Turtle Crossing

This Crossy Road-style game project is a Python-based application where players control a character, represented by a turtle, attempting to cross a busy road filled with moving obstacles. Built using the Turtle graphics module, this game offers an engaging experience with dynamic obstacles and progressively increasing difficulty levels. The main objective is to navigate the character across the road without colliding with moving cars. This project showcases a blend of game development concepts, object-oriented programming (OOP), user input handling, basic AI behaviours, and data persistence.

The motivation behind this project was to create an interactive and enjoyable game that would allow me to deepen my understanding of programming concepts such as collision detection, game loop mechanics, and dynamic object generation. By building a game from scratch, I aimed to sharpen my skills in Python game development and create a visually engaging application that would test a player's reflexes and strategic thinking.

The game features user input control through the arrow keys, enabling players to move the turtle character up, down, left, and right. As the player navigates through the road, cars are generated at random intervals with varying speeds and colours. This dynamic obstacle generation involves randomization and object-oriented design, as each car is treated as an individual object managed by a `CarManager` class. To enhance gameplay, basic AI behaviours were implemented, such as random lane-changing and speed adjustments based on proximity to the player, adding an extra layer of unpredictability and challenge.

A critical part of the game is collision detection. If the player comes too close to a moving car, the game ends, providing immediate feedback and raising the stakes for each move. The game also incorporates a level progression system, where each successful crossing increases the player's level. As levels progress, the cars move faster, increasing the difficulty and rewarding player skill. High score tracking is integrated, with the current level and high score displayed on the screen and saved to a file for persistence across sessions. This required the use of file handling and data management techniques.

Throughout the development of this project, I encountered and overcame challenges such as balancing game difficulty, optimising real-time obstacle movement, and ensuring smooth user input response. Building this game allowed me to apply and deepen my understanding of game logic, user interaction, data persistence, randomization, and basic AI behaviours. It demonstrates my ability to create interactive applications that are both challenging and enjoyable, reflecting my commitment to enhancing user experiences through thoughtful programming and design.