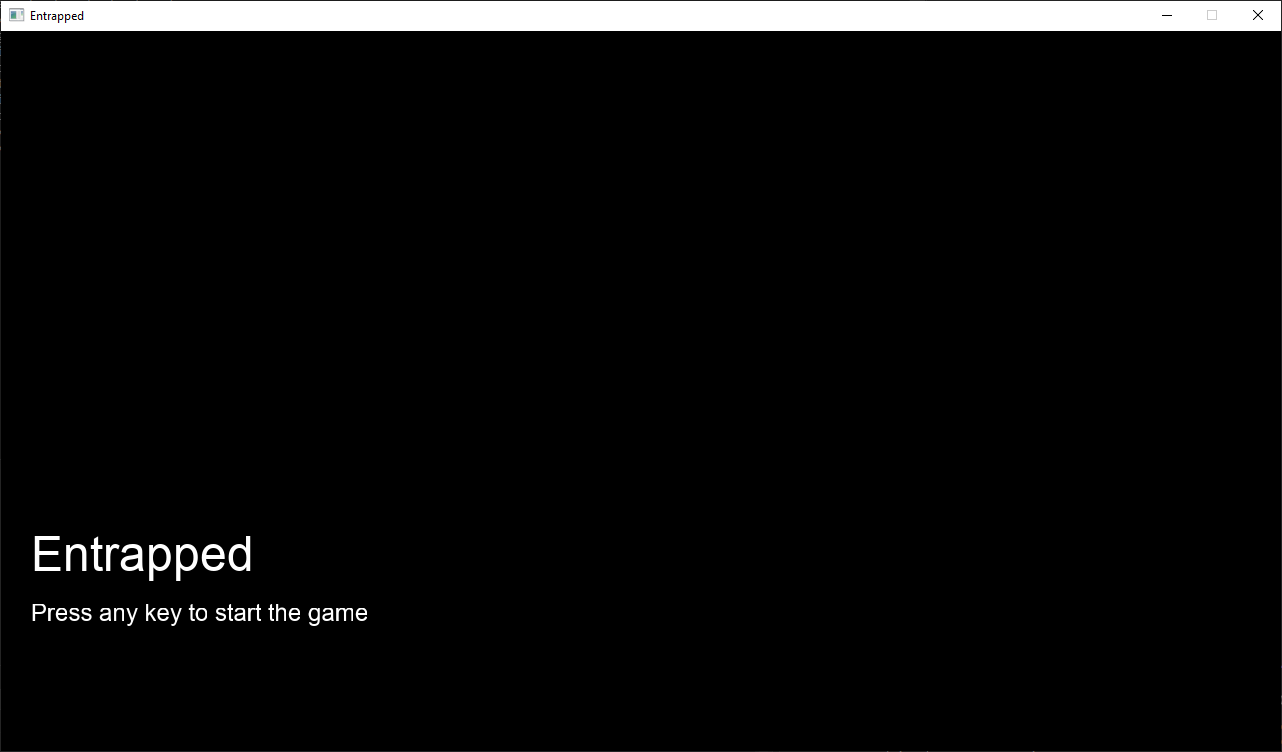
# Test Plan - M1

The expected behavior of M1 submission is as follows.

## Main menu



### Layout

* The game title "Entrapped" should appear at the bottom left of the screen.
* A smaller prompt text "Press any key to start the game" should appear below the game title.

### Operation

* Press any key on the keyboard and release to leave the main menu and enter the game scene.
* Mouse click not included.

## Game scene



### Layout

* The player (the entity on the left with green helmet and a gun) and the enemy (the entity on the right in yellow) will be spawned on a maze-like map.
* The map is tiled in 48x48 grid. The lighter part of the map is wall and the darker part is ground.
* The map is enclosed with walls.
* A looping background music will be played in the back.

### Collision

* The wall, the player and the enemy will collide with each other.
* Entities' movement will be restricted if they collide with the wall. In particular, they will not be able to move into the wall.
* When the player collide with the enemy, the player will receive damage and play damage effect
* The player can overlap with the enemy

### Damage effect

* When the player overlaps with the enemy, the player will keep receiving damage.
* The player's sprite will be tinted red and transparent.
* There is an internal cooldown for how frequent the player can receive damage. The cooldown is shown by a linear interpolation between the normal state of the sprite and the damaged state of the sprite i.e. the red and transparent version.

### Player movement

* Press WASD to move the player
  + W to move the player up
  + A to move the player left
  + S to move the player down
  + D to move the player right
* Press space to dash, the player will move with double velocity for a short period.
* Hold left shift to run, the player will keeping moving in double velocity until the key is released.
* Holding keys for adjacent directions will make the player move diagonally
* Holding keys for opposite directions will stop the player from moving along that specific direction
* Player cannot walk through the wall.

