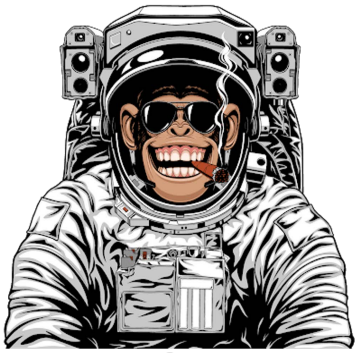
Game Design Document: CS4043

Project Name: Galactic Primates

Team Name: Team Badger

Team Members:



Konan Heany(ID:20242603)/(Workload share ~ 20%) – User Interface Designer

Dennis Kolomiyets(ID:20250762)/(Workload share ~ 20%) -Game Designer/Programmer

Jai(ID:20045247)/(Workload share ~ 20%) – Lead Programmer

Tim Hannon(ID:20276028)/(Workload share ~ 20%) – Project Manager / Animator

Rohan Arya(ID:20086377)/(Workload share ~ 20%) – Audio Engineer

TABLE OF CONTENTS

Introduction/Inital Pitch

Our CS4043 game project straddles the Indie and 2D Space Shooter genres. It was developed using the 3.2.3 Godot engine and is entitled 'Galactic Primates'. The game combines primates,space,and lasers;what could be more exciting? The game pays homage to a host of different video games such as Tomoshiro Nishikado's 1978 classic 'Space Invaders' and playfully parodies film franchises such as 'Star Wars' and 'Alien'. The game features a strong storyline and makes great utilization of the godot engine node system.It was our intention to create a project which was simultaneously challenging, stimulating, and fulfilling for participants. We hope to release the game on the android platform in the coming months. We hope you love our project but beware,In Space nobody can hear you banana!

Design Goals

Our design goals included:

1. Creating a unique game-play experience which would serve as a great introduction to the Godot engine which would also provide an enjoyable and high enough quality experience to release on android devices.

2. Incorporating the key elements of effective gameplay outlined in class into our project. Gameplay is defined as the challenges a player must face in order to arrive at the objective of the game, as well as the actions a player is permitted to take in order to address those challenges. This entails:
    a. **Fairness:**In order to maintain the optimum user engagement and satisfaction, it is essential that fairness is paramount. If the game is too challenging or too easy, players may become similarly disinterested. Game design must conform to established social meta rules concerning fairness. In other words, balance is at the core of what makes a great game, the desire for reward and challenge must both be satiated. Our project "Galactic Primates" is deeply informed by this rational and players will receive rewards after completing each level.
    b. **Asymmetry:** This refers to the different rules and relative victory conditions associated with a particular game. Certain games such as golf have ways to alter the difficulty of a given session through the use of a system of handicaps. Asymmetry is also a feature of the majority of quality video-games. We hope to integrate asymmetry into our project by utilizing a level system whereby the strength,speed,and number of enemy combatants will gradually increase.
    c. **Symmetry:**This refers to the established/constant rules and victory conditions which inform player decisions and expectations. While the game may vary gradually overtime(eg.different level),the games' central objective will remain the same, in the case of our project, this may mean eliminating all enemy combatants and staying alive.

3. **Simplicity:** Our Godot project had to be simple given our limited experience, our plan had to be realistic if we would be able to meet our deadlines.
4. **Player Rewarding:** As well as being a great learning exercise, we hope to create a product which will be enjoyable to play.
5. **Core Mechanic:** It was our intention to centre our project around a single core game mechanic (eg. Be able to move vertically and horizontally and shoot projectiles).Players will be utilizing this core mechanic continually,hence,it must be effective.
6. **Focus on Target Player:** Different games have different target markets,many PS5 games are intensive and time-consuming (i.e Cyberpunk and Ghosts of Tsushima).Our project by contrast will have simple mechanics and graphics; and be aimed at casual players of all ages and genders with limited time.

Influences & Sources

Our project has had a wide array of different influences. Foremost amongst them is the 1978 classic 'Space Invaders' created by Tomoshiro Nishikado. Because microcomputers in Japan were not powerful enough at the time to perform the complex tasks involved in designing and programming Space Invaders, Nishikado had to design his own custom hardware and development tools for the game. He created the arcade board using the latest microprocessors from the United States. Adjusted for inflation, the many versions of the game are estimated to have grossed over $13 billion in total revenue as of 2016, making it the highest-grossing video game of all time. An urban legend states that Space Invaders' popularity led to a shortage of 100-yen coins in Japan. 'Space Invaders' is widely considered the quintessential arcade game; even by many non-game enthusiasts.

Other key influences include:

- "Freedom Finger": A far more recent title in the Space Shooter genre, it's comedic aspects and epic soundtrack featuring the rapper Aesop Rock make it a very entertaining product. (https://www.youtube.com/watch?v=hldldMBFIYQ)
- "Everspace": Is an amazing title which uses 3D-graphics and superb mechanics to push the genre to it's limits. (https://www.youtube.com/watch?v=ItPzK4dzVKU)
- 'Donkey Kong'(1981), originally created by Shigeru Miyamoto (https://www.youtube.com/watch?v=KJkcNP4VkiM)
- The Star Wars/Alien film franchises are also a key influence, regularly parodied.

Target Market

Our ideal user has the following characteristics:

1. Male;
2. 13 to 16 years old;
3. Has access to android device;
4. Likes games to begin with;
5. Likes comics/Sci-fi films(eg.Star Wars);
6. Plays mobile games;
7. Consumes many types of content on mobile;
8. Purchases content through apps, or online;

Although young males are the primary targeted user, the game at its fundamental level has a more universal appeal. It may also have a nostalgic quality and may be appreciated by those who played many of the Arcade games (in 1970/80s) which have inspired our project.

