

☰  c0a241414f / ProjExD\_3

🔍  Issues 1  Pull requests  Actions  Projects 

💻 main ➔ ProjExD\_3 / ProjExD / ex3 / fight\_kokaton.py ⚙️ 

 c0a241414f 追加機能2：連射 5482c12 · 13 minutes ago 

188 lines (152 loc) · 5.54 KB

Code Blame   Raw    

```
1 import os
2 import random
3 import sys
4 import time
5 import pygame as pg
6
7
8 WIDTH = 1100
9 HEIGHT = 650
10 NUM_OF_BOMBS = 5
11 os.chdir(os.path.dirname(os.path.abspath(__file__)))
12
13
14 def check_bound(obj_rct: pg.Rect) -> tuple[bool, bool]:
15     yoko, tate = True, True
16     if obj_rct.left < 0 or WIDTH < obj_rct.right:
17         yoko = False
18     if obj_rct.top < 0 or HEIGHT < obj_rct.bottom:
19         tate = False
20     return yoko, tate
21
22
23 class Bird:
24     delta = {
25         pg.K_w: (0, -5),
26         pg.K_s: (0, +5),
27         pg.K_a: (-5, 0),
28         pg.K_d: (+5, 0),
29     }
30     img0 = pg.transform.rotozoom(pg.image.load("fig/3.png"), 0, 0.9)
31     img = pg.transform.flip(img0, True, False)
32     imgs = {
33         (+5, 0): img,
34         (+5, -5): pg.transform.rotozoom(img, 45, 0.9),
35         (0, -5): pg.transform.rotozoom(img, 90, 0.9),
36         (-5, -5): pg.transform.rotozoom(img0, -45, 0.9),
37     }
```

```
37         (-5, 0): img0,
38         (-5, +5): pg.transform.rotozoom(img0, 45, 0.9),
39         (0, +5): pg.transform.rotozoom(img, -90, 0.9),
40         (+5, +5): pg.transform.rotozoom(img, -45, 0.9),
41     }
42
43     def __init__(self, xy: tuple[int, int]):
44         self.img = __class__.imgs[(+5, 0)]
45         self.rct = self.img.get_rect()
46         self.rct.center = xy
47
48     def change_img(self, num: int, screen: pg.Surface):
49         self.img = pg.transform.rotozoom(pg.image.load(f"fig/{num}.png"), 0, 0.9)
50         screen.blit(self.img, self.rct)
51
52     def update(self, key_lst: list[bool], screen: pg.Surface):
53         sum_mv = [0, 0]
54         for k, mv in __class__.delta.items():
55             if key_lst[k]:
56                 sum_mv[0] += mv[0]
57                 sum_mv[1] += mv[1]
58         self.rct.move_ip(sum_mv)
59         if check_bound(self.rct) != (True, True):
60             self.rct.move_ip(-sum_mv[0], -sum_mv[1])
61         if not (sum_mv[0] == 0 and sum_mv[1] == 0):
62             self.img = __class__.imgs[tuple(sum_mv)]
63         screen.blit(self.img, self.rct)
64
65
66     class Beam:
67         def __init__(self, bird: "Bird"):
68             self.img = pg.image.load("fig/beam.png")
69             self.rct = self.img.get_rect()
70             self.rct.centery = bird.rct.centery
71             self.rct.left = bird.rct.right
72             self.vx, self.vy = +5, 0 # 速めてもOK
73
74         def update(self, screen: pg.Surface):
75             self.rct.move_ip(self.vx, self.vy)
76             screen.blit(self.img, self.rct)
77
78         def is_out(self) -> bool:
79             """画面外判定"""
80             yoko, tate = check_bound(self.rct)
81             return not (yoko and tate)
82
83
84     class Bomb:
85         def __init__(self, color: tuple[int, int, int], rad: int):
86             self.img = pg.Surface((2*rad, 2*rad))
87             pg.draw.circle(self.img, color, (rad, rad), rad)
88             self.img.set_colorkey((0, 0, 0))
```

```
89         self.rct = self.img.get_rect()
90         self.rct.center = random.randint(0, WIDTH), random.randint(0, HEIGHT)
91         self.vx, self.vy = +5, +5
92
93     def update(self, screen: pg.Surface):
94         yoko, tate = check_bound(self.rct)
95         if not yoko:
96             self.vx *= -1
97         if not tate:
98             self.vy *= -1
99         self.rct.move_ip(self.vx, self.vy)
100        screen.blit(self.img, self.rct)
101
102
103    class Score:
104        def __init__(self):
105            self.fonto = pg.font.SysFont("hgp創英角ﾎﾟｯﾌﾟ体", 30)
106            self.color = (0, 0, 255)
107            self.scr = 0
108            self.img = self.fonto.render("表示させる文字 ", 0, self.color)
109            self.rct = self.img.get_rect()
110            self.rct.center = 100, HEIGHT-50
111
112        def update(self, screen: pg.Surface):
113            self.img = self.fonto.render(f"スコア:{self.scr}", 0, self.color)
114            screen.blit(self.img, self.rct)
115
116
117    def main():
118        pg.display.set_caption("たたかえ！こうかとん")
119        screen = pg.display.set_mode((WIDTH, HEIGHT))
120        bg_img = pg.image.load("fig/pg_bg.jpg")
121
122        bird = Bird((300, 200))
123        bombs = [Bomb((255, 0, 0), 10) for _ in range(NUM_OF_BOMBS)]
124        beams = [] # ← Beam リスト
125
126        clock = pg.time.Clock()
127        score = Score()
128        tmr = 0
129
130        while True:
131            for event in pg.event.get():
132                if event.type == pg.QUIT:
133                    return
134                if event.type == pg.KEYDOWN and event.key == pg.K_RSHIFT:
135                    beams.append(Beam(bird)) # ← 追加
136
137            screen.blit(bg_img, [0, 0])
138
139            # こうかとん死亡判定
140            for bomb in bombs:
```

```
141     if bird.rct.colliderect(bomb.rct):
142         bird.change_img(8, screen)
143         fonto = pg.font.Font(None, 80)
144         txt = fonto.render("Game Over", True, (255, 0, 0))
145         screen.blit(txt, [WIDTH//2-150, HEIGHT//2])
146
147         pg.display.update()
148         time.sleep(1)
149         return
150
151     # ビームと爆弾の衝突判定
152     for i, beam in enumerate(beams):
153         for j, bomb in enumerate(bombs):
154             if beam is not None and bomb is not None:
155                 if beam.rct.colliderect(bomb.rct):
156                     beams[i] = None
157                     bombs[j] = None
158                     score.scr += 1
159
160     # None を削除
161     beams = [b for b in beams if b is not None]
162     bombs = [b for b in bombs if b is not None]
163
164     # ビームが画面外に出たらリストから削除
165     beams = [b for b in beams if not b.is_out()]
166
167     key_lst = pg.key.get_pressed()
168     bird.update(key_lst, screen)
169
170     # ビーム描画
171     for beam in beams:
172         beam.update(screen)
173
174     # 爆弾描画
175     for bomb in bombs:
176         bomb.update(screen)
177
178     score.update(screen)
179     pg.display.update()
180     tmr += 1
181     clock.tick(50)
182
183
184 if __name__ == "__main__":
185     pg.init()
186     main()
187     pg.quit()
188     sys.exit()
```