c0a2205445 / ProjExD_05 Public

Code Issues Pull requests Actions Projects ···

← Files

main ···

ProjExD_05 / air_game.py

c0a2205445 課題5用

4 minutes ago

202 lines (181 loc) · 7.92 KB

Code Blame

Raw

```python
import pygame
import time
from pygame.locals import *

pygame.init()

# 色の定義
white = (255, 255, 255)
black = (0, 0, 0)
green = (0, 150, 0)
red = (255, 0, 0)
blue = (0, 0, 255)
light_blue = (147, 251, 253)

# Clockの初期化
clock= pygame.time.Clock()
# 画面のサイズ
screen= pygame.display.set_mode((800, 600))
# 枠の区切り線
divline1 = screen.get_width()/2, 0
divline2 = screen.get_width()/2, screen.get_height()
# ゲームの名前
pygame.display.set_caption('Air Hockey!')
# フォントサイズ
smallfont = pygame.font.SysFont("comicsansms", 25)
medfont = pygame.font.SysFont("comicsansms", 45)
largefont = pygame.font.SysFont("comicsansms", 65)

# ゲームオブジェクトの作成
goalheight = 50
goalwidth = 20
goal1_x = 0
goal1_y = screen.get_height()/2 - 50
goal2_x = screen.get_width() - 10
goal2_y = screen.get_height()/2 - goalheight
goal1 = pygame.Rect(goal1_x, goal1_y, 10, 100)
goal2 = pygame.Rect(goal2_x, goal2_y, 10, 100)
paddle1 = pygame.Rect(screen.get_width()/2 - 200, screen.get_height()/2, 20, 20)
paddle2 = pygame.Rect(screen.get_width()/2 + 200, screen.get_height()/2, 20, 20)
paddleVelocity = 4
disc = pygame.Rect(screen.get_width()/2, screen.get_height()/2, 20, 20)
discVelocity = [5, 5]
```

```python
img = pygame.image.load('./ex05/disc.png')
bluepadimg = pygame.image.load('./ex05/bluepad.png')
redpadimg = pygame.image.load('./ex05/redpad.png')

# スコア
score1, score2 = 0, 0
serveDirection = 1

# パックをリセットする関数
def resetPuck():
    discVelocity[0] = 5 * serveDirection
```

```python
    print(score1, score2)
    disc.x= screen.get_width()/2
    disc.y= screen.get_height()/2

# テキストオブジェクトを作成する関数
def text_objects(text, color, size):
    if size == "small":
        textSurface = smallfont.render(text, True, color)
    if size == "medium":
        textSurface = medfont.render(text, True, color)
    if size == "large":
        textSurface = largefont.render(text, True, color)
    return textSurface, textSurface.get_rect()

# ポーズ画面
def pause():
    paused = True
    message_to_screen("Paused", green, -100, size = "large")
    message_to_screen("Press c to continue , q to quit", green, 25)
    pygame.display.update()
    while paused:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                quit()

            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_c:
                    paused = False

                elif event.key == pygame.K_q:
                    pygame.quit()
                    quit()
        clock.tick(5)

# メッセージを画面に表示する関数（得点とプレイヤー名の表示）
def message_to_screen(msg, color, y_displace = 0, x_displace = 0, size = "small"):
    textSurf, textRect = text_objects(msg, color, size)
    textRect.center = (screen.get_width()/2 + x_displace), ((screen.get_height()/2) + y_displace)
    screen.blit(textSurf, textRect)

# ゲームループ
def gameLoop():
    gameExit = False
    gameOver = False
    score2, score1 = 0, 0

    while not gameExit:

        for event in pygame.event.get():
            down2, up2, up, down, left2, right2, right, left = 0, 0, 0, 0, 0, 0, 0, 0
            print(event)
            if event.type == pygame.QUIT:
                gameExit = True
            keys = pygame.key.get_pressed()
            if keys[K_LEFT]:
                left = 1
            if keys[K_RIGHT]:
                right = 1
            if keys[K_UP]:
                up = 1
            if keys[K_DOWN]:
                down = 1
            if keys[K_a]:
                left2 = 1
            if keys[K_d]:
                right2 = 1
            if keys[K_w]:
                up2 = 1
            if keys[K_s]:
                down2 = 1
            if keys[K_p]:
                pause()

        # パドル1の更新
        paddle1.y += (down2 - up2) * paddleVelocity
        paddle1.x += (right2 - left2) * paddleVelocity
        # パドル1が範囲外に出ないように指定
        if paddle1.y < 0:
            paddle1.y = 0
        if paddle1.y > screen.get_height() - paddle1.height:
            paddle1.y = screen.get_height() - paddle1.height
        if paddle1.x < 0:
            paddle1.x = 0
        if paddle1.x > screen.get_width()/2 - paddle1.width:
            paddle1.x = screen.get_width()/2 - paddle1.width

        # パドル2の更新
        paddle2.y += (down-up) * paddleVelocity
        paddle2.x += (right-left) * paddleVelocity
        # パドル2が範囲外に出ないように指定
        if paddle2.y < 0:
            paddle2.y = 0
        if paddle2.y > screen.get_height() - paddle2.height:
            paddle2.y = screen.get_height() - paddle2.height
        if paddle2.x > screen.get_width() - paddle1.width:
            paddle2.x = screen.get_width() - paddle1.width
        if paddle2.x < screen.get_width()/2:
            paddle2.x = screen.get_width()/2

        # パックの更新
        disc.x += discVelocity[0]
        disc.y += discVelocity[1]
        if (disc.x <= goalwidth/4) and (disc.y <= screen.get_height()/2 + goalheight) and (disc.y >= screen.
            score2 += 1
            serveDirection = -1
            resetPuck()
        if (disc.x >= screen.get_width() - goalwidth - disc.width) and (disc.y <= screen.get_height()/2 + go
            score1 += 1
            serveDirection = 1
            resetPuck()
        if disc.x - 10 < 0 or disc.x + 25 > screen.get_width() :  # 左右の画面にディスクが衝突したとき
            discVelocity[0] *= -1;
        if disc.y - 10 < 0  or disc.y + 10 > screen.get_height() - disc.height:  # 上下の画面にディスクが衝突し
            discVelocity[1] *= -1
        if disc.colliderect(paddle1) or disc.colliderect(paddle2):  # プレイヤーとディスクが衝突したとき
            discVelocity[0] *= -1.2
            discVelocity[1] *= -1.2



        # 画面表示
        screen.fill(black)
        message_to_screen("Player 1", white, -250, -150, