
Software Requirements Specification

for

Echoed Nights

By: Landon Armstrong

New Game Horror (NGH)

4/30/2024

Table of Contents (Page 2-3)

1. Introduction (Page 4)

2. Overall Description (Page 5)

- 2.1 Product Perspective
- 2.2 Features
- 2.3 User Classes
- 2.4 Design Constraints
- 2.5 User Documentation

3. Use Cases (Page 6-9)

- 3.1 Business needs
- 3.2 Changes and Impact on System
- 3.3 Actors
- 3.4 Starting the Night
- 3.5 Exploring the Environment
- 3.6 Encountering AI Enemies
- 3.7 Completing Objectives
- 3.8 Surviving until Dawn
- 3.9 Learning and Adaptation
- 3.10 End of Night Analysis
- 3.11 End of Game Analysis
- 3.12 Audience
- 3.13 Procedure

4. External Interface Requirements (Page10)

- 4.1 Player Interface
- 4.2 Hardware Platforms
- 4.3 Network Interfaces
- 4.4 Integration with External Services
- 4.5 User Documentation
- 4.6 Game Configuration

5. Nonfunctional Requirements (Page 11)

- 5.1 Performance
- 5.2 Safety
- 5.3 Security
- 5.4 Software Quality Attributes

6. Visuals (Page12-14)

- 6.1 Front-end Environment
- 6.2 Main Menu Screen
- 6.3 Models
- 6.4 Object Hierarchy
- 6.5 Assets/Packages
- 6.6 Back-end

7. Wireframe (Page 15)

- 7.1 Main Menu
- 7.2 Night 1+

8. Innovativeness and Creativity (Page 16)

- 8.1 Originality

9. Similar References (Page 17)

- Games

10. Problem Solving (Page 18)

- 10.1 Assets
- 10.2 Integration
- 10.3 Legality
- 10.4 Coding
- 10.5 AI Training
- 10.6 Visual Resources
- 10.7 Asset Management

Introduction

The purpose of this project is to develop a first-person POV horror game set in an environment filled with enemy AI monsters. Players will find themselves trapped in this eerie setting for a predetermined number of in-game nights, where they must complete objectives, uncover the story, and evade attacks from the lurking monsters.

These AI monsters will feature various mechanics such as roaming, chasing, triggering events, setting traps, and more. A unique aspect of the game is the adaptive nature of these monsters, learning from the player's decisions and patterns. This dynamic AI ensures that players cannot rely on repeating the same techniques to easily overcome each night's challenges, adding depth and tension to the gameplay.

My motivation for creating this first-person horror game stems from my background in video game development courses, experience with Unity, and a passion for both playing and creating games. By focusing on player mechanics, game environment, enemy mechanics, and storytelling elements, I aim to deliver a high-quality gaming experience. Additionally, I see this project as an opportunity for personal growth and learning, particularly in implementing original and unique AI elements into the game's monsters.

2. Overall Description

2.1 Product Perspective:

- Software operates within standalone gaming environments.
- No direct interaction with external systems.
- Relies on Unity 3D engine and Visual Studio for development.

2.2 Features:

- Immersive first-person gameplay.
- Dynamic AI-driven enemies.
- Adaptive AI-driven
- Eerie environments.
- Compelling storyline.
- Objectives and threats avoidance.

2.2 User Classes:

- Casual gamers.
- Horror enthusiasts.
- Survival horror genre fans.
- Interact primarily through gameplay interactions.

2.3 Design Constraints:

- Technical constraints: Compatibility with Unity 3D and Visual Studio.
- Regulatory constraints: Compliance with age rating requirements.
- Resource constraints: Time, budget, staffing, 3D model/animation experience, GPU processing power for AI training

2.4 User Documentation:

- Audio log in-game
- No manuals or online help
- After publication on Steam – comment section for user-to-user interaction

3 Use Cases

3.1 Business needs

The development of this horror game is driven by several factors:

- Meeting the demand for immersive gaming experiences in the horror genre.
- Catering to casual gamers, horror enthusiasts, and fans of the survival horror genre.
- Offering a unique gameplay experience through dynamic AI-driven enemies and adaptive AI behavior.
- Leveraging the developer's expertise in game development, Unity, and storytelling to create a high-quality gaming experience.
- Providing opportunities for personal growth and learning in game development and AI implementation.