### Enemy movement

* Enemy will randomly wander on the ground. Every certain period, the enemy will pick a random direction to move. The velocity of the enemy is fixed.
* Enemy cannot walk through the wall.

### Creative features

#### Simple rendering effects

* The damage effect shader will change the player's color and opacity over time based on an internal cooldown. The current cooldown is 0.5 second.

#### External integration

* Freetype is integrated to render the text in the main menu. The installation should work on windows and mac

#### Basic integrated assets

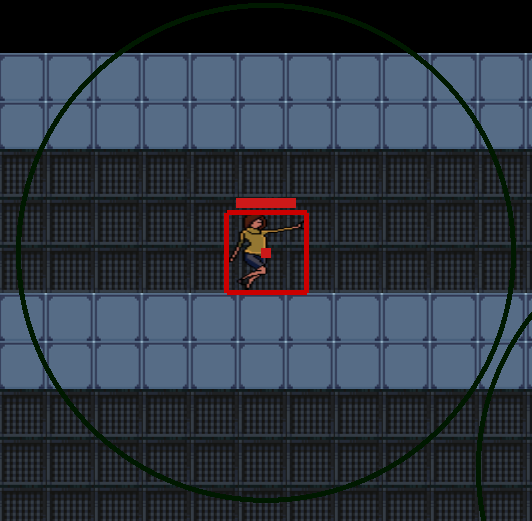
* The following assets are created and processed
  + The tiles for ground and walls. Tilesets are obtained on itch.io and processed into tiles of 48x48 in size. These tiles are rendered as textured squares on screen based on the map grid that we defined
  + The player's animated sprite. The frames of this sprite are drawn as a textured rectangle and animated in game by switching texture while the player is walking. This asset is drawn by us.
  + The enemy sprite. The texture is obtained on itch.io. The enemy sprite is drawn as a textured rectangle.

# Test Plan - M2

## Debug Mode

* **Action**: Hold semicolon key during gameplay
* **Outcome**: The detection radius of the enemies and all the bounding boxes should be shown.
* **Action**: Press semicolon upon entering the game
* **Outcome**: The mesh of the entities will be displayed explicitly on screen.

## Game AI



Enemies' AI is consisted of three states: wandering, chasing and fleeing. The large circle in debug mode represents the detection radius and the center square represents the actual position of the entity.

### wandering state

Green circle in debug mode indicates wandering state.

* **Action**: The player's center position is outside of the enemy's detection radius.
* **Outcome**: The enemy will move to a random direction every certain period.
* **Action**: The player's center position is inside of the enemy's detection radius.
* **Outcome**: The enemy will switch to chasing state. The circle in debug mode will turn red.
* **Action**: The player's center position is inside of the enemy's detection radius and the enemies' health is less than 30%.
* **Outcome**: The enemy will switch to chasing state. The circle in debug mode will turn blue.

### chasing state

Red circle in debug mode indicates wandering state.

