c0a22080fa / ProjExD_05-1

Code Pull requests Actions Projects Wiki Security Insights

main ProjExD_05-1 / Super_Danio.py Go to file t

c0a22120 11 minutes ago

227 lines (188 loc) · 7.75 KB

Code Blame

```python
import sys
import random
import pygame as pg
from pygame.math import Vector2
import time

# ゲームのウィンドウサイズ
WIDTH, HEIGHT = 800, 600

# 色の定義
WHITE = (255, 255, 255)

# ゲームの初期化
pg.init()
screen = pg.display.set_mode((WIDTH, HEIGHT))
pg.display.set_caption("横スクロールジャンプゲーム")
clock = pg.time.Clock()
jump_sound = pg.mixer.Sound("ex05/fig/jump.mp3")  # 2. Load the "power.mp3" file into a Sound o


# 背景画像の読み込み
bg_img = pg.image.load("ex05/fig/pg_bg.jpg")
bg_img = pg.transform.scale(bg_img, (WIDTH, HEIGHT))
bg_imgs = [bg_img, pg.transform.flip(bg_img, True, False)] * 4

score = 0

class Player(pg.sprite.Sprite):
    def __init__(self):
        super().__init__()
        self.image = pg.image.load("ex05/fig/danieru.png")
        self.image = pg.transform.rotozoom(self.image, 0, 0.5)
        self.rect = self.image.get_rect()
        self.rect.topleft = (200, 400)
        self.velocity = pg.Vector2(0, 0)
        self.gravity = 0.4
        self.jumping = False
```

```
        self.jump_count = 0  # 二段ジャンプの回数を追加
        self.invincible = False
        self.invincible_timer = 0
        self.power_up_effect_active = False  # パワーアップの効果が発動中かどうか
        self.power_up_effect_duration = 300  # パワーアップの効果が持続する時間

    def update(self):
        if self.power_up_effect_active:
            self.power_up_effect_duration -= 1
            if self.power_up_effect_duration <= 0:
                self.end_power_up_effect()


        self.handle_gravity()
        self.rect.move_ip(self.velocity)

        if self.rect.top >= HEIGHT:
            self.rect.bottom = HEIGHT
            self.velocity.y = 0
            self.jumping = False
            self.jump_count = 0  # 地面に着地したらジャンプ回数をリセットする

        # 無敵状態の処理
        if self.invincible:
            self.invincible_timer -= 1

            if self.invincible_timer <= 0:
                self.invincible = False
                self.image = pg.image.load("ex05/fig/danieru.png")  # 元の画像に戻す
                self.image = pg.transform.rotozoom(self.image, 0, 0.5)

        # パワーアップ状態の画像を更新
        if self.power_up_effect_active:
            self.image = pg.image.load("ex05/fig/power.png")
            self.image = pg.transform.flip(pg.transform.rotozoom(self.image, 0, 0.4),True, Fals

    def handle_gravity(self):
        if self.jumping:  # ジャンプ中のみ重力をかける
            self.velocity.y += self.gravity

    def jump(self):
        if not self.jumping or self.jump_count < 2:  # 二段ジャンプの条件を修正
            self.velocity.y = -17
            self.jumping = True
            self.jump_count += 1
            jump_sound.play()

    def set_invincible(self):
        if not self.invincible:
            self.invincible = True
            self.invincible_timer = 300
            invincible_image = pg.image.load("ex05/fig/supadanieru.png")
            invincible_image = pg.transform.rotozoom(invincible_image, 0, 0.7)
            self.image = pg.transform.flip(invincible_image, True, False)
```

```python

        def set_power_up(self, score):
            if not self.power_up_effect_active and score >= 100:
                self.power_up_effect_active = True

        def end_power_up_effect(self):
            self.power_up_effect_active = False
            self.image = pg.image.load("ex05/fig/danieru.png")  # 元の画像に戻す
            self.image = pg.transform.rotozoom(self.image, 0, 0.5)


    # コインクラス
    class Coin(pg.sprite.Sprite):
        def __init__(self):
            super().__init__()
            self.image = pg.image.load("ex05/fig/coin.png")
            self.image = pg.transform.scale(self.image, (40, 40))
            self.rect = self.image.get_rect()
