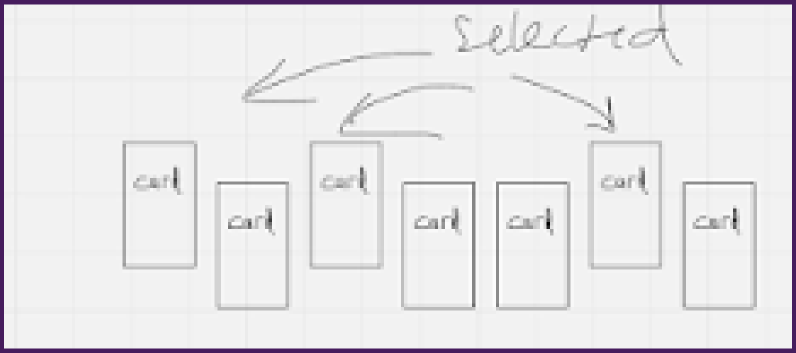
AP CSP Project:
Brainstorm, & Investigate an Idea

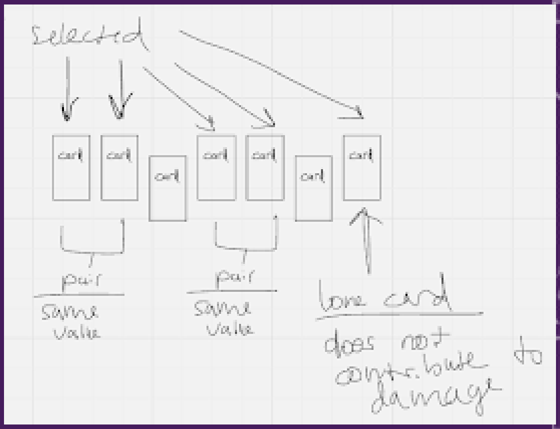
- An RPG game
- There will be a player and enemies
- Player will do damage using combos from playing cards
- User pick the card
    - Damage depends on the card and the combo created from said card
- ~~Can choose power up to modify stats or damage~~

