Functional Design

# Story

* Not intended as a continuation of Level 1, different assets and different idea
* Lone spaceship (Player) flies through space, avoiding and shooting Enemy ships that appear
* The character or Enemy ships are never given a name, totally anonymous and no script

# Game Mechanics

## Basic Mechanics

* Player is able to control spaceship at bottom of the screen, moving left or right
* Unable to move forwards or backwards
* Also able to rotate the ship to shoot at a different angle
* You can shoot a CONTINUOUS stream of bullets, never runs out
* The user has ONE spaceship to control, Enemy AI spawns a random amount of Enemy ships progressively through the game
* Enemy spaceships will slowly move down the screen as if flying past you, shooting you as well
* You lose a life when shot by an Enemy or hit by a ship
* You are awarded 1 point for every second you last, and 25 for each Enemy you destroy by shooting
* The game gets progressively harder as you last longer, as the enemies spawn at a faster rate
* You are able to 'warp' through space by moving to the left hand side, and coming out on the right
  + Bullets also warp through space

## Unit Mechanics

* Player
  + Uses quite a large sprite, unique and not used by anything else
  + Moves and rotates quite slowly
  + Player behaviours
    - Move left and right
    - Rotate left and right to a certain angle
    - Shoot unlimited bullets
    - Warp from left to right, right to left
    - Collect power ups
* Enemies
  + Uses a small sprite, each spaceship is the same sprite
  + Enemy behaviours:
  + Move downwards towards Player
  + Move sidewards at random intervals
  + Rotate body to be facing Player at all times
  + Fire random amount of bullets at random intervals

## Game Physics

* Player collides with Enemy
  + Player loses a life, Enemy is destroyed
* Player collides with an Enemy bullet
  + Player loses a life, bullet is destroyed
* Player hits left hand side of the screen
  + Comes out on the right hand side
* Player hits right hand side of the screen
  + Comes out on the left hand side
* Enemy collides with Player bullet
  + Enemy's life is reduced by 1, it takes 10 bullets to destroy a Enemy ship
* Enemy collides with Enemy bullet
  + Nothing happens
* Enemy hits left hand side of the screen
  + Comes out on the right hand side
* Enemy hits right hand side of the screen
  + Comes out on the left hand side
* Any bullet hits any other bullet
  + Nothing happens
* Any bullet hits the left hand side of the screen
  + Comes out on the right hand side
* Any bullet hits right hand side of the screen
  + Comes out on the left hand side

## Artificial Intelligence

* Enemies contain the only aspect of AI
  + They will track the Player's spaceship, and face it at all times
    - They will fire a random amount of bullets in that direction towards the Player
  + They will occasionally move from side to side on their way down, randomly selected

# User Interface

## Flowchart of Navigation

Main Menu

Level 1

Level 2

Pause

High Scores

Level 1 Loss

Level 1 Win

Level 2 Win

Level 2 Loss

## Screen Mock-ups

Main Menu/other menus (victory, losses)

Main Menu

Level 1

Level 2

High Scores

Exit

Pause

Paused

Resume Game

Quit Game

Level 2

Score:

Lives:

Time left:

High Scores

High Scores

1. ---
2. ---
3. ---
4. ---
5. ---
6. ---
7. ---
8. ---
9. ---
10. ---

Exit

# Art and Video

* Art style to have a distinct space and futuristic style, but with a retro feel, so everything is simple
* List of assets
  + Space background
    - Blank black canvas with occasional stars and star systems placed realistically
  + Player spaceship
    - Larger than Enemy spaceships
    - Predominantly blue
    - Needs some kind of symbol on it
  + Enemy spaceship
    - Smaller than Player spaceship
    - Predominantly red
  + Player bullets
    - Square
    - Green
    - Very small
  + Enemy bullets
    - Square
    - Red
    - Very small
* No cinematics/introductions/animations

# Sound and Music

* Sound style to be futuristic, space orientated, and ambient (as much as possible)
* List of assets
  + Main ambient background song
    - Track name by artist