 c0a21023 / ProjExD

Public

<> Code

Issues 19

Pull requests

Actions

Projects


Security

...

 main

...

ProjExD / ex06 / test.py / <> Jump to

 c0a21023

基本機能 + 追加機能実装

History

1 contributor

184 lines (158 sloc) | 5.88 KB

```
1  # -*- coding:utf-8 -*-
2  import pygame
3  from pygame.locals import *
4  import sys
5
6  # ボールの動きを計算
7  def calc_ball(ball_x, ball_y, ball_vx, ball_vy, bar1_x, bar1_y, bar2_x, bar2_y,wall_x,wall_y,wall_1,ball_1):
8      global flag
9      #プレイヤーのバーにあたった時
10     # wall_rct=wall_1.get_rect()
11     # ball_rct=ball_1.get_rect()
12     if ball_x <= bar1_x + 10.:
13         if ball_y >= bar1_y - 7.5 and ball_y <= bar1_y + 42.5:
14             ball_x = 20.
15             ball_vx = -ball_vx
16     #AIのバーにあたった時
17     if ball_x >= bar2_x - 15.:
18         if ball_y >= bar2_y - 7.5 and ball_y <= bar2_y + 42.5:
19             ball_x = 605.
20             ball_vx = -ball_vx
21     #障害物にあたった時
22     #右側にあたった時
23     if wall_1.colliderect(ball_1) and not flag:
24         ball_vx=-ball_vx
25         flag = True
26     if not wall_1.colliderect(ball_1):
27         flag = False
28
29
30     #画面外に出たとき
31     if ball_x < 5.:
32         ball_x, ball_y = 320., 232.5
33     elif ball_x > 620.:
34         ball_x, ball_y = 307.5, 232.5
35     if ball_y <= 10.:
36         ball_vy = -ball_vy
37         ball_y = 10.
38     elif ball_y >= 457.5:
39         ball_vy = -ball_vy
40         ball_y = 457.5
41
42     return ball_x, ball_y, ball_vx, ball_vy
43
44 # AIの動きを計算
45 def calc_ai(ball_x, ball_y, bar2_x, bar2_y):
46     dy = ball_y - bar2_y
```

```
47     if dy > 80: bar2_y += 20
48     elif dy > 50: bar2_y += 15
49     elif dy > 30: bar2_y += 12
50     elif dy > 10: bar2_y += 8
51     elif dy < -80: bar2_y -= 20
52     elif dy < -50: bar2_y -= 15
53     elif dy < -30: bar2_y -= 12
54     elif dy < -10: bar2_y -= 8
55
56     if bar2_y >= 420.: bar2_y = 420.
57     elif bar2_y <= 10.: bar2_y = 10.
58     return bar2_y
59
60 # プレイヤーの動き
61 def calc_player(bar1_y, bar1_dy):
62     bar1_y += bar1_dy
63     if bar1_y >= 420.: bar1_y = 420.
64     elif bar1_y <= 10.: bar1_y = 10.
65     return bar1_y
66
67 #障害物の動き
68 def wall_mov(wall_x,wall_y,wall_vx,wall_vy):
69     if wall_y <= 15.:
70         wall_vy = -wall_vy
71         wall_y = 15.
72     elif wall_y >= 400.5:
73         wall_vy = -wall_vy
74         wall_y = 400.5
75     return wall_x, wall_y, wall_vx, wall_vy
76
77 # 得点の計算
78 def calc_score(ball_x, score1, score2):
79     if ball_x < 5.:
80         score2 += 1
81     if ball_x > 620.:
82         score1 += 1
83     return score1, score2
84
85 # イベント処理
86 def event(bar1_dy):
87     for event in pygame.event.get():
88         if event.type == QUIT: # 閉じるボタンが押されたら終了
89             pygame.quit()
90             sys.exit()
91         if event.type == KEYDOWN: # キーを押したら
92             if event.key == K_UP:
93                 bar1_dy = -10
94             elif event.key == K_DOWN:
95                 bar1_dy = 10
96         elif event.type == KEYUP: # キーを押し終わったら
97             if event.key == K_UP:
98                 bar1_dy = 0.
99             elif event.key == K_DOWN:
100                 bar1_dy = 0.
101     return bar1_dy
102
103 def main():
104     global flag
105     # 各パラメータ
106     bar1_x, bar1_y = 10. , 215.
107     bar2_x, bar2_y = 620., 215.
108     wall_x,wall_y=325,215.
109     ball_x, ball_y = 307.5, 232.5
110     bar1_dy, bar2_dy = 0. , 0.
111     ball_vx, ball_vy = 250., 250.
```

```
112     wall_vx, wall_vy = 250., 250.
113     score1, score2 = 0,0
114     ball_r = 7
115     flag = False
116
117     # pygameの設定
118     pygame.init() # Pygameの初期化
119     screen = pygame.display.set_mode((640,480),0,32) # 画面の大きさ
120     pygame.display.set_caption("PONG") # 画面タイトル
121     clock = pygame.time.Clock()
122     font = pygame.Font.SysFont(None,40) # 画面文字の設定
123
124     # 背景の設定
125     back = pygame.Surface((640,480))
126     background = back.convert()
127
128     # ボールを打つバーの設定
129     bar = pygame.Surface((10,50))
130     bar1 = bar.convert()
131     bar1.fill((255,255,255))
132     bar2 = bar.convert()
133     bar2.fill((255,255,255))
134
135     # ボールの設定
136     circ_sur = pygame.Surface((20,20))
137     pygame.draw.circle(circ_sur,(255,255,255),(ball_r, ball_r), ball_r)
138     ball = circ_sur
139     ball.set_colorkey((0,0,0))
140
141     #障害物の設定
142     wall_s = pygame.Surface((10,90))
143     wall = wall_s.convert()
144     wall.fill((255,255,255))
145
146     while (1):
147         # 各オブジェクトの描画
148         scr=screen.blit(background,(0,0))
149         screen.fill((75,0,125))
150         pygame.draw.aaline(screen,(255,255,255),(330,5),(330,475)) # 中央線の描画
151         bar_1=screen.blit(bar1,(bar1_x,bar1_y)) # プレイヤー側バーの描画
152         bar_2=screen.blit(bar2,(bar2_x,bar2_y)) # CPU側バーの描画
153         wall_1=screen.blit(wall,(wall_x,wall_y)) #
154         ball_1=screen.blit(ball,(ball_x, ball_y)) # ボールの描画
155         screen.blit(font.render(str(score1), True,(255,255,255)),(250.,10.))
156         screen.blit(font.render(str(score2), True,(255,255,255)),(400.,10.))
157
158
159
160     # プレイヤー側バーの位置
161     bar1_dy = event(bar1_dy)
162     bar1_y = calc_player(bar1_y,bar1_dy)
163
164     # ボールの移動
165     time_passed = clock.tick(30)
166     time_sec = time_passed / 1000.0
167     ball_x += ball_vx * time_sec
168     ball_y += ball_vy * time_sec
169     wall_y += wall_vy * time_sec
170
171     # 得点の計算
172     score1, score2 = calc_score(ball_x, score1, score2)
173
174     # CPUのバー速度を計算
175     bar2_y = calc_ai(ball_x, ball_y, bar2_x, bar2_y)
176
```

```
177     # ボールの速度・位置を計算
178     ball_x, ball_y, ball_vx, ball_vy = calc_ball(ball_x, ball_y, ball_vx, ball_vy, bar1_x, bar1_y, bar2_x,
bar2_y,wall_x,wall_y,wall_1,ball_1)
179     wall_x, wall_y, wall_vx, wall_vy = wall_mov(wall_x,wall_y,wall_vx,wall_vy)
180     pygame.display.update() # 画面を更新
181
182
183 if __name__ == "__main__":
184     main()
```

