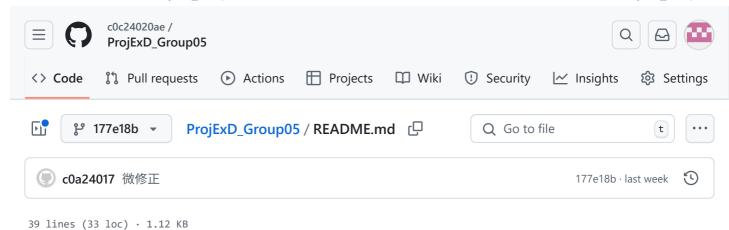
Raw □ ± 0 - :=

Preview





♂ 実行環境の必要条件

Code Blame

- python >= 3.10
- pygame >= 2.1

ゲームの概要

• 主人公のこうかとん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

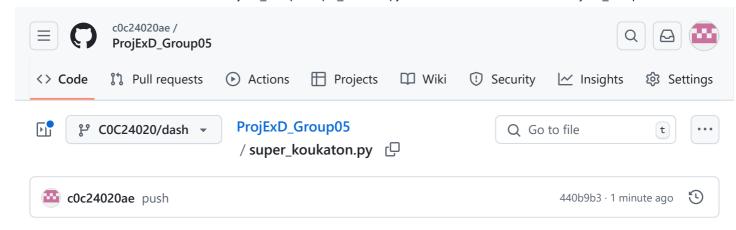
- 背景画像と主人公キャラクターの描画
- 背景がキャラとともに動く
- 十字キーでキャラを動かす
- 地形を作る
- ゴールでゲーム終了
- 地形に乗ることができる
- ジャンプできる

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)(東田)
- アイテムの出現 (東田)
- 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



```
156 lines (125 loc) · 4.29 KB
```

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

```
self.is_jumping = False
42
43
           def jump(self):
45
               if not self.is_jumping:
46
                    self.speed_y = self.jump_power
                    self.is_jumping = True
47
48
49
       class Enemy(pygame.sprite.Sprite):
           def __init__(self, x, y):
50
51
               super().__init__()
               self.image = pygame.Surface([40, 40])
52
               self.image.fill(GREEN)
53
               self.rect = self.image.get_rect()
54
               self.rect.x = x
55
56
               self.rect.y = y
               self.speed_x = -5
57
58
           def update(self):
59
               self.rect.x += self.speed_x
60
61
               if self.rect.right < 0:</pre>
                    self.kill()
62
63
      def main():
64
           pygame.init()
65
           screen = pygame.display.set mode((SCREEN WIDTH, SCREEN HEIGHT))
66
           pygame.display.set_caption("スーパーこうかとんブラザーズ")
67
68
           clock = pygame.time.Clock()
69
           # 背景画像
70
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()
           enemies = pygame.sprite.Group()
75
           bird = Bird()
76
77
           all_sprites.add(bird)
78
           scroll x = 0
79
           running = True
80
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:
               for event in pygame.event.get():
89
                    if event.type == pygame.QUIT:
90
91
                        running = False
                    if event.type == pygame.KEYDOWN:
92
93
                        if event.key == pygame.K_SPACE and not game_over:
                            bird.jump()
94
95
               if not game_over:
96
97
                    keys = pygame.key.get_pressed()
98
                    base\_speed = 5
                    speed = base_speed * 1.5 if keys[pygame.K_LSHIFT] else base_speed
```

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