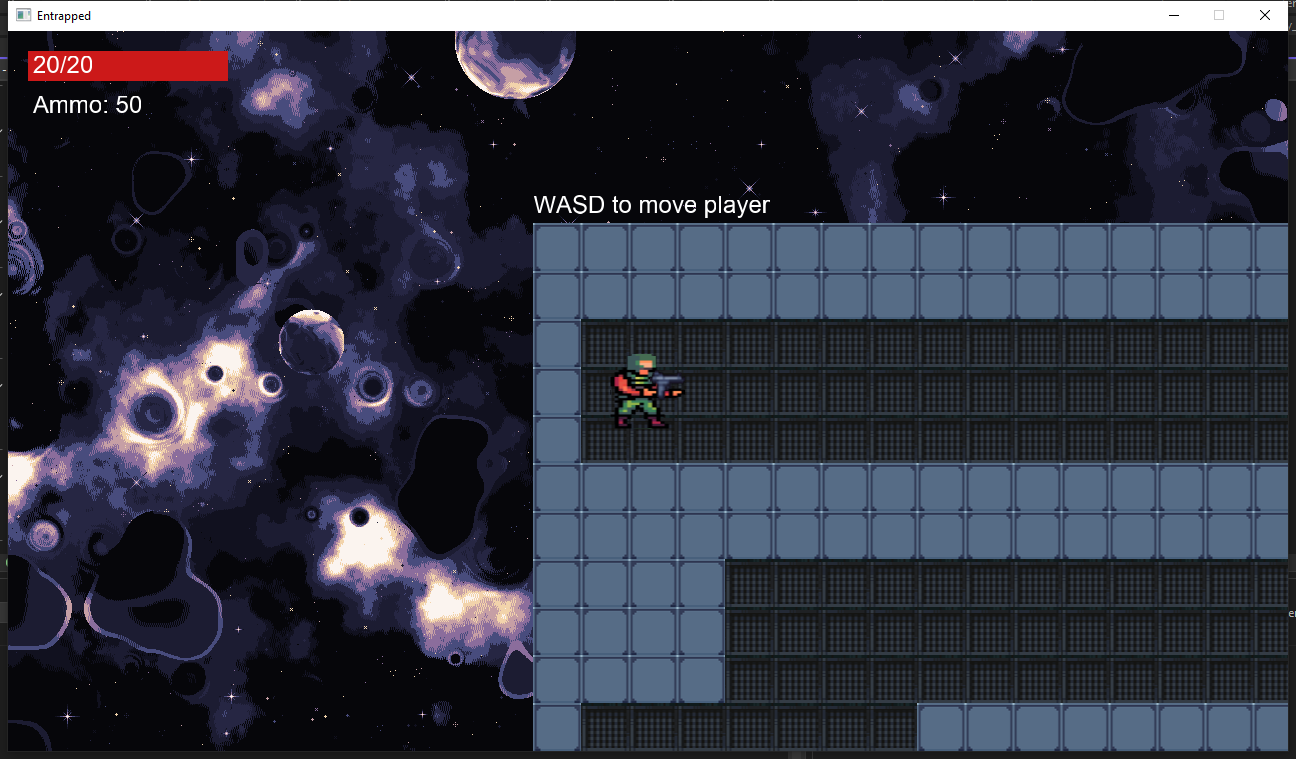
* **Action**: The player's center position is inside of the enemy's detection radius.
* **Outcome**: The enemy will move towards the current location of the player.
* **Action**: The player's center position is outside of the enemy's detection radius for less than a certain period (3 seconds)
* **Outcome**: The enemy will move towards the last seen location of the player in the detection radius.
* **Action**: The player's center position is outside of the enemy's detection radius for more than a certain period (3 seconds)
* **Outcome**: The enemy will switch to wandering state. The circle in debug mode will turn green.
* **Action**: The enemies' health is reduced to less than 30%.
* **Outcome**: The enemy will switch to fleeing state. The circle in debug mode will turn blue.

### fleeing state

* **Action**: The player's center position is inside of the enemy's detection radius.
* **Outcome**: The enemy will move away from the current location of the player.
* **Action**: The player's center position is outside of the enemy's detection radius for less than a certain period (3 seconds)
* **Outcome**: The enemy will move away from the last seen location of the player in the detection radius.
* **Action**: The player's center position is outside of the enemy's detection radius for more than a certain period (3 seconds)
* **Outcome**: The enemy will switch to wandering state. The circle in debug mode will turn green.

## Animation and Assets

* **Action**: Reduce the enemy's health to zero
* **Outcome**: The enemy death animation will be played and the enemy sprite will fall downwards.
* **Action**: Move the mouse.
* **Outcome**: The gun sprite in the hand of the player will rotate to aim the mouse position.
* 
* **Action**: Press E near the chest.
* **Outcome**: The chest's sprite will be changed to open. If the chest contains healing item, the player's health will be restored by 5. If the chest contains ammo, the player's ammo will increase by 10.

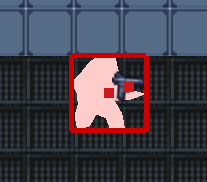


* **Action**: Start the new game.
* **Outcome**: A nebula background will be displayed in the background. It should not move with the camera.
* **Action**: The player's health is reduced to zero.
* **Outcome**: The player will lay down on the ground. The screen will be darkened and transit to the death scene.



* **Action**: Left click during game scene
* **Outcome**: Bullet will be shot from the gun, towards the mouse position

## Mesh-based Collision



The pink shape in debug mode represents the mesh of the sprite

* **Action**: Entities move towards the wall.
* **Outcome**: When the mesh touches the wall, the entity movement should be blocked. Except for bullet, the bullet will be removed upon touching the wall.
* **Action**: Entities move towards the chest.
* **Outcome**: When the mesh touches the chest, the entity movement should be blocked. Except for bullet, the bullet will be removed upon touching the chest.
* **Action**: The player mesh collides with the enemy mesh
* **Outcome**: The player will receive damage and the health bar in the top left corner will decrease in size.
* **Action**: The bullet mesh collides with the enemy mesh
* **Outcome**: The enemy will receive damage and the health bar on the enemy will decrease in size.
* **Action**: The player mesh collides with the bullet mesh
* **Outcome**: Nothing happens



* **Action**: The player mesh collides with the stairs
* **Outcome**: If the player is in the first stage, the player will move to the next stage. If the player is in the second stage, the game will transit to the ending screen.

