


 c0b22061 / ProjExD\_03



[Code](#) [Issues](#) 1 [Pull requests](#) [Actions](#) [Projects](#) [...](#)

 master [ProjExD\\_03 / fight\\_kokaton.py](#) 

Go to file  [t](#) [...](#)

 c0b22061 未完成

1 hour ago  

210 lines (179 loc) · 6.62 KB

 master [ProjExD\\_03 / fight\\_kokaton.py](#) [↑ Top](#)

Code

Blame

Raw



```
1  import random
2  import sys
3  import time

9  HEIGHT = 900 # ゲームウィンドウの高さ
10 NUM_OF_BOMBS = 5 # 爆弾の数
11
12
13  def check_bound(obj_rct: pg.Rect) -> tuple[bool, bool]:
14      """
15      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
16      引数: こうかとん, または、爆弾SurfaceのRect
17      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)
18      """
19      yoko, tate = True, True
20      if obj_rct.left < 0 or WIDTH < obj_rct.right:
21          yoko = False
22      if obj_rct.top < 0 or HEIGHT < obj_rct.bottom:
23          tate = False
24      return yoko, tate
25
26
27  class Bird:
28      """
29      ゲームキャラクター (こうかとん) に関するクラス
30      """
31      delta = { # 押下キーと移動量の辞書
32          pg.K_UP: (0, -5),
33          pg.K_DOWN: (0, +5),
34          pg.K_LEFT: (-5, 0),
35          pg.K_RIGHT: (+5, 0),
36      }
37
38  def __init__(self, num: int, xy: tuple[int, int]):
39      """
40      こうかとん画像Surfaceを生成する
41      引数1 num: こうかとん画像ファイル名の番号
42      引数2 xy: こうかとん画像の位置座標タプル
43      """
44      self.img = pg.transform.flip( # 左右反転
45          pg.transform.rotozoom( # 2倍に拡大
46              pg.image.load(f"ex03/fig/{num}.png"),
47              0,
48              2, 0)
```

```

49         True,
50         False
51     )
52     self.rct = self.img.get_rect()
53     self.rct.center = xy
54
55     def change_img(self, num: int, screen: pg.Surface):
56         """
57         こうかどん画像を切り替え、画面に転送する
58         引数1 num: こうかどん画像ファイル名の番号
59         引数2 screen: 画面Surface
60         """
61         self.img = pg.transform.rotozoom(pg.image.load(f"ex03/fig/{num}.png"), 0, 2.0)
62         screen.blit(self.img, self.rct)
63
64     def update(self, key_lst: list[bool], screen: pg.Surface):
65         """
66         押下キーに応じてこうかどんを移動させる
67         引数1 key_lst: 押下キーの真理値リスト
68         引数2 screen: 画面Surface
69         """
70         sum_mv = [0, 0]
71         for k, mv in __class__.delta.items():
72             if key_lst[k]:
73                 sum_mv[0] += mv[0]
74                 sum_mv[1] += mv[1]
75         self.rct.move_ip(sum_mv)
76         if check_bound(self.rct) != (True, True):
77             self.rct.move_ip(-sum_mv[0], -sum_mv[1])
78         screen.blit(self.img, self.rct)
79
80
81     class Beam:
82     def __init__(self, bird: Bird):
83         """
84         ビーム画像Surfaceを生成する
85         引数 bird: こうかどんインスタンス (Birdクラスのインスタンス)
86         """
87         self.img = pg.image.load(f"ex03/fig/beam.png")
88         self.rct = self.img.get_rect()
89         self.rct.left = bird.rct.right # こうかどんの右横座標
90         self.rct.centery = bird.rct.centery # こうかどんの中心縦座標
91         self.vx, self.vy = +5, 0
92
93     def update(self, screen: pg.Surface):
94         """
95         ビームを速度vxにしたがって移動させる
96         引数 screen: 画面Surface
97         """
98         self.rct.move_ip(self.vx, self.vy)
99         screen.blit(self.img, self.rct)
100
101
102
103     class Score:
104     def __init__(self):
105         self.font = pg.font.SysFont("hgp", 30)
106         co = (0, 0, 255)
107         score = 0
108         self.img = self.font.render("表示させる文字列", 0, co)
109         self.rct = pg.display.get_rect()
110         self.rct.center = (100, 50)
111         screen=

