

c0a22043 /  
ProjExD\_05

&lt;&gt; Code

Pull requests

Actions

Projects

Wiki

Security



BGM



ProjExD\_05 / README.md



c0a22043 BGM機能

3 minutes ago



23 lines (22 loc) · 617 Bytes

Preview

Code

Blame

Raw



# ゲームのタイトル

## 実行環境の必要条件

- python >= 3.10
- pygame >= 2.1

## ゲームの概要

- 線と球をよける

## ゲームの実装

### ### 共通基本機能

- 基本行動に関するクラス
- 線とドットの親クラス
- 垂直の線に関するクラス
- 水平の線に関するクラス
- ドットに関するクラス

## 担当追加機能

- BGM

## ToDo

- ☐ BGM機能

## メモ

### ## 追加機能BGM

- ゲームがスタートするとBGMが再生される
- Tabキーを押すとBGMが切り替わる
- ゲームオーバーするとゲームオーバーのBGMが流れる

c0a22043 /  
ProjExD\_05

&lt;&gt; Code

Pull requests

Actions

Projects

Wiki

Security

...



BGM

...

ProjExD\_05 / test1.py



c0a22043 BGM機能

4 minutes ago



289 lines (263 loc) · 10.3 KB

Code

Blame

Raw



```
1  import pygame
2  import sys
3  import random
4  import numpy as np
5  import datetime
6
7  OCTO_CAT_VELOCITY = 4
8  OCTO_CAT_JUMP = 20
9  COLORS = [[255,0,0],[255,165,0],[255,255,0],[0,255,0],[0,255,255],[0,0,255],[128,0,128]]
10
11  pygame.init()
12  screen = pygame.display.set_mode((640, 480))
13  pygame.display.set_caption("Jump the Rope")
14  heart_image = pygame.image.load("ex05/images/heart.png")
15  clock = pygame.time.Clock()
16
17  class Octo_Cat:
18      def __init__(self,x,y):
19          #position
20          self.x = x
21          self.y = y
22          #image before scaling
23          self.rough_image = pygame.image.load("ex05/images/greenoctocat.png").convert()
24          #properly scaled image
25          self.image = pygame.transform.scale(self.rough_image, (20,20))
26          #image to show when the player is jumping
27          self.immune_image = pygame.transform.scale(self.rough_image, (30,30))
28          #about movements
29          self.move_right = False
30          self.move_left = False
31          self.move_up = False
32          self.move_down = False
33          #if jumping or not
34          self.immunity = False
35          #the player cannot jump more than a specified amount of time
36          self.immunity_count = 0
```

```

38         self.life = 4
39         #if the player is inflicted with damage, it cannot be inflicted again for a certain amou
40         self.life_lost_time = 0
41
42     #movements
43     def update(self,event):
44         if event.type == pygame.KEYDOWN:
45             if event.key == pygame.K_RIGHT:
46                 self.move_right = True
47             if event.key == pygame.K_LEFT:
48                 self.move_left = True
49             if event.key == pygame.K_UP:
50                 self.move_up = True
51             if event.key == pygame.K_DOWN:
52                 self.move_down = True
53             if event.key == pygame.K_SPACE:
54                 self.immunity = True
55         elif event.type == pygame.KEYUP:
56             if event.key == pygame.K_RIGHT:
57                 self.move_right = False
58             if event.key == pygame.K_LEFT:
59                 self.move_left = False
60             if event.key == pygame.K_UP:
61                 self.move_up = False
62             if event.key == pygame.K_DOWN:
63                 self.move_down = False
64
65     #defiens ropes and dots
66     class Rope:
67         def __init__(self,x=0,y=0,velocity=0,tilt=0,color=0):
68             self.x = x
69             self.y = y
70             self.velocity = velocity
71             self.tilt = tilt
72             self.color = color
73         def update(self):
74             return
75         def judge(self,octo_cat):
76             return
77
78     #vertical ropes
79     class Straight_Rope(Rope):
80         def update(self):
81             if(self.x > 635):
82                 self.direction = "LEFT"
83             elif(self.x < 5):
84                 self.direction = "RIGHT"
85             if(self.direction == "RIGHT"):
86                 self.x += self.velocity
87             elif(self.direction == "LEFT"):
88                 self.x -= self.velocity
89             pygame.draw.line(screen, COLORS[3], [self.x, 0], [self.x, 480], 5)
90
91     #checks if the player and the rope collided
92     def judge(self,octo_cat):

