Contents

Name : Arian Jafari-Tadi

Game Design Document

Project: Escape Terror onto the Stars

Platform: PC

Controls: Mouse and keyboard (maybe controller support will be added later)

Genre: 2D Platformer

X statement

My game, Escape Terror onto the Stars, is a 2D platform with a lore and artistic style inspired from Star Wars. This game draws gameplay inspiration from 2D platform games such as Super Meat Boy, in terms of the hard puzzley nature of overcoming obstacles, and Hollow Knight in terms mystical elements of the world.

Elevator statement

The story is about an Imperial Storm Trooper who is found out to be a spy. A long time ago, he had embarked on a journey to sabotage the empire from within, but in the quest of finding out more information about the Death Star, he was discovered and now he is on the run. He must fight his way out of the Death Star being guided by the player through a 2D Side Scrolling Platformer adventure, where he will need to use his abilities and weapons to fight and platform his way out of the Death Star. Using a variety of tools to sneak out of the planet Destroying monstrous of a space station, the player will be engaged in the world that he is playing in.

Razor statement (with images for reference)

1. Challenging platforming puzzles similar to Super Meat Boy

2. Fighting different types of enemies along the way, as well as the mythical ambiance of Hollow Knight

3. Having similar feel to the side scrolling sections of the Super Star Wars games from SNES

USP (Unique Selling Point)

- Physics based combat
- Lore
- Funny Animations
- Star Wars inspired platformer

Design Pillars

- 2D physics based Platformer
- Star Wars like aesthetics
- Variety of enemies with different abilities

- Challenging puzzle platforming

Scope

Mechanics – Low

- Basic movement, left, right, jump for platforming
- Press to shoot to kill enemies
- Platform to get in a position to be able to fire at the enemy or other objects
- Force power activator key to activate for power to damage enemy (might not happen)

Story – Low

- The story isn't necessarily going to be told clearly, instead the story will be told through short written explanations through the levels, the story itself is pretty basic (see elevator statement) (might not happen)

Aesthetics – Low

- 2D pixel art style
- Star wars inspired
- Basic 2D Sprites

Technology – Medium

- In terms of technology, I will be using the game engine Unity to develop this game.
- The game will be 2D and have some 2D Physics elements incorporated in it.
- In terms of physics, I am looking at creating ragdolls by making the enemy sprites with parts connected with joints and have them fall down to their death in a satisfying ragdolly way.
- Another element of physics that I want to have in my game is destructible objects, things such as objects that can get destroyed.
- The last element that I want to have is a force power, force push, which would push enemies backwards and also damage them if they are within range

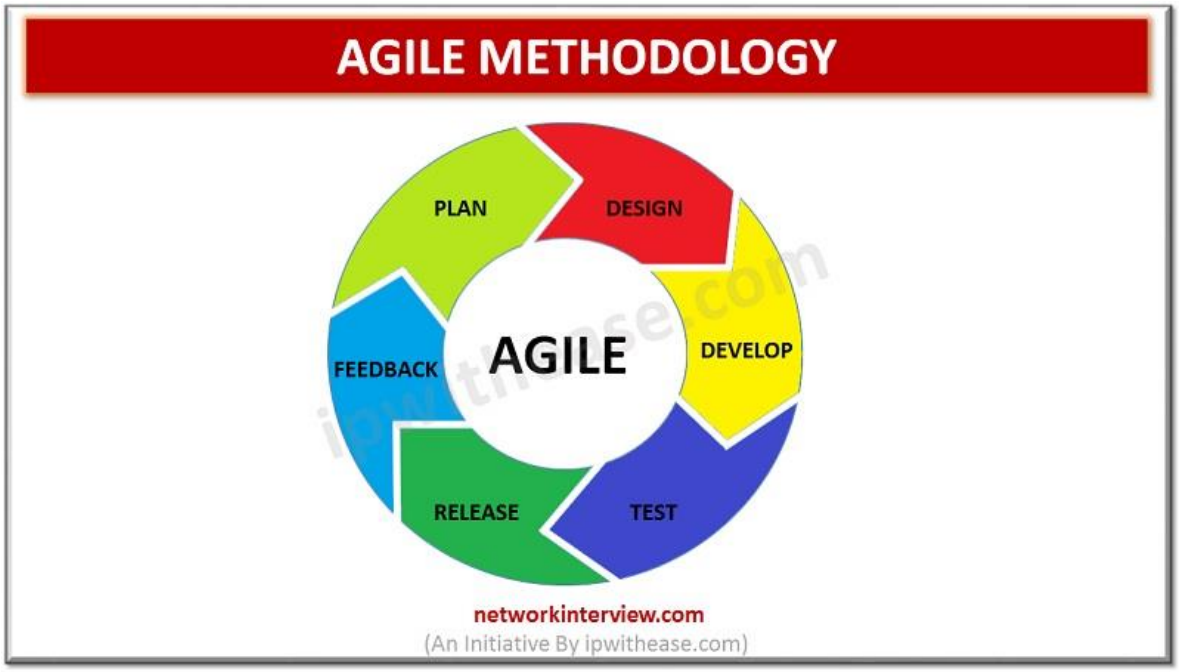
Development Process:

Agile

- Agile development is a great process of developing video games, as it allows for a very flexible and adaptive way to create and design, making sure that you don't get fixated on one aspect of the game too much, and when you hit a roadblock, even stuck at a certain stages of the development process without being able to move on, you are actually able to come back later to the part you were stuck on, and move on to something else.
- The process is great because it "points to a software development methodology consolidated to the idea of repetitive development, where requirements and solutions are developed through fraternizing between self-organizing and cross-functional teams. The utmost value in Agile development is that it allows teams to deliver faster value, with good quality and predictability, and greater ability to respond to change.