            self.rect.topleft = (WIDTH, random.randint(HEIGHT // 2, HEIGHT - 30))
            self.speed = -5

        def update(self):
            self.rect.move_ip(self.speed, 0)
            if self.rect.right <= 0:
                self.reset()

        def reset(self):
            self.rect.topleft = (WIDTH, random.randint(HEIGHT // 2, HEIGHT - 30))

    # 障害物クラス
    class Obstacle(pg.sprite.Sprite):
        def __init__(self):
            super().__init__()
            self.image = pg.image.load("ex05/fig/unnko.png")
            self.image = pg.transform.rotozoom(self.image, 0, 2)
            self.rect = self.image.get_rect()
            self.rect.topleft = (WIDTH, HEIGHT - self.rect.height)
            self.speed = 10

            # 障害物の判定用矩形を設定
            self.hitbox = pg.Rect(self.rect.x + 10, self.rect.y + 10, self.rect.width - 40, self.re

        def update(self):
            self.rect.move_ip(-self.speed, 0)
            self.hitbox.move_ip(-self.speed, 0)  # 判定用矩形も移動させる
            if self.rect.right <= 0:
                self.reset()

        def reset(self):
            # 障害物の出現位置をランダムに設定
            self.rect.topleft = (WIDTH, random.randint(HEIGHT // 2, HEIGHT - 30))
            self.hitbox.topleft = (WIDTH, random.randint(HEIGHT // 2, HEIGHT - 30))  # 判定用矩形も

    def main():
```

```
    # スプライトグループ
    all_sprites = pg.sprite.Group()
    coins = pg.sprite.Group()
    obstacles = pg.sprite.Group()
    score = 0  # スコアを初期化
    player = Player()
    all_sprites.add(player)

    # ゲームループ
    running = True
    bg_x = 0  # 背景画像のx座標
    bg_speed = 2  # 背景画像のスクロールスピード
    next_obstacle_time = 170# 次の障害物が出現するまでの時間
    obstacle_interval = 2000  # 障害物の出現間隔（ミリ秒）
    while running:
        for event in pg.event.get():
            if event.type == pg.QUIT:
                running = False
            elif event.type == pg.KEYDOWN:
                if event.key == pg.K_SPACE:
                    player.jump()
                elif event.key == pg.K_LSHIFT:
                    if score >= 50:
                        score -= 50
                        player.set_invincible()
        player.set_power_up(score)
        player.update()

        # コインを追加
        if len(coins) < 5 and random.randint(0, 100) < 10:
            coin = Coin()
            coins.add(coin)
            all_sprites.add(coin)

        # 障害物を追加
        current_time = pg.time.get_ticks()
        if current_time > next_obstacle_time:
            obstacle = Obstacle()
            obstacles.add(obstacle)
            all_sprites.add(obstacle)
            next_obstacle_time = current_time + random.randint(obstacle_interval // 1, obstacle


        # 衝突判定
        hits = pg.sprite.spritecollide(player, coins, True)
        if hits:
            score += 10

        hits = pg.sprite.spritecollide(player, obstacles, False)
        if hits and not player.invincible:
            running = False

        all_sprites.update()
```

```python
            # 背景画像のスクロール
            bg_x -= bg_speed
            if bg_x <= -WIDTH:
                bg_x = 0

            # 画面描画
            screen.fill((0, 0, 0))
            screen.blit(bg_imgs[0], (bg_x, 0))
            screen.blit(bg_imgs[1], (bg_x + WIDTH, 0))
            all_sprites.draw(screen)
            font = pg.font.Font(None, 36)
            score_text = font.render("Score: {}".format(score), True, (255, 255, 255))
            screen.blit(score_text, (10, 10))
            pg.display.flip()
            clock.tick(60)

        # ゲームオーバー画面
        screen.fill((0, 0, 0))
        font = pg.font.Font(None, 64)
        game_over_text = font.render("Game Over", True, (255, 0, 0))
        screen.blit(game_over_text, (WIDTH // 2 - 150, HEIGHT // 2 - 32))
        pg.display.flip()
        time.sleep(2)  # 2秒間待機してから終了
        pg.quit()
        sys.exit()

    if __name__ == "__main__":
        main()
```