3.2 Changes and Impact on System

- Having a desktop and peripherals with high performance components and plenty of GPU's to allow for mass training. Not enough processing power doesn't allow for mass training for the AI and may cause lag, impacting proper learning.
- As the gaming industry evolves, there may be changes in technology, player preferences, and market trends.
- Technological advancements may lead to improvements in graphics, AI capabilities, and virtual reality integration, which could enhance the game's features and player experience.
- Changes in player preferences may impact the demand for horror games and influence the design and content of future updates or sequels.
- Market trends such as the rise of digital distribution platforms and subscription services may affect the distribution and monetization strategies for the game.
- The system should be designed with scalability, adaptability, and flexibility to accommodate future changes and updates, ensuring its longevity and relevance in the gaming market.

3.3 Actors

- Player – The user who interacts with the game to progress through the night, complete objectives, and survive until dawn.
- AI Enemies - Non-player characters within the game environment that pose threats to the player and react to the player's actions.
- Environment - The virtual world within the game where the player navigates, explores, and encounters challenges
- Audio Log: In-game recordings that provide backstory, hints, or instructions to the player. (STILL UNDERDEVELOPMENT)

3.4 Starting the Night:

- Player begins a new in-game night, Night 1.
- Player is introduced to game through a “audio log”
- Objectives for the night are displayed, providing direction for the player's actions.

3.5 Exploring the Environment:

- Player navigates the environment using first-person controls.
- Atmospheric effects such as creaking floorboards and distant whispers enhance the sense of dread.
- Player searches for clues, resources, and safe hiding spots to evade enemies.

3.6 Encountering AI Enemies:

- AI enemies(monsters) roam throughout the night, each with distinct behaviors and mechanics.
- Enemies adapt to the player's tactics and patterns, making them more challenging as the game progresses for the following night.
- Player must evade detection by enemies through stealth, flee to safety, or mislead the enemies.

3.7 Completing Objectives:

- Player progresses through the night by completing various objectives scattered throughout the environment.
- Objectives involve finding collectibles within the time limit.
- Failure to collect before the time limit results in a loss, restarting the night.
- Each completed objective brings the player closer to surviving the night and advancing the story.

3.8 Surviving until Dawn:

- Player must survive until dawn to progress to the next night.
- As the night/game progresses, enemies become more aggressive and persistent.
- Player's actions and decisions influence the AI's behavior, shaping the difficulty and intensity of subsequent nights.

3.9 Learning and Adaptation:

- AI enemies dynamically adapt to the player's actions and strategies throughout the night through the reinforcement learning and neural network that is developed through training.
- Enemies learn from successful player evasion tactics, adjusting their patrol routes and search patterns accordingly through triggers to alert and inform the monster agents.
- Player's choices have consequences, influencing the AI's behavior in future encounters or future nights.

3.10 End of Night Analysis:

- After surviving until dawn and completing all objectives, the night concludes.
- Player is introduced into the next night with another audio log that drags the player more into the storyline.
- Player reflects on their experiences and strategizes for the next night based on lessons learned.

3.11 End of Game Analysis:

- The player has completed all nights, the final answer to the story is revealed.

3.12 Audience:

- Players seeking a horror, first-person, or AI implementation video games
- Any age or background
- Average Skill Difficulty

3.13 Training

NAME	Landon Armstrong	
USE CASE NO.	001	
DESCRIPTION	Training the AI	
ACTORS	Landon Armstrong	
PRECONDITIONS	Unity Ready Training packages Virtual Environment config file Run ml agents	
POST CONDITIONS	Observe AI training/movement and Debug.Log Critique reward values or parameters in Visual Studio Try again	
TRIGGER	Failed to achieve proper brain/neural network AI would make considerable progress, then regress in progress made	
DESCRIPTION	STEP	ACTION
	1	Setup the Unity Environment
	2	Install pytorch and other packages
	3	Setup virtual environment file for testing
	4	Setup config.yaml file for parameters
	5	Run ml agents command
STEP EXTENSIONS	STEP	BRANCHES
	1	Using Unity and Visual Studio for object and actors
	2	Making sure packages are a compatible version
	3	Open command prompt, setup directory, and setup venv file localiton
	4	Adjust the config steps to allow for a longer training period, due to bigger task for AI
	5	Run mlagents-learn config.yaml id=Test1
ALTERNATE COURSES	STEP	ACTION
	1	N/A
	2	Follow videos for installation
	3	Follow videos for setup
	4	File Explorer and video tutorial
	5	Use command prompt and run command