It splits the products into small incremental builds. These builds are provided in iterations. Duration of each iteration lasts from about one to three weeks. Each iteration involves working of cross functional teams working simultaneously on various areas.

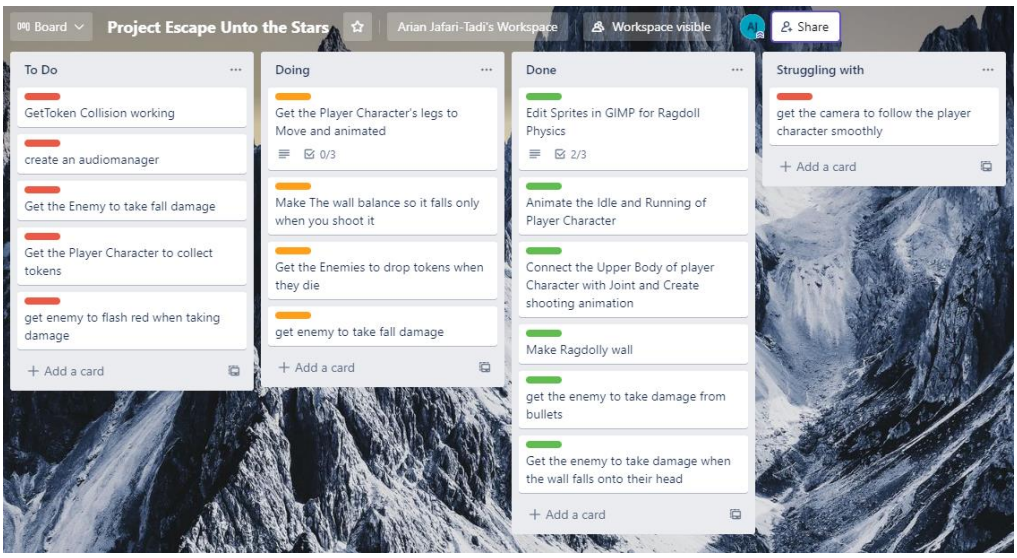
1. Planning
2. Requirements Analysis
3. Design
4. Coding
5. Unit Testing and
6. Acceptance Testing.” [2]



[2]

Tracking:

In the development of this Video Game, Trello has been used to a large extent to establish clear goals and deadlines, and to also keep track of those tasks to make sure that they are not left undone, or make sure that the scope of the project doesn’t extend beyond a point where I won’t be able to develop the game within the given time constraints. Due to Trello’s stellar ease of use, I was able to successful track my tasks for the development of this game.



MVP

- **Player can move and face different directions**
- **Player can jump**
- **Player animations for moving and jumping look smooth**
- **Wall ragdoll and all animations work and look good**
- **Limited platforming level**

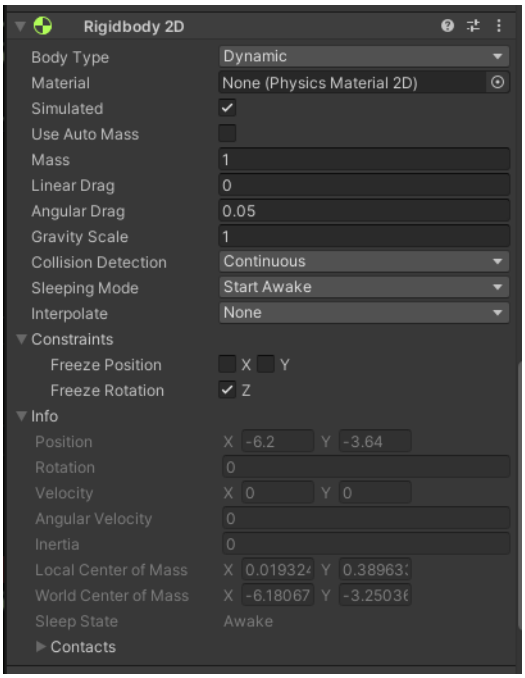
Art/Assets

The assets for this game are going to be basic 2D sprites. The art style in general, will be inspired by Star Wars and will have mystical air to it. The assets are going to be procured from open licences as well as copyrighted materials, the author of the copyrighted materials will be given credit in those cases.

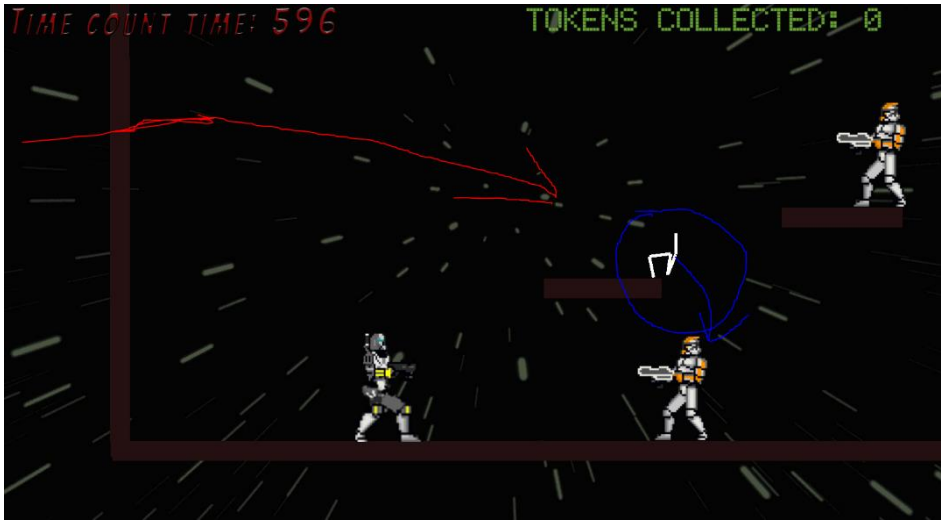
Physics

Part of the task for creating the Video Game for this module was that we were asked to include physics-based gameplay elements, so given that I had to do all of this in 2D, it was an interesting process. I have, in my opinion, introduced quiet a few enjoyable physics interactions into my game, as I will outline below (not all physics based elements that are part of my game will be highlighted in this document).

