



UNIVERSITY OF THE PHILIPPINES
CMSC 13 - Survey of Programming Paradigms
First Semester AY 2025-2026
Department of Natural Sciences and Mathematics

Merry Grace T. Potot
Geoffrey Adam A. Tomagan
Member

September 5, 2025
Date of Submission

I. Game Description

The proposed project is a software application inspired by the popular game show “Are You Smarter than a 5th Grade?”, but redesigned to focus on programming concepts. In place of general knowledge questions, this game will challenge players with different concepts, theories, applications, and principles of the various programming paradigms. With more challenging questions at higher levels, each round assesses the player's comprehension of coding concepts.

This project aims to develop an engaging and interactive quiz game that replicates a quiz show experience and include the diverse concepts of multiple programming paradigms into the design and implementation of the software. In addition to being entertaining, the game will strengthen the player's understanding of programming paradigms, including procedural, object-oriented, functional, and event-driven programming. The design will also revolve or utilize symbols around programming, ensuring that both the content and aesthetics stay consistent with the topic.

With this game, we aim to give students and programming enthusiasts an entertaining education resource. This game is a showcase of both technical and creative application, as it will also demonstrate how various programming paradigms can be used in software design.

II. Game Features

The game will include the following features to establish both engagement and functionality.

1. Splash Screen

- A welcoming screen that shows the game title alongside a themed background.
- Background music plays
- A “Click the screen to start” to proceed to the main menu

2. Main Menu

Options include:

- Play – Starts a game session.
- **Classmates** – Shows the different “classmate”
- School record – Shows the names, scores, and dates of all sessions played.
- How to Play – Displays instructions and rules of the game.
- Exit – Closes the program

3. Gameplay Panel

- Classmates Selection – After selecting the Play button, the player will choose an “classmate” that can help the player through the game.
- Category Selection – After choosing the “classmate”, the player will choose a category. Either theoretical questions or programming questions. This is also where the player is required to input a username.

4. Choices Panel

- Question Selection: The player can choose different topics with their corresponding level of difficulty.

5. Question Panel

- Question Area – Shows the current question.
- Questions - The questions will be drawn from randomized categories, with each category containing 10 questions across levels 1 to 5. The 10 questions will be divided into two parts: 5 theoretical questions and 5 practical questions.
- Answer Buttons – Four clickable options (A, B, C, D)
- Scoreboard – Tracks the player's score and progress
- Classmate – The classmate you chose from the start, it is a themed character that provides feedback.
- Debug Tools (Cheat)
 - Refactor - Removes 2 wrong answers from the choices.
 - Console Log (Peek) - Player can see what the “Classmate” (AI classmate or sidekick character) answered.
 - Ctrl + C (Copy) - Player can copy the “Classmate” answer without thinking.
 - Auto-Debug (Save) - If the player gets it wrong, but the classmate was correct, they're automatically saved.
- Option Buttons (Top Left)
 - Retry – Restarts the game from the beginning.
 - Main Menu – Returns to the main menu.
 - Mute Sound – Toggles sound on/off
- Sound Effects – Sound effect triggers depending if the answer is correct or not.

6. Scoring and Levels

- Each question is worth a set number of points
- Final score is displayed when game is over, with messages based on the score
- Each level have different number of points
 - Level 1: 100
 - Level 2: 250
 - Level 3: 500
 - Level 4: 750
 - Level 5: 1000

7. How to Play Panel

- Displays clear and step-by-step instructions

8. Background Music and Sound Effects

- Background music similar to a quiz show plays throughout gameplay
- A short sound effect is triggered on correct or incorrect answers.

9. Game Over Panel

When the game is finished, a pop-up menu appears displaying:

- Final Score – Shows the player's total accumulated score and number of correct answers.
- Play Again button – Restarts the game.
- Main Menu button – Returns to the main menu.
- Message to the player – Shows different messages depending on whether the player won the game or not.

