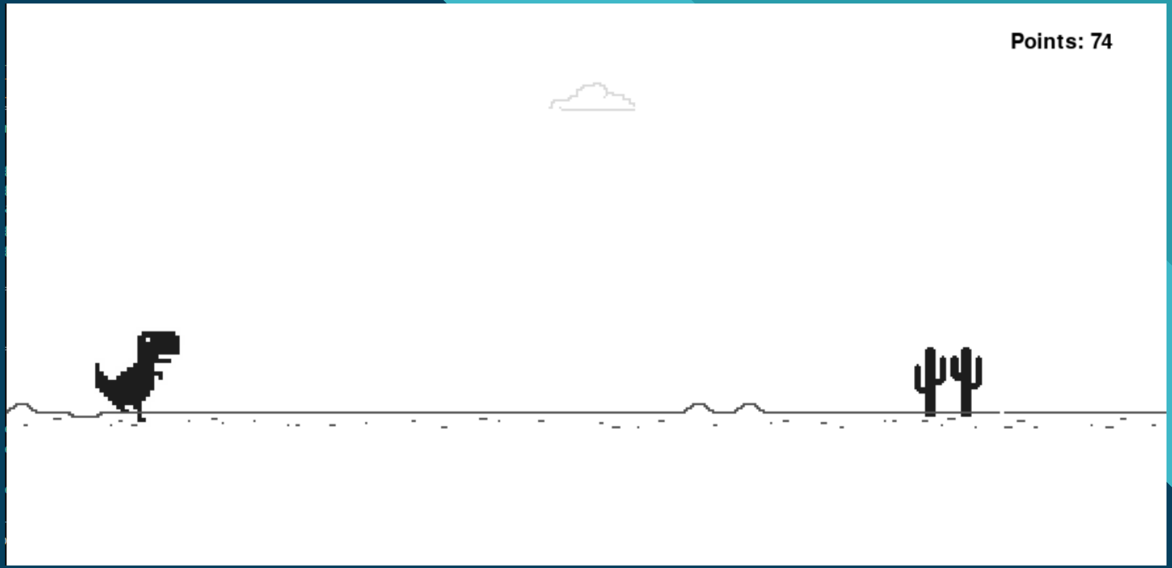
Dino T-Rex

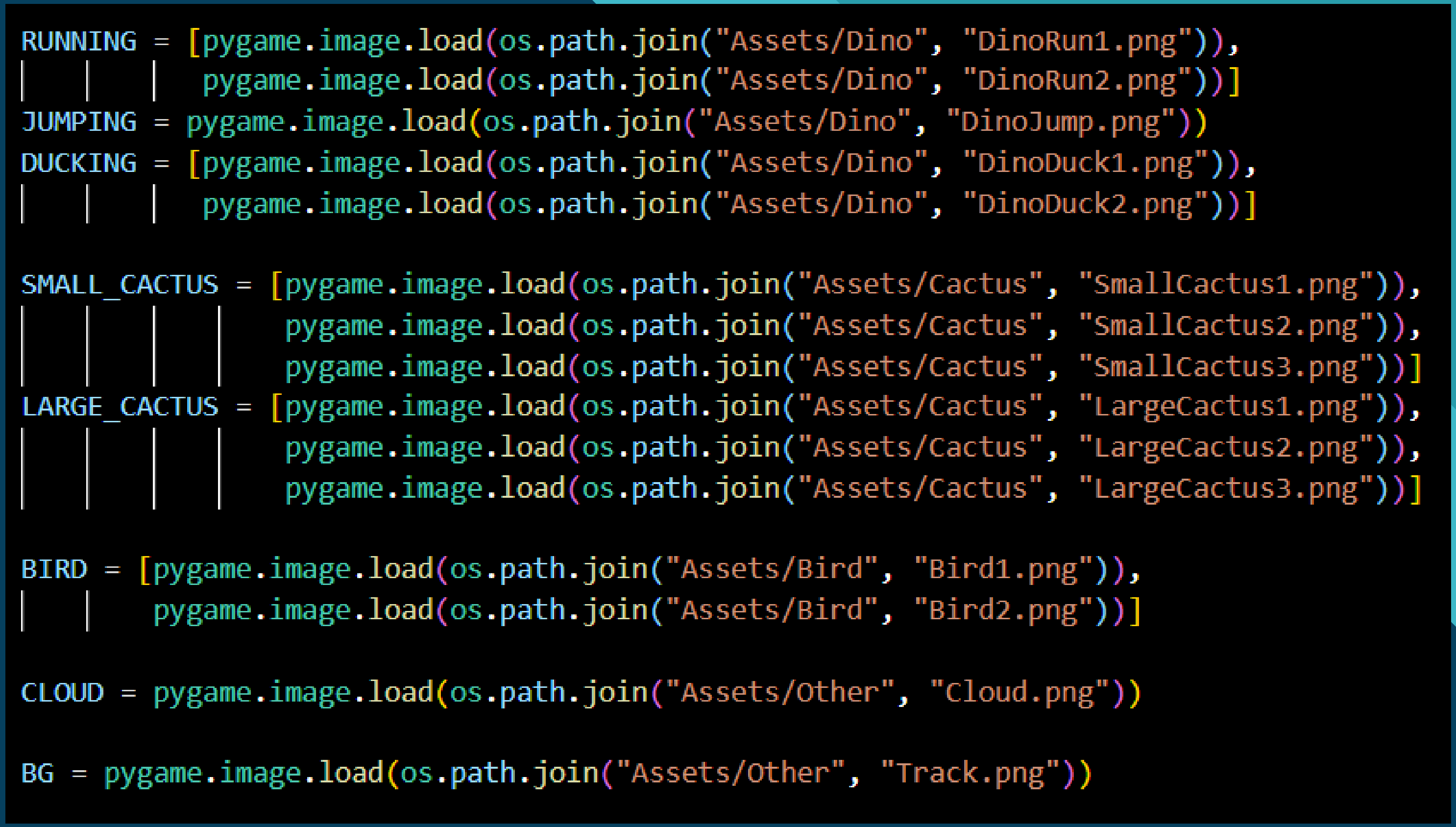
Game Overview

The Dinosaur Game is a simple side-scrolling platformer where player controls a T-Rex. The objective is to navigate the dinosaur through