder enemies will spawn and give the player a



challenge. These enemies might have different speeds as well, meaning they will travel through the map faster, putting pressure on the player for them to be quick and defeat the enemies before they reach the end goal.

## Maps

We have discussed the maps in detail, as we believe they are going to be the foundation of the game and how we are going to control all the tower placements, enemy pathing and other key factors that are going to create the game. We are designing the maps very basic at first to get an understanding of how we are going to move the enemies in terms of flags and grid references, however we are going to have 3 maps that will all get harder when the player completes one and moves to the next. I mentioned previously that we might have areas where a player can't place a turret, such as a river or a lake in which there is no way a player can place a turret as it is meant to be positioned on land. (We might include towers that can only be placed on water in the future to give the game a little more detail)

The maps are going to be based around a certain path, the design is up to us on how the path changes throughout the map, as there are going to be twists and turns that the enemies follow. The enemies will always follow that specific path so that the user knows where he or she can or can't place the turrets to fight the enemies. The path will change on each map and might be longer or shorter which will add to the difficulty of the game.

## Difficulties

The difficulties are self-explanatory, we are going to have Easy, Normal and Hard. In terms of how that changes the game, the difficulties will increase the health of the enemies and possibly the speed. The main thing that will change is the health of the player, as the player loses once a certain number of enemies pass through the end goal, so the amount it takes for the player to lose will lower the harder the difficulty is. Therefore, to explain a little better, Easy mode will have 150 health, Normal will have 100 and Hard will have 50, this makes the game more intense for the player and allows us to increase the difficulty of the game automatically without making any drastic changes. We can also increase the number of enemies that spawn within the levels when we change the difficulty so that it is not always the same. Hard mode might even have a bonus final round in which you have to defeat an overall boss to win the game.

We believe that adding multiple difficulties gives the player a sense of accomplishment once they have completed a map, they will then feel like they are progressing through the game as they go through the levels and increase the difficulty if they choose.

## Shop

The shop is a very important part of the game, it is the main area of player interaction outside of the map, we have to make sure that the shop is integrated perfectly into the game, so that there are no issues when the player is attempting to spend his points / currency to get a different tower or ammunition.

The shop must be very easy to access as the player will need to purchase towers as the waves of enemies are progressing, the shop will have towers and ammunition to purchase, we might update this later so that we have upgrades although that will be once the main aspects of the game are implemented.

## Currency

The currency is closely related to the shop as it is going to be how we purchase towers and ammunition, the currency will be gathered by defeating enemies, a certain amount of currency will be given to the player once an enemy has been defeated. The harder the enemy is to defeat the more currency the player will receive once it has been defeated. The player should be able to spend their currency in the shop as the game / waves are progressing, meaning we must make sure that the shop is always accessible for the player so that they can spend their earned currency.

## Pseudocode

### Turret

```
ABSTRACT CLASS Turret(x, y)
```

```
    FILE texture
    FLOAT rateOfFire
    INTEGER range
    INTEGER x
    INTEGER y
    INTEGER level
    INTEGER MAX_UPGRADE = 3
    INTEGER damage

    FUNCTION ABSTRACT upgrade()

    FUNCTION getX()
        RETURN x
    END

    FUNCTION getY()
        RETURN y
    END

    FUNCTION fire_at_enemy(enemy)
        AIM_AT(enemy)
        FIRE()
        WAIT(rateOfFire)
    END

END
```

```
END
```

```
CLASS Turret_I EXTENDS Turret
```

```
    INTEGER range = 500
    INTEGER level = 1
    INTEGER damage = 10
    FLOAT rateOfFire = 0.5

    FUNCTION create(x, y)
        PUSH_TO_ABSTRACT_CLASS(x, y, "Turret_I.png")
    END

    FUNCTION getX()
        RETURN x
    END

    FUNCTION getY()
        RETURN y
    END

    FUNCTION fire_at_enemy(enemy)
        AIM_AT(enemy)
        FIRE()
        WAIT(rateOfFire)
    END

END
```