10. Interactive Classmate

- Each “classmate” could belong to a paradigm (e.g., *Procedural Pete*, *OOP Olivia*, *Functional Finn*).
- Their hints match their paradigm (Functional one might give mathematical hints, OOP one might give analogies about objects, etc.).

11. Compile and Run Animation

- After each answer:
 - Show a fake “compiler” terminal (animation) with output:
“Compiling answer...” → “Build successful!” (if correct) or “Error 404: Wrong answer” (if wrong).
- Adds a programming flavor instead of just “Correct/Incorrect.”

III. Game Mechanics

• Control

The game implements mouse-click navigation for ease of use.

• Starting a Game

Click the play button on the main menu to start a new game. The Player will then be asked to choose their classmate, after that the player will choose a category and will input a valid Username. There are two options: theoretical questions or programming questions.

• Answering Questions (Mouse clicks only)

The Player should carefully read the question and choices. Button can be hovered over to be highlighted and clicking it will finalize the player’s choice. There is a timer which counts down for every question and changes depending on the difficulty of the round. If a player does not select an answer on time, it is automatically rendered as incorrect. There would be a pop-up panel which shows whether the answer of the player is correct or not.

• Debug Tools

If a player is having difficulty answering a question, they can utilize the Debug Tools, which can help them.

• How to Play

1. Choose Your Classmate

- At the start of the game, select one classmate. Each classmate represents a different style of programming and can provide unique hints or support.

2. Select Your Category

- Pick between **Theoretical Questions** or **Programming Questions**. Your chosen category will remain the same for the entire game session.

After completing the chosen category, the player will then finish the remaining category

3. Enter Your Username

- Type in a valid username. This will be recorded in the School Records along with your score and rank.

4. Pick a Topic and Difficulty

- Within your chosen category, select a **programming paradigm topic** (Procedural, OOP, Functional, Event-Driven, etc.).
- Each topic can be played at varying levels of difficulty: Level 1- Level 5

5. Answering Questions

- You will be given **10 main questions** and **1 final bonus question**.
- Each question has four choices (A, B, C, D). Click on your answer to lock it in.
- A timer may be active for harder levels. If time runs out, the question is automatically marked incorrect.

6. Using Debug Tools

- Players can use Debug Tools to assist them:
 - **Refactor (50/50)** – removes two wrong answers.
 - **Console Log** – can see what the “Classmate” answered.
 - **Ctrl + C** – copy the “Classmate” answer without thinking.
 - **Auto-Debug** – automatic save.
- Debug Tools are limited. Once they run out, you must rely on your own knowledge.

7. Scoring System

- Correct answers award points based on difficulty
- Incorrect answers deduct the point value of that question from your total score (if no Debug Tools remain).
- If you decide to “return” and stop playing. Your points will not get deducted and you keep your final score.

