

AokiYoshiaki / ex05 Public

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [...](#)[main](#) [ex05 / tower.py](#)

Go to file

t

...



c0b21181 7 minutes ago

...



298 lines (242 loc) · 10.1 KB

[main](#)[ex05 / tower.py](#)[↑ Top](#)

Code

Blame



...

```
162 def main():
4     import time
5     from typing import Any
6     import pygame
7     import pygame as pg
8     from pygame.sprite import AbstractGroup
9     import pygame.mixer
10
11     WIDTH = 1600
12     HEIGHT = 900
13
14     class tower(pg.sprite.Sprite):
15         """
16         自分と敵のタワーに関するクラス
17         1,init
18         引数はHPと位置を示すタプル
19         バベルの塔の画像を表示させ指定された位置に置く
20         読み取り用属性は"Tower"である
21         2,update
22         HPが0になったときグループから消去する
23         """
24     def __init__(self, hp, xy: tuple[int, int]):
25         super().__init__()
26         self.image = pg.transform.rotozoom(pg.image.load("ex05/fig/tower.png"), 0.5, 0.5)
27         self.hp = hp
28         self.rect = self.image.get_rect()
29         self.rect.center = xy #位置X,Y
30         self.zokusei = "Tower" #読み取り用属性
31
32     def update(self, screen: pg.Surface):
33         if self.hp < 0:
34             self.kill()
35     class Chicken(pg.sprite.Sprite):
36         """
37         鶏肉に関するクラス
38         """
39     def __init__(self, side):
40         super().__init__()
41         self.image = pg.transform.rotozoom(pg.image.load(f"ex05/fig/chicken.png"), 0, 0.3)
42         self.rect = self.image.get_rect()
43         self.life = 50
44         if side == 0:
```

```

45         self.rect.center = random.randint(WIDTH/2, WIDTH), random.randint(HEIGHT/2,HEIGHT)
46     elif side == 1:
47         self.rect.center = random.randint(0, WIDTH/2), random.randint(HEIGHT/2,HEIGHT)
48     else:
49         self.rect.center = random.randint(0, WIDTH), random.randint(HEIGHT/2,HEIGHT)
50     def update(self):
51         self.life -=1
52         if self.life < 0:
53             self.kill()
54
55
56 ✓ class Chara(pg.sprite.Sprite):
57     """
58     出撃するこうかとんに関するクラス
59     1,init
60     引数はHPと位置を示すタプルとdx,反転させるか(敵であるか)を判定するa
61     こうかとんの画像を表示させ指定された位置に置く
62     dxはこれの重さや強さを決める数値である
63     大きいほど重く防御の堅いキャラクターになる
64     読み取り用属性は"Chara"である
65     2,update
66     dxの量分移動する
67     HPが0になるとグループから消える
68     """
69 ✓ def __init__(self, hp, xy: tuple[int, int], dx, a = False):
70     super().__init__()
71     self.a = a
72     self.image = pg.transform.flip(pg.transform.rotozoom(pg.image.load("ex05/fig/2.png"), abs(dx)*0.1, a
73     self.hp = hp
74     self.rect = self.image.get_rect()
75     self.rect.center = xy #位置X,Y
76     self.dx = 20/dx #dxが小さいほど速い
77     self.weight = abs(dx) #dxが大きいほど重い
78     self.zokusei = "Chara" #読み取り用属性
79
80 ✓ def update(self,screen: pg.Surface):
81     self.rect.move_ip(self.dx, 0)
82     screen.blit(self.image, self.rect)
83     if self.hp < 0:
84         self.kill()
85
86 ✓ class Hit(pg.sprite.Sprite):
87     """
88     当たり判定に関するクラス
89     1,init
90     引数はものとノックバック発生時間
91     ヒットが発生したもののHPを5減らす
92     2,update
93     読み取り属性が"Chara"なら
94     繰り返し当たらないよう、重さに応じてノックバックする
95     ノックバック発生時間が終わるとグループから消える
96     """
97 ✓ def __init__(self, obj: "Chara|tower", life: int):
98     super().__init__()
99     self.obj = obj
100     self.obj.hp -= 5
101     self.life = life
102 ✓ def update(self):
103     self.life -= 1
104     if self.obj.zokusei == "Chara":