## Group 62

```
FUNCTION upgrade()
    IF level < MAX_LEVEL THEN
        rateOfFire = rateOfFire - 0.2
        range = range + 100
        damage = damage + 2
        level = level + 1
    END
END
END

CLASS Turret_II EXTENDS Turret

    INTEGER range = 600
    INTEGER level = 1
    INTEGER damage = 20
    FLOAT rateOfFire = 0.5

    FUNCTION create(x, y)
        PUSH_TO_ABSTRACT_CLASS(x, y, "Turret_II.png")
    END

    FUNCTION getX()
        RETURN x
    END

    FUNCTION getY()
        RETURN y
    END

    FUNCTION fire_at_enemy(enemy)
        AIM_AT(enemy)
        FIRE()
        WAIT(rateOfFire)
    END

    FUNCTION upgrade()
        IF level < MAX_LEVEL THEN
            rateOfFire = rateOfFire - 0.2
            range = range + 120
            damage = damage + 3
            level = level + 1
        END
    END
END

CLASS Turret_III EXTENDS Turret

    INTEGER range = 300
    INTEGER level = 1
    INTEGER damage = 70
    FLOAT rateOfFire = 2.5

    FUNCTION create(x, y)
        PUSH_TO_ABSTRACT_CLASS(x, y, "Turret_III.png")
    END

    FUNCTION getX()
        RETURN x
    END

    FUNCTION getY()
        RETURN y
    END
```

Group 62

END

FUNCTION fire\_at\_enemy(enemy)

    AIM\_AT(enemy)

    FIRE()

    WAIT(rateOfFire)

END

FUNCTION upgrade()

    IF level < MAX\_LEVEL THEN

        rateOfFire = rateOfFire - 0.2

        range = range + 50

        damage = damage + 7

        level = level + 1

    END

END

END

## Group 62

### Enemy

ABSTRACT CLASS enemy\_1(INTEGER x, INTEGER y)

```
FILE texture
FLOAT rateOffire
INTEGER range
INTEGER x
INTEGER y
INTEGER speed
INTEGER damage
```

```
FUNCTION ABSTRACT upgrade()
```

```
FUNCTION getX()
    RETURN x
END
```

```
FUNCTION getY()
    RETURN y
END
```

```
FUNCTION move_to(INTEGER pos_x, INTEGER pos_y)
    IF pos_x != x AND x THEN
        MOVE FORWARD
    ELSE
        RETURN FALSE
    END
END
```

END

CLASS enemy\_2(INTEGER x, INTEGER y) EXTENDS enemy\_1

```
INTEGER range = 25
INTEGER x = 0.1
INTEGER y = 0.4
INTEGER speed = 0.35
INTEGER damage = 25
```

```
FUNCTION ABSTRACT upgrade()
```

```
FUNCTION getX()
    RETURN x
END
```

```
FUNCTION getY()
    RETURN y
END
```

```
FUNCTION move_to(INTEGER pos_x, INTEGER pos_y)
    IF pos_x != x AND x THEN
        +pos_x
    ELSE
        +pos_y
    END
```

```
FUNCTION takeDamage()
    IF arrowhit
        health - 100
    ELSE
```

Group 62

```
        health = health
    END
END
```

END

CLASS enemy\_3(INTEGER x, INTEGER y) EXTENDS enemy\_1

```
    INTEGER range = 35
    INTEGER x = 0.1
    INTEGER y = 0.4
    INTEGER speed = 0.50
    INTEGER damage = 35
```

FUNCTION ABSTRACT upgrade()

```
FUNCTION getX()
    RETURN x
END
```

```
FUNCTION getY()
    RETURN y
END
```

```
FUNCTION move_to(INTEGER pos_x, INTEGER pos_y)
    IF pos_x != x AND x THEN
        +pos_x
    ELSE
        +pos_y
    END
```

```
FUNCTION takeDamage()
    IF arrowhit
        health - 100
    ELSE
        health = health
    END
END
```