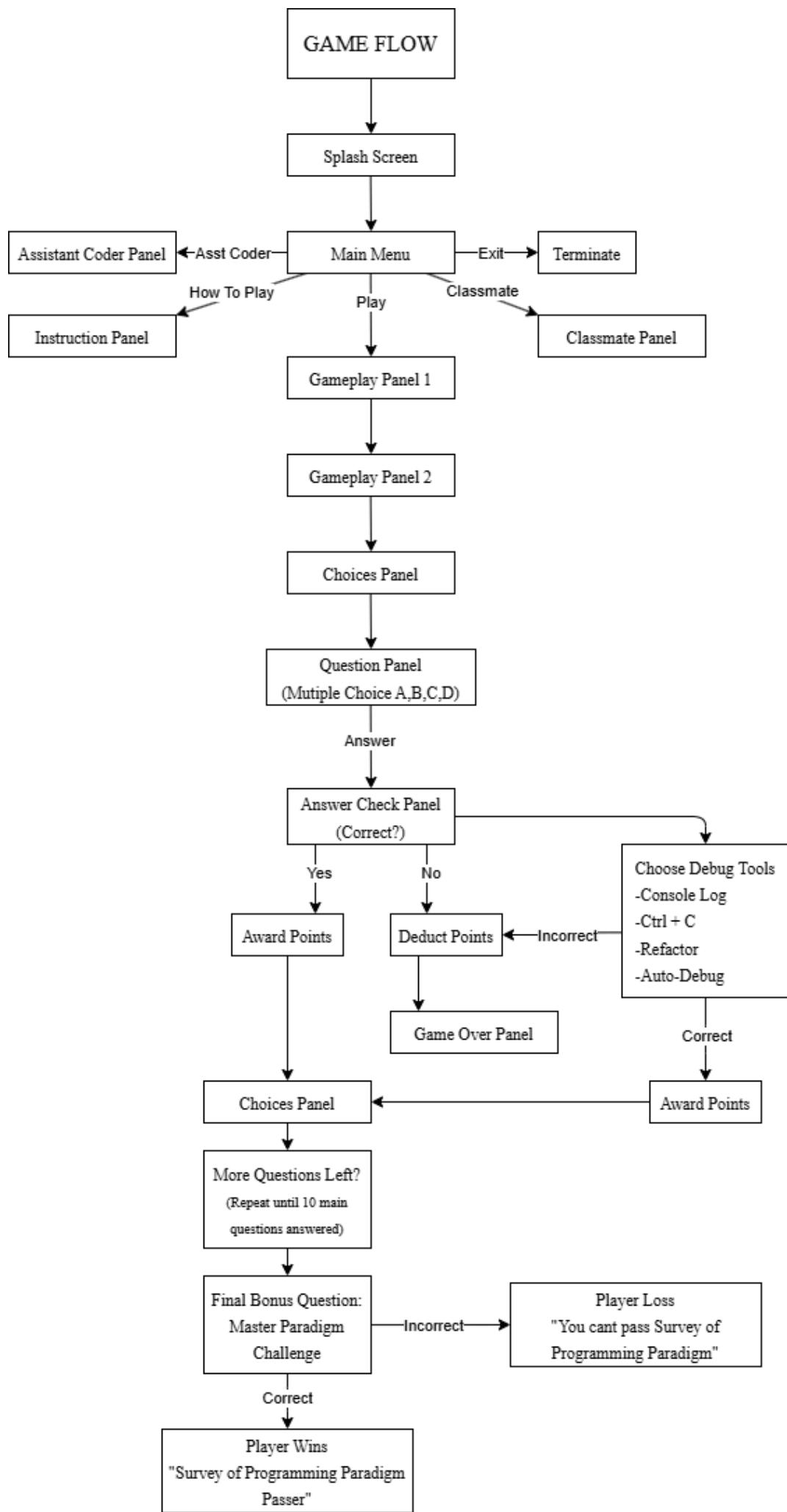
8. Winning Condition

- To **win the game and earn the title** of “*Survey of Programming Paradigms Passer*”:
 - Answer all 10 main questions correctly.
 - Correctly answer the final bonus question.
- The player needs to finish both the theoretical and practical category
- If you fail a question and cannot recover with Debug Tools, your accumulated score will be finalized and recorded.