4. External Interface Requirements

4.1 Player Interface:

- Objectives HUD
- Startup Menu, Options Menu
- Keyboard and Mouse Controls

4.2 Hardware Platforms:

- PC-Only Compatible game
- Minimum system requirements – Unknown (Most users will meet system requirements)
- Optimization guidelines for performance and resource usage on different hardware platforms – Unknown
- Local save file

4.3 Network Interfaces:

- Singleplayer local game
- No internet connection is required for play

4.4 Integration with External Services:

- None

4.5 User Documentation:

- **Audio logs and objective list for player** - Help systems to guide players through gameplay mechanics and controls
- **None** - Specifications for external documentation, such as digital manuals, online forums, or community wikis
- **English Only** - Localization requirements for translating user documentation into multiple languages

4.6 Game Configuration:

- **Volume Only** - Options for configuring game settings, including graphics quality, audio settings, and gameplay preferences
- **Local Save Only** - Specifications for save/load functionality, including cloud saves and cross-platform compatibility.

5. Nonfunctional Requirements:

5.1 Performance:

- The game should target a minimum frame rate of 30 frames per second (FPS) on default settings
- Load times for transitioning between levels or scenes should not exceed 10 seconds to maintain player immersion.
- The game should efficiently utilize system resources, proper and efficient CPU and GPU usage during gameplay.

5.2 Safety:

- Unknown – proper age rating
- Minimal - disturbing or excessively violent content

5.3 Security:

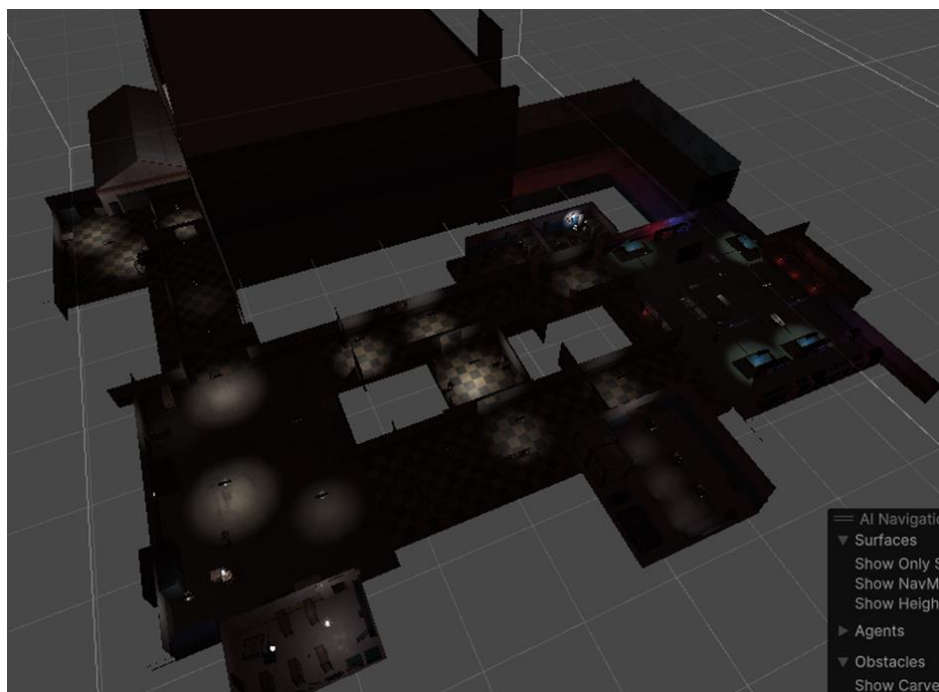
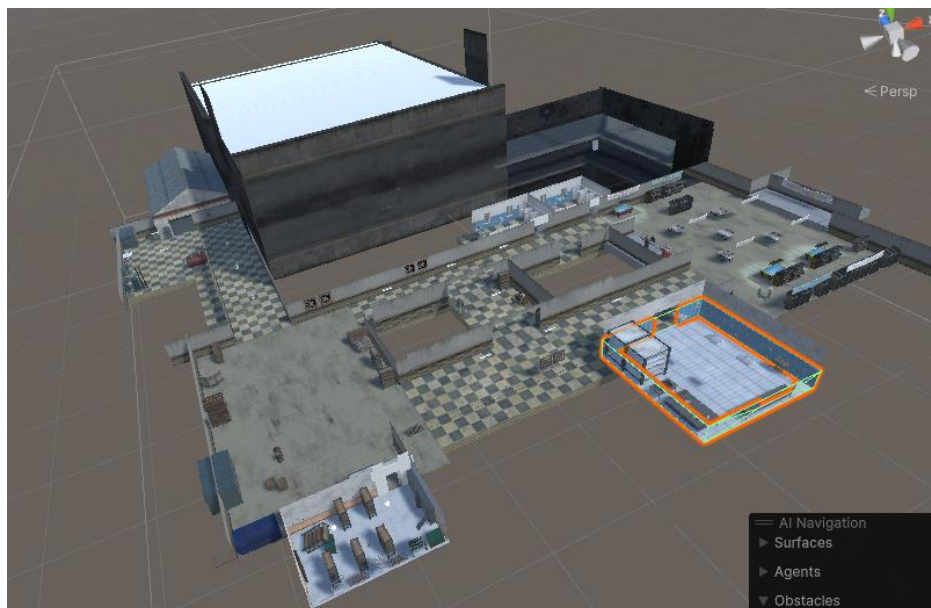
- Player data is all local save file
- None - Anti-cheat mechanisms
- None - Network communications should use secure protocols (e.g., SSL/TLS)

