

MomonoKawabata / ProjExD

Public

<> Code

Issues 8

Pull requests

Discussions

Actions

Projects

Wiki

Security

Insights

Settings

main

ProjExD / ex04 / dodge_bomb.py / <> Jump to

MomonoKawabata 訂正 #8 #7

History

1 contributor

130 lines (107 sloc) | 4.07 KB

```
1 import pygame as pg
2 import random
3 import sys
4 import time
5
6 def check_bound(obj_rct, scr_rct):
7     yoko, tate = +1, +1
8     if obj_rct.left < scr_rct.left or scr_rct.right < obj_rct.right:
9         yoko = -1
10    if obj_rct.top < scr_rct.top or scr_rct.bottom < obj_rct.bottom:
11        tate = -1
12    return yoko, tate
13
14
15 def main():
16     clock =pg.time.Clock()
17     # 練習 1
18     pg.display.set_caption("逃げる！ こうかとん")
19     scrn_sfc = pg.display.set_mode((1600, 900))
20     scrn_rct = scrn_sfc.get_rect()
21     pgbg_sfc = pg.image.load("fig/pg_bg.jpg")
22     pgbg_rct = pgbg_sfc.get_rect()
23
24     # 練習 3
25     tori_sfc = pg.image.load("fig/6.png")
26     tori_sfc = pg.transform.rotozoom(tori_sfc, 0, 2.0)
27     tori_rct = tori_sfc.get_rect()
28     tori_rct.center = 900, 400
29     # scrn_sfcにtori_rctに従って, tori_sfcを貼り付ける
30     scrn_sfc.blit(tori_sfc, tori_rct)
31
32
33
34     # 練習 5
35     bomb_sfc = pg.Surface((20, 20)) # 正方形の空のSurface
36     bomb_sfc.set_colorkey((0, 0, 0))
37     pg.draw.circle(bomb_sfc, (255, 0, 0), (10, 10), 10)
38     bomb_rct = bomb_sfc.get_rect()
39     bomb_rct.centerx = random.randint(0, scrn_rct.width)
40     bomb_rct.centery = random.randint(0, scrn_rct.height)
41     scrn_sfc.blit(bomb_sfc, bomb_rct)
42     vx, vy = +1, +1
43
44     bomb_sfc2 = pg.Surface((20, 20)) # 2つめの爆弾
45     bomb_sfc2.set_colorkey((0, 0, 0))
46     pg.draw.circle(bomb_sfc2, (0, 0, 255), (0, 255), (10, 10), 10)
47     bomb_rct2 = bomb_sfc2.get_rect()
48     bomb_rct2.centerx = random.randint(0, scrn_rct.width)
49     bomb_rct2.centery = random.randint(0, scrn_rct.height)
50     scrn_sfc.blit(bomb_sfc2, bomb_rct2)
51     ax, ay = -5 , -5
52
53
54
```

```

55 # 練習2
56 while True:
57     scrn_sfc.blit(pgbg_sfc, pgbg_rct)
58
59
60     for event in pg.event.get():
61         if event.type == pg.QUIT:
62             return
63
64 # 練習4
65 key_dct = pg.key.get_pressed()
66 if key_dct[pg.K_UP]:
67     if tori_rct.collidect(bomb_rct) or tori_rct.collidect(bomb_rct2): #爆弾に当たったら
68         tori_rct.centery -= 0 #こうかとんが動かない
69     else:
70         tori_rct.centery -= 1
71 if key_dct[pg.K_DOWN]:
72     if tori_rct.collidect(bomb_rct) or tori_rct.collidect(bomb_rct2): #爆弾に当たったら
73         tori_rct.centery += 0 #こうかとんが動かない
74     else:
75         tori_rct.centery += 1
76 if key_dct[pg.K_LEFT]:
77     if tori_rct.collidect(bomb_rct) or tori_rct.collidect(bomb_rct2): #爆弾に当たったら
78         tori_rct.centerx -= 0 #こうかとんが動かない
79     else:
80         tori_rct.centerx -= 1
81 if key_dct[pg.K_RIGHT]:
82     if tori_rct.collidect(bomb_rct) or tori_rct.collidect(bomb_rct2): #爆弾に当たったら
83         tori_rct.centerx += 0 #こうかとんが動かない
84     else:
85         tori_rct.centerx += 1
86 if check_bound(tori_rct, scrn_rct) != (+1, +1):
87     # どこかしらはみ出ていたら
88     if key_dct[pg.K_UP]:
89         tori_rct.centery += 1
90     if key_dct[pg.K_DOWN]:
91         tori_rct.centery -= 1
92     if key_dct[pg.K_LEFT]:
93         tori_rct.centerx += 1
94     if key_dct[pg.K_RIGHT]:
95         tori_rct.centerx -= 1
96 scrn_sfc.blit(tori_sfc, tori_rct)
97
98 # 練習6
99 bomb_rct.move_ip(vx, vy)
100 scrn_sfc.blit(bomb_sfc, bomb_rct)
101 yoko, tate = check_bound(bomb_rct, scrn_rct)
102 vx *= yoko
103 vy *= tate
104
105 bomb_rct2.move_ip(ax, ay) #2つ目の爆弾の動き
106 scrn_sfc.blit(bomb_sfc2, bomb_rct2)
107 yoko, tate = check_bound(bomb_rct2, scrn_rct)
108 ax *= yoko
109 ay *= tate
110
111
112 # 練習8
113 if tori_rct.collidect(bomb_rct) :
114     vx *= 0 #赤い爆弾の動きとめる
115     vy *= 0
116 if tori_rct.collidect(bomb_rct2) :
117     ax *= 0 #青い爆弾の動きとめる
118     ay *= 0
119
120
121
122 pg.display.update()
123 clock.tick(1000)
124
125
126 if __name__ == "__main__":
127     pg.init()
128     main()

```

```
129 pg.quit()  
130 sys.exit()
```