



 c0a22017 / ProjExD\_05


🔍 Type  to search >\_

+ ▾









<> Code

 Issues

 Pull requests

 Actions

 Projects

 Wiki

 Security

 Insights

 Settings

 shield ▾

ProjExD\_05 / README.md 

🔍 Go to file t

...

 c0a22017 自分の書いた

1180556 · now

 History

...

Preview

Code

Blame

71 lines (54 loc) · 3.98 KB

Raw





 ▾



# ゲームのタイトル

Banging Splash

## 🔗 実行環境の必要条件

- python >= 3.10
- pygame >= 2.1

## ゲームの概要

プレイヤーは画面下部に配置されています。プレイヤーは左右に移動し、スペースキーを押すことで弾を発射できます。敵は画面上部からランダムに出現し、左右に移動します。プレイヤーが弾を当てると、新しい敵がランダムな位置に再出現します。ゲームの目標は、できるだけ多くの敵を倒してスコアを稼ぐことです。

# ゲームの実装

操作方法: 左矢印キー: プレイヤーを左に移動 右矢印キー: プレイヤーを右に移動 スペースキー: 弾を発射 左シフトキー: スコアを消費してバリアを生成 右シフトキー: スコアを消費して散弾銃モード

要素:4 プレイヤー (playerImg) : 左右に移動できる宇宙船。 敵 (enemyImg) : 画面上部から左右に移動する敵宇宙船。 弾 (bulletImg) : プレイヤーと敵が発射する弾。 バリア (shield) : プレイヤーが生成するバリア

得点: 敵宇宙船を倒すごとに、得点が加算されます。得点は画面左上に表示されます。

注意点: 敵宇宙船が放つ弾がプレイヤーに当たると、ゲームが終了します。

コードの構造: Pygameライブラリを使用してウィンドウを作成し、ゲームのメインループを実装しています。 イベントハンドリングを使用して、キーボードの入力に応じてプレイヤーの動きや弾の発射を制御しています。 isCollision 関数は、弾と敵の衝突を検出しています。 スコアは画面左上に表示され、敵を倒すと増加します。

クリア条件: 敵が放つ弾に当たらずプレイヤーは敵を打ち落とし最高点を目指す

参考サイト: [https://www.youtube.com/watch?v=fAJ\\_BjLd3Ro](https://www.youtube.com/watch?v=fAJ_BjLd3Ro)

## ###共通基本機能

- 主人公キャラクターに関するクラス 方向キーで左右に移動 スペースキーで弾の放出
- 敵に関するクラス ランダムで左右に移動一定の時間ごとに弾の放出

## 担当追加機能

- 伊藤 ゲージをある量使用して発生させるバリア生成機能

## ToDo

- ☐ バリア機能

☐ 弾の無効化機能

## メモ

シールドの状態や描画は、shield\_state変数で管理されています。shield\_stateが 'ready' のときはシールドが非アクティブで、'active' のときはシールドがアクティブです。



キー入力 ('S'キー) によってシールドをアクティブにする処理があります。具体的には、キーが押されたとき (pygame.KEYDOWN イベント)、'S'キーが押された場合、shield\_stateを 'active' に設定し、shieldXおよびshieldYをプレイヤーの位置に設定しています。



シールドがアクティブの場合、draw\_shield関数が呼び出されて、シールドが描画されます。この関数は指定された位置 (x + 33, y) に半径 shield\_radius の円を描画します。

敵がプレイヤーのシールドに当たったかどうかの判定は、isCollision関数で行われています。この関数は、円と円の衝突判定を行います。具体的には、isCollision(enemyX, enemyY, playerX, playerY, shield\_radius, 32)のように呼び出されています。もしシールドがアクティブでかつ敵との衝突が検出されれば、shield\_stateを 'ready' に設定し、新しい敵の位置をランダムに設定しています。

これにより、プレイヤーが 'S' キーを押すことでシールドをアクティブにし、敵の攻撃を防ぐことができるようになります。

今日は、ほかの人のゲージ生成と敵の弾が放出されて当たるコードをマージしていない とりあえずシールドを生成して敵が自分に当たったら無効化されるコードの作成をした

  c0a22017 / ProjExD\_05


🔍 Type  to search >\_ + ▾ 🕒 🔗 📁 🏠 

<> Code 🕒 Issues 🔗 Pull requests 2 🕒 Actions 📁 Projects 📖 Wiki 🛡 Security 📈 Insights ⋮

 shield ▾ **ProjExD\_05** / Banging\_Splash.py 

🔍 Go to file

t

 c0a22017 シールド完成





4ba6d0d · 12 minutes ago

🕒 History



133 lines (108 loc) · 3.51 KB

Code Blame

Raw    ▾ 

```
1  import pygame
2  from pygame import mixer
3  import random
4  import math
5
6  pygame.init()
7
8  screen = pygame.display.set_mode((800, 600))
9  # screen.fill((150, 150, 150))
10 pygame.display.set_caption('Invaders Game')
11
12 # Player
13 playerImg = pygame.image.load('ex05/player.png')
14 playerX, playerY = 370, 480
15 playerX_change = 0
16
17 # Enemy
18 enemyImg = pygame.image.load('ex05/enemy.png')
19 enemyX = random.randint(0, 736)
```

## Symbols



Find definitions and references for functions and other symbols in this file by clicking a symbol below or in the code.