obstacles by jumping and ducking while collecting points.

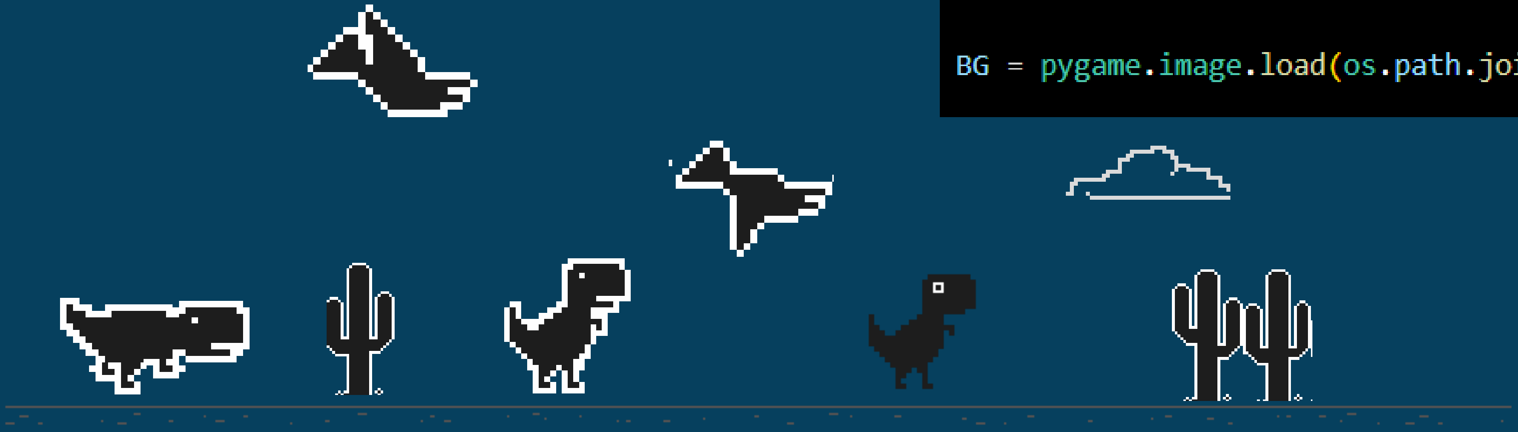


Game Assets

Let's take a look at the assets used in our game, including images for the dinosaur character, obstacles, and background elements.



