

Gridlock - Postmortem

Team Member: Zachary Goldberg

Introduction:

Gridlock is a first-person 3D maze escape game developed in Godot. The main objective of the game is for the player to navigate through the maze, collect a required number of gems, avoid the flying drone orbs, and reach an exit door before the timer runs out.

Gridlock was outlined in the original Game Design Document as a low-poly maze game with a win/lose ending, flying drones that attack, and gem collectibles that emphasize exploration and getting around obstacles. Due to limited time and bugs, a number of changes were made during development, even though the finished game was similar to the initial idea. Some planned features were simplified or removed, while others were expanded and added beyond the original design.

Overall, Gridlock expanded from a conceptual maze-escape game into a fully playable 3D experience featuring first-person controls, floating animated threats, save-and-load functionality, sound effects, background music, HUD/UI systems, and multiple menus. The final version reflects both the original plan and the reality of solo development.

In addition, Gridlock represents a learning-focused development rather than a feature-driven one. Because the game was developed entirely by a single person, every feature, from player movement to UI updates, design, implementation, testing, and debugging, was handled. This influenced many decisions throughout development.

While the original proposal emphasized a simple maze design, the final version evolved into a more structured game loop with progress, win/lose features, and multiple UIs. Features such as save-and-load functionality, a centralized GameManager, and a MusicManager were not initially planned. Still, they became necessary as the project grew into a playable game rather than a concept.

Interactivity and Rules:

The player interacts with the game through multiple systems:

- Movement: The player can move forward, backward, left, and right using the keyboard.
- Camera Control: Mouse movement controls the player's view in first-person.
- Jumping & Sprinting: The player can jump and sprint, with sprinting increasing movement speed.
- Gem Collection: Gems are collected by walking into them. Each gem increments a counter.
- Pause Menu: The game can be paused at any time, stopping gameplay and unlocking the mouse.
- Save Game: The player can save progress through the pause menu.

- Continue Game: The player can load a previously saved run from the main menu.
- Quit Game: The player may exit the game at any time.

The player's interactions are intentionally limited to keep the challenge:

- The player cannot attack enemies.
- The player must collect all required gems before the exit unlocks.
- Being caught by a flying drone orb resets the run.
- Running out of time results in a loss.

These rules enforce careful decision-making and create tension in line with the game's design.

Beyond basic movement and interaction, Gridlock shows rules and controls on menus to inform players of what to do next. The HUD/UI constantly displays the player's current stats, including remaining time and gem progress, ensuring the player always understands their objective.

Player limitations play an essential role in shaping the game. The inability to attack enemies forces the player to rely on navigation and awareness. Players can sprint, but doing so slows down their reaction time in confined spaces, increasing the chance of being captured by a floating drone.

Objectives & Outcomes:

The main objective of Gridlock is to collect all required gems and get to the exit archway before the timer reaches zero. The player wins by reaching the exit after collecting all gems. The player loses if the timer expires or is caught by a flying drone orb. Upon winning or losing, the game displays a clear on-screen message and provides menu options to restart or return to the main menu. UI elements continuously display the player's gem progress and remaining time, removing ambiguity about their current progress. When the exit becomes available, the player can see that all collected gems unlock it. Failure outcomes are designed to be readable. If the timer runs out, the game stops and shows a loss screen. If the player touches a floating drone orb, the game is reset after a delay, allowing time for audio and animation to register. These outcomes ensure that success and failure are never unclear.

Conflict:

Conflict in Gridlock is created in many different ways. Floating drone orbs patrol sections of the maze and restart the player on contact. A countdown timer forces the player to keep advancing in the maze. Tight hallways, dead ends, and limited visibility prevent the player from reaching the exit archway easily. The drones are animated, colored in red, and have audio, so they show that they are a threat.

