**Name (unsure): Pusheen’s Adventures**

**TP1 UPDATE:** No updates have been made as of the TP1 deadline.

**TP2 UPDATE:** Changed the name of the project to ‘Pusheen’s Adventures.’

Description: This is a platform side scroller game based on Donkey Kong Country on the SNES. In this game, the player is tasked with maneuvering the main character across a series of enemies and terrain to complete a level. Along the way, the player will encounter different types of terrain, interactable objects, and collectible items.

Similar Projects: Much of the idea for this project is inspired by the platform games developed by Nintendo, most notably Donkey Kong Country and Super Mario Bros. One of the most successful aspects of these games is the amount of autonomy and skill expression it allows despite being based on the fundamental idea of a simple side scroller, in fact, Super Mario Bros. is one of the first ever game to put this concept into use. The player is rewarded for purposeful interaction with the environment and ability to maneuver through different types of terrain; for example, the player can complete a level quicker by utilizing the floating terrain and pipes. By allowing the players to have more of the world during their progression through a level compared to the preexisting platform games at their respective releases, both these games received with critical acclaim and cemented itself as a staple in the SNES era games. For this project, I would like to emulate these aspects of Donkey Kong Country and Super Mario Bros. that make them such an enjoyable experience for both the casual and serious gamer audience.

Structural Plan: The core aspect of the game will probably be divided into 3 main functionalities: the side scrolling gameplay, object interaction, and terrain. The scrolling of the screen can be dictated by a function within the main program, while the main character and enemies can be stored in a class of all characters in the game. The characters appearance can be represented with sprites fetched from an external file. Additionally, the objects and terrain will most likely be stored in classes as well, perhaps in separate files to the main program. This is because the types of objects and terrain of this game have key properties similar to each other. For example, every interactable object could have a ‘carried’ status and call upon an effect attribute when dropped. Similarly, there are mainly two types of terrain: floor and ropes. The floor behaves the same as any basic terrain in most other platform games; the players can stand, run, and jump on the floor. Ropes behave a little differently, as they facilitate vertical movement instead of horizontal movement.

Algorithmic Plan: The hardest part of the project seems to be the implementation of object interaction and physics of rope dynamics. Although I am not exactly sure how these features would be implemented, I have thought of some preliminary ideas. Object interaction would most likely rely on the use of lists and many subclasses. The list would be a relatively simple way keep track of the objects that are currently in the game, while the different subclasses would determine the different types of objects and their properties (e.g., TNT, vehicles, barrels). Rope dynamics would rely on mathematical calculations which factor in the character’s velocity, the rope properties, and the position of contact. For example, the rope will swing more if the character contacts it further away from the pivot point with a higher velocity.

Timeline Plan of Implementation:

* Week 1 (14-20th November): Basic game functionalities such as walking around, jumping on and off terrains of different height, and going up down rudimentary ropes.
* Week 2 (21-27th November): Side scrolling function and object interaction such as carrying, dropping, and riding. Moving enemy characters and other threats. Start/end game features such as goal, lives, and coins; this implies a basic user interface of start and game over screens. The game should be at the MVP stage by the end of this week.
* Week 3 (28th November-4th December): Any backlog from the previous week’s work before the TP2 deadline to add to the MVP. Rope dynamics. In-game aesthetics, mainly character sprites, background, and music. A more refined user interface.
* Final Stretch (5-7th December): Touch-ups to appearance and user experience. Extra object or character features if time allows it.

Version Control Plan: I have created a GitHub repository to deal with backing up versions of my project. A screenshot of a computer

Description automatically generated

Module List: pygame (for playing audio only)