9. Game Over and Records

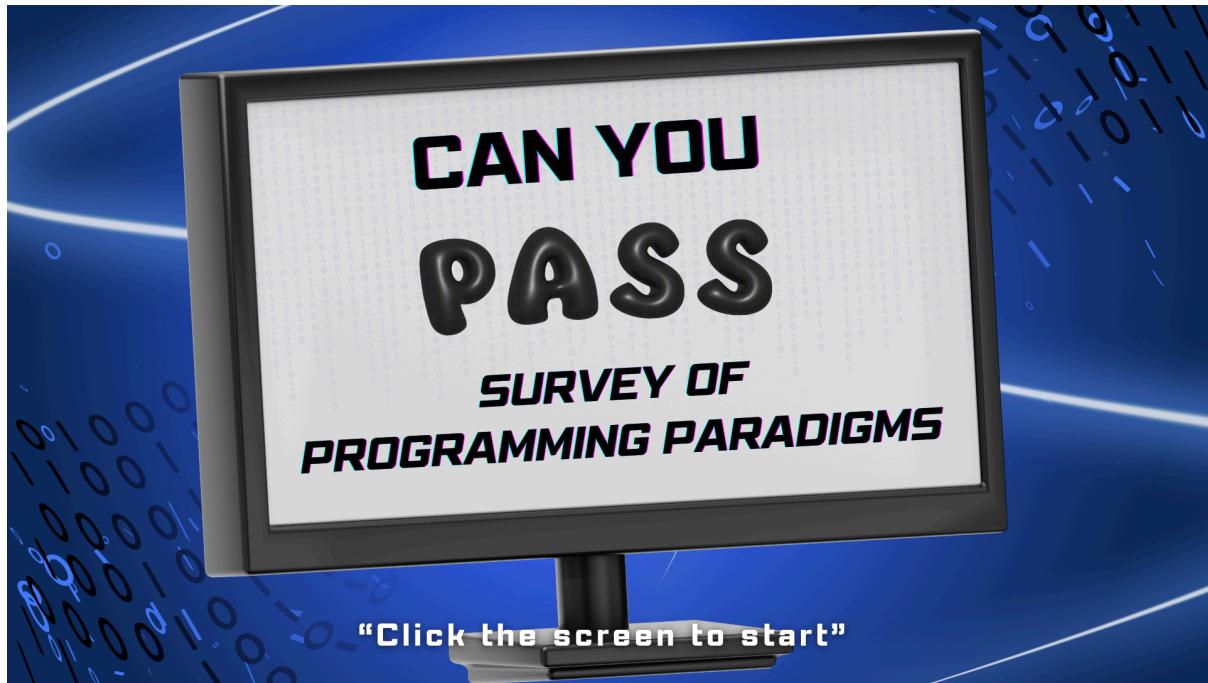
- At the end of the session, your final score, username, and rank will appear in the School Records.

