



Dungeon Destroyer

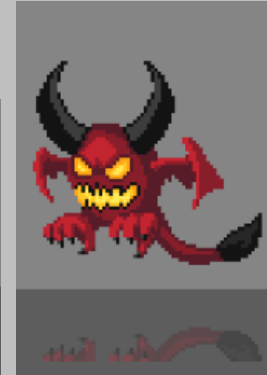
Husain Patanwala (CS210)

General Overview

I am considering making a turn-based RPG style game called “Dungeon Destroyer”. The main premise of the game is to traverse through the map which will be implemented using a graph, and the player has to battle monsters in turn-base combat. To learn new skills, players must gain experience and work their way through the map. Certain skills will be more effective against certain monsters, and thus, players will have to focus on unlocking different parts of the skill tree to ensure they can pass levels. Also, user related info will be stored using hashmaps and lists, and the turn based system between player and ai enemies will be implemented using a queue. The end goal of the game is to reach the end of the map and defeat the boss. The story is still somewhat incomplete, but these are the general mechanics I would like to implement.

Data Structures:

- Hashmap
 - Entity animations
 - Game colors
- Queue
 - Each level has a queue to simulate turn based combat
- List
 - Store frames for each animation sequence
- Tree
 - A skill tree used to show player progression; new skill will unlock based on level completion
- Graph
 - Used to represent the map structure



```

1  def main_menu():
2      start_button = Button('Assets/UI/border 2.png', SCREEN_WIDTH//2, 450, 117, 59, 2.5, 'Start')
3      skills = Button('Assets/UI/border 2.png', SCREEN_WIDTH//2, 600, 117, 59, 2.5, 'Skills')
4
5      while True:
6          CLOCK.tick(FPS)
7
8          screen.fill(COLORS['GREY'])
9          draw_title(screen, 'Dungeon Destroyer', 'white', SCREEN_WIDTH//2, 275)
10
11         if start_button.draw(screen):
12             map()
13         if skills.draw(screen):
14             print('skill tree')
15
16         for event in pygame.event.get():
17             if event.type == pygame.QUIT:
18                 pygame.quit()
19             elif event.type == pygame.KEYDOWN and pygame.key.get_pressed()[pygame.K_ESCAPE]:
20                 pygame.quit()
21
22         pygame.display.update()

```

Knight constructor →

← Main menu function

```

1  class Knight(pygame.sprite.Sprite):
2      def __init__(self, x, y, max_hp):
3          pygame.sprite.Sprite.__init__(self)
4
5          self.animations = {
6              'IDLE' : [animation_parser('Assets/Player/IDLE.png', 7, 96, 84, 2.5), 0],
7              'HURT' : [animation_parser('Assets/Player/HURT.png', 4, 96, 84, 2.5), 0],
8              'DEATH' : [animation_parser('Assets/Player/DEATH.png', 12, 96, 84, 2.5), 0],
9              'ATTACK 1' : [animation_parser('Assets/Player/ATTACK 1.png', 6, 96, 84, 2.5), 30],
10             'ATTACK 2' : [animation_parser('Assets/Player/ATTACK 2.png', 5, 96, 84, 2.5), 40],
11             'ATTACK 3' : [animation_parser('Assets/Player/ATTACK 3.png', 6, 96, 84, 2.5), 50],
12             'DEFEND' : [animation_parser('Assets/Player/DEFEND.png', 6, 96, 84, 2.5), 0]
13         }
14
15         self.action = 'IDLE'
16
17         self.max_hp = max_hp
18         self.curr_hp = max_hp
19         self.alive = True
20
21         self.image = self.animations[self.action][0][0]
22         self.rect = self.image.get_rect()
23         self.rect.center = (x, y)
24
25         self.update_time = pygame.time.get_ticks()
26         self.curr_frame = 0
27