Visual & Audio Assets:

Menus & Interfaces:

The game includes the following menus:

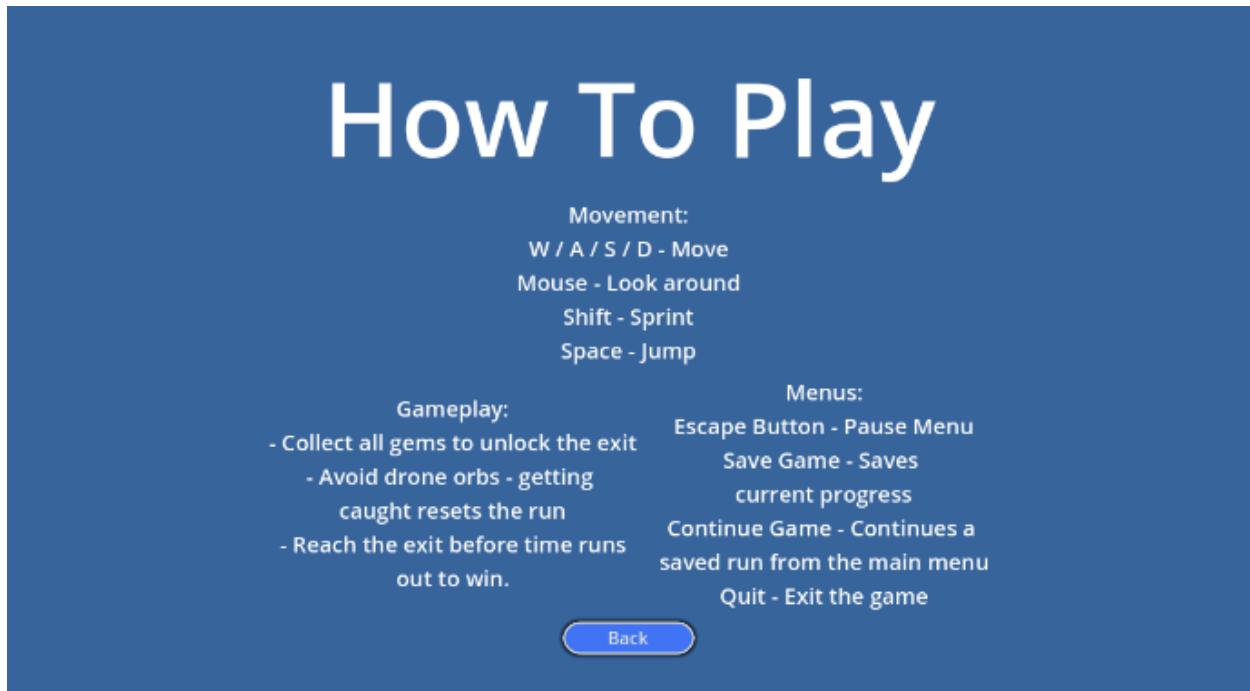
- Main Menu: Displays the game title and buttons to start the game or go to other menus.



- About Menu: Lists the game title and developer credits.



- How to Play Menu: Explains movement, objectives, enemies, saving/loading, and controls.



- Pause Menu: Allows the player to resume, save, or quit.



- Win/Lose Screens: Display clear feedback and what to do next.

