COSC 499 Project Plan

Project Proposal for PROJECT OPTION 4

Team Number: 14

Team Members: Gabriel Mercier 48255368, Jesse Lazzari 83173385, Darion Pescada 87753638, Justin McKendry 90213778, Kibele Sebnem Yildirim 11236312

1 Overview

Our minimum viable product is a 2D top-down shooter game designed for 1-2 players in local multiplayer mode, where they team up to battle against AI-controlled bots. Players will progress through 10 levels, each level will have increasingly challenging terrain and enemies. For gameplay progression, players must defeat all bots to move on to the next level. The core gameplay mechanics will include being able to shoot bullets that bounce off of the level's walls/obstacles, and using the level's terrain to the player's advantage. The primary purpose of this software is to provide an engaging and challenging gaming experience that promotes cooperation and strategy.

What sets our solution apart is its combination of fixed camera 2D gameplay, progressively challenging levels, a wide range of player stats (speed, health, attack speed) and a variety of enemy attributes. Players can customize their gameplay experience by selecting between 3 different difficulties: easy, medium, and hard (gameplay difficulty changes may involve increasing enemy health, enemy movement speed, enemy projectile velocity, and enemy fire rate). Since the game is local multiplayer vs. bots we will also implement controller support.

Our MVP aims to deliver an entertaining local multiplayer experience, offering 10 levels of increasing difficulty, unique gameplay mechanics, and dynamic enemy AI. It aims to promote teamwork and strategic thinking, since players can strategize the best way to eliminate all enemies for a given level, with the goal of completing the 10 levels as fast as possible.

Minimum Viable Product Features:

- 2D top down shooter with a fixed camera
- 1-2 players local multiplayer vs bots
- Gameplay progression, players must defeat all bots to move on to the next level
- 10 levels, each level has progressively harder obstacles and more difficult enemies
- Main menu
- Difficulty selection menu
- 1 or 2 player selection menu
- Pause menu
- Options menu
- Game over menu
- Win menu
- In game UI (displaying player health, points, current level, etc.)
- Level template (an empty level that will be used as a template to easily create other levels)
- Level creation (10 unique levels the progressively get more difficult)
- Level transitions (moving players from one level to another)
- Interactable objects (breakable walls, moving walls, doorway to the next level, etc.)
- Player controls (getting user inputs from keyboard, mouse, and controllers)

- Player movement
- Player attacks
- Player bullets
- Player collisions with other objects
- Coop
- Multiple enemy types (each item in the sublist may work differently for each enemy type)
    - Enemy movement
    - Enemy attacks
    - Enemy bullets
    - Enemy collisions with other objects
    - Enemy targeting (enemies need to choose which player to attack if playing coop)
- Boss battle on final level
- Upgrade system (a way for players to acquire upgrades)
- Upgrades (each individual upgrade)
- Respawn system for coop (a dead player should respawn when other player enters next level)
- 3 difficulties: Easy, medium, and hard (increases enemy health, projectile velocity, etc.)
- Game controller support (required for coop)
- Music
- Sound effects
- Simple animations (walking, attacking, and dying)
- Difficulty balancing (making adjustments to ensure a challenging but fair experience)

Stretch Goal Features:

- Particle effects (trails, explosions, blood splatter, etc.)
- Character selection (each with different abilities, stats, and aesthetics)
- Character selection menu
- Special abilities (dash ability, bomb ability, shield ability)
- UI for abilities (visual feedback so the player knows if they can use their ability or not)
- Additional enemy types
- Additional levels (10 might not be enough)
- Additional upgrades
- Endless survival game mode
- Game mode selection menu
- Online leaderboards for how fast you can complete the game
- Leaderboard menu
- Save system (allows for player to quit and then continue playing from where they left off)
- Other effects to improve the games quality (screen shake, post processing, menu transitions)
- Friendly AI (a bot that fights alongside you instead of against you)
- Cut scene at end of game
- Tutorial