```

```
112
113     def update(self):
114
115         screen.blit(kazu,[100,50])
116
117
118     class Bomb:
119         """
120         爆弾に関するクラス
121         """
122         colors = [(255, 0, 0), (0, 255, 0), (0, 0, 255), (255, 255, 0), (255, 0, 255), (0, 255, 255)]
123         directions = [-5, +5]
124
125     def __init__(self):
126         """
127         ランダムな色, サイズの爆弾円Surfaceを生成する
128         """
129         rad = random.randint(10, 50) # 半径をランダムに設定
130         color = random.choice(__class__.colors)
131         self.img = pg.Surface((2*rad, 2*rad))
132         pg.draw.circle(self.img, color, (rad, rad), rad)
133         self.img.set_colorkey((0, 0, 0))
134         self.rct = self.img.get_rect()
135         self.rct.center = random.randint(0, WIDTH), random.randint(0, HEIGHT)
136         self.vx = random.choice(__class__.directions)
137         self.vy = random.choice(__class__.directions)
138
139     def update(self, screen: pg.Surface):
140         """
141         爆弾を速度ベクトルself.vx, self.vyに基づき移動させる
142         引数 screen: 画面Surface
143         """
144         yoko, tate = check_bound(self.rct)
145         if not yoko:
146             self.vx *= -1
147         if not tate:
148             self.vy *= -1
149         self.rct.move_ip(self.vx, self.vy)
150         screen.blit(self.img, self.rct)
151
152
153     def main():
154         score=0
155         pg.display.set_caption("たたかえ！こうかとん")
156         screen = pg.display.set_mode((WIDTH, HEIGHT))
157         bg_img = pg.image.load("ex03/fig/pg_bg.jpg")
158         bird = Bird(3, (900, 400))
159         Score()
160         bombs = [Bomb() for _ in range(NUM_OF_BOMBS)]
161         beam = None
162
163         clock = pg.time.Clock()
164         tmr = 0
165         while True:
166             for event in pg.event.get():
167                 if event.type == pg.QUIT:
168                     return
169                 if event.type == pg.KEYDOWN and event.key == pg.K_SPACE:
170                     # キーが押されたら, かつ, キーの種類がスペースキーだったら
171                     beam = Beam(bird)
172
173             screen.blit(bg_img, [0, 0])
174
175
```

```
176         for bomb in bombs:
177             if bird.rct.colliderect(bomb.rct):
178                 # ゲームオーバー時に、こうかとん画像を切り替え、1秒間表示させる
179                 bird.change_img(8, screen)
180                 pg.display.update()
181                 time.sleep(1)
182                 return
183         for i, bomb in enumerate(bombs):
184             if beam is not None:
185                 if beam.rct.colliderect(bomb.rct): # ビームと爆弾の衝突判定
186                     # 撃墜=Noneにする
187                     beam = None
188                     bombs[i] = None
189                     bird.change_img(6, screen)
190                     pg.display.update()
191                     score +=1
192                     Score.update(screen)
193         bombs = [bomb for bomb in bombs if bomb is not None]
194
195         key_lst = pg.key.get_pressed()
196         bird.update(key_lst, screen)
197         for bomb in bombs:
198             bomb.update(screen)
199         if beam is not None:
200             beam.update(screen)
201         pg.display.update()
202         tmr += 1
203         clock.tick(50)
204
205
206 if __name__ == "__main__":
207     pg.init()
208     main()
209     pg.quit()
210     sys.exit()
```