

Graded Unit 2: Development Document

Stephan Aldhous

Contents

[Analysis 4](#_Toc451359196)

[Impact of requirements 4](#_Toc451359197)

[Customizable Towers 4](#_Toc451359198)

[Art Style 4](#_Toc451359199)

[Login Mechanics 4](#_Toc451359200)

[AI Pathfinding 4](#_Toc451359201)

[Map Generation 4](#_Toc451359202)

[Spawning Mechanics 4](#_Toc451359203)

[Areas of Risk 5](#_Toc451359204)

[AI Pathfinding 5](#_Toc451359205)

[Map Generator 5](#_Toc451359206)

[Unique enemies 5](#_Toc451359207)

[Character Combat System 5](#_Toc451359208)

[Tower Building Mechanics 6](#_Toc451359209)

[Wi-Fi or Server Issues 6](#_Toc451359210)

[Design 7](#_Toc451359211)

[Environments 7](#_Toc451359212)

[Example Map (with marked spawn locations) 7](#_Toc451359213)

[Level Environments (1 – 4, no spawn locations, just the map) 8](#_Toc451359214)

[Map Generator (Visual and Unplaceables) 10](#_Toc451359215)

[Positioning of Actors 11](#_Toc451359216)

[Examples of Actor Positions 11](#_Toc451359217)

[Explanation of positioning 12](#_Toc451359218)

[Ally spawning mechanics 12](#_Toc451359219)

[Alien Spawning Mechanics 12](#_Toc451359220)

[Tower Management Mechanics 13](#_Toc451359221)

[General Positioning 14](#_Toc451359222)

[Player Completion Paths 15](#_Toc451359223)

[Completion Path Examples 15](#_Toc451359224)

[Map Generator (Functional Paths) 16](#_Toc451359225)

[Level Flow 17](#_Toc451359226)

[Explanation 17](#_Toc451359227)

[Concept Art 19](#_Toc451359228)

[Allies 19](#_Toc451359229)

[Enemies 20](#_Toc451359230)

[Tower Parts 21](#_Toc451359231)

[Speech and Narration 24](#_Toc451359232)

[Intro 24](#_Toc451359233)

[Ending 1 24](#_Toc451359234)

[Ending 2 25](#_Toc451359235)

[Ending 3 25](#_Toc451359236)

[Narrative discussion of elements 26](#_Toc451359237)

[Theme 26](#_Toc451359238)

[Goals 26](#_Toc451359239)

[Elements of the Environment 26](#_Toc451359240)

[Components 26](#_Toc451359241)

[Testing 28](#_Toc451359242)

[Test Strategy 28](#_Toc451359243)

[Test Plan 28](#_Toc451359244)

[Test Data (See Excel Document) 28](#_Toc451359245)

# Analysis

## Impact of requirements

### Customizable Towers

Towers must be ‘constructable’ by the player and built piece by piece. This is a major unique selling point of the game, that there are no pre-set towers. The idea of custom made towers also ties in to how the game ends, by judging how you played it or rather the play style you used.

### Art Style

The art style used in the game will also impact how the player feels while playing. A pixel based style will give a good retro feeling to the game and also create an enjoyable experience. This would have an impact on the game for the player.

### Login Mechanics

The player should be able to login and play. This is however limited to game progress and achievement progress since saving towers was far too time consuming.

### AI Pathfinding

The characters in the game must be able to find and follow paths made of tiles. This is because the player is charged with guiding the VIP to the finish and the enemies must track the VIP as they travel. The VIP’s guards must also be able to find the VIP if the finish a fight and follow the VIP generally.

### Map Generation

The ability to generate a map is required to make the game enjoyable and varied. This means the player will never (or rather extraordinarily unlikely) to generate the same map. This makes playing the maps not only fun but hook the player to replay since every map is new.

### Spawning Mechanics

The aliens must be able to spawn in waves to attack the VIP. The best way for this is to have a spawn manager handle this, and spawn them on either side of the map on walkable tiles. On top of this the manager cannot spawn them all at once. It must spawn them smoothly over time, to make the spawning feel complete and accurate.

## Areas of Risk

### AI Pathfinding

Pathfinding mechanics may pose a sizable risk to the completion of the game since pathfinding is often known for being so work intensive to debug properly. This means lots of time must be spent to ensure it works correctly.

### Map Generator

The player must be able to generate maps. This is a risk to the project’s completion timescale wise. This is because the generator must be able to make decent and playable maps for the player to enjoy.

#### Generating the paths

The generator must generate the paths from the left of the map to the right. This is a timescale issue because there are many things that could go wrong with this. This means that lots of testing is required to ensure it is stable for the final hand in.

#### Generate Rough (Unplaceable) Terrain and Foliage

Generating unplaceable terrain must be done to limit the player from building towers everywhere. This is a risk because multiple things could go wrong with this such as generating in such a way that tower placing is impossible. The foliage is much easier to implement and does not carry much risk.

