


 c0b220698f /
ProjExD_05



[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

 C0B22069/score ▾ ProjExD_05 / dinosaur.py 

Go to file

t

...



c0b220698f スコア実装完了

41 minutes ago



161 lines (131 loc) · 4.27 KB

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

Raw     

```
1  import sys
2  import os
3  import random
4
5  import pygame as pg
6
7
8  SCREEN_HEIGHT = 600
9  SCREEN_WIDTH = 1100
10 SCREEN = pg.display.set_mode((SCREEN_WIDTH, SCREEN_HEIGHT))
11
12 RUNNING = [
13     pg.image.load(os.path.join("ex05/Assets/Dino", "DinoRun1.png")),
14     pg.image.load(os.path.join("ex05/Assets/Dino", "DinoRun2.png")),
15 ]
16 JUMPING = pg.image.load(os.path.join("ex05/Assets/Dino", "DinoJump.png"))
17 DUCKING = [
18     pg.image.load(os.path.join("ex05/Assets/Dino", "DinoDuck1.png")),
19     pg.image.load(os.path.join("ex05/Assets/Dino", "DinoDuck2.png")),
20 ]
21
22
23 BG = pg.image.load(os.path.join("ex05/Assets/Other", "Track.png"))
24
25
26 class Dinosaur:
27     X_POS = 80
28     Y_POS = 310
29     Y_POS_DUCK = 340
30     JUMP_VEL = 8.5
31
32     def __init__(self):
33         self.duck_img = DUCKING
34         self.run_img = RUNNING
35         self.jump_img = JUMPING
36
37         self.dino_duck = False
38         self.dino_run = True
39         self.dino_jump = False
40
41         self.step_index = 0
42         self.jump_vel = self.JUMP_VEL
43         self.image = self.run_img[0]
44         self.dino_rect = self.image.get_rect()
45         self.dino_rect.x = self.X_POS
46         self.dino_rect.y = self.Y_POS
47
48     def update(self, userInput):
```

```

49         if self.dino_duck:
50             self.duck()
51         if self.dino_run:
52             self.run()
53         if self.dino_jump:
54             self.jump()
55
56         if self.step_index >= 10:
57             self.step_index = 0
58
59         if userInput[pg.K_UP] and not self.dino_jump:
60             self.dino_duck = False
61             self.dino_run = False
62             self.dino_jump = True
63         elif userInput[pg.K_DOWN] and not self.dino_jump:
64             self.dino_duck = True
65             self.dino_run = False
66             self.dino_jump = False
67         elif not (self.dino_jump or userInput[pg.K_DOWN]):
68             self.dino_duck = False
69             self.dino_run = True
70             self.dino_jump = False
71
72     def duck(self):
73         self.image = self.duck_img[self.step_index // 5]
74         self.dino_rect = self.image.get_rect()
75         self.dino_rect.x = self.X_POS
76         self.dino_rect.y = self.Y_POS_DUCK
77         self.step_index += 1
78
79     def run(self):
80         self.image = self.run_img[self.step_index // 5]
81         self.dino_rect = self.image.get_rect()
82         self.dino_rect.x = self.X_POS
83         self.dino_rect.y = self.Y_POS
84         self.step_index += 1
85
86     def jump(self):
87         self.image = self.jump_img
88         if self.dino_jump:
89             self.dino_rect.y -= self.jump_vel * 4
90             self.jump_vel -= 0.8
91         if self.jump_vel < -self.JUMP_VEL:
92             self.dino_jump = False
93             self.jump_vel = self.JUMP_VEL
94
95     def draw(self, SCREEN):
96         SCREEN.blit(self.image, (self.dino_rect.x, self.dino_rect.y))
97
98
99     def main():
100         global game_speed, x_pos_bg, y_pos_bg, points, obstacles
101         run = True
102         clock = pg.time.Clock()
103         player = Dinosaur()
104         game_speed = 20
105         x_pos_bg = 0
106         y_pos_bg = 380
107         points = 0
108         font = pg.font.Font("freesansbold.ttf", 20)
109         obstacles = []
110         pg.display.set_caption("恐竜ゲーム")
111
112     def score():
113         global points, game_speed

```

```
113     global points, game_speed
114     points += 0.1
115     if points % 100 == 0:
116         game_speed += 1
117
118     text = font.render(f"ScorePoint:{points:.0f}", True, (0, 0, 0))
119     textRect = text.get_rect()
120     textRect.center = (1000, 40)
121     SCREEN.blit(text, textRect)
122
123     def background():
124         global x_pos_bg, y_pos_bg
125         image_width = BG.get_width()
126         SCREEN.blit(BG, (x_pos_bg, y_pos_bg))
127         SCREEN.blit(BG, (image_width + x_pos_bg, y_pos_bg))
128         if x_pos_bg <= -image_width:
129             SCREEN.blit(BG, (image_width + x_pos_bg, y_pos_bg))
130             x_pos_bg = 0
131         x_pos_bg -= game_speed
132
133     while run:
134         for event in pg.event.get():
135             if event.type == pg.QUIT:
136                 run = False
137                 sys.exit()
138
139         SCREEN.fill((255, 255, 255))
140         userInput = pg.key.get_pressed()
141
142         player.draw(SCREEN)
143         player.update(userInput)
144
145         for obstacle in obstacles:
146             obstacle.draw(SCREEN)
147             obstacle.update()
148             if player.dino_rect.colliderect(obstacle.rect):
149                 pg.time.delay(2000)
150
151         background()
152         score()
153         clock.tick(30)
154         pg.display.update()
155
156
157 if __name__ == "__main__":
158     pg.init()
159     main()
160     pg.quit()
161     sys.exit()
```