



c0a2407017 /
プロジェクトExD_グループ05



<> コード

🕒 問題

🔗 プルリクエスト

🎬 アクション

📁 プロジェクト

📖 ウィキ



コミット 48e0a21



c0a24017 コミットした 7 minutes ago

機能:敵の出現

🔗 [C0A24070/敵の出現](#)

親1人[177e18b](#)専念48e0a21

📁 4つのファイルが変更されました +87 -15 行が変更されました

↑ トップ



🔍 ファイルをフィルタリング...



📄 README.md

▼ 📁 イチジク

📄 0.png

📄 3.png

📄 super_koukaton.py

📁 4つのファイルが変更されました +87 -15 行が変更されました

🔍 コード内検索



▼ README.md

+ 3 00000



```
↑... @@ -37,3 +37,6 @@
37 37 - 登場キャラの決定
38 38 ☐ - アイテム決定
39 39
40 + ### ✖モ
41 + * 敵のクラス名はEnemy
42 +
```

▼ fig/0.png

6.53 KB





▼ fig/3.png

5.32 KB



...



▼ super_koukaton.py

+ 84 - 15 ...

```
...    @@ -1,4 +1,6 @@
1      - import pygame
      1  + import pygame
      2  + import sys
      3  + import random
2      4
3      5      # 画面のサイズ
4      6      SCREEN_WIDTH = 1100
@@    @@ -14,11 +16,23 @@
14     16      # ブロックの上端のy座標（背景画像に合わせて調整）
15     17      GROUND_Y = 610 # 必要に応じて微調整してください
16     18
19     + def check_bound(obj_rct:pygame.Rect) -> tuple[bool,bool]:
20     +     """
21     +     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
22     +     引数：こうかとんや敵などのRect
23     +     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）
24     +     """
25     +     yoko, tate = True, True
26     +     if obj_rct.left < 0 or SCREEN_WIDTH < obj_rct.right:
27     +         yoko = False
28     +     if obj_rct.top < 0 or SCREEN_HEIGHT < obj_rct.bottom:
29     +         tate = False
30     +     return yoko, tate
31     +
17     32      class Bird(pygame.sprite.Sprite):
18     33          def __init__(self):
19     34              super().__init__()
20     -         self.image = pygame.Surface([30, 50])
```

```

21      -        self.image.fill(RED)
35      +        self.image = pygame.image.load("fig/3.png")
22      36        self.rect = self.image.get_rect()
23      37        self.rect.x = 50
24      38        self.rect.bottom = GROUND_Y  # ブロックの上に乗せる
      ↓
      ↑
      @@ -50,28 +64,74 @@ def jump(self):
50      64            self.is_jumping = True
51      65
52      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.height)) != (True, True):
109 +                 self.kill()
110 +
53 111 def main():
54 112     pygame.init()
55 113     screen = pygame.display.set_mode((SCREEN_WIDTH, SCREEN_HEIGHT))
56 114     pygame.display.set_caption("スーパーこうかとんブラザーズ")
57 115     clock = pygame.time.Clock()
58 116
59 117     # 背景画像のロード
60     -     bg_img =
        pygame.transform.rotozoom(pygame.image.load("ex5/fig/pg_bg.png").convert(), 0,
2.92)
118 +     bg_img =
        pygame.transform.rotozoom(pygame.image.load("fig/pg_bg.png").convert(), 0, 2.92)
61 119     bg_width = bg_img.get_width()
62 120     bg_height = bg_img.get_height()
63 121
64     -     all_sprites = pygame.sprite.Group()
65 122     bird = Bird()
66     -     all_sprites.add(bird)
123 +     emys = pygame.sprite.Group()
67 124
68 125     scroll_x = 0 # 背景のスクロール量
69 126
70     -     running = True
71     -     while running:
127 +         tmr = 0
128 +         #emys.add(Enemy())
129 +
130 +
131 +         while True:
72 132             for event in pygame.event.get():
73 133                 if event.type == pygame.QUIT:
```

```

74      -            running = False
      134  +            return 0
75      135            if event.type == pygame.KEYDOWN:
76      136            if event.key == pygame.K_LEFT:
77      137                bird.speed_x = -5
      ⚡ @@ -85,8 +145,6 @@ def main():
85      145            if event.key == pygame.K_RIGHT and bird.speed_x > 0:
86      146                bird.speed_x = 0
87      147
88      -            # ゲームループ
89      -            all_sprites.update()
90      148
91      149            # 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
92      150            center_x = SCREEN_WIDTH // 2
      ⚡ @@ -107,14 +165,25 @@ def main():
107     165            else:
108     166                bird.rect.x = bird.world_x
109     167
110     168  +            if tmr%20 == 0:
111     169  +                emys.add(Enemy())
112     170  +
113     171  +
114     172            # 描画
115     173            screen.blit(bg_img, (-scroll_x, 0))
116     174  +            all_sprites.draw(screen)
117     175  +            bird.update()
118     176  +            screen.blit(bird.image, bird.rect)
119     177  +            emys.update()
120     178  +            for emy in emys:
121     179  +                emy.rect.x = emy.world_x - scroll_x # スクロール補正
122     180  +                screen.blit(emy.image, emy.rect)
123     181            pygame.display.flip()
124     182
125     183            clock.tick(FPS)
126     184  +            tmr += 1
127     185  +            clock.tick(50)
128     186
129     187            pygame.quit()
130     188
131     189            if __name__ == '__main__':
132     190  +            main()
133     191
134     192  +            main()