#### Generating Corners on Tiles

This is a rather large risk to timescale as texture editing is very technical and accuracy intensive. If the editing is even slightly off, it will be very noticeable. This means lots of care must be taken to ensure that this is done as accurately as possible. This of course requires large amounts of repetitive testing.

### Unique enemies

Enemies have unique appearances and also have melee and ranged combat mechanics. This allows all to attack the player’s VIP and only one that can attack towers.

#### Scout

The scout is special out of the aliens. The reason for this is that it prioritizes the destruction of towers over the killing of the ally characters. This also means it must be able to fire projectiles at the towers to damage them. This will cause extra work in trying to make the unique behaviour and new projectile just for this alien.

### Character Combat System

This is a rather large risk to the timescale of the project. This is because a combat system would not only be incredibly time consuming to test properly, but also has a very large potential for annoying and/or combat breaking bugs. Since the game has a heavy focus on tower defence, this means any bugs in the mechanics are not necessarily game breaking, but could seriously harm the player’s view and opinion of the game.

### Tower Building Mechanics

This is an area of very large risk since this will be an excessively complicated process. This would allow the player to make towers from parts and thus make them very varied. A good but more complicated and extensive example of this would be the spore creature creator.

#### Unique Projectiles and Effects

This system would require that each module within the tower have a unique effect or action. This means each module would be unique and would require coding many different behaviours.

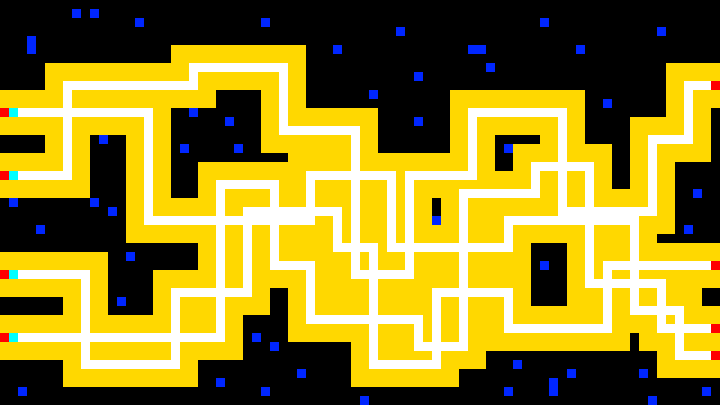
### Wi-Fi or Server Issues

If the Wi-Fi at home goes down or Moodle has server issues, uploading the final game will be stopped. This is a massive risk because, if the game and development document are not uploaded by the deadline, this entire project will be a failure.

# Design

## Environments

### Example Map (with marked spawn locations)



Black – Unplaceable tiles (Towers cannot be placed on these)

Yellow – Placeable tiles (Towers can be place on these tiles)

White – Walkable tiles

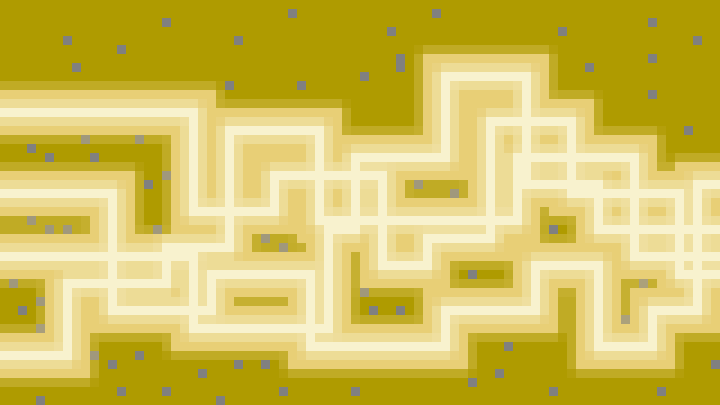
Deep Blue – Map foliage

Cyan – Potential Ally Spawn Points

Red – Potential Enemy Spawn Points

### Level Environments (1 – 4, no spawn locations, just the map)

#### Level 1 Map



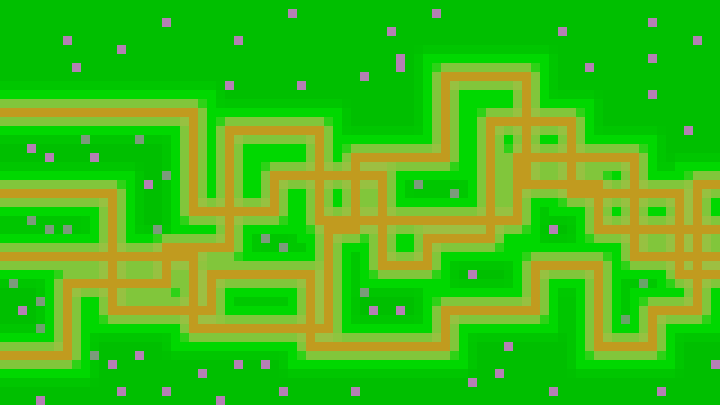
Brown – Unplaceable tiles (Towers cannot be placed on these)

Cream – Placeable tiles (Towers can be place on these tiles)

