

shooter!

CMSI 1010 Final Project Proposal

General

Project timeline: November 10 - December 13

Members: Raihana Zahra, Rayane Tarazi, Lauren Campbell

Project Name: "shooter!"

Project Type: platform shooter game

Description

01. Basic Gameplay

shooter! is a 8-bit platform shooter game where your only goal is to stay alive as long as possible while avoiding obstacles in your journey from one side of the level to the other. If the player is shot by the enemy, they lose the game. If the player makes it to the designated finish line at the end of the level, then they win the game.

02. Minimum Target

Our minimum deliverables are as follows:

- move the player using WASD keys
- allow the player to shoot ammo at the enemy
- have the enemies shoot ammo at the player
- include on screen point system
- build a level of the game

03. Possible Extensions

Possible extensions for the game in future iterations, or if there is enough time include:

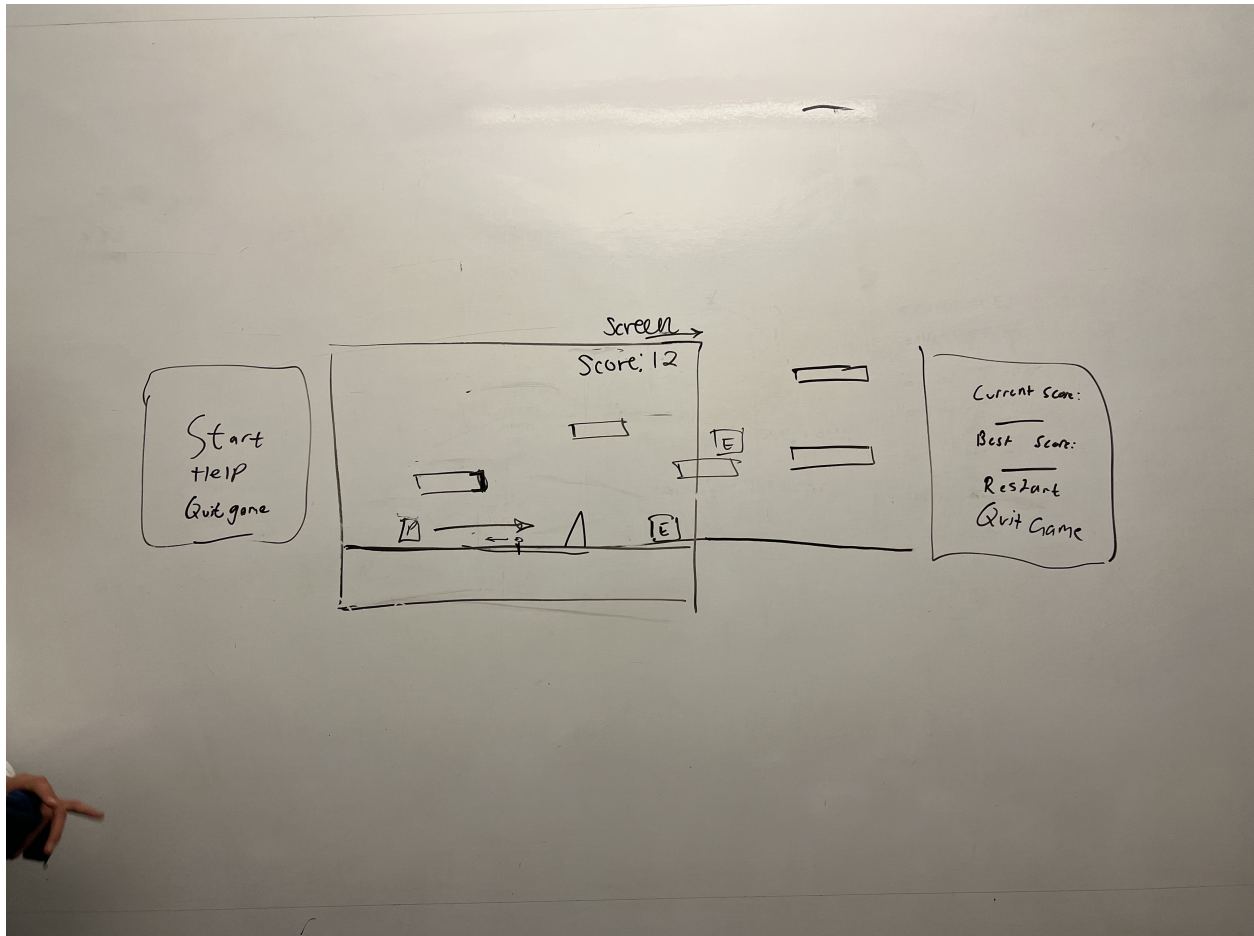
- adding power ups to modify game play (eg. speed multipliers, extra ammunition, boosting strength of ammunition, shield, etc)
- adding a health bar (which would allow for the player to lose health due to hits, but also gain health from a possible health bar throughout the level)
- allowing players to choose their character
- adding different types of enemies (variety in size, hit points, strength, speed, etc)
- adding different types of weapons (eg. a bow and arrow would have different benefits to using a slingshot, a sword would accommodate for short-range combat better, etc)
- adding a boss battle at the end of the level