Rigidbody 2D



One of the most important elements of physics in Unity, when it comes to 2D is Rigidbody 2D, because the “Rigidbody2D class essentially provides the same functionality in 2D that the Rigidbody class provides in 3D. Adding a Rigidbody2D component to a sprite puts it under the control of the physics engine. By itself, this means that the sprite will be affected by gravity and can be controlled from scripts using forces. By adding the appropriate collider component, the sprite will also respond to collisions with other sprites. This behaviour comes entirely from Unity's physics system; very little code is required to get impressive and authentic physical behaviour and allows for ‘emergent’ gameplay that was not explicitly coded into the game.” [3]



Ragdoll Physics 2D

Because my game was a 2D game, I had to approach Ragdoll from a different way, you couldn't just use the Ragdoll Wizard from Unity 3D, as this was a 2D game, so things had to be done 'more manually', but still taking advantage of Unity's excellent built-in features. So, using a combination of Unity's Rigid Body 2D, Colliders (things such as Box Collider 2D, Capsule Collider 2D, and other types of 2D Colliders), Hinge Joints 2D, and many other features, I have been able to design brick walls that when shot, will collapse at the point where the Hinge Joints are and as they fall down, they can damage enemies.

Physics force, double jump, fall damage, collision

In my game, I have been taking advantage of quiet a few built-in Unity 2D features, for example, I have used .AddForce(), which applies a force to rigid body, to propel my Character Player when jumping, as well as using an empty child object placed at the base of the feet of the Character, to check for ground Collision, with this I used an if statement to stop the Character from jumping more than 2 times, and forced the second jump to have less force. The force that is being applied to the Rigidbody2D of my Character "is specified as two separate components in the X and Y directions (there is no Z direction in 2D physics). The object will be accelerated by the force according to the law force = mass x acceleration - the larger the mass, the greater the force required to accelerate to a given speed." [4]

Also by messing around with velocity on the Y axis, I have added fall damage to the enemies, which allows the Player character to push enemy characters off the edge from a high place in order to kill them.

Character Design

When it comes to the Characters that I have used, most of them are Free Art available on the internet, but the characters are heavily inspired by Star Wars, so I will mention that in the referencing section of this document. The characters are mostly basic 2D Sprites created with SpriteSheets and in some cases with Joints 2D as well.

Gameplay

Gameplay goals for the game

- Fighting your way through the level to reach the exit
- Collecting as many tokens as possible from dead enemies
- Experimenting with the physics-based combat (things such as destructible walls being used to crush enemies)
- Platforming

Second to second gameplay:

- Player solves platforming challenges to progress further onto the level
- While solving challenges the player has to constantly deal with a lot of enemies trying to harass him along the way

Minute to minute gameplay:

- Player goes through different sections of the level with different platforming challenges and different mix number of enemies
- Player finds different kinds of collectables throughout the Death Star that give him rewards

Hour to hour gameplay:

- Player goes back through levels to find all collectables not found in previous playthroughs
- Player tries to beat their times from previous playthroughs

Controls

The controls are:

- Left arrow key and right arrow key for movement, pretty typical for this style of game
- F for shooting
- Space for jumping, double Space for double jumping, once again, pretty typical for this style of game

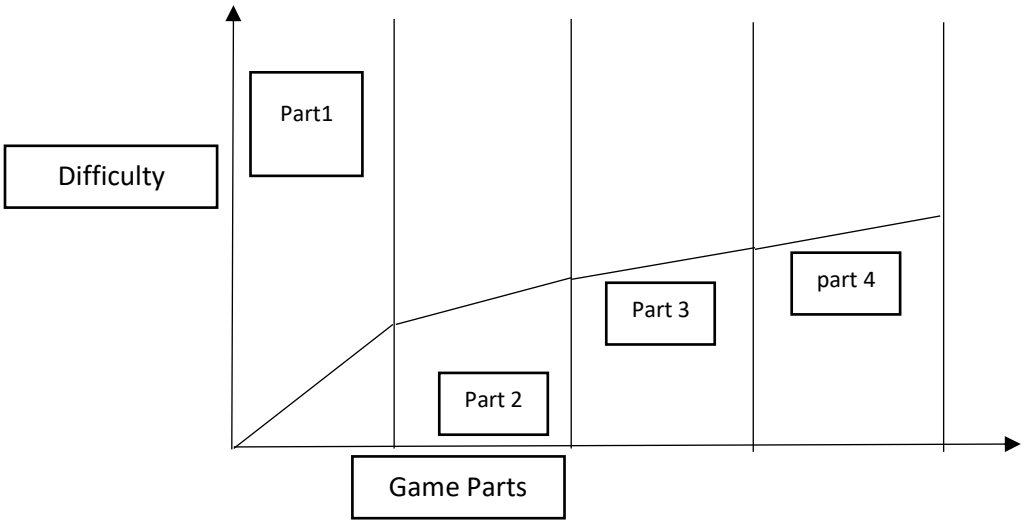
Combat

The combat in my game mainly consists of the player using a gun to shoot blaster shots at enemies and objects. But there are a variety of ways that you could approach this ‘seemingly’ simple combat style.