Magnolia – Walkable tiles

Dark Grey – Map foliage

#### Level 2 Map



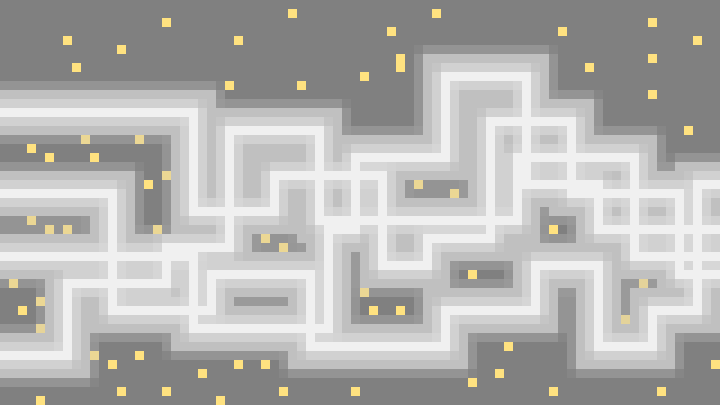
Deep Green – Unplaceable tiles (Towers cannot be placed on these)

Green – Placeable tiles (Towers can be place on these tiles)

Brown – Walkable tiles

Grey-Purple – Map foliage

#### Level 3 Map



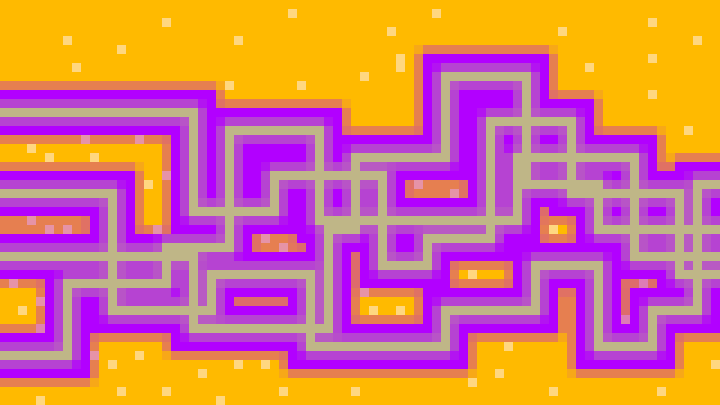
Dark Grey – Unplaceable tiles (Towers cannot be placed on these)

Grey – Placeable tiles (Towers can be place on these tiles)

Light Grey – Walkable tiles

Golden-Yellow – Map foliage

#### Level 4 Map



Bright Orange – Unplaceable tiles (Towers cannot be placed on these)

Purple – Placeable tiles (Towers can be place on these tiles)

Dark Cream – Walkable tiles

Very Light Orange – Map foliage

### Map Generator (Visual and Unplaceable Tiles)

#### Making ‘Unplaceable Tiles’

The generator will iterate through all the map tiles. On each iteration it will iterate through a random sized box of neighbours (this box being a search through tile indexes). If a single walkable tile is found, the algorithm will break out the method and move on to the next tile. If none are found, the method will make the current tile unplaceable at the end of the loop. This will continue until all tiles in the map have been iterated through.

#### Making the foliage

Again the map generator will iterate through all the map tiles. If the tile is unplaceable then there is a 1 in 10 chance of turning the current tile into a foliage tile.

#### “Rounding” the corners

Once all the tiles have been dealt with in the map generation, some aesthetic touches are added to make the map feel more complete and polished. This was done by “rounding” the corners. More accurately, where corners differ, the tile frames are blended and makes the corner look like it has been sliced at a 45 degree angle. This gives the impression of a round corner until zoomed in, however even then, the corners still look nicely finished.

##### The process

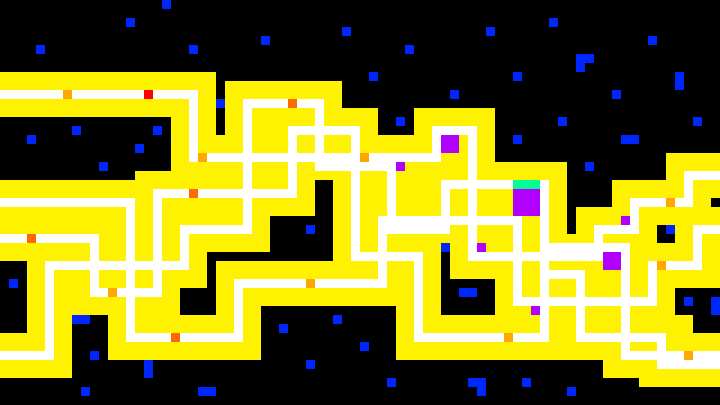
The way the corners are created is by getting the adjacent tile frame and blending it with the current tile frame. The algorithm searches through all the tiles in the map. For each iteration, it will look at the tiles neighbouring the current tile (The neighbouring tiles are connected. For example, for the top left corner, look at the tiles to the left and above.). It will look at the neighbouring tiles of each corner of that tile and move to the next.