```

GUI Mockups + Progress so Far

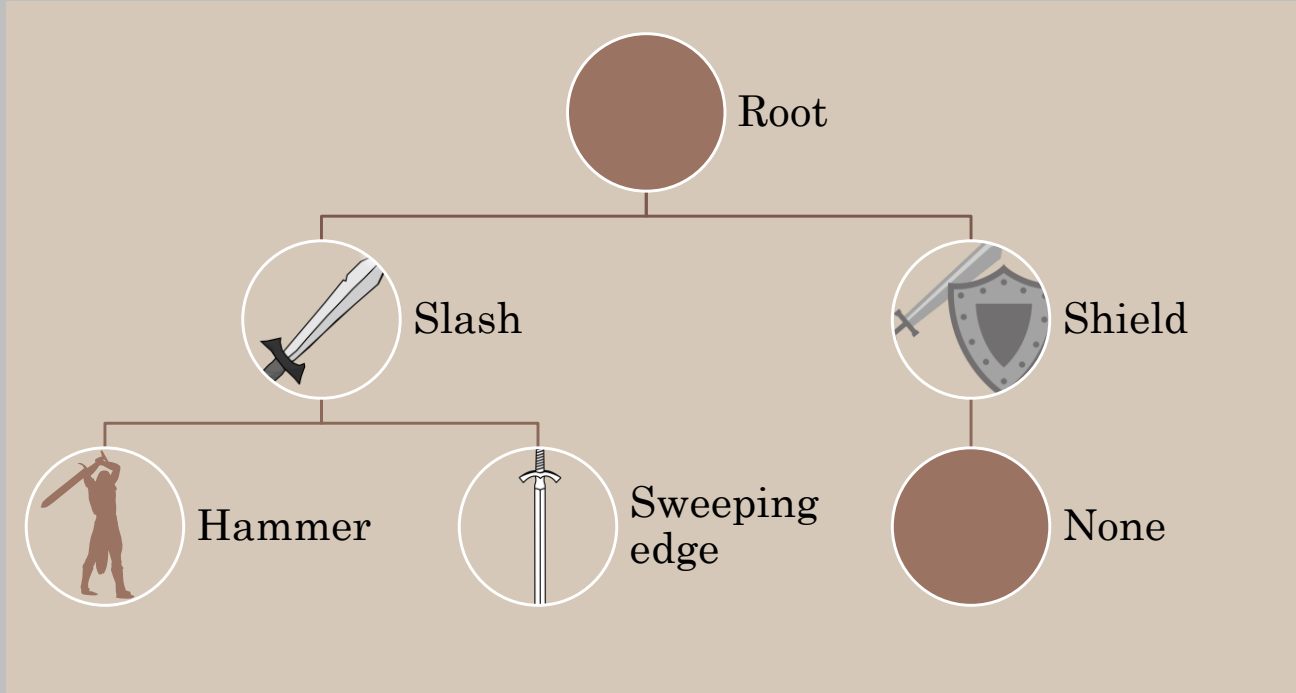


This is the main menu/start screen of the game. From here you can either view the game map (in progress) or the skill tree (tbd).

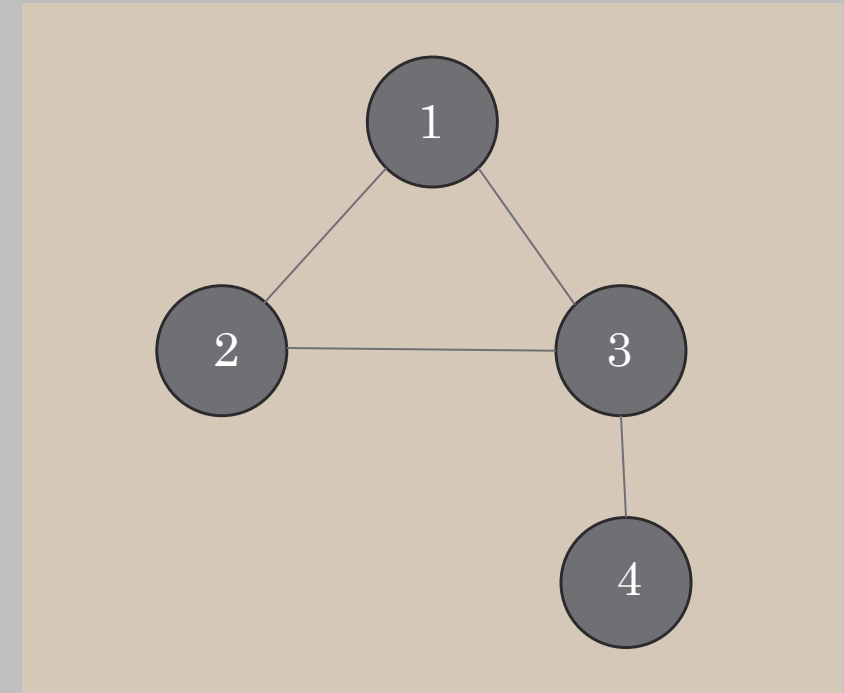


This is an example of a possible level. The health bars are shown in green, and adapt to damage changes. The red buttons are placeholders for attack input (they work btw).

GUI Mockups + Progress so Far cont.



This is a mockup of the skill tree. What I'm thinking is that for each level you can possibly unlock a new level on the skill tree...



This is a simple graph, and this is how I want to lay it out. Although, I have to figure out how I'm going to implement it.

