

Milestone III Submission

Team Name:

Octagon

Proposed Level of Achievement:

Gemini

Final Product

For Windows:

https://drive.google.com/file/d/1VcHGPITLExp7Rp4H6gCvSwiG0XS_FC/view?usp=sharing

Troubleshooting for exe:

[Cannot open EXE files - Windows Server | Microsoft Learn](#)

For macOS:

https://drive.google.com/file/d/1_Mxyz71m6DIqiA174vvXOzQBGIbJ0pb2/view?usp=sharing

Troubleshooting for macOS:

Steps:

1. Click the link above and download the zip file.
2. After unzipping and opening the "cc_milestone1MacOS" folder, open the "Campus_Crisis" application.
3. If a warning notification shows up saying "Campus_Crisis" can't be opened because Apple cannot check it for malicious software:
 - a. Open System Preferences
 - b. Go to Security and Privacy
 - c. Under "General", click "Open Anyway"
4. Troubleshooting:
<https://ask.godotengine.org/67755/couldnt-open-after-exporting-mac-says-couldnt-load-data-path>

To play:

Use the WASD keys or arrow keys on the keyboard to move around. Use the spacebar key to stop the player.

To pause, click the top-right button. This gives you the option to either resume or quit.

When you come into contact with another NPC, the tag button appears for you to tag that person.

The ally's persona is displayed on the top of the screen. Use it to locate and tag the ally.

You have a minute to get the ally. The coins are computed based on the speed of catching the ally.

You have to avoid being too close to the lion chasing you.

Pick up weapons along the way to deter the lion temporarily.

Use the coins collected to buy power-ups in the inventory shop.

Problem Statement:

We want to create a game that helps incoming students, local or international, to familiarise themselves and get to know NUS better. We also want to develop a challenging puzzle runner game that celebrates the vast diversity and variety of academic departments offered by NUS, as well as the appreciation of teamwork and collaboration to achieve a noble accomplishment. On a personal level, our team also wants to make this game a personal passion project over the holidays. We believe this project will also provide a learning experience for using industry-standard tools and software while expanding our horizons by giving us a glimpse of what a career in game development might look like. We also want to learn what it's like working on both front-end and back-end aspects in terms of development, and learning how to overcome challenges associated with them.

Motivation:

We were initially inspired by classic platformer games, specifically endless run games such as Temple Run and Subway Surfers. Additionally, we found inspiration from video games such as the Insomniac Spider-Man PS4 game whereby the gameplay places are based on real-life locations. Most importantly, however, our team was motivated by the children's puzzle game "Where's Waldo" in which players look for a target individual within a massive crowd of people.

These inspirations form the motivation and foundation of our ideation for the game we want to develop: "Campus Crisis". Similar to the Spider-Man PS4 game, this will take place in certain places inspired by real-life venues within the campus at NUS, namely the Central Library, Prince George's Park and University Town. We feel that there aren't many games based in NUS, so we thought it would be a fun and captivating challenge to model our game location after NUS. Also, as the game involves running, we wish to somehow incorporate aspects from the endless run games into our game, though over time we have decided to make a mix of a platformer and RPG game

Students who missed the orientation camps before the start of the semester, along with international students and those who received late offers to NUS, may find it difficult to navigate the campus and find their way around once the semester begins. Without the opportunity to participate in these camps, they may encounter challenges in familiarising themselves with the campus and its surroundings once the semester begins. As such, "Campus Crisis" will fill in the missing introduction to new students in terms of familiarising themselves with NUS.

In addition, university life can be rather monotonous and boring, and some may wish for more spice in their day-to-day lives at NUS for current students. There aren't many games that are based on our NUS campus either. In our game, student life has more excitement, as it involves the main character running around campus as they complete missions to fight a lion monster rampaging through the campus.

Another major aspect we want to showcase in the game is the diversity of different academic majors within NUS. In the game, we wish to form a team of different students from various academic departments, whereby they will not only work on developing the cure but also on working to create power-ups for the main character, which will be discussed later in the story.

Vision:

Campus Crisis is a single-player 2D computer video game which takes place in NUS. We aim to produce a fun game for new and existing students alike. It is an adventure puzzle-solving game packed with action and challenges. The incoming students can familiarise themselves with the NUS campus while existing students can get the thrill of adventure in familiar territory.

Storyline:

In the middle of a relatively peaceful environment in Central Library, you are studying for an upcoming assessment. You keep receiving what appears to be anonymous spam messages, which contain unknown instructions for looking for specific individuals from different majors. Upon a quick glance, the messages have a mysterious but cautionary warning of a disaster of sorts, and that these people would help alleviate the catastrophe. Weirded out by the texts, you ignore them and continue with your studies.

Just as you are about to pack your things to head back home, a lion monster appears out of nowhere to wreak havoc and chaos. Everyone is confused, frightened and panicked by such an unusual and cataclysmic event, running away from the lion. All of a sudden whilst you are running for your life, the spam messages make sense. You start to frantically scroll through those texts and read through images and descriptions of those individuals. In bold, you read that all these people are important in developing a cure to neutralise the lion and that they must be found in a specific order. This would've been easy if it weren't for that pesky lion making it difficult to look for the team of individuals you wish to assemble. This is where the game begins.

Your first person to look for, or a key individual, is a physics student with a specialisation in quantum technologies who has recently developed prototype devices to teleport to his lab. After finding him successfully, you provide a full dissertation on the situation and he is convinced to lend you a prototype of his teleportation device, which you will use to teleport to the base of operations in his lab.

Time is running short, and you still have to find more people to help create a cure for the lion. Next, you will look for a student from Management who will help organise and manage the team, as well as a biochemical engineering student. After successfully finding these allies, the biochemical engineering student will kickstart the development of the cure. He also donates a pepper spray can to use against the lion.

You then level up to a different location, this time in PGP which is larger than the previous location. Next on your list are a Sports Sciences major and a chemistry student who just so happens to be neighbours in Pioneer House. The sports sciences student, who was rather

difficult to catch up to earlier, will give you advice on how to run faster, unlocking a speed-up ability. He also starts strategizing on how best to coordinate your way around in order to evade the lion. You now have the option to use your coins to buy a communications device to communicate with him while running, essentially giving hints for the game.

The chemistry student helps the biochemical engineering student provide insight and knowledge to better improve upon the formula, and she also has access to more chemicals and compounds in her personal lab which they can teleport to. The biochemical engineering student has also just created a prototype which needs to be tested on the lion, warning that its concentration is not high enough to neutralise the lion. He and the physics student have created a tranquiliser dart gun for you to experiment on the lion on your next encounter.

Afterwards, you look for a Mechanical Engineering student and a veterinary sciences student to enlist in your team. After successfully recruiting them, the mechanical engineering student said she already had a trap in development to catch the lion and will work on it to improve it further, while the veterinary sciences student joins the cure development team. The mechanical engineering student and the veterinary sciences student were also able to whip up an air horn that releases a specific frequency to hinder the lion.