The first thing that is done is to check if the corner has tiles that have lower indexes than the current tile’s index. If so, it will get the current tile index and the highest neighbouring index. Then it creates a new texture for that tile. It does this because of the way textures are loaded using the XNA Framework. The XNA Framework loads the texture into a memory location on the first load call, then any other code that tries to load that texture, is given the same memory to save memory space. This is a problem, because if a corner is edited on one tile, it gets edited on all tiles as well since they share the texture memory. This is why a new texture is made for the current tile and it is given the old textures pixel data.

The algorithm then executes a method to take the corner data of the tile frame of the higher index found. Then sets the corner of the current tile frame with the corner data of the other frame. This way the frames blend nicely and also wrap nicely at edges, since the corner made is consistent with the rest of the area.

## Positioning of Actors

### Examples of Actor Positions



Black – Unplaceable tiles (Towers cannot be placed on these)

Yellow – Placeable tiles (Towers can be place on these tiles)

White – Walkable tiles

Deep Blue – Map foliage

Turquoise – Friendly characters

Purple – Towers (Varying sizes)

Light Orange – Regular Aliens

Orange – Scout Aliens

Red – Tank Aliens

### Explanation of positioning

#### Spawning

At each side of the map aliens will spawn on walkable tiles. After they spawn, they will look for the VIP. If they cannot find the VIP, they will reposition themselves and look again. This will occur until they do find the VIP, at which point they will move to intercept them.

At the left side the VIP and his guards will spawn. From here the player must guide the VIP to the other side of the map. Once at the other side the level will be considered complete.

#### Towers

The towers cannot be placed just anywhere on the map. They can only be placed on what are called “Placeable Tiles” (Yellow on diagram). This means they cannot be placed intersecting the path (white on diagram) or on rough terrain/foliage (black for rough terrain on the diagram, deep blue for foliage on diagram). This is a way of not only making the map look nice but also limit placement locations for larger towers.

### Ally spawning mechanics

When the game map is generated, the player ally characters (VIP and the guards) are placed on the map (and added to a list of friendlies).

This is done by searching through the leftmost tiles, from top to bottom. Any tiles that are walkable are added to a list of walkable tiles to select from. Then, a random one of these tiles are selected to spawn the allies on. Then the VIP is placed first, a tile out from the selected tile. Then both guards are added to either side of the VIP.

### Alien Spawning Mechanics

When the spawn manager is constructed, it creates alien spawn zones for the aliens to spawn in. The reason these are needed is to prevent aliens from spawning next to the VIP. These spawn zones are created on both sides of the map.

When the game is in play and no enemies are spawning, a timer counts down till the next wave (30 seconds). When the next wave begins spawning aliens, the numbers of each alien spawning and the time between spawning them onto the map will be different. When spawning begins, a boolean value is then set to true so that waves don’t spawn into another wave already spawning.

First the system finds a random spawn zone to spawn the alien, then checks if the VIP is with the zone. If so, then pick another spawn zone. Otherwise spawn an alien on that tile and decrement the number of that type of alien to spawn. Then resets a mini timer for that alien to type (prevents spawning all at once). The countdown to spawn another alien begins again. This cycle continues until the number of that alien to spawn hits 0. This process occurs for all three types of alien.

When all aliens have been spawned, set the spawning boolean value to false, reset wave time, reset mini timers and increment the current wave and start the process over.

### Tower Management Mechanics

#### Buy and Placing Mechanics

If a tower button is pressed, the tower placing mechanics activate. When this happens, a Boolean value for placing is set to true. When placing or moving is true, display overlay set of tiles. These are the tiles that will be referred to from now on here. Then the system gets the dimensions of the tower and checks the surrounding tiles where applicable if they are placeable (and not walkable). It also highlights the tiles with another colour appropriate to the “placeableness” of the tile.

If all the tiles within the area of the tower dimensions are placeable and not walkable, then allow the player to place the tower there, root the tower into those tiles and take the appropriate amount of cash from the player.

#### Moving Tower Mechanics

When the movement button is pressed on a tower menu, the moving bollean is set to true and the tower is “uprooted”. Once this is the case, the system will check the “placeableness” of the neighbouring tiles surrounding the tile that contains the mouse. If they are all placeable, then if the left mouse button is pressed, then root the tower on those tiles and thus the tower is placed.

#### Checking Surrounding Tiles

The method starts by declaring a minimum and a maximum variables. Then adjusts them to fit the placing/moving tower’s dimensions. If the dimensions are 1, only check the mouse over tile. If the dimensions are 2, it will check the mouse over tile and the tiles to the right and bottom of the tile (including the corner). If the dimensions are 3, it will check the mouse over tiles as well as the surrounding tiles (including the corners). If all tiles are not walkable and placeable, set the canPlace variable to true to allow for placement of the tower in question.

#### Rooting and Uprooting Towers

The rooting and uprooting works in a similar fashion to the checking of surrounding tiles. It would get the dimensions of the tower. If the tower is being moved then it would set all the checked tiles to unoccupied. If the tower is being placed in that location, then set all tiles within the checked are to being occupied.