END

## Player

CLASS player\_shop

```
    INTEGER itemPrice
    STRING itemName
    INTEGER stock
    INTEGER playerCurrency
```

```
FUNCTION purchaseItem()
    IF itemPurchased
        playerCurrency - itemPrice
        stock - 1
        player +itemName
    ELSE
        playerCurrency = playerCurrency
    END
```

```
FUNCTION sellItem()
    IF itemSold
```

Group 62

```
        playerCurrency + itemPrice / 3
    END
END
```

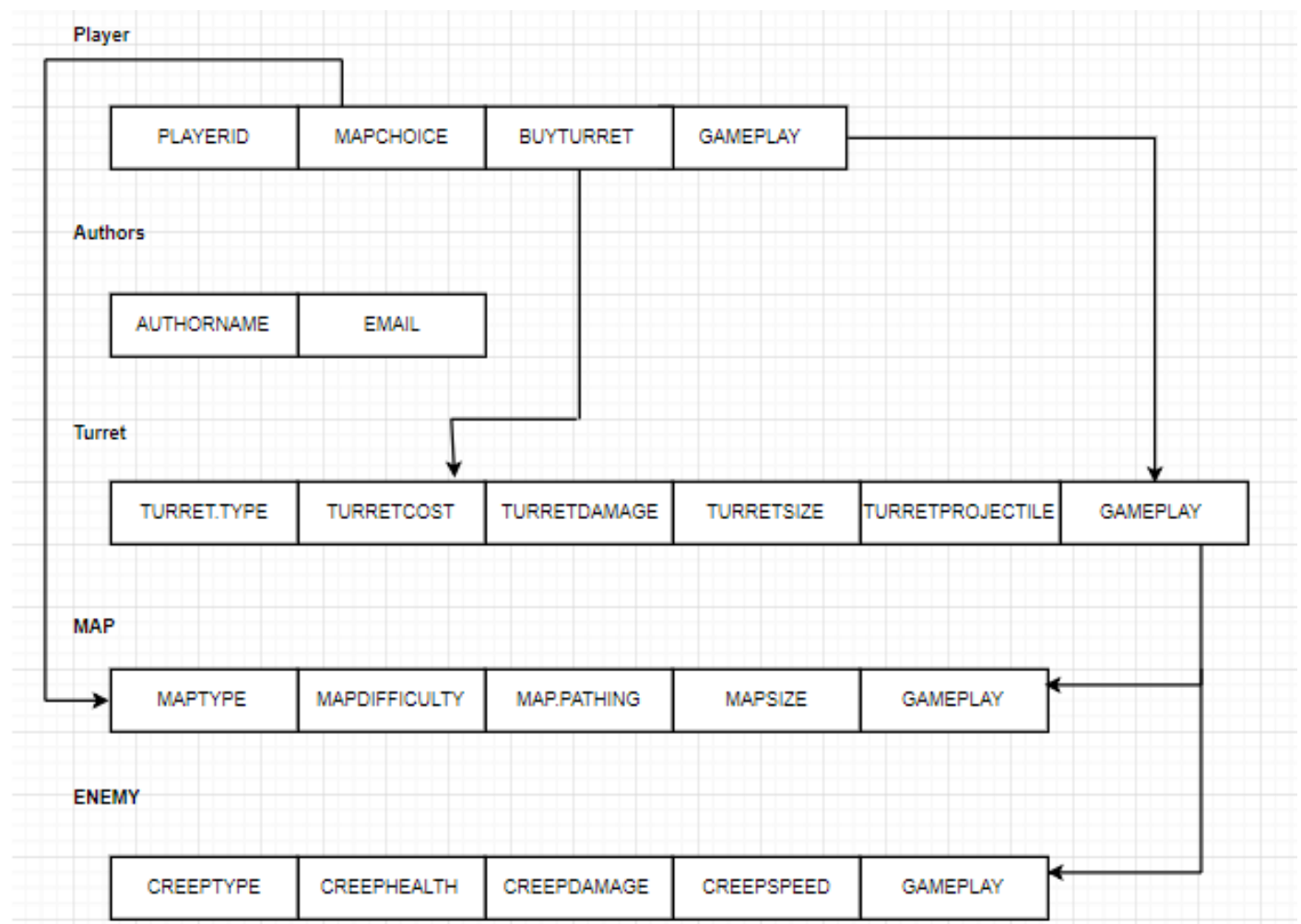
Class player

```
INTEGER playerCurrency
INTEGER playerHealth
STRING playerName
INTEGER playerLevel
INTEGER ammoCount
STRING ammoType
INTEGER towerAvailable
```

```
FUNCTION enemyKilled()
    IF enemy_health = 0
        enemyKilled +1
        playerCurrency +50
    END
```

```
FUNCTION enemyCrossed()
    IF enemyAtEnd
        playerHealth - 1
    END
END
```