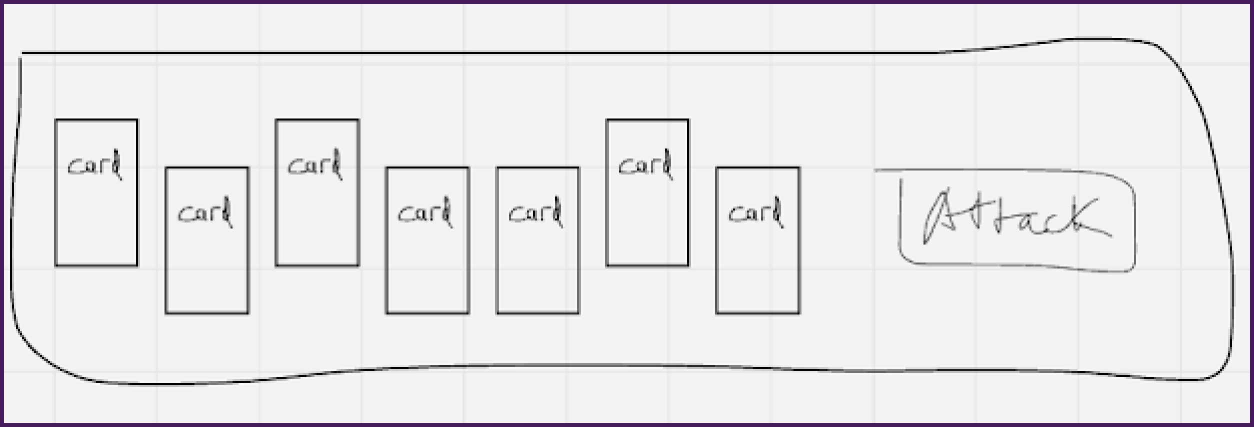
AP CSP Project: Plan

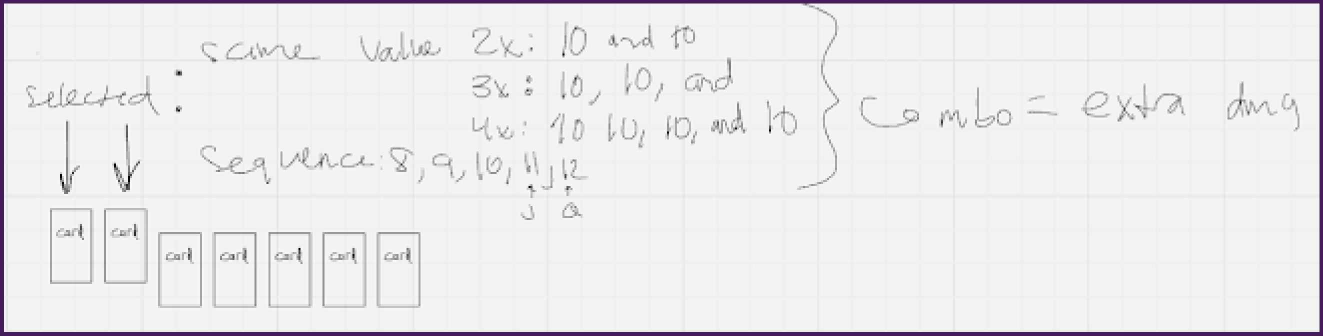
1. We need to create a system to pick the card
2. The card need to be identifiable by its value and suits
3. Build a deck with all the card
4. The card are randomly drawn
5. Indicators for then the card is picked/hovered
6. Generate 8 random cards in hand
7. Each card is rearrange by their value
8. The player can select and deselect each card by clicking
9. When cards are selected, give indication of certain combo if a combination of card is found
10. Calculate the damage based on combo and card selected
11. Discard the cards either after doing damage or clicking the discard button
12. Draw the player and enemy on screen
13. The player and enemy should have animation depending on what action is occurring
    a. Attacking, getting hurt, death, idle
14. After the enemy dies, a map is drawn to show where the user can go
15. The user can click on a point to start the next gameplay loop
16. There are multiple areas that the user can go to
    a. Encounter, Rest, Mini-Boss, Boss
17. Some point can have different option of where the user can go
    a. The user can either battle or rest to regain health
18. When the player dies, a death screen shows up, asking if the user wants to continues or quit

