

Space Tower Defense – Group 2

Team Members:

Jesse Ohler
Jonathan Hernandez
Joshua Roberts
Sydney Melhorn

Project Description:

Space Tower Defense is a modern reinterpretation of a classic Galaga-style shooter. The game preserves the core arcade mechanics of dodging enemies, firing projectiles, and progressing through increasingly difficult levels while adding upgrade-based progression and modern visual elements. Players control a ship and must eliminate waves of enemy blocks before they reach the bottom of the screen or collide with the player. After each level, the user may choose from several upgrades—such as increased damage, faster fire rate, enhanced bullet speed, or multi-shot capability—that customize gameplay and enhance survivability. The combination of classic arcade movement with modern upgrade options creates a dynamic, engaging, and progressively challenging experience.

Prototype link: <https://johnnybass.com/johnny/index.html>

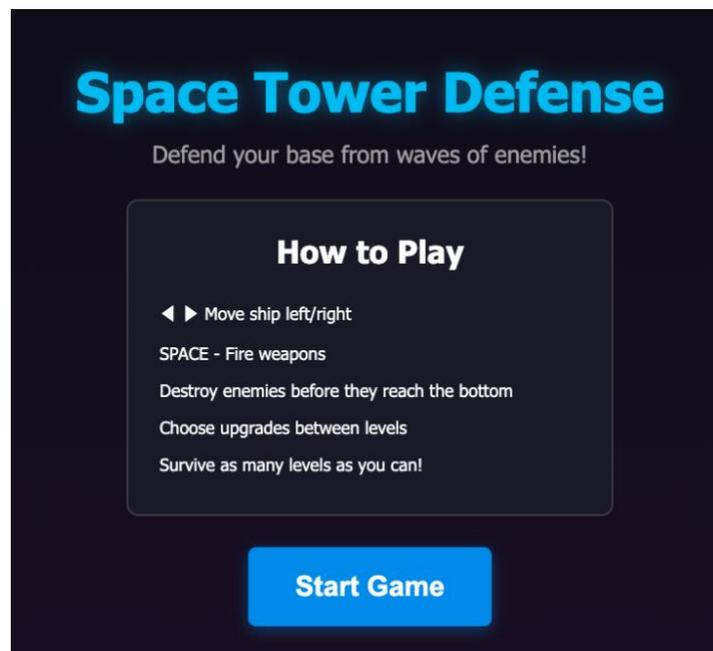


Figure 1 Main Menu Screen

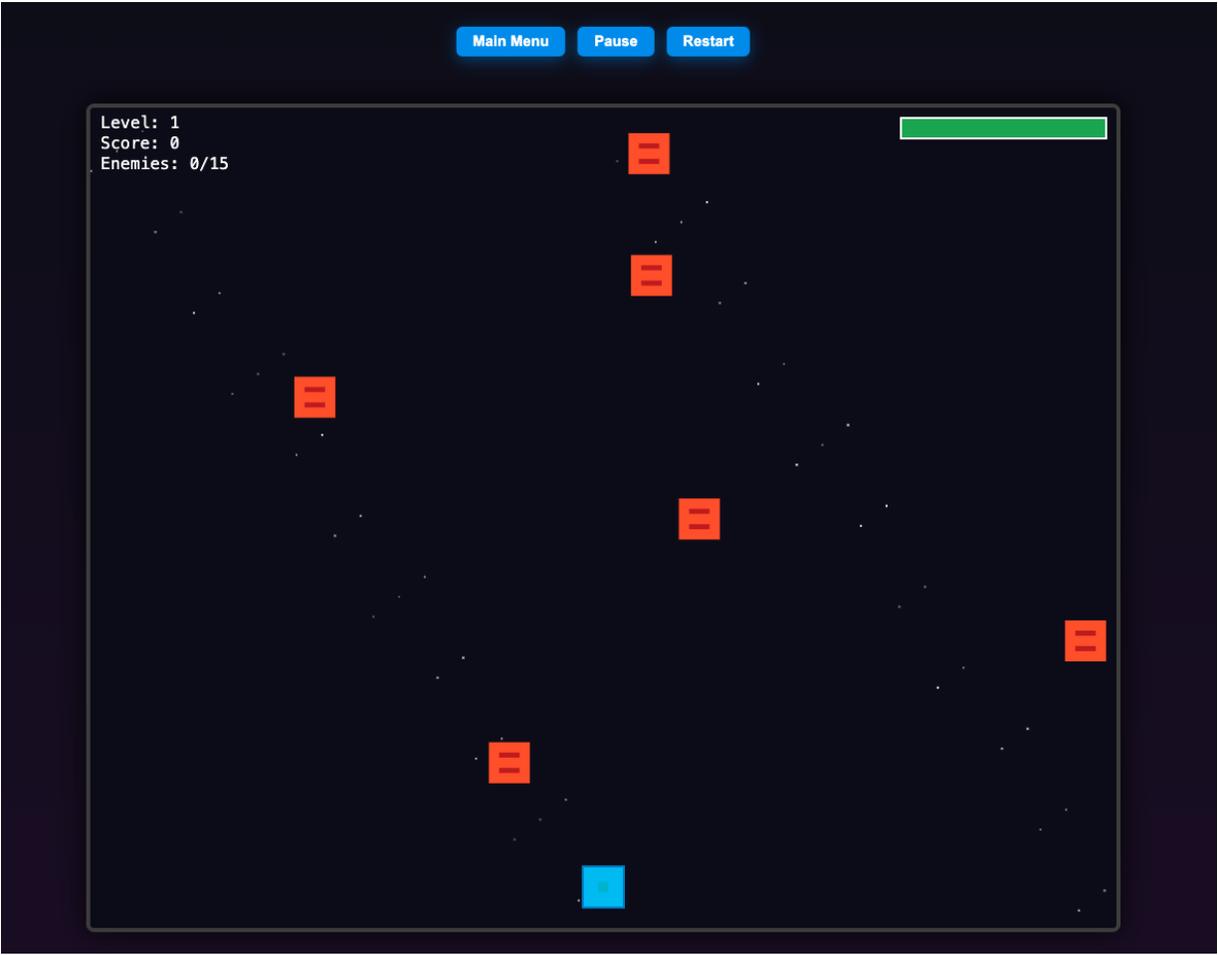


Figure 2 Gameplay Screen

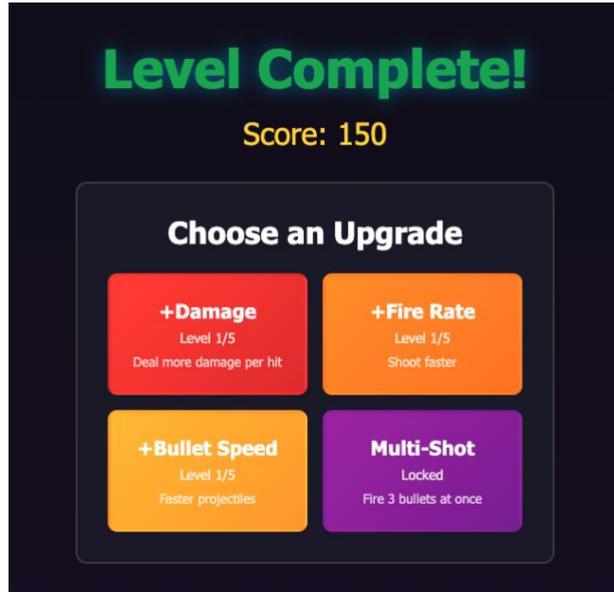


Figure 3 Upgrade Selection Screen

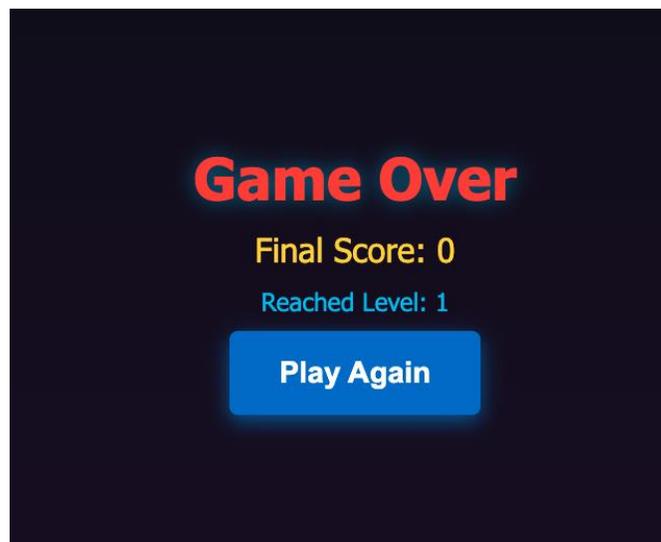


Figure 4 Game Over Screen

Use Case: Play Space Tower Defense

Use Case Name:

Play Space Tower Defense

Primary Actor:

User / Player

Goal:

To destroy all enemy ships required for the level before they reach the bottom of the screen or collide with the player, and then advance to the next level.

Preconditions:

The user has access to a keyboard (arrow keys and space bar).

The game is loaded in the web browser and running.

The user has an active connection if the game is hosted online.