Functional Specifications

Game Mechanics

Core Game Play

The PC (Playable Character) moves in a 2D (two dimensional) plane, exploring the depths of space and the 'Galactic Primates' universe that extends from the left to the right of the screen. Enemies(Gorilla Minions) spawn immediately at the beginning of each level, and the players must defeat them to move along. The player may also encounter obstacles(eg.meteorites) which cannot be destroyed by attacks but can be temporarily pushed away. The game was designed with the intention that it be challenging as well as entertaining and falls neatly under the Bullet Hell/Space Shooter genre.
As the PC advances through the levels, specific actions and events will trigger achievements that are rewarded with 'Galactic Primates' virtual trophies, which can be retrieved at any time through the achievements menu screen. The PC's actions will also be rewarded with experience points which will allow it to "level up and receive stats points, which can then be applied to increase the three different stats that govern the PC's capabilities(Strenght,Speed,Special Attack).
Gameplay related dialogue scenes take place and are displayed for the player during gameplay through the HUD, and cut scenes are displayed between levels/stages, advancing an overarching storyline that corresponds to the events in the 'Galactic Primates' graphic novel. The graphic novel will be made available for free digitally Online (this will help promote the future franchise).
At the end of each level,players will face a mid-level boss(a close confident of the Dark Lord 'Gorillatron') and must overcome the Dark Lord 'Gorillatron' himself in the final level and throw off the Gorilla yoke once and for all.

Game Flow

Actions that the PC can perform are:

1. Up Arrow (Forward Thrust)
2. Left and Right arrows (Turn Left and Right)
3. Spacebar (Attack once)
4. Perform a special attack move with z key(similar to a sub-machine gun), the icon goes on cool down and starts counting down when the player uses the ability .

Characters / Units

1. ***Coconut Queen:*** A distinguished tribal chieftain from the rainforest's of the chimpanzee home-planet Maboo. She enjoys using a coconut rocket launcher and banana staff. Stats should be very high health,average strength, and low special,and she should have average speed.

2. ***Captain Banana:*** Once a simple and content orangutan named Kevin,genetic experiments transformed Kevin into an ace fighter pilot and greatest hope of the primate republic. Stats should be very high strength,high health , and average special, speed stats are high.

3. ***Paddy Higgins:*** Just an ordinary Irish farmer from Co.Cavan. Paddy was walking his land one day and was abducted by a gorilla spacecraft which wanted to experiment on him. Although reused by the resistance,will this humble homo-sapian survive the conflict and make it home to Cavan? Stats should be very low strength,weak health,and average special, speed stats should be very low too.

4. ***Gorillatron:*** Boss and Commander of the gorilla of horde. If the player encounters him more than once, his stats should increment with each appearance. He's a very large character who will have at least three different attacks of varying speed and power, which should require skill for the PC to evade.

Game Play Elements

These are elements present in the game level that the PC can interact with:

- ***Health recovery item:*** Items that recover the player's health bar. Health items improve the quality of gameplay by stimulating player attention constantly. Health recovery items also prolong the possibility of remaining alive and the chances of winning even with low health. This encourages players to continue playing even when their prospects may at times not be great.

- ***Power recovery item:*** Items that fills the player's power bar. Power recovery items that may increase strength, speed,and special attack contribute to gameplay by helping players in difficult situations and making things more interesting and less repetitive.

- **O*bstacles:*** Obstacles that hinder the player's progress and must be attacked until destroyed.(eg.meteorites/black holes).Obstacles provide a structure to another blank arena;in this respect they contribute immensely to gameplay.

Additionally we have elements that communicate the PC's game state:

- ***Health bar:*** Bar that represents the PC's health, which remains the same size but increases/decreases differently depending on the health stats of the character. Once all of their health is the depleted the PC dies.

- ***Power bar:*** Bar that represents the PC's special power move. Once the bar is completely filled that character's special move is available for the player to use.

- ***Dialog scenes:*** These will be small panels presenting dialogue in the form of comic book panels with text that will be displayed for the player on the HUD during gameplay. The player will receive relevant information through these dialogues.

There also exist elements that determine the PC's stats:

- ***Experience Points:*** Points that the character earns every time he defeats an enemy and/or performs some special task. Once the character has acquired enough experience points the character "levels up".

- ***Level:*** This shows the progress that a character has made, as he gains a level once he acquires enough experience points, and for every level that the character gains, the player receives a certain amount of Stat Points that can be used to improve the character's attributes. Maximum player level is 50.

- ***Stat Points:*** Points that the player earns every time he gains a level that can be used to increase the character's attributes.

- ***Attack attribute:*** This attribute determines the damage caused by the PC's attack. Each character starts with a set number, but this number can be improved as the character gains levels, by using the Stat Points.

- ***Health attribute:*** This attribute governs the number of health units the character has in its' health progress bar. Each of the characters starts with a set number in this attribute, but this number can be improved as the character gains levels, by using the Stat Points.

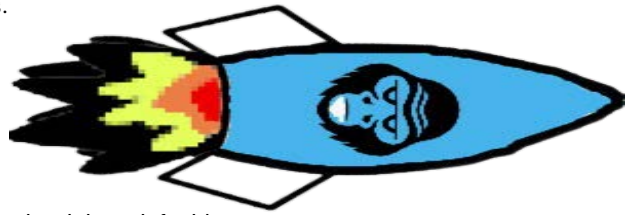
- ***Power attribute:*** This attribute governs the damage caused by the PC's special power. Each character starts with a set number, but this number can be improved as the character gains levels, by using the Stat Points.

- ***Round Number:*** Game progression and difficulty is represented through a round number which is displayed at the beginning of each new round.

Game Physics and Statistics

Physics in the game work as such:

- PCs move along a 2D static arena , either from left to right, or up and down(with aid of the WASD buttons).
- PCs can cross paths with enemy characters unless it is hit by an attack, which destroys both parties.
- Gorilla Minions(servants of Gorillatron) are obliterated when struck, these mindless zombie-like cretins hurl themselves relentlessly like Kamikaze pilots at the PC.
- Boss characters(eg.Gorillatron) should be mostly impervious to attacks, except when they present a weakness and provide an opportunity for the PC.
- Asteroids can collide and slide against each other, and are the only things in the game which cannot be destroyed. There are always exactly five asteroids in the game. Asteroids can be repelled temporarily with blasts from the spaceship.
- We have done our best to emulate the effects of an outer-space/low gravity environment with the design of Captain Bananas' spaceships' mechanistic movements.



Artificial Intelligence

Regular enemies:

- Enemies spawn(Gorilla Minions) on screen either from the right or left side.
- Some enemies will be located in specific positions in a level(eg. the boss at the end of a level will remain at the centre of the screen and only make minimal movements).
- Enemies may pursue the PC on sight or wait until it is within a certain range.
- Enemies attack the PC as soon as they are in range. The frequency of their attacks will vary, but they will be able to attack repeatedly.