1.1 Envisioned Usage

1. Single User Scenario (Carla) :

   Carla decides she wants to try the MVP shooter game. She's into shooter games and is eager to test her skills over increasingly challenging levels. She launches the game from her computer and is prompted by

an intriguing menu screen. She clicks on the 1-Player button and is presented with three difficulty options: Easy, Medium, and Hard. She chooses “Medium” because she knows she has some experience in gaming but wants a balanced challenge. The first level starts in an arena style map and with an opponent across from her. Carla familarizes herself with the controls using WASD keys and her mouse for aiming and shooting. She quickly completes the first level and the game allows her to move on to the next level. As she continues to progress through the levels, she notices the obstacles in the map are getting more tricky, there are more enemies, and they are getting more aggressive. She has to be careful as she only has three hearts and she doesn't want to have to start over. After finally completing each unique level as well as the Final Boss she is presented with a victory screen displaying her time and score. She is excited from the win and decides that on her next game she’ll try playing the game on “Hard” difficulty to test her new found skills.

2. Two-User Scenario (Sean and Chris):

   Two friends, Sean and Chris, boot up the game for the first time choosing 2-player and Hard difficulty as they feel confident in their abilities and would like a challenge. They plug in their game controllers to their computer and quickly familiarize themselves with the joystick controls on the first level. After a few levels, Chris needs the bathroom so they pause the game to come back to it a bit later. After coming back to the game they find themselves struggling and lose both of their three hearts. They are presented a Game Over screen and decide they should try an easier difficulty on their next attempt.

2 Major Milestones

Deadline	Deliverable
Term 1 week 9: Mini Presentation Oct 31-Nov 2	For the first milestone, we plan to have a basic implementation of the player character. This includes: ● Player controls (getting user inputs) ● Controller support ● Player movement ● Player attacks ● Player bullets ● Player collisions with other objects ● Player animations (walking, attacking, dying) Additionally, we will create: ● The level template that will be used in the future for creating other levels ● The main menu ● The pause menu Many of these features will be improved later on as we implement other features. At the end of this milestone, we will have a simple prototype where you can start a game from the main menu, move around as the player in an empty room, shoot bullets, and pause the game.
Term 1 week 13: Design submission	In this milestone, we will implement the first enemy. This includes: ● Enemy movement ● Enemy attacks ● Enemy bullets

	● Enemy collisions with other objects ● Enemy targeting (enemies need to choose which player to attack if playing coop) ● Enemy animations (walking, attacking, dying) To allow player and enemy interactions, we will add collisions between: ● Player bullets and enemies, allowing players to shoot and kill enemies ● Enemy bullets and players, allowing enemies to shoot and kill players ● Enemies and players, causing players to take damage when colliding with enemies We will also implement: ● 2 player coop ● An option in the main menu to select 1 or 2 players ● The in game UI to display player health and other important information ● The game over menu At the end of this milestone, 2 players can play together and interact with the enemy AI. They will be able to shoot bullets at enemies to destroy them, and the enemy will be able to do the same to the players. Additionally, visual feedback from the in game UI will inform the players of important information and the game over menu will appear when no players are left alive.
Term 2 week 4: Peer Testing	In this milestone, we will be adding 2 new enemy types. The creation of each additional enemy includes multiple features which are stated above in the previous milestone features list. To prepare for peer testing, we will include the following: ● A small set of test levels ● A transition system to move players from one level to the next ● Interactable objects to improve the diversity of each level. This includes: ○ Breakable walls ○ Moving walls ○ Doorways that takes the players to the next level ● A win screen that displays after the players complete all levels Additionally, the game will have music, and an options menu to change the volume. At the end of this milestone, our game will feel more like a real game, rather than a simple prototype, with various enemy types, and a few interactive levels to play through. There will also be music to improve the games immersion and an options menu to change the volume.
Term 2 week 8: Peer Testing	In this milestone, our main focus will be creating 6 different upgrades as well as a system that allows players to select an upgrade at the end of each level. We will also add: ● 3 difficulty levels ● A difficulty selection menu ● A respawn system for coop ● Sound effects At the end of this milestone, there will be a fully implemented upgrade system allowing players to choose different upgrades to increase their strength and allow for different play styles. We will have a working

	respawn system so if a player dies, they will respawn once the other player completes the current level. The game will include 3 difficulty levels, as well as sound effects to further improve the games immersion.
Term 2 week 13: Final project submission	In this milestone, the majority of our time will be spent polishing the game, fixing minor issues, and balancing the difficulties. New features that will be implemented include: ● Creating all of the levels ● A boss battle on the final level ● Other additional features, depending on the time remaining At the end of this milestone, the game will be complete with a full set of levels, and a boss battle on the final level. The game will be polished and balanced for an enjoyable yet challenging experience.