5.4 Software Quality Attributes:

- Reliability: The game will undergo testing to identify and address software bugs, crashes, and errors, ensuring a stable and reliable gaming experience.
- Maintainability: Well-organized code and documented to facilitate future updates, patches, and expansions, allowing for efficient maintenance and expansion
- Scalability: The game architecture designed to accommodate future content updates, additional features, and retain some performance or better with updates
- Usability: The user interface should be user-friendly, straightforward, easy access to game settings and controls.
- Portability: The game is only planned for PC compatibility

6. Visuals

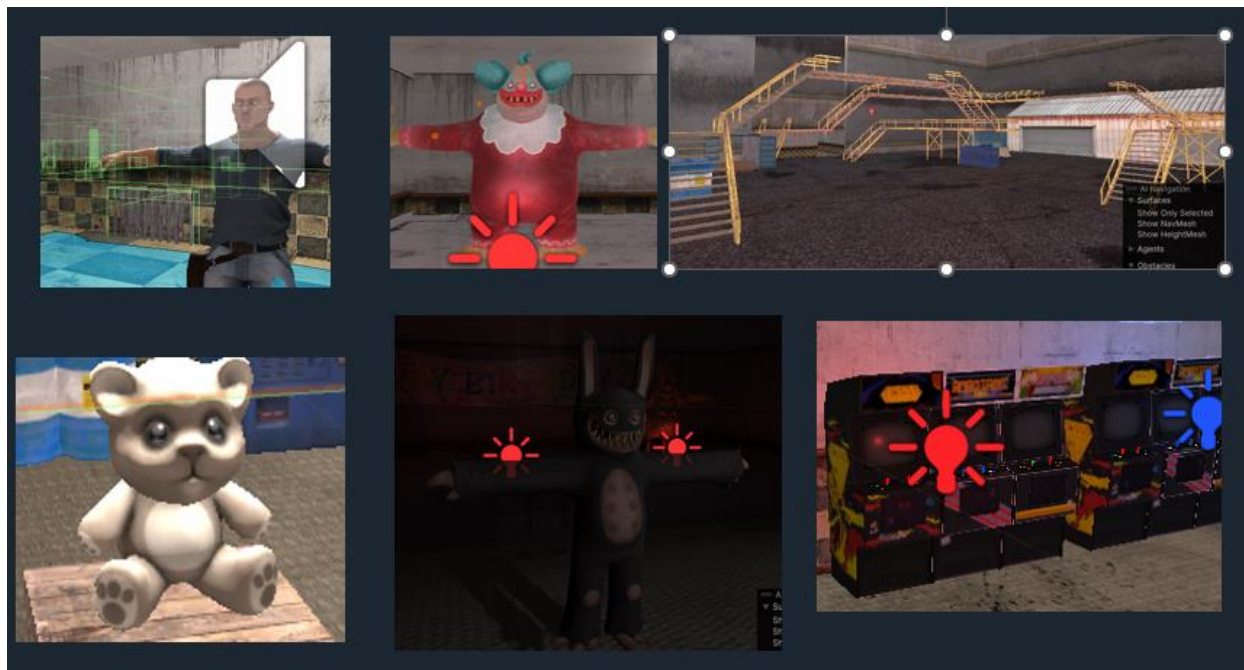
6.1 Front-end: Environment



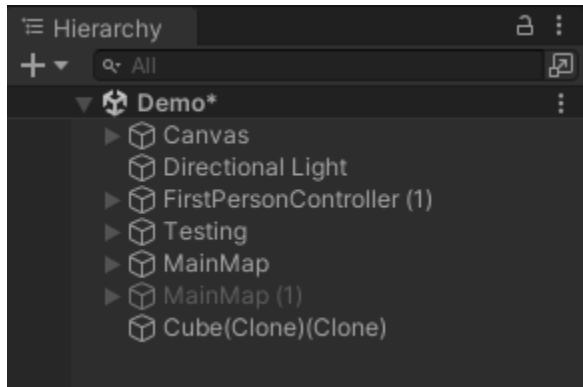
6.2 Main Menu Screen:



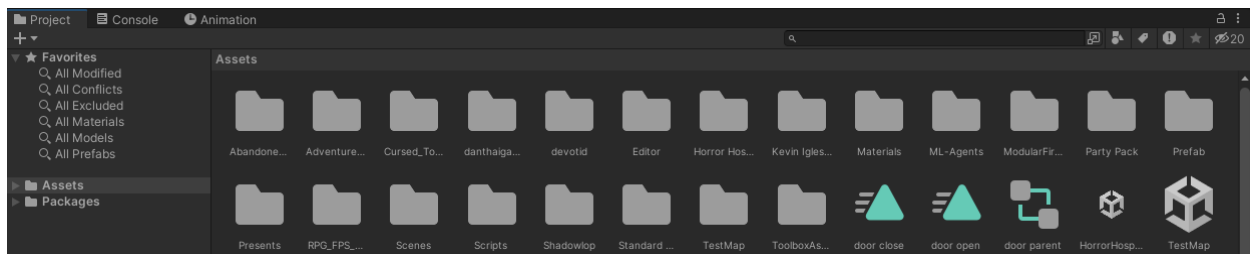
6.3 Models



6.4 Object Hierarchy



6.5 Assets/Packages



6.6 Back-end:

```

@ Unity Script (1 asset reference) | 0 references
public class MonsterAgent : Agent
{
    #region
    private Vector3 playerDummyLocation;
    private Transform playerDummy;
    private bool hasCollidedWithPlayer = true;
    private float previousDistanceToPlayer;
    private Rigidbody rb;
    public float moveSpeed = 1.0f;
    private float timeSinceLastUpdate = 0.0f;

    5 references
    public override void Initialize()...

    10 references
    public override void OnEpisodeBegin()...