## Table Structures





## Group 62

### Player

PLAYERID	MAPCHOICE	BUYTURRET	GAMEPLAY

### Authors

AUTHORNAME	EMAIL

### Turret

TURRET.TYPE	TURRETCOST	TURRETDAMAGE	TURRETSIZE	TURRETPROJECTILE	GAMEPLAY

### MAP

MAPTYPE	MAPDIFFICULTY	MAP.PATHING	MAPSIZE	GAMEPLAY

### ENEMY

CREEPTYPE	CREEPHEALTH	CREEPDAMAGE	CREEPSPEED	GAMEPLAY

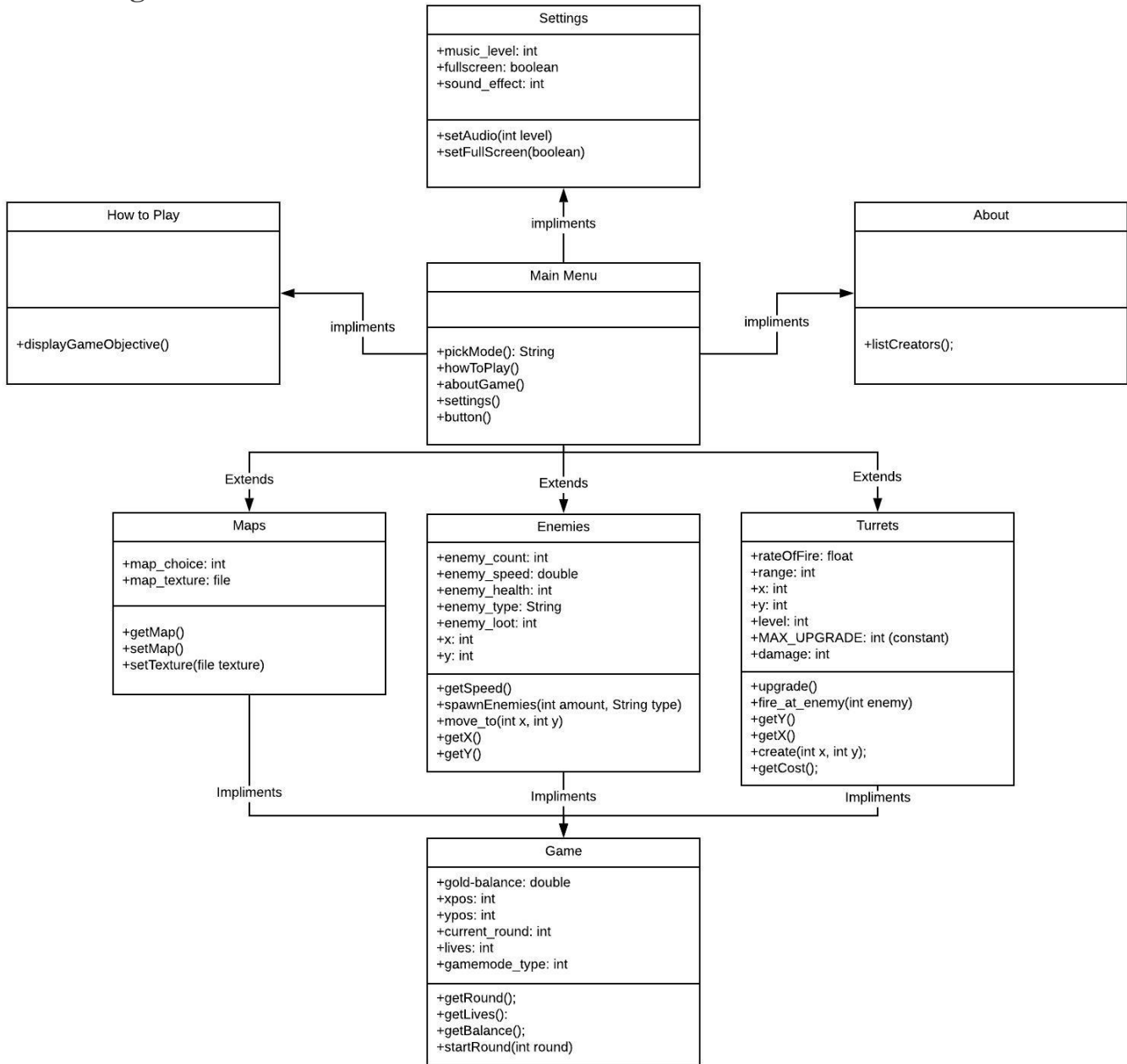
Data Dictionary - Turriss				
Name	Data Type	Length	Scope	Purpose
player_lives	Integer	1 - 20	public	This value holds the remaining number of lives the player has. Once this value is at 0 the game will be over.
gold_balance	double	0.00 - 20000.00	public	When the player kills an enemy, they receive a gold reward to buy and upgrade towers. This value represents the total amount of gold the player has.
xPosition	Integer	Width of frame	public	xPosition represents the location of where the turret is along the width of the frame.
yPosition	Integer	Height of frame	public	yPosition represents the location of where the turret is along height of the frame.
Turret_x	Integer	0-800	public	This value represents where the player is location along the x axis giving the x position of the player on the frame.
Turret_y	Integer	0-600	public	This value represents where the player is location along the y axis giving the position of the player on the frame.
Current_round	Integer	1 – 30 or if on endless mode its unlimited	public	This value represents the current round that the player is on, as the round value increases so does the enemy count.