After a few days, the cure has been developed with a high chance of neutralising the lion, and the trap has been set in UTown to finally bring an end to the lion's terror. Using all the weapons and power-ups, you make one final showdown against the lion. In this boss level, you must run fast enough and make sure you don't get trapped yourself. Once you successfully tranquilise the lion, the lion is finally sent back to the local zoo, and your team will be awarded for the journey you have made along the way.

User Stories:

1. As a local incoming student who lives far away, I want to be more familiar with the campus without having to physically travel to campus. I want to immerse myself within realistic NUS locations while playing the game.
2. As an international incoming student, I want to experience moving around campus in my own time so I can adapt to a new country more easily.
3. As an incoming student who did not get a chance to attend orientation camps, I want to be familiar with the campus so I can get around easily.
4. As a current student at NUS, I want to play a game set on campus and get a taste of adventure. I want to feel the adrenaline and urgency of searching for specific people while being hunted down by a lion.
5. As a player, I want to play a game similar to Where's Waldo that can train my eyes and visual coordination to find the key ally. There aren't many games like Where's Waldo, much less one that involves a moving crowd which adds a layer of challenge to it.
6. As a student, I want to get a feel of working in a team of different students from different academic departments and backgrounds. I want to know what it's like to

work with different faculties as this will reflect working with a diverse group of people in the future, career-wise.

Project Scope:

“Campus Crisis” is a single-player 2D game based in real-life locations within NUS, whereby the characters consist of NUS students and staff from various academic faculties, the key allies and the lion.

Mainly these action game mechanics will be utilised: namely up, down, left, right and tagging the key ally. In addition, the backstory of the lion’s mystery will gradually be explained as the game goes on. In the meantime, power-ups such as teleportation, speed and visibility will be introduced with the help of the allies step by step.

Features to be completed by mid-June:

1. Finished designs and animations for the crowd of non-playable characters
 - a. We aim to ideally have a total of 15 characters per scene for the player to sift through to look for the key ally
 - b. Animations of the characters will be haphazard to add a layer of difficulty in searching and navigating through the location
2. Finished all maps’ graphic designs
 - a. Detailed and near-accurate representation of the NUS campus locations, such as assets and positioning of the assets
 - b. Will be completed based on whether the collisions and character interactions are completed
 - c. Priority given to Central Library, followed by PGP then UTown
 - d. The map designs to be completed are UTown, Central Library and PGP, whereby we look to having UTown be the final boss level in defeating the lion.
3. Completely learn how to properly update version control on repository
 - a. Unfortunately, not much effort is being used on this aspect of the project, with us only knowing the basics of it
4. All power-ups and weapons to be completed
 - a. Power-ups to include visibility, speed and teleportation, with more to be added
 - b. Weapons include fire torch, pepper spray, tranquilliser dart and air horn, more to be added.
 - c. Complete designs and animations of weapons and main player using weapons

Features to be completed by mid-July:

5. Completion of the player tutorial game and cutscenes
 - a. Playable tutorial section with instructions on how to use mechanisms/ basic inventory/ basic weapons before the start of the game
 - b. Completed animated cutscene sequences to be used in between levels
 - c. Dialogues to be written for cutscenes

6. Addition of music and audio sound effects for gameplay
 - a. Includes lion roaring, crowd screaming, video game sound effects etc.
 - b. Possible background music which can be borrowed from royalty-free library
7. Possible inclusion of final boss level set in UTown
 - a. Completely different game mechanics required
 - b. No longer searching for allies, instead acting as bait to trap the lion and neutralise with a cure
 - c. UTown is the largest place to be designed, and since it is more crowded more characters need to be designed

Proposed Core Features:

1. Game map: Campus Crisis features 3 locations in NUS: UTown, Central Library and PGP Residences. The game map allows the player to have a good overview of the layout of the locations to spot the allies and complete the missions.
2. Reward system: By completing missions and finding allies, the players can collect coins redeemed at the inventory shop to enhance their character and player experience. The coin count will be based on how fast the player finds the ally and the number of attempts needed to find the ally.
3. Inventory shop: Players use the collected coins to buy additional weapons developed by the team or increase their player skills here. The weapons will be used to temporarily slow down the lion for a significant amount of time. These weapons will include a fire torch, Also, power-ups can be bought to boost the player's abilities, which include visibility, coin multiplier and speed.
4. Action buttons: The direction buttons: up, down, left, right, control the direction of movement of the player. The action buttons allow the player to tag the allies and complete tasks.
5. Save and load system: A save and load system will be utilised to allow players to pause and resume the game at any time, retaining their progress.

Design:

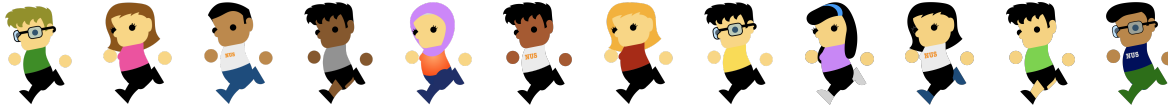
The characters will be designed to look as similar as possible to each other in terms of body shape and size. The characteristics that will set them apart would be clothing, hair, skin colour, accessories etc. Unlike many Godot-developed games, we are attempting to steer away from a pixelated look because the main aspect of the game is to differentiate between characters, which would be severely limited with pixelated designs. Hence, we aim to use vector graphics to design the characters and assets.

Originally for the video for Mission Control 1, Keynote was used to design the characters as well as their animation frames. Keynote was also used for designing the assets that were

seen in the video. As such, the animations, character designs and assets are reused for the video game itself.



