## **Capstone Project Proposal**

**General Information**

Project name: Hydra Tanks

Author: Jamie Lewis

Project Organization: BeachBlock Games

Project Manager: Jamie Lewis

Date: March 17, 2024

**Project Overview and Project Objectives**

**State the Problem**

My studies in Software Development have led me toward a specific focus on Game Development. The point of most games, this one being no exception, is to be fun and engaging. I am working toward game development as a career, so this will be an exercise toward that end. I want to make Hydra Tanks a good gaming experience. That being said, I will not make games that I would not want my own kids to play, and as such, the games I produce must conform to a Christian worldview.

**Background**

Hydra Tanks started as a very small game I developed for a 10-day game jam back in May of 2022. At the time of its initial release, I had thought about building it out into a more full-featured title. This senior project seems like a good time to do that. The current state of the game contains the core gameplay and basic mechanics. The object of the game is to survive an ever-growing enemy onslaught for as long as possible while killing enemy tanks, collecting power-ups, and using tactical maneuvers.

It can be debated whether or not a tank combat game could fit into the Christian worldview. My perspective is that it is a game no different than playing chess. In fact, military strategy is even mentioned in parts of the Old Testament scriptures. As a developer/publisher, I can make games from a Christian worldview in intentionally including or not including certain content. I can refrain from including things like foul language, drug references, sexual references and/or interactions, or allowing players to commit other heinous acts. I can promote a Christian worldview by rewarding good moral behavior, incorporating Bible concepts and teachings into storylines, and keeping language clean. Hydra Tanks does not lend itself to adding things to the game which demonstrate a Christian worldview, but it will exclude that which the Bible says is offensive to God.

**Project Objectives**

List objectives that will be used to measure project success.

* The project will be completed on time.
* The game will run without crashing.
* The list of required features will be completed.

**Challenges**

List the known challenges that will be used to measure project success.

* There is a limited timeline in which to complete the project.
* Since I have no financial budget for this project, I will be using free and already owned art and audio assets to complete the game project. I will have to find appropriate assets for each feature I want to implement into the game.

**Benefits and Opportunities**

Successful completion of this project will provide me with positive experience in support of my goal of becoming an independent game developer. It will also benefit players of my game by providing entertainment value. The game’s genre is also an exercise in tactics and real-time strategy.

Opportunities begin with the potential profit from publishing. However, even if the game is not profitable, there is potential for player feedback so that the next game can be improved. The timeline for this project does not allow for a lot of player input or the potential to grow an audience during development. However, any future players of this game could become that audience of early players for the next game.

**Project Scope**

The scope of the project involves developing the original iteration of Hydra Tanks into a fully featured game. This will include a full menu system, including Main, Pause, Settings, and Level Select. There should be at least 5 levels of gameplay to choose from. These levels should also be bigger than a single screen for the player to roam around in, where the single existing level is the size of the screen. There are currently 3 power-ups that players can collect while playing a level. That should be expanded to include as many as 8 power-ups, including variations of existing ones. The game’s current state only supports keyboard and mouse, so controller support should be added. The final minimum requirement should be a save system for high scores.

An optional nice-to-have feature that could be included, if development goes smoothly and time allows, is a tank upgrade and customization system. This would include adding a menu item for the player to build out their tank, as well as adding this customization to the save system.

Out-of-scope features would include any kind of multi-player support. There simply is not enough time to implement that within the timeframe that this project must be completed. This means local or online and cooperative or competitive multi-player game modes. Another out-of-scope feature would be online leaderboards for the high scores. These features could be added in a future version of the game beyond this class project.

|  |  |  |
| --- | --- | --- |
| Stakeholder Name | Role(s) | Responsibilities |
| Jamie Lewis | Project Owner | Design, Development, Testing, & Documentation |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Work Breakdown Structure | | | | | | | | | | |
| ID | Task | Dependencies | Status | Effort Hours | Cost | Start Date | Planned Completion | Estimate to Completion | Actual Completion | Resource |
| 1 | Project Proposal Draft 1 |  | Complete |  | N/A | 1/29/24 | 2/4/24 |  | 2/4/24 |  |
| 2 | Requirements Analysis |  | Complete |  | N/A | 2/5/24 | 2/11/24 |  | 2/11/24 |  |
| 3 | Architectural Plan |  | Complete |  | N/A | 2/12/24 | 2/18/24 |  | 2/18/24 |  |
| 4 | Development Iteration 1 |  | Complete |  | N/A | 2/19/24 | 2/25/24 |  | 2/25/24 |  |
| 5 | Development Sprint 2 |  | Pending |  | N/A | 2/26/24 | 3/3/24 |  |  |  |

**Project Completion**

My goal in this project is to complete a functional, bug-free game within scope and timeline.

|  |
| --- |
| Project Completion Criteria |
| 1. The project will be completed on time. |
| 2. The game will run without crashing. |
| 3. The list of required features will be complete and bug-free. |
| 4. All project documentation will be current and complete. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Assumptions and Constraints | | | | | |
| ID | Description | Comments | Type | Status | Date Entered |
| 1 | I have enough experience using Unity to have determined how much I can complete in the allotted timeframe for this class. | I have been working with Unity for 2 to 3 years in my spare time and completed a few projects. The game for this project is being built off of one of those. | Assumption |  | 2/3/24 |
| 2 | I could get derailed by wanting to add more features than time allows. | I tend to be overly ambitious and get out of scope. I need to stay focused on established development plan. | Constraint |  | 2/3/24 |

**Project Controls**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk Management | | | | |
|  | **Risk Probability** | **Risk Impact** |  |  |
| **Event Risk** | **(high, medium, low)** | **Risk Mitigation** | **Contingency Plan** |
| Life events could impact the time available to put into development work | Low | I may have to work around some schedule interruptions. I am usually very flexible about these things though. | Only respond to issues that really matter and don’t make excuses to put off work on the project. | I will try to get things done as early in the week as possible so that I am not crunched to finish the week’s tasks on the weekend. |
| I may have difficulty choosing assets to bring the game design to life in the way I envision it. | Medium | The game may not look like I really want it to, or I may waste too much time looking for art/audio assets. | I need to start looking for these assets early on, before the first week of development. | I can be flexible in the desired outcome of the look and feel of the game. |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Change Control Log | | | | | | | | | |
| **ID** | **Change Description** | **Priority** | **Originator** | **Date Entered** | **Date Assigned** | **Evaluator** | **Status** | **Date of Decision** | **Included in Rev. #** |
| 1 |  |  |  |  |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Roles and Responsibilities | | | |
| Name | Team | Project Role | Responsibility |
| Jamie Lewis | Solo | Owner | Design, Development, Testing, & Documentation |

**Project Schedule**

Project tasks have been outlined, prioritized, and organized into sprints in the Sprint Backlog Excel document. There is a tab for each sprint.

**Cost Estimate (if applicable)**

There are no planned costs associated with the development of this project. I will only be using free or already owned resources to complete it. Credits are in Appendix A.

**Issue Log**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Issues Log | | | | | | | | |
| **ID** | **Issue Description** | **Project Impact** | **Action Plan/Resolution** | **Owner** | **Importance** | **Date Entered** | **Date to Review** | **Date Resolved** |
| 1 | Did not complete a task to the desired level. Adjusted Sprints accordingly. | Minor | Complete task in next Sprint | Jamie Lewis | *Moderate* | *3/3/24* | *3/10/24* | *3/10/24* |
| 2 | System Logical Model (page 10) and Appendix B (Credits) need brought up to date. | Minor | Complete task in next Sprint | Jamie Lewis | *Low* | *3/3/24* | *3/17/24* | *3/17/24* |
| 3 | Requirements & Design Documentation need reworked to provide better communication. | Moderate | All documentation will be reviewed in detail and reworked as needed to better reflect the intended outcome of the project. | Jamie Lewis | *Major* | *3/10/24* | *3/17/24* | *3/17/24* |
| 4 | Test Cases & Traceability Matrix need to be completed | Moderate | This will get done in the coming week! | Jamie Lewis | *Major* | *3/17/24* | *3/24/24* | *3/24/24* |
| 5 | Test Cases are still incomplete. Traceability Matrix would need updated once that was complete. | Moderate | There is not enough time left in this class to complete these. However, the workflow between documents has been refined to demonstrate an understanding of the process. | Jamie Lewis | *Major* | *3/24/24* | *N/A* | *N/A* |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

**Integrated Instructor Feedback into Project Documentation**

Yes  No

**Project Approval**

Instructor Donna Jackson (CST-451 – Topic 1: No feedback provided other than grading.)