Mid-level boss:

- The mid-level boss will have a similar behavior to regular enemies, but will have better range, and more powerful attacks, and may also have a special attack.(eg.Servant Spider Monkey, a close confidant of Gorillatron).Players will face mid-level bosses at the end of each level.

Boss:

- The final boss(Gorillatron) is faced at the end of the game and has a variety of special attack moves with a lot of range, and that have varying speeds and strength.
- The Boss attacks with his long-range special power whenever the player is away from him.
- When the player comes closer to him the boss charges and attacks the player with his melee attack.
- The time the boss is most vulnerable is while he is using his special power.(Dark Coconut Blast)

Multiplayer

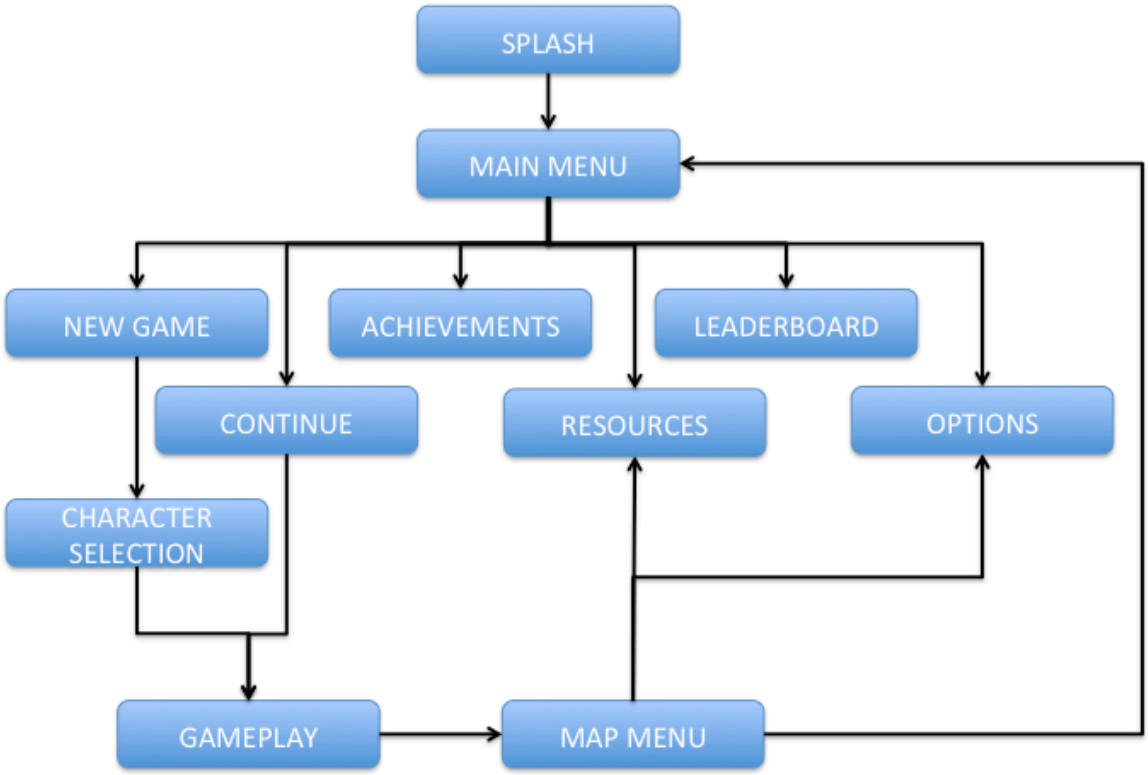
The game includes a leader-board to display the best results obtained by the player and compare them to others. The leaderboard will be accessible in the GUI/homescreen.

If we were to eventually incorporate a multiplayer versus battle system,it could work like this:

1. Choose the character you want to battle with.
2. Choose an opponent from the list of connected opponents.
3. The game goes to the versus battle screen that looks much like other fighting games such as Street Fighter. The two characters are located on opposite sides of the screens in a background related to the game, with health bars and special bars on top of the screens.(Eg, Paddy Higgins Vs Captain Banana
4. On the bottom of the screen you have a bar of available attack motions, they tell you what kind of attack the player can perform and what finger motion they must do on the android device in order to use it.
5. Once one player depletes the others' health bar he is the winner.
6. Every time the player battles in a multi-player battle his multi-player stats change, for battles fought and battles won. This could eventually lead to a vibrant Online culture.

User Interface

Flowchart



Created by Konan Heany with the aid of Draw.io

Functional Requirements

Splash: This is the initial screen and introduces the 'Galactic Primates' logo.

Main Menu: Presents all of the possible choices for the user as well as a captivating graphic image that introduces the look and feel of the game and of the series.

New Game / Character Selection: Shows the character selection screen where the user can choose the character he wants to play with as well as brief information on the character's bio and stats.

Continue: Displays all of the user's games with each character and gives the option to continue from where he last left off.

Achievements: Gallery where the user can check all of the prizes he has achieved as well as the ones he can still unlock. This section will first present a gallery of thumbnail images, and you can see a larger version of the image by interacting with the thumbnail.

Resources: Page in which the user can check further information on different characters, this information is unlocked during gameplay.

Leaderboard: Ranks players in order according to total points acquired.

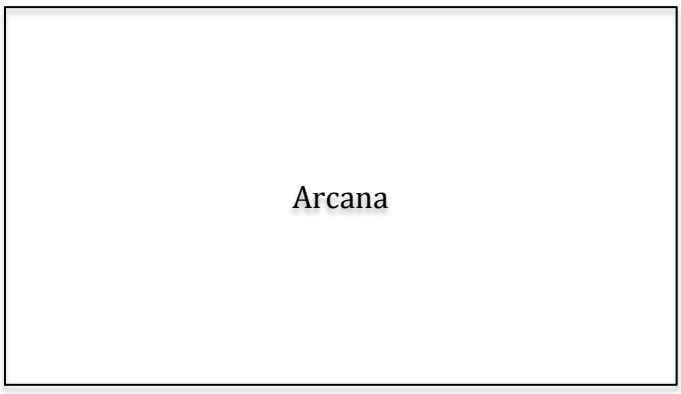
Options: Gives the user the ability to control some of the game's attributes, such as sound volume and music volume.

