

CMSI 1010
Shaun Seah
John Chang
Brendan Deasy
Dec, 13 2022

FINAL PROJECT REPORT

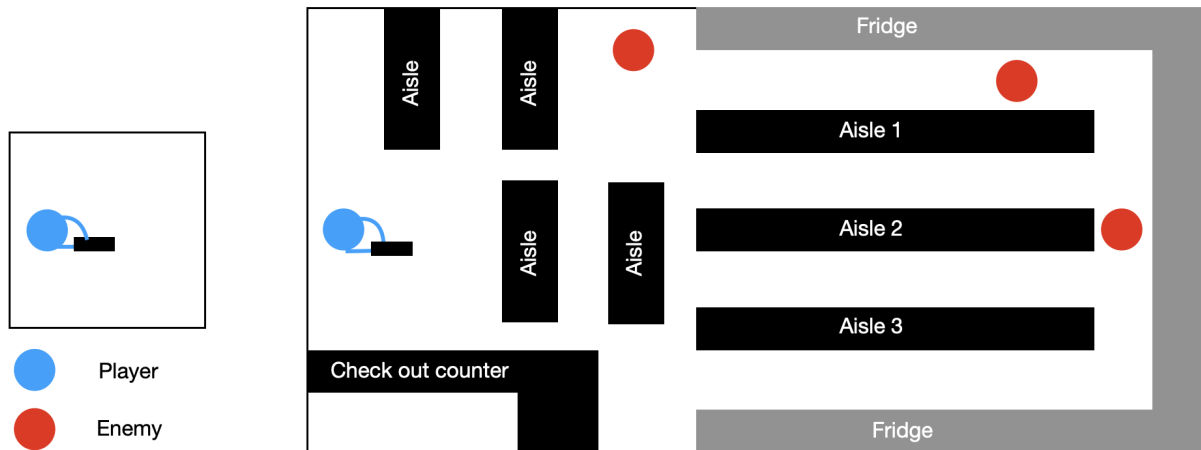
Project Proposal

We plan to create a simple top down shooter game. The idea of the game is for the human player to survive and kill the 3 enemies. The game consists of 1 Player, 3 Enemies. The idea for the map is a convenience store. The game contains a working health and damage system for both players and enemies, and solid boundaries on maps.

Schedule

1. Milestone 1 complete 11/29
 - a. Game runs, player can move, shoot, and look around
 2. Milestone 2 complete 12/6
 - a. Map and boundaries
 - b. Working health and enemies
 3. Milestone 3 (optional) 12/13
 - a. Interactive map
 - b. point system
 - c. Health pickup
- Minimum target
 - Top down shooter, 1 player 3 enemies
 - Map with solid boundaries
 - Player can shoot and die
 - Health system
 - Extensions
 - Multiple gun types
 - Items (health armor etc)
 - Interactive map (doors)
 - Field of view mechanics
 - Enemy ai
 - Arrest mechanics
 - Point system
 - Idea description
 - Milestones
 - Game runs, and player can move, look around and shoot
 - Health system, map boundaries, static enemies
 - Interactive map and point system
 - Item pickups, gun types
 - Enemy ai

Level Design



The first room on the left is the tutorial stage, where users will be walked through the control scheme and how to maneuver, aim and fire their weapon. The second room will be introduced once the tutorial has been completed. The player starts with a handgun and has to survive and eliminate the three enemy AI players on the map. The second map is a small convenience store with thoughtfully placed aisles to introduce different angles and combat scenarios.

Project Milestone 1

To create our game, we started by creating a moveable player in a level. Along with this, we added a mouse-pointer based aiming system for the player to look around. To go with the aim mechanic, we added a projectile which the player can fire in the direction they are looking. To top the player character off, we added a couple of sound mechanics. The first is a yelling command which can be activated by the player, as well as footstep sounds to follow their movement. Next, we dealt with the sound effects of the weapons, adding weapon firing and shell casing sound effects, adding to the playing experience.

We are currently working on an addition which would see a camera always able to track the player from a top-down view. Currently the camera does not perfectly follow the player's movements, but this addition will be a massive step in the direction of completing our project. Because of the accessories and quality of life changes we have already added to the game, we are pleasantly ahead of schedule.

Description of how and why it changed from its original form

- We originally envisioned having the camera center itself on the player. However, it proved to be very challenging and time consuming and we were unable to implement the self centering camera on the player along with other critical game mechanics due to conflicts.
- Instead, we implemented map boundaries on a static map and used the screen edges as the “walls”.
- Due to the issues with merge conflicts and uploading process on Github, we were not able to implement interactive map items (doors), point systems, and item pickups
- We originally hoped to add a large variety of weapons to the game however, we only managed to implement the handgun.
- We were unfortunately unable to implement an interactive map, health system, and points system due to time constraints.
- We originally planned to implement a map with solid objects, multiple levels, and characters with animations, however we were only able to implement the player’s walking animation.
- Enemies were initially intended to walk in intelligent paths around the map. This implementation, however, resulted in enemies losing track of the player’s location and being unable to aim effectively.
- The map was intended to have boundaries for players and bullets, however its implementation caused major bugs with the offsets of enemy and player images with regards to their intended position.
- Enemies were supposed to have “cones of vision” which dictated where they could shoot the player based on their position with respect to the vision. This implementation could not be completed due to bugs resulting from enemies entering one another’s fields of vision