## Help

* **Action**: Move close to certain entities in the first stage (tutorial stage). This includes
  + Enemies, will display "Left click to shoot!"
  + The top right chest, will display "Chest with healing item. Press E to open"
  + The bottom left chest, will display "Chest with ammo. Press E to open"
  + The top left corner, will display "WASD to move player"
  + Around the bottom of the map, will display "Hold shift to sprint"
  + The top left corner in the second stage, will display "Look for the exit of the maze!"
* **Outcome**: The texts above will be displayed near the entity.

## FPS Counter

* **Action**: Press F.
* **Outcome**: A FPS counter text will be displayed at the top left corner.

## Creative Elements

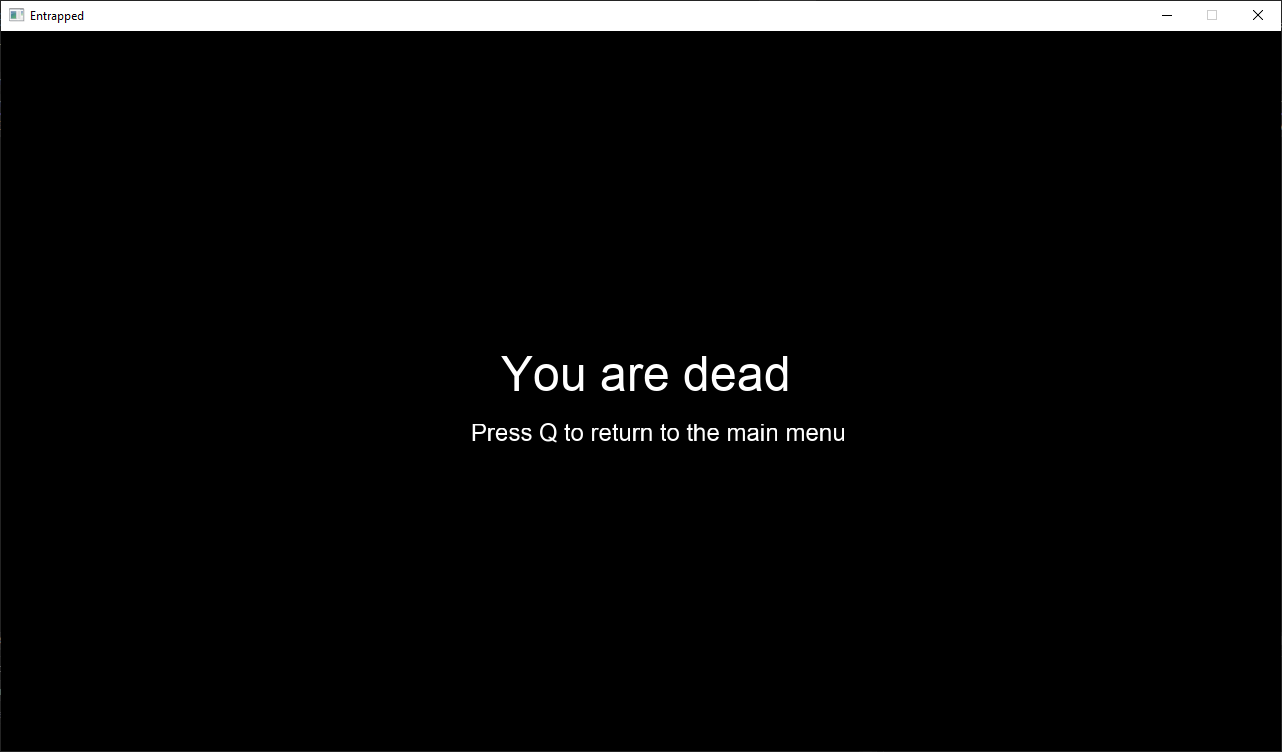
### Audio Feedback

* **Action**: Left click to shoot bullet
* **Outcome**: The bullet shouting sound will be played
* **Action**: Player receives damage
* **Outcome**: The player damage sound will be played
* **Action**: Open chest
* **Outcome**: If the chest contains healing item, the healing sound will be played. If the chest contains ammo, the item pickup sound will be played.
* **Action**: Enemy dies
* **Outcome**: The enemy death sound will be played.
* **Action**: Player dies
* **Outcome**: The player death sound will be played.