Map Menu: Can be accessed during gameplay and works as a sort of pause menu, from this page the user can also see which mission he is currently in or he can quit the game.

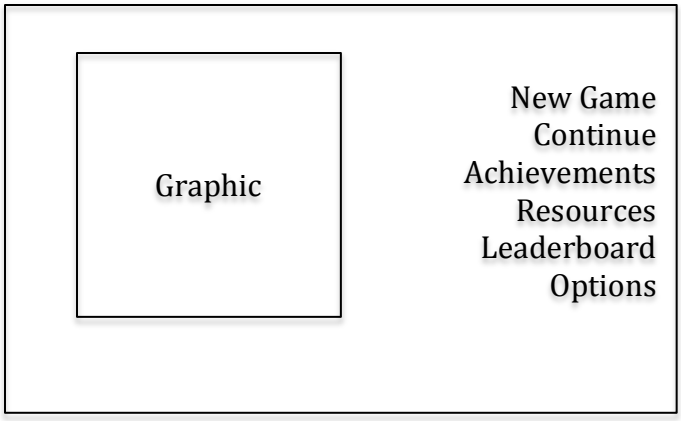
Gameplay: Where the game happens, has a fixed Heads Up Display that contains a character portrait, the total points, a health and a power bar as well as a level display.

Mockups

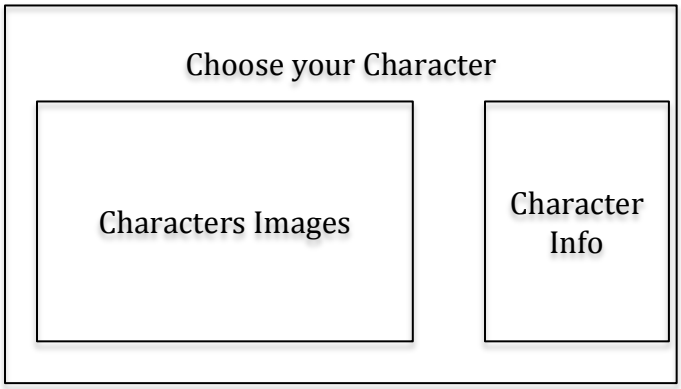
1. Splash Page



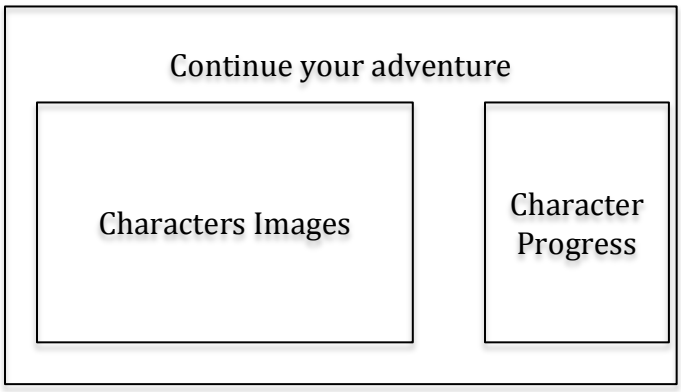
2. Main Menu



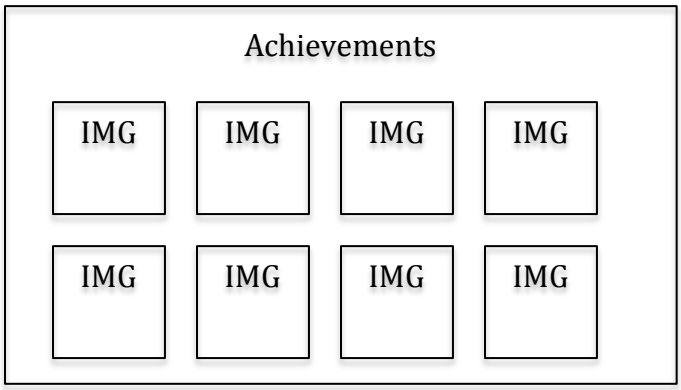
3. Character Selection



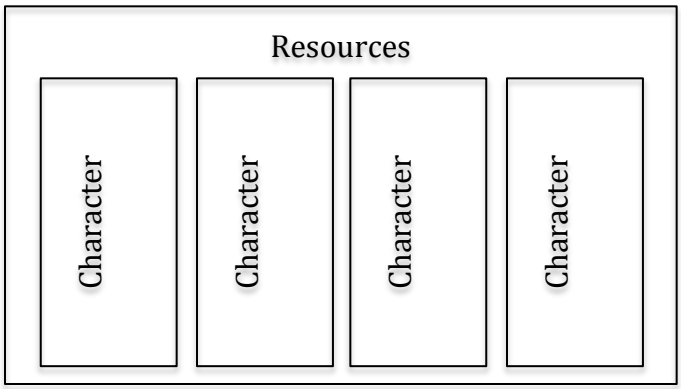
4. Continue



5. Achievements

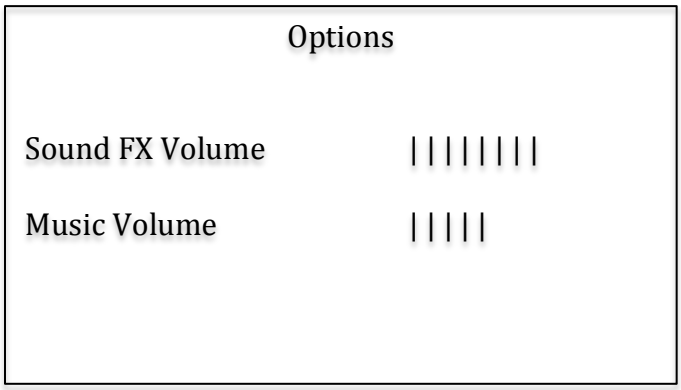


6. Resources

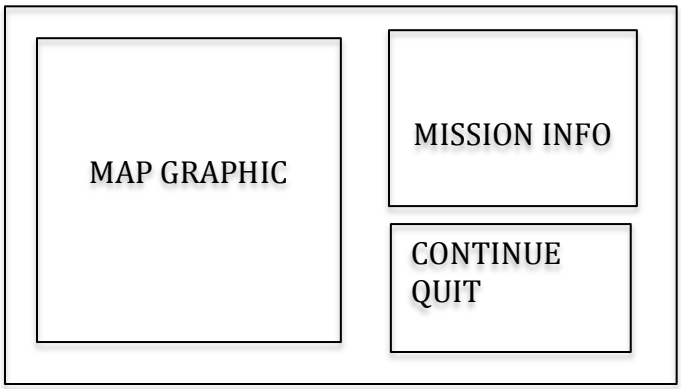


7. Leaderboard

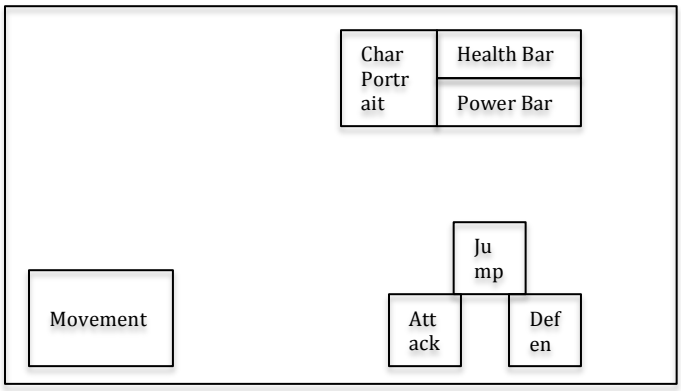
8. Options



9. Map Menu



10. Gameplay



This GUI Mockup was created by Tim Hannon with the aid of Microsoft Word

GUI Objects

The GUI of the game is divided in a few sections. First we have the HUD (Heads Up Display). The HUD/ health bar can be moved to the left/right.This is the information that appears on screen when the user is actually playing the game, the HUD contains:

HUD Image

- Character image – displays the character currently in use.
- Health bar – represents the total health the character has and the current amount of health.
- Power bar – shows the current level of the power bar,
- Special bar – it is a special button that becomes available only when the bar is completely filled, the user interacts by tapping on it in order to use the special power.
- JOYSTICK 1/2 – The user taps the arrows to move the character in that direction.
- Attack – It is an attack button and the user must tap it.
- Defend – The user taps this button to make the character defend.

Art and Video

Overall Goals

From the onset we have sought to achieve an art style that resembles a comic book.
Our art style has a slightly dark character yet maintains a cartoony , sci-fi, and humorous mood.
We created our own art assets such as the Captain Banann's star-fighter and Gorilla Minion craft to complement our projects' unique sensibility.

Current main menu, art work was obtained from the open gamers art repository and the manipulated by Tim Hannon with the aid of Adobe Photoshop. The font displayed is 'Broadway' and was chosen because of it's space-age vibe.

Main menu image(Still in development), art work obtained from open gamers art repository (https://opengameart.org/)

2D Art & Animation

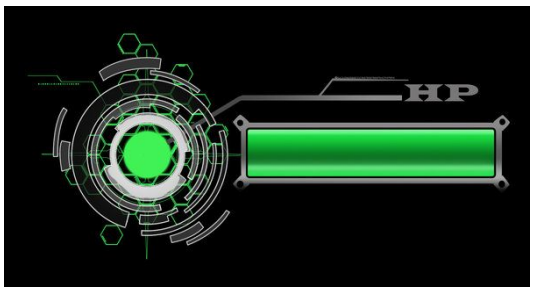
GUI 2D Art

Achievement Unlocked Alert

Level Up Alert

Space age text is consistent with the games overall theme/design

Art work obtained from open gamers art repository(https://opengameart.org/)



Character Health and Power bar.

- Character portrait;
- Health bar;
- Power bar;
- Points system;
- Level display;
- Enemy health display;
- Boss health display;
- Power attack button;
- Map menu button;
- Level Up alert
- Achievement Unlocked alert