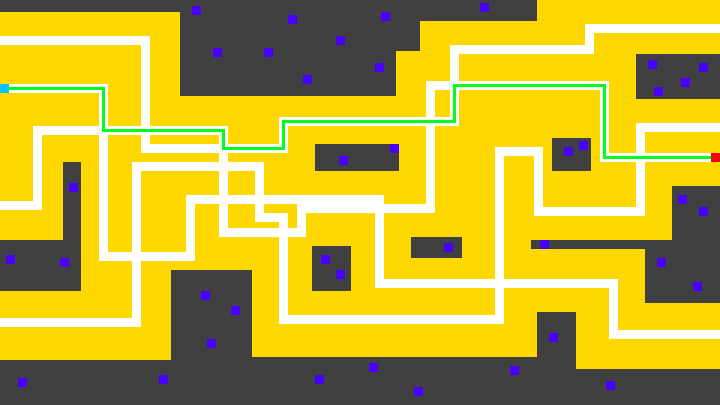
### General Positioning

Aliens will only ever reside on the walkable paths of the map. This same goes for the VIP and guards. The towers can only be placed on tiles between the rough terrain and walkable paths.

## Player Completion Paths

### Completion Path Examples

#### Open Path



Dark Grey – Unplaceable tiles (Towers cannot be placed on these)

Yellow – Placeable tiles (Towers can be place on these tiles)

White – Walkable tiles

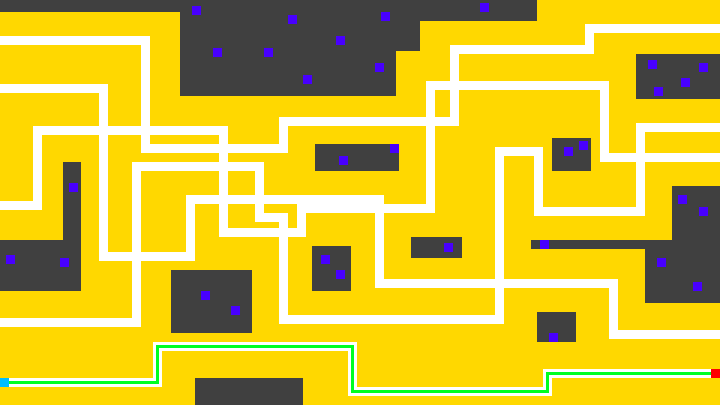
Blue – Map foliage

Light Blue – VIP and guards start

Red – VIP end goal

Green – Completion path

#### Isolated Path



Dark Grey – Unplaceable tiles (Towers cannot be placed on these)

Yellow – Placeable tiles (Towers can be place on these tiles)

White – Walkable tiles

Blue – Map foliage

Light Blue – VIP and guards start

Red – VIP end goal

Green – Completion path

### Map Generator (Functional Paths)

The map generator generally works based around integers and switch statements.