All game assets (audio, scripts, canvas, and UI components) are successfully loaded.

Postconditions:

The user completes the level, selects an upgrade, and progresses to the next level, or the user loses all health and the game transitions to the Game Over screen.

The system records the player's final score for the session.

Main Flow:

1. The user opens the application and views the main menu.
2. The user reads the instructions and selects Start Game.
3. The system loads Level 1 and initializes the player ship.
4. The user moves the ship using the left and right arrow keys.
5. The user fires bullets using the space bar.
6. Enemy ships spawn and move downward.
7. The user must destroy enemies before they reach the bottom or collide with the player.
8. The user gains points for each enemy destroyed.
9. After destroying the required number of enemies, the system marks the level as complete.
10. The system displays four available upgrades.
11. The user selects one upgrade.
12. The system loads the next level with increased difficulty.
13. Steps 4–12 repeat until the user loses all health or chooses to stop.

Alternative Flow A1 – User Pauses the Game:

- A1.1 The user clicks the pause button or the canvas.
- A1.2 The game suspends animations, enemy spawning, and music.
- A1.3 The user may choose Resume, Restart Level, or Main Menu.
- A1.4 If Resume is selected, the game continues play.

- A1.5 If Restart Level is selected, the system resets the current level.
- A1.6 If Main Menu is selected, the system returns to the main menu.

Alternative Flow A2 – User Loses the Game:

- A2.1 The player's health reaches zero.
- A2.2 The game stops all activity and displays the Game Over screen.
- A2.3 The user may restart the game from Level 1.