

1. Document Overview

- Author(s): names Louis-Viktor Lacombe 2330080, Kheo Chen 2330059 and Karl-Michael Coulombe (2346334)
- Revision History:
 - Version 0.1 – [initial commit]
 - Version 0.2 – [enemies and enemies scriptable object include]
 - Version 0.3 – [camera controller]
 - Version 0.4 – [turns, enemies a.i and shooting added]
 - Version 0.5 – [state machine complete]
 - Version 0.6 – [nav mesh and enemy pathfinding]
 - Version 0.7 – [UI, game scenes added]
 - Version 0.8 – [Levels implemented]

2. Game Overview

- Game Title: Carnage Clock
- Game Genre: Turn-based horror shooter
- Target Platforms: PC
- Game Mode(s): Single-player
- Timeline: 2-week development, broken down as follows:
 - 3 days for player mechanics
 - 3 days for enemy design
 - 2 days for game systems
 - 2 days for level design
 - Remaining time for bug fixes and polish

- **Game Summary:**

This is a third-person, turn-based horror shooter where players must strategically manage 20 action points per turn to navigate eerie, monster-infested environments. Players can move, take cover, and shoot at terrifying creatures during their turn. However, the enemies are nightmarish entities that don't use firearms. Instead, they pursue and surround the player, creating tense and terrifying encounters as the player plans every move to avoid being overwhelmed.

- **Key Features:**

- Free roam movement.
- Lite turn-based combat that adds tension.
- Minimalist UI focuses on core information, such as turns, player action points remaining, and enemies turns.

3. Technical Requirements

- **Game Manager Specifications:**

- The game will track various player metrics:
 - Total player score from enemies killed.
 - Loads scene levels and results scenes

- **Turn Manager Specifications:**

- List all turn-based entities on the map.
- Keep track of turns.
- Manages turn order.

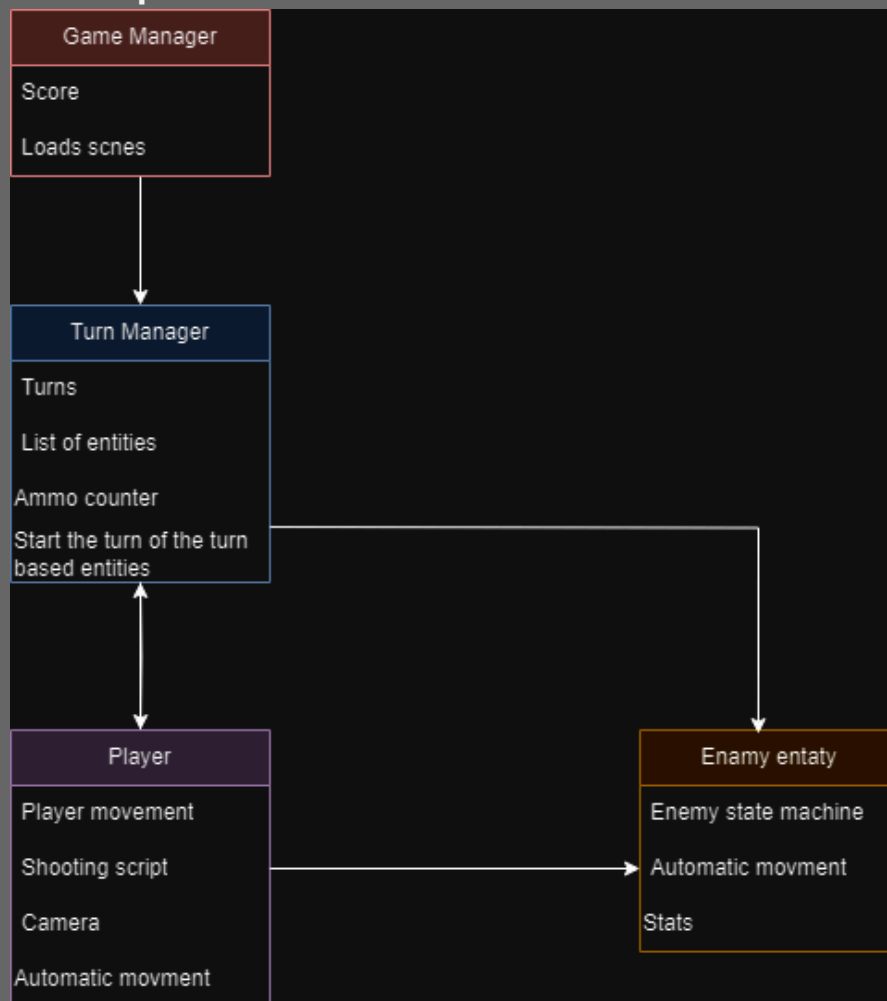
- **Player Requirements:**

- Keyboard and mouse input:
 - Movement: WASD keys for movement.
 - Aiming: Mouse-based tracking, where the player rotates to aim toward the cursor.
 - Actions: Left mouse button to shoot, press 'space' to skip the player's turn.

- **Enemy Requirements:**

- Look for the player.
- It moves randomly if it can't find the player.
- It will move towards the player if found playing when getting within the target range.
- While in attack range, it will start attacking.

4. Technical Graph



5. Architecture Overview

- **Engine Overview:**
 - The game will be built using Unity Engine 2022, selected for its low system requirements, ease of prototyping, and familiarity with the team members.
 - Unity's grey boxing tools will streamline the creation of 3d levels.

- **Key Components:**

- **Player Controller System:** This system will manage all player actions, including movement, aiming, shooting, and skip their turn.
- **Input Handling:** The system will manage all player inputs via keyboard and mouse, ensuring smooth and responsive controls, particularly during combat.
- **AI System:**
 - Enemy AI will be programmed to pursue the player when in range, using basic pathfinding to avoid obstacles.
- **Collision Detection:** Unity's physics system will precisely detect collisions between players, enemies, and environmental objects like doors and barriers.

6. Production Requirements

- **Core Gameplay Systems:**

- **Combat System:** Players will fight enemies using point-and-click shooting, while enemies will use simple melee attacks. Managing the player turn timer will be critical in how they approach combat.
- **Movement System:** Players can move in four directions using WASD. Meanwhile, aiming and shooting will be handled independently using mouse tracking for precision combat.

- **Game UI/UX:**

- **HUD Design:**
 - The player's time for their turn remaining will be shown via a 20 second timer.
 - turn prompts will update as the player ends their turn and the enemy starts.
- **Menus:**
 - The start screen will have options for 'Start' and 'Quit.'
 - End screens will display player stats, including total score and retry.

- **Input Mappings:**

- **Keyboard:**
 - WASD for movement.
 - 'Space' to end turn.
- **Mouse:**
 - Left-click to shoot.
 - Cursor to aim and rotate the player.

7. Tools and Development Environment

- **Version Control:**
 - The development will be tracked on a GitHub repository (TDD-Carnage Clock). The repository will have a main branch for stable builds and testing branches for experimental features and bug fixes.
 - The team will follow a feature branch workflow, where all features are developed in their respective branches and only merged into the main branch after review.
- **Build Process:**
 - The game will be packaged into a build folder before being committed to the GitHub repository.
 - .gitignore will exclude unnecessary files from the repository, such as temporary build files and personal settings.

8. Appendix

- **README.MD:** The README file will control mappings, description of the game and the development team.