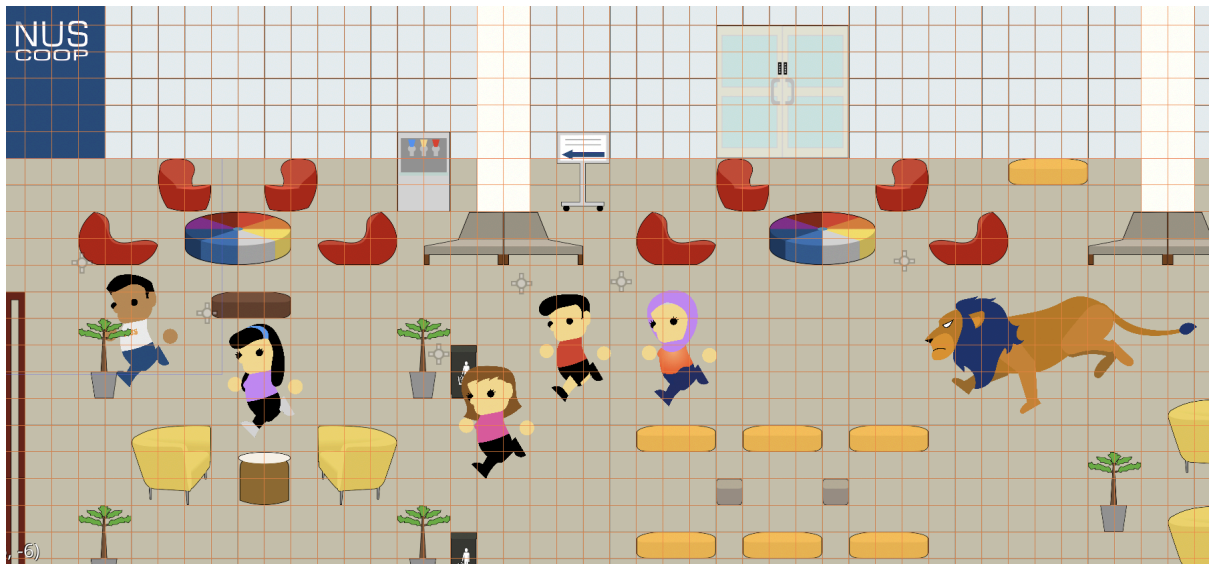
Frames for animating the main character, which were reused to create a sprite sheet.



Non-playable characters, some of whom will be key allies

As for the game design itself, we initially wanted to make a platformer game similar to Mario as it was easy, but after some consideration, we decided to challenge ourselves by making an RPG game. We initially wanted to make it a mobile app game too, but due to time constraints, we decided to make it a computer game instead.

The design of the game is inspired by real-life locations within NUS. Image and video references were used to create near-accurate depictions of the assets and locations for the game. Again, it is not 100% accurate but will be similar enough to provide a good introduction to NUS to newcomers. The assets also have to be positioned in such a way that it would make way for a crowd of people to run in, so the distancing of the assets would be different from their real-life counterparts.



Depiction of the ground floor of Central Library

Core Features Developed

1. Crowd of non-playable characters with finished designs and animations

Added crowd of 3 NPCs and 1 ally instead of 15 NPCs to make it clearer to the player which NPC they are tagging

2. Complete graphic designs for all maps in the game

4. Full implementation of power-ups and weapons

Weapons will be on the ground around the game map for players to collect and use to enhance their player statistics

5. Integration of audio sound effects for enhanced gameplay experience

Software Engineering practices utilised:

1. Separation of Concerns

Each section of code only does 1 thing. The code for each node is more organised and easier to read. In the case where only a certain property of the node does not work, it is faster and less confusing to identify the section of code that presents the problem.

2. Open Closed Principle

Each node is open for extension but closed for modification. Additional functionalities can be introduced without having to change the base code. This allows the nodes to be more flexible and less susceptible to bugs that may take a longer time to fix.

3. Loose Coupling

Each file of code and functionality for a node is interconnected with code for other nodes mainly through the 'world' scene. In case a node fails it can just be removed from the 'world' scene and the other components can continue functioning.

4. Model, View and Controller (MVC) model

The model: The functionalities of each individual nodes like the main player, NPCs detects and updated the states of variables like in collision detection to trigger the appropriate outcomes on receiving user input

The view: we used a separate node for the components of the user interface like the tag button, pause button, pause menu, messages from the game etc.

The controller: The user interface node contains buttons to receive user input and signals in the scripts are triggered by the user input to produce the respective outputs.

5. User testing

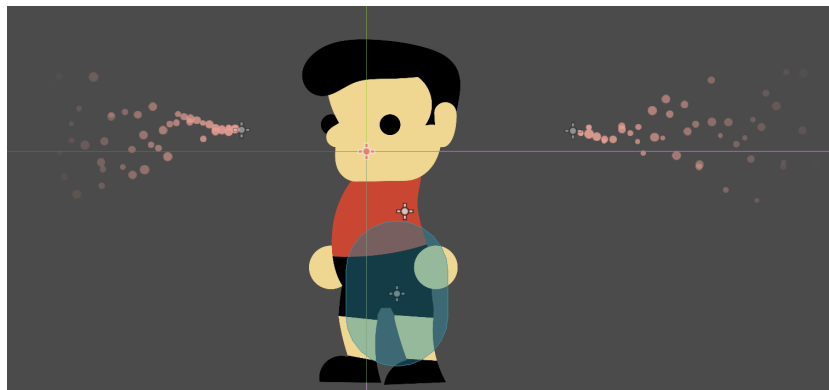
The prototype of the game was tried out with different players and feedback was gathered on the experience of playing the game so we could adjust the game difficulty level and adjust the layout of certain user interfaces to make it more comfortable for the player.

6. Version Control

Version control was handled via branching as a sandbox for developing new features so bugs in the new features will not interfere with the basic functionalities of the game in the main branch

Product Features:

1. Main player



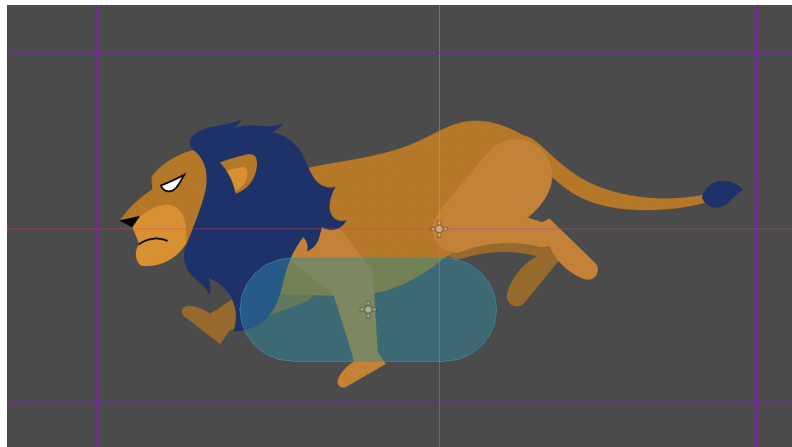
The player node was a `CharacterBody2D`, and has three child nodes: `AnimatedSprite2D` for animations such as “running”, “idle” and using weapons, `CollisionShape2D` for it to collide with the environment and the lion, and `Camera2D` so the camera follows the player and his movements. Under `AnimatedSprite2D` are the child nodes for spray particles as shown above and audio nodes for whenever each weapon is used. A speed of 600 is given to the player. Below is how the player velocity is computed:

```
func move_body(current_dir):
    var moveDirection = Vector2.ZERO
    if Input.is_action_pressed("up"):
        moveDirection.y -= 1
    elif Input.is_action_pressed("down"):
        moveDirection.y += 1
    if current_dir == "left":
        moveDirection.x -= 1
    elif current_dir == "right":
        moveDirection.x += 1
    velocity = moveDirection.normalized() * speed
```

