



Addis Ababa Institute of Technology
School of Information Technology and Engineering

Department of SWE Eng.

Computer Graphics final project
Tower Defence Game

Group member's name	ID
1. Dibora taye	UGR/ 2376/15
2. Eleni abebe	UGR/ 2233/15
3. Hilina zemedkun	UGR/1331/15
4. Ruth yeshitila	UGR/ 8377/15
5. Yohannes worku	UGR/2047/15

Submitted To: Addisalem Genta
Submission-Date: June 22/2025 GC

Contents

Introduction.....	1
1. Objective	3
2. Core mechanism.....	3
3. Features	4
4. Project File Breakdown.....	5
5. Setup and Installation.....	6
6. How to play	7
7. Improvement plan	8
Conclusion.....	9

Introduction

Tower Defense is a 3D strategy game that runs directly in your web browser using **Three.js** and WebGL. In the game, your mission is to defend a small village from continuous waves of enemies by placing different types of towers along the path they follow. Each tower has its own way of attacking, and placing them in the right spots is key to stopping the enemies before they reach the village. If too many enemies get through, you lose health—and when your health runs out, the game is over.

One of the main features of this game is that the environment changes every time you play. Instead of using fixed, pre-made maps, the terrain is created using procedural generation, which means the ground has realistic hills and shapes that are different in each round. The placement of houses, trees, and other objects in the village also changes each time, making each playthrough feel fresh and unique. This helps keep the game interesting and adds variety to your strategy.

This project was built to demonstrate how powerful modern web technologies can be when creating games. It uses WebGL for 3D graphics, custom shaders for effects, and modular JavaScript for clean and organized code. The design is also responsive, so the game adjusts to different screen sizes and devices. Tower Defense is not only fun to play, but it also shows how a complete, interactive 3D game can be built and played fully in the browser—no downloads or installations needed.

1. Objective

The objective of Tower Defense is to protect your village from a constant and escalating cycle of enemy attacks by strategically placing towers along a pre-determined path that the enemies follow. With each successive wave of enemies, each successive wave becomes more challenging—doubling in number, faster speed, and greater resilience—so that gamers need to continuously change their approach. Succeeding in the game depends on strategic planning, keen observation, and good decision-making. The players have to study the landscape design, define tower placement spots, and block enemies from the end of the path. Since each tower can only be within a given range and possess specific attributes, each placement matters and contributes towards making the defense as efficient as possible. The game encourages on-the-fly problem-solving, spatial awareness, and efficient use of limited placement opportunities, and thus it is as much a test of tactical reasoning as it is the ability to quickly respond to changing threats. Ultimately, Tower Defense takes the core elements of strategy and survival and packages them into a streamlined, browser-friendly experience that grows in difficulty and reward by wave

2. Core mechanism

Health System

- The player begins the game with a fixed amount of health (e.g., 100).
- Each enemy that successfully reaches the village reduces this health by a set amount.
- When health drops to zero, the game ends immediately.
- This system creates urgency and pressure, especially in later waves where enemies appear in greater numbers.

Wave Progression

- Enemies spawn in waves that increase in difficulty over time.
- Each new wave features a larger number of enemies, and they move faster, requiring faster responses and better tower placement.
- There may be a short delay between waves to allow the player to adjust their defense, depending on the game design.

Tower Defense Mechanics

- Players can place defensive towers on grid cells marked as valid (usually in green).
- Towers automatically attack any enemy that comes within their detection radius.
- Different types of towers (optional in future updates) may include various fire rates, damage levels, and range capabilities.
- Towers cannot be placed on the enemy path or terrain obstacles.

- Strategic tower distribution is essential to covering the entire enemy path efficiently..

Score Tracking

- The game tracks the number of enemies defeated and overall wave progression.
- Players are encouraged to beat their previous best scores or attempt perfect defense runs for added replay value..

Victory Conditions

- Complete a round or the full game without allowing a single enemy to reach the village

Defeat Conditions

- The game ends immediately when the player's health drops to zero, regardless of the wave count.

3. Features

3D Graphics

Tower Defense is built using Three.js, a powerful 3D graphics library that enables rich visual experiences directly in the browser. The game world is rendered entirely in 3D, with smooth animations, realistic lighting, dynamic shadows, and a movable camera. Players can rotate, zoom, and pan the camera to explore the battlefield from different angles, helping them plan tower placement more effectively. This level of visual control adds depth and immersion to the gameplay, making the experience more engaging than traditional 2D tower defense games.

Procedural Environment

Every time the game starts, the environment is procedurally generated, meaning it's different each time you play. This ensures that no two game sessions feel exactly the same. Procedural elements include:

- **Terrain Generation:** The ground surface is created using custom shaders that simulate realistic hills and elevation. This terrain isn't flat—its natural variations give the game world a more organic and lifelike feels.
- **Village Decorations:** Visual elements like houses, trees, rocks, and other decorations are randomly placed across the map. These features are not just decorative—they help create a believable environment and may influence how you view and plan tower placements.

Dynamic Enemy Waves

Enemies attack in structured waves, and each wave is more difficult than the last. As the game progresses:

- The number of enemies per wave increases.
- Enemies move faster, reducing the time you have to react.
- The difficulty scales up gradually, making early-game strategy and preparation critical to long-term success.

