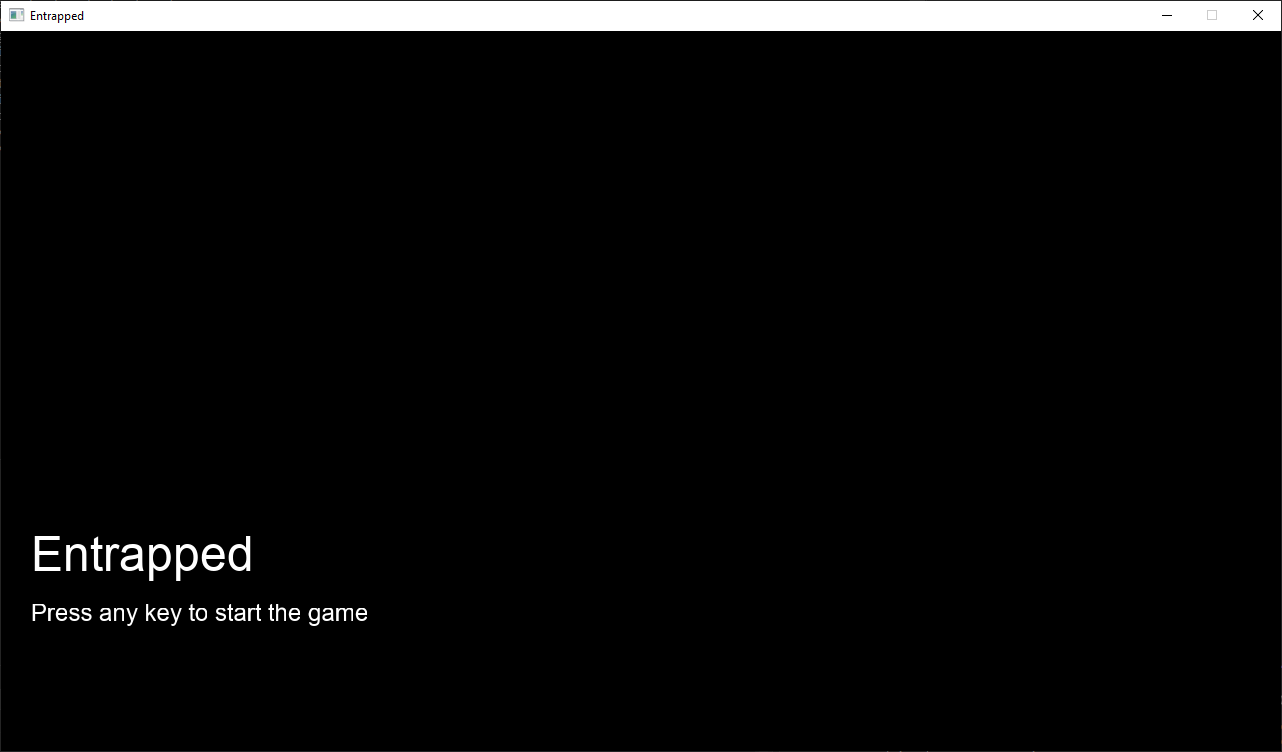
# Test Plan - M1

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

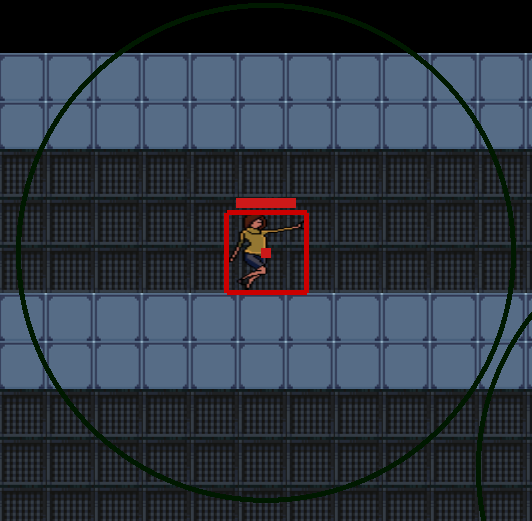
* The following assets are created and processed
  + The tiles for ground and walls. Tilesets are obtained on itch.io and processed into tiles of 48x48 in size. These tiles are rendered as textured squares on screen based on the map grid that we defined
  + The player's animated sprite. The frames of 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.

