TCSS 360 Software Development & Quality Assurance

Cool Trivia Maze By: Renzo Aquino, Shiraz Arif, and Caleb Ernst

Team contributions: (Total time tracked via YouTrack)

Renzo Aquino: Total Time - *85h 45m*

I called for a handful of meetings and constantly bugged my teammates on our progress for each sprint. In each meeting, we checked in with each other on how we are doing and what we have. I contributed in communicating on what we have to prioritize for each sprint, and considered other views so everyone has a say in this project. I did my best to ask questions for clarification, for not only me, but the team as a whole. I worked with my team to manage the github repository, and communicated when I made changes to my branch or when I started any pull requests.

In terms of the project work itself, my primary role was starting us off. When learning Godot, I provided multiple links for tutorials and introductions regarding Godot. I whipped up the first room concept and player controls during our first sprint. I also practiced the SQLite plugin for Godot (godot-sqlite), and presented what I learned to the rest of the team, making sure they understood my code from the first pause menu build I made. I also started up the Unit Tests by learning how the Godot plugin GUT works, and provided tutorials to my teammates as well. I did my best to develop the pause menu (which I presented during the code review meeting), and my teammates were able to take that code I developed, and connect it with the game using the factory pattern. For extra credit, I worked on all of the music (not including sound effects), as well as all artworks for the game, as art is a primary skill of mine.

Here are some *additional* things I did:

- Created about 40 questions (worked with caleb with some) for the database
- Created the .csv file converter (csvParser.py)
- Win Scene + Script
- Lose Scene + Script + edited together sound effects
- Main Menu Scene + Script
- Fade to black/white scene transition animations + scripts

Shiraz Arif: Total Time - *71h 40m*

Set up and maintained the YouTrack project, including the agile board and time reports. Handled merge conflicts and set up gitconfigs to enable significantly easier merges for godot scenes. Started up the spreadsheet to be used as the question database and provided formatting guidelines to ensure consistency. Created the logo. Fixed some unit tests (minor).

Implementations:

- Systems for maze data, room data, room traversal, door data, and door hitboxes/detection
- SaveManager, Question, and QuestionFactory
- Save/Load function via serialization/deserialization
- Menu functions (excluding volume sliders) and layout (all found within the save_select scene and script)

Caleb Ernst: Total Time - *55h 57m*

Directed the project structure and managed the github repository. Balanced the importance of creating a substantial project while meeting the suggestions and personal requests of my fellow team members. I focused most of my time on the core aspects of the model for a cool trivia maze.

- I created around 180+ questions,
- Made the game installable via docker
- Connected the database to the view
- Made the controllers
- Handled the door functions
- Made the UML Diagrams

Problems we had to overcome:

Since Godot doesn't inherently have the MVC Pattern, we had to do what we could to organize our project to match up with that pattern as much as possible.

Unit Testing was a huge pain for Godot, as it is an extension we had to include in our project called Godot Unit Testing (GUT). We ended up with sufficient tests, however.

We struggled with making our game installable via docker. He had github issues regarding file sizes surpassing the limit for commits. The build of the game initially didn't even load the questions for some reason, leaving just the menu with a button that's always correct. We had issues with docker itself taking a long time to handle our changes as well. We were able to

Project Shortcomings:

We wanted to add more features such as items, a map with an interface rather than coordinates, and a hard mode. The hard mode would've had a timer system, and hard mode exclusive items to handle the timer, but a lower chance of finding items. Of course, the maze would be bigger as well for hard mode. However, due to the time we had, we weren't able to get to those. Other additional UI would've been nice too, such as having a sprite that would appear that would display the controls and would disappear as soon as the player moves, or a pause button that would sit in a corner of the screen so pressing escape isn't the only option. Resolution size handling would've been nice as well, as Renzo's TV monitor had the game too large for the screen for some reason.

Extra Credit:

An overwhelming majority of our assets were produced strictly by team members.

All music was composed by Renzo, most art was drawn by Renzo as well.

Renzo has composed the soundtrack consisting of 5 tracks, that being the following: MainMenu, SaveSelect, Victory (Win Scene), Defeat (Loss Scene), Maze (In Game Maze Scene).

The art Renzo has drawn consist of the following: All door assets (Correct, Incorrect, Question), all tiles used for the tilemap (all rooms), character design + spritesheet animations, Main Menu

Art, Win Scene Art, Loss Scene Art, Trash Can sprite, Win Sprite (the sprite that appears at the center of the maze).
Shiraz also made the logo.
We also included funny sound effects we gathered from memes. The sound effects are the only assets we did not originally produce.
We thought of all of our questions, having our database consist of 200 Questions.
Since we not only have custom assets that we made, those assets were used to make a GUI.
We also made the game installable via docker. Instructions on how to install are provided in the readme.md file.
We added a few quality of life features such as a run button, and having the player teleported to the next room immediately if the player answers a question correctly.