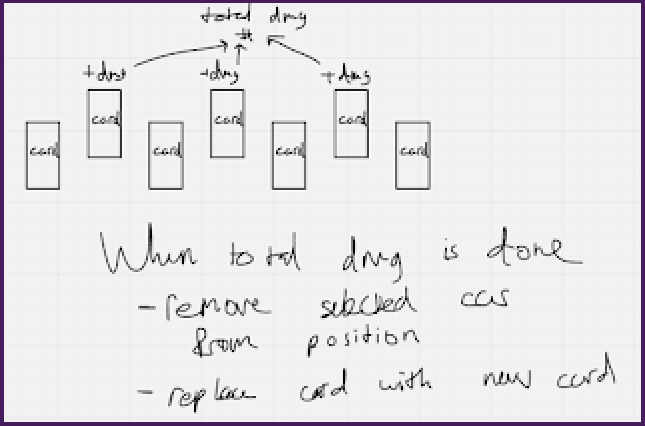
AP CSP Project: Design



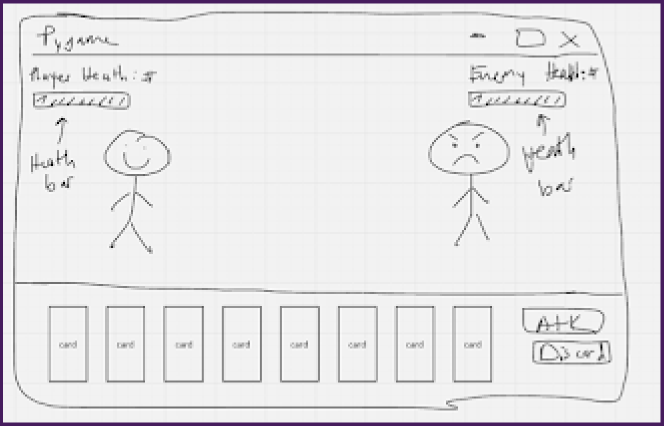




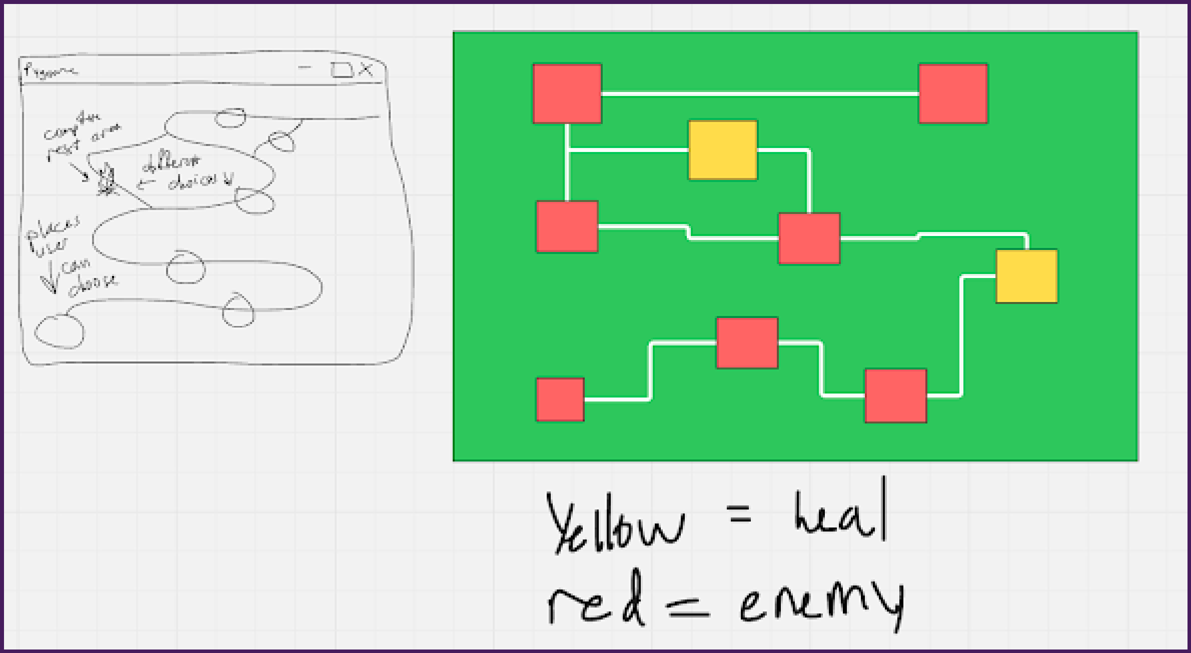




When total dmg is done
- remove selected cards from position
- replace card with new card



AP CSP Project: Design

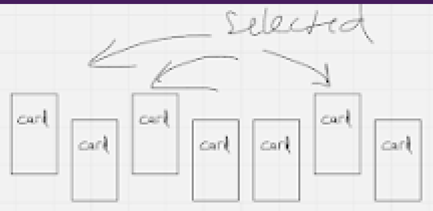


AP CSP Project: Design

Priority	Feature	Est. Time
1	Card System	12 hours
2	Damage / Combo System	8 hours
3	Damage Popup System	8 hours
4	Card Replacement System	8 hours
5	Character Setup - Health, Strength, Health Bar	9 hours
6	Map System	10 hours
7	Character Animations - Including creating sprites	2 days
8	Rest Area (Restoring HP)	1 hour
8	Defeat Screen	1 hour

AP CSP Project: Create and Test

This is where you write your pseudocode:



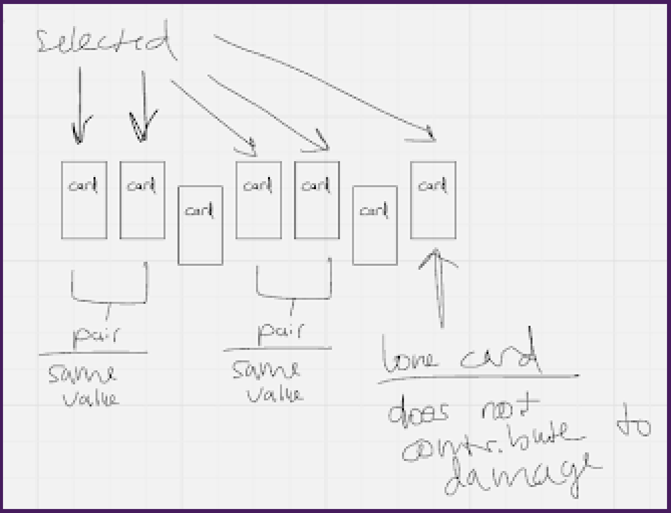
```
when selected:
    append card value in a list
    calculate the value to
        get damage
    remove card, when unselected
```

