9 Mazes of Hell Documentation

Creators:

Alquicer, Randall
Dela Cruz, Leian Carl
Lapaz, Jermel
Nuñez, Bazer Timothy
Vaflor, Adrian

1. Project Overview

- **Game Title**: *9 Mazes of Hell*
- **Inspiration**: Based on Dante Alighieri's *Divine Comedy* and *Pac-Man*.
- **Objective**: The player, controlling Raphael, must navigate nine increasingly complex mazes, outwit AI-driven demons, and rescue Constance.

2. Introduction of the Game

- *9 Mazes of Hell* places players in the shoes of Raphael, a hero who dives into hell to rescue his beloved. Players must navigate nine labyrinthine mazes, each crawling with demons that use different AI search and pathfinding algorithms. The player must outmaneuver these demons to reach the end of each maze and proceed to the next.

1. Setup Instructions

Step 1: Download and Extract the Project Files

1. Download the provided zip file from the submission bin in Google Classroom.
2. Once downloaded, unzip the file. You should see the following contents:
   - `9-Mazes-of-Hell` folder
   - `INSTRUCTIONS.txt`
   - `Gameplay.mp4`

Step 2: Open the Project in Your IDE

1. Open the `9-Mazes-of-Hell` folder in your preferred IDE (e.g., VS Code).

Step 3: Ensure npm is Installed

1. Open Command Prompt (cmd) and type: npm -v
   - If a version number is displayed, npm is already installed.
   - If not, follow the steps below to install Node.js and npm.

Step 4: Install Node.js and npm (If Not Installed)

1. Visit the official Node.js website: https://nodejs.org/
2. Download the **LTS (Long-Term Support)** version for better stability.

3. Run the installer and follow the setup instructions (click "Next" until finished).

Step 5: Install Dependencies and Run the Game
1. Open the terminal and navigate to the main project folder: cd path/to/9-Mazes-of-Hell
2. Install the required dependencies by running: npm i
3. Start the development server with: npm run dev
4. Once the server starts, it will provide a URL (e.g., `http://localhost:8080/`). Open this URL in your web browser.

Step 6: Verify the Game
- The game should resemble the visuals in the `Gameplay.mp4` file.

Step 7: Enjoy!
Have fun playing 9-Mazes-of-Hell!

2. Game Design

- Mechanics
    - Where does the game take place?
      The game is situated in mazes of hell. Currently only 1 maze is made and we plan to add more.
    - When does the game end?
      The goal of the game is to reach the endpoint in the maze to proceed to the next level or next maze. There is nothing specific that marks the exit, but the first maze is obvious, and it is intentional to add difficulty to the game. The game would finally end after the player had completed all 9 mazes.
    - How to play and win?
      The rules are simple, move the warrior up, down, left, and right using arrow keys. Try not to collide with the demons upon reaching the exit. Colliding the demon results in game over and reaching the exit proceeds to the next maze.

3. Technical Specs

- Assets used
  https://craftpix.net/freebies/free-undead-tileset-top-down-pixel-art/
  https://foozlecc.itch.io/lucifer-warrior
- Tile animation plugin:
  https://raw.githubusercontent.com/nkholski/phaser-animated-tiles/master/dist/AnimatedTiles.js
- Tools Used
    - **Javascript**: Main tech stack
    - **Phaser**: Game framework

- **Tiled**: For designing the map tilesets.
- **Canva**: For game posters (storyline and main menu)
- **Bandlab**: For soundtrack composition.
- **Pixabay:** For royalty-free music and sound effects

