Game: Plants vs Zombie (simple version)

Project Description:
The game was originally developed by PopCap Games and published by Electronic Arts. I will make a simplified version of that, which will only include a few types of plants, zombies, and two levels (the day level and the night level).
Plants vs Zombies is a tower-defending game, where the player's goal is to successfully defeat waves of zombies using the plants that are provided for each level.

Normal Level:
The player will be provided five types of plants(sunflower, peashooter, walnut and potato mine, kernel pult). The player needs to defend 3 waves of zombies.

Hard Level:
The player will be provided four types of plants(sunflower, peashooter, walnut and kernel pult)
Waves increase (4 waves)

Plants:

- Peashooter $100
- Sunflower $50
- Walnut $50
- Potatomine $25
- Corn pult $100 (each shot takes $50)

Zombies:

- Normal zombies
- Conehead zombies
- Buckethead zombies

Tools/Settings:

- sunlights (randomly drop every 8 secs)
- sun → store number of sunlights (each sun is 25)
- Two waves of zombies(each wave contains 15 zombies)
- shovel → give rid of a planted plant

- GameStarting Page
- 'Game over' page → if player defeat all waves of zombies
- 'Quit' the game → when player presses 'q' or 'Q' on keyboard

Structure Plan:
One file contain all the plant functions

- The class for plants and many subclasses(peashooter, sunflower, walnut, puff-shroom, potato mine and coconut cannon)
- Function for placing the plant on the grid
- Function for removing the plant on the grid
- Function to detect the zombie is on the lawn (for each row) and the plant will attack

One file for all the zombie functions

- A class for normal zombies and its subclasses (conehead zombies and buckethead zombies)
- A function to detect if a zombie is near a plant

One main file for calling all these files

- Contains function for the main page and the playing page
- Important functions like keyPressed, mousePressed and draw, etc.
- Contain function for checking winning and losing

Algorithme Plan:

- Kernel Pult will attack on the same row
- Cattail will can attack three rows(row it's on; row above and row below)

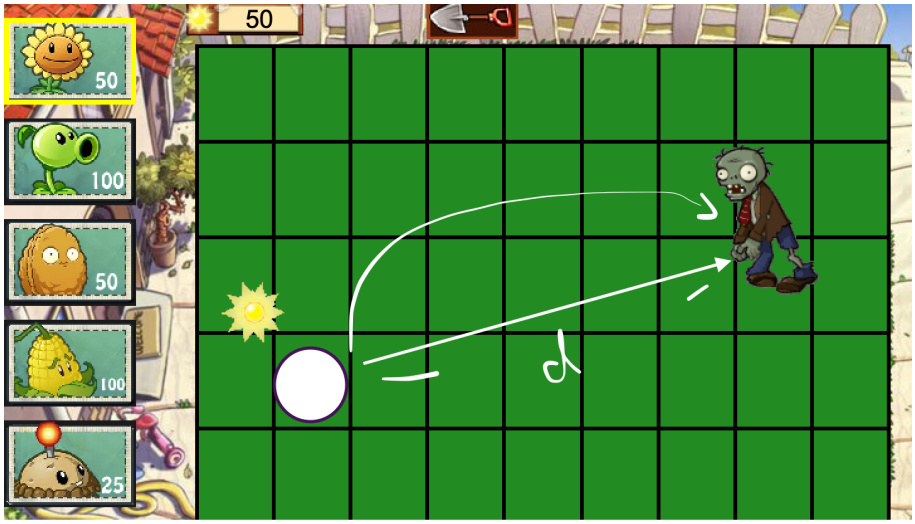
The Kernel Pult will shoot the bullet in a projectile motion. To track the position of the bullet and also draw that on the canvas, we can use the formula

$$x_t = x_0 + v_0\, t\, cos(\Theta)$$

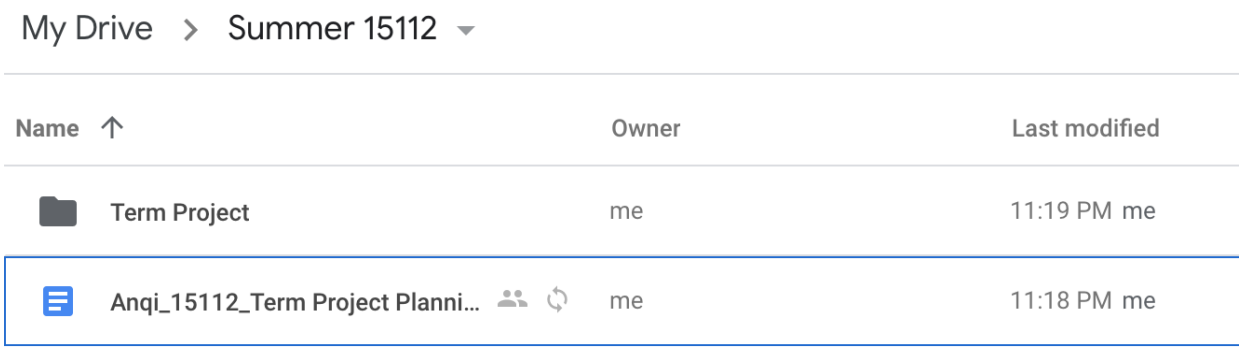
$$y_t = y_0 + v_0\, t\, sin(\Theta) - \frac{1}{2} g t^2$$

To calculate the x and y position at any given time. The g is the gravity (9.81m/s^2); v is the velocity.
We will also use math.cos and math.sin for calculating the angle the bullet is moving. Each plant will be represented in a different number, and kernel pult is represented with number 4. Using the 2d list to create the grid and store the number for each grid. If app.board[row][col] == 4 (that means that grid contains a plant and it is kernel pult), then the bullet image will be launched from that grid to the zombie in a projectile. In addition, if there are many zombies in a row, the bullet will only hit the zombie that is nearest to the kernel pult.

For cattail, it will also attack the zombie that is the nearest (compared among three rows). Its attack will also be in projectile motion. Using the two points distance formula for the ‘d’, and use ‘d’ to find the launch angle. Then we will use the same formula above to find the x and y coordinates of the bullet.



Vision Control Plan:
Backup files will be stored on google folder (school account). Will update my codes every two hours(when working)



TP2 Update:
For complexity, I decided to add two other features after discussed with TA:

1. Zombie intelligence (based on the player’s move, zombie will choose the weak row)
2. Cooling time for the plant card after a plant is being selected

TP3:
Remove cooling time for the card

Added key presses:

- ‘L’ return losing board of the game
- ‘W’ return winning page of the game
- ‘P’ will pause the game