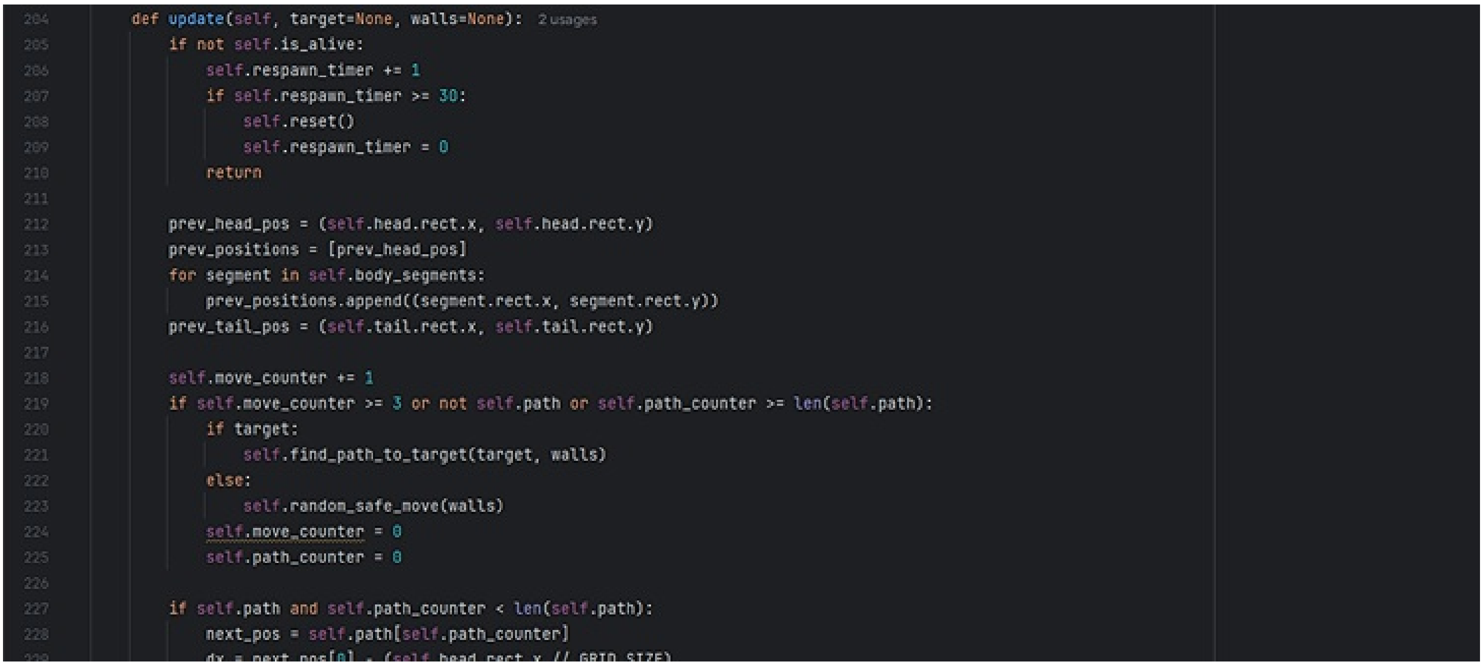
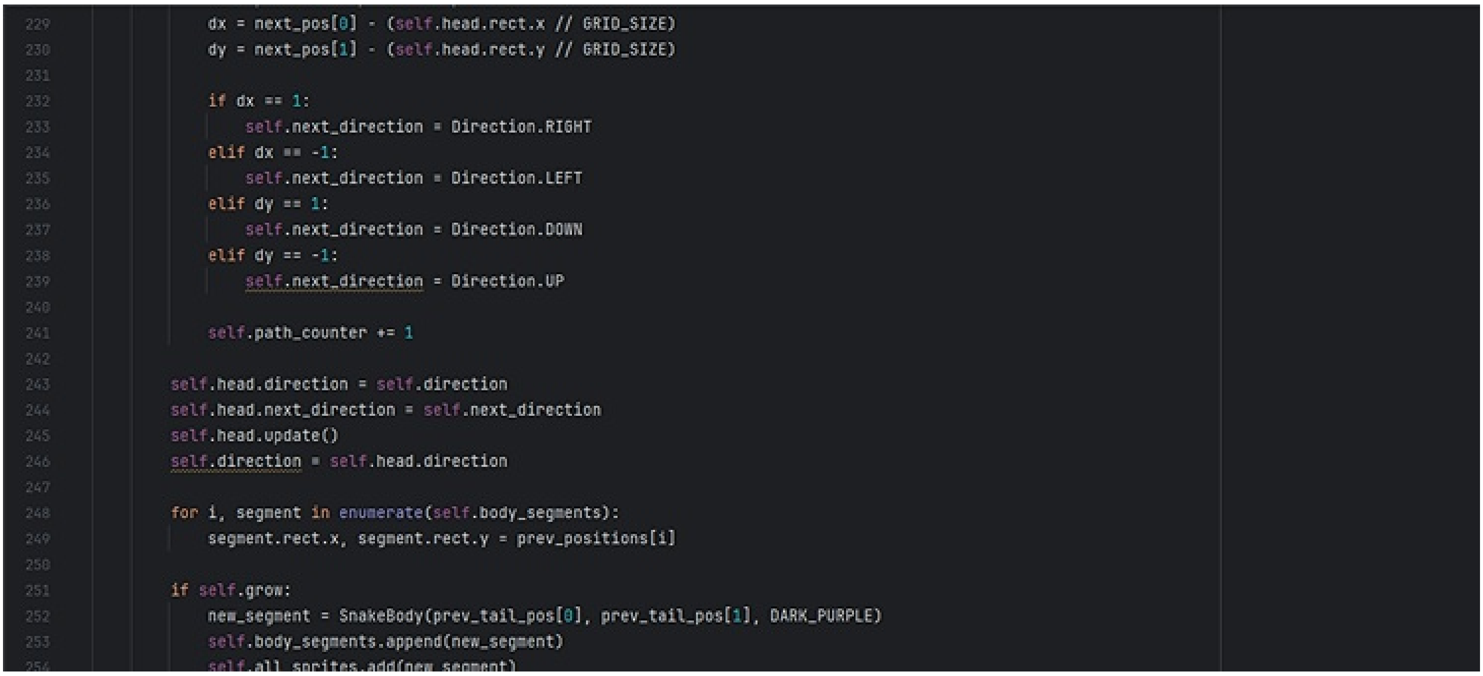
## Component C. Personalized Project Reference.