Enemy_count	Integer	unlimited	public	The enemy count represents how many enemies are to be spawned in on that round. As there is an endless mode there is no fixed maximum number of enemies that can be spawned.
enemy_speed	Double	0.01 – 5.00	public	This value represents the speed that the enemies move along the path.
Turret1_price	Double	200.00	Public	This value represents the total cost for Turret 1
Turret2_price	Double	500.00	Public	This value represents the total cost for Turret 2
Turret3_price	double	1000.00	public	This value represents the total cost for Turret 3
Turret1_level	Integer	1 – 5	Public	This value represents the current tower level for tower 1
Turret2_level	Integer	1 – 5	Public	This value represents the current tower level for tower 2
Turret3_level	Integer	1 – 5	public	This value represents the current tower level for tower 3
turret_damage	Integer	0 - 100	public	The integer value represents the amount of damage a tower does to an enemy per each shot from the tower.
map_choice	Integer	1 - 3	public	The map selected is represented as an integer value. As there is three maps the user can choose between 1, 2 and 3 to select which map they want to play.
enemy_health	Integer	0 - 100	public	This value represents how much health each of the enemies have.
enemy_type	String	“Gladiator”, “Warrior”, “Fighter”	public	The enemy type can be chosen between three types of enemies. These enemies are defined by Strings.