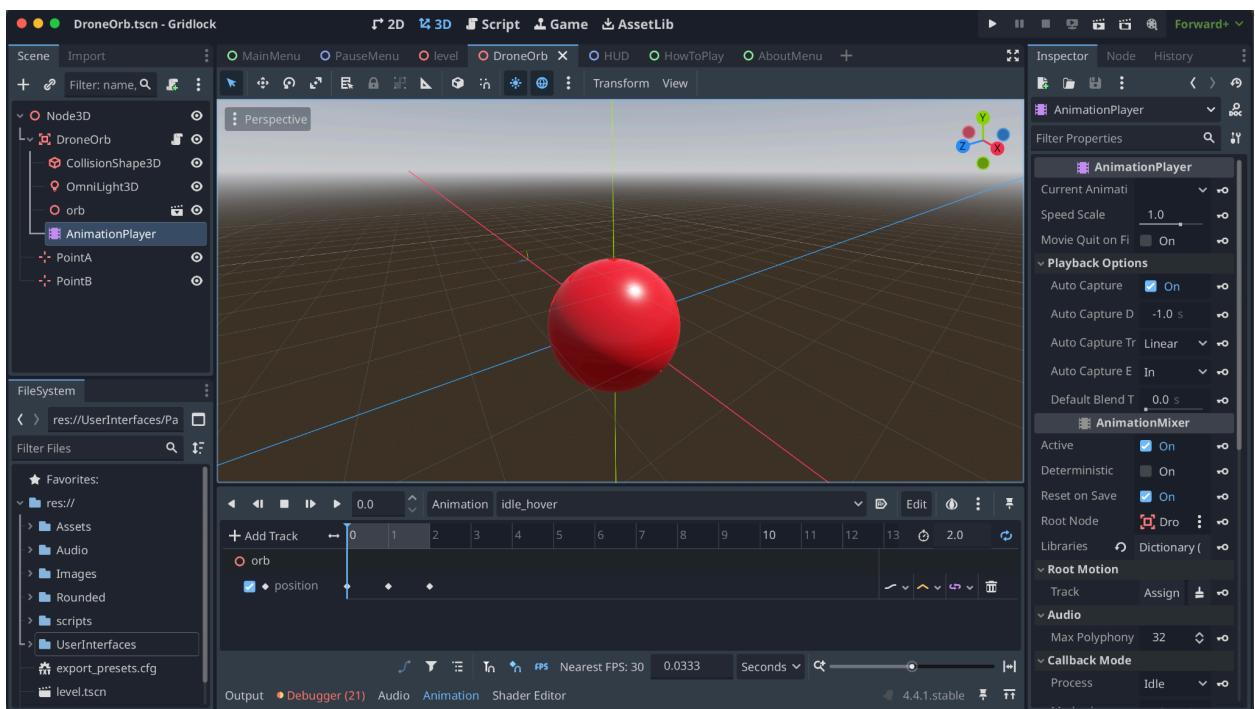
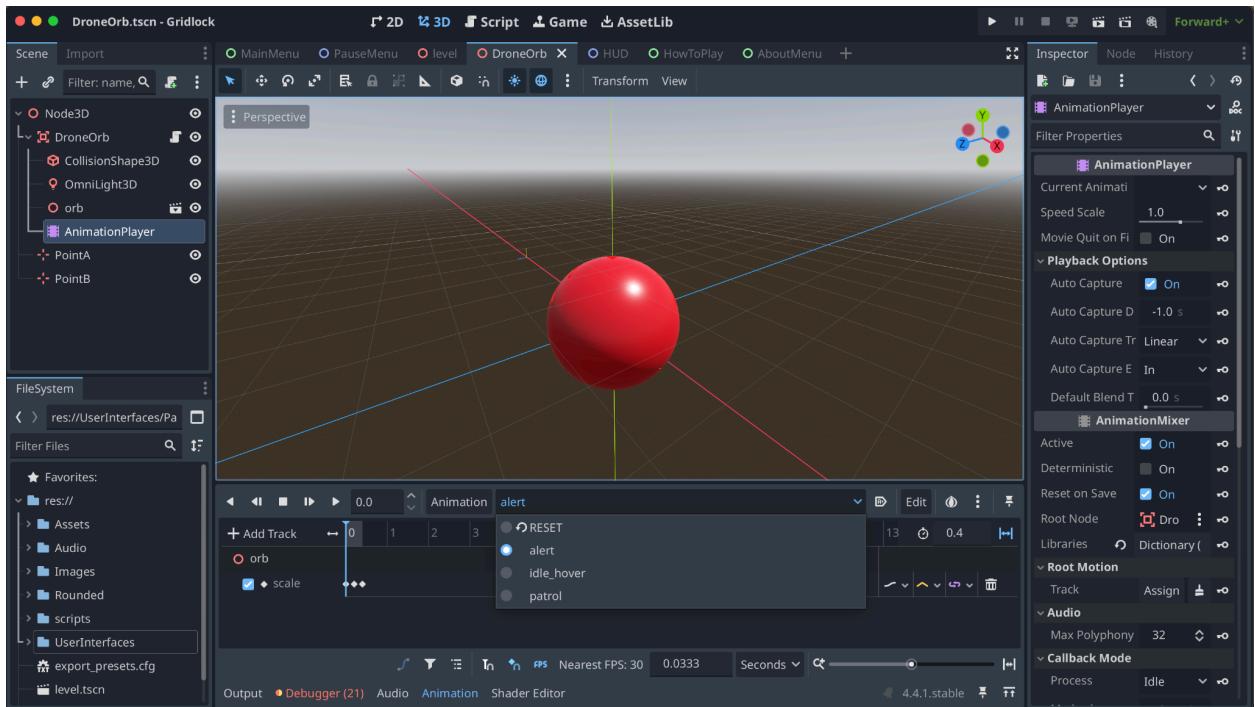