NOTE: The Health/Power Bar is still in development but will be included when we release the game on android.

Marketing and Packaging Art

Splash Image

- Splash screen;
- Android icon.

Terrain/background

LEVEL 1 Background - 'The Monkey Way'

LEVEL 8 Background - 'Maboo Skies'

Sound and Music

Overall Goals

The goal of the music within the game is to set establish the desired mood of the world as well as indicate a call to action for the player during gameplay events. In the case of our project, the intention was to reinforce our projects' sci-fi and classic arcade aspects. Two clearly different music soundtracks were created by the teams' sound engineer Rohan Arya. Rohan has dabbled with audio engineering for many years and we decided that our own independently produced soundtrack would avoid copyright concerns when we release the game on android devices in the coming months. Rohan has created a basic gameplay music loop that's not too fast, that is very minimal and subtly layered with synthesizer textures and digital samples (Monkey Limbo, Gorilla Quake); and a 'Banana Blaze' song loop that is considerably faster, and could has an industrial rock feel. The music and has been normalized, mixed, and compressed as MP3 with a sampling rate that we should aspire to be 192kbps, and should be no lower than 128kbps, and consists of loops that can be cycled repeatedly.

Sound FX

The GUI interactions are indicated with sounds for Menu button, Stats button, Reward button, and Play Game button. Our game uses special effects to indicate powerful actions(special attacks) and to establish the game's ambiance, such as distant explosions and various animal noises(eg.a shouting gorilla). The sound FX within the game must enhance the game-play and clearly identify each character. The sound FX must contribute to the essential aspects of effective game-play outlined in class and be conducive to an enjoyable experience for players. Even though the actions that PCs can perform are the same, their special moves are different, and require different sounds.

What takes place in the game is indicated with sound effects, such as the appearance of new characters. For instance, a jungle beat indicates the presence of Captain Banana; while the presence of Paddy Higgins is presaged by a sounds of various animals.

When a PC recovers health it is also indicated with a sound effect. Sound FXs also indicate the characters' attacks, when they receive damage, and when they die. There are sounds for when a character picks up an item, receives an achievement, finds a hidden area, or rescues a hostage.

The basic enemy characters (Gorilla Minions) are able to cycle randomly through a few grunt sounds that will be heard when they are struck the first time in a combo, and have a particular sound to indicate when they die. They also have a basic grunt sound for when they are not engaged in combat, and another grunt for when launching an attack. Vibrant sound effects ensure that the game is totally emmersive.