## Procedure:

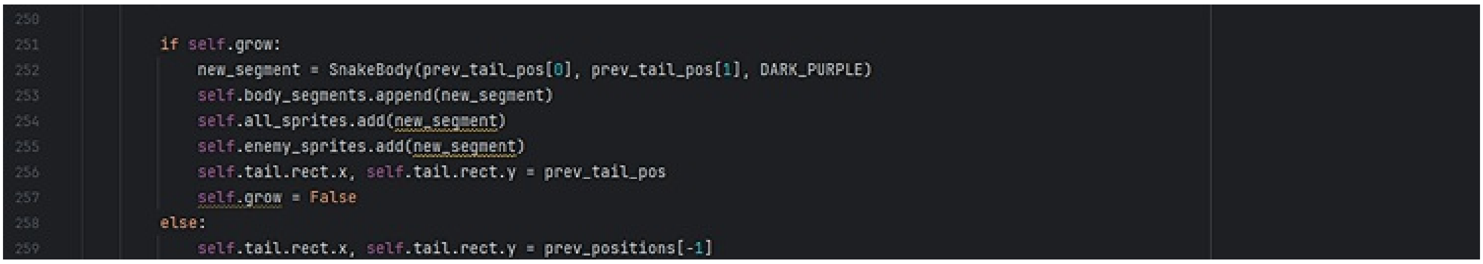
### i.



```
    def update(self, target=None, walls=None):  2 usages
        if not self.is_alive:
            self.respawn_timer += 1
            if self.respawn_timer >= 30:
                self.reset()
                self.respawn_timer = 0
            return

        prev_head_pos = (self.head.rect.x, self.head.rect.y)
        prev_positions = [prev_head_pos]
        for segment in self.body_segments:
            prev_positions.append((segment.rect.x, segment.rect.y))
        prev_tail_pos = (self.tail.rect.x, self.tail.rect.y)

        self.move_counter += 1
        if self.move_counter >= 3 or not self.path or self.path_counter >= len(self.path):
            if target:
                self.find_path_to_target(target, walls)
            else:
                self.random_safe_move(walls)
            self.move_counter = 0
            self.path_counter = 0

        if self.path and self.path_counter < len(self.path):
            next_pos = self.path[self.path_counter]
```