**Projects Requirements Document**

**Hydra Tanks**

**Use Cases**

See the Sprint Backlog (Excel document).

**System Design**

Top-down system design flowchart.

A diagram of a game

Description automatically generated

Flowchart of the game’s menu system.

A diagram of a computer program

Description automatically generated

**Technical Requirements**

Unity Game Engine

Unity is more than just a game engine, but it is a very popular choice for indie game developers. There is a massive community around it with a lot of resources available for learning to use it. It comes with a lot of built in tools for speeding up game development. For example, Hydra Tanks is a 2D top-down style game. Unity has a Tile Map editor that aids in fast and easy designing of tile-based environments for game levels. Its code library includes the GameObject and Monobehaviour classes, which come with a lot of built-in functionality. Methods include things like Start and Update. Start is called once when a GameObject is instantiated, and Update is called every frame.

C# & Visual Studio

Unity also supports C# as the scripting language. This means using Visual Studio is one of the best options for an IDE. These are tools I have become familiar with, both in classes at GCU, and while using Unity over the past couple of years.

Audio & Visual Assets

Interactive games are very dependent on an immersive audio and visual experience. Since I am not a musician or a graphical artist, I can use pre-made assets to make games. This is common practice for independent developers, and particularly for solo developers. There are many freely available assets online, particularly in the Unity Asset Store, although that is not where I got the assets that I have in the existing project.

A\* Pathfinding Project

The enemies in my game need a way to track down the player in a level. I have written out an A\* pathfinding algorithm in a previous tutorial project which I could use for Hydra Tanks as well, but the A\* Pathfinding Project (see Appendix B) has a free version that I can use which also includes additional functionality already built in that will allow for finer control and faster setup of the enemy AI. This is what is already in use in the existing version of Hydra Tanks.

**System Logical Model**

Block Diagram of system components and their relationships and interactions. Details about each component can be found in the Final Architectural Design under Detailed Solution Architecture.

A screenshot of a computer

Description automatically generated

**Reports**

The closest thing to a report that will be provided by Hydra Tanks is a local record of high scores for each level of the game. These scores will be saved using a simple text file and there will be a “High Scores” button on the main menu which will open a screen to display the scores for each level. See the “High Scores Menu” wireframe, above, for reference.

**Screen Definitions and Layouts**

Main Menu

A screenshot of a video game

Description automatically generated

Level Select Screen

A screenshot of a game

Description automatically generated

Settings Menu

A screenshot of a computer

Description automatically generated

High Scores Menu

A screenshot of a computer screen

Description automatically generated

In-game Overlay

A screenshot of a computer

Description automatically generated

Pause Menu

A screenshot of a computer

Description automatically generated

**Security**

There are no security issues for this game. For the scope of this project, it will be a single-player, offline experience. There will be no logging into any services of any kind.

**Other (as dictated by the context and scope of the project)**

N/A

## **Final Plan**

Hydra Tanks

Prepared by Jamie Lewis

|  |
| --- |
| Design Planning Summary |

Game development is a challenging field. I’ve heard it said that it is one of the hardest development fields to break into. In the early years of game development, a company had to make a final version of a game for their specific hardware, copy it onto physical media, and sell those copies at physical retail locations. With widely available development tools and marketing platforms, any developer can make a game and release it onto the internet. However, it is still a very real challenge to get your game discovered because of the massive number of games that are available to take up players’ time. As an independent developer, or “indie dev”, it is important to just keep making games and putting them out there. A would-be game developer could also apply for a job with an existing company, but that is still a challenge. There are only so many jobs available in that field, and it is not exactly stable. In fact, there have been thousands of layoffs within some major game companies in recent months (Parrish, 2023). But, with either path that a game developer chooses to attempt, they absolutely must have a portfolio of games, both for a resume in case they need to find a job, and/or for marketing their own products.