Gorillatron has very deep and reverberating sounds to accompany his attacks. The character is also identified with distinctive evil grunts and such that sound muffled since the character is inside of armor,as well as a beastly howl for when the character loses a battle. We also have a number of different sound effects, such as explosions and fire that are used in conjunction with other sound effects in the final boss battle scenario.

NOTE: As well as developing the majority of our visual assets ourself, we also went to the effort to create our own soundtrack. Rohan had experience dabbling with audio engineering and is an accomplished musician;he created the games' unique track so copyright issues would not be a concern.

Story Synopsis

In a galaxy far far away but not unlike our own. Primates of different varieties contend for domination of their system. Once allies,lust for power has driven the gorillas to betray their chimpanzee allies and declare war. As a seasoned chimpanzee Space Force ace; will you be able to overcome the Gorilla hordes and protect your home-planet Maboo? But be beware, In Space nobody can hear you banana!

Player Characters

1.Coconut Queen: A distinguished tribal chieftain from the rainforest's of the chimpanzee home-planet Maboo. She enjoys using a coconut rocket launcher and banana staff. Stats should be very high health,average strength, and low special,and she should have average speed.

2.Captain Banana: Once a simple and content orangutan named Kevin,genetic experiments transformed Kevin into an ace fighter pilot and greatest hope of the primate republic. Stats should be very high strength,high health , and average special, speed stats are high.

3.Paddy Higgins: Just an ordinary Irish farmer from Co.Cavan. Paddy was walking his land one day and was abducted by a gorilla spacecraft which wanted to experiment on him. Although rescued by the resistance,will this humble homo-sapian survive the conflict?Stats should be very low strength,weak health,and average special, speed stats should be very low too.

Other potential PCs(Future Development): Dr. Neanderthal, Pr. Bonobo,Elon Monkey .

Enemy Characters

Gorillatron: He is essentially a very powerful ancient gorilla demon who was banished to the universe's outer rim by the primate gods many eons ago. However,he has returned to reign over The Monkey Way and has assembled a horde of gorilla minions, death gods, and possessed beings, with the intent of tearing down the wall between the abyss and reality. Gorillatron has a human like appearance and completely encased in a suit of armor. In his true original demonic form, Gorillatron has spiked wings, horns, and hellfire.

Gorilla Minions: These are possessed warriors of Gorillatron who follow every command from their demonic overlords. They are empty shells of former gorillas, and they crumble to black dust when destroyed. There are minions of different types, with different strengths, who may be recognized by their markings or their colors.

Game Mechanics

Game engine

We have used a game engine to develop our game called Godot 3.2.3. A game engine is a system designed to develop games for various platforms like consoles, computers and hand-held devices like smart-phones. Godot is a 2D and 3D, cross-platform, free and open-source game engine released under the MIT license. Godot aims to offer a fully integrated game development environment. It allows developers to create a game, needing no other tools beyond those used for content creation (visual assets, music, etc.). The engine's architecture is built around the concept of a tree of "nodes". Nodes are organized inside of "scenes", which are reusable, instanceable, inheritable, and nestable groups of nodes. Godot is very simple an is great for amateurs or those just beginning their game development career. Games using Godot can be created with a variety of programming languages including C++, C#, and any other language with GDNative bindings such as Rust, Nim, and D. However, we just used the GD script for our project. It is our in intention to use godot to learn new languages in the future.

Platform and OS

Godot provides the possibility to export the game to various platforms, these includes: iOS, Mac Standalone, Windows Standalone, Web, Nintendo Switch, and Android.

Galactic Primates was developed for the android/ios devices in mind, given that the average mobile gamer very much pertains to our own ideal target market. One could easily envision “Galactic Primates” resonating with this particular target market which generally prefers inexpensive and casual games. The Godot engine has also been widely used by the developers of many of the greatest Indie mobile games. The low technical and financial barriers to entry into the mobile gaming market are also very attractive. There is an amazing 2 billion android users alone!

The Nintendo Switch has a similar target market to android/ios games and apparently has great tools which support external developers. However,the userbase of NS devices is a fraction of the ios/android one. Also,the NS developer tools are expensive(450$).We may release our project on the Nintendo Switch store in the future.

Code Objects

Player Object:
The player object is named after the current character(eg.Captain Banana) being played and can be identified by the “Player” tag. The object has the Player.cs script on it, which is used to act according to the input from the user, this is done by listening to events from the ButtonManage.cs.

Enemy Objects:
The enemy objects come in 3 different variations and is identified by the tag “Enemy”.(eg.Gorilla Minions) Its type is decided by a Boolean value in the Enemy.cs script that is attached to the enemy object. The difference between the 3 types of enemies is the amount of experience they provide the user with after getting killed, the amount of health they have and the amount of damage they deal. This means that enemyOne is the weakest and enemyThree is the strongest. The Enemy.cs script also provides the enemy with an AI.

Boss Object:
This object is identified by the tag “Boss” and is named Gorillatron. The boss has a very simple AI just like the enemy.

Camera Object:
The camera Object is an orthographic camera that exists in every scene of the game. This object has attached 2 scripts to it. The first is called rayCastScript.cs this script manages all the raycasts made when the user is touching the screen. The scripts purpose is to let the ButtonManager.cs script, which is also attached to the camera object, know if the user is pressing a button or not.

Button Objects:
The game has 6 in game button objects, these are: JOYSTICK1, JOYSTICK2, Attack, Special Attack, IngameMenu. All these objects are linked to the buttonManager script on the camera object.

Power Up Objects:
The game has 2 different types of Power Up objects, one recovers the player's lost health, and the other fills up the player's special power bar that allows him to use the special power attack. Both Power Ups have a 25% chance to spawn each time the player kills an enemy. The Power Ups have the PowerUp.cs script attached to them where a Boolean value decides its type.

GameManager:
This object is named GameManage and has the GameManager.cs script attached to it. The Object manages things such as enemy spawning and the points the player is able to spend on stats after leveling up. It handles the enemies by making a list with all the enemies in the current scene. Then the Trigger objects can ask the game manager for an enemy on the list when the trigger needs to spawn one.