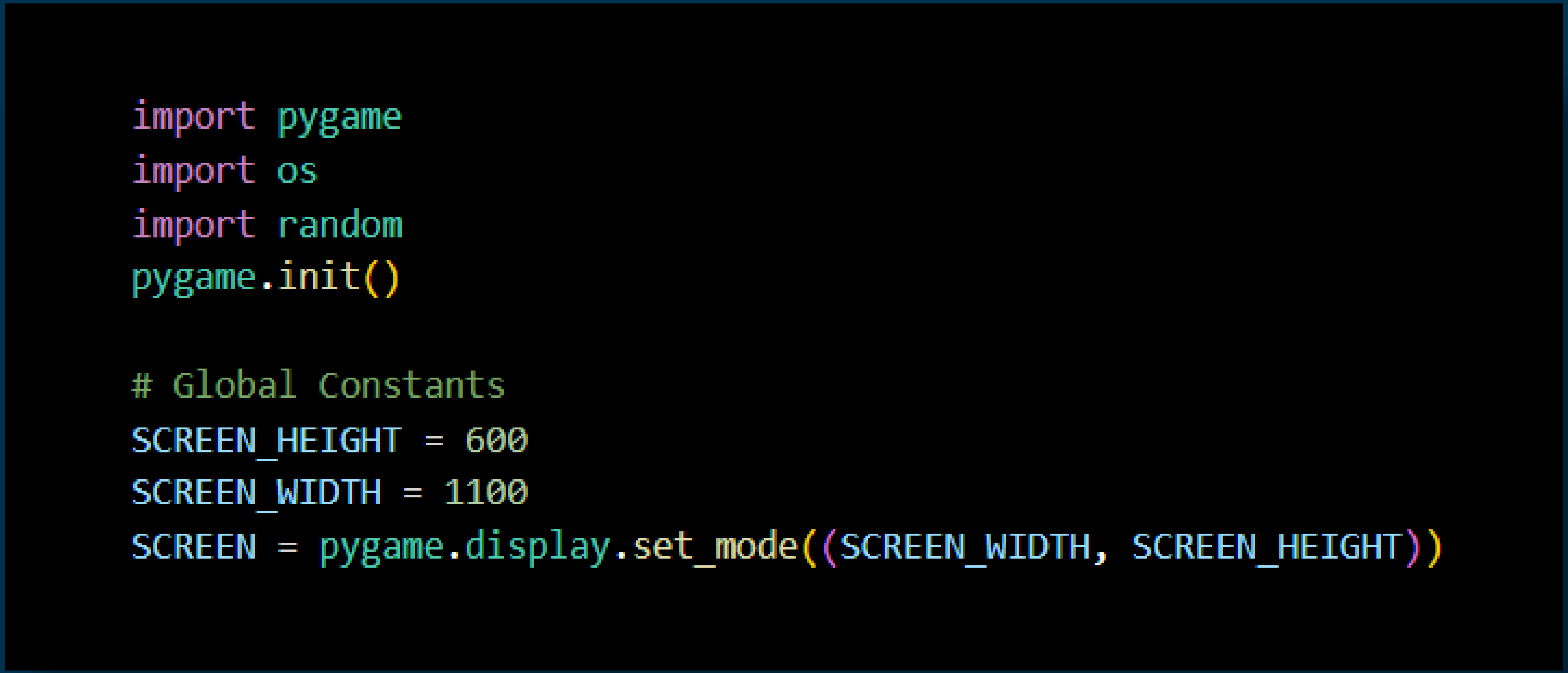
```
RUNNING = [pygame.image.load(os.path.join("Assets/Dino", "DinoRun1.png")),
           pygame.image.load(os.path.join("Assets/Dino", "DinoRun2.png"))]
JUMPING = pygame.image.load(os.path.join("Assets/Dino", "DinoJump.png"))
DUCKING = [pygame.image.load(os.path.join("Assets/Dino", "DinoDuck1.png")),
           pygame.image.load(os.path.join("Assets/Dino", "DinoDuck2.png"))]

SMALL_CACTUS = [pygame.image.load(os.path.join("Assets/Cactus", "SmallCactus1.png")),
                pygame.image.load(os.path.join("Assets/Cactus", "SmallCactus2.png")),
                pygame.image.load(os.path.join("Assets/Cactus", "SmallCactus3.png"))]
LARGE_CACTUS = [pygame.image.load(os.path.join("Assets/Cactus", "LargeCactus1.png")),
                pygame.image.load(os.path.join("Assets/Cactus", "LargeCactus2.png")),
                pygame.image.load(os.path.join("Assets/Cactus", "LargeCactus3.png"))]

BIRD = [pygame.image.load(os.path.join("Assets/Bird", "Bird1.png")),
        pygame.image.load(os.path.join("Assets/Bird", "Bird2.png"))]

CLOUD = pygame.image.load(os.path.join("Assets/Other", "Cloud.png"))

BG = pygame.image.load(os.path.join("Assets/Other", "Track.png"))
```



Setup process

Our game is built using the Pygame library, a tool for game development in Python.

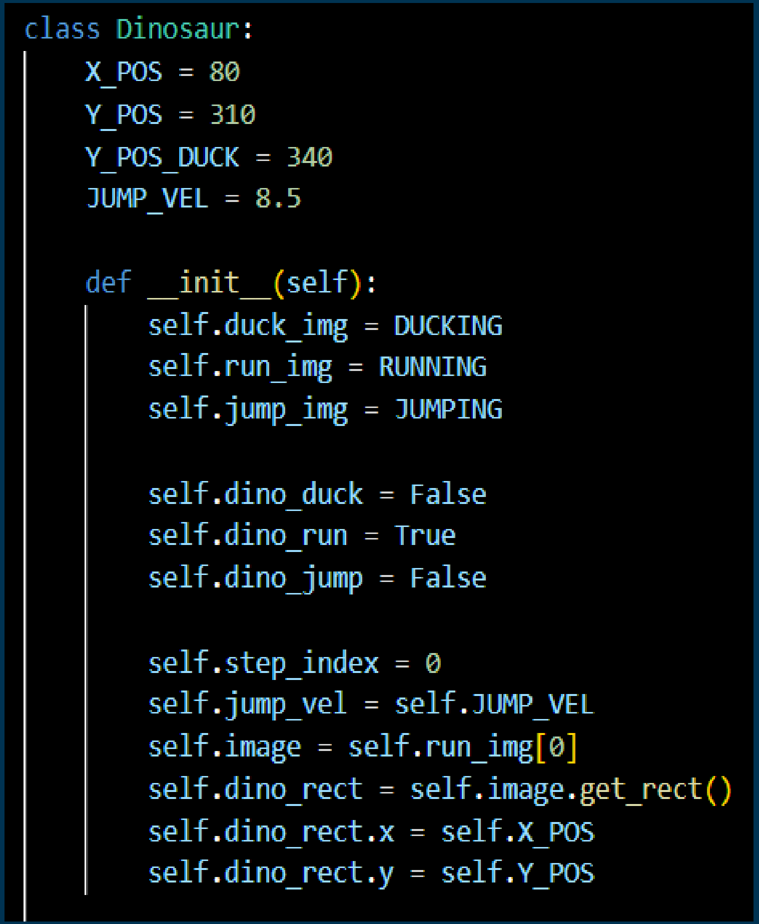
We define several global constants to establish the basic parameters of our game environment. These constants include the screen dimensions and the paths to our game assets.