It is connected by signal to the weapons buttons on clicked, causing it to change animation, velocity, weapon, flip orientation and audio. Additionally, signals from the lion allows us to determine where the lion is relative to the player. For instance, the code below shows when the torch button is pressed.

```
func _on_torch_button_pressed():
    isTorch = true
    stopped = true
    if distance_to_lion >= 406:
        $AnimatedSprite2D.flip_h = true
    else:
        $AnimatedSprite2D.flip_h = false
    $AnimatedSprite2D.play("fire_torch")
    $AnimatedSprite2D/torchAudio.play()
    $AnimatedSprite2D/torchAudio.play()
    $AnimatedSprite2D/torchAudio.play()
    await get_tree().create_timer(3).timeout
    $AnimatedSprite2D.flip_h = !$AnimatedSprite2D.flip_h
    $AnimatedSprite2D/torchAudio.stop()
    $AnimatedSprite2D.play("running")
    isTorch = false
    stopped = false
```

2. Lion



The lion node has similar child nodes to the player, which are AnimatedSprite2D and CollisionShape2D. The lion position relative to the player is computed easily, which is just the difference between the lion and player position. The sign of the integer returned determines if the lion is to the left or right of the player. Once this number hits zero, a signal is sent over that the player is killed by the lion, ending the game. The velocity of the lion is also computed similarly to the player. It also has signals connected to the use of weapons. Whenever a weapon is pressed, the lion changes direction and runs away from the player for a set time of 3 seconds.

3. Ally and NPCs



The ally and NPCs have AnimatedSprite2D and CollisionShape2D nodes as children nodes. Signals for the collision shape and the hud scene are used, whereby once the collision shape is entered, the tag button will appear. In terms of code, they are all similar in that they have signals for whenever the collision shapes are entered or exited, as well as an algorithm to compute their velocity. However, the main difference is that for the ally a global boolean variable will be set to true, whereas the other NPCs have it to false. Hence, if the collision shape for the ally is entered and the tag button is pressed, since the global variable is true the ally would be successfully tagged. This global variable also takes into account the current location node, as different locations will have different allies.

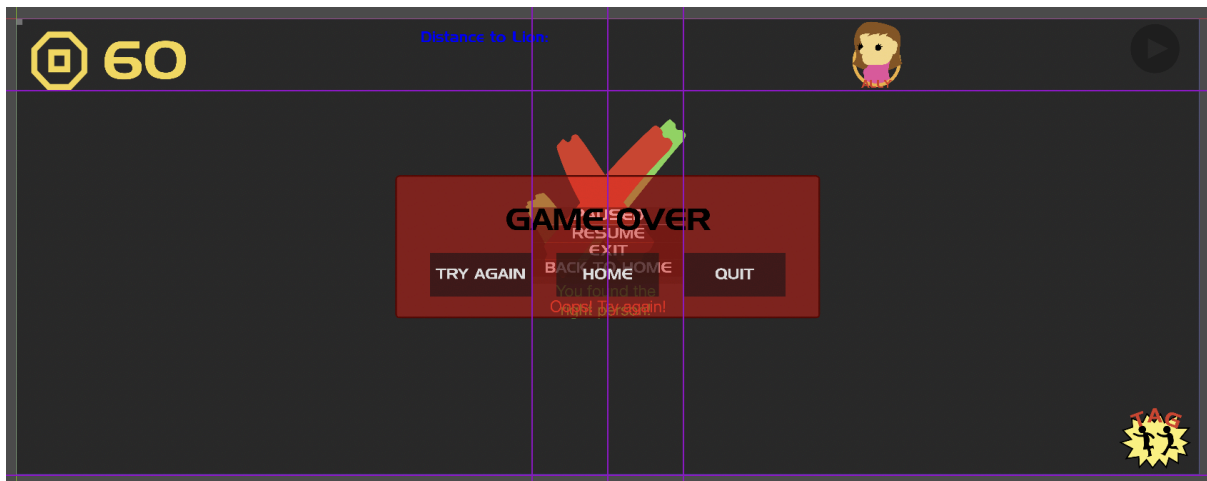
4. Start menu



This is the scene that shows before the game starts. The top corner shows the coin count, which would increase after collecting coins from the game. Only the tutorial is available upon beginning the game, and future levels would be unlocked based on the success of the previous level. There is a scroller that allows players to navigate their way to certain NUS

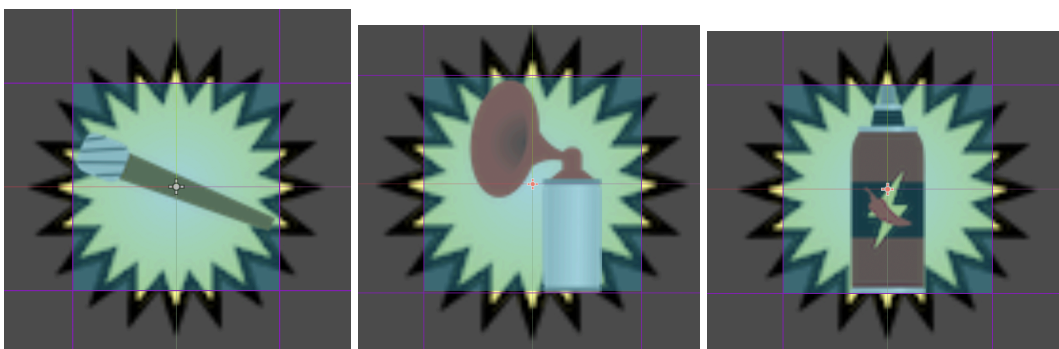
locations, though currently only the Central Library Level 1 and Prince George's Park are available.

5. HUD



The HUD, or heads-up display, show several features that would follow the player as the player runs in the game. Firstly is the coin counter, which also serves as a countdown timer. It starts from 60 and counts down to 0, meaning the player has one minute to find the ally. Next is the distance-to-lion, which shows how far the player is from the lion. There is also the sprite for the ally, which shows the profile of the ally to catch. On the top right corner is the pause button, which then shows three options to either resume, exit or return to start menu. The bottom right corner shows the tag button which is enabled whenever the player comes into contact with another person. There are also a tick and cross in the centre, which show up based on whether the right person is tagged or not. Finally there is the game over panel, which appears when the player is eaten by the lion or when the time is up. The three options available are “Try Again”, “Home” and “Quit”.