## Group 62

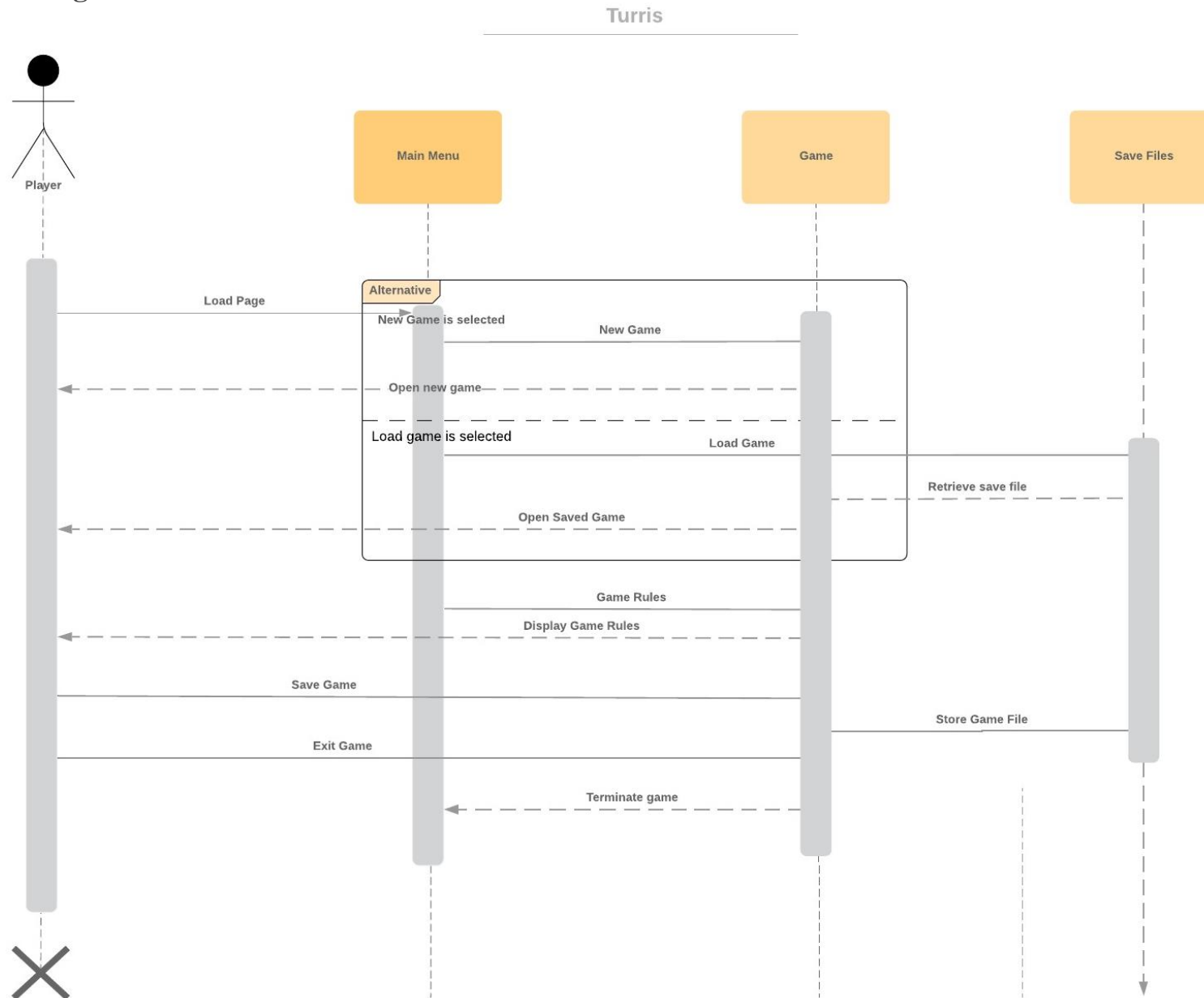
turret_range	Integer	Amount of blocks the turret can reach	public	This value represents how many blocks the tower projectiles can reach from their position.
Enemy_loot	Integer	The amount of gold an enemy drops when killed	public	This value represents how much gold the enemy will when it is
gamemode_type	Integer	1 - 2	public	This value represents which game mode has been selected.

