Helix Jump Project

TP2 update:
There are no changes made to my design

TP1 update:
Rather than creating a 3d game, I plan on creating a 2d one. This means my project will not have a helix like shape, rather it will just look like a tower.

Project Description [2.5 pts]: The name of the term project and a short description of what it will be.
Name: Helix Jump
Description: This is a game where players guide a bouncing ball through a helical tower maze, avoiding obstacles and navigating through gaps by rotating the structure.

Similar projects [2.5 pts]: A 1-2 paragraph analysis of similar projects you've seen online, and how your project will be similar or different to those.
I am recreating the online game Helix Jump. The biggest similarity between our games are the main functionality of the game and its rules. Unlike the original, my version will incorporate various elements like booster shields and collectible coins to enhance the gaming experience. Additionally, I'm considering the inclusion of an anti-gravity section, where the ball doesn't jump, requiring players to navigate it in all directions. Taking inspiration from Isabella Shi's Flappy Bird 112 project, I intend to follow a similar approach by adding unique features to make my Helix Jump game stand out. I will be creating a UI that looks different from the original game, and something that matches my interests more.

Structural Plan [2.5 pts]: A structural plan for how the finalized project will be organized in different functions, files and/or classes.
Classes I will create:

1. Game status:
    a. Includes functions that keeps track of scores, current level, and whether the game is lost or not
2. Tower:
    a. Includes functions that manages layout and rotation of the tower
    b. Function that checks collision
3. Ball:
    a. Includes functions that manages the location of the ball
    b. Includes functions that manages the speed of the ball (applying gravity)
4. Obstacles:
    a. Includes functions that manages the forbidden areas that the user must avoid
5. PowerUps:
    a. Includes functions that manages the various different power ups available for the user to use
    b. Changes speed, creates protections against losing
    c. Collect coins

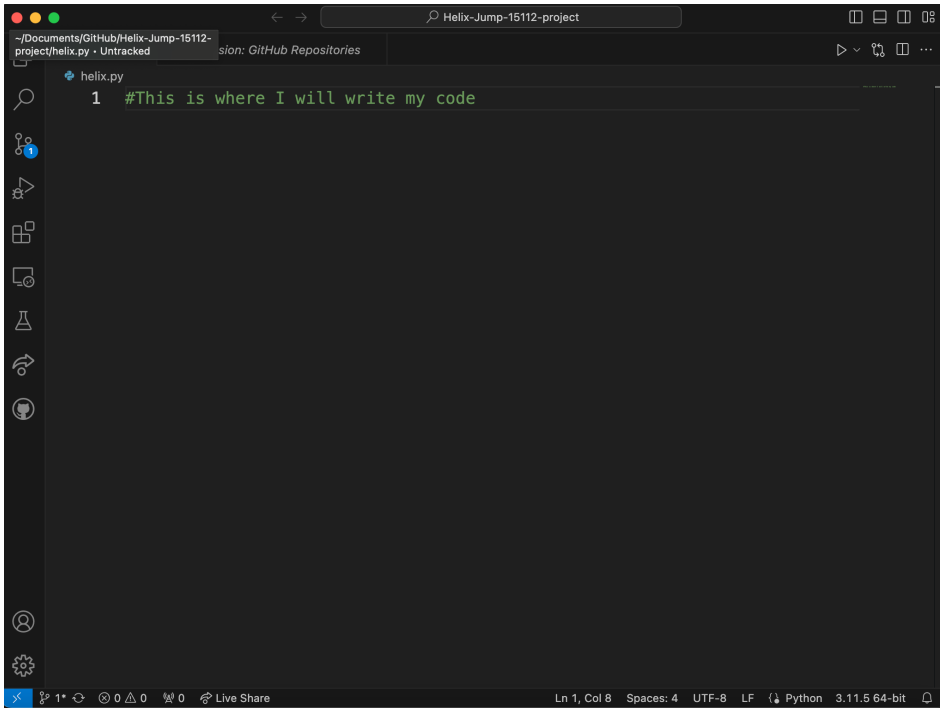
Algorithmic Plan [2.5 pts]: A plan for how you will approach the trickiest part of the project. Be sure to clearly highlight which part(s) of your project are algorithmically most difficult, and include some details of how you expect to implement these features.