• Game Panel Flow Chart



IV. GUI

1. Splash Screen



2. Main Menu



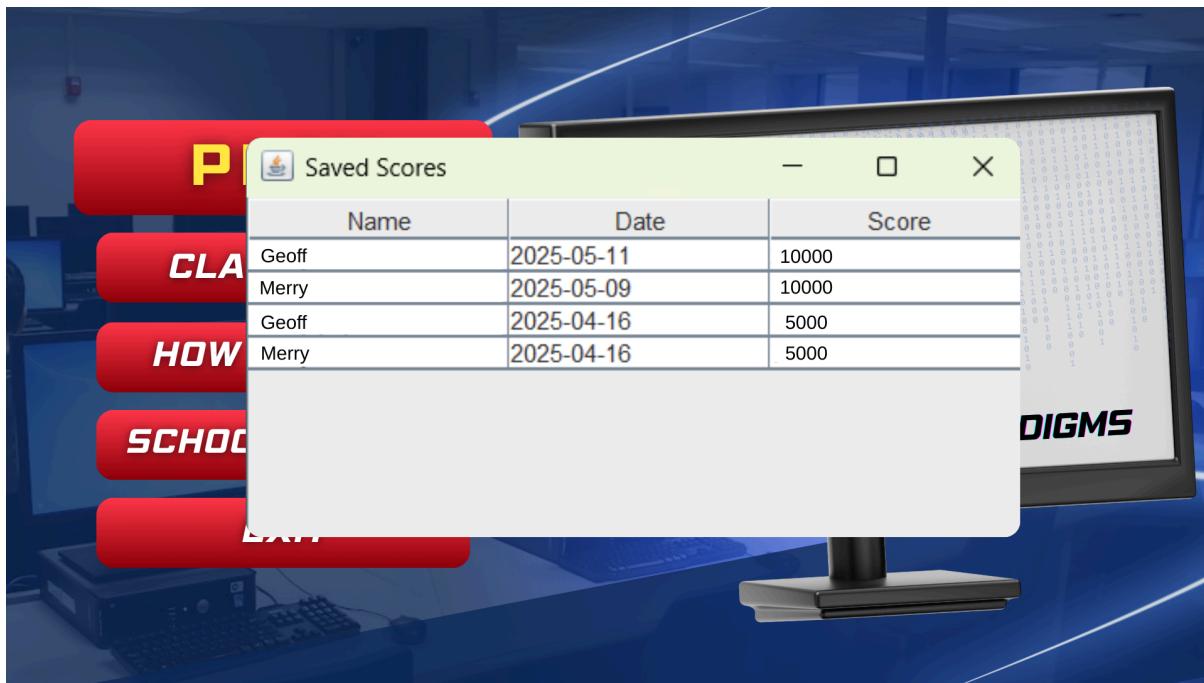
3. Classmate Panel/Gameplay Panel 1



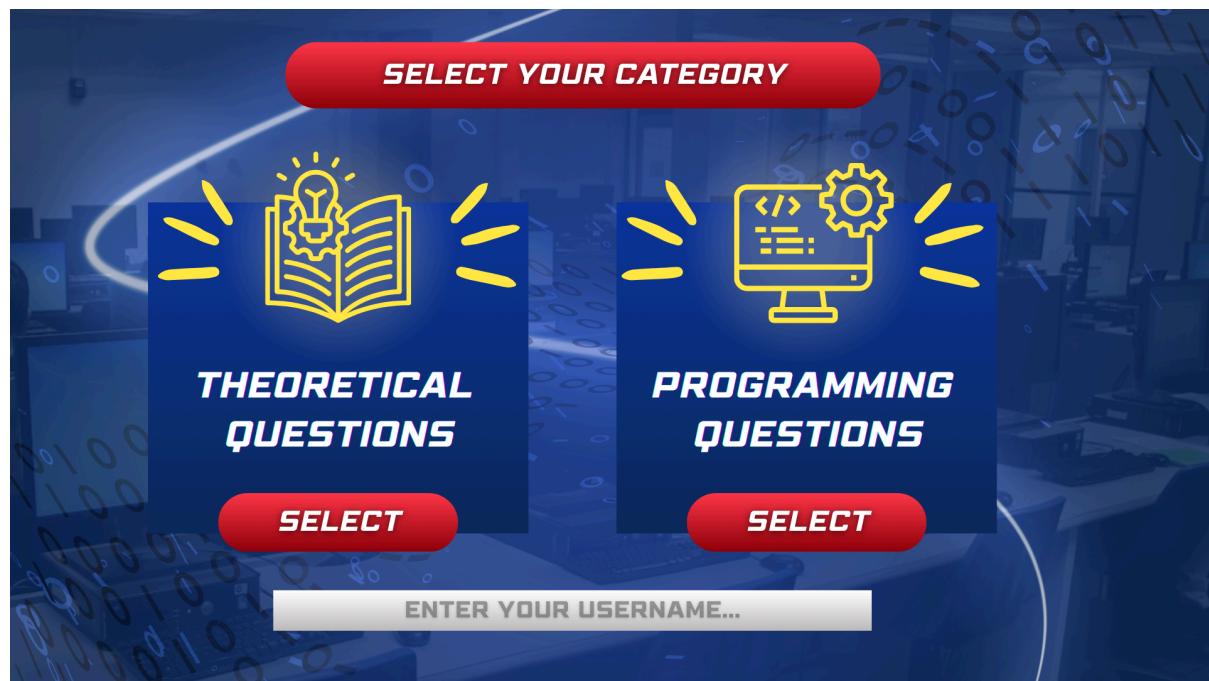
4. How To Play Panel



5. School Record Panel



6. Gameplay Panel 2



7. Choices Panel



8. Question Panel

A screenshot of a mobile application's question panel. The background is a dark blue gradient with binary code. At the top, there are three circular icons: a refresh symbol, a menu symbol, and a speaker volume symbol. To the right of these is a white button labeled "LEVEL 1: PROCEDURAL". Further right is a white text area showing a character named "GEOFF" with a small portrait and a list of five items: "Refactor", "Console Log", "Ctrl + C", "Auto-debug", and "Return". In the center, there is a large gray rectangular box containing a question in Latin: "What is Lorem ipsum dolor sit amet, consectetur adipiscing elit? Maecenas vel lectus sed elit venenatis laoreet. Morbi malesuada lectus nec urna finibus hendrerit." Below the question are four answer options: A. Lorem ipsum, B. Dolor sit amet, C. consectetur adipiscing, and D. Maecenas vel lectus. At the bottom, there are four colored buttons labeled A (red), B (yellow), C (blue), and D (green). The word "SCORE: 0" is displayed in the top right corner.

The screenshot shows a mobile game interface. At the top, there are three circular icons: a refresh symbol, a menu icon, and a volume icon. In the center, the text "LEVEL 1: PROCEDURAL" is displayed in a yellow and white font. On the right, the word "SCORE: 0" is shown above a character named "GEOFF". Geoff is a cartoonish character with brown hair, glasses, and a black hoodie. To his right is a vertical sidebar with five buttons: "Refactor", "Console Log", "Ctrl + C", "Auto-debug", and "Return". The main content area contains a question in Latin: "What is Lorem ipsum dolor sit amet, consectetur adipiscing elit? Maecenas vel lectus sed elit venenatis laoreet. Morbi malesuada lectus nec urna finibus hendrerit." Below the question are four options: A. Lorem ipsum, B. Dolor sit amet, C. consectetur adipiscing, and D. Maecenas vel lectus. Under each option are two buttons: "A" (red) and "Disabled" (gray), and "C" (blue) and "Disabled" (gray). The background of the main screen is a blurred image of a futuristic control room.

What is *Lorem ipsum dolor sit amet, consectetur adipiscing elit?* *Maecenas vel lectus sed elit venenatis laoreet. Morbi malesuada lectus nec urna finibus hendrerit.*

