Project Description:

1) Name - Python Jump Quest
2) Python Jump Quest is a platformer game written on python using cmu graphics library. Goal of the Player would be navigating through a screen, overcoming obstacles & enemies. Game will encourage players to utilize power-ups and special items to progress.

Competitive Analysis:

There are classic platformers such as Kirby's Adventure and Sonic the Hedgehog. "Kirby's Adventure" is popular for its creative level design and power-ups, while "Sonic the Hedgehog" features high-speed action and time-limited challenges.

Python Jump Quest retains the elements of an addictive platformer. Unlike "Sonic the Hedgehog," the game has no time limit, allowing players to explore at their own pace. Players will collect coins and overcome obstacles without the timebound, which sets the game apart from the other two.

Structural Plan:

The Python Jump Quest project will be organized into several modules to ensure clarity and functionality, utilizing the CMU Graphics library. The main game frame will be initialized inside the def main() function. The `Player` class, defined before the main functions, will handle player attributes, movements, and drawing. The `Level` class will manage the layout of the levels, including parsing block positions and drawing the level elements. Utility functions for tasks such as collision detection and boundary checks will reside in inFrame(app, dx=0, dy=0). Constants like block size, and colors that are not classified in Player and Level Classes will be stored in onAppStart(app) and reset(app) functions.

Algorithmic Plan:

The most challenging aspects of my project now are implementing side scrolling and generating random blocks dynamically:

For side scrolling, I will enhance the onKeyPress function to adjust the view based on the player's movement. This will involve adding an additional variable, app.scrollX, to track the horizontal scrolling offset. By modifying the rendering logic to account for app.scrollX, I will ensure a smooth scrolling experience as the player moves through the level.

Additionally, to replace hardcoded blocks with randomly generated ones, I will utilize the random library. I will define specific criteria for block generation to ensure they are placed logically and enhance gameplay. This will involve creating an algorithm that randomly positions blocks while adhering to these constraints, thus providing a more dynamic and engaging game environment.

Timeline Plan:

I plan to complete all the main framing features, such as side-scrolling and integrating the background image, before the start of the one-week break, specifically by Saturday. In the first part of the break, I will focus on adding new characters that present challenges to the player, as well as implementing a coin collection system. In the second half of the week, my focus will shift to enhancing character development, incorporating additional onKey features, and increasing the overall complexity of the game's functionality.

Version Control Plan:
The whole code will be uploaded to Github for the purpose of backup as well as additional images. All updates and new versions will be uploaded there too.

Module List:
I am not planning to use any additional modules.

TP2 Update:
No.

TP3 Update:

- I have incorporated additional enemies (shooting) and power-ups (mushroom), both of them are positioned on floating blocks.
- My coins and star power-ups are now in constant motion, up & down and from left to the right respectively.
- I've enhanced the player character with animations and images for a more engaging visual experience.
- The player now has three lives, which are displayed on the screen.