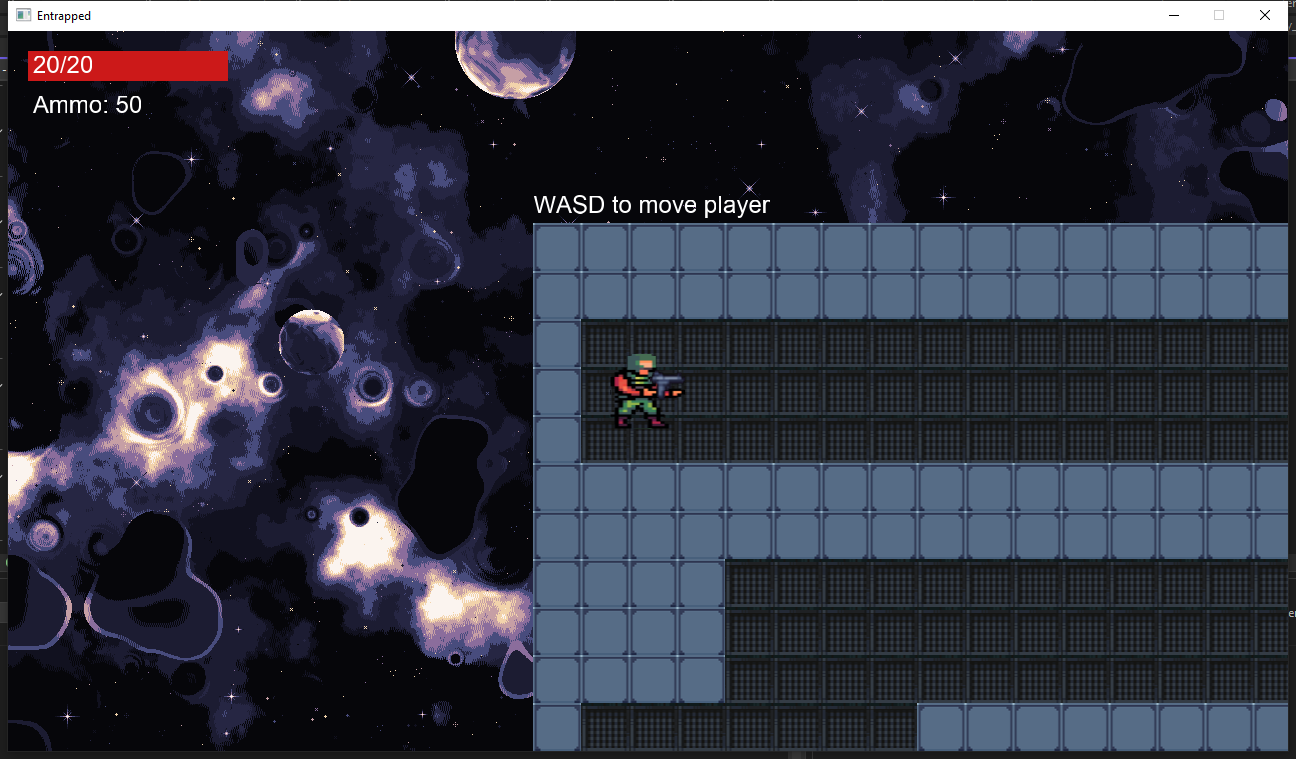
* **Action**: The player's center position is inside of the enemy's detection radius.
* **Outcome**: The enemy will move towards the current location of the player.
* **Action**: The player's center position is outside of the enemy's detection radius for less than a certain period (3 seconds)
* **Outcome**: 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.

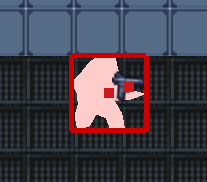


* **Action**: Start the new game.
* **Outcome**: A nebula background will be displayed in the background. It should not move with the camera.
* **Action**: The player's health is reduced to zero.
* **Outcome**: The player will lay down on the ground. The screen will be darkened and transit to the death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

