187 lines (155 loc) · 5.35 KB

Code Blame Raw

```python
import pygame
import sys
import random

# 画面のサイズ
SCREEN_WIDTH = 1100
SCREEN_HEIGHT = 700
FPS = 60

# 色
BLACK = (0, 0, 0)
WHITE = (255, 255, 255)
RED = (255, 0, 0)
GREEN = (0, 255, 0)

# ブロックの上端のy座標（背景画像に合わせて調整）
GROUND_Y = 610

def check_bound(obj_rct:pygame.Rect) -> tuple[bool,bool]:
    """
    オブジェクトが画面内or画面外を判定し，真理値タプルを返す関数
    引数：こうかとんや敵などのRect
    戻り値：横方向，縦方向のはみ出し判定結果（画面内：True／画面外：False）
    """
    yoko, tate = True, True
    if obj_rct.left < 0 or SCREEN_WIDTH < obj_rct.right:
        yoko = False
    if obj_rct.top < 0 or SCREEN_HEIGHT < obj_rct.bottom:
        tate = False
    return yoko, tate

class Bird(pygame.sprite.Sprite):
    def __init__(self):
        super().__init__()
        self.image = pygame.image.load("fig/3.png")
        self.rect = self.image.get_rect()
```

```
37              self.rect.x = 50
38              self.rect.bottom = GROUND_Y  # ブロックの上に乗せる
39              self.speed_x = 0
40              self.speed_y = 0
41              self.gravity = 1
42              self.jump_power = -20
43              self.is_jumping = False
44              self.world_x = 50  # ワールド座標
45
46          def update(self):
47              # 左右移動
48              self.world_x += self.speed_x
49              # プレイヤーの画面上のx座標は後で調整
50
51              # 重力
52              self.speed_y += self.gravity
53              self.rect.y += self.speed_y
54
55              # 地面との衝突判定（ブロックの上）
56              if self.rect.bottom > GROUND_Y:
57                  self.rect.bottom = GROUND_Y
58                  self.speed_y = 0
59                  self.is_jumping = False
60
61          def jump(self):
62              if not self.is_jumping:
63                  self.speed_y = self.jump_power
64                  self.is_jumping = True
65
66
67      class Enemy(pygame.sprite.Sprite):
68          """
69          敵のクラス
70          """
71          def __init__(self):
72              img = pygame.image.load("fig/0.png")
73              super().__init__()
74              self.image = img
75              self.rect = self.image.get_rect()
76
77              self.rect.x = random.randint(100, SCREEN_WIDTH - 100)
78              self.world_x = random.randint(100, SCREEN_WIDTH - 100)  # ワールド座標
79              self.rect.y = -self.rect.height
80
81              self.speed_y = 0
82              self.gravity = 10
83              self.speed = 0
84              self.is_landed = False
85
86          def update(self):
87              """
88              敵の速度self.speed
```

```
89              落下速度 self.speed_y
90              引数  screen:画面Surface
91              """
92              if not self.is_landed:
93                  self.speed_y += self.gravity
94                  self.rect.y = self.speed_y
95
96                  if self.rect.bottom >= GROUND_Y:
97                      self.rect.bottom = GROUND_Y
98                      self.is_landed = True
99                      self.speed_y = 0
100                 else:
101                     self.rect.x += self.speed
102             else:
103                 self.speed = -3
104                 self.world_x += self.speed
105
106
107                 self.rect.move_ip(self.speed,0)
108             if check_bound(pygame.Rect(self.world_x, self.rect.y, self.rect.width, self.rect.h
109                 self.kill()
110
111 def main():
112     pygame.init()
113     screen = pygame.display.set_mode((SCREEN_WIDTH, SCREEN_HEIGHT))
114     pygame.display.set_caption("スーパーこうかとんブラザーズ")
115     clock = pygame.time.Clock()
116
117     # 背景画像のロード
118     bg_img = pygame.transform.rotozoom(pygame.image.load("fig/pg_bg.png").convert(), 0, 2.
119     bg_width = bg_img.get_width()
120     bg_height = bg_img.get_height()
121
122     bird = Bird()
123     emys = pygame.sprite.Group()
124
125     scroll_x = 0  # 背景のスクロール量
126     tmr = 0
127
128
129     while True:
130         for event in pygame.event.get():
131             if event.type == pygame.QUIT:
132                 return 0
133             if event.type == pygame.KEYDOWN:
134                 if event.key == pygame.K_LEFT:
135                     bird.speed_x = -5
136                 if event.key == pygame.K_RIGHT:
137                     bird.speed_x = 5
138                 if event.key == pygame.K_SPACE:
139                     bird.jump()
140             if event.type == pygame.KEYUP:
```

```python
                if event.key == pygame.K_LEFT and bird.speed_x < 0:
                    bird.speed_x = 0
                if event.key == pygame.K_RIGHT and bird.speed_x > 0:
                    bird.speed_x = 0


        # プレイヤーが画面中央より右に行ったら背景をスクロール
        center_x = SCREEN_WIDTH // 2
        if bird.world_x > center_x:
            scroll_x = bird.world_x - center_x
        else:
            scroll_x = 0
        # 背景の範囲外に行かないように制限
        max_scroll = bg_width - SCREEN_WIDTH
        if scroll_x > max_scroll:
            scroll_x = max_scroll
        if scroll_x < 0:
            scroll_x = 0

        # プレイヤーの画面上のx座標を調整
        if bird.world_x > center_x:
            bird.rect.x = center_x
        else:
            bird.rect.x = bird.world_x

        if tmr%20 == 0:
            emys.add(Enemy())


        # 描画
        screen.blit(bg_img, (-scroll_x, 0))
        bird.update()
        screen.blit(bird.image,bird.rect)
        emys.update()
        for emy in emys:
            emy.rect.x = emy.world_x - scroll_x  # スクロール補正
            screen.blit(emy.image, emy.rect)
        pygame.display.flip()

        tmr += 1
        clock.tick(50)


if __name__ == '__main__':
    main()
    pygame.quit()
    sys.exit()
```