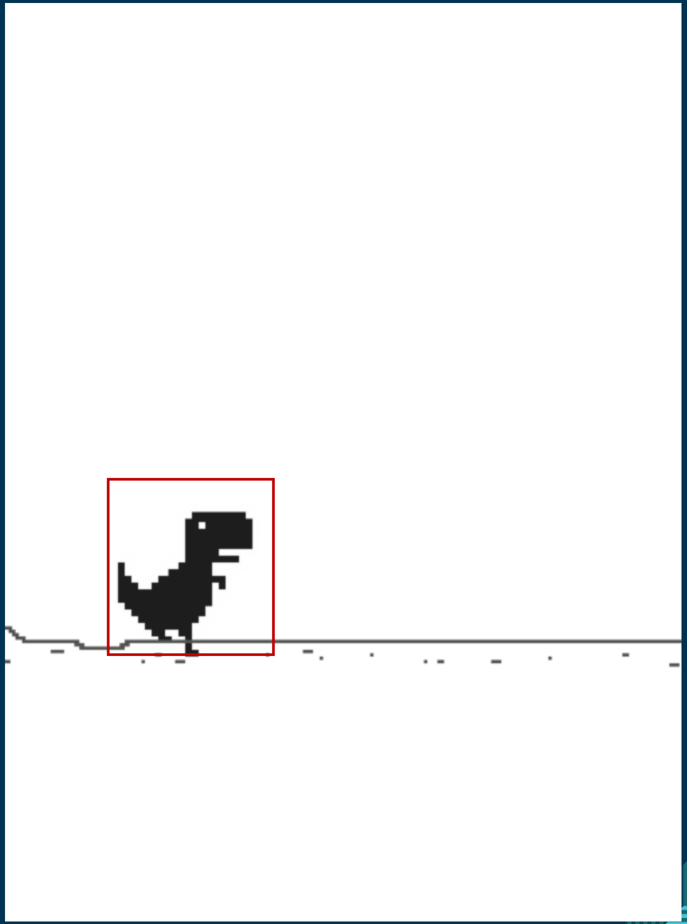
```
import pygame
import os
import random
pygame.init()

# Global Constants
SCREEN_HEIGHT = 600
SCREEN_WIDTH = 1100
SCREEN = pygame.display.set_mode((SCREEN_WIDTH, SCREEN_HEIGHT))
```

Setting up the T-Rex

The Dinosaur class represents the main character of the game. Its attributes define various parameters related to the dinosaur's position, animation, and behavior. By default, the dinosaur is set to be running and not ducking or jumping.



```
class Dinosaur:
    X_POS = 80
    Y_POS = 310
    Y_POS_DUCK = 340
    JUMP_VEL = 8.5

    def __init__(self):
        self.duck_img = DUCKING
        self.run_img = RUNNING
        self.jump_img = JUMPING

        self.dino_duck = False
        self.dino_run = True
        self.dino_jump = False

        self.step_index = 0
        self.jump_vel = self.JUMP_VEL
        self.image = self.run_img[0]
        self.dino_rect = self.image.get_rect()
        self.dino_rect.x = self.X_POS
        self.dino_rect.y = self.Y_POS
```



Behaviour of T-Rex

❑ Update Method:

-Updates dinosaur's behavior based on user input.

-Checks if it's ducking, running, or jumping.