So, for example, you could shoot directly at the enemy, and keep shooting until they die, or you could shoot at the walls that are located above the enemies, to try to get the wall to collapse onto the enemy, and in the process damaging the enemy, or you could try to push the wall from a higher place so it falls on top of the enemy, then jump on the collapsing wall to make sure that in its collapse it damages the enemy, or you could try to push the enemy off the ledge from a higher altitude place to try to kill him, so there are quiet a few ways to approach combat in the game.

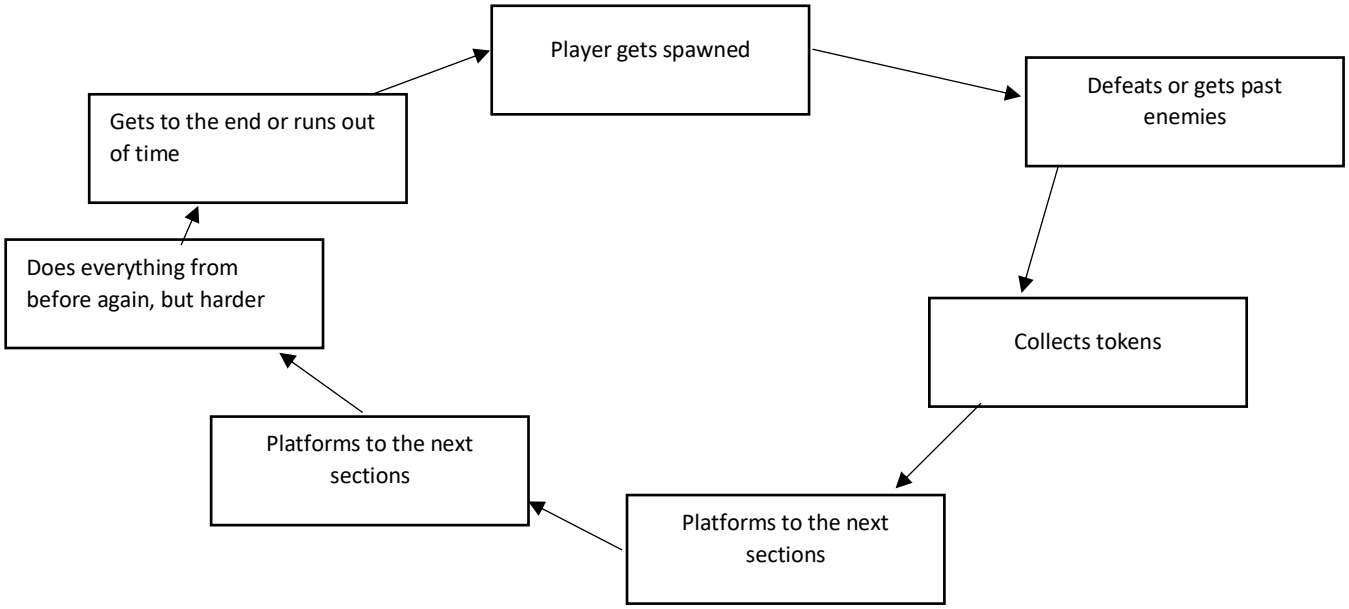
Difficulty curve

The game progressively gets harder as you progress into other parts of the game, but it doesn't become that much harder than at the beginning, it's more that you have more stuff to deal with, like for example, more walls, more enemies, more platforms to parkour.



Core Gameplay Loop

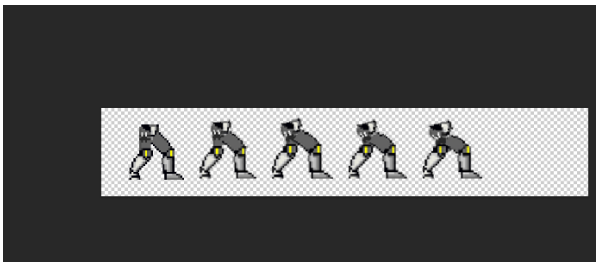
The loop of the game consists of player gets spawned, defeats enemy or gets past it, collects tokens, platforms to the next session, gets to the end or runs out of time. This is an illustration of that:



Prototypes

1. This is one of my earliest prototypes that is using the legs sprite sheet that I created, here I was trying the get collision working as well as get the idle animation working

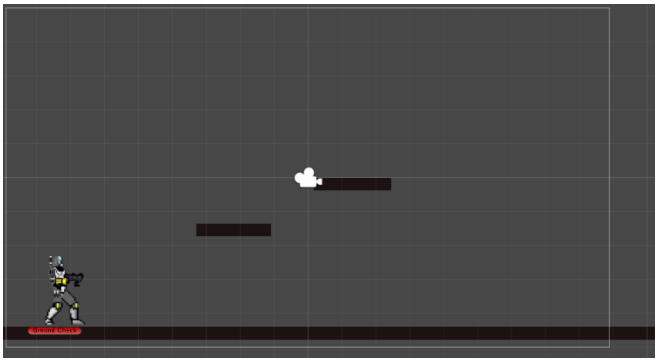
This is me creating the sprite sheet in GIMP