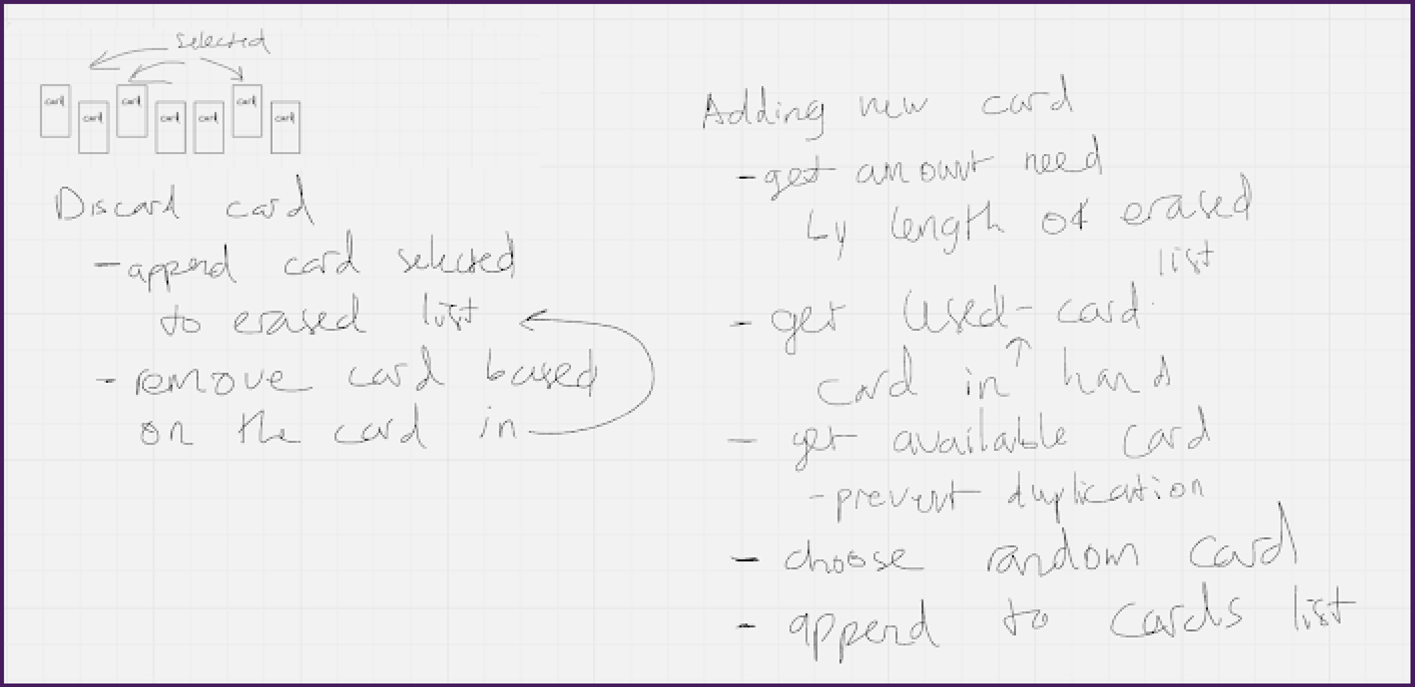
```
Attack

when attack is click
    call function
    calculate the damage
    display damage
```

```
when combo happen, gives
    extra damage:
        - create a list of combo
        - compare selected card to
          combo
        - disregard any lone card
          that does not contribute
          to combo
```

