


 TOKYOLIUNIAN /  
ProjExD\_2



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

  main [ProjExD\\_2 / README.md](#) 

 [t](#) [...](#)





TOKYOLIUNIAN `dodge_bomb.py`

5f05273 · 1 hour ago



89 lines (73 loc) · 2.33 KB

[Preview](#) [Code](#) [Blame](#)

 [Raw](#)     

```
1  import os
2  import random
3  import sys
4  import pygame as pg
5
6
7  WIDTH, HEIGHT = 1100, 650
8  DELTA = {
9      pg.K_UP: (0, -5),
10     pg.K_DOWN: (0, +5),
11     pg.K_LEFT: (-5, 0),
12     pg.K_RIGHT: (+5, 0),
13 }
14 os.chdir(os.path.dirname(os.path.abspath(__file__)))
15
16
17 def check_bound(rct: pg.Rect) -> tuple[bool, bool]:
18
19     yoko, tate = True, True # 横, 縦方向用の変数
20     # 横方向判定
21     if rct.left < 0 or WIDTH < rct.right: # 画面外だったら
22         yoko = False
23     # 縦方向判定
24     if rct.top < 0 or HEIGHT < rct.bottom: # 画面外だったら
25         tate = False
26     return yoko, tate
27
28
29
30
31 def main():
32     pg.display.set_caption("逃げろ! こうかといん")
33     screen = pg.display.set_mode((WIDTH, HEIGHT))
34     # こうかといん初期化
35     bg_img = pg.image.load("fig/pg_bg.jpg")
36     kk_img = pg.transform.rotozoom(pg.image.load("fig/3.png"), 0, 0.9)
37     kk_rct = kk_img.get_rect()
38     kk_rct.center = 300, 200
39     # 爆弾初期化
40     bb_img = pg.Surface((20, 20))
41     pg.draw.circle(bb_img, (255, 0, 0), (10, 10), 10)
42     bb_rct = bb_img.get_rect()
```

```
43     bb_rct.centerx = random.randint(0, WIDTH)
44     bb_rct.centery = random.randint(0, HEIGHT)
45     bb_img.set_colorkey((0, 0, 0))
46     vx, vy = +5, +5
47
48     clock = pg.time.Clock()
49     tmr = 0
50     while True:
51         for event in pg.event.get():
52             if event.type == pg.QUIT:
53                 return
54             screen.blit(bg_img, [0, 0])
55
56             # こうかとん
57             if kk_rct.colliderect(bb_rct):
58                 print("Game Over")
59                 return
60
61             key_lst = pg.key.get_pressed()
62             sum_mv = [0, 0]
63
64             for key, mv in DELTA.items():
65                 if key_lst[key]:
66                     sum_mv[0] += mv[0] # 左右方向
67                     sum_mv[1] += mv[1] # 上下方向
68
69             kk_rct.move_ip(sum_mv)
70             if check_bound(kk_rct) != (True, True): # 画面外だったら
71                 kk_rct.move_ip(-sum_mv[0], -sum_mv[1]) # 画面内に戻す
72             screen.blit(kk_img, kk_rct)
73             bb_rct.move_ip(vx, vy) # 爆弾の移動
74             yoko, tate = check_bound(bb_rct)
75             if not yoko: # 左右
76                 vx *= -1
77             if not tate: # 上下
78                 vy *= -1
79             screen.blit(bb_img, bb_rct) # 爆弾の描画
80             pg.display.update()
81             tmr += 1
82             clock.tick(50)
83
84
85 if __name__ == "__main__":
86     pg.init()
87     main()
88     pg.quit()
89     sys.exit()
```