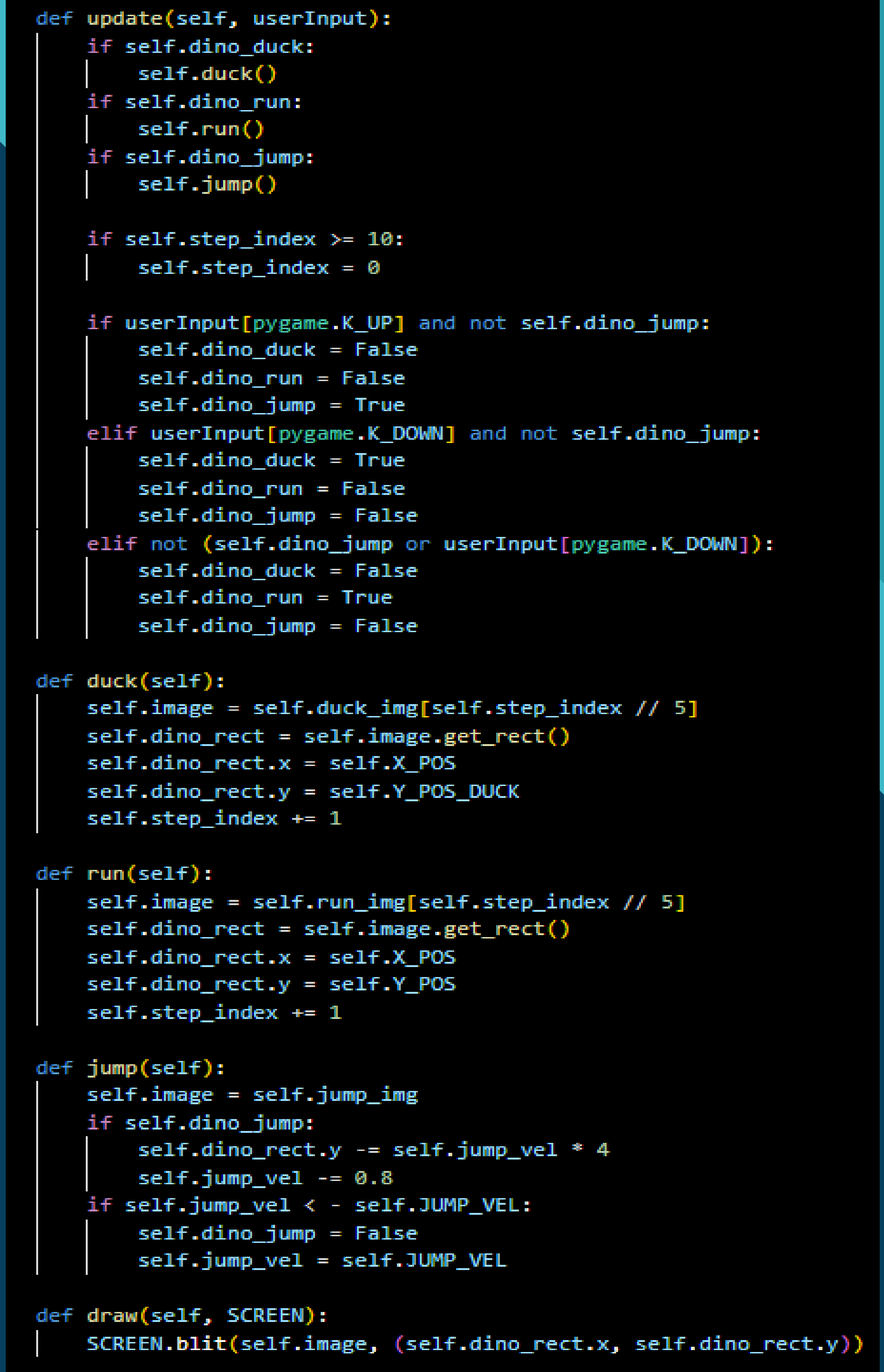
-Responds to jump (pygame.K_UP) or duck (pygame.K_DOWN).

❑ Duck, Run and Jump Method:

-Updates image and position when ducking, running and jumping.

-Adjusts bounding box for positions.