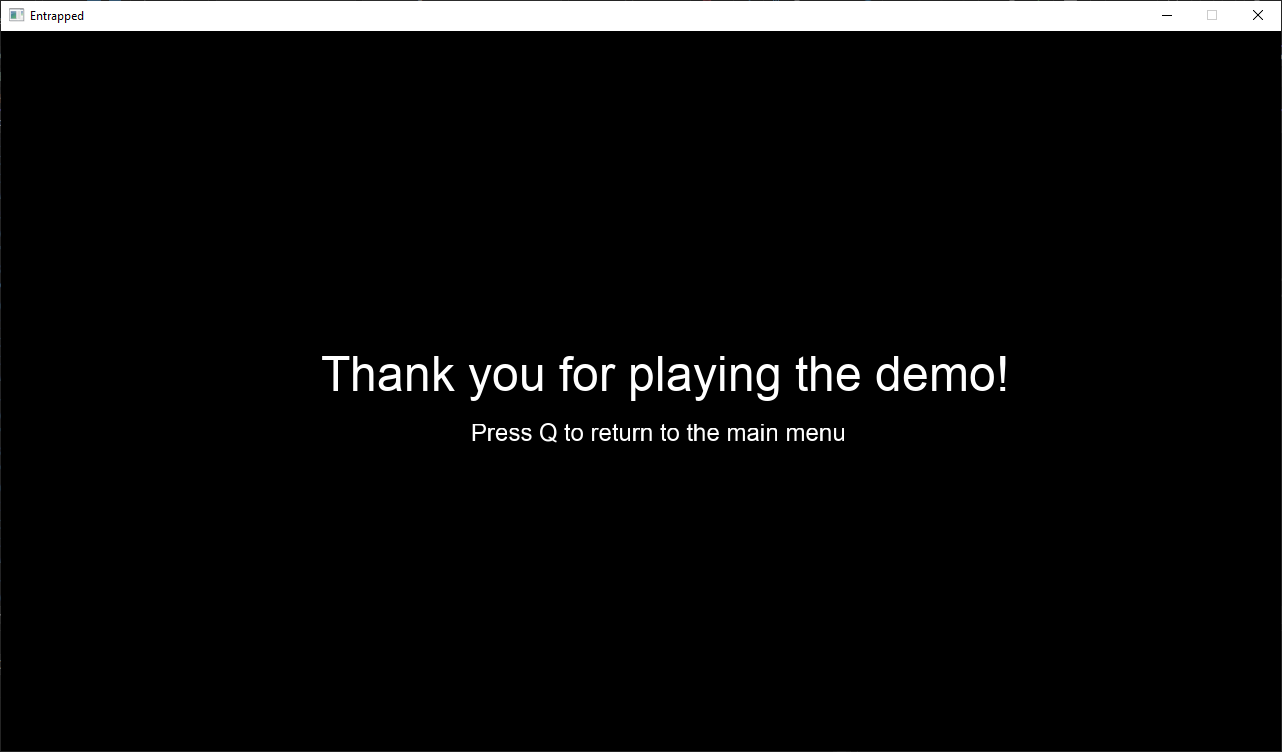
### Camera Matrix

* **Action**: Move the player
* **Outcome**: The camera will move with the player with the player always in the center. This should be computed based on camera matrix.

## Miscellaneous



* **Action**: Press and release Q during death scene (The scene shown after the player died)
* **Outcome**: Return to the main menu



* **Action**: Press and release Q during ending scene.
* **Outcome**: Return to the main menu