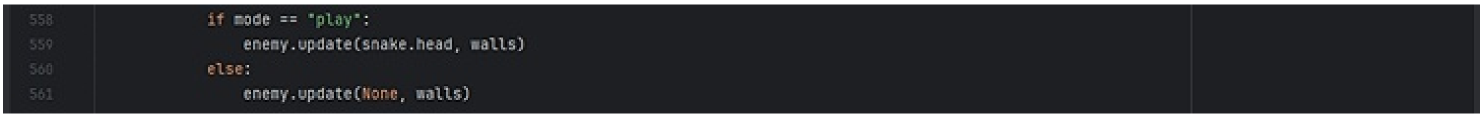


```
            dx = next_pos[0] - (self.head.rect.x // GRID_SIZE)
            dy = next_pos[1] - (self.head.rect.y // GRID_SIZE)

            if dx == 1:
                self.next_direction = Direction.RIGHT
            elif dx == -1:
                self.next_direction = Direction.LEFT
            elif dy == 1:
                self.next_direction = Direction.DOWN
            elif dy == -1:
                self.next_direction = Direction.UP

            self.path_counter += 1

        self.head.direction = self.direction
        self.head.next_direction = self.next_direction
        self.head.update()
        self.direction = self.head.direction

        for i, segment in enumerate(self.body_segments):
            segment.rect.x, segment.rect.y = prev_positions[i]

        if self.grow:
            new_segment = SnakeBody(prev_tail_pos[0], prev_tail_pos[1], DARK_PURPLE)
            self.body_segments.append(new_segment)
            self.all_sprites.add(new_segment)
```



```
if self.grow:
    new_segment = SnakeBody(prev_tail_pos[0], prev_tail_pos[1], DARK_PURPLE)
    self.body_segments.append(new_segment)
    self.all_sprites.add(new_segment)
    self.enemy_sprites.add(new_segment)
    self.tail.rect.x, self.tail.rect.y = prev_tail_pos
    self.grow = False
else:
    self.tail.rect.x, self.tail.rect.y = prev_positions[-1]
```

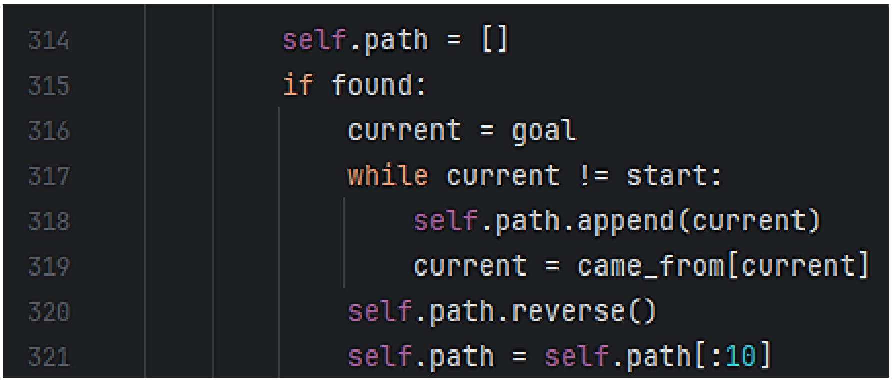
### ii.



```
if mode == "play":
    enemy.update(snake.head, walls)
else:
    enemy.update(None, walls)
```