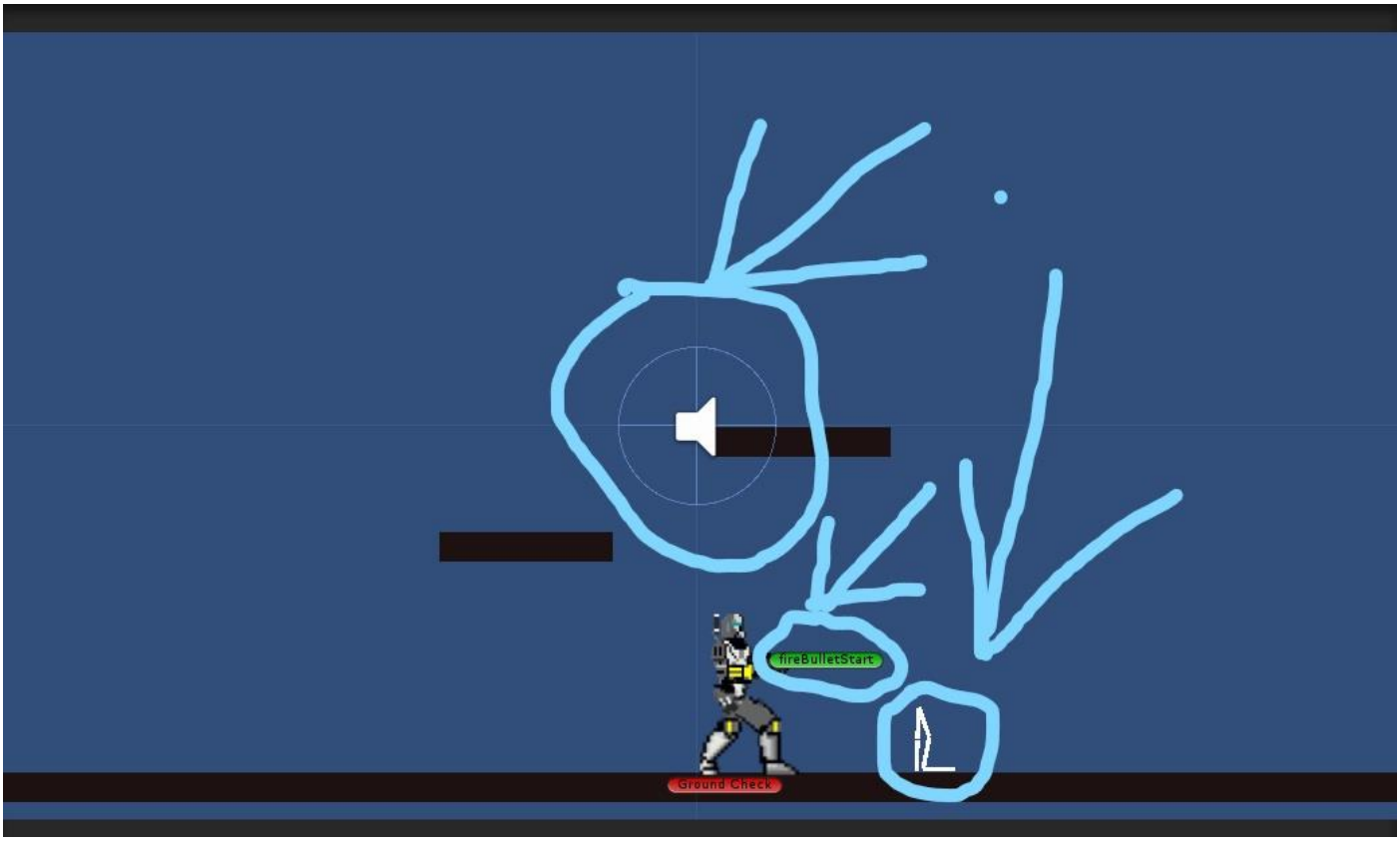
2. This is a prototype where I was trying to get the running animation for the legs to work from the Running Sprite sheet that I create using GIMP

Here is me trying to create the Sprite Sheets using GIMP:

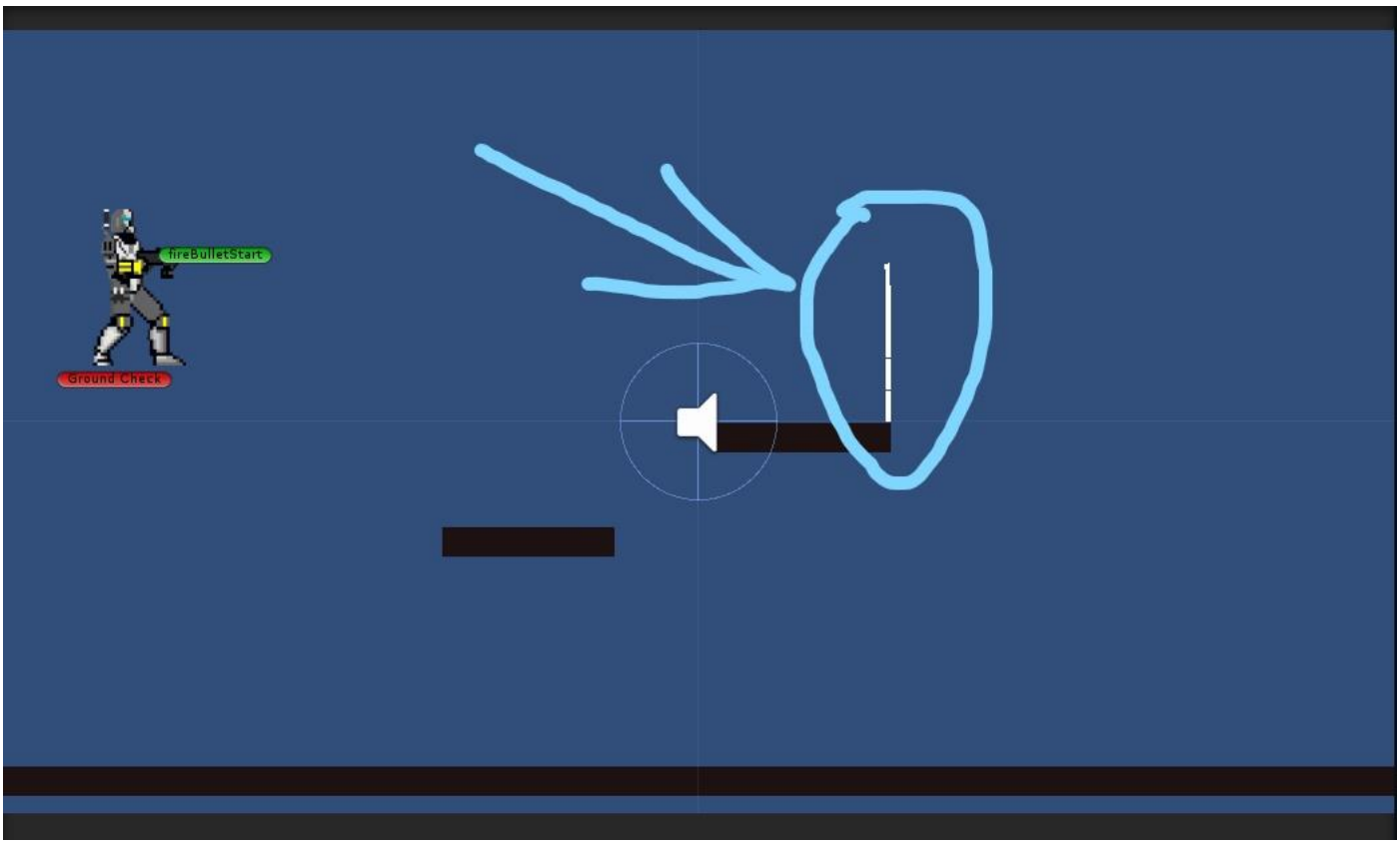
3. This is a prototype where I was trying to get the Ground Check to work, to check if ground collides with characters, so I can prevent the player character from doing more than 2 jumps continuously



4. Here, it is a prototype where I already got my Audio Manager, the firing (using an empty object to determine the position from where it will start firing), and Ragdoll wall working



5. In this prototype, I managed to get the Wall to stand balance itself to stay still, until I shoot and then it start the disintegrating in a Ragdolly way