3 Technology Stack

The user is going to be able to play the game on either a web browser or downloadable option. The user might be able to access it on mobile devices. We will be using vs code as our IDE to develop the game in c# while using UNITY's engine. For unit testing we will be using the UNITY testing framework (industry standard). If we implement the leaderboard and saving features we would be using the LootLocker API.

Our project (which I will refer to as the game from now on) will be primarily developed for computers. However, we may add compatibility for mobile devices later in the development of the game. We believe this is best as splitting efforts between the development for multiple devices in the beginning could lead to a lower overall quality of the game but a port of the full game once completed could lead to two great products. The game will be downloadable so it can be played even without wifi, which allows the user more freedom with where they want to play.

Our team plans on using VS code as our IDE for the development of the game. We choose VS code because it is one of the most user friendly IDE's, it has endless extensions that can help with development and it is one of the most widely used IDE's which means that if any problems occur with the IDE itself then there should be easily found solutions. Along with that, it is seamlessly used with Github for version control which will be helpful for the management of the development of our game.

The game will be made in C#. We choose this because we are using Unity's engine to build the game and C# is supported by Unity. We are using Unity's engine to build the game because it is extremely user friendly, uses C# which is similar to Java which we all have experience in and unlike Unreal engine it focuses more on the game opposed to graphics. It also has many free assets which can allow us to spend less time designing graphics and more time on the game itself. Along with that it is generally known as the best for new independent developers.

The unit testing done for our game will be made using Unity Test Framework. This is a package developed by Unity to allow for unit testing on our game. It uses a Unity integration of the NUnit library. NUnit library is a library that is made for unit testing on .Net languages. Along with that we may use the Test Runner API which allows the developer to run tests from any script. It also allows the developer to get a list of tests that will run in edit mode, play mode or both all while not running any actual script. This all is some of the most up to date software for Unity that allows the developer to unit test.

Finally, depending on which additional features we end up developing we may use an API called loot locker. This API is used as a backend platform and functions similarly to a database. It can save and update

leaderboards, individual progression, and accounts. Along with that it is fairly simple to use and is widely used by independent game developers. This means that not only will it not slow down the development of the game but it should also be fairly easy to get advice and work towards solutions allowing us to stay on track.

4 Teamwork Distribution and Anticipated Hurdles

Category	Jesse Lazzari	Darion Pescada	Justin McKendry	Kibele Sebnem Yildirim	Gabriel Mercier
Experience	COSC 304: Basic e-commerce website developed using HTML, Java, and SQL. COSC 310: Uber clone developed in Android Studio, following Agile methodology and branching workflow COSC 341: Local mobile application similar to iClicker	-Game development using Unity and C# -COSC 304: simple website using Java, SQL, and HTML -COSC 341 and 310: basic mobile applications using Android Studio -Additional group projects using GitHub and a branching workflow	COSC 341: Mobile app development using Java that focused heavily on front end development. COSC 310: Developed a library app using an agile development style and branching workflow. COSC 304: Simple e-commerce website developed using HTML, JAVA and SQL COSC 322: Developed an AI using an agile development style and branching workflow Managerial skills from being a crew lead at an industrial flooring company.	COSC341: simple mobile app development using Javascript, css, html COSC310: managing and finalizing project, unit testing, kanban and scrum COSC360 :Web Development, dynamic website building COSC304: SQL ddl database development using Docker	-COSC 341: Created a mobile app as a group of four students in Android Studio. -COSC 360: Created a forum, Reddit-like, website (front and backend) using PHP, HTML,CSS, JavaScript, and SQL. -Brief experience using Radiant game engine to create custom game levels. -Github used in various projects
Good At	- Project planning - Video Editing - Task organization - Java - Sql - Basic Html - Project management - Public speaking - Presentations - Data Structures - Teamwork - Coordination - Communication - Schedule flexibility	-Coding -Java -C# -Python -SQL -JUnit -Debugging -Data structures -GitHub -Unity -Game development -Project planning -Time management	- Team planning and coordination - Managerial skills - Communication - Java - Python - Basic HTML - SQL - Basic C - Data Structures - Presentations - Public speaking - Time management - Project planning	- Documentation - Presentation - Project planning - communication - Unit testing with JUnit - Java - Html - Css - Javascript - Sql ddl - Relational - Research and development - PHP - R - Data structures	- Public speaking -Communication -Working as a team -Presentations -Delegating -Coding -Java -SQL -HTML/CSS -GitHub -Figma

