**Name of Project**

**Benjamin Cao, Rizwan Sadique**

# PROJECT PROBLEM STATEMENT

Create a simple platforming game where you need to jump onto platforms and avoid obstacles to get to a certain point, moving on to the next level.

# CONCISE DESIGN OVERVIEW

Develop a prototype or model of what your final solution would look like, providing abstract information about the individual components.

For our main technologies, we are going to use Python to script our prototype, using mainly the libraries of Pygame and Pyglet. For the components of our solution, we are going to create simple platforming levels, that will have an end objective to reach to get to the next level. We will first create four initial platforming levels, each with obstacles, backgrounds, and platforms to jump on for the player. The player’s properties initially will be that the character will be able to jump up and down through platforms, as well as moving side to side. We will also create a final level, which will be more difficult than the previous four, and will have its own unique design and background.

# SCOPE STATEMENT

SMART goals or objectives would include deliverables. Clarify as necessary what the scope includes and does not include.

Conceptualize a simple game with a simple plot, a main character, and setting by 5/24.

Design a character model draft by 5/26

Design 4 obstacles/dangers by 5/28

Design platforming parts of “regular” levels by 6/1

Design backgrounds for regular levels by 6/2

Design platforming parts for final level by 6/4

Design background for final level by 6/5

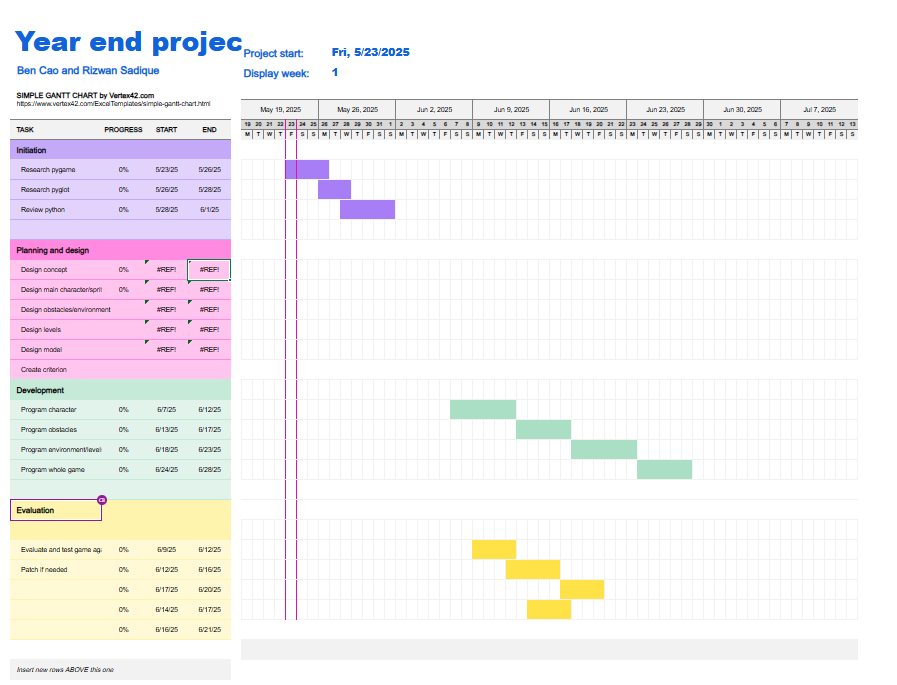
Program player functionality by 6/7

Program platforming/level functionality 6/10

Program obstacles/dangers by 6/12

# TIMELINE

Draw a timeline, including milestones to serve as the basis for a work breakdown structure (WBS) and appropriate Gantt or PERT charts. For this class, you can refer to the Work Breakdown Excel Sheet. This can be a table with anticipated tasks listed for each school day of the project. You will submit the completed chart(or Record of Tasks) along with your final deliverables.



# TEST PLAN

Describe how you will test your solution. The plan should include testing the whole solution and testing individual components of the solution. Include details such as if you will automate testing or you will manually test with positive and negative test cases.

We will manually test each individual component in a testing level. Then, if all tests are passed, we will test the whole solution. Starting with player functionality, we will first test to make sure that the character’s properties function correctly, by checking if the player can move up and down and side to side with Pygame functionality. We will also test to make sure the character is limited by the window’s borders. Then, we will make sure the player can platform correctly, by checking if the player does not fall through a platform once the player is on top of it. We will test to make sure that when the player hits obstacles, the player is forced to restart the level.

# RISKS AND CONTINGENCY PLANS

List known and potential risks by estimated probability, with mitigation plans.

Game does not save progress (High probability) - Search the internet for help and troubleshoot