**Name: 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.’ The concept of the rope terrain is replaced with vines instead. The holes and platforms throughout the map are randomly generated at the start of each game, instead of having a set of terrain for specific levels.

**TP3 UPDATE:** No longer using any external modules. Rope/vine dynamics are no longer in the implementation plan. The power ups in the game are size-up, speed boost, teleport, and guns; the power ups are scattered randomly across the map. Added a mud terrain along side the floor and vines. Platforms now randomly vary in height and length. Enemies jump randomly once the player is nearby, and also now vary in sizes.

Description: This is a platform side scroller game revolving around a fat cat called Pusheen. In this game, the player is tasked with maneuvering Pusheen across a map to complete a level. Along the way, the player will encounter a series of different types of terrain, interactable objects, and moving enemies.

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, power ups, 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 vines. The floor behaves the same as any basic terrain in most other platform games; the players can stand, run, and jump on the floor. Vines behave a little differently, as they facilitate vertical movement instead of horizontal movement. Additionally, a mud terrain that hinders the player’s ability to jump can be placed around the map as well. All of these terrain types will be generated at random locations at the start of the game.

Algorithmic Plan: The hardest part of the project seems to be the implementation of object interaction and the character physics. 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 dictionaries. A list would be a relatively simple way keep track of the objects that are currently in the game, while the dictionary would determine the different types of objects and their properties (size-up, speed boost, teleportation, and guns). The physics of the game would rely on some rudimentary calculations based on the kinematic equations, factoring in the acceleration from gravity, jumping, and running. Doing so, the character can move in respect both axes simultaneously, allowing for parabolic movement.

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 vines.
* 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. In-game aesthetics, mainly character sprites and background. 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: None