Expect to Learn	- Unity - Game development - Frameworks - Debugging - Coding within Unity - Unit Testing	-Team based game development -Unity testing framework -Test driven development -Improve my overall game development skills	- Unity - Game development - More detailed unit testing - Frameworks - C# - Additional workflow tools for organization.	- CICD - Unity - Game development - Frameworks - Debugging - Coding - communication	- Unity -Game design -Better unit testing -Working on one goal at a time/ breaking down work -logging work

Category of Work/Features	**Jesse Lazzari**	**Darion Pescada**	**Justin McKendry**	**Kibele Sebnem Yildirim**	**Gabriel Mercier**
Project Management: Trello	✓	✓	✓	✓	✓
Technical Direction: Time Estimation, Team Logs	✓	✓	✓	✓	✓
Technical Help: Finding Techni-cal Solutions	✓	✓			
Troubleshooting: The Go-To When Others Are Stuck		✓			✓
System Architecture Design (Repo Setup)		✓		✓	
Player controls					✓
Controller support		✓			
Player movement					✓
Player attacks		✓			
Player bullets			✓		
Player collisions with other objects			✓		
Player animations	✓				
Level template	✓				
Create main menu				✓	
Create pause menu				✓	
Enemy 1 movement		✓			
Enemy 1 attacks					✓
Enemy 1 bullets			✓		
Enemy 1 collisions			✓		
Enemy 1 targeting		✓			
Enemy 1 animations	✓				
Collisions between players and enemies	✓				
Collisions between player bullets and enemies	✓				
Collisions between enemy bullets and players	✓				
Coop		✓			

1 or 2 player selection menu				✓	
In game UI					✓
Game over menu				✓	
Enemy 2 movement		✓			
Enemy 2 attacks					✓
Enemy 2 bullets			✓		
Enemy 2 collisions			✓		
Enemy 2 targeting		✓			
Enemy 2 animations	✓				
Enemy 3 movement		✓			
Enemy 3 attacks					✓
Enemy 3 bullets			✓		
Enemy 3 collisions			✓		
Enemy 3 targeting		✓			
Enemy 3 animations	✓				
Create test levels	✓				
Transition system		✓			
Interactable objects					✓
Win menu				✓	
Options menu				✓	
Music					✓
Sound effects					✓
Upgrade 1					✓
Upgrade 2			✓		
Upgrade 3			✓		
Upgrade 4	✓				
Upgrade 5		✓			
Upgrade 6		✓			
Upgrade system		✓			
Difficulty levels				✓	
Difficulty menu				✓	
Respawn system	✓				
Level creation	✓	✓	✓	✓	✓
Boss battle	✓	✓	✓	✓	✓
Difficulty balancing		✓			
Additional bug fixing and improvements	✓	✓	✓	✓	✓
Presentation Preparation	✓	✓	✓	✓	✓
Design Video Creation	✓				
Design Video Editing					✓
Design Report				✓	
Final Video Creation	✓				✓
Final Video Editing		✓		✓	✓
Final Team Report		✓	✓	✓	
Final Individual Report	✓	✓	✓	✓	✓