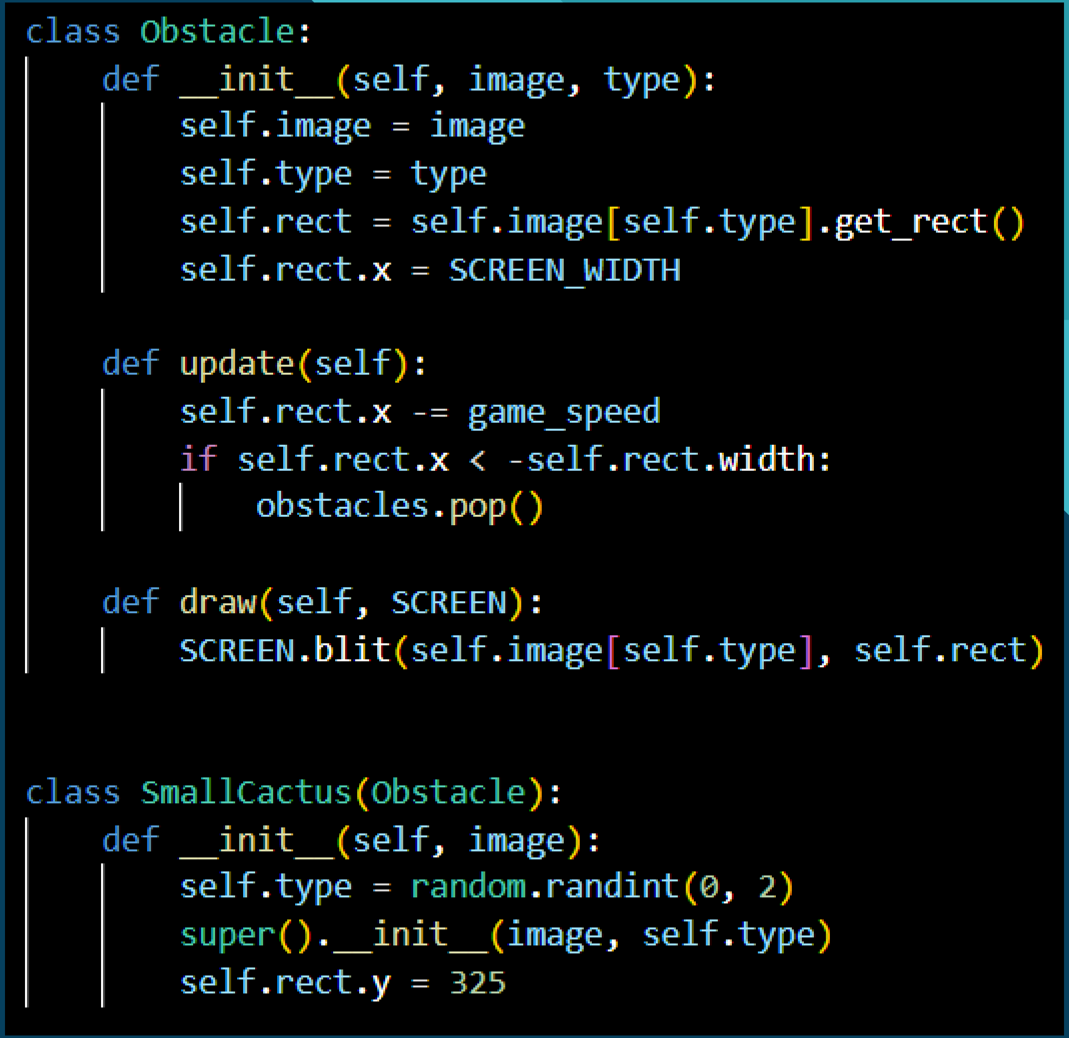
-Decreases jump velocity until minimum.



```python
def update(self, userInput):
    if self.dino_duck:
        self.duck()
    if self.dino_run:
        self.run()
    if self.dino_jump:
        self.jump()

    if self.step_index >= 10:
        self.step_index = 0

    if userInput[pygame.K_UP] and not self.dino_jump:
        self.dino_duck = False
        self.dino_run = False
        self.dino_jump = True
    elif userInput[pygame.K_DOWN] and not self.dino_jump:
        self.dino_duck = True
        self.dino_run = False
        self.dino_jump = False
    elif not (self.dino_jump or userInput[pygame.K_DOWN]):
        self.dino_duck = False
        self.dino_run = True
        self.dino_jump = False

def duck(self):
    self.image = self.duck_img[self.step_index // 5]
    self.dino_rect = self.image.get_rect()
    self.dino_rect.x = self.X_POS
    self.dino_rect.y = self.Y_POS_DUCK
    self.step_index += 1

def run(self):
    self.image = self.run_img[self.step_index // 5]
    self.dino_rect = self.image.get_rect()
    self.dino_rect.x = self.X_POS
    self.dino_rect.y = self.Y_POS
    self.step_index += 1

def jump(self):
    self.image = self.jump_img
    if self.dino_jump:
        self.dino_rect.y -= self.jump_vel * 4
        self.jump_vel -= 0.8
    if self.jump_vel < - self.JUMP_VEL:
        self.dino_jump = False
        self.jump_vel = self.JUMP_VEL

def draw(self, SCREEN):
    SCREEN.blit(self.image, (self.dino_rect.x, self.dino_rect.y))
```

Obstacles

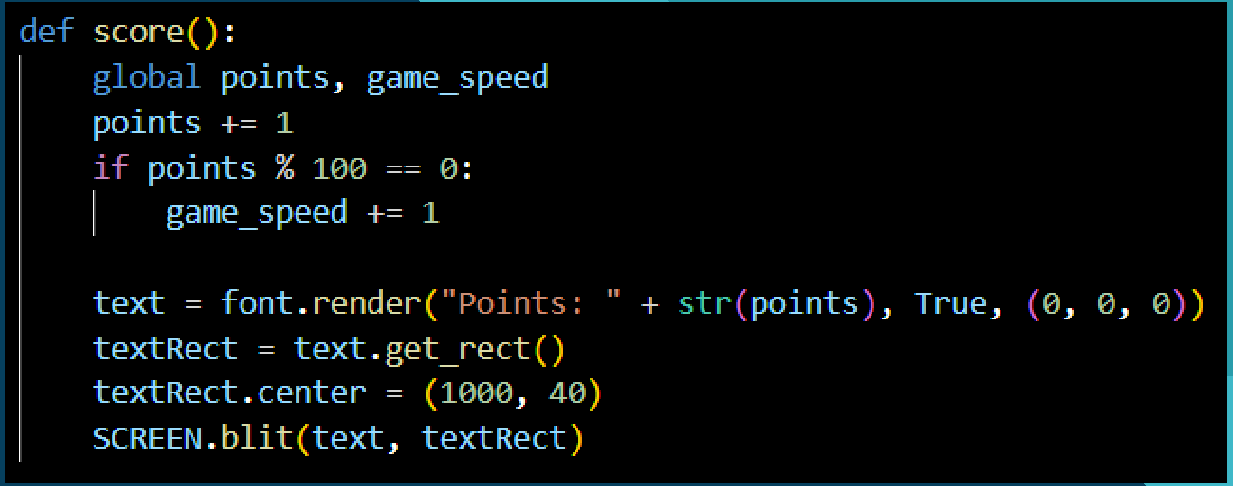
- ❑The Obstacle class manages the behavior of obstacles in the game environment. It takes an image list and a type parameter, representing the various obstacle variations and their specific type.
- ❑The update method moves the obstacle towards the left of the screen based on the game speed.
- ❑The draw method renders the obstacle image onto the screen at its current position.