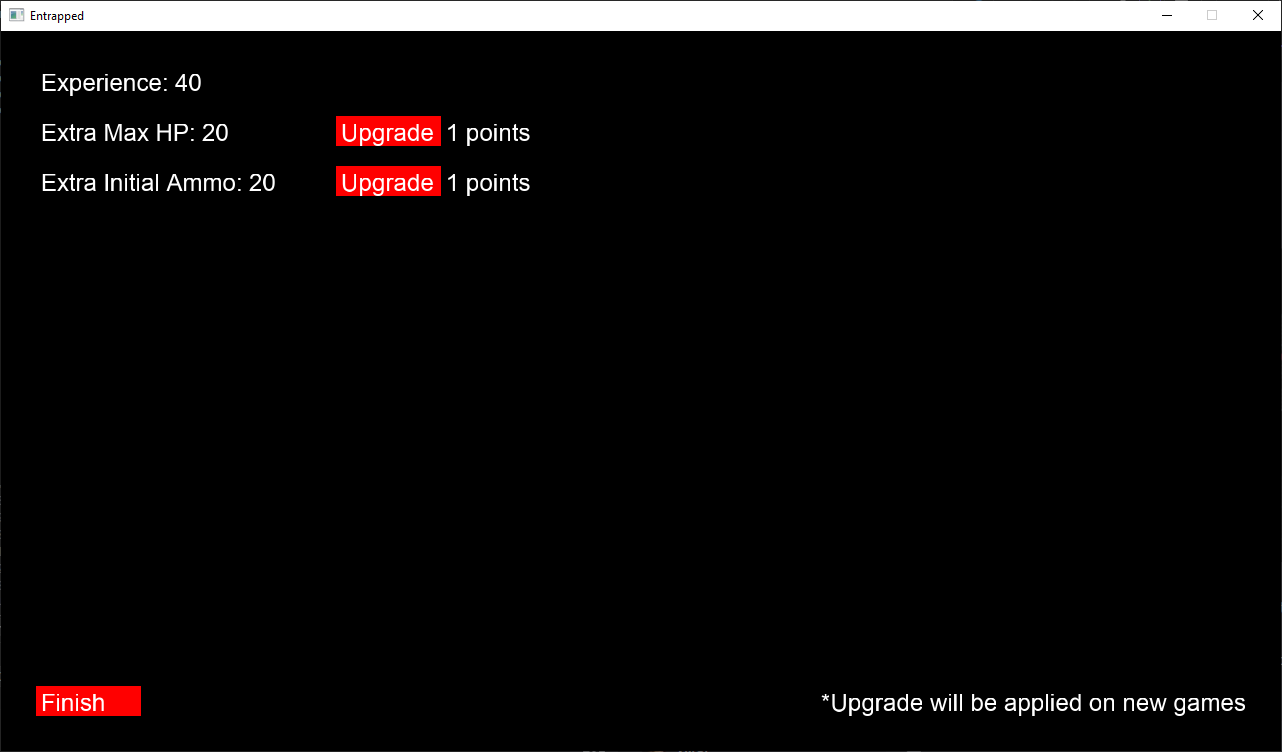
# Test Plan - M3

## Main Menu



* **Action**: Press the "New Game" button
* **Outcome**: A new game will start and the player will enter the profession selection scene.
* **Action**: Press the "Continue Game" button
* **Outcome**: The player will enter the game scene. The player will be spawned at the last map that he/she reached last time. This button will not show if there is no available save
* **Action**: Press the "Upgrade" button
* **Outcome**: The player will enter the upgrade scene
* **Action**: Press the "Debug: Off" button
* **Outcome**: The button will be changed to "Debug: On". Pressing again will change the button back to "Debug: Off". This button toggles the display of mesh data in game. Debug mod will also disable the FOV shader mentioned below in the first map