Trigger Objects:
There are 3 different kinds of these object, they are StartTrigger, SpawnTrigger and WinTrigger. The start trigger makes sure that the player can't leave the defined level when he is colliding with it; the WinTrigger ends the level when the player collides with it. The spawn trigger spawns an enemy the first time the player collides with it. The trigger will ask the GameManager for an enemy with the stats defined on the specific Trigger the player is colliding with, if the enemy is available it will spawn it.

Core Gameplay Loop

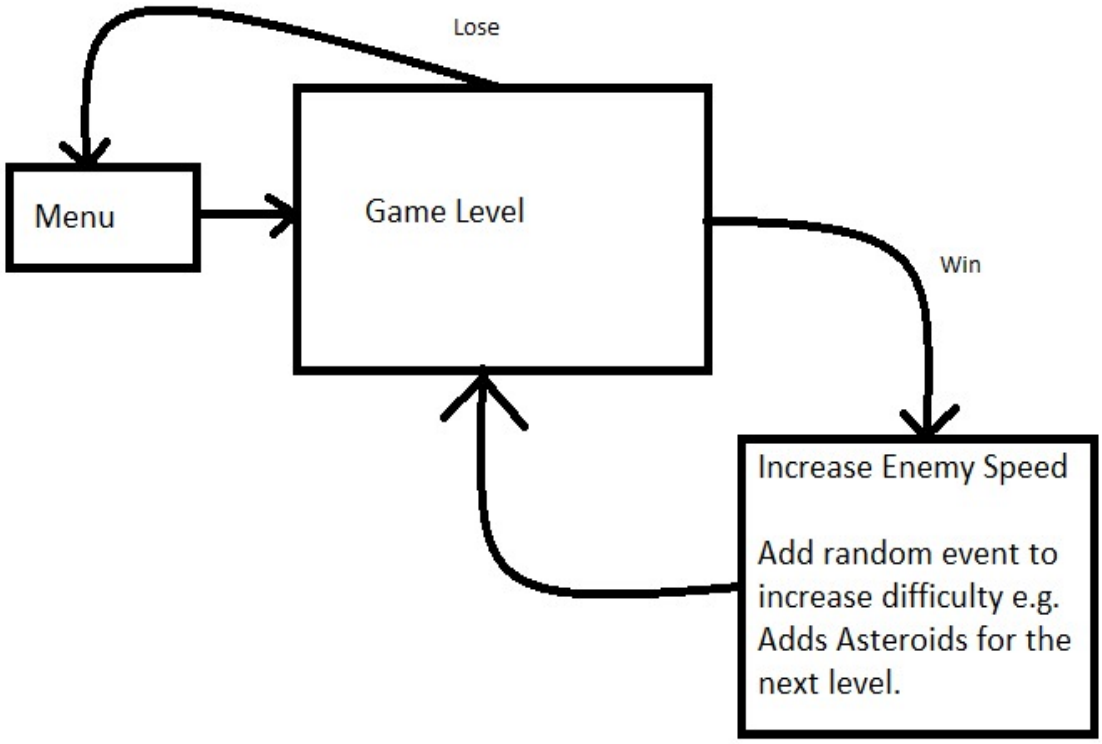
The Core Gamplay Loop in our project is the following gd script and is defined as follows:

```
$MobPath/MobSpawnLocation.offset = randi()
var mob = Mob.instance()
add_child(mob)
var direction = $MobPath/MobSpawnLocation.rotation + PI / 2
mob.position = $MobPath/MobSpawnLocation.position
direction += rand_range(-PI / 4, PI / 4)
mob.rotation = direction
mob.linear_velocity = Vector2(rand_range(mob.min_speed, mob.max_speed), 0)
mob.linear_velocity = mob.linear_velocity.rotated(direction)
```

This core gameplay loop allows for Mob Spawning and Movement. Where both are random but the speed is fixed and declared elsewhere.

```
func _process(delta):
    if Input.is_action_just_pressed("ui_down"):
        var bullet := preload("res://Bullet.tscn").instance()
        bullet.global_position = Global.player.global_position
        bullet.rotation = $Player.bullet_dir
        add_child(bullet)
```

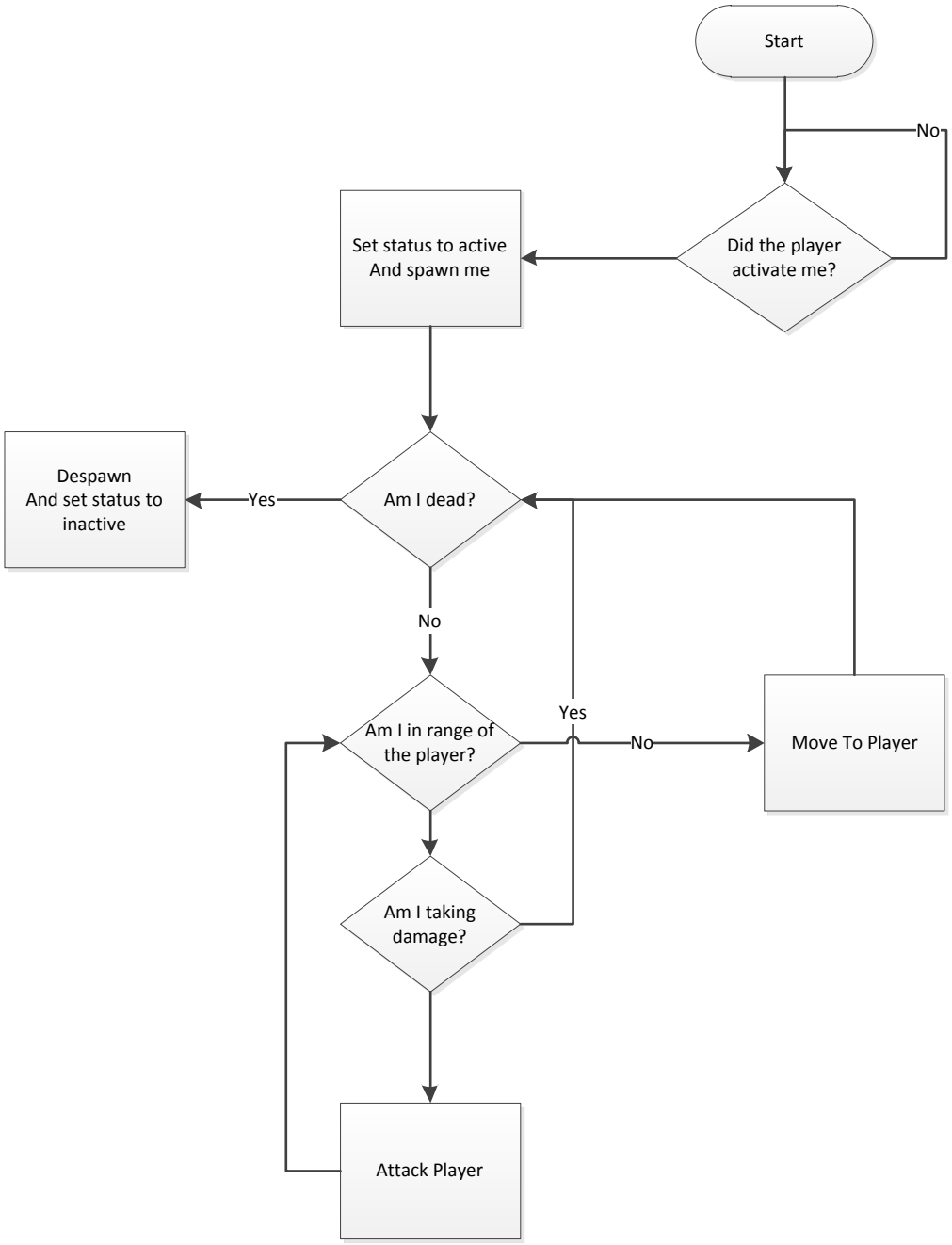
This core gameplay loop enables bullet input and spawn commands. Where it sends on player position and rotation.



