

c0c24020ae /

ProjExD\_Group05

<> Code

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

177e18b

ProjExD\_Group05 / README.md

Go to file

t

c0a24017 微修正

177e18b · last week

39 lines (33 loc) · 1.12 KB

Preview

Code

Blame

Raw

# スーパーこうかтонブラザーズ

## 実行環境の必要条件

- python >= 3.10
- pygame >= 2.1

## ゲームの概要

- 主人公のこうかтонがさらわれたこうか姫を助けるゲーム

## ゲームの遊び方

- 十字キーでキャラを操作する
- スペースキーでジャンプ

## ゲームの実装

### 共通基本機能



- 背景画像と主人公キャラクターの描画
- 背景がキャラとともに動く
- 十字キーでキャラを動かす
- 地形を作る
- ゴールでゲーム終了
- 地形に乗ることができる
- ジャンプできる




### 分担追加機能

- 敵を踏んで倒せる (り)
- 穴に落ちたら死ぬ (り)
- 敵の出現 (加藤)
- 敵が壁に反射 (加藤)
- ダッシュ (河辺)
- 敵に当たったら死ぬ (河辺)
- ブロックを叩ける(体が大きければ壊せる) (束田)
- アイテムの出現 (束田)
- 食べると体が大きくなるアイテム (波部)

## ToDo

- キャラクターの変数名の統一
- 敵キャラの変数名決め
- 登場キャラの決定
- アイテム決定

 c0c24020ae / ProjExD\_Group05



[Code](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

 C0C24020/dash ▾

ProjExD\_Group05

Go to file  t 

/ super\_koukaton.py 



c0c24020ae push

440b9b3 · 1 minute ago




156 lines (125 loc) · 4.29 KB

```
1  import pygame
2  import random
3
4
5  # 画面のサイズ
6  SCREEN_WIDTH = 1100
7  SCREEN_HEIGHT = 700
8  FPS = 60
9
10 # 色
11 BLACK = (0, 0, 0)
12 WHITE = (255, 255, 255)
13 RED = (255, 0, 0)
14 GREEN = (0, 255, 0)
15
16 # 地面のY座標
17 GROUND_Y = 610
18
19 class Bird(pygame.sprite.Sprite):
20     def __init__(self):
21         super().__init__()
22         self.image = pygame.Surface([30, 50])
23         self.image.fill(RED)
24         self.rect = self.image.get_rect()
25         self.rect.x = 50
26         self.rect.bottom = GROUND_Y
27         self.speed_x = 0
28         self.speed_y = 0
29         self.gravity = 1
30         self.jump_power = -20
31         self.is_jumping = False
32         self.world_x = 50
33
34     def update(self):
35         self.world_x += self.speed_x
36         self.speed_y += self.gravity
37         self.rect.y += self.speed_y
38
39         if self.rect.bottom > GROUND_Y:
40             self.rect.bottom = GROUND_Y
41             self.speed_y = 0
```

```
42         self.is_jumping = False
43
44     def jump(self):
45         if not self.is_jumping:
46             self.speed_y = self.jump_power
47             self.is_jumping = True
48
49     class Enemy(pygame.sprite.Sprite):
50     def __init__(self, x, y):
51         super().__init__()
52         self.image = pygame.Surface([40, 40])
53         self.image.fill(GREEN)
54         self.rect = self.image.get_rect()
55         self.rect.x = x
56         self.rect.y = y
57         self.speed_x = -5
58
59     def update(self):
60         self.rect.x += self.speed_x
61         if self.rect.right < 0:
62             self.kill()
63
64     def main():
65         pygame.init()
66         screen = pygame.display.set_mode((SCREEN_WIDTH, SCREEN_HEIGHT))
67         pygame.display.set_caption("スーパーこうかとんブラザーズ")
68         clock = pygame.time.Clock()
69
70         # 背景画像
71         bg_img = pygame.transform.rotozoom(pygame.image.load("ex5/fig/pg_bg.png").convert(), 0, 2.92)
72         bg_width = bg_img.get_width()
73
74         all_sprites = pygame.sprite.Group()
75         enemies = pygame.sprite.Group()
76         bird = Bird()
77         all_sprites.add(bird)
78
79         scroll_x = 0
80         running = True
81         game_over = False
82
83         enemy_spawn_timer = 0
84         enemy_spawn_interval = 90 # 約1.5秒
85
86         font = pygame.font.Font(None, 80) # フォント準備
87
88         while running:
89             for event in pygame.event.get():
90                 if event.type == pygame.QUIT:
91                     running = False
92                 if event.type == pygame.KEYDOWN:
93                     if event.key == pygame.K_SPACE and not game_over:
94                         bird.jump()
95
96                 if not game_over:
97                     keys = pygame.key.get_pressed()
98                     base_speed = 5
99                     speed = base_speed * 1.5 if keys[pygame.K_LSHIFT] else base_speed
```


```
100
101     if keys[pygame.K_LEFT]:
102         bird.speed_x = -speed
103     elif keys[pygame.K_RIGHT]:
104         bird.speed_x = speed
105     else:
106         bird.speed_x = 0
107
108     all_sprites.update()
109     enemies.update()
110
111     enemy_spawn_timer += 1
112     if enemy_spawn_timer >= enemy_spawn_interval:
113         enemy_spawn_timer = 0
114         enemy_x = scroll_x + SCREEN_WIDTH + 20
115         enemy_y = GROUND_Y - 40
116         enemy = Enemy(enemy_x, enemy_y)
117         enemies.add(enemy)
118         all_sprites.add(enemy)
119
120     center_x = SCREEN_WIDTH // 2
121     if bird.world_x > center_x:
122         scroll_x = bird.world_x - center_x
123     else:
124         scroll_x = 0
125
126     max_scroll = bg_width - SCREEN_WIDTH
127     scroll_x = max(0, min(scroll_x, max_scroll))
128
129     if bird.world_x > center_x:
130         bird.rect.x = center_x
131     else:
132         bird.rect.x = bird.world_x
133
134     if pygame.sprite.spritecollideany(bird, enemies):
135         game_over = True
136         game_over_time = pygame.time.get_ticks() # ゲームオーバーになった時間
137
138     screen.blit(bg_img, (-scroll_x, 0))
139     all_sprites.draw(screen)
140
141     if game_over:
142         text = font.render("ゲームオーバー！", True, (255, 0, 0))
143         rect = text.get_rect(center=(SCREEN_WIDTH // 2, SCREEN_HEIGHT // 2))
144         screen.blit(text, rect)
145
146         # ゲームオーバーから2秒経過したら終了
147         if pygame.time.get_ticks() - game_over_time > 2000:
148             running = False
149
150     pygame.display.flip()
151     clock.tick(FPS)
152
153     pygame.quit()
154
155 if __name__ == "__main__":
156     main()
```


 C0C24020/dash ▾ [ProjExD\\_Group05](#) / super\_koukaton.py ↑ Top


Code

Blame


Raw







▾



64

`def main():`