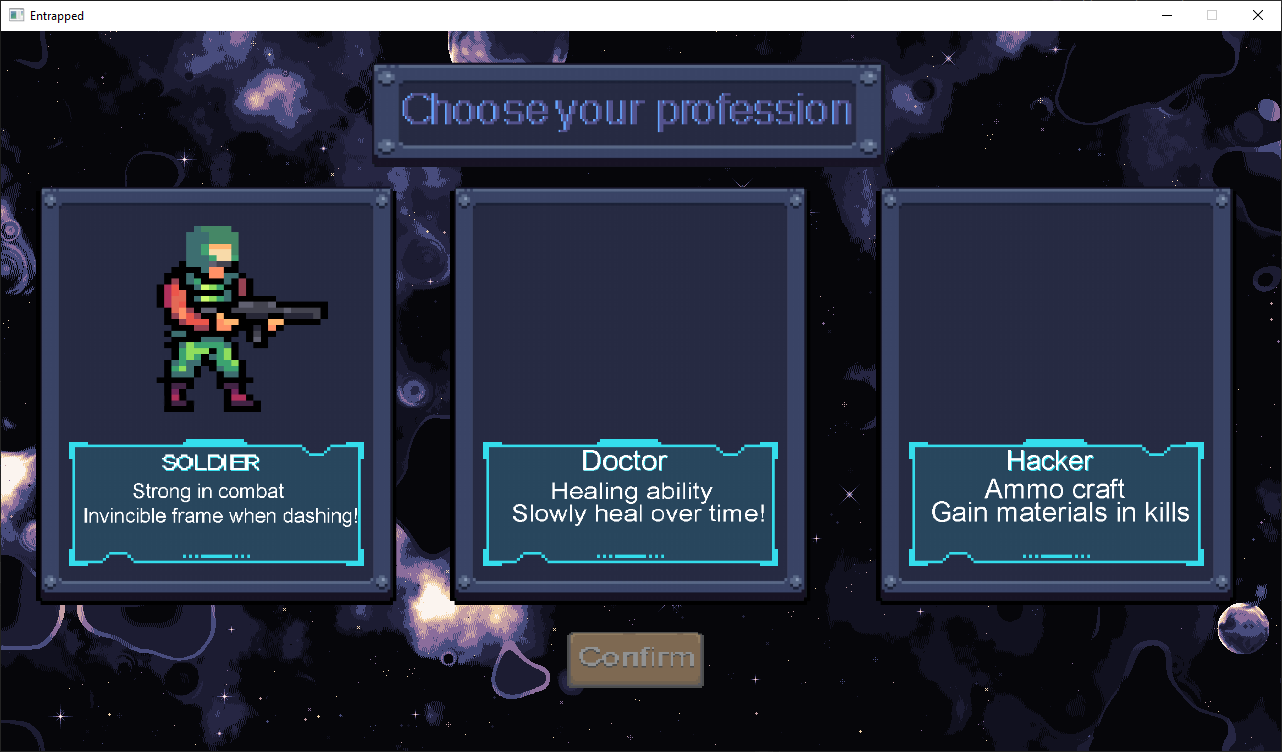
## Upgrade Scene



This is where the play may spend the experience points they gained in game to upgrade themselves for future games.

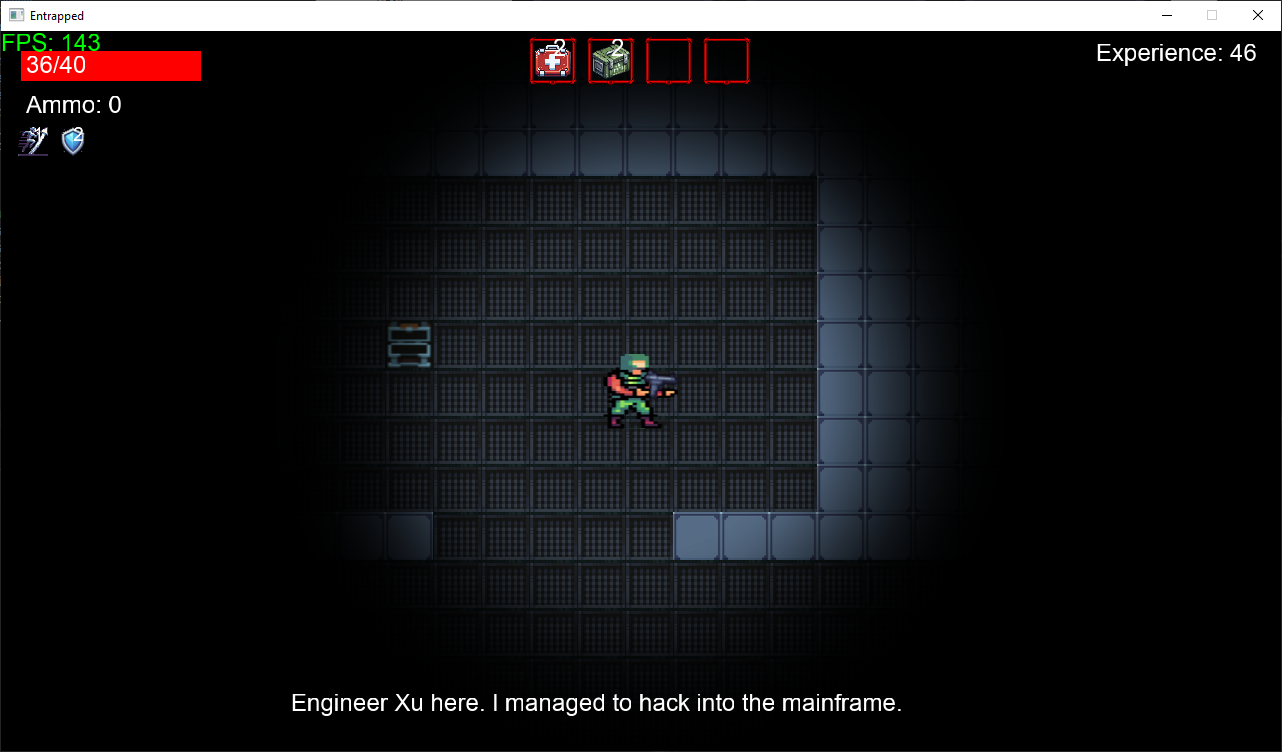
* **Action**: Press the "Upgrade" button beside the "Extra Max HP"
* **Outcome**: Experience points will be deducted by 1 and the extra max HP will be increased by 10. For the next run, the player will have X more max HP, where X is the value displayed near "Extra Max HP". If the current experience point is less than 1, there would be not effect.
* **Action**: Press the "Upgrade" button beside the "Extra Initial Ammo"
* **Outcome**: Experience points will be deducted by 1 and the extra initial ammo will be increased by 10. For the next run, the player will start with X more ammo, where X is the value displayed near "Extra Initial Ammo". If the current experience point is less than 1, there would be not effect.
* **Action**: Press the "Finish" Button
* **Outcome**: The player will return to the main menu and the upgrades will be saved. Note that closing the window in upgrade menu will not save the changes.