## Group 62

# Class Diagram

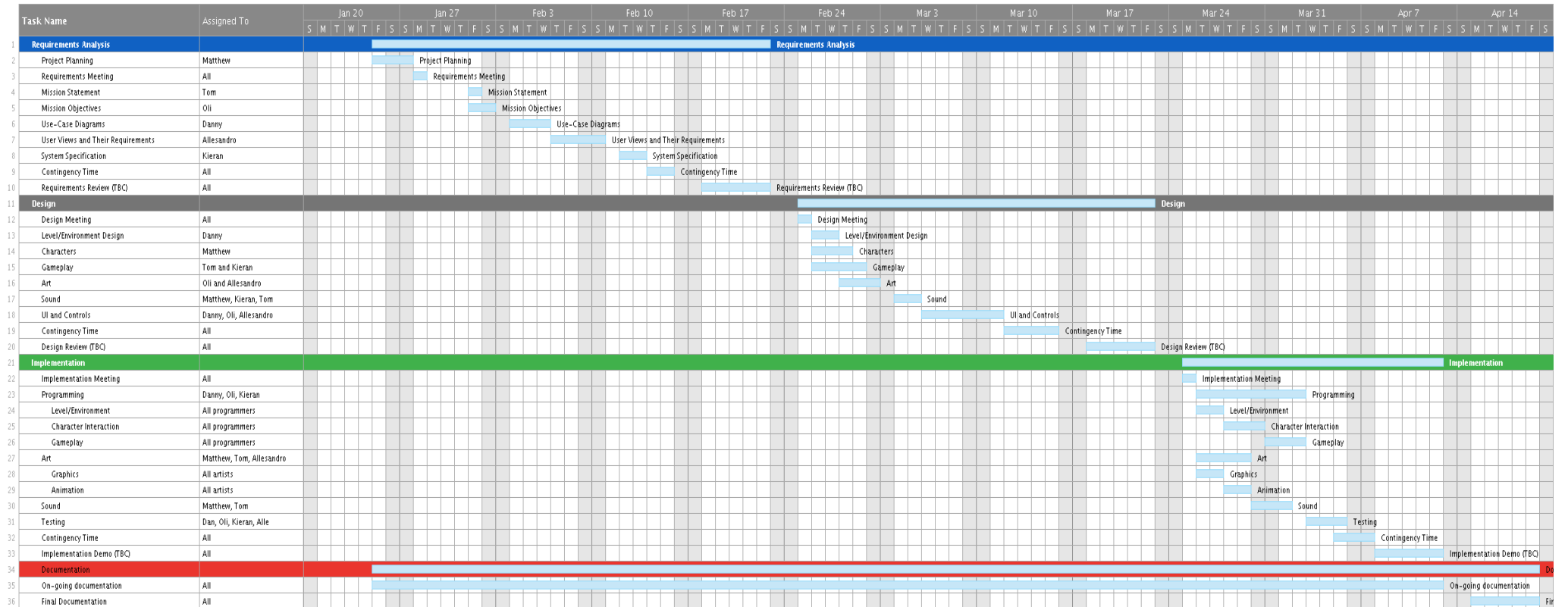


Group 62  
Message Sequence Diagram



Group 62

**Original Gantt Chart**

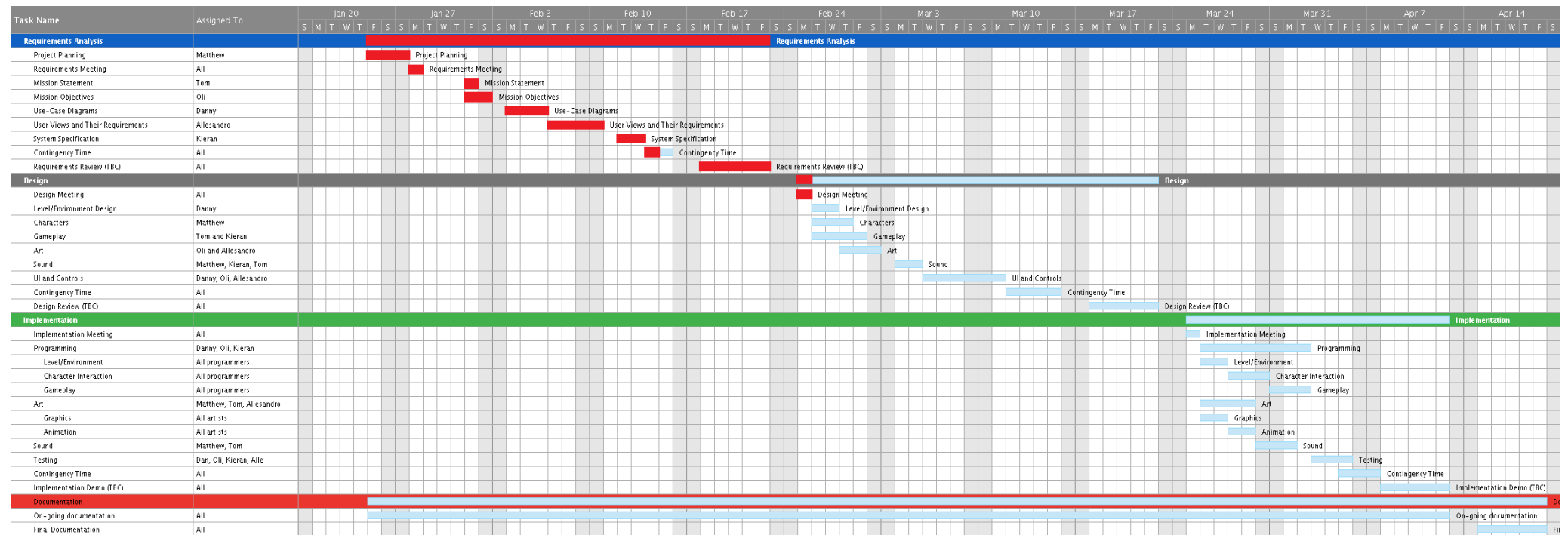


## Group 62

## Updated Gantt Chart

Up to now we have completed every task within the time constraints set out by our original plan. This includes all tasks associated with the requirements analysis. We had our review of our requirements document and have been given valuable feedback. Currently we are on track to completing our design document. The review has already been decided to take place on 22<sup>nd</sup> March.