4. Implementation

- Creating the map

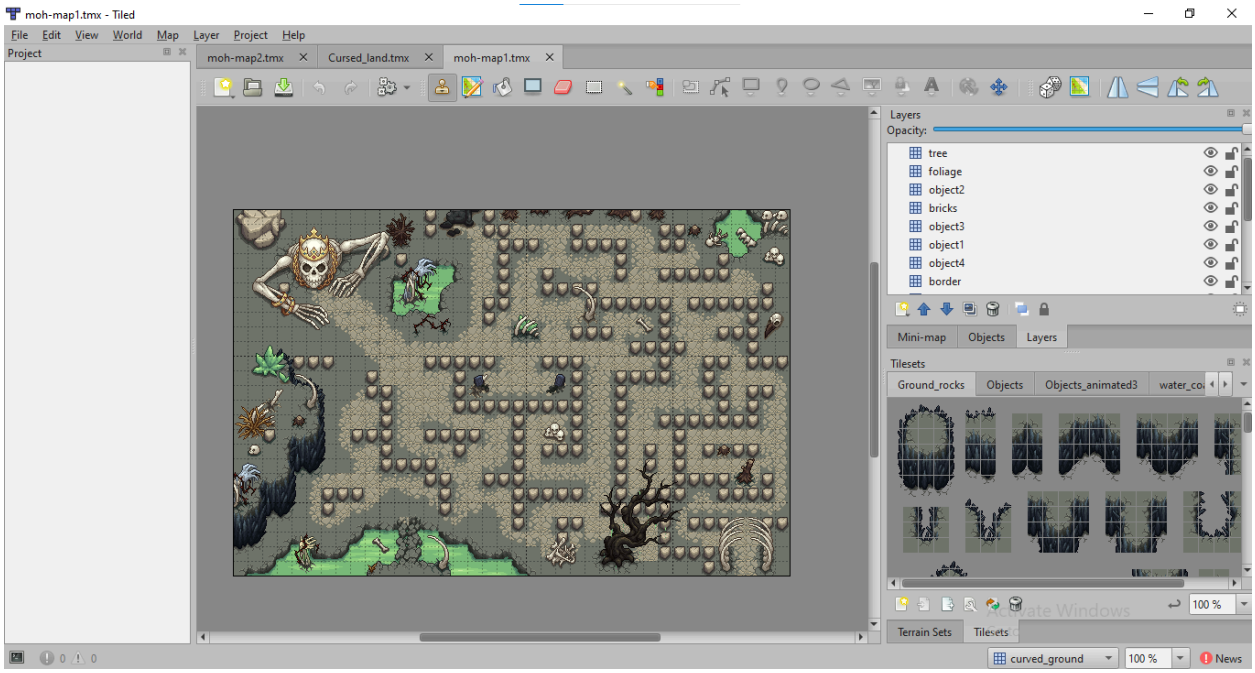

Figure 1. Map created using Tiled software

From the assets freely available online, a map was made using Tiled software. In tiled, layers such as tree, water, ground, bridge, foliage, etc. are created. Those layers used the tilesets we downloaded. Each tileset was edited to add a collision property which is needed to help indicate when a sprite hits a particular object.

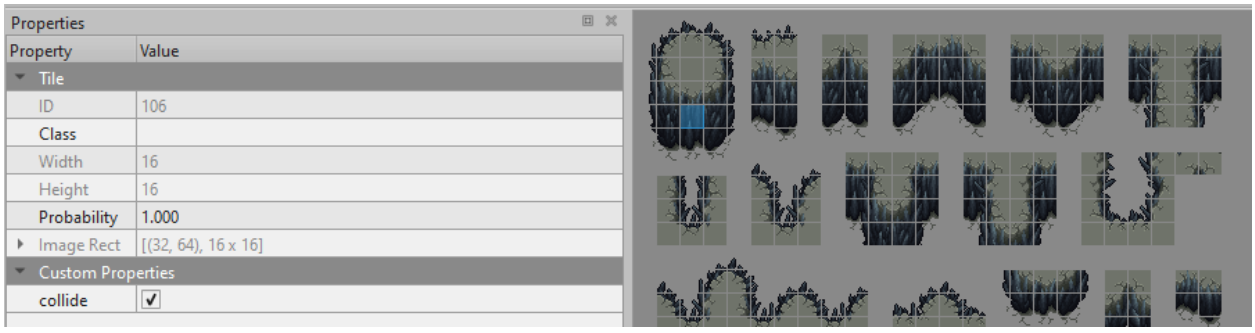

Figure 2. A particular area of the Ground_rocks tileset marked as collide (a player can not pass through it)

After the map was created, it was then exported in a json file. In our code, the exported json file was used. Phaser rebuilds the map by way of knowing which images are used, and what layers need to be created. So, in our code, in Game.js, those information were declared.