List of which team members did what during the project

- Jonathan
 - Artwork
 - Enemy Sprites, Animations, Map, Death Screen
 - Sourcing Images
 - Map Boundaries
 - Camera Scroll Testing
 - Death Screen
 - Game Demo Video
 - Color Correction
 - Game Design
- Shaun
 - Player Class
 - Player Aiming Mechanic
 - Player Sprite Rotation

- Basic Player Movement
- Shooting Key
- Reload Key & Mechanic
- Yell Key & Mechanic
- All SFX, Soundmanager (weapon, reload, yell, walk, turn SFX), keybinds
- Different sound channels for walk and turn so they don't clash with weapon sounds
- General Game Setup
- Brendan
 - Enemy Class
 - Enemy Bullet Class
 - Enemy Movement (AI)
 - Enemy Spawns
 - Enemy Health
 - Enemy Shooting
 - Player Health
 - Player Death
 - Slideshow presentation
 - Damage screen color change

A description of any difficulties you had and how you overcame them. Also include anything interesting you learned about Python, programming, or working on a team.

- Communication
 - Our team has a group chat on iMessage.
 - Communication and arranging times for group work whether in person or on zoom was challenging.
- Player Sprite Rotation & Shooting Mechanic
 - Implementation of the rotating player sprite in the direction of the mouse pointer.
 - It took a lot of research because we started off using rectangles instead of sprites. Using sprites requires a different coding approach in order for it to work smoothly.
- Scroll Mechanic
 - We tried for a long time to implement the scroll mechanic because it was central to the game design.
 - We got it to work separately, but every time we tried to merge the concept with the main file it would break
 - One possible reason for the issue could be the fact that the shooting angle and mouse cursors and enemies were based on coordinates, but since the camera scroll is a variable and is relative to the movement of the player, the tangent function could have issues working with variables where numbers should have been

- Github committing, pushing and merging
 - Overall working on Github was mainly a “figure out as we went” type issue, given we didn’t have much prior experience, which made it very tough to effectively collaborate through Github
- Drawing and animations
 - Drawing each sprite and coloring them by hand took hours, and sometimes exporting the files would lower quality, which needed to be worked around
 - Sometimes, limbs would take over an hour if not more, depending on mistakes in coloring and shading
 - Each frame for the shooting animations that weren’t implemented due to time constraints was tough due to hand shading, drawing, and adding details
 - Each frame was hand drawn which on average took between 3-7 hours each and each animation was roughly 12-18 frames each
- Enemies
 - Enemy shooting and taking damage was incredibly difficult as the implementation of hitboxes proved troublesome when included with movement
 - This problem was solved by scrapping hitboxes in favor of an xy coordinate radius which would result in damage being taken if a bullet entered the given range
 - Movement was difficult to implement, as enemies would lose nearly all aiming ability once they turned corners
 - This was solved by keeping the enemies moving in straight lines so that the bullets could simply be reversed to still target the player
 - Transitioning the enemies from static rectangles to images made the hitboxes extremely buggy
 - This was once again solved by simply scrapping the hitboxes in favor of a “hitbox radius”

References

 Pygame Top Down Shooter Tutorial #1 - Player Setup
<https://www.youtube.com/watch?v=sVbFS9qEl4Y>

<https://stackoverflow.com/questions/28005641/how-to-add-a-background-image-into-pygame>

<https://stackoverflow.com/questions/72914802/pygame-top-down-shooting>

<https://replit.com/@Rabbid76/PyGame-SpriteRotateToMouse#main.py>

<https://stackoverflow.com/questions/30799522/how-to-run-a-program-in-atom-editor>

<https://stackoverflow.com/questions/57641012/how-to-import-backgrounds-and-images-into-pygame>

<https://stackoverflow.com/questions/20842801/how-to-display-text-in-pygame>

<https://stackoverflow.com/questions/62142988/how-do-i-make-a-start-death-screen-in-pygame>

<https://stackoverflow.com/questions/29640685/how-do-i-detect-collision-in-pygame>

<https://stackoverflow.com/questions/28005641/how-to-add-a-background-image-into-pygame>

<https://stackoverflow.com/questions/8874276/trouble-with-making-background-of-image-transparent-in-python-using-pygame>

<https://stackoverflow.com/questions/69634473/replace-a-rectangle-with-an-image-in-pygame>

<https://stackoverflow.com/questions/74728081/how-do-i-replace-a-rectangle-with-an-image-in-a-class-for-pygame>

<https://stackoverflow.com/questions/42743482/how-to-draw-an-image-in-pygame?newreg=51e0becbe4814c45bcec5cba44556b57>

<https://stackoverflow.com/questions/8767129/how-to-have-an-image-appear-in-python-pygame>

<https://stackoverflow.com/questions/42743482/how-to-draw-an-image-in-pygame>

<https://stackoverflow.com/questions/43267799/typeerror-required-argument-dest-pos-2-not-found>

<https://stackoverflow.com/questions/48304195/what-are-the-u-and-m-file-markers-in-visual-studio-code>

<https://stackoverflow.com/questions/55642923/how-do-i-change-a-sprite-from-a-rectangle-to-an-image>

<https://stackoverflow.com/questions/57641012/how-to-import-backgrounds-and-images-into-pygame>

<https://stackoverflow.com/questions/46877667/how-to-add-a-new-project-to-github-using-vs-code>

<https://stackoverflow.com/questions/19723118/what-type-of-animation-file-should-i-add-to-a-pygame-game>

<https://stackoverflow.com/questions/69634473/replace-a-rectangle-with-an-image-in-pygame>