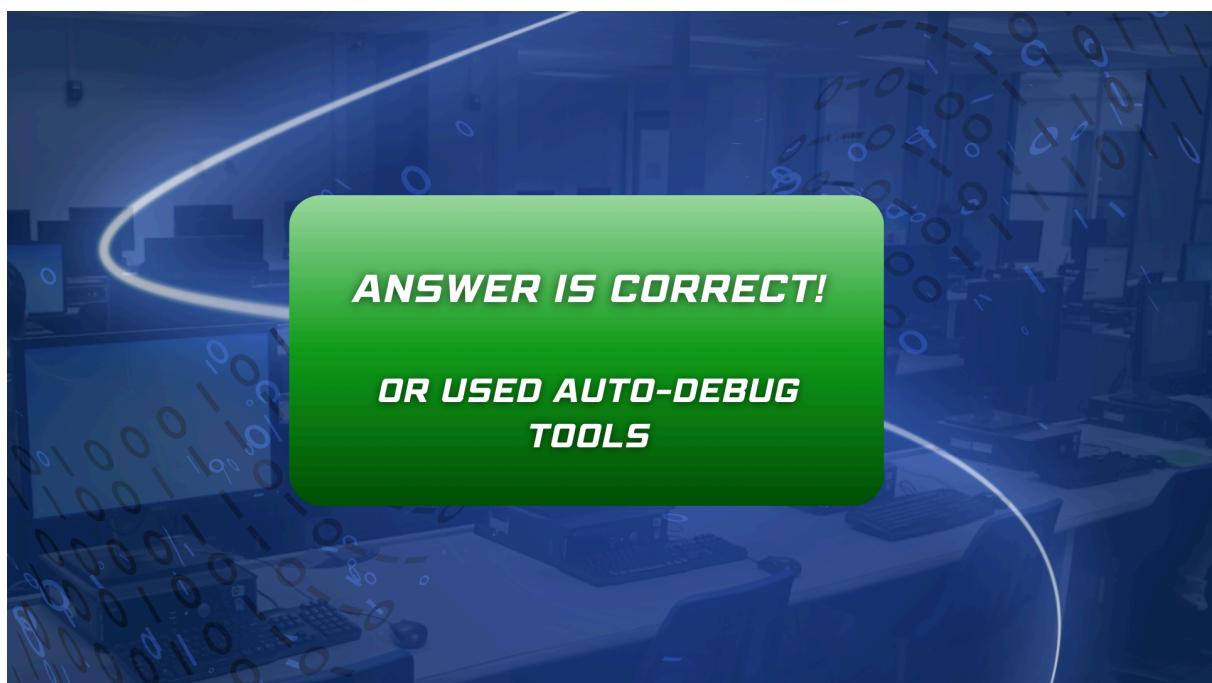
A. *Lorem ipsum* B. *Dolor sit amet*
C. *consectetur adipiscing* D. *Maecenas vel lectus*