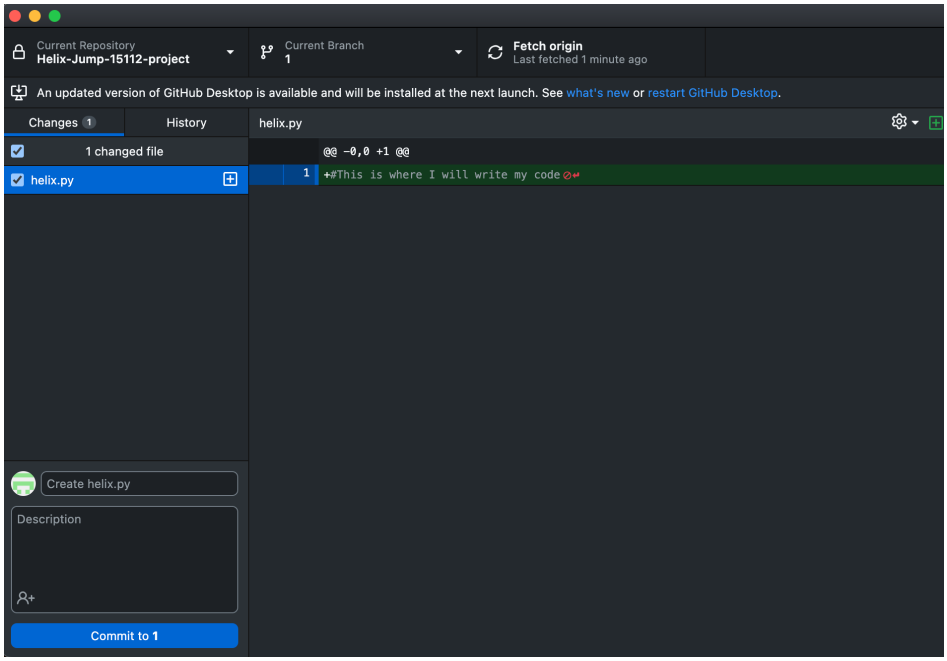
One of the trickiest parts of my project will be simulating realistic ball movements and determining collisions within the game. To create the ball's movement, I will need to consider gravity, bouncing, and velocity changes.I plan on researching numerical integration methods to figure out how I can best update the ball's position based on its velocity and acceleration. Additionally, I will need to consider the helix tower's rotation when updating the ball's coordinates. In order to approach collision detection, I plan on creating a function that constantly checks if the ball and the tower have any intersections. Then I would have a different function that determines what the ball has collided with, and determine whether the ball should bounce or if the game is over.

Timeline Plan [2.5 pts]: A timeline for when you intend to complete the major features of the project. (16 days in total)

- **Days 1-2: Project Setup and Planning**
  - Create a game design and outline
  - Research ways to approach the physics and animations of the game
- **Days 3-6: Physics and Basic Gameplay:**
  - Implement basic physics for ball movement
  - Create the helix tower
  - Create collision detection
- **Days 7-11: User Interface and Score**
  - Score display + words display
  - Level change
  - Highest score tracker
- **Days 11-14: Visuals**
  - Animations for balls, platforms, and level ups
- **After:**
  - Testing my code and refining it
  - Fixing bugs

Version Control Plan [1.5 pts]: A short description **and image** demonstrating how you are using version control to back up your code. Notes:
I will be using GitHub to backup my code from VS code. Whatever changes I make to my original code will be then saved and updated to my github repository.





Module List [1 pts]: A list of all external modules/hardware/technologies you are planning to use in your project. Note that any such modules must be approved by a tech demo. If you are not planning to use any additional modules, that's okay, just say so!

- I am not using modules