I started publishing small game projects on a known indie dev platform called Itch (http://www.itch.io) back in 2022. My last title(s), “Puzzle Cube” and “Puzzle Cube Mobile” were also published on Steam and Google Play, respectively. One of the earlier games I posted on Itch was called “Hydra Tanks”. It was made in 10 days for a game jam, an event where indie devs are given a set timeframe in which to make a game, usually around a given theme. The theme for Hydra Tanks was “Death is only the beginning”. In Hydra Tanks, a player starts out against a single enemy, and then two more spawn in for every one destroyed. The goal is to survive as long as possible and receive a score for the number of enemies destroyed.

Hydra Tanks gameplay screenshot

A video game map of a video game

Description automatically generated

This project takes Hydra Tanks to a more feature-complete state, with the features already mentioned in this proposal. The goal is to add another more polished title to my resume, and to potentially market this one for profit on Steam. There are several technical and design concepts to consider for making a game.

By creating a full-featured and polished game to add to my portfolio, I stand the best chance of my games gaining popularity in the market or standing out to a potential employer. There are many more features that could be added to any game, but by focusing on making a smaller, better quality game at first and then expanding on it later, I can add more games to my repertoire faster.

|  |
| --- |
| Overview of Design Concepts |

Development Environment

For this project, I will be using the Unity Game Engine editor version 2022.3.19f1, which is the most current LTS version. The engine is written in C++ but supports game scripts being written in C#, which I have become very familiar with over the past few years. Visual Studio 2022 will be my code editor. The code will be refactored significantly from the original project since this project involves a much longer development cycle than the 10-day game jam it was originally created for.

Core Game Loop

It is imperative that the core game loop is a fun and engaging experience, or nobody will want to play the game. The gameplay is designed in a kind of retro arcade style. I have already built the core game mechanics, including:

• Basic menu allows player to start a new game or pause a game in progress.

• Moving, aiming, and shooting the tank.

• Enemy tank movement, aiming, and shooting.

• Player can pick up power-ups, giving them various effects such as:

o Increase max hit points (health),

o Repair health,

o Increase fire rate.

• Player death ends the level.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Deliverable Acceptance Log | | | | | |
| ID | Deliverable Description | Comments | Evaluator (internal or external as applicable) | Status | Date of Decision |
| 1 | Final Proposal, Requirements, & Design Documentation |  |  |  |  |
| 2 | Initial Development Build |  |  |  |  |
| 3 | Sprint 2 |  |  |  |  |
| 4 | Sprint 3 |  |  |  |  |
| 5 | Final Project Presentation |  |  |  |  |

|  |
| --- |
| Detailed Solution Architecture |

This list of design features aligns with the major components included in the System Logical Model (page 10, colorized items in the diagram).

**Main Scene**

1. Menu System

Controls showing/hiding various UI panels and loading/unloading levels. Menu objects include:

* Level Select – handles selecting a level to load,
* Settings – handles audio settings which auto-save when going back to the previous menu,
* How To Play – includes buttons to access sections for Controls, Objectives, and Upgrades,
* High Scores – displays the top score saved for each level,
* Credits – displays a list of citations for assets used in development of the game,
* Pause Menu – pause/un-pause the game and access to Settings,
* Game Over screen – allows level restart or unloading the current level and returning to the Main Menu,
* Quit Button - exits the game.

1. Audio Settings

Handles settings for master, music, and sound effects volume levels. Handles music track selection. Settings are autosaved to a simple text file via the Save System.

1. Save System

Handles reading and writing settings and high scores to save files. This system also uses settings and high scores data models and a data service to handle each data model.