## Profession Scene



* **Action**: Press on the one of the three panels
* **Outcome**: The corresponding profession will be selected and the confirm button will be activated.
* **Action**: Press on disabled confirm button
* **Outcome**: Nothing should happen.
* **Action**: Press on activated confirm button
* **Outcome**: The player will enter the game scene and start the new game with the selected profession. Each profession will have a unique passive described in the profession scene.

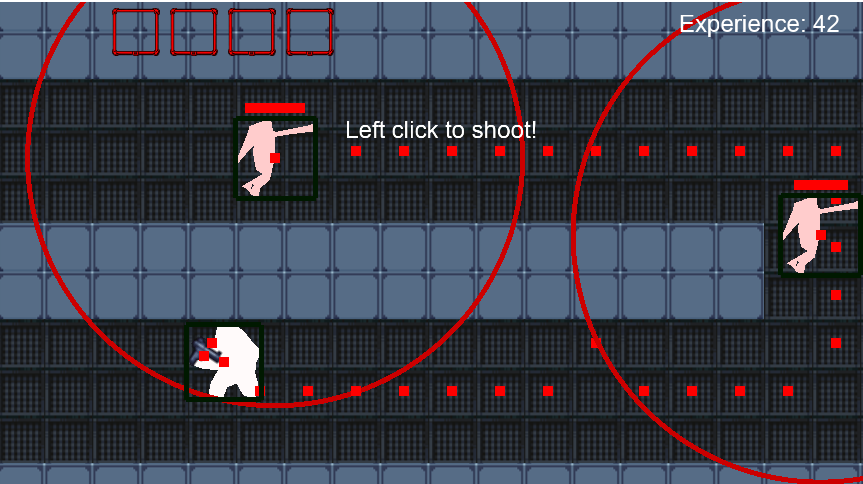
## Game Scene



### FOV Shader

* **Action**: Enter the game scene
* **Outcome**: The player's vision will be displayed as a dynamic radius, as shown by the picture.

### Path Finding



* **Action**: The player enters the detection radius of the enemies and the enemy's health is above 30%
* **Outcome**: The enemy will shift to chasing state and start to look for a path towards the player. The path will be shown as dotted line in debug mode. The enemy will attempts to compute a path every 0.75s with a modified A\* algorithm.

### Precise Collision

Minimum Translation Vector (MTV) and a modified ray marching algorithm are combined to achieve precise collision.

* **Action**: The entity moves into the wall
* **Outcome**: Entities' movement should be completely blocked by the wall and no longer stuck in the wall. There should not be any tunneling when the game freezes for too long as well.

### Consumables



Consumable is a stackable active item that will take effect upon using.

* **Action**: Press key 1-4
* **Outcome**: From left to right, the consumable item in the corresponding slot will be used. If there is no item in the slot, nothing will happen
* **Action**: Use the med kit (first one in the picture)
* **Outcome**: The player will heal for 10 health and the number of med kit will decrease by 1.
* **Action**: Use the ammo pack (second one in the picture)
* **Outcome**: The player will gain 10 more ammos and the number of ammo pack will decrease by 1.
* **Action**: Open Chest
* **Outcome**: There is a chance that the player can get med kit or ammo pack from the chest.

### Power Ups

fig:

Power up is a stackable passive ability that will take effect upon collecting.

* **Action**: Open Chest
* **Outcome**: There is a chance that the player can get a random power up.
* **Action**: Obtain speed boost power up
* **Outcome**: The player's movement speed will increase by 10%. Obtain the power up again will increase its stacked layers. Each layer will increase the player's movement speed by 10%
* **Action**: Obtain shield power up
* **Outcome**: The next damage that the player receives will be blocked and decrease the charge by 1. Obtain the power up again will increase its charge by 1.

### Story Elements - Tapes



* **Action**: Press E around the tape items on the map
* **Outcome**: An audio will be played to explain the background story of the game. A caption will also be displayed at the bottom of the screen.
* **Action**: Play other tapes while a tape is playing
* **Outcome**: The current audio will be interrupted and the new audio will be played immediately
* **Action**: Exit the map while a tape is playing
* **Outcome**: The current audio will keep playing.
* **Action**: Exit the last map while a tape is playing
* **Outcome**: The current audio will be interrupted.

## Miscellaneous

* **Action**: Kill an enemy
* **Outcome**: The player will get 1 experience point.

### Reloadability

* **Action**: Enter a new map in the game scene
* **Outcome**: The game will be saved. The following data will be saved and recovered when the player continue the game
  + Player's current health
  + Player's current max health
  + Player's current ammo
  + Player's consumables
  + Player's power ups
  + Player's experience
* Player's upgrade status will be saved as well, except they cannot be adjusted in game scene so there will be no effect.