This flowchart was created by Dennis Kolomiyets and seeks to describe visually other aspects of the game functionality.The game runs the same recursive functions over and over again for every round until the player loses.

Artificial Intelligence

Below is a flowchart of the enemy AI to explain how it works. If the player runs through a trigger the trigger will ask the game manager for an enemy. If there is an enemy that fulfills the triggers request the enemy will spawn. The enemy will then run through the cycle seen below until it's killed.



Made with the aid of draw.io by Tim Hannon

Production Schedule

Scope

Our project Scope includes the following:

1. Designing and development of an Android Game;
2. Game is to have a Space Shooter/Arcade style of gameplay;
3. Minimum of three playable characters;
4. One type of regular enemy character;
5. One boss type enemy character;
6. Three slightly different background images that can be expanded in several possible game levels(eg,"Maboo Skies/The Monkey Way";
7. Prize giving mechanic with several loaded prizes;

Plan for Hypothetical Developer Team

In the event that another game developer team would wish to construct our project from scratch independently, we have prepared a brief production and post production plan which we believe would be of immense help:

- Before commencing production,familiarize/dabble with the game engine of your choice before you begin your work(eg.Unity/Godot),we found this helped us immensely.
- In order to make this game yourself. You need to create a 2D plain for your character to traverse Character models are designed, rendered, and iterated on to look exactly how they should in the story.
- Your character must then be given movement and the ability to fend himself off. In our case we have a ranged attack.
- Following that, add enemies that spawn in and move randomly that if they hit you, they then send a signal that they've killed you. Add a ranged attack that deletes the enemies if they get hit by it, neutralizing them in the process.
- Then you might want to add a timer that counts towards zero. Once it reaches zero have a new round start that increases the enemies speed stat. Make sure to leave time between different round ending events to allow the player to catch their breath.
- Once the majority of production has ceased and the core game is complete. Now you can add objects to float around the screen as obstacles we used asteroids sprites. Make these indestructible but deadly to both you and the foe.
- Then we add a special ability. Ours was the same basic attack except it fires very quickly(simple from a technical point of view).
- Finally, create a 'Home' and 'Game Over' screen, where the home screen has all the buttons and info needed and the game over screen displays the amount of time you lived for.

Post-production recommendations:

- Many talented developers neglect the post-production phase of game development. However, more often than not; post-production is equally important if a project is to prove successful and achieve significant market penetration.
- The pre-launch stage is a stressful time for game developers. Questions of self-doubt may seep in as you wonder how the public will react to your first functional product. At this stage,critical resources such as human capital may be redirected from game development and put into advertising/promotion or bug-squashing. Developers should concentrate on how people will hear about their project(press-conference/social-media).It is our intention to utilize social media intensively before we release our game on the android app store. A hype video should be created which can be shared easily and which engenders suspense.
- For games with many bugs, developers will create a hierarchy of bugs to squash. This hierarchy will include "game-crashing" bugs near the top and minor bugs near the bottom. In addition to bug squashing, developers will typically polish the game as much as possible before it launches. Maybe Captain Monkey's Banana Star-fighter could have more depth. Perhaps Gorillatron's obsidian armor could be more textured.
- After the game is officially released, the game development team is not completely finished working on the project. It's not uncommon for video games to launch with batches of minor bugs. This is all part of post-launch support. Hypothetical developers following our plan should create mechanisms for the video game community to reach them with their concerns.
- Another part of post-launch is to provide regular software updates for the game. These updates range from game-balancing patches to new downloadable content, or DLCs (eg.'**Galactic Monkeys:The Panda Insurgency**').

Other useful tips for effective game development and recreation of our project:

- Divide up work between all members as equally as possible and allocate work based on peoples' interests where possible(eg. Rohan had dabbled with audio engineering before he created our unique sound-track.
- Set weekly goals to ensure the team completes the project within a reasonable time-frame.
- Have a reasonable idea where you are going to source your game assets from. Will you create your or will you rely exclusively on open -source resources(we used both in our case)?
- Have realistic goals and expectations about what you will be able to accomplish.

Sources/References

- Godot Docs (https://docs.godotengine.org/en/stable/index.html)
- https://opengameart.org/ (Open-source art resource for game developers)
- Pixelorama is a free & open source pixel art (https://orama-interactive.itch.io/pixelorama)
- https://www.playonloop.com/ (An open-source music resource)
- https://pixelprospector.com/ (A great resource for Indie game developers)
- http://dig.ccmixter.org/ (An open-source music resource)