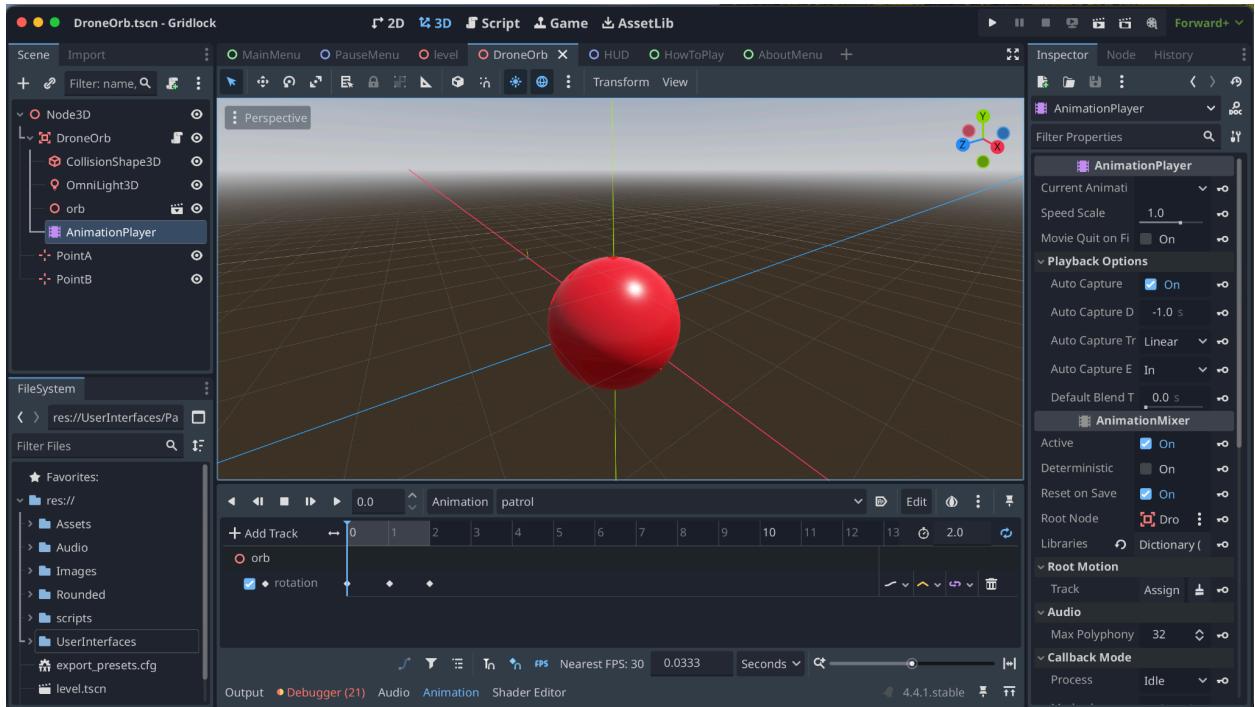
Gridlock has many UIs designed to give information and allow the player to continue through the game. The main menu displays the game's title and offers options to start a new game, continue a saved game, view the About menu, view the How to Play menu, or quit the game. The About menu lists the game title and developer credit, while the How to Play menu explains the controls, objectives, enemies, and save/load functionality in a readable menu. During gameplay, the player can see a HUD/UI that displays the remaining time left and the number of gems collected out of the required total. This information updates in real time, ensuring the player always knows their current progress. A pause menu can be opened at any point during the game, allowing the player to resume, save their progress, or quit to the main menu. When the player wins or loses, a UI message is displayed with options to restart or return to the main menu.