6. Weapons



There are three weapons that can be used against the lion, which are the fire torch, air horn and pepper spray. They are scattered across the library for the player to pick up. They have collision shape signals to check whether the player has come into contact with it. Once the player does do so, the weapon's button would appear in the lower part of the screen. In terms of code, the world gdscrip has three variables: firstWeapon, secondWeapon and thirdWeapon. Whenever a weapon is picked up, the following functions are called:

```
func index_weapons(weapon: String):
    if first_weapon == "none":
```

```

        first_weapon = weapon
    elif second_weapon == "none":
        second_weapon = weapon
    elif third_weapon == "none":
        third_weapon = weapon

```

```

func organise_individual_weapon(index: int, weapon: String):
    var buttonPos = Vector2(6923, 2654)
    buttonPos.x -= (index - 1) * 330
    var _button
    if (weapon == "torch"):
        _button = $hud/torchButton
    elif (weapon == "spray"):
        _button = $hud/sprayButton
    else:
        _button = $hud/hornButton
    _button.position = buttonPos
    _button.disabled = false
    _button.visible = true

```

```

func organise_weapons():
    if (first_weapon != "none"):
        organise_individual_weapon(1, first_weapon)
    if (second_weapon != "none"):
        organise_individual_weapon(2, second_weapon)
    if (third_weapon != "none"):
        organise_individual_weapon(3, third_weapon)

```

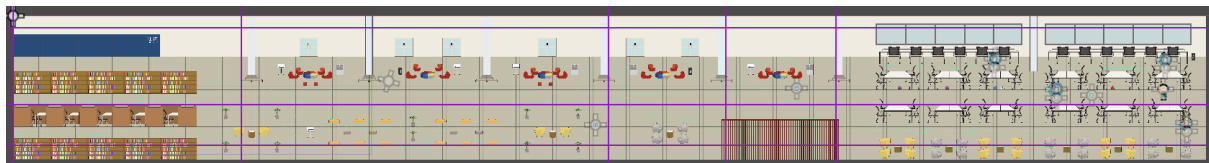
In a nutshell, the algorithm determines which index weapon the weapon picked up belongs to, then the buttons would appear based on the weapon itself and its index. After it is picked up and the button clicked, the weapons will be reorganised in terms of index and button positions. Signals are also emitted to the lion script and player script for each of them to react accordingly to the button pressed.

7. Inventory



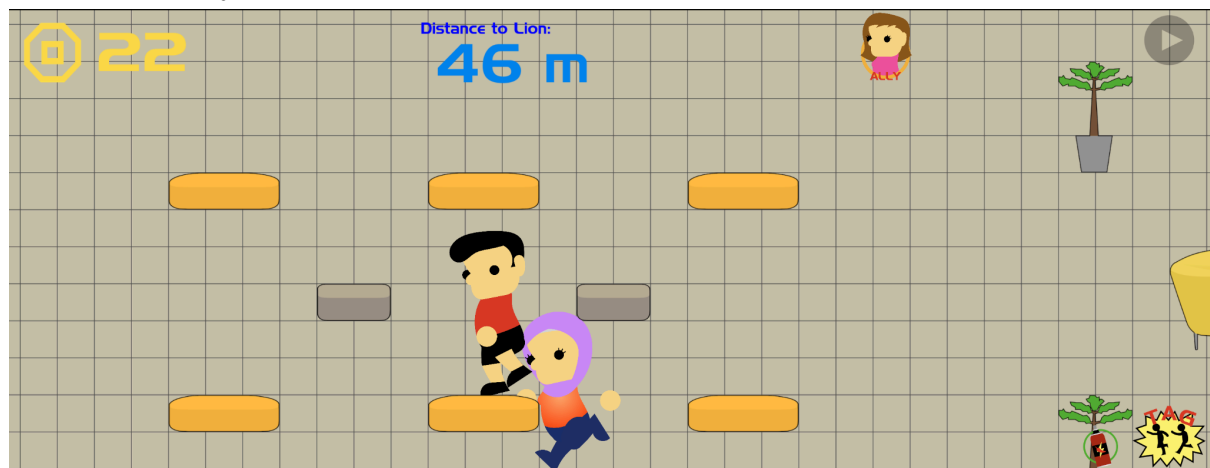
The coins collected can be used to purchase power-ups and weapons in the inventory shop. In the inventory shop, there are weapons boosts whereby buying them would allow for a free weapon of either air horn, fire torch or pepper spray at the start of the game, effectively doubling the possible weapons available. The weapons have to be picked up during gameplay for them to appear in the inventory shop. In addition to the weapons, there are also power-ups, which include an increased speed boost and an extended arm to tag the ally from further away. They cost 100 coins each and can be unlocked at any stage of the game.

8. Main map scene



The tile map for the library is inspired by the real life location. The player has to run around the map to find for the ally and run away from the lion. There are assets such as tables, couches and books in the library which are divided into several physics layers. Collision shapes and Y-sorting are also utilized so the player does not simply run over the assets. This is the main scene where all the nodes are organised, such as the main player, npcs, lion, hud etc.

9. Gameplay



The player is able to move the main character using the WASD keys or arrow keys, as well as the spacebar to make the player stop running. Once the main character comes into contact with another person, the tag button appears. There are also weapon buttons at the bottom of the screen for the player to use if the lion gets too near. If the player wants to pause, the player can just click the pause button. The distance to lion helps the player know how far the lion is, while the coin count timer shows how many seconds are left and how many coins will be obtained when the correct ally is tagged.

User testing feedback:

1. Lion often gets stuck or cannot reach player

Due to the lion's large size, it sometimes gets stuck in the library assets. This means we may have to reposition the assets or decrease the lion's collision shape size.

2. Screen size ratio

The current screen size is 7800 by 3000, which is 5:13 ratio. Suggestions included making it into a more standard computer screen ratio or even smartphone screen ratio.

3. Weapons and tag button too small

Users complained that it is difficult and inconvenient to run while having to click the small buttons. We concluded that the buttons should be larger or they could be integrated into the keyboard input map.

4. Confusing pause and space bar key

Some users clicked the spacebar button on instinct to pause the game, though it only made the player stop running while the game was still playing. This made us rethink the key input design and consider other approaches to have the player stop without confusing the user.

5. Unusual pricing of weapons

The weapons were priced at 100, 200 and 300 coins, which some users felt didn't make sense. Some said that they should be priced equally as the weapons felt equal in function and effectiveness. Others said it was too expensive and suggested lowering their prices.

6. Music and audio issues

Some users felt that the music did not fit the tone of the game. They felt that it felt too "cliche" for an adventure runner game, which makes sense as the audio was mostly pulled from royalty-free music sources, and relatively less time was dedicated to researching better music and audio files.

7.

Problems Encountered:

1. Due to the many signals used across different scripts, it was initially a little confusing. We counteracted this issue by only using 1 'world' node to connect signals across different nodes rather than connecting the 2 different scripts directly
2. Some version control commands only partially updated the GitHub repository which resulted in sometimes non-functional versions when the repository is pulled into another computer. When nodes are re-parented in the tscn files, it only gets partially updated when it is pulled from github.
3. The world.tscn file kept failing to open randomly, possibly due to broken dependencies. Fortunately, an older file was retrieved from version control on Git and the necessary changes were performed, albeit time-consuming.
4. Game over panel does not work. The panel has three buttons which are "Try Again", "Home" and "Quit", none of which work as desired.
5. Signals between nodes were disconnected on merging the branches to main so those signals had to be reconnected. It was difficult to identify which signals had been disconnected given the many scenes we had and many signals across different nodes.

