Blake Freer

Program Description

- My project will be to create the framework for a 2D game engine.
- I will be referencing the Unity game engine to structure my engine. My engine will be different as it will not have a graphical scene editor (all GameObjects will be created through scripts) and will be a 2D engine, whereas Unity is a 3D engine.
    - I want to create some graphical world editing tools, such as a tilemap editor, to make the world creation process smoother.
- Once I have created the necessary framework, I will make a basic game using the components and classes that I have made. It will likely be a platformer game, similar to Mario.

Programming Concepts (in addition to basic ones like arrays and loops)

- Inheritance: All objects will inherit some methods and properties from the GameObject class, and all object components (renderers, colliders, custom scripts, etc.) will inherit from the general Component.
- Queues and Sorts: To draw the objects in the correct order, I will use a queue that holds all of the rendering jobs, which will be sorted based on their layer height.
- 2D Graphics and Swing: I will be implementing my own methods for drawing objects to the screen. I will use Swing for game UI elements.
- Recursion: The Game World will be organized as a tree, where each object is the child of another one. To update all of the objects, a recursive algorithm will be used.
- Listeners: In order for the user to control the game, I must be able to handle input
- Packages: I will be creating too many classes to hold in one package, so I will use other packages to group content.
- Files: To store level creation data, I will use files (ex if I use tile sprites to make the world, they could be saved and read as an integer array, referring to the type of tile in each grid cell)

Research

- I will need to research Vector math operations and Vector class implementations in order to represent objects' positions and velocities on screen.
- Collisions: I will research methods for detecting collisions of circular and rectangular shapes.
- Raycasting: This will involve detecting collisions between a line and the edge of a collider, but I need to research more.