We have not made any major changes to the Gantt chart as when we had our design meeting, every task we discussed fulfilled the plan. We have indicated in red what tasks have already been completed. You may notice we did not use all our contingency time for requirements analysis. This is because we worked efficiently and didn't come across too many hurdle (this has been indicated on the Gantt Chart below).

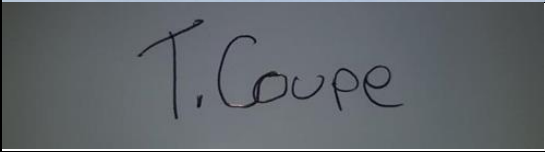

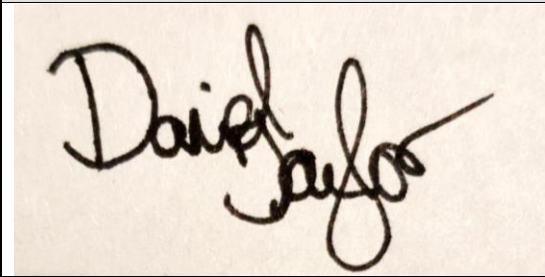


Group 62

**Contributing Team members:**

- Oliver Legg
- Daniel Taylor
- Thomas Coupe
- Matthew Smith
- Kieran Baker
- Alessandro Wang

**Signatures:**

<u>Alessandro Wang</u>		
<u>Thomas Coupe</u>		
<u>Matthew Smith</u>		
<u>Oliver Legg</u>		
<u>Daniel Taylor</u>		
<u>Kieran Baker</u>	