Bugs Squashed

1. Game Over panel

The "Try Again" reloads the current scene, " Home" goes back to level selection screen and "Quit" closes and saves the game. All buttons work as desired.

2. Player not able to move while using weapons

Initially the player seems to be able to move and slide even when the animation stops playing on using the weapons. Now the player will remain stationary and unable to move and slide by inputs while the weapons are being used.

3. Weapon button count decreasing beyond zero

After getting a power-up, the weapons count is increased to 2. However, after using both weapons of the same kind, the number displayed indicating the counter kept decreasing beyond zero, despite already including a code for checking the count if its more than zero or not. We realised that this was caused by the queue-free function which set the button node to null, resulting in unexpected behaviour. A condition checking if the button node is null is added, fixing the bug.

4. Incorrect ally between different locations

Due to the fact that there must be different allies to tag for each location (tutorial, PGP and CLB), we faced the issue whereby the global boolean variable checking for whether the ally is tagged or not acted abnormally. As such, another global variable was added for updating the current location node. This allows us to easily update the boolean variable based on the npc script to see if the npc is an ally or not. In addition, having a location global variable proved useful for other matters, such as setting the camera tracker limits.

5. Unable to pause correctly

Due to using the pause feature in Godot, follow-up inputs after pausing could not be taken in. The built-in pause function could not be used as it would not allow for the exit or home button to work. To solve this issue, we added yet another global boolean variable to check if the pause button is pressed. Each npc, the lion and player check for this variable. If true, the physics process determining the animation and motion of them are effectively stopped, as well as the path follow nodes being disabled.

Edge features developed

1. Inventory shop

Players collect weapons and coins during the gameplay and the collected weapons will be available for purchase in the inventory shop. There are currently 3 different types of weapons available: torch, pepper spray and an airhorn. Power-ups are also available for purchase in the inventory shop: speed boost for the player, and an extended arm which allows the player to tag the ally from further away.

2. Auto save and load system

As the player enters the starting level selection menu, the previous saved data will be auto loaded. When the quit buttons are pressed, whether in the Game Over panel, pause menu or in the level selection screen, game data will be saved. Data saved include the coins collected, levels unlocked and items purchased from the inventory shop.

3. Background music and sound effects

Background music was added and sound effects for different outputs in the game to enhance gameplay experience.

4. Gameplay tutorial

The tutorial adopts a sample gameplay style, pausing and guiding the player along with instructions.

5. Level system

Players need to tag the ally from the earlier levels before unlocking the subsequent levels. Players are required to learn the gameplay from the tutorial level before unlocking the other levels. The levels starting from the lowest level are as follows: tutorial, clb, pgp.

6. New scene (Pioneer House)

An additional level was implemented, which was Pioneer House. Gameplay is similar to the Central Library scene already implemented in milestone 2.

Timeline and Development Plan:

2nd week of May:

- Familiarise with Orbital Project
- Meet with our advisor
- Design of assets, characters (main player, lion, non-playable characters) and animations using Keynote
- Design the poster and edit the video for Liftoff submissions.

3rd week of May:

- Pick up skills and learn Godot game development basics.
- Follow online tutorials and instructions to develop a game based on an already existing project.
- Learn GDScript language for coding.
- Begin working on Milestone I submission. Update README document for submission.

4th week of May:

- Design a basic prototype for "Campus Crisis" for Milestone I.
- Designing more assets, tile sets and sprite sheets
- Tag and play/pause mechanism as well as coin counter
- Familiarising with using the GitHub repository for version control updates.

1st week of June:

- Begin using Git in conjunction with sending version updates on Github repository
- Complete designs and animations for all 15-25 different unique characters
- Include separate node for comparing the tagged person and wanted key person, comparing differences and percentage wrong
- Design and animation of weapons for inventory

2nd week of June:

- Further refinement of gdscript code and gameplay, as well as collision with surrounding assets
- Random wandering of crowd of non-playable character sprites
- Continue designing more additional assets to make the location more accurate

3rd week of June:

- Improve on prototype game with further refinements and changes through testing
- Animation of cutscene for characters and game storyline
- Learn save and load system for pausing and resuming the game

- Inclusion of inventory menu and incorporation of updated inventory into gameplay
- Begin working on A1 size poster and video editing

4th week of June:

- Continuation of previous week
- Checks and feedback with advisor to improve on game

1st week of July:

- Development and addition of sound system into the game
- Refinement of levels by adding new levels (or boss level) if previous level is satisfactory

2nd week of July:

- Testing and debugging

3rd week of July:

- Continuation of any unfinished tasks
- Testing and debugging
- User experience and feedback