```

```
105         if self.obj.dx >= 0:
106             self.obj.rect.centerx -= 5/self.obj.weight #dxが大きいほどノックバックしにくい
107         else:
108             self.obj.rect.centerx += 5/self.obj.weight #dxが大きいほどノックバックしにくい
109     if self.life < 0:
110         self.kill()
111     if self.obj.hp <= 0:
112         explosion_sound()
113
114     def BGM():
115         pygame.mixer.init() #初期化
116
117         pygame.mixer.music.load("ex05/fig/Tropical.mp3") #読み込み
118
119         pygame.mixer.music.play(-1) #再生
120
121     def duck_sound(): # こうかとんに声を追加した
122         pygame.mixer.init() #初期化
123
124         dse = pygame.mixer.Sound("ex05/fig/duckvoice.mp3") #読み込み
125
126         dse.play() #再生
127
128
129     def explosion_sound():
130
131         ese = pygame.mixer.Sound("ex05/fig/explosion_sound.mp3") #読み込み
132
133         ese.play() #再生
134
135
136     class Cooldown():
137         """出撃タイマーの設定"""
138         def __init__(self, cooltime):
139             self.cooltime = cooltime
140             self.timer = 0
141             self.color = [240, 128, 128], [50, 50, 50]
142
143
144     def flag(self, now):
145         if (now - self.timer >= self.cooltime) or self.timer == 0:
146             self.timer = now
147             return True
148         else:
149             return False
150
151     def update(self, now, surface, n):
152         font1 = pg.font.SysFont("hg正楷書体pro", 150)
153         n += 2
154         text1 = font1.render(f"{n-2}", True, (255, 255, 255))
155         if now - self.timer >= self.cooltime or self.timer == 0:
156             surface.fill(self.color[0], (HEIGHT / 8 * n, 700, 100, 100))
157         else:
158             surface.fill(self.color[1], (HEIGHT / 8 * n, 700, 100, 100))
159         surface.blit(text1, (HEIGHT / 8 * n, 676))
160
161
162     def main():
163         screen = pg.display.set_mode((WIDTH, HEIGHT))
164         bg_img = pg.image.load("ex05/fig/pg_bg.jpg")
```

```
165     Pltower = pg.sprite.Group()
166     Entower = pg.sprite.Group()
167     cooltimes = [Cooldown(10), Cooldown(40), Cooldown(200)]
168     Plchara = pg.sprite.Group()
169     Enchara = pg.sprite.Group()
170
171     hits = pg.sprite.Group()
172     chickens = pg.sprite.Group()
173     tmr = 0
174     clock = pg.time.Clock()
175
176     Pltower.add(tower(500, (100, 400)))
177     Entower.add(tower(500, (1500, 400)))
178
179     while True:
180
181         key_lst = pg.key.get_pressed()
182         """
183         敵は一定時間でキャラクターが生まれる
184         """
185         if tmr != 0 and tmr % 200 == 0:
186             if tmr != 0 and tmr % 400 == 0:
187                 if tmr != 0 and tmr % 800 == 0:
188                     Enchara.add(Chara(75, (1500, 400), -15, True))
189                 else:
190                     Enchara.add(Chara(50, (1500, 400), -10, True))
191             else:
192                 Enchara.add(Chara(50, (1500, 400), -5, True))
193
194         for event in pg.event.get():
195             if event.type == pg.QUIT:
196                 return 0
197             """
198             押したボタンの数値が大きいくほど
199             強いけど遅いキャラクターが生まれる
200             """
201
202
203             if event.type == pg.KEYDOWN and event.key == pg.K_0 and cooltimes[0].flag(tmr):
204                 Plchara.add(Chara(50, (100, 400), 5))