Project Timeline

Week	Goals	
Week 1 [11/14 → 11/20]	Design screens that the player interacts with (start, end, background)	Design player and enemy sprites
Week 2 [11/21 → 11/27]	Start developing player class, design methods	Start developing enemy class, design methods
Week 3 [11/28 → 12/4]	Implement player and enemy objects in game, design and implement obstacles to hinder player	Work on physics and logic for the player inside the game environment. Sort out collisions and hit boxes
Week 4 [12/5 → 12/11]	Compose score for the game, implement shooting from enemies, create a camera to follow the player	Compile README, demo the game in class, take suggestions to finish debugging and cleaning up the game
Week 4.5 [12/12 → 12/13]	Last minute debugging, tidying up the code files, finalizing commented descriptions	Finalize all code files, documentation, merge all branches, and submit assignment

Interface Mockups

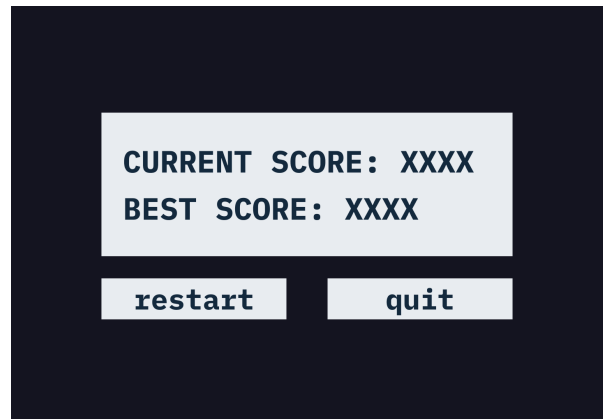
Initial Interface Brainstorm

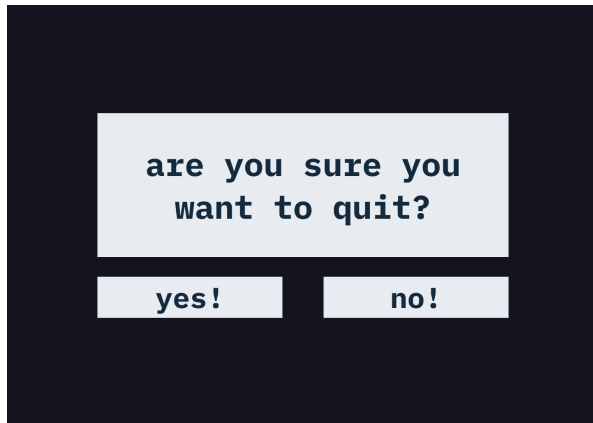


Static Interface Mockups [as of 11/14]

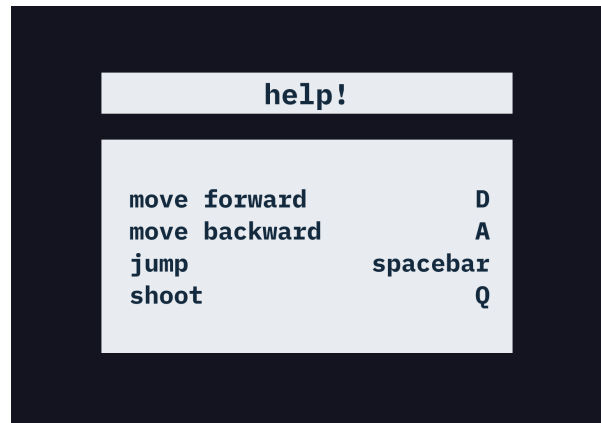


shooter! start screen

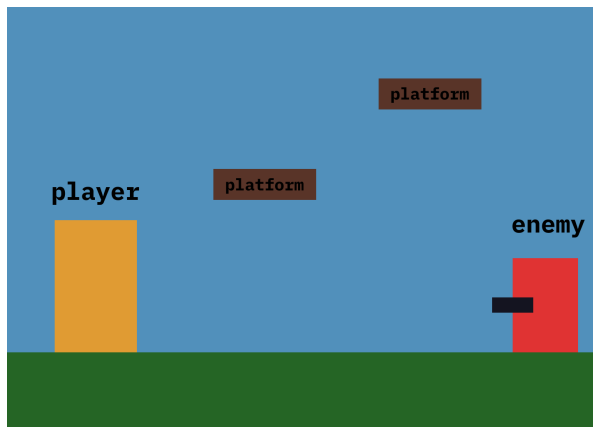




shooter! quit assurance screen



shooter! help screen



shooter! gameplay mockup

Live Interface Mockups [via Figma]

<https://www.figma.com/proto/x89mwasuP3EyW35LHXmka1/CS-Final-Mockup?node-id=3%3A39&scaling=scale-down&page-id=0%3A1&starting-point-node-id=3%3A39>

Classes and Objects

In order to execute our game, we plan on using the following Classes and Objects, as well as including the non-exhaustive list of attributes and methods included:

- Class → Weapon

- Child Class → Gun
- Class → Ammunition (Ammo)
 - Object → Bullet
 - Method → Fire
- Class → Player
 - Attribute → Hitbox
 - Attribute → Movement
- Class → Enemy
 - Attribute → Hitbox
 - Attribute → Movement
 - Attribute → Fire Rate
- Class → Obstacles
 - Object → Spike
 - Object → Dirt

The relationships between these classes are as follows:

- Player has-a Weapon
- Enemy has-a Weapon
- Gun is-a Weapon
- Spike is-a Obstacle
- Dirt is-a Obstacle

Libraries and Execution

After careful consideration, we have decided to use the `pygame` library, and the built in Python graphics library (`graphics.py`). We will also use some of the built-in Python modules, including `random` and `math`.

The game will be built in Visual Studio Code, and can be played in a pop-up window after being run from the Terminal