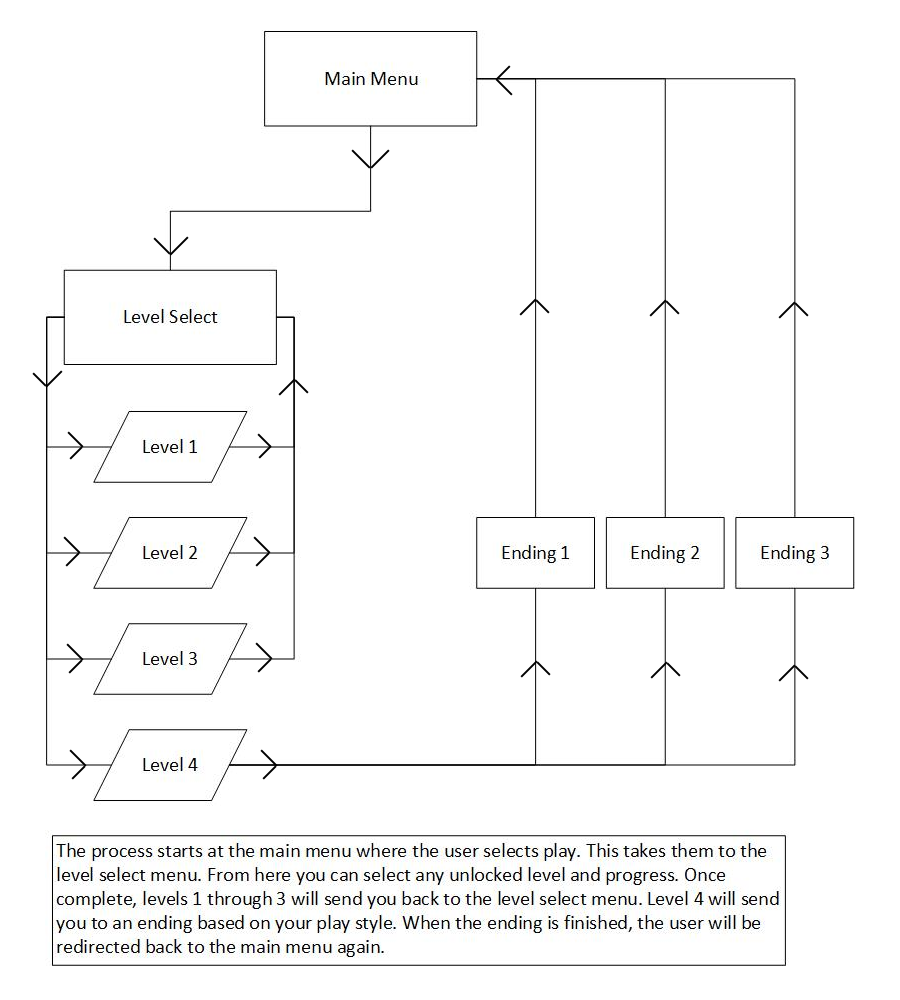
#### Picking random number of paths

The generator picks a random number of paths to generate (ranging from 3 to 6). Then it tries to start path creation by picking a random tile on the left (and making sure that it and the neighbouring tiles are not walkable).

#### Generating a path

Once the starting index is picked, the path generator will always generate a horizontal section first. It will generate a random length (ranging from 3 to a number the player chooses [3 – 30]) of walkable tiles to the right. After this it will pick a random number ranging from 0 to 2 (0 = up, 1 = right, 2 = down), if the number selected is the same as the current direction, it will generate another until it gets a valid direction. After the new direction is picked, it will then calculate a new length and so on. This process will continue until the generated path reaches the right hand side of the map. When this happens the generator will start a new path and continue until the number of paths required are made.

## Level Flow



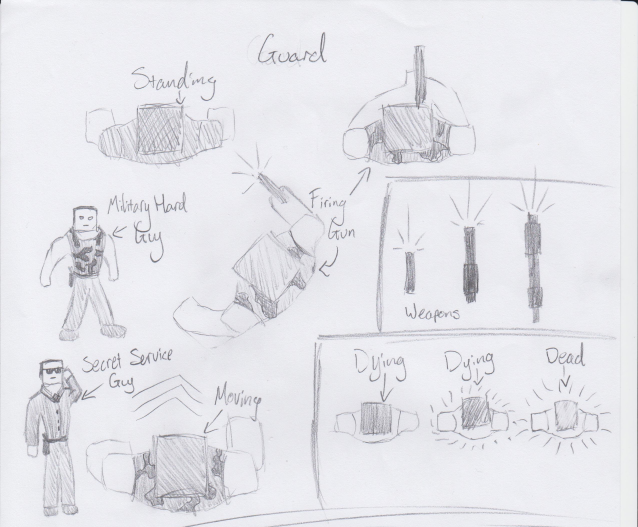
### Explanation

The player starts at the main menu. When they press the play button, the player will be taken to the level select menu where they can select a level to play. In order to play a level it must be unlocked first. Once a level is complete it will take you back to the level select menu and increment level progress and thus unlock the next level. Levels can also be replayed. Once the 4th level is complete, the appropriate ending will be calculated and shown to the player. The difficulty will also be incremented at the end of the game so the player can replay on a harder mode. Once the 4th level is complete, progression will reset and the player will restart the levels and the difficulty is incremented.