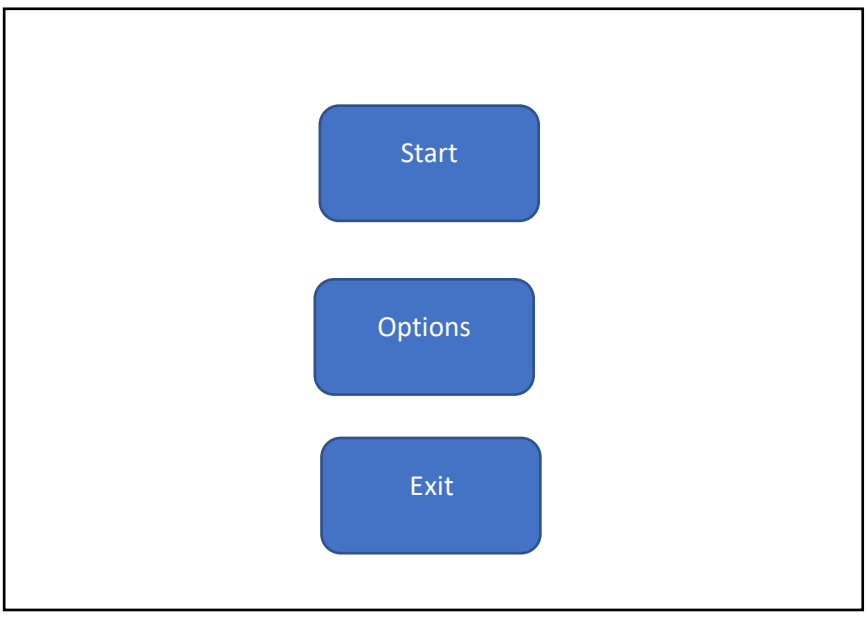


Design problems

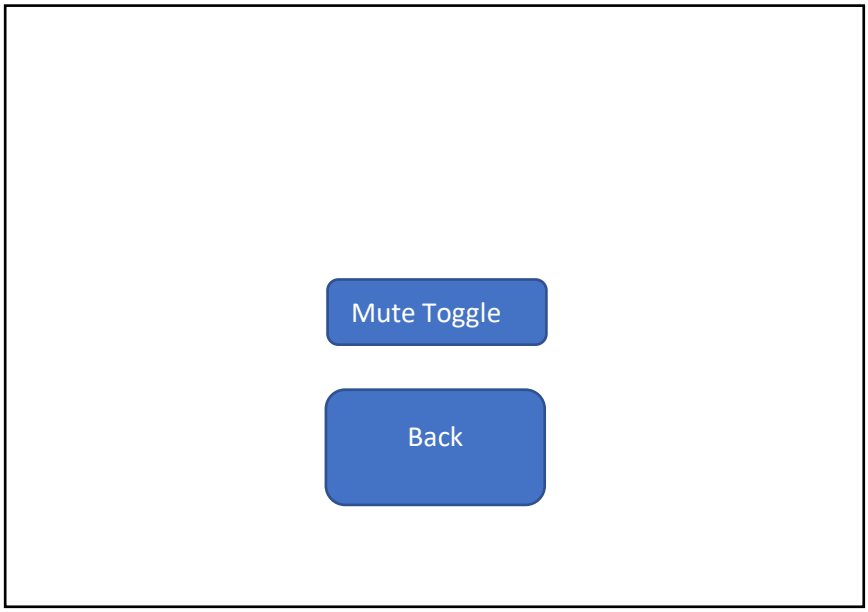
My game, Escape Terror onto Stars, was a quiet different game at the beginning of development, I was aiming for all characters, including the player character to be created out of individual pieces connected by hinge joints, and for all characters when they die, to fall in a ragdolly way. But, I realized the complexity of getting something like this done within the time constraints that I had, also a game centred around that wouldn't be that fun. So changing ideas and design concepts as I went along, due to the use of the Agile Development, has helped me quiet a bit in determining what my game should eventually be. Once I settled on what I wanted to do, I had more focused vision, that wasn't as expansive, but ultimately, it was better.

Wireframes

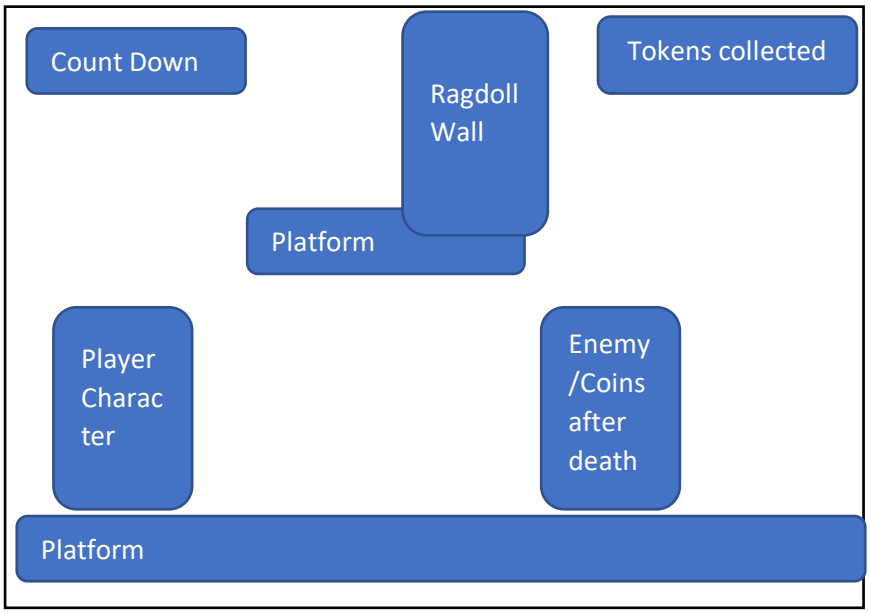
Main Menu:



Settings Menu



Game

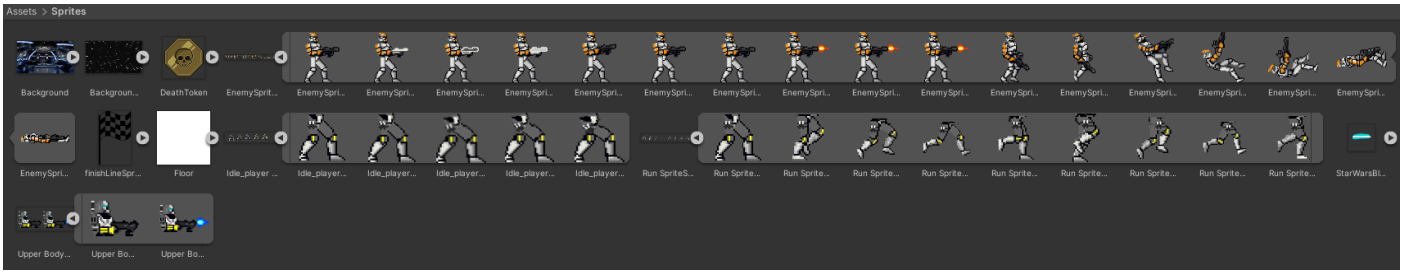


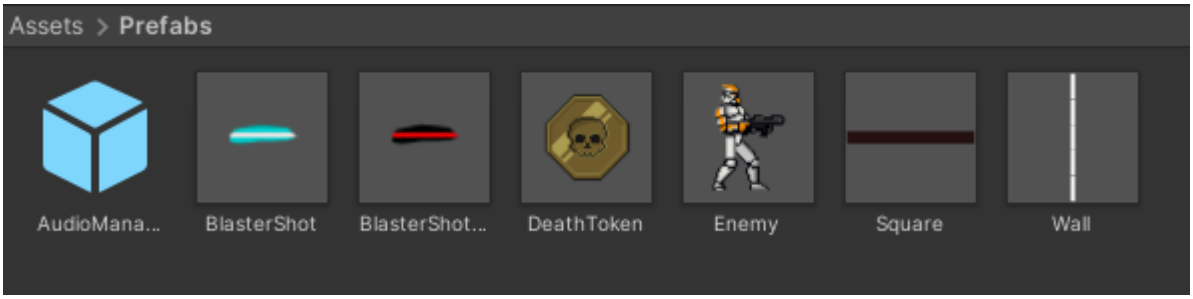
Asset List

Sprites:

- Background (this is the background for the Main Menu)
- Background Game (This is the background for the Game Scene)
- Death Token (this is the token that spawns when the enemy is killed, they can be picked up by the Player Character) (also a prefab)
- Enemy Sprite Sheet (this is the Sprite Sheet of the Enemy, also a prefab)
- Finish Line Sprite (this is the Finish line flag that triggers the “You Won screen”)
- Floor (ground for player and enemies) (also a prefab)
- Idle player Sprite Sheet (Legs for Player when Idle)
- Run Sprite Sheet (Legs for Player when Running)

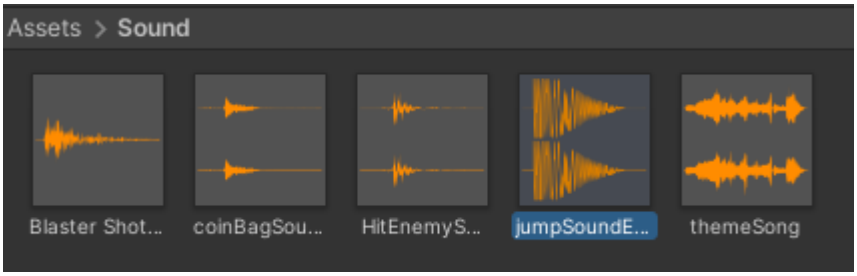
- Star Wars blaster Shot (the bullet of the Player Character)
- Upper Body Sprite (the Sprite sheet for the Torso)
- Wall (prefab for destructible wall)
- Square (Floor prefab)
- Audio Manager (prefab) (empty object with script, create by me to manage lots of audio files very easily)





Sound:

- Blaster Shot Sound (sound used for when the player character shoots a blaster shot)
- Coin Bag Sound (sound used for when the player character picks up a token dropped by enemy)
- Hit Enemy Sound (sound used for when the player character shoot a blaster shot that hits enemy)
- Jump Sound Effect Player (sound used for when the player character presses the Space Bar in order to jump)
- Theme Song (sound used for the whole game, as ambience music)



Research

I have done quiet a bit of research into these genre of gaming, from studying the SNES releases of Super Star Wars, specifically the 2D platforming sections, to actually playing 2D platforms, watching YouTube videos analysing the game design of successful games from this genre, as well as asking other average gamers what they would like to see in a simpler version of a 2D Video Game Platformer.

When it comes to the SNES games, I have looked into the gameplay styles they used, as I was interested in seeing what a similar game that is based on Star wars would do, but I didn't end up taking much from it.

When it comes to actually playing games from the genre, I played Super Meat Boy, and Hollow Knight. What I took away from Super Meat boy was the very interesting, and challenging platforming sections that require the player to think in order to find the solution to getting past an obstacle. The things that I took away from Hollow Knight, were

the collectibles, and how it can incentivize the player to take risks and venture where they otherwise wouldn't, I also took away the interesting mechanics that the enemies had, as well the mythical vibe and art style that was on display.

When it come to watching game design analysis of similar 2D platformers, the things that I took away the most was the incentives that keep the player playing the game are important. For example, collectables are a way of making the player think that trying to go places that they otherwise wouldn't want to go to, will give them the opportunities to find more of those collectibles, and then the game rewards the player for going there with a very interesting event, or an interesting place to explore. Another way of keeping the player engage, is enemies having interesting combat abilities with challenging encounters, this will keep the player aware at all times, not losing focus, as he needs to stay on his toes in order to not be killed by the enemies and potentially lose quiet a bit of progress, so it is a balance between rewarding exploration and trying to keep the player constantly aware of the environment as well as danger, and maybe even opportunity.

Target audience

The target audience for this type of game tend to be more of the hardcore gamers, usually mainstream audiences don't tend to play a lot of 2D Side-Scrolling Platformers. More likely than not, the overwhelming bulk of the player base for these game would be 18-34 year old males, that are really into gaming, so still not everyone from that age group. But there would also be a considerable number of female players of 18 to 34 years old, but not as much as the second largest audience which would probably still be males, but older, somewhere between 34-50 years old.

Bibliography:

In terms of assets and art style, the game is inspired by Star Wars which is owned by Disney, but there is not financial benefit to me for creating this game.

Information:

1. networkinterview. (Un). *Introduction to Agile Methodology : Comprehensive Explanation.* Available: https://networkinterview.com/introduction-to-agile-methodology/ . Last accessed 13/05/2022.[2]
2. Unity. (Un). Rigidbody2D.. Available: https://docs.unity3d.com/ScriptReference/Rigidbody2D.html%20%5b3 . Last accessed 13/05/2022. [3]
3. Unity. (Unknown). *Rigidbody2D.AddForce.* Available: https://docs.unity3d.com/ScriptReference/Rigidbody2D.AddForce.html . Last accessed 13/05/2022. [4]

Images:

1. networkinterview. (Un). *Introduction to Agile Methodology : Comprehensive Explanation.* Available: https://networkinterview.com/introduction-to-agile-methodology/ . Last accessed 13/05/2022.[2]

Game:

1. fghow. (2011). Commando Scorch Sprite. Available: https://scratch.mit.edu/projects/1805058/ . Last accessed 13/05/2022.
2. CloneCommando1. (2011). Clone Trooper Sprite Pack. Available: https://scratch.mit.edu/projects/1955863/ . Last accessed 13/05/2022.