Immune Escape

Emma, Walter, Anastasia and Colin

Background

Motivation/vision for the game:

- Create a two-player, turn-based battle game featuring the immune system
- Customize the game such that different “characters” have different attacks, defenses, and other features that are true to their biological role
- Create a game that has an educational component for learning about immunology and microbiology concepts

Pygame - www.pygame.org

What is Pygame:

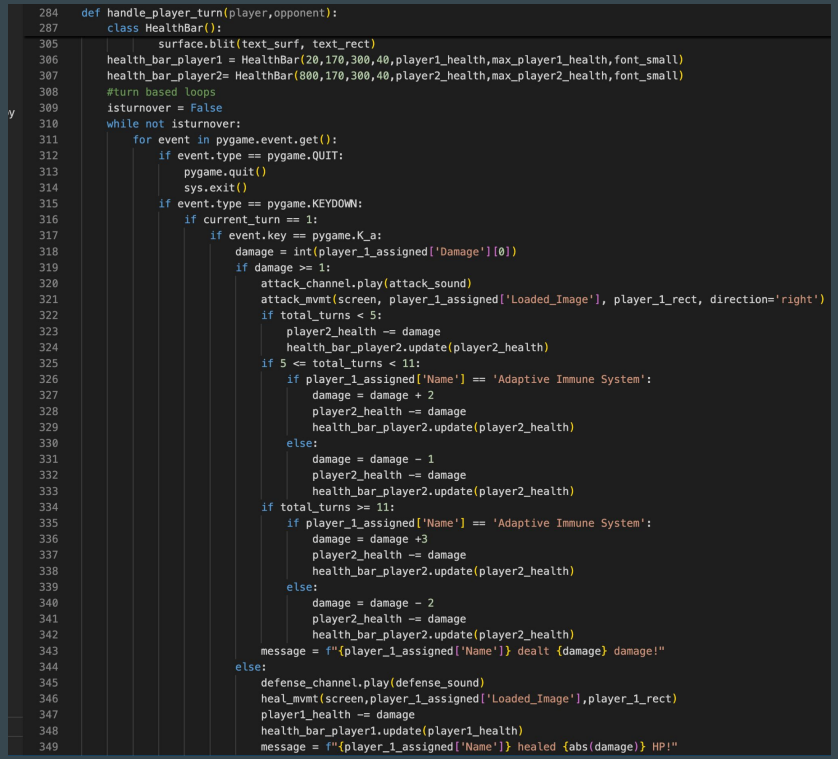
- **Pygame** is a Python library designed for making 2D games and interactive applications.
  a. It provides simple tools for handling:
     i. Graphics – drawing images, shapes, and text on the screen
     ii. Sound – playing background music and sound effects
     iii. Input – responding to keyboard, mouse, or controller actions
     iv. Timing & Events – controlling game speed and player interactions

Why It's Useful for Our Game:

- Beginner-friendly — reinforces Python fundamentals (loops, functions, objects, logic)
- Immediate visual feedback — helps connect code to on-screen results
- Allows for an interactive "educational" game experience

Python Coding Elements Used

- File I/O
- Data structures
- For and while loops
- Nested logic
- Class creations
- Functions
- Global variables
- Event-driven programming
- State management and game logic
- Graphing and animation
- Sound integration
- Timing and frame control



```
284  def handle_player_turn(player,opponent):
287      class HealthBar():
305              surface.blit(text_surf, text_rect)
306      health_bar_player1 = HealthBar(20,170,300,40,player1_health,max_player1_health,font_small)
307      health_bar_player2= HealthBar(800,170,300,40,player2_health,max_player2_health,font_small)
308      #turn based loops
309      isturnover = False
310      while not isturnover:
311          for event in pygame.event.get():
312              if event.type == pygame.QUIT:
313                  pygame.quit()
314                  sys.exit()
315              if event.type == pygame.KEYDOWN:
316                  if current_turn == 1:
317                      if event.key == pygame.K_a:
318                          damage = int(player_1_assigned['Damage'][0])
319                          if damage >= 1:
320                              attack_channel.play(attack_sound)
321                              attack_mvmt(screen, player_1_assigned['Loaded_Image'], player_1_rect, direction='right')
322                              if total_turns < 5:
323                                  player2_health -= damage
324                                  health_bar_player2.update(player2_health)
325                              if 5 <= total_turns < 11:
326                                  if player_1_assigned['Name'] == 'Adaptive Immune System':
327                                      damage = damage + 2
328                                      player2_health -= damage
329                                      health_bar_player2.update(player2_health)
330                                  else:
331                                      damage = damage - 1
332                                      player2_health -= damage
333                                      health_bar_player2.update(player2_health)
334                              if total_turns >= 11:
335                                  if player_1_assigned['Name'] == 'Adaptive Immune System':
336                                      damage = damage +3
337                                      player2_health -= damage
338                                      health_bar_player2.update(player2_health)
339                                  else:
340                                      damage = damage - 2
341                                      player2_health -= damage
342                                      health_bar_player2.update(player2_health)
343                              message = f"{player_1_assigned['Name']} dealt {damage} damage!"
344                          else:
345                              defense_channel.play(defense_sound)
346                              heal_mvmt(screen,player_1_assigned['Loaded_Image'],player_1_rect)
347                              player1_health -= damage
348                              health_bar_player1.update(player1_health)
349                              message = f"{player_1_assigned['Name']} healed {abs(damage)} HP!"
```