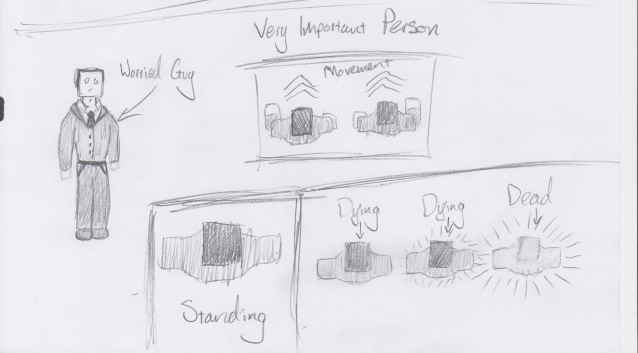
## Concept Art

### Allies

#### Guards

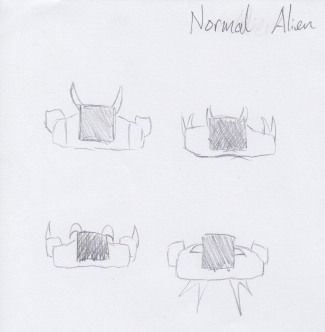


#### VIP

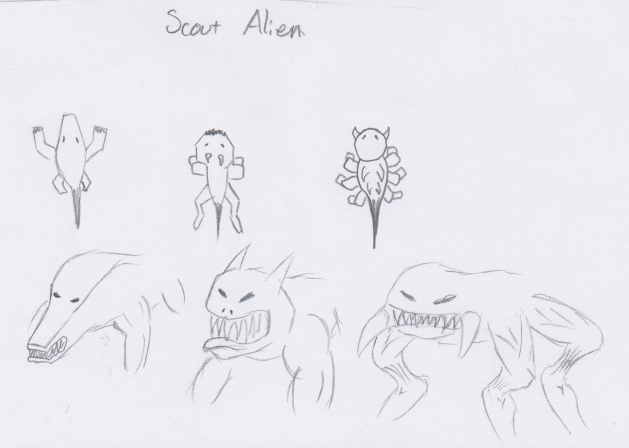


### Enemies

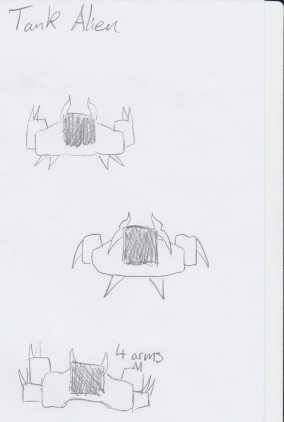
#### Regular Alien



#### Scout Alien

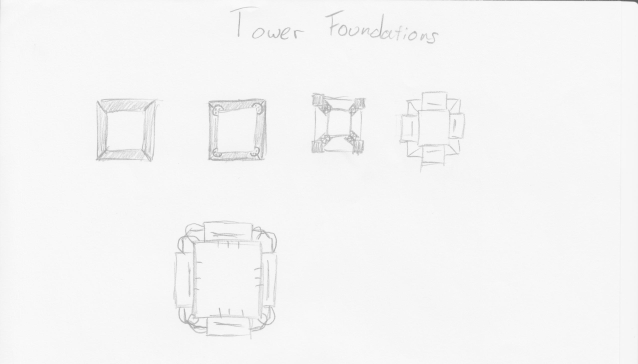


#### Tank Alien

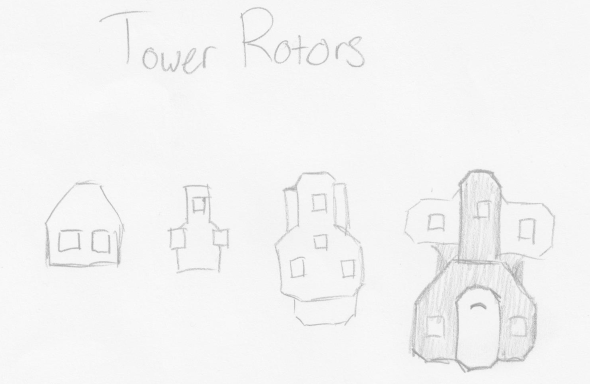


### Tower Parts

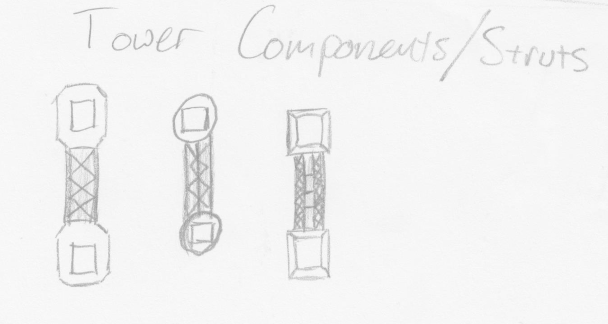
#### Foundations



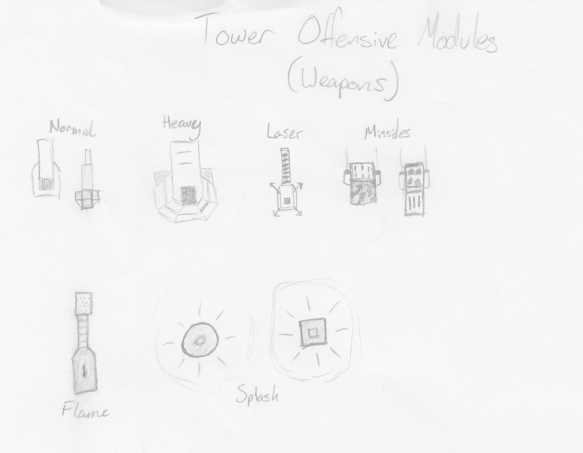
#### Rotors



#### Components / Struts



#### Weapons



#### Utilities / Defences



## Speech and Narration

### Intro

This is the first cut scene the player will see as they start the game.

#### Narrative Text

“In the middle of nowhere, a rift opens…”

“Pouring out of the rift, creatures never seen before in the universe.”

“They travelled to human populated planets, destroying the civilizations residing on them…”

“Only to disappear as quickly as they arrived.”

“However, standing between them and human extinction…”

“Is YOU, whoever you are!”

“Escape this hell and ensure humanity’s survival!”

### Ending 1

In this ending, humanity curses and accuses the other races in the universe for the disaster. The human race then begins a crusade across the stars, annihilating the other races. After eons, and many evolutions, the humans have become god-like beings. Realizing the magnitude of their actions they attempt to set things right by sending minions to destroy themselves. This then completes the time loop.