Because phaser does not have a feature to animate moving tiles in the map. We added this plugin we found online https://raw.githubusercontent.com/nkholski/phaser-animated-tiles/master/dist/AnimatedTiles.js

- Character Movements
  The character, which is a warrior, is animated by using the spritesheets we found online. The sprite sheets were loaded and adjusted the frame per second and other measurements. To add velocity to the sprites, we bind keyboard keys (arrow up, arrow, down, arrow left, arrow right).
- Demon Movements
  Same as how we created the warrior but the movements are automated. Because we added a “collide” property in each of our tilesets, we are able to detect when the sprite hits an object that it cannot pass through. If the demon hits an object, it randomly chooses a direction to walk to. To make the movement natural, we also added an idle state of the demon movement which happens 3% of the time.
- Collision Box
  Setting a collision box is important in order to add markings of the sprite to determine

what to consider a collision and what to do with the collide.

Figure 3. Entities with collision boxes (demon: violet, warrior: red, success area: green)

Figure 4. Warrior colliding the maze (left), demon colliding the maze(center), success area (right)

The collide boxes of entities was intentionally made to only cover the lower part of the body in order to have this semi-realistic view of the map by allowing areas of the entities to overlap “collidable” objects.

AI - Component

Vampires have 3 classes.

- Vampire_1 creates instances of vampires which moves in a straight line and turns to a random direction when bumping a wall
- Vampire_2 creates instances which moves along walls. More predictable movement, but wider coverage in map.
- Vampire_3 creates somewhat smart instances. It knows the coordinates of the player at all times and how far it is. It tries to travel in a diagonal direction to the player and slides on walls upon collision. Kinda like how Vampire_2 does. But only for a specific amount of time. This is just as to avoid getting stuck on pockets.

5. Game Screens

Story.js

- This scene presents an introductory sequence before transitioning to the main menu. The story unfolds through a series of text lines that are displayed one at a time, simulating a typing effect.

MainMenu.js

- This scene serves as the entry point for the player, providing options to start the game or view instructions.

Game.js

- The Game.js file is the main game scene of our game. Here's how it was implemented:

1. Loads Assets in preload()

- Loads audio (background music).
- Loads tilemap and tileset images for the game environment.
- Loads spritesheets for animations of the player and enemies.

2. Initializes Game in create()

- Plays background music.
- Creates a player (Warrior) and four Vampire enemies using the Warrior and Vampire classes.
- Sets up collision detection between the player and vampires.
- Configures keyboard input:
    - Arrow keys (← ↑ ↓ →) for the character.
- Adjusts depth for proper layering of sprites.
- Prevents characters from moving outside the game world.
- Sets the pixel filtering mode to NEAREST for a pixelated retro look.
- Defines a success area in the lower right corner (possibly a win condition).
- Calls createMap() and createAnimations() (not fully shown).

3. Handles Player-Vampire Collision (handlePlayerVampireCollision())

- If the player touches a vampire, the game stops and transitions to a Game Over screen.

4. Loads and Creates the Tilemap (createMap())

- Loads a tilemap (map1.json) and its associated tilesets (ground, objects, water, etc.).

.

6. Future Plans

Create more maps, continue the progress of player attacks, add AI, and introduce more demons.

7. List of Credits

Undead Tileset (Top-Down Pixel Art) by **CraftPix.net**
Source: https://craftpix.net/freebies/free-undead-tileset-top-down-pixel-art/

Lucifer Warrior by FoozleCC
Source: https://foozlecc.itch.io/lucifer-warrior

Phaser Animated Tiles by nkholski
Source:
https://raw.githubusercontent.com/nkholski/phaser-animated-tiles/master/dist/AnimatedTiles.js