In this project, I created a Flappy Bird–style game where the player controls a pigeon navigating through pipes. This project gave me the chance to explore how games can be built using classes, objects, and loops. I was able to understand better how organising code into classes makes it easier to manage and expand. Thinking in terms of classes enabled me to view the game as a body of independent objects interacting with each other.

I learnt to implement game states like ‘start’, ‘playing’, and ‘game over' using if/else statements, allowing the game to behave differently depending on the state. For instance, the start screen shows instructions and pauses the game until the player presses space, while the game over screen shows the player’s score and a restart message. Using these conditional statements helped me understand how to control the flow of a game and make the experience a smoother one for the player. Using push() and pop() has also helped me isolate style changes without affecting other parts of the game, allowing me to understand better how temporary changes can be controlled.

Adjusting the bird’s speed and flap strength helped me see how small changes in numbers can affect how the game feels, like making it easier or more challenging. I also worked with arrays and loops to handle multiple pipes, updating their positions and checking for collisions. Learning to use preload() to load assets (i.e., background image, character, fonts) before the game started, and then placing them correctly on the canvas, allowed me to visualise the relationship between visuals and code. Playing with image sizes, positioning, and layering made the game look more elevated.

Overall, this project helped me combine what I had learnt about classes, loops, vectors, and game logic with creative choices.

(298 words)