AP CSP Project: Create and Test

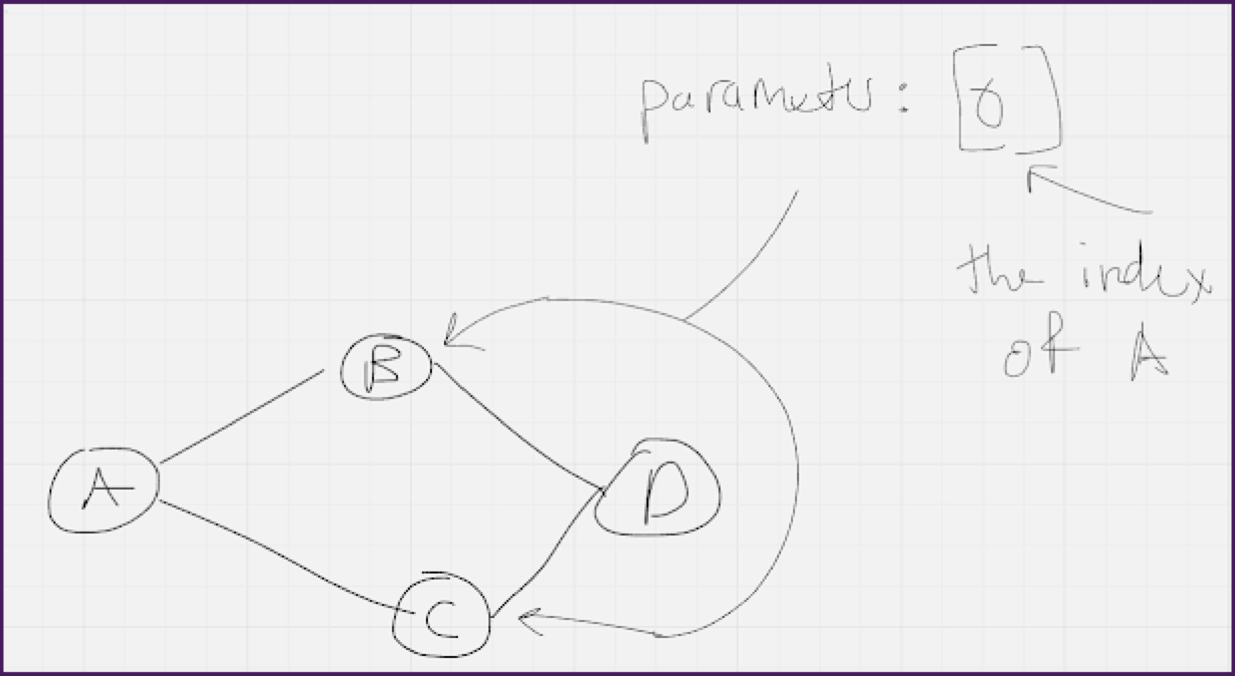
This is where you write your pseudocode:

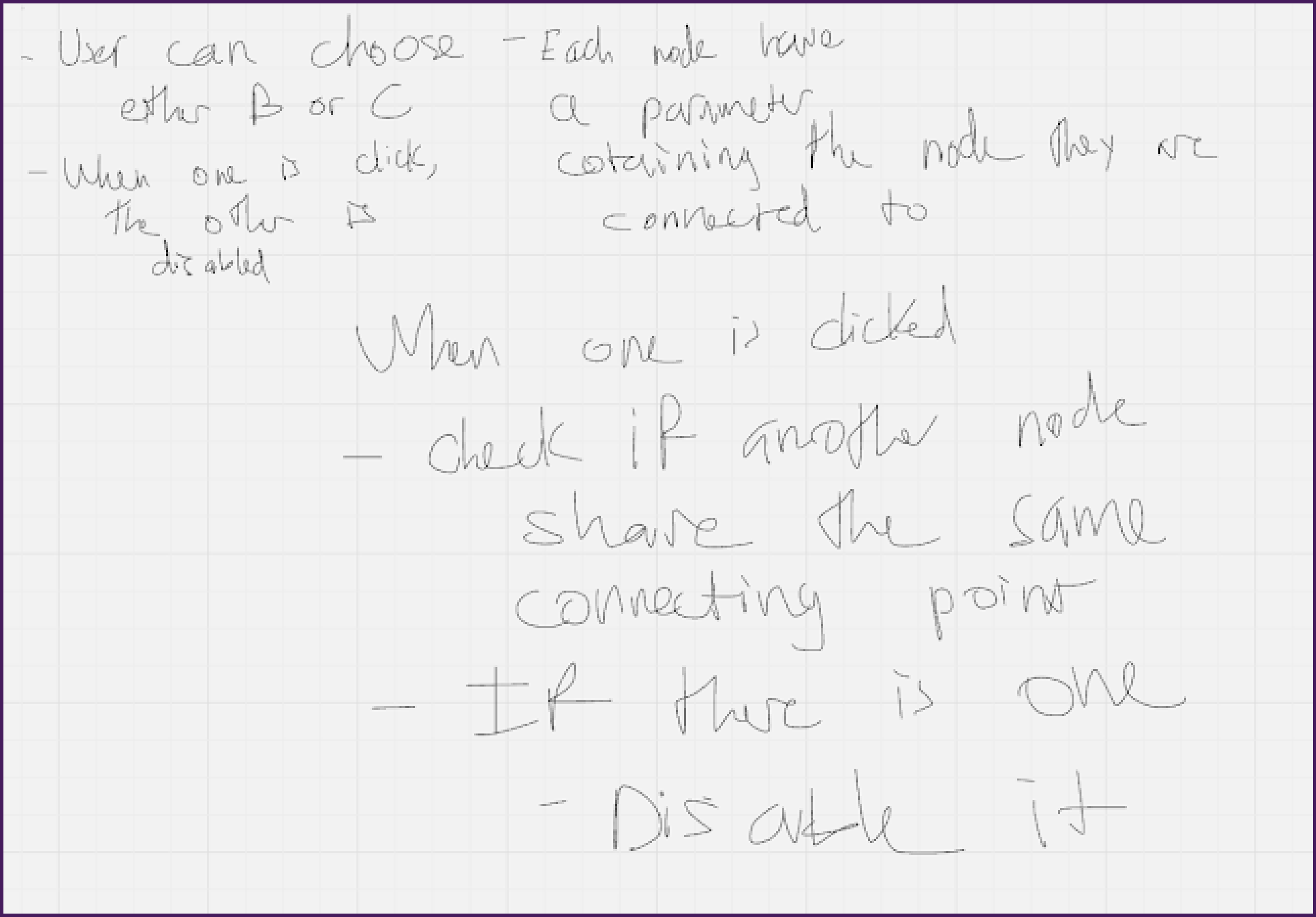



selected
card
card
card
card
card
card
card

```
Discard card
    -append card selected
        to erased list
    -remove card based
        on the card in

Adding new card
    -get amount need
        by length of erased
                            list
    - get used-card
        card in hand
    - get available card
        -prevent duplication
    - choose random card
    - append to cards list
```

AP CSP Project: Create and Test




- User can choose either B or C
- When one is click, the other is disabled

- Each node have a parameter cotaining the node they are connected to

When one is clicked
- check if another node share the same connecting point
    - If there is one
        - Disable it

AP CSP Project: Evaluate the Solution

Were there any issues or challenges that you faced on this project? How did you overcome them?

- The game must know when the mouse cursor hovers over the card. However, the first iteration of the collision system between the mouse cursor and the card did not work as the card have extra space that was not visible. When the mouse goes over this space, the card is hovered, but the mouse cursor was not on the card. This space was large enough that it interfered with the other card
    - I consulted Coding With Russ tutorial on Pygame Collisions
    - Instead of having the plain mouse cursor, I opted for a sprite one
    - With the sprite one, I can use the *overlap()* method of pygame to check if two sprite is overlapping, ie. the card sprite and mouse sprite
    - With this, I am able to calculate the when they overlap based on the mouse and card's x and y coordinates

- When users click on the card, the card is appended to a *selected_card* list, where it will be use to calculate the damage and performs combos. When the user click the card again, it is deselected and removed from the *selected_card* list. After shortening the conditional algorithm that adds and remove the cards from the list, the card would not longer be removed.
    - I looked back into my method that checks whether a card is clicked and either selects or deselects it
    - The program was a logic error
    - I realized that when a card is selected, the method returns True, meaning that the second half of the method, which deals with a card being remove, is ignored
    - When I shorten the conditional statement, my first conditional was to check if the *clicked()* method is True, which it always is, so a card would never get removed because the second half of the method would never get executed

AP CSP Project: Evaluate the Solution

Were there any issues or challenges that you faced on this project? How did you overcome them?

When the user have selected their cards and click the "Attack Button," there should be a short period between in which the text travels to the center and the combo/damage is counted. This did not happen, and those two events occur simultaneously

- Watched a YT and learned about Custom Events: https://youtu.be/eQDi3h61In4?t=253
- Added a Boolean variable that is set to True after the "Attack" button is pressed
    - The boolean is used to call the *total_damage_calculation()* function which contain the user event that has been activated
    - When the function is called, the delay occurs

Because of the different sprite sheet length, the animations between the two character on screen doesn't add up. Also, the enemy are all different sizes, oriented differently, and have different scale value, so a fixed point does not work for all.

- Used a dictionary to stored the enemy's attributes, such as:
    - Resolution
    - Scale
    - Animation Speed Offset
    - Flip
    - Y Offset
    - Strength
    - HP
- Call to that dictionary, getting the enemy's name and attribute to create the enemy