    // Implementing the OnActionReceived method to respond to actions from the agent
    15 references
    public override void OnActionReceived(ActionBuffers actions)...

    8 references
    public override void CollectObservations(VectorSensor sensor)...

    /*
    public override void Heuristic(in ActionBuffers actionsOut)
    {
        ActionSegment<float> continuousActions = actionsOut.ContinuousActions;
        continuousActions[0] = Input.GetAxisRaw("Horizontal");
        continuousActions[1] = Input.GetAxisRaw("Vertical");
    }
    */

    @ Unity Message | 0 references
    private void OnTriggerEnter(Collider other)...

    3 references
    private IEnumerator SwapGroundMaterial(Material mat, float time)...

```

```

@ Unity Script (1 asset reference) | 3 references
public class FirstPersonController : MonoBehaviour
{
    private Rigidbody rb;
    private Animator anim;

    Camera Movement Variables
    Movement Variables
    Head Bob

    @ Unity Message | 0 references
    private void Awake()...

    @ Unity Message | 0 references
    void Start()...

    @ Unity Message | 0 references
    private void Update()...

    @ Unity Message | 0 references
    void FixedUpdate()...

    1 reference
    private void HandleCameraMovement()...

    1 reference
    private void HandleCameraZoom()...

    1 reference
    private void HandleSprint()...

    1 reference
    private void HandleMovement()...

    1 reference
    private void Idle()...

    1 reference
    private void Walk()...

