



c0a2205337 / ProjExD\_5

🔍 Type to search



&lt;&gt; Code

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

ProjExD\_5 / README.md



c0a2205337 READMEを追加

797f3d1 · 4 minutes ago



History



Preview

Code

Blame

30 lines (20 loc) · 1.22 KB

Raw



# 吹き飛ば！！こうかとん！！！！

## 実行環境の必要条件

- python >= 3.10
- pygame >= 2.1

## ゲームの概要

いわゆるボンバーマンのオマージュ。デフォルトで二人プレイにするのが特徴。長方形のマップがあり、等間隔に侵入できないブロックがある。特定のボタン（左SHIFTと右SHIFT）を押すと爆弾を設置することができ、十字の形に爆発する。

# ゲームの実装

---

## 共通基本機能

- 背景と破壊できないブロックが等間隔にあるマップの作成 \*1Pキャラクターの作成：1PキャラクターはWASDで操作できるものとする。なめらかに動くのではなく、1マスずつ移動を行う。 \*1Pキャラクターの移動機能：WASDに対応して、上下左右に移動を行う。

## 担当追加機能

- 破壊可能な壁を設置する機能（担当：喜多村）：初期マップにて、プレイヤーが通行可能なマスに破壊可能な壁をランダムに配置する機能。

## ToDo

[ ] 勝敗判定機能 [ ] score

## メモ

## 参考

\*<https://www.youtube.com/watch?v=CTwmZ6h21dk> \*グループ内のイメージを確立するために利用した動画



c0a2205337 / ProjExD\_5

Type / to search



&lt;&gt; Code



Pull requests



Actions



Projects



Wiki



Security



Insights



Settings

ProjExD\_5 / original\_game.py



c0a2205337 壊れる壁の実装

3aacf86 · 8 minutes ago



History



115 lines (97 loc) · 3.48 KB

Code

Blame

Raw



```
1  import math
2  import os
3  import random
4  import sys
5  import time
6  import pygame as pg
7
8  WIDTH = 1000 # ゲームウィンドウの幅 25
9  HEIGHT = 680 # ゲームウィンドウの高さ 17
10 SQ_SIDE = 40 # マス一辺
11 TATE = 17 # マス数
12 YOKO = 25
13 MAIN_DIR = os.path.split(os.path.abspath(__file__))[0]
14 TYPE_DICT = {0:"floor",1:"wall",2:"block",3:"bomb"}
15
16
17 def check_bound(obj,map_lst:list,mv):
18     """
19     obj:対象のインスタンス(座標としてself.x,self.yを定義してあるもの)
20     map_lst:マップ
```

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 WIDTH

const HEIGHT

const SQ\_SIDE

const TATE

const YOKO

const MAIN\_DIR

const TYPE\_DICT

func check\_bound

class Player

```
21     mv:動く距離
22     """
23     if map_lst[obj.x+mv[0]][obj.y+mv[1]] == 0:
24         return obj.x+mv[0],obj.y+mv[1]
25     else:
26         return obj.x,obj.y
27
28
29
30  ✓ class Player():
31  ✓     def __init__(self):
32         self.x = 3
33         self.y = 11
34         self.img = pg.transform.rotozoom(pg.image.load(f"{MAIN_DIR}/fig/player.png"), 0,
35         self.rect = self.img.get_rect()
36         self.rect.center = (self.x*SQ_SIDE,self.y*SQ_SIDE)
37
38     def update(self,mv,screen: pg.Surface,map_lst):
39         self.x,self.y = check_bound(self,map_lst,mv)
40         self.rect.center = (self.x*SQ_SIDE,self.y*SQ_SIDE)
41         screen.blit(self.img,self.rect.center)
42
43
44
45
46
47  ✓ def main():
48     pg.display.set_caption("吹き飛ば!!こうかとん!!!")
49     player = Player()
50     screen = pg.display.set_mode((WIDTH, HEIGHT))
51     bg_img = pg.image.load(f"{MAIN_DIR}/fig/pg_bg.jpg")
52     wall_image = pg.transform.rotozoom(pg.image.load(f"{MAIN_DIR}/fig/wall.png"),0, 2.5)
53     dwall_image = pg.transform.rotozoom(pg.image.load(f"{MAIN_DIR}/fig/damaged_wall.png")
54     map_lst = [[0 for i in range(17)] for j in range(26)]
55     for x in range(YOKO):
56         for y in range(TATE):
```

```
func __init__
func update
func main
```

```
57         num = random.randint(0,2)
58         if num != 0:
59             if not((player.x-1 <= x <= player.x+1)and(player.y-1 <= y <= player.y+1))
60                 map_lst[x][y] = 2
61
62     while True:
63         screen.blit(bg_img, [0, 0])
64         # 壁設置
65         for x in range(YOKO):
66             for y in range(TATE):
67                 if x == 0 or x == YOKO-1:
68                     map_lst[x][y] = 1
69                 elif y == 0 or y == TATE-1:
70                     map_lst[x][y] = 1
71                 elif x%2 == 0 and y%2 == 0:
72                     map_lst[x][y] = 1
73                 if map_lst[x][y] == 1:
74                     screen.blit(wall_image,(x*SQ_SIDE,y*SQ_SIDE))
75                 # 壊れる壁配置
76                 if map_lst[x][y] == 2:
77                     screen.blit(dwll_image,(x*SQ_SIDE,y*SQ_SIDE))
78
79
80     key_lst = pg.key.get_pressed()
81     mv = [0,0]
82     for event in pg.event.get():
83         if event.type == pg.QUIT:
84             return 0
85         if event.type == pg.KEYDOWN:
86             if event.key == pg.K_UP:
87                 mv[1] -= 1
88             if event.key == pg.K_DOWN:
89                 mv[1] += 1
90             if event.key == pg.K_RIGHT:
91                 mv[0] += 1
92             if event.key == pg.K_LEFT:
```

```
93         mv[0] -= 1
94
95     player.update(mv, screen, map_1st)
96     pg.display.update()
97     pass
98     # score = Score()
99     # bird = Bird(3, (900, 400))
100    # bombs = pg.sprite.Group()
101    # beams = pg.sprite.Group()
102    # shield = pg.sprite.Group()
103    # exps = pg.sprite.Group()
104    # emys = pg.sprite.Group()
105    # gravitys = pg.sprite.Group()
106    # num = 3
107    # tmr = 0
108    # clock = pg.time.Clock()
109
110
111    if __name__ == "__main__":
112        pg.init()
113        main()
114        pg.quit()
115        sys.exit()
```