#### Narrative Text

“As you escape, you vow to unleash fury on all those that were unaffected by the crisis…”

“Believe them to be part of some wide plan to destroy humanity.”

“As time went on, species upon species would be destroyed by humanity’s crusade.”

“As more time passed, human’ evolve tremendously, becoming god-like beings. However they were severed from the fabric of conventional time.”

“Upon the realization of the magnitude of their actions…”

“They sent out the remaining beings back in time to wipe themselves out. In an effort to correct their grave mistake!”

“And with that the loop is complete!”

### Ending 2

In this ending, after the disaster, humanity disappears into the stars for good, becoming a myth or legend in the process. Those that dare search for them would return, remembering next to nothing. As if the encounter was more like a dream. After eons, the humans became celestial beings that simply observed the cosmos and its residents till the end of time.

#### Narrative Text

“Escaping their doom, humanity went on a great journey into the stars.”

“Few civilizations made contact with them from then on, and those who did could not accurately recall the event.”

“Through their enlightenment, humans figured out that it was in fact they, who sent those monsters to destroy the human race.”

“As time went, they evolved into god-like beings of incredible power, and become one with the cosmos.”

“Now cosmic entities they observed the goings on in the universe from afar, or rather from everywhere.”

“And with that an alternate reality of observers was created!”

### Ending 3

In this ending, humanity decide that no other race should have to suffer the same fate and went on to become great protectors of all life in the universe.

#### Narrative Text

“Escaping their doom, humanity decided that it was going to protect other races in the cosmos from events much like this.”

“They became great defenders, great guardians of all life. Taking up the mantle of responsibility of care and nurture.”

“And with that an alternate reality of defenders was created!”

## Narrative discussion of elements

### Theme

The theme that must be in the project is glow, however the main theme of the project is sci-fi.

#### How does the theme impact the player

The theme impacts the player through story and also helps for suspension of disbelief, because many modules in game would make absolutely no sense in real life. An example of such a case is the healing module that miraculously heals nearby allies. Another example is the time dilation module that slows nearby enemies. The theme also impacts the story heavily as the entire story is far into the realms of science fiction.

#### What makes it effective

What makes the sci-fi theme effective is its safe and interconnectedness with the theme of glow. The glow theme fits into sci-fi games and movies quite nicely and complements the feeling of the game. The main areas glow have been used are in the lasers and viewing ranges of towers and ally characters.

#### Why choose this theme

Not only does a sci-fi theme sit nicely with the extra glow theme requirement, I simply prefer this theme to something dull and boring.

### Goals

#### What makes the goals effective

The goals are made effective by forcing the player to challenge their coordination and management skills to complete the game.

#### How do the goals impact the player

The main goal of each level makes the player push their ability to manage. To manage characters and towers together is not an easy task and is required to win.

### Elements of the Environment

#### Player Impact

The environment can really affect a player’s opinion of the game and affect whether or not the player wants to play again. This also ties in with the player’s opinion on how complete the game is. If the environment looks bad art wise and is functionally unusable, the player will not

### Components

#### Towers

The towers are defensive buildings that help defend allies in the game.

##### Effect on gameplay

Towers have a rather large effect on the gameplay as they are the only real way of getting the VIP out of combat. Although guards defend the VIP, they aren’t very effective at getting the VIP out of the situation.

##### What makes the towers effective

The towers in this game are player built. This means that the parts a tower has are customizable as they are built by the player. This allows they player free reign over the kind of towers they have and is one of the key selling points of the game.

#### Enemies

The enemies are the main barrier in the game between the player and completion.

##### Effect on gameplay

The enemies come in 3 types. Regulars, Scouts and Tanks. Regulars and Tanks simply get into melee combat with any character nearby. The Scouts on the other hand will aim for the towers over the player. This means that they can be a big strategic threat.

##### What makes the enemies effective

They are made effective by the fact that the player must actively pay attention to the types of enemies he/she are dealing with.

#### Allies

##### Effect on gameplay

The allies are your lifeline. If they are not defended, you will lose. This means they are key to the game and are the primary focus of the gameplay.

# Testing

## Test Strategy

The only testing needing to be done for the game is a unit test since it is only a single player game. Aside from interruption of the testing for any variety of reasons (meal, distraction, a problem is found etc.) there isn’t any major risks to testing.

The main idea is to test various sections of GUI and gameplay in order. If a bug is found, spend time fixing it until it is fixed or it is deemed to be taking too much time to fix.

## Test Plan

The plan for the testing is to first test the login system. Then the entire menu system as fully as possible, without diving into the gameplay. Then move on to test the menus within the gameplay. Once this is done the tutorial, settings and achievements will be checked. After that the enemy AI will be tested to see if they path-find and attack properly. Then the tower building mechanics will be tested to see if each part adds correctly. Then the tower updating will be tested to see if each unique part acts as it should. Testing how they interact with the environment will be next. Next testing level completion mechanics will be done to see if each level completes correctly. Then testing of endings will be done to see if the correct endings are shown.

## Test Data (See Excel Document)