

SorataOda /
ProjExD2023_ex05

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights



main

ProjExD2023_ex05 / game.py

Go to file

t



C0A22088 10 minutes ago



333 lines (280 loc) · 9.9 KB

Code

Blame



```
1 import sys
2 import random
3 import shelve
4 import sys
5 import time
6 from typing import Any
7 import pygame as pg
8 from pygame.sprite import AbstractGroup
9
10
11 WIDTH = 1600
12 HEIGHT = 900
13
14 def check_bound(obj: pg.Rect) -> tuple[bool, bool]:
15     """
16     オブジェクトが画面内か画面外かを判定し、真理値タプルを返す
17     引数 obj: オブジェクト hito SurfaceのRect
18     戻り値: 横方向, 縦方向のはみ出し判定結果 (画面内: True/画
19     """
20     yoko, tate = True, True
21     if obj.rect.left < 0: # 横方向のはみ出し判定
22         yoko = False
23     if obj.rect.top < 0 or HEIGHT < obj.rect.bottom: # 縦
24         tate = False
25     return yoko, tate
26
27 class Hito(pg.sprite.Sprite):
28     """
29     操作する人に関するクラス
30     """
31
32     def __init__(self):
33         """
34         初期化メソッド
35         """
36         self.img=pg.image.load("ex05/fig/hito1.png")
37         self.img = pg.transform.rotozoom(self.img,0,0.25)
38         self.img=pg.transform.flip(self.img,True,False)
39         self.rect = self.img.get_rect()
40         self.rect.center = WIDTH/2,(HEIGHT/2)+165
41         self.type = "run"
42         self.state = "nomal"
43         self.janptop = -27
```

Symbols



Find definitions and references for functions and other symbols in this file by clicking a symbol below or in the code.

Filter symbols

r

const WIDTH

const HEIGHT

func check_bound

class Hito

func __init__

func change_state

func item_use

func change_img

func update

class Score

func __init__

func score_up

func update

class Item

func __init__

func update

class Coin

func __init__

func update

class Obstacle

func __init__

func update

func main

```

45         self.item = False
46         self.vx = -1
47         self.item_life = 0
48         self.hyper_life = -1
49
50     def change_state(self, state: str, hyper_life: int):
51         self.state = state
52         self.hyper_life = hyper_life
53
54
55     def item_use(self, life: int):
56         self.item = True
57         self.item_life = life
58
59     def change_img(self, name: str, num: int, screen: pg.Surface):
60         """
61         主人公の画像を差し替えるメソッド
62         引数1: name ファイル名.拡張子
63         引数2: screen
64         """
65         self.img = pg.transform.rotozoom(pg.image.load(f"
66         #self.rect = self.image.get_rect()
67
68         screen.blit(self.img, self.rect)
69
70
71
72     def update(self, screen: pg.Surface):
73         """
74         updateメソッド
75         self.typeを3つにわけ、ジャンプを表現
76         """
77         if self.state == "hyper":
78             self.hyper_life -= 1
79
80         if self.hyper_life == 0:
81             self.change_state("normal", -1)
82             self.img = pg.image.load("ex05/fig/hito1.png")
83             self.img = pg.transform.rotozoom(self.img, 0, 0.
84             self.img = pg.transform.flip(self.img, True, False)
85
86         if self.type == "janpup":
87             self.janp -= 1
88             if self.janp < self.janptop:
89                 self.type = "janpdown"
90                 self.janp = 0
91         elif self.type == "janpdown":
92             self.janp += 1
93             if self.rect.centery >= (HEIGHT/2)+165 :
94                 self.type = "run"
95                 self.janp = 0
96         if self.item == True:
97             if self.rect.centerx > WIDTH:
98                 self.rect.move_ip(0, +self.janp)
99             else:
100                 self.rect.move_ip(+2, +self.janp)
101             self.item_life -= 1
102             if self.item_life == 0:
103                 self.item = False
104         if self.item == False:

```

```
105         self.rect.move_ip(-1,+self.janp)
106
107
108
109
110         screen.blit(self.img, self.rect)
111         #screen.blit(self.img, [WIDTH/2,HEIGHT/2])
112
113
114 ✓ class Score:
115     """
116     取得コインに対応するスコアを表示するクラス
117     """
118 ✓ def __init__(self):
119     self.font = pg.font.Font(None, 50)
120     self.color = (0, 0, 0)
121     self.score = 0
122     self.image = self.font.render(f"Score: {self.score}")
123     self.rect = self.image.get_rect()
124     self.rect.center = 100, 50
125
126     def score_up(self, add): #スコアを増やす関数
127         self.score += add
128
129 ✓ def update(self, screen: pg.Surface): #スコアを更新、表示
130     if 100<self.score<500:
131         self.font=pg.font.Font(None, 55)
132         self.color=(80,0,0)
133     elif 500<=self.score<1000:
134         self.font=pg.font.Font(None, 60)
135         self.color=(160,0,0)
136     elif 1000<=self.score:
137         self.font=pg.font.Font(None, 65)
138         self.color=(255,0,0)
139     self.image = self.font.render(f"Score: {self.score}")
140     screen.blit(self.image,self.rect)
141
142
143
144 ✓ class Item(pg.sprite.Sprite):
145     """
146     アイテムに関するクラス
147     """
148 ✓ def __init__(self):
149     super().__init__()
150     self.image = pg.image.load("ex05/fig/item.png")
151     self.image = pg.transform.rotozoom(self.image,-10,1)
152     self.rect = self.image.get_rect()
153     self.rect.center = WIDTH,random.randint((HEIGHT/2-50),(HEIGHT/2+50))
154     self.vx = -6
155     self.use = "not"
156 ✓ def update(self,):
157     """
158     updateメソッド
159     """
160     self.rect.centerx += self.vx
161     if self.use == "on":
162         self.kill()
163
164 ✓ class Coin(pg.sprite.Sprite):
165     """
```

```

165     """
166     coinに関するクラス
167     """
168     imgs = [pg.image.load(f"ex05/fig/coin{i}.png") for i
169     def __init__(self,num):
170         """
171         """
172         super().__init__()
173         self.image = __class__.imgs[num]
174         #self.image = random.choice(__class__.imgs)
175         #self.image=pg.image.load("ex05/fig/coin1.png")
176         self.image=pg.transform.rotozoom(self.image,0,0.1
177         self.rect=self.image.get_rect()
178         self.rect.centerx=WIDTH
179         self.rect.centery=random.randint(HEIGHT/3,HEIGHT*
180         self.vx = -10
181
182     def update(self):
183         self.rect.centerx+=self.vx
184
185
186     class Obstacle(pg.sprite.Sprite):
187         """
188         障害物に関するクラス
189         """
190     def __init__(self):
191         super().__init__()
192         self.image = pg.image.load("ex05/fig/enemy.png ")
193         self.image = pg.transform.rotozoom(self.image,0,0
194         self.rect = self.image.get_rect()
195         self.rect.center = WIDTH,random.randint((HEIGHT/2
196         self.vx = -10
197     def update(self):
198         """
199         updateメソッド
200         """
201         self.rect.centerx += self.vx
202
203     def main():
204         pg.display.set_caption("gmae")
205         screen = pg.display.set_mode((WIDTH, HEIGHT))
206         clock = pg.time.Clock()
207         score = Score()
208         bg_img = pg.image.load("ex05/fig/pg_bg.jpg")
209         bg_img_2=pg.transform.flip(bg_img,True,False)
210         yuka = pg.image.load("ex05/fig/renga.png")
211
212         items = pg.sprite.Group()
213         hito = Hito()
214         coins1 = pg.sprite.Group()
215         coins2 = pg.sprite.Group()
216         coins3 = pg.sprite.Group()
217         score = Score()
218         coins=pg.sprite.Group()
219         obstecles = pg.sprite.Group()
220
221         tmr = 0
222         x = 0
223
224         while True:
225

```

```

225
226
227     for event in pg.event.get():
228         if event.type == pg.QUIT:
229             return 0
230         if event.type == pg.KEYDOWN and event.key ==
231             if score.score >= 100:
232                 score.score -= 100
233                 hito.change_state("hyper",300)
234                 hito.change_img("UC.NTD.png",1.0,scre
235                 print(hito.state)
236         if event.type == pg.KEYDOWN and event.key ==
237             if hito.type == "run":
238                 hito.type = "janpup"
239
240     #for i, coin in enumerate(coins):
241         #if coin.rect.colliderect(hito.rect):
242             #coins[i] = None
243             #print("b")
244             #pg.display.update()
245
246     x = tmr%3200
247     screen.blit(bg_img, [-x, 0])
248     screen.blit(bg_img_2,[1600-x,0])
249     screen.blit(bg_img,[3200-x,0])
250     screen.blit(yuka,[-x,HEIGHT/2+261])
251     screen.blit(yuka,[1600-x,HEIGHT/2+261])
252     screen.blit(yuka,[3200-x,HEIGHT/2+261])
253     #score.score=10*len(pg.sprite.spritecollide(hito,
254     for i in pg.sprite.spritecollide(hito,coins1,True
255         score.score_up(10)
256     for i in pg.sprite.spritecollide(hito,coins2,True
257         score.score_up(100)
258     for i in pg.sprite.spritecollide(hito,coins3,True
259         score.score_up(500)
260     score.update(screen)
261
262     if tmr%5000 == 0:
263         items.add(Item())
264
265     for item in pg.sprite.spritecollide(hito,items,Tr
266         item.use = "on"
267         hito.item_use(200)
268
269     if tmr%(random.randint(300,1000 )) == 0:
270         obstacles.add(Obstacle())
271
272     for obstacle in pg.sprite.spritecollide(hito,obst
273         if hito.state == "hyper":
274             score.score_up(300)
275         else:
276             gameover_str = pg.image.load("ex05/fig/ga
277             screen.blit(gameover_str,[WIDTH/2-562,HEI
278             hito.change_img("die.png",0.5,screen)
279             obstacles.update()
280             obstacles.draw(screen)
281             #score.update(screen)
282             pg.display.update()
283             time.sleep(2)
284             return
285

```

```
286         yoko,tate = check_bound(hito)
287         if not yoko or not tate:
288             hito.change_img("die.png",0.5,screen)
289             #score.update(screen)
290             gameover_str = pg.image.load("ex05/fig/gameov
291             screen.blit(gameover_str,[WIDTH/2-562,HEIGHT/
292             pg.display.update()
293             time.sleep(2)
294             return
295
296         hito.update(screen)
297
298         items.update()
299         items.draw(screen)
300         if tmr%random.randint(1,1500)==0:
301             num=random.randint(0,3)
302             if num==0:
303                 coins1.add(Coin(num))
304             elif num==1:
305                 coins2.add(Coin(num))
306             elif num==2:
307                 coins3.add(Coin(num))
308
309         coins1.update()
310         coins1.draw(screen)
311         coins2.update()
312         coins2.draw(screen)
313         coins3.update()
314         coins3.draw(screen)
315
316         coins.update()
317         coins.draw(screen)
318
319         obstecles.update()
320         obstecles.draw(screen)
321
322
323         pg.display.update()
324
325         tmr += 10
326         clock.tick(100)
327
328
329 if __name__ == "__main__":
330     pg.init()
331     main()
332     pg.quit()
333     sys.exit()
```