```

```
188 +     pygame.quit()
```

```
189 +     sys.exit()
```



コメント 0

 会話をロックする



Comment



Unsubscribe

You're receiving notifications because you're subscribed to this thread.

c0a2407017 /
ProjExD_Group05

<> Code

Issues

Pull requests

Actions

Projects

Wiki



C0A24070/Enem... ▾

ProjExD_Group05 / super_koukaton.py



c0a24017 修正

e30f808 · 1 minute ago



187 lines (155 loc) · 5.35 KB

Code

Blame



Raw



```
1  import pygame
2  import sys
3  import random
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 # ブロックの上端のy座標（背景画像に合わせて調整）
17 GROUND_Y = 610
18
19 ▾ def check_bound(obj_rct:pygame.Rect) -> tuple[bool,bool]:
20     """
21     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
22     引数：こうかとんや敵などのRect
23     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）
24     """
25     yoko, tate = True, True
26     if obj_rct.left < 0 or SCREEN_WIDTH < obj_rct.right:
27         yoko = False
28     if obj_rct.top < 0 or SCREEN_HEIGHT < obj_rct.bottom:
29         tate = False
30     return yoko, tate
31
32 ▾ class Bird(pygame.sprite.Sprite):
33     ▾ def __init__(self):
34         super().__init__()
35         self.image = pygame.image.load("fig/3.png")
36         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
72     def __init__(self):
73         img = pygame.image.load("fig/0.png")
74         super().__init__()
75         self.image = img
76         self.rect = self.image.get_rect()
77
78         self.rect.x = random.randint(100, SCREEN_WIDTH - 100)
79         self.world_x = random.randint(100, SCREEN_WIDTH - 100) # ワールド座標
80         self.rect.y = -self.rect.height
81
82         self.speed_y = 0
83         self.gravity = 10
84         self.speed = 0
85         self.is_landed = False
86
87     def update(self):
88         """
89         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:
```

```
141         if event.key == pygame.K_LEFT and bird.speed_x < 0:
142             bird.speed_x = 0
143         if event.key == pygame.K_RIGHT and bird.speed_x > 0:
144             bird.speed_x = 0
145
146
147     # 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
148     center_x = SCREEN_WIDTH // 2
149     if bird.world_x > center_x:
150         scroll_x = bird.world_x - center_x
151     else:
152         scroll_x = 0
153     # 背景の範囲外に行かないように制限
154     max_scroll = bg_width - SCREEN_WIDTH
155     if scroll_x > max_scroll:
156         scroll_x = max_scroll
157     if scroll_x < 0:
158         scroll_x = 0
159
160     # プレイヤーの画面上のx座標を調整
161     if bird.world_x > center_x:
162         bird.rect.x = center_x
163     else:
164         bird.rect.x = bird.world_x
165
166     if tmr%20 == 0:
167         emys.add(Enemy())
168
169
170     # 描画
171     screen.blit(bg_img, (-scroll_x, 0))
172     bird.update()
173     screen.blit(bird.image, bird.rect)
174     emys.update()
175     for emy in emys:
176         emy.rect.x = emy.world_x - scroll_x # スクロール補正
177         screen.blit(emy.image, emy.rect)
178     pygame.display.flip()
179
180     tmr += 1
181     clock.tick(50)
182
183
184 if __name__ == '__main__':
185     main()
186     pygame.quit()
187     sys.exit()
```