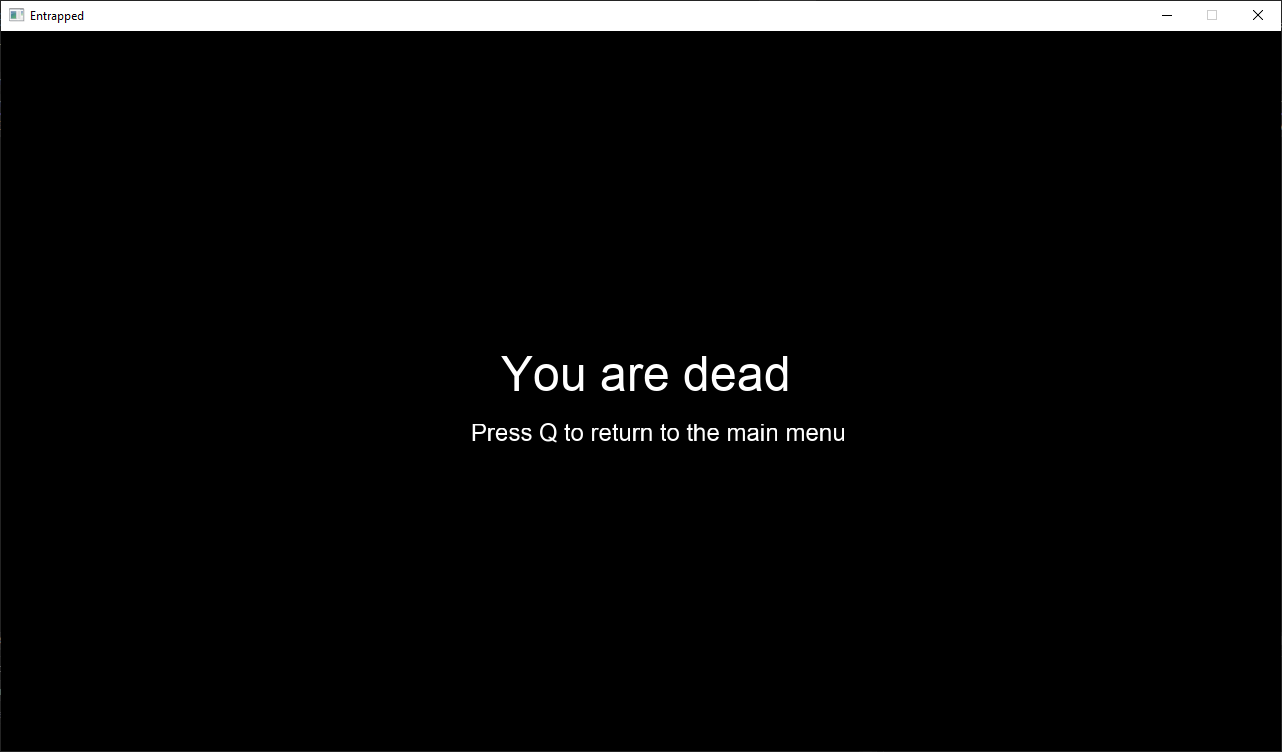
### Audio Feedback

* **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.

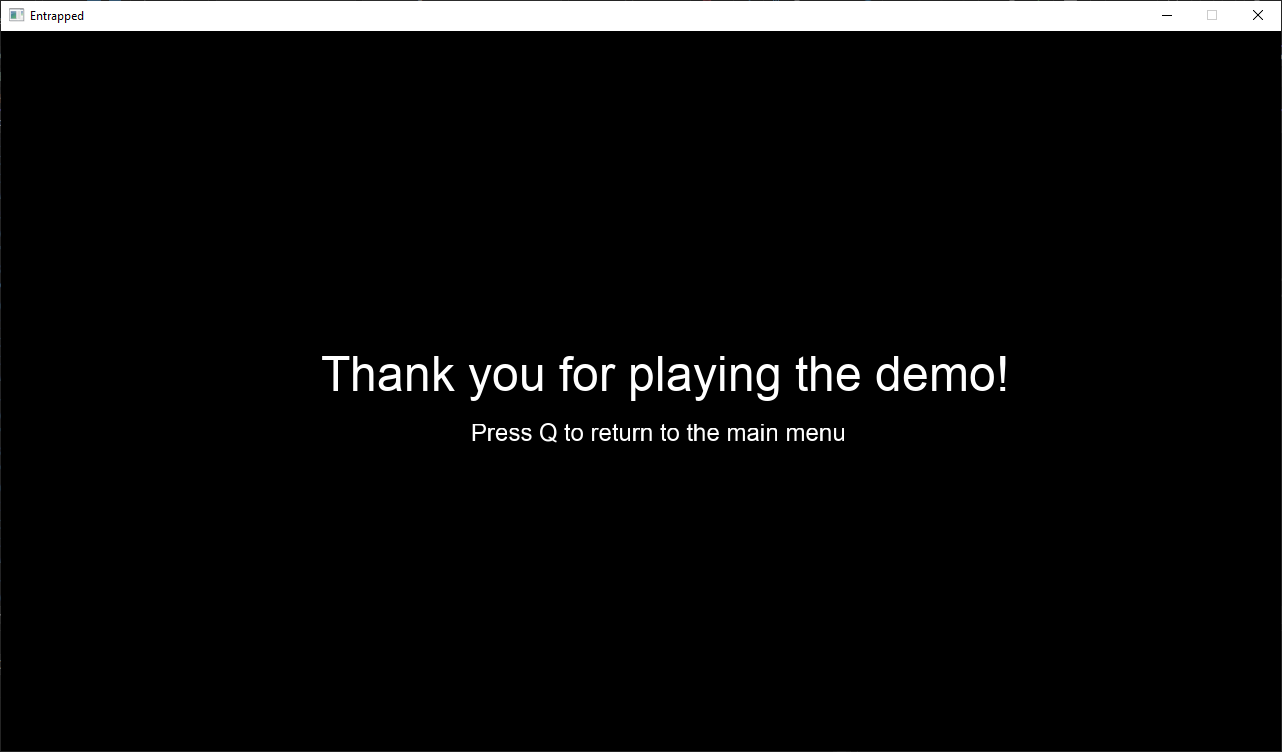
### Camera Matrix

* **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