**Galaga**

Plan #1: if I can, I am going to try and make Galaga which is basically space invaders

How the game works:

You control a spaceship that can shoot. It can move right to left and a tiny bit up and down staying at the bottom of the map. I am going to limit the shooting speed.

There are aliens that come down and shoot lasers at the player.

You can shoot lasers back at the aliens and the aliens die after they get hit by 1-2 lasers.

You get points by killing the aliens and you lose a life if you get hit by a laser.

You have three lives or maybe I will make it five.

Aliens spawn in from the side and go to a tile in the top half of the screen.

*Objective is to get the most points possible and not die. Or try to win by killing all the waves of Aliens.*

*Gameplay is a fast shooter that has a spaceship shooting aliens.*

External API:

I am going to add an Event API to allow the game to understand what the user is trying to do listening to the key presses.

I am going to add java swing API to allow for a screen to appear before the user and to utilize visuals.

I am going to add an image processing API so I can make premade images be used in the game, so I don’t have to lag the game with redrawing the image every frame.

I am adding awt API so I can utilize colors and lists.

I assume I will have to make an API that works with collisions, but I think I can do it without a collision API by comparing x and y values.

Classes:

Galaga:

This holds the main class and allows me to run the game. It makes the display appear and starts the game to be running by calling the other classes

Display:

This extends jpanel and it makes all the visuals and updates and redraws all the other objects in the class. This basically runs the whole game by updating and redrawing all the objects in a loop that goes on until the game ends. This updates all the objects in the game at 60 FPS making everything work fast.

KeyController:

This checks if any key is being pressed and it will make a variable set as true if that key is being pressed. Display will update this every time the loop is laps. It sets the variables true when the key is being pressed and it changes it to false when the key is let go of. (Except space bar for shoot is when you let go.)

Collision detector:

This checks to see if the objects that are passed through are touching each other by checking their y and x values.

Entity:

This class holds all the variables that are the basis of each entity. This is a super class.

Player:

This class extends entity and holds all the variables the player would need holding health, number of lasers, it could shoot at once, position of the player, the images of the player, and when the display tells it to update it will check to see if any keys are being clicked by checking which buttons are being pressed by looking at the variables in KeyController and runs the code accordingly to the button that was pressed. WASD is movement keys and spacebar shoots a laser, so it calls the laser class. This class updates and redraws the image every lap that the loop in the display class runs and it will move at that speed. It will also need a collision detector that checks to see if the player is hitting any lasers.

Laser:

This extends entity class and it goes in the direction that it is called going up if it is called to go up and goes down if it is called to go down. This will check for collisions, and it will move across the screen downwards if the aliens call this and move up if the player calls it. This will be put in a list that gets updated and redrawn in the display while loop every lap.

Alien:

I don’t know if I will make separate alien classes or just one for all the aliens that I make. Currently I am trying to do a blue and red alien. And they will need to hold a damage variable, health, shot intervals, the location of where it is and the location of where it wants to be. I might also make it so the aliens have certain paths that they could take so their flight paths are less static, and they move more. I also make the aliens form in a line, and I will have to make it so new waves don’t fly in the same path ending up in the same location. They also shoot lasers by calling the laser class. They are updated and redrawn in every frame of the game.

Hearts:

This displays the hearts based on the players hearts. This is an object that is updated and redrawn every frame. I will put it in a list that is updated in the display class.

Wave:

Creates and calls the waves of enemies based on if there have been enough enemies killed, enough time has passed, or you reach a certain score. You gain score when killing an enemy.

I might also put score in this class and I might even make a score multiplier that gives you a higher score the less lives you have so a strategy would be to lose all your lives then kill all the aliens with one life.