205                 duck_sound()
206             if event.type == pg.KEYDOWN and event.key == pg.K_1 and cooltimes[1].flag(tmr):
207                 Plchara.add(Chara(75, (100, 400), 10))
208                 duck_sound()
209             if event.type == pg.KEYDOWN and event.key == pg.K_2 and cooltimes[2].flag(tmr):
210                 Plchara.add(Chara(100, (100, 400), 15))
211                 duck_sound()
212
213         for plt in pg.sprite.groupcollide(Pltower, Enchara, False, False).keys():
214             hits.add(Hit(plt, 20)) #敵に襲われて自分のタワーにダメージ
215
216         for ply in pg.sprite.groupcollide(Plchara, Enchara, False, False).keys():
217             hits.add(Hit(ply, 20)) #敵のキャラクターと戦って自分のキャラにダメージ
218
219         for ply in pg.sprite.groupcollide(Plchara, Entower, False, False).keys():
220             hits.add(Hit(ply, 20)) #敵のタワーの反撃で自分のキャラにダメージ
221             chickens.add(Chicken(0))
222         for ent in pg.sprite.groupcollide(Entower, Plchara, False, False).keys():
223             hits.add(Hit(ent, 20)) #自分のキャラが襲撃して敵のタワーにダメージ
224
225     ~~~
```

```
225
226     for enm in pg.sprite.groupcollide(Enchara, Plchara, False, False).keys():
227         hits.add(Hit(enm, 20)) #自分のキャラクターと戦って敵のキャラにダメージ
228
229
230     for enm in pg.sprite.groupcollide(Enchara, Pltower, False, False).keys():
231         hits.add(Hit(enm, 20)) #自分のタワーの反撃で敵のキャラにダメージ
232         chickens.add(Chicken(1))
233     if len(Pltower) == 0: #自分のタワーがやられたとき、少し止まって終了
234         explosion_sound()
235
236     font1 = pygame.font.SysFont("hg正楷書体pro", 400) # 敗北ロゴ生成
237     font2 = pygame.font.SysFont(None, 300)
238
239     text1 = font1.render("敗北", True, (255,0,0))
240     text2 = font2.render("LOSE", True, (255,0,0))
241     screen.blit(text1, (WIDTH/2-400,HEIGHT/2-400))
242     screen.blit(text2, (WIDTH/2-300,HEIGHT/2+100))
243
244     pygame.display.update() #描画処理を実行
245     pg.display.update()
246     pygame.display.update() #描画処理を実行
247
248     time.sleep(2)
249     return
250
251     if len(Entower) == 0: #敵のタワーがやられたとき、少し止まって終了
252         explosion_sound()
253
254     font1 = pygame.font.SysFont("hg正楷書体pro", 400) # 勝利ロゴ生成
255     font2 = pygame.font.SysFont(None, 300)
256
257     text1 = font1.render("勝利", True, (255,255,0))
258     text2 = font2.render("WIN", True, (255,255,0))
259     screen.blit(text1, (WIDTH/2-400,HEIGHT/2-400))
260     screen.blit(text2, (WIDTH/2-200,HEIGHT/2+100))
261
262     pygame.display.update() #描画処理を実行
263     pg.display.update()
264
265     time.sleep(2)
266     return
267
268
269     screen.blit(bg_img, [0, 0])
270
271     Pltower.update(screen)
272     Pltower.draw(screen)
273     for i in range(len(cooltimes)):
274         cooltimes[i].update(tmr, screen, i)
275     Entower.update(screen)
276     Entower.draw(screen)
277
278     Plchara.update(screen)
279     Plchara.draw(screen)
280     Enchara.update(screen)
281     Enchara.draw(screen)
282
283     chickens.update()
284     chickens.draw(screen)
285
286     # 画面の更新
```

```
285
286         hits.update()
287
288         pg.display.update()
289         tmr += 1
290         clock.tick(50)
291
292     if __name__ == "__main__":
293         BGM()
294         pg.init()
295         main()
296         pg.quit()
297         sys.exit()
298
```