Tower Placement

At the heart of the game is a grid-based tower placement system. Players can build towers on predefined grid tiles (typically marked in green), which serve as defensive structures against the incoming enemies. Important aspects of tower placement include:

- **Automatic Targeting:** Towers automatically detect and shoot at enemies within their range.
- **Range and Effectiveness:** Each tower is most effective when placed at locations that cover large sections of the enemy path.
- **Restrictions:** Towers cannot be placed directly on the enemy path, ensuring players must think strategically about their positioning.

UI and Game State

The User Interface keeps players informed about their current progress and game state. The UI includes:

- **Health Indicator:** Displays the player's remaining health. Every enemy that reaches the village lowers this value. When it reaches zero, the game ends.
- **Score Tracker:** Shows the player's current score based on the number of enemies defeated. Higher scores reflect better efficiency and defense.
- **Wave Counter:** Indicates the current wave number, giving the player insight into how far they've progressed.

4. Project File Breakdown

Root Directory

- **index.html**
The main HTML file and entry point of the game. It defines the canvas for rendering, includes UI elements (like health, score, and control buttons), and loads JavaScript modules using an import map for Three.js.

- **style.css**
Contains all visual styling for the game, including layout, fonts, UI panels, buttons, and the game-over modal. It ensures the interface is clean, responsive, and visually consistent.

js/ Directory

- **main.js**
Central control of the game. It initializes the Three.js scene, camera, lighting, game loop (animate), and user interactions. It acts as the coordinator, calling functions from all other modules.
- **gameState.js**
Manages core game variables like health, score, and wave count. It provides methods to update these values and handles end-game conditions.
- **ground.js**
Generates the game terrain using shaders. The vertex shader creates a natural, bumpy surface, while the fragment shader gives it a grassy appearance—entirely procedural, no textures used.
- **path.js**
Defines the enemy path as an array of 3D coordinates. Also generates a visible mesh to represent the route, helping players plan their defenses.
- **enemies.js**
Handles enemy creation, movement along the path, and removal upon reaching the village or being defeated. Each wave is managed through this module.
- **towers.js**
Controls tower placement and behavior. Towers detect nearby enemies, fire projectiles, and handle targeting and damage logic.
- **village.js**
Adds visual details like houses and trees to enhance the environment. Objects are placed randomly but avoid interfering with the enemy path.
- **utils.js**
Contains reusable helper functions, such as createParticleSystem for effects like enemy death explosions.

5. Setup and Installation

To run the game properly, you need to use a local web server. This is because the game uses modern JavaScript features that don't work correctly when opened directly from your file system.

1. Download the Project

Get all the required files, including index.html, style.css, and the entire js folder. Place them together in a single directory on your computer.

2. Start a Local Server

Use a simple local server to host the project on your computer. You can do this with tools like Python or Node.js, depending on what you have installed.

3. Open the Game in Your Browser

Once the server is running, open your browser and go to the local server address (usually something like `http://localhost:8000`). The game will load and you can start playing.

Note: Opening the game by double-clicking the HTML file won't work correctly due to how modern browsers handle modules and file access. Always use a local server.

6. How to play

Controls

- **Camera:**
 - Rotate the view by left-clicking and dragging.
 - Pan the view by right-clicking and dragging.
 - Zoom in and out using the mouse scroll wheel.
- **Gameplay:**
 - Place towers by left-clicking on green tiles, which are valid spots on the grid.
 - Towers automatically attack enemies within their range.
 - You cannot place towers on red tiles, which represent the enemy path.
 - Start the next wave by clicking the “Start Next Wave” button.
 - Reset the game anytime by clicking the “Reset Game” button.

Basic Steps to Play

1. Use the camera controls to explore the map and get a good view of the enemy path.
2. Identify green tiles along the enemy's route where you can place towers.
3. Click on these green tiles to build towers that will attack enemies automatically.
4. When ready, click the button to start the next wave of enemies.
5. Between waves, adjust your tower placement and plan your defense.
6. Continue placing towers and starting waves until the game ends.

How to Win

You win by successfully defending your village through all the enemy waves without losing all your health. This means placing towers strategically to stop every enemy before they reach the end of the path. Achieving a high score by defeating many enemies efficiently also counts as a successful game.

How You Can Lose

You lose the game if your health reaches zero. Each enemy that manages to reach your village reduces your health, so if too many get through, your defense will fail .

When your health reaches zero, the game immediately ends. A "Game Over" modal appears with a message letting you know your health ran out. All game actions stop enemies freeze, towers stop firing, and the animation pauses. You can restart the game by clicking "Play Again" or "Reset Game," which resets your health, score, wave, clears the battlefield, and starts a new game session.

7. Improvement plan

- Add new tower types with special effects (e.g., splash damage, slow enemies).
- Introduce different enemy types with varying strengths and abilities.
- Implement tower upgrades and in-game currency system.
- Include a pause menu and audio controls.
- Optimize performance by using instanced rendering for repeated objects.
- Save game state using local storage or session data.

Conclusion

Tower Defense successfully combines strategic gameplay with immersive 3D graphics, creating an engaging and challenging experience that runs entirely in the browser. The procedural generation of the environment and dynamic enemy waves ensure that each round feels unique, requiring players to constantly adjust their tactics and tower placements. With its modular architecture and modern web technologies, the project showcases what is possible with browser-based 3D games and has potential for future development. Planned enhancements like new tower types, enemy variations, upgrades, and performance optimizations will further enhance this game and create better player experience .