**Level Scenes**

A Level is a Unity Scene asset consisting of its own Level Manager, A\* pathfinding system, UI Manager, tile map graphics, various obstacles, upgrade crates, and spawn points for tanks. Each level will be loaded and unloaded asynchronously within the Main Scene. Each level will be designed to be progressively more difficult.

1. Level Manager

Handles player and enemy spawning, score tracking, and ending the game.

1. Spawn Points

Handles spawning a player or enemy tank at some point on the map. There are settings to determine which direction the spawned tank should be facing and how much time should pass between spawns at a single spawn point.

1. Level Map

This is a tile map of sprites that make up the “ground” that the tanks drive on. It also has physics colliders to keep the tanks within boundaries.

1. Player & Enemy Tanks

The tank object consists of various child objects allowing the turret to operate independent of the body, the muzzle flash effect that plays when the tank shoots, a Damageable class health system, a TankController class, and an appropriate input module. The player tank uses Unity’s Input System and a PlayerInputHandler translation layer. The enemy tank gets EnemyAI and PlayerDetector classes.

* 1. Input System & PlayerInputHandler

The player controls their tank through Unity’s Input System using either a gamepad controller, such as for an Xbox or PlayStation, or a mouse and keyboard.

* 1. Enemy AI Controller and PlayerDetector

The EnemyAI class accesses the A\* Pathfinding Project package (Appendix B: A\* Pathfinding Project) to track the player. The PlayerDetector handles line of sight to the player for aiming and shooting.

1. Projectiles

Handles instantiating, moving, and destroying projectiles when players or enemies shoot. The projectile determines what it hit and communicates with the object’s health system.

1. Health System (Damageable class)

Both the player and enemy tank use a health system. The player can get powerups to restore health though, while enemies cannot. This system also communicates with the Hud (UI Manager) component.

1. Destructible Obstacles

The first level has destructible barricades and fences, which alter the available paths of travel for the player and enemies when destroyed. The enemy pathfinding simply recalculates anytime an object is destroyed.

1. Crates and Upgrade Pickups

A crate object that drops a random upgrade item when shot and then deactivates. Will reactivate after a given amount of time. There are 6 powerups available that modify various properties of the player tank:

* 1. MaxHP – increases the player’s maximum hit points and then heals the player by half of the new MaxHP minus their current HP level,
  2. Repair – fills player’s hit points to full,
  3. Reload Speed – decreases the time it takes for the player to be able to fire consecutively,
  4. Move Speed – increases the movement and rotation of the player’s tank body,
  5. Aim Speed – increases the rotation speed of the player’s tank turret,
  6. Damage – increases the amount of damage a single shot does to an enemy tank.

The upgrades for reload speed, move speed, aim speed, and damage are indicated by a rank. The player starts with rank 1, and can upgrade to rank 5, by 1 rank per upgrade picked up.

1. Hud (UI Manager)

The Hud gets an update whenever the player takes damage, picks up an upgrade, or destroys an enemy tank.

|  |
| --- |
| Hardware and Software Technologies |
| 1 – Workstation PC running Windows (hardware) |
| 2 – Controllers for testing inputs (hardware) |
| 3 – Unity game engine - latest LTS version (software) |
| 4 – Visual Studio 2022 (software) |

|  |
| --- |
| Revision and Signoff Sheet |

**Change Record**

|  |  |  |
| --- | --- | --- |
| **Date** | **Editor** | **Revision Notes** |
| 2/18/24 | Jamie Lewis | Initial draft for review/discussion |
| 2/25/24 | Jamie Lewis | Updated **Detailed Solution Architecture** and **System Logical Model** sections |
| 3/3/24 | Jamie Lewis | Updated all Excel sheets for Sprint 2: Sprint Backlog, Sprint Burndown Chart, Requirements Traceability Matrix, and Test Cases |
| 3/10/24 | Jamie Lewis | Updated all Excel sheets for Sprint 3: Sprint Backlog & Burndown Charts, Requirements Traceability Matrix, and Test Cases. The Sprint Documents have been combined to remove redundancy. |
| 3/17/24 | Jamie Lewis | Updated all diagrams, user requirements, and design architecture in this document. Updated Sprint Backlog & Sprint 4 Burndown chart. Started reworking Requirements Traceability Matrix, and Test Cases. |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