```

```
93         if(self.x > (octo_cat.x) and self.x < (octo_cat.x + 20)):
94             return True
95         else:
96             return False
97
98     #horizontal ropes
99     class Straight_Rope_Horizontal(Rope):
100     def update(self):
101         if(self.y > 475):
102             self.direction = "DOWN"
103         elif(self.y < 5):
104             self.direction = "UP"
105         if(self.direction == "DOWN"):
106             self.y -= self.velocity
107         elif(self.direction == "UP"):
108             self.y += self.velocity
109         pygame.draw.line(screen, COLORS[3],[0, self.y], [640, self.y], 5)
110
111     #checks if the player and the rope collided
112     def judge(self,octo_cat):
113         if(self.y > (octo_cat.y) and self.y < (octo_cat.y + 20)):
114             return True
115         else:
116             return False
117
118     #dots
119     class Shooting_Star(Rope):
120     def update(self):
121         self.x += self.tilt
122         self.y += self.velocity
123         pygame.draw.circle(screen, COLORS[self.color], [self.x, self.y], 6)
124
125     #checks if the player and the dot collided
126     def judge(self,octo_cat):
127         if((self.y > (octo_cat.y) and self.y < (octo_cat.y + 20)) and (self.x > (octo_cat.x) and
128             return True
129         else:
130             return False
131
132     def open():
133         endFlag = False
134         font1 = pygame.font.SysFont(None, 80)
135         text1 = font1.render("Jump the Rope", False, (255,255,255))
136         font2 = pygame.font.SysFont(None, 40)
137         text2 = font1.render("Press Any Key to Start", False, (255,255,255))
138
139         while endFlag == False:
140             screen.fill((0,0,0))
141             screen.blit(text1,(30,50))
142             screen.blit(text2,(20,150))
143             pygame.display.update()
144             for event in pygame.event.get():
145                 if event.type == pygame.QUIT:
146                     endFlag = True
147                 elif event.type == pygame.KEYDOWN:
148                     endFlag = True
```

```
149         main()
150
151     def main():
152         global is_new_bgm_playing
153         endFlag = False
154         octo_cat = Octo_Cat(400,400)
155         time_elapsed = 0
156         force_quit = False
157         BGM = pygame.mixer.Sound('ex05/fig/BGM.mp3')
158         BGM.set_volume(0.2) # 音量を設定 (0.0から1.0の範囲)
159         BGM.play(-1) # -1を渡すと無限ループ再生
160         New_BGM = pygame.mixer.Sound('ex05/fig/maou.mp3')
161         is_new_bgm_playing = False # 新しいBGMの再生状態をトラッキング
162         game_over_sound = pygame.mixer.Sound('ex05/fig/dead.wav')
163
164
165         ropes = []
166
167         while endFlag == False:
168             clock.tick(60)
169             time_elapsed += 1
170             screen.fill((0,0,0))
171
172             for event in pygame.event.get():
173                 if event.type == pygame.QUIT:
174                     endFlag = True
175                     force_quit = True
176                 elif event.type == pygame.KEYDOWN:
177                     if event.key == pygame.K_TAB:
178                         if pygame.mixer.get_busy():
179                             BGM.stop() # 現在のBGMを停止
180                             New_BGM.stop()
181                             New_BGM.play(-1) # 新しいBGMを再生
182                         octo_cat.update(event)
183
184             #move the player
185             if octo_cat.move_right == True:
186                 if octo_cat.x < 620:
187                     octo_cat.x += OCTO_CAT_VELOCITY
188             if octo_cat.move_left == True:
189                 if octo_cat.x > 00:
190                     octo_cat.x -= OCTO_CAT_VELOCITY
191             if octo_cat.move_up == True:
192                 if octo_cat.y > 00:
193                     octo_cat.y -= OCTO_CAT_VELOCITY
194             if octo_cat.move_down == True:
195                 if octo_cat.y < 460:
196                     octo_cat.y += OCTO_CAT_VELOCITY
197
198             #make the instances of ropes and dots
199             if (time_elapsed == 20):
