**Code Quest**

Mohammed Al-Hosni

**Game Concept:** *Code Quest* is a fast-paced, text-based dungeon crawler where players battle their way through programming-themed challenges by quickly typing correct C/C++ answers. It combines rapid decision-making with coding knowledge, offering a fun and unique experience that mixes learning with gameplay.

**Explanation of Novel Idea:** Unlike traditional dungeon crawlers, *Code Quest* adds a timed coding twist—players progress by solving C/C++ questions under pressure. Each encounter requires players to quickly respond to syntax, logic, and concept-related prompts within a strict time limit. The game will mainly revolve around concepts taken in the course so that anyone that has paid attention should be able to beat the game, and as such, the game will also include references and easter eggs from class, making it relevant and relatable to students who have followed the course. By incorporating C/C++ questions as attack prompts, it turns your knowledge and understanding of the concepts and uses them as a weapon to attack the in-game enemies and to navigate the “CSB” dungeon.

The game combines the idea of a dungeon crawler, a typing game like typeracer, and speed based programming. Heavy inspiration for the idea was taken from a cool indie game cool “The Texorcist”

**Outline of Gameplay:** The game is set within a CLI, divided into five interactive boxes:

1. **Bottom Box (Input Box)**: This is where players type their responses. It’s where the players must type their answers as quickly and accurately as possible.
2. **Middle Box (Narration Box)**: This box narrates the unfolding story, including descriptions of enemies, events, and room transitions. It also displays the coding questions and challenges players must answer.
3. **Top Right Boxes**:
   * **Item Box**: Displays the player’s inventory, with items like “Syntax Shield” or “Debug Hammer” that can be used for different bonuses. Certain items can be acquired by beating bosses on through luck by entering certain rooms. They will allow you to skip questions, have typos, slow down time (creating “breakpoints”)
   * **Health/Stamina Box**: Tracks the player’s health and stamina, with each wrong answer reducing health and correct answers boosting stamina.
4. **Top Left Box (Main Game Box)**: This is the largest box, representing the current “room” of the dungeon. ASCII-based visuals will depict enemies, items, and special events, adding a visual layer to the narrative.

A white sheet of paper with black text

Description automatically generated

Example of CLI outline

The game begins with the player entering a CSB, the power is off, but you forgot to take your code with you after saving it, so you must go to the computer your worked with on the 11th floor to retrieve codebase. The player must navigate through rooms filled with different coding challenges, ranging from basic syntax to poiter arithmetic, silly questions such as “C library or Dutch words?” etc. Each enemy is a coding problem—answering correctly damages the enemy, while incorrect answers lead to player damage. The final boss, *Professor Lee* himself, is stopping you from getting back your code… because… haven’t figured out that part yet.

As silly as it is, I think it would be a fun way to remember the concepts we worked on in class, and it should (hopefully, but probably not, we’ll see) be somewhat humorous while we’re at it.

Thanks :)