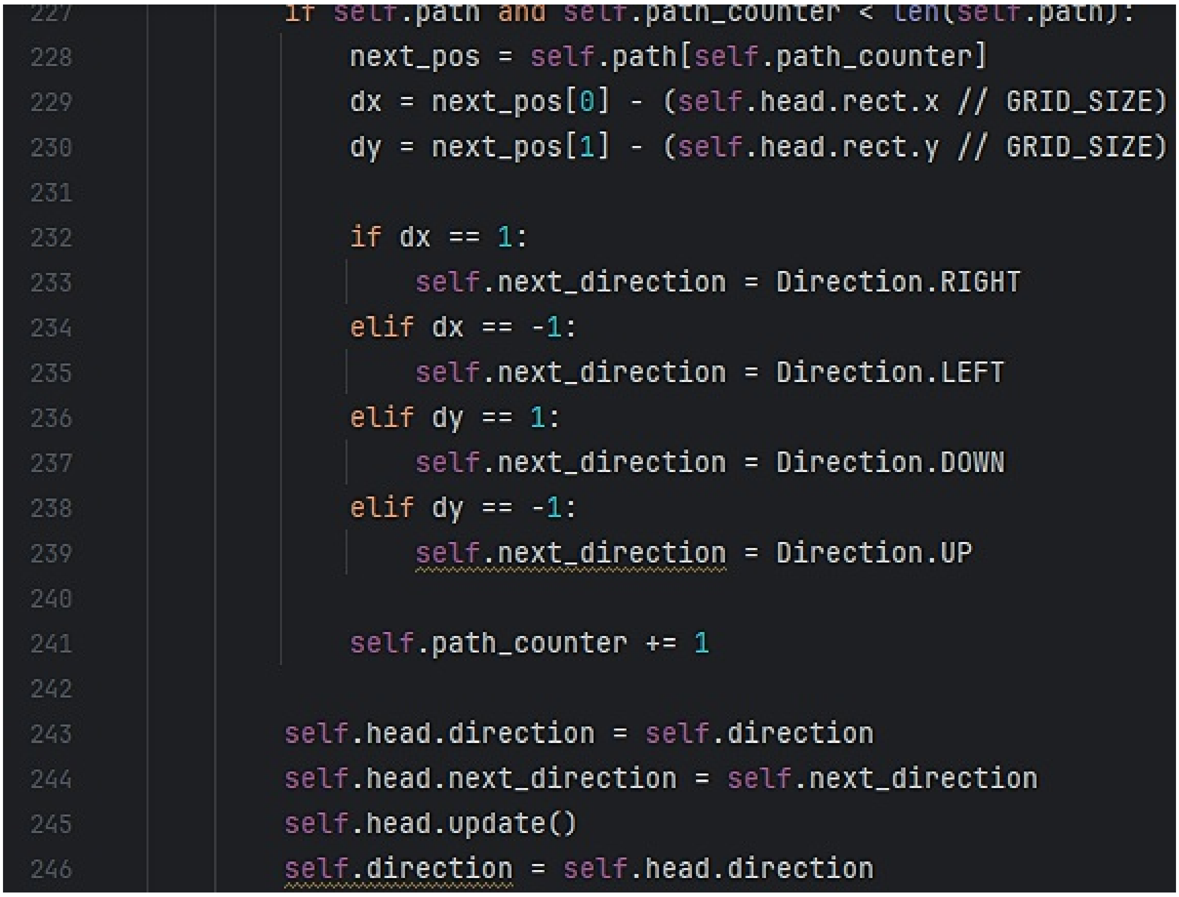
## List:

### i.



```
self.path = []
if found:
    current = goal
    while current != start:
        self.path.append(current)
        current = came_from[current]
    self.path.reverse()
    self.path = self.path[:10]
```

ii.



```
        if self.path and self.path_counter < len(self.path):
            next_pos = self.path[self.path_counter]
            dx = next_pos[0] - (self.head.rect.x // GRID_SIZE)
            dy = next_pos[1] - (self.head.rect.y // GRID_SIZE)

            if dx == 1:
                self.next_direction = Direction.RIGHT
            elif dx == -1:
                self.next_direction = Direction.LEFT
            elif dy == 1:
                self.next_direction = Direction.DOWN
            elif dy == -1:
                self.next_direction = Direction.UP

            self.path_counter += 1

        self.head.direction = self.direction
        self.head.next_direction = self.next_direction
        self.head.update()
        self.direction = self.head.direction
```