200                 straight_rope = Straight_Rope(0,0,3)
201                 ropes.append(straight_rope)
202             if (time_elapsed == 300):
203                 straight_rope_horizontal = Straight_Rope_Horizontal(0,0,3)
```

```

204         ropes.append(straight_rope_horizontal)
205     if (time_elapsed == 600):
206         straight_rope = Straight_Rope(0,0,3)
207         ropes.append(straight_rope)
208     if (time_elapsed == 860):
209         straight_rope_horizontal = Straight_Rope_Horizontal(0,0,3)
210         ropes.append(straight_rope_horizontal)
211     if(random.randrange(200) < 6):
212         shooting_star1 = Shooting_Star(random.randrange(640),0,random.randrange(5) + 5,random.randrange(5) + 5)
213         ropes.append(shooting_star1)
214         shooting_star2 = Shooting_Star(random.randrange(640),0,random.randrange(5) + 5,random.randrange(5) + 5)
215         ropes.append(shooting_star2)
216         shooting_star3 = Shooting_Star(10,random.randrange(480),random.randrange(5) + 5,random.randrange(5) + 5)
217         ropes.append(shooting_star3)
218         shooting_star4 = Shooting_Star(10,random.randrange(480),random.randrange(5) + 5,random.randrange(5) + 5)
219         ropes.append(shooting_star4)
220
221     #move all the ropes and dots
222     for rope in ropes:
223         rope.update()
224         if (rope.x < 0 or rope.x > 640) or (rope.y < 0 or rope.y > 480):
225             ropes.remove(rope)
226         if(time_elapsed % 1000 == 0) and time_elapsed != 0:
227             rope.velocity += 1
228
229     #if the player is jumping, do not check if it collided with ropes or dots
230     if(octo_cat.immunity == True):
231         octo_cat.immunity_count += 1
232         if (octo_cat.immunity_count < OCTO_CAT_JUMP ):
233             screen.blit(octo_cat.immune_image,(octo_cat.x,octo_cat.y))
234         else:
235             octo_cat.immunity = False
236             octo_cat.immunity_count = 0
237             screen.blit(octo_cat.image,(octo_cat.x,octo_cat.y))
238     else:
239         screen.blit(octo_cat.image,(octo_cat.x,octo_cat.y))
240     for rope in ropes:
241         if(rope.judge(octo_cat) == True) and (octo_cat.life_lost_time + 30 < time_elapsed):
242             octo_cat.life_lost_time = time_elapsed
243             octo_cat.life -= 1
244             if octo_cat.life == 0:
245                 endFlag = True
246     if octo_cat.life == 0:
247         endFlag = True
248     if pygame.mixer.get_busy():
249         BGM.stop() # ゲームオーバー時にBGMを停止
250         New_BGM.stop()
251         game_over_sound.play() # ゲームオーバー音楽を再生
252
253     for i in range(octo_cat.life - 1):
254         screen.blit(heart_image,(i * 30,50))
255     pygame.display.update()
256     quit(time_elapsed,force_quit)
257
258     #when quitting the game
259     def quit(force_quit):

```

```
259  def quit(score, force_quit):
260      if force_quit == False:
261          endFlag = False
262          yourScore = "your score: " + str(score)
263          font1 = pygame.font.SysFont(None, 40)
264          text1 = font1.render(yourScore, False, (255,255,255))
265          font2 = pygame.font.SysFont(None, 40)
266          text2 = font1.render("Press Any Key to Re-Start", False, (255,255,255))
267
268      while endFlag == False:
269          screen.fill((0,0,0))
270          screen.blit(text1,(20,50))
271          screen.blit(text2,(20,150))
272          pygame.display.update()
273          for event in pygame.event.get():
274              if event.type == pygame.QUIT:
275                  endFlag = True
276              elif event.type == pygame.KEYDOWN:
277                  endFlag = True
278                  main()
279      #retrieve the highest score
280      data = np.loadtxt("score/score.tsv",dtype="str",delimiter=",")
281      highest_score = data[1]
282      print("The highest score so far: " + highest_score)
283      if(score > int(highest_score)):
284          save_data = np.array([str(datetime.datetime.today()),str(score)])
285          np.savetxt('score/score.tsv',save_data,delimiter=',', fmt="%s")
286      pygame.quit()
287
288  if __name__ == "__main__":
289      open()
```