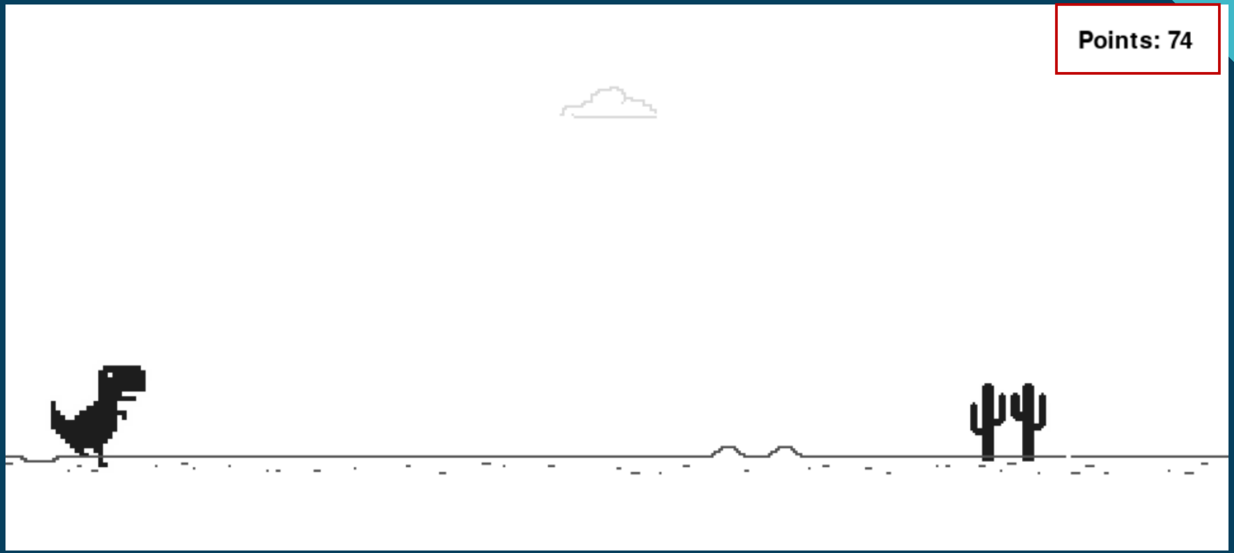
```python
class Obstacle:
    def __init__(self, image, type):
        self.image = image
        self.type = type
        self.rect = self.image[self.type].get_rect()
        self.rect.x = SCREEN_WIDTH

    def update(self):
        self.rect.x -= game_speed
        if self.rect.x < -self.rect.width:
            obstacles.pop()

    def draw(self, SCREEN):
        SCREEN.blit(self.image[self.type], self.rect)


class SmallCactus(Obstacle):
    def __init__(self, image):
        self.type = random.randint(0, 2)
        super().__init__(image, self.type)
        self.rect.y = 325
```

Score

The score() function increments the points variable by one for every frame rendered. If the points reach a multiple of 100, it increases the game speed, adding to the game's difficulty over time.Using the pygame.font.render() function, it creates a text surface displaying the current points.