Code Structure

Loading Player Type Dictionaries

Each player category (immune system or pathogen) is encoded in a dictionary of dictionaries that holds character specific attributes. The structure is as follows: Dictionary: Immune System or Pathogen Keys: 'Character' Dictionaries (i.e. virus, bacteria, parasite for the pathogen dictionary) Keys: Attributes of the Character (Actions, Damage, Health, Image) Lists: Of possible actions, damage amounts, starting health, and image paths.

Text and Image Import

In this section, we specify all the text used and render the text objects for display on the screen. We do the same for all images, specifying the image path and assigning them to variables that can be rendered on the appropriate screen.

Creating Clickable Icons

We created a class of clickable icons that are used during the character selection process.

Drawing Screens

We created functions that will 'draw' the desired screens that we use at different points in the game, and then set those states as arbitrary values so that they can be defined and called upon during the game loop.

Game Loop

Game Loop

- We created a game loop which, depending on which characters player 1 and player 2 chose, inserts them into a battle.
- Each character has their own actions that are used to deal different levels of damage.
- The actions are chosen by specifying key choices on the keyboard. a, s, and d are used for player 1 and up, down, and left for player 2.
- Depending on which action is chosen, the associated damage will be taken from the opponent's health bar or if the action is healing, the associated heal will be added to the own players health.
- This loop also contains instructions to display text depending on which action was chosen and how much damage it dealt. Movements for attacks is back and forward again and for a heal it's up and down.
- Once in the loop which specified what happens during the battle the turn based loop allows the players to go back and forth, one player at a time, until one players health falls below 0 points.
- In this event, the screen changes to the associated winner of the game.

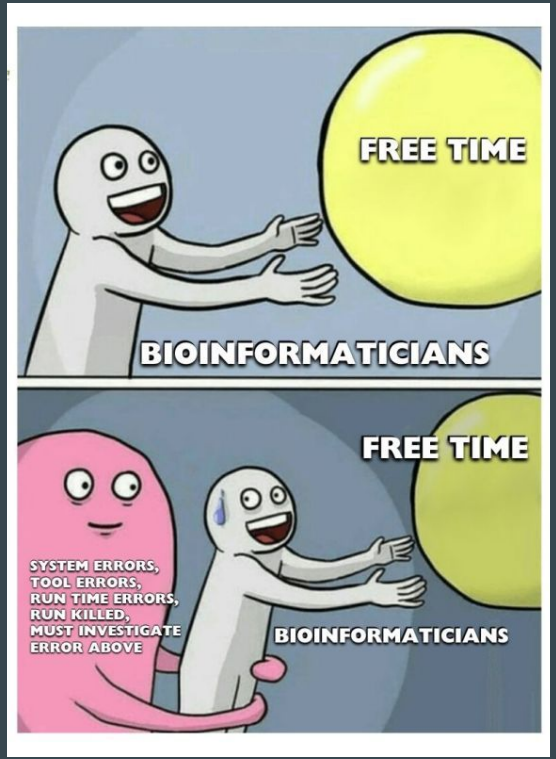
Challenges and future opportunities

Challenges:

- Steep learning curve for pygame language, functions, and structure.
- Creating the correct format for loops and when to end the loops.
- Keeping music running after restart screen.
- Learning how to work with a group github repo.

Future opportunities:

- Creating more advanced gameplay
- Learning from live demo to evaluate if any players have unfair advantages.



Live Demo


Welcome to ImmuneEscape!
Enter
Where the outcomes is literally life or death
PLAY AT YOUR OWN RISK