### Proposal

1. Project Description

The project is named “Jumping to the Moon.” A reference to Bruno Mars’ song “Talking to the Moon,” the title conveys the project’s essence: a game in which the protagonist has to jump all the way up to the moon. Starting at ground level, or maybe someplace else, the character has to jump his way up buildings, mountains, magnetic drones and windy heights until he reaches the moon. There are multiple checkpoints throughout the map in which the user can save his or her work.

In the game, the movement (jumping motion) of the character will accurately follow the laws of physics acting in each situation. Beyond the basic laws of velocity and acceleration, special conditions such as magnetic force, air resistance, varying gravitational constants, and movements in fluid–such as water–will be implemented. On the side of the screen, velocity, acceleration, and equations of forces that are acting upon the character will be visualized.

1. Similar Projects

I was originally inspired by the game “Jumpking.” In the game, the character has to jump his way up a huge map, comprising building interiors, drainpipes, tree trunks and many more. Reaching the end of the map, the Princess awaits, and the objective of the game is to reach and rescue the princess on the top of a castle. The game is famous for its notorious difficulty and tedious gameplay, and the catharsis of clearing such a game.

While “Jumping to the Moon” is similar to “Jumpking” at first sight, there are various factors that differentiate the two. “Jumping to the Moon” puts its focus on its variety of maps and modes that each require different physical rules. Unlike “Jumpking” which requires monotonous, repeated motions of jumping and thus has a very narrow and committed fanbase, the change of pattern in between each map will make “Jumping to the Moon” more enjoyable. The game will even be educationally beneficial since it is also a tool to visualize physical motion in different conditions.

1. Structural Plan

1. Laws and information for the character’s motion will be stored in class Frame. In class Frame, there will be methods for calculating the displacement, velocity and acceleration after a certain time. Also there will be a list–or another data type–that represents what the frame’s map looks like.
2. Since for each frame, the map, environmental conditions, and physical laws will vary, each frame will be an instance of class Frame.
3. The game will be based on and visualized through cmu 112 graphics.
4. Leaderboard and high scores, user information will be stored in an external text file.
5. Algorithmic Plan

The trickiest part of the project will be implementing multiple laws of physics into the motion of the character. I will make a separate class that deals with disposition, velocity, and acceleration, and give all necessary input parameters such as gravitational constant and magnetic force.

1. Timeline Plan

~ 11/18: finish top-down design of project, an outline comprising class names and function names

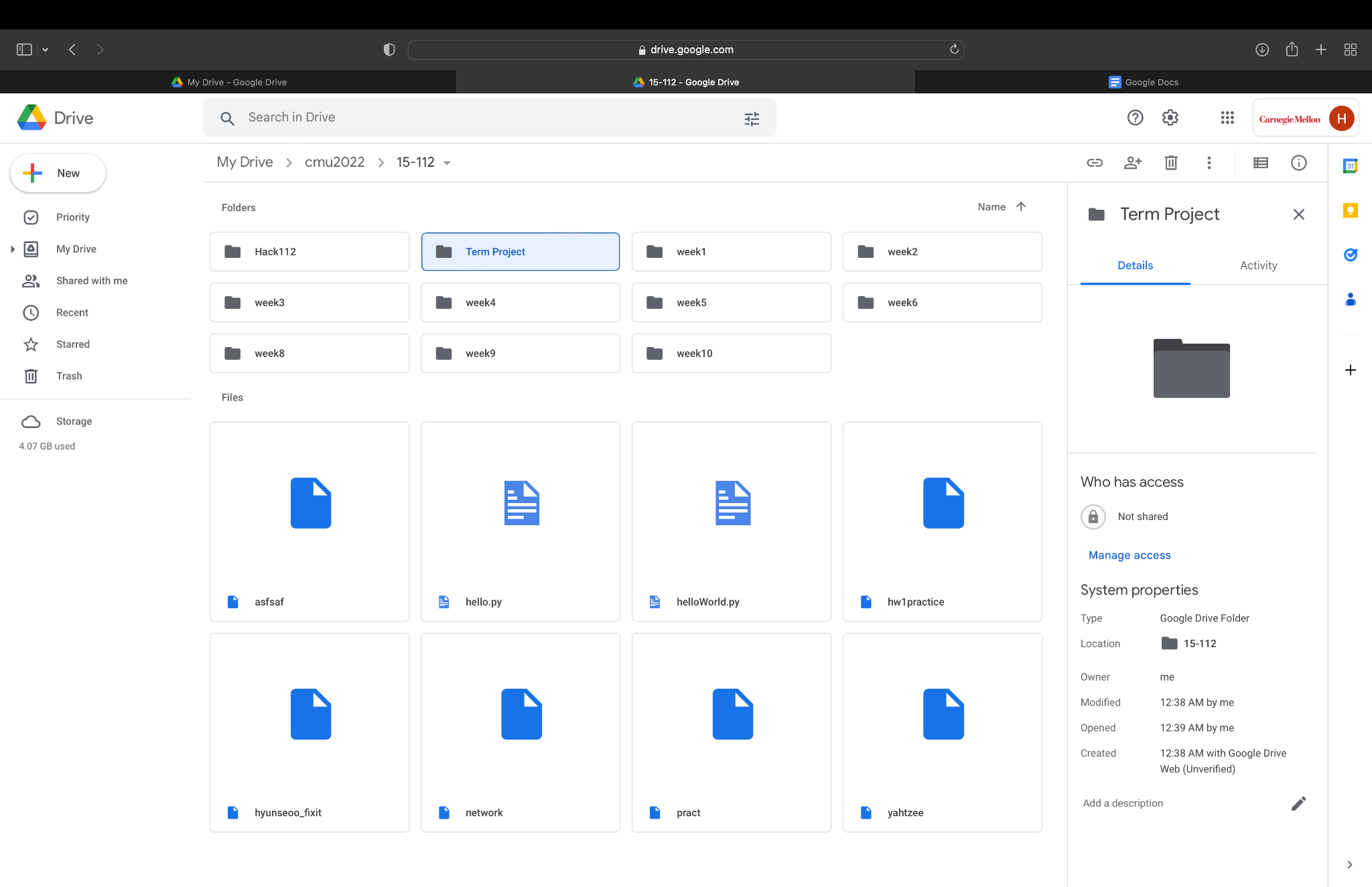
~ 11/22: write codes for classes

~ 11/26: design and work on animation code

~ 11/30: combine all parts together and debug for MVP (TP2)

~ 12/7: take TP2 feedback and improve code

1. Version Control Plan



Each version of the project code will be saved as a backup file once a day, into my google drive folder “Term Project.”

1. Module List

No external module will be used for the project.

1. TP1 Update

No changes were made.