Description of Minigame Project

Sarvesh Sudharshan Prabhu

Why This Project?

I opted for Python, coupled with Tkinter and the webbrowser module, for my minigame project due to Python's reputation for simplicity and readability. Tkinter is a powerful toolkit that facilitates the swift development of graphical interfaces, allowing for efficient prototyping of the game's visual elements. Python's cross-platform compatibility ensures broad accessibility across different devices without the need for additional installations. Incorporating the webbrowser module in Python enables seamless integration of web-based features into the game, contributing to a more dynamic user experience. The extensive support and resources within the Python and Tkinter communities provide a valuable knowledge base, enhancing the overall development process.

Minigame #1: Snake

Snake is a classic arcade game where players control a growing snake, aiming to eat food while avoiding collisions with the snake's own body and walls. In Python, I created a simple version of Snake using the Pygame library for graphical elements and user input. I implemented the game logic to handle the snake's movement, food generation, and collision detection, ensuring a dynamic and challenging gaming experience.

Minigame#2: Hangman

Hangman is a word-guessing game where one player thinks of a word, and the other player attempts to guess it by suggesting letters. In Python, I created a Hangman game by selecting a random word from a predefined list, initializing variables to track guesses and game state, and implementing a loop for user input. The game displays the word's current state, accepts letter guesses, and reveals correct guesses while incrementally drawing a hangman figure for incorrect attempts. The game continues until the player guesses the word or exceeds the maximum allowed incorrect attempts.

Minigame#3: Tic Tac Toe

Tic-Tac-Toe is a two-player game where opponents take turns marking an empty three-by-three grid with their designated symbol, typically "X" or "O." In Python, I created a Tic-Tac-Toe game by defining the game board, handling player input, and implementing logic to check for win conditions or a draw. The program continuously prompts players for moves, updates the board, and declares the winner or a draw when the game concludes. The code involves conditional statements and loops to manage the game flow and ensure a functional and interactive experience.

Cited Information / Reflection

No citations needed for this project. All code solely produced by me. I had a really fun and challenging experience for the 2 months that I have worked on this project. I could have done a lot of things differently if I had started working on the project earlier. For example, I could tried to compile this program into a mobile application or a website. In addition. I could have used other GUIs in python along with tkinter to style the UI better. Also, I could have added additional minigames for a more fun and dynamic experience for the user.