```

7. Wireframe

7.1 Main Menu

- Background Screen
- Start – Starts the Night1 Scene
- Options – Volume (Under Development)
- Exit – (Closes Game)

7.2 Night 1+

- Level Layout – Corridors, Arcade Room, Security Office, Main Stage, Secondary Stage, Kitchen, Storage Room, Back Room, Warehouse Room
- Stages consist of Monsters on Display
- Depending on the Night, Monsters spawn in front of their respective Stage
- Player Spawns in Security Office
- Obstacles – Tables, Arcade Machine, Stairs, Pillars, Boxes, Screens, Pallets, and more.
- Collectables – Glowing stuffed animals required for the player to complete the night
- Flow – Player wanders, no hints or given direction. Monsters wander as well

8. Innovativeness and Creativity

8.1 Originality

1. **Adaptive AI Implementation:** While some horror games feature reactive AI, my game would take it a step further by incorporating a sophisticated learning system that allows enemies to dynamically adapt to the player's actions and strategies over multiple in-game nights. This level of adaptiveness creates a truly unique and immersive gameplay experience that constantly challenges players to think on their feet and adjust their tactics.
2. **Multiple monsters, each with its own distinct mechanics and behaviors.** Unlike many horror games that focus on a single antagonist, Echoed Nights offers a diverse array of enemies, each presenting unique challenges and requiring different strategies to overcome all at once in one environment. This adds depth and replay value to the experience, as players must contend with a variety of threats throughout the game.
3. **Extended Night Cycle:** The structure of my game, which spans over several in-game nights in the same environment, similar to the popular Five Nights at Freddy's series. This extended night cycle provides a different pacing and progression compared to other horror games, allowing for a deeper exploration of the environment, narrative, and enemy mechanics over time.
4. **Player-Driven Gameplay:** My game puts a strong emphasis on player agency and decision-making. Unlike traditional horror games where scripted sequences dictate the flow of gameplay, your game empowers players to shape their own experience through their actions and choices. However, giving the player to shape their own experience doesn't necessarily mean beating the game, since they will have to shape their playstyle around the adaptive AI through the nights. The adaptive AI system ensures that no two playthroughs are exactly alike, offering a high degree of replayability and unpredictability.

Through all these aspects the specific combination of features, including adaptive AI, the extended night cycle, multiple enemy AI monsters, and the emphasis on player-driven gameplay, sets it apart from existing games in the horror genre.

By integrating these elements in a cohesive and innovative way, my game concept offers a fresh and unique experience for players, providing them with a new set of challenges and opportunities for exploration and immersion, as well as multiple unique AI in one single environment. This originality is a key strength of my game idea and has the potential to captivate audiences in the gaming community who are seeking new and innovative experiences in the horror genre.

9. Similar References:

Five Nights at Freddy's – Extended Night Cycle, Multiple Monsters

Alien Isolation (A.I) - The two AIs are known as the director AI, and the Alien's behavior tree. The director AI knows the location of the Alien and the player at all times. When the Alien is nearby, the director will periodically give the Alien a hint of your general location, but it will never tell you exactly where you are.

F.E.A.R (A.I) - Fear's AI system relies on three core states within a finite state machine: moving towards a location, interacting with smart objects, and playing animations. These states dictate the behavior of the AI-controlled characters in the game.

10. Problem Solving

10.1 Assets: I've used the Unity Asset Store to implement a lot of 3D models, textures, animations, due to the creation of all these would be a separate learning experience that would exceed and not allow project completion within the time. I chose the assets that best fit the style of restaurant, warehouse, hallways, security office, arcade, atmosphere.

10.2 Integration: Importing assets from the unity asset store sometime have scripts or keywords that are outdated with the current version of Unity and require updating or removal to optimize and allow the project to run while in testing.

10.3 Legality: Using assets published and created by others can have legal limitations. All free unity assets can be used for commercial and non-commercial purposes. No non-free assets have been purchased. More research will be done in order to comply with publication requirements on the Steam Store.

10.4 Coding: Errors and bugs may not allow the game to run to be tested or the code isn't being implemented properly to affect the game. There are plenty of forums, videos, and tutorials out there about how to do certain things in Unity and properly using the game engine with Visual Studio.

10.5 AI Training: Computer doesn't have the processing power it needs to allow for a proper AI training environment. Only one environment could be created in Unity to allow for training. AI would make noticeable progress, than regress on the goal of reaching the player. Mass training and more work on the reward system is needing to fix this.

10.6 Visual Resources: A lot of videos and tutorial teaching about AI training are outdated, lack enough information for certain situations, and only show AI training in a smaller environment. The best I could do I take what I knew and watched various videos and try to apply the best possible route for applying AI training and reward system for the AI.

10.7 Asset Management:

Making sure there is a hierarchy of game objects. Clear asset management system, including folder structures, naming conventions, and asset labels or tags, to keep the project organized and easy to navigate.