3D Models & Animations:

The game features multiple 3D models, including:

- Flying Drone Orbs (animated, floating, moving, & attacking)

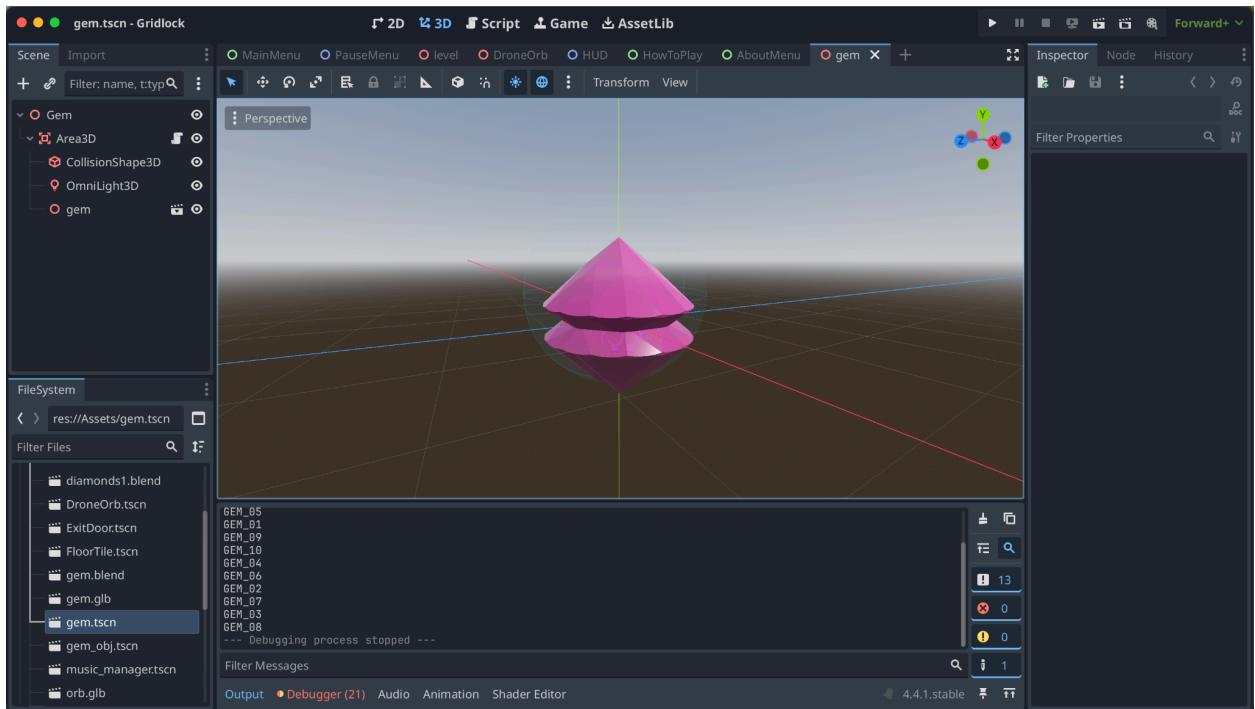




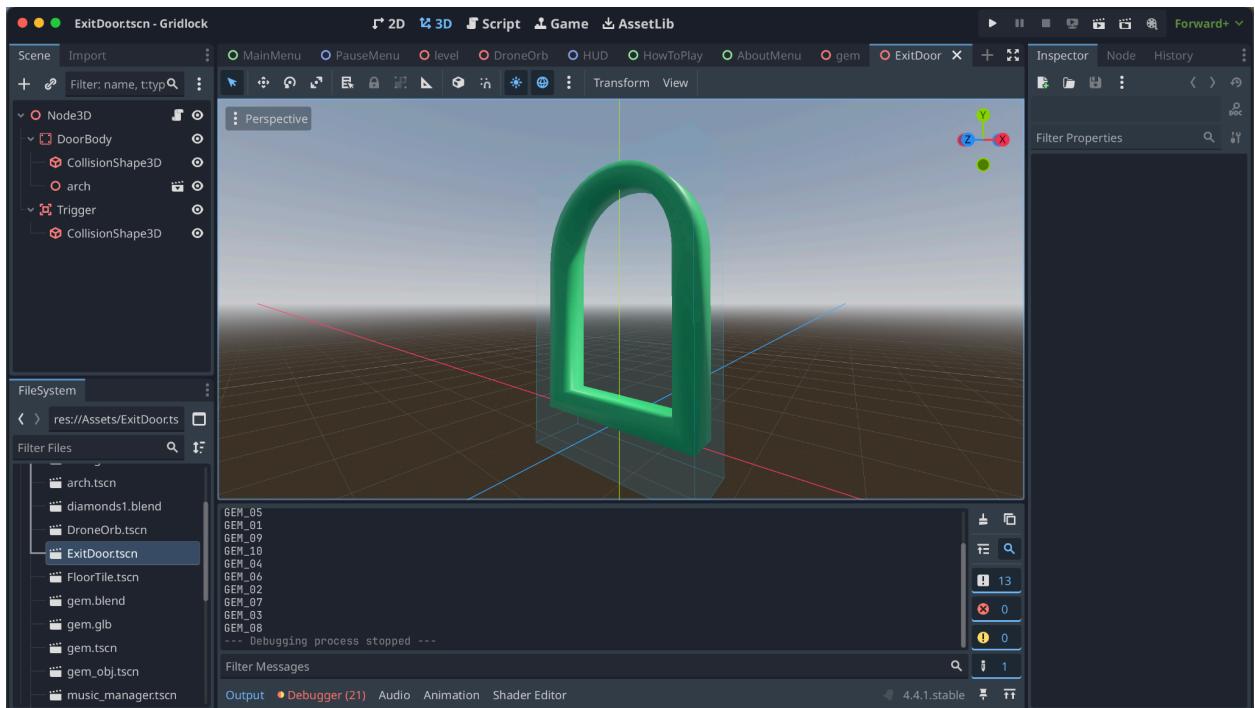
The drone enemy features multiple animations:

- Idle hover
- Patrol movement
- Attack/Alert

- Gems (animated fade-out on collect)



- Exit archway



Gridlock has many 3D models created to support gameplay. The first animated character is the flying drone orb enemy. The drone orb has an idle hover animation when not moving, a patrol movement while driving on its path, and an alert or attack when the player touches it. These animations help the drone appear threatening. Other 3D models include the collectible gems and the exit door. Gems feature a floating and spinning animation and fade out when collected. The exit door is a simple archway colored green, indicating to players that it is the maze's end goal.

Audio Assets:

Audio is very important to the gameplay. The game has a looping background music track that plays throughout gameplay. Sound effects were added for gem collection, drone catching, exit door unlocking, player footsteps, and menu button clicking. All audio assets enhance gameplay without being distracting or overwhelming.

Features Removed:

Several planned features from the Game Design Document were removed or simplified: Advanced Drone Behaviors, Multiple maze levels, Complex animations, and more. The drones were initially planned to chase the player, but were simplified to patrol paths. Only one main maze was implemented to maintain stability and polish. Drone animations were simplified. These features were simplified to keep the project stable, achievable, and within scope for a solo developer.

Problems Encountered:

Throughout development, a few problems occurred. I encountered a collision issue that caused the player to fall through the floors due to an incorrect collision hierarchy. I resolved this by properly assigning collision shapes to StaticBody3D nodes. Another issue was that the mouse camera rotation slowed down at certain angles, caused by scaling the player's parent node. This was fixed by resetting the scale to 1,1,1 and not shaping it differently. The save-and-load feature initially did not restore collected gems correctly. This was fixed by assigning unique IDs to each gem and tracking their collected values when saving and loading. Music also stopped playing when switching scenes, so a global MusicManager was created to keep audio playing. Finally, save files did not work in exported builds due to incorrect file paths; switching to user:// paths fixed this issue. Each of these problems required me to debug, research, and deepen my understanding of how to use Godot.

Conclusion:

Developing Gridlock was a challenging but rewarding experience. Creating a complete 3D game from a concept to a fully running game required problem-solving through programming, design, UI, audio, and user experience. Working by myself increased the workload but also provided complete creative control and a deep understanding of the engine. Debugging the saving/loading and player movements were the biggest challenges for me. Overcoming these challenges led to a better finished product and a greater understanding of game development. The experience reinforced both technical and project management skills. If given the opportunity, I would absolutely do this again.