≡ Filter symbols

r

const screen

const playerImg

const playerX\_change

const enemyImg

const enemyX

const enemyY

const bulletImg

const bullet\_state

const shield\_radius

```
21 enemyX_change, enemyY_change = 4, 40
22
23 # Bullet
24 bulletImg = pygame.image.load('ex05/bullet.png')
25 bulletX, bulletY = 0, 480
26 bulletX_change, bulletY_change = 0, 3
27 bullet_state = 'ready'
28
29 #ここ
30 # Shield
31 shield_radius = 80
32 shield_color = (218, 12, 95) # Blue color for the shield
33 shield_state = 'ready'
34
35 # Score
36 score_value = 0
37
38 def player(x, y):
39     screen.blit(playerImg, (x, y))
40
41 def enemy(x, y):
42     screen.blit(enemyImg, (x, y))
43
44 def fire_bullet(x, y):
45     global bullet_state
46     bullet_state = 'fire'
47     screen.blit(bulletImg, (x + 16, y + 10))
48
49 #ここ
50 def draw_shield(x, y):
51     pygame.draw.circle(screen, shield_color, (x + 33, y), shield_radius)
52
53 def isCollision(obj1X, obj1Y, obj2X, obj2Y, obj1_radius, obj2_radius):
54     distance = math.sqrt(math.pow(obj1X - obj2X, 2) + math.pow(obj1Y - obj2Y, 2))
55     if distance < (obj1_radius + obj2_radius):
56         return True
```

```
const shield_color
const shield_state

const score_value

func player
func enemy
func fire_bullet
```

```
57         else:
58             return False
59
60     running = True
61     while running:
62         screen.fill((0, 0, 0))
63
64         for event in pygame.event.get():
65             if event.type == pygame.QUIT:
66                 running = False
67
68             if event.type == pygame.KEYDOWN:
69                 if event.key == pygame.K_LEFT:
70                     playerX_change = -1.5
71                 if event.key == pygame.K_RIGHT:
72                     playerX_change = 1.5
73                 if event.key == pygame.K_SPACE:
74                     if bullet_state is 'ready':
75                         bulletX = playerX
76                         fire_bullet(bulletX, bulletY)
77                 #ここ
78                 if event.key == pygame.K_s: # Press 'S' to activate shield
79                     if shield_state == 'ready':
80                         shieldX, shieldY = playerX, playerY
81                         shield_state = 'active'
82
83             if event.type == pygame.KEYUP:
84                 if event.key == pygame.K_LEFT or event.key == pygame.K_RIGHT:
85                     playerX_change = 0
86
87     # Player
88     playerX += playerX_change
89     if playerX <= 0:
90         playerX = 0
91     elif playerX >= 736:
92         playerX = 736
```

```
93     #ここ
94     # Shield
95     if shield_state == 'active':
96         draw_shield(playerX, playerY)
97
98
99     # Enemy
100    if enemyY > 440:
101        break
102    enemyX += enemyX_change
103    if enemyX <= 0: #左端に来たら
104        enemyX_change = 4
105        enemyY += enemyY_change
106    elif enemyX >=736: #右端に来たら
107        enemyX_change = -4
108        enemyY += enemyY_change
109
110    #ここ
111    if shield_state == 'active' and isCollision(enemyX, enemyY, playerX, playerY, shield_
112        shield_state = 'ready'
113        enemyX = random.randint(0, 736)
114        enemyY = random.randint(50, 150)
115
116    # Bullet Movement
117    if bulletY <=0:
118        bulletY = 480
119        bullet_state = 'ready'
120
121    if bullet_state is 'fire':
122        fire_bullet(bulletX, bulletY)
123        bulletY -= bulletY_change
124
125    # Score
126    font = pygame.font.SysFont(None, 32) # フォントの作成 Noneはデフォルトのfreesansbold.tt
127    score = font.render(f"Score : {str(score_value)}", True, (255,255,255)) # テキストを描
128    screen.blit(score, (20,50))
```

```
129  
130     player(playerX, playerY)  
131     enemy(enemyX, enemyY)  
132  
133     pygame.display.update()
```