## **Unit and Integration Test Plans**

**Hydra Tanks**

|  |
| --- |
| Test Tools |
| 1 – Play Testing: test new features and as they are implemented into the game and retest related features to ensure no new conflicts/bugs |
| 2 – Debug Logging in code (these code lines removed after testing completed for code cleanup) |
| 3 - Current testing data is included in the **Test Cases** and **Requirement Traceability Matrix** documentation |

|  |
| --- |
| COMPONENT TESTING |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| COMPONENT: | | |  | | | | | | | |  | Name of Developer: | | | | | | | | |  | | | | | |  | |
|  |  | | |  |  | |  |  |  | | | | |  |  |  | | | | | | | |  |  |  | | | | |  |
|  |  | | |  |  | |  |  |  | | | | |  | Name of Reviewer: | | | | | | |  | | | | |  | | |
|  |  | | |  |  | |  |  |  | | | | |  |  |  | | | |  | | | | |  |  |  |
| Component Test Checklist | | | | | | | | | | | | | | | | | | | | | | | | | | |
| N/A | Pass | | |  | | | | | | | | | Reviewed by Developer | | | | | | Comments | | | | | | | |
| Functionality | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  |  | | |  | | | | | | | | |  | | | | |  | | | | | | | | |
|  |  | | |  | | | | | | | | |  | | | | |  | | | | | | | | |
| Date - | | |  | | | | | | | | | | | | | | | | | | | | | | | |
| Performance | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  |  | | |  | | | | | | | | |  | | | | |  | | | | | | | | |
| Date - | |  | | | | | | | | | | | | | | | | | | | | |
| **Final Sign-Off and Approval** | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  |  | | |  | |  |  |  |  | Project Name- | | | | | | |  | | | | | | | | | |
|  |  | | |  | |  |  |  |  | Date- | | | | | | |  | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **UNIT TESTING** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| **Project:** |  | | | | | | | | | | |  | **Name of Developer:** | | | | | | | | | | |  | | | | | | |  | |
|  |  |  |  | |  | |  | | |  | | | | |  | | | |  |  | | | | | |  |  |  | | | | | | |  |
|  |  |  |  | |  | |  | | |  | | | | |  | | | | **Name of Reviewer:** | | | | | |  | | | | | | |  | |
|  |  |  |  | |  | |  | | |  | | | | |  | | | |  |  | | |  | | | |  |  |  | | | | | | |
| **Unit Test Checklist** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| **N/A** | **Pass** |  | | | | | | | | | | | | **Reviewed by Developer** | | | | | | | | **Comments** | | | | | | | | |
| **Functionality** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  |  |  | | | | | | | | | | | |  | | | | | | | |  | | | | | | | | |
|  |  |  | | | | | | | | | | | |  | | | | | | | |  | | | | | | | | |
|  |  | | | | | | | | Date - | | | | | | | | |  | | | | | | | | | | | | |
| **Other Validations** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  |  |  | | | | | | | | | | | |  | | | | | | | |  | | | | | | | | |
|  |  |  | | | | | | | | | | | |  | | | | | | | |  | | | | | | | | |
|  |  | | | | | | | | Date - | | | | | | | |  | | | | | | | | | | | | | |
| **Processing** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  |  |  | | | | | | | | | | | |  | | | | | | | |  | | | | | | | | |
|  |  |  | | | | | | | | | | | |  | | | | | | | |  | | | | | | | | |
|  |  | | | | | | | | Date - | | | | | | |  | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| **Code QA Checklist** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| **N/A** | **Pass** |  | | | | | | | | | | | | **Reviewed by Developer** | | | | | | | | **Comments** | | | | | | | | |
| **Format and Style** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  |  |  | | | | | | | | | | | |  | | | | | | | |  | | | | | | | | |
|  |  |  | | | | | | | | | | | |  | | | | | | | |  | | | | | | | | |
|  |  | | | | | | | | Date - | | | | | | |  | | | | | | | | | | | | | | |
| **Coding Standards & Practices ( General I)** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  |  |  | | | | | | | | | | | |  | | | | | | | |  | | | | | | | | |
|  |  |  | | | | | | | | | | | |  | | | | | | | |  | | | | | | | | |
| **Performance (if applicable)** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  |  |  | | | | | | | | | | | |  | | | | | | | |  | | | | | | | | |
|  |  | | | | | | | | Date - | | | | | | |  | | | | | | | | | | | | | | |
| **Final Sign-Off and Approval** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|  |  |  |  |  | |  | |  | | | Project Name- | | | | | | | | | |  | | | | | | | | | | |
|  |  |  |  |  | |  | |  | | | Date- | | | | | | | | | |  | | | | | | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ACCEPTANCE TEST FINAL REPORT** | | | | |
| **Project Name:** | | | **Date:** | |
| **Final Sign-Off and Approval (if applicable):** | | | | |
| General description of the acceptance test effort. | | | | |
| **Unresolved Defects** | | | | |
| **Issue/Defect** | **Impact**  **(H, M, L)** | **Risk Mitigation (If known)** | | **Work Around**  **(If known)** |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Requirements Testing** | | | | | |
| Step | Operator Action | Expected Results | Observed Results | Pass/Fail | Date |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |

|  |
| --- |
| **Reference Documents** |

1. If you are going to use established, documented test processes or procedures or if you will need to tailor tests specifically for this project, name these documents and include them in this reference section.

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

**Appendix A – References**

Parrish, A. (2023, December 30). 2023’s great games were overshadowed by a dark cloud of layoffs. *The Verge*. https://www.theverge.com/24009039/video-game-layoffs-2023

**Appendix B – Copyright Compliance (Credits)**

Unity Game Engine - <https://unity.com/legal>

Kenny.nl Art Assets - <https://www.kenney.nl/assets/top-down-tanks-redux>

OpenGameArt.org

Licensing/Attribution FAQ - <https://opengameart.org/content/faq#q-how-to-credit>

Background Music - <https://opengameart.org/content/8bit-title-screen>

<https://opengameart.org/content/8-bit-crisis>

<https://opengameart.org/content/the-good-fight>

<https://opengameart.org/content/5-chiptunes-action>

Sound Effects - <https://opengameart.org/content/cannon-fire>

<https://opengameart.org/content/soundpack-03>

Iconduck.com

License - <https://choosealicense.com/licenses/bsd-3-clause-clear/?ref=iconduck.com>

Speed Icon - <https://iconduck.com/icons/137518/speed>

Repair Icon - <https://iconduck.com/icons/137449/repair>

A\* Pathfinding Project (free version)

Source – <https://arongranberg.com/astar/download>

Licensing info is not on the website, but I found a forum post where the author states that it falls under the Free Asset license in the Unity Asset Store.

Forum Post - <https://forum.arongranberg.com/t/licensing-question/13520>

Unity Asset Store – <https://unity.com/legal/as-terms>

It is common practice in game development to use a pre-built game engine, especially for independent developers. It takes a long time to develop a new game engine, and it is not always necessary. Some larger companies will build their own for a franchise of game titles.

It is also common for independent developers who do not have artistic or musical skills to buy pre-made assets. This is especially encouraged during a prototyping stage, but it is also acceptable to use those in final production provided it is not an “asset flip”. An asset flip means to buy an asset pack that is essentially an entire game already, such as an engine for a specific genre that contains all of the necessary art and audio assets. Hydra Tanks is not an asset flip, because I am creating all of the gameplay mechanics and designed the game from scratch.

**Appendix C – External Resources**

|  |  |
| --- | --- |
| **GIT URL:** | <https://github.com/jtlewis81/HydraTanks>  The code is under /Assets/Scripts  The documentation is under /CST-452 Docs |
| **Hosting URL:** | <https://beachblockgames.itch.io/hydra-tanks> |
| **Presentation Link:** | <https://youtu.be/cso-e3Tnr9g> |