Tech Stack

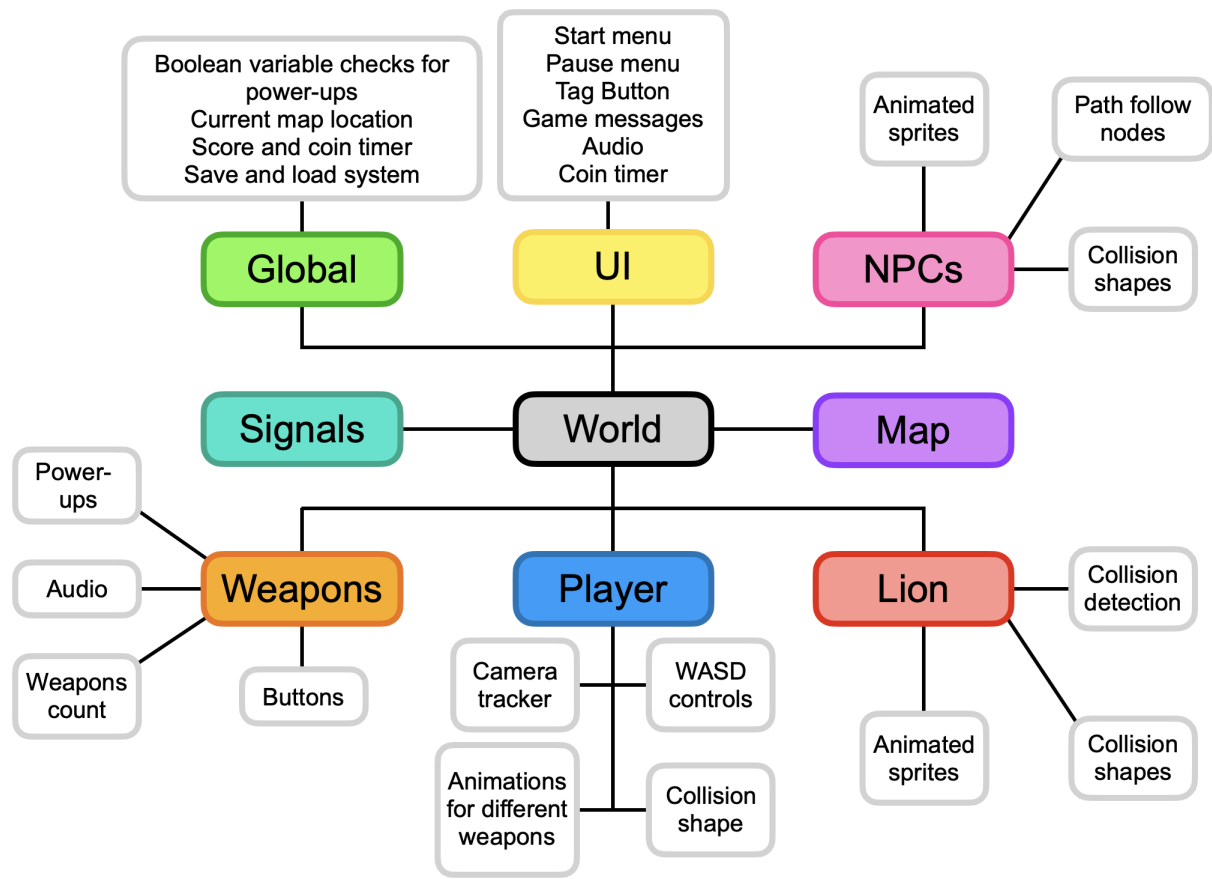
Godot

Git

Keynote

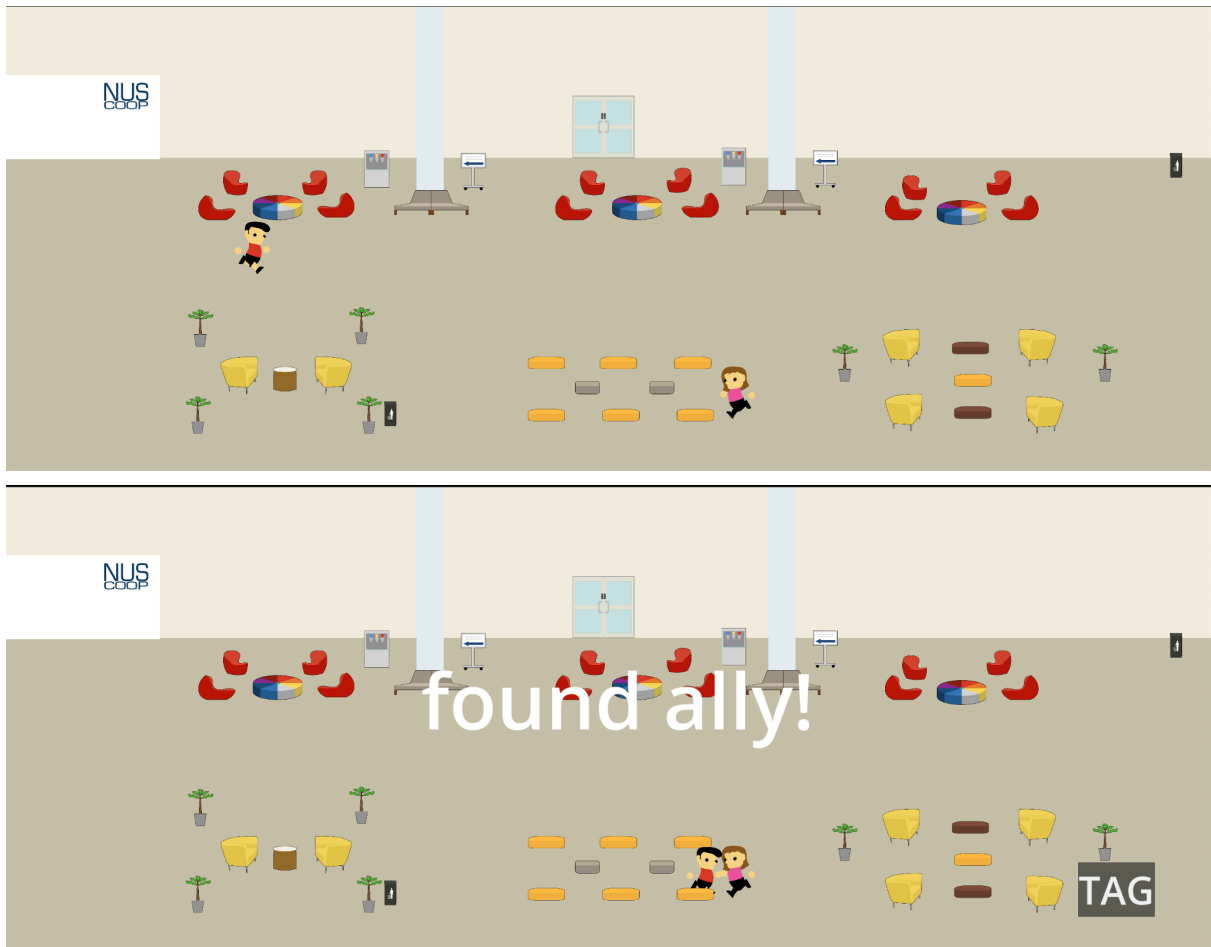
Blender

Node relationship diagram



Mockup

Milestone 1:



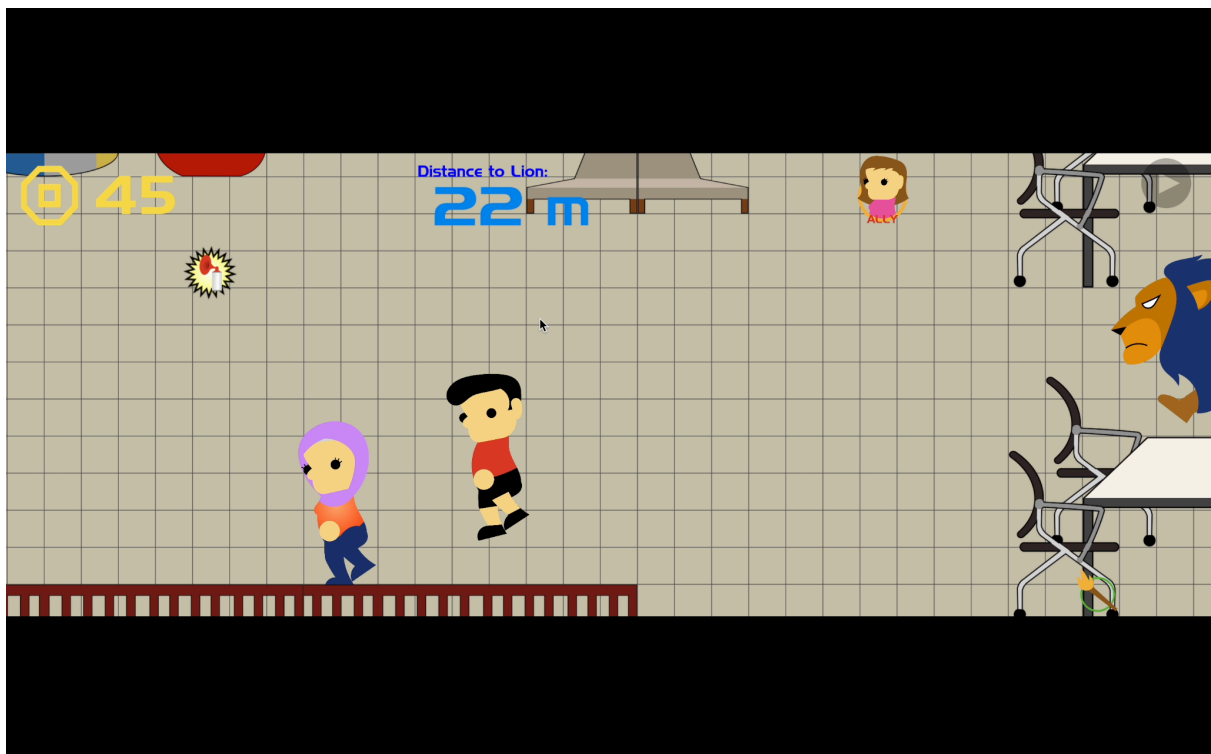
Milestone 2:
Intro scene



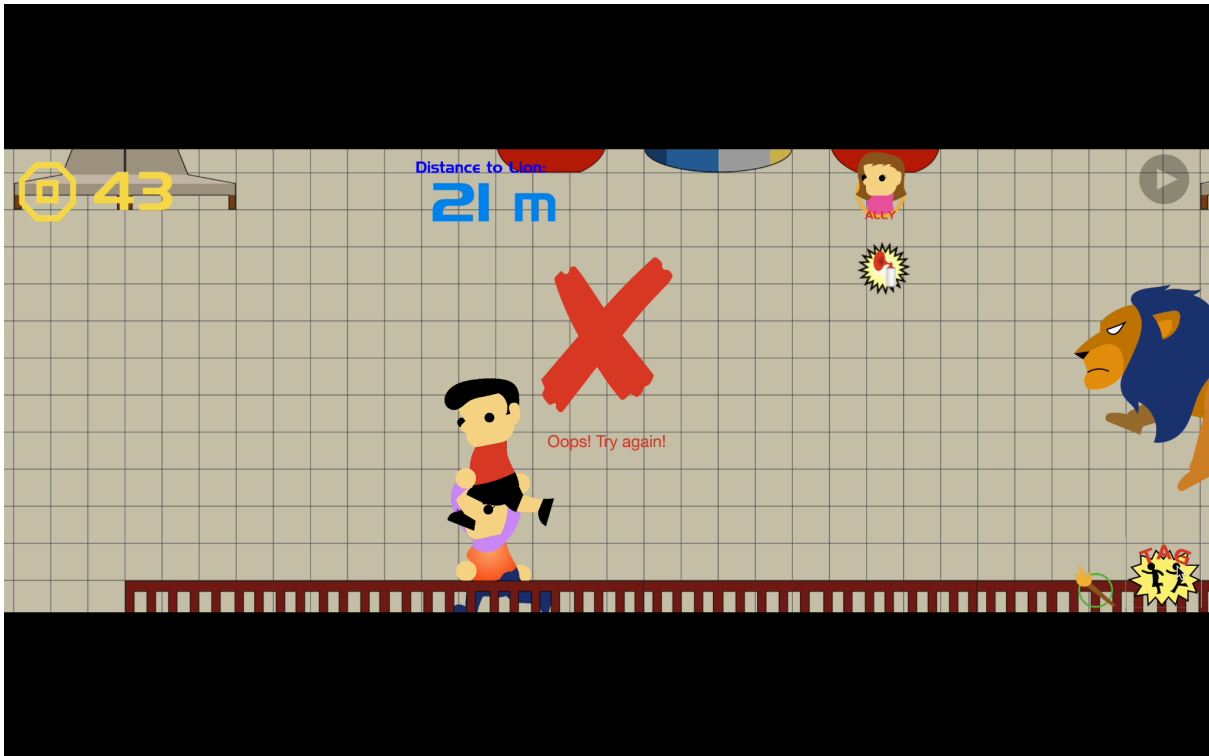
Start menu



Gameplay



Tagged wrong person



Tagged correct person



Eaten by lion- game over



Times up, game over



Milestone 3

Inventory shop with only the torch weapon collected through gameplay



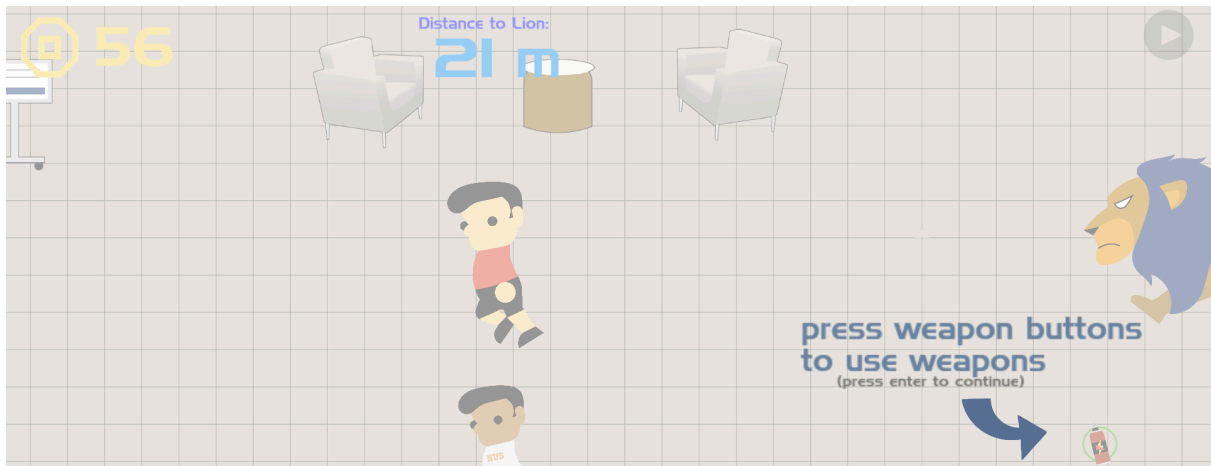
After purchasing the weapon



Power-ups in inventory



Tutorial gameplay



Level system



On clicking locked level