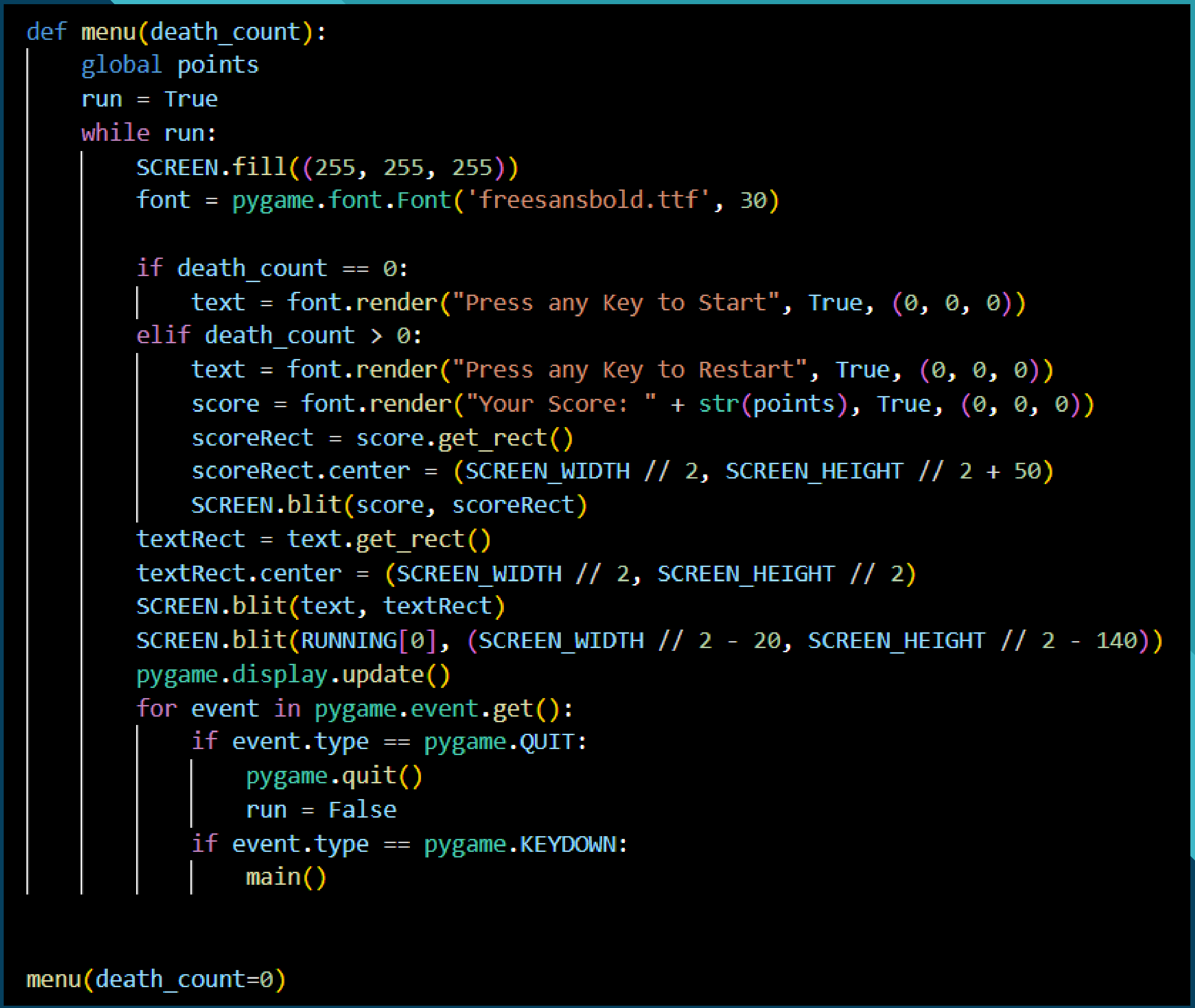


```python
def score():
    global points, game_speed
    points += 1
    if points % 100 == 0:
        game_speed += 1

    text = font.render("Points: " + str(points), True, (0, 0, 0))
    textRect = text.get_rect()
    textRect.center = (1000, 40)
    SCREEN.blit(text, textRect)
```

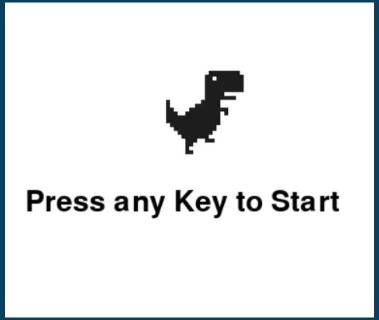


Death and Menu

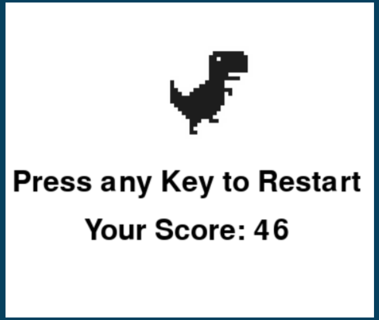
The `menu()` function manages the game menu, taking `death_count` as input. If `death_count` is 0, it prompts the user to start the game. Otherwise, it prompts to restart along with the player's score. Upon key press, it starts or restarts the game by calling `main()`.



```python
def menu(death_count):
    global points
    run = True
    while run:
        SCREEN.fill((255, 255, 255))
        font = pygame.font.Font('freesansbold.ttf', 30)

        if death_count == 0:
            text = font.render("Press any Key to Start", True, (0, 0, 0))
        elif death_count > 0:
            text = font.render("Press any Key to Restart", True, (0, 0, 0))
            score = font.render("Your Score: " + str(points), True, (0, 0, 0))
            scoreRect = score.get_rect()
            scoreRect.center = (SCREEN_WIDTH // 2, SCREEN_HEIGHT // 2 + 50)
            SCREEN.blit(score, scoreRect)
        textRect = text.get_rect()
        textRect.center = (SCREEN_WIDTH // 2, SCREEN_HEIGHT // 2)
        SCREEN.blit(text, textRect)
        SCREEN.blit(RUNNING[0], (SCREEN_WIDTH // 2 - 20, SCREEN_HEIGHT // 2 - 140))
        pygame.display.update()
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                run = False
            if event.type == pygame.KEYDOWN:
                main()


menu(death_count=0)
```



Press any Key to Start



Press any Key to Restart
Your Score: 46

Conclusion

This presentation shows how Pygame makes it easy to create fun 2D games in Python. With things like moving characters, buttons, and keeping track of scores, it’s easier to make games that people enjoy playing. Pygame's simple language and helpful tools make it perfect for anyone who wants to try making games.

Thank you

Presented by

Name	:	Minhazur Rahman
ID	:	2215151032
Section	:	5A2

Name	:	Kayes Mahmood
ID	:	2215151030
Section	:	5A2

Name	:	Redwanul Islam Rafi
ID	:	2215151043
Section	:	5A2