c0a21023 / ProjExD

Public

<> Code

Issues 19

Pull requests

Actions

Projects

Security

main

ProjExD / ex06 / README.md

c0a21023 基本機能 + 追加機能実装

History

1 contributor

20 lines (20 sloc) | 1.19 KB

## 卓球ゲーム（ex06/test.py）

### ゲーム概要

- ex05/test.pyを実行すると。紫の背景に左端と中央、右端に3つのバーが出現し、左端のバーはプレイヤーが、中央のバーは上下に移動し、右端のバーはコンピューターが操作し、中央から出現するボールをバーで跳ね返し遊ぶゲームが実行される。
- また、横端にボールが到達するとその逆にいるプレイヤーのスコアが加算される

### 操作方法

- 矢印キーでこうかとんを上下に移動する

### 基本機能

- 両端にあるバーがボールを跳ね返すことができる機能
- 右端にあるバーが自動的にボールを追いかけてボールを返す機能
- 両端の壁にボールがあたることでスコアが加算される機能
- スコアの表示

### 追加機能

- 中央に出現するバーにボールがあたることでボールを跳ね返すことのできる機能
- 背景色の変更
- 中央のバーが上下に自動で移動する機能

### ToDo（実装しようと思ったけど時間がなかった）

- ☐ ボールを増やす機能
- ☐ 勝ち負け判定

### メモ

- とくになし