A *Disabled*
C *Disabled*

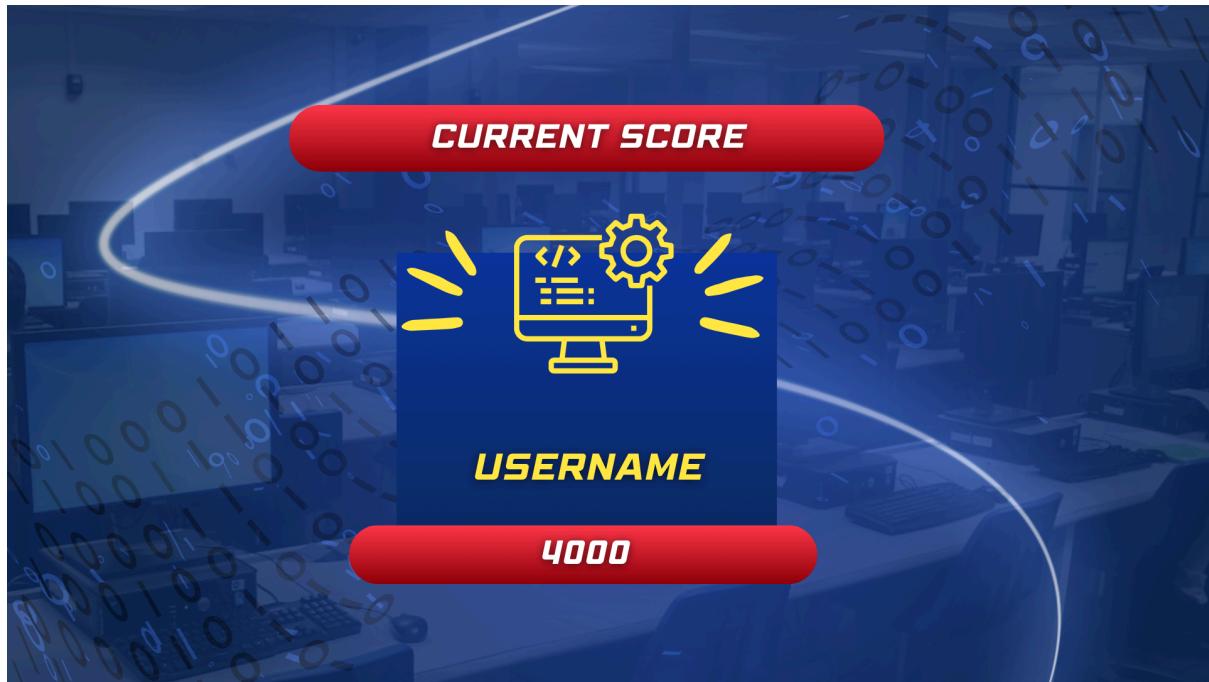
GEOFF

Refactor
Console Log
Ctrl + C
Auto-debug
Return

9. Answer Check Panel

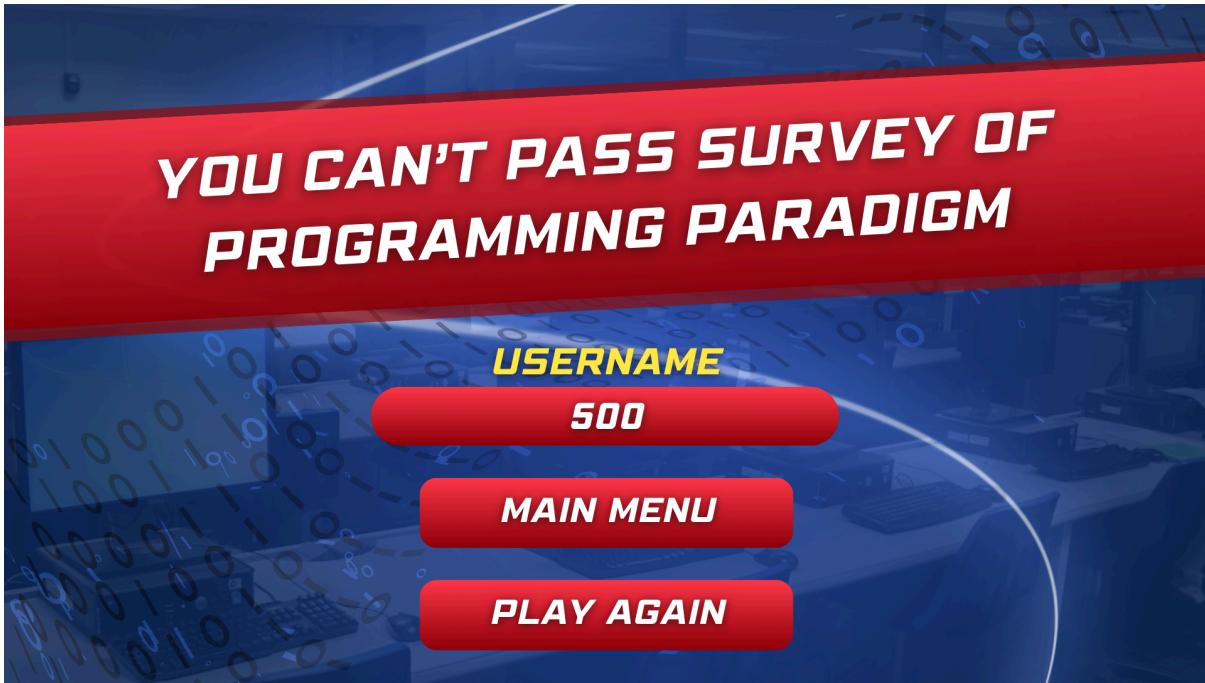


10. Points Panel



11. Game Over Panels





V. Conclusion

This proposed game show converts programming paradigms into an entertaining and instructive interactive quiz format. The game illustrates how various paradigms can be used in practice with features like classmates, debug tools, and visuals with a paradigm theme. We, the students and developers, were able to apply our knowledge and comprehension of the subjects covered in Survey of Programming Paradigms to the design and questions of the game through this project. All things considered, CMSC 13 students will find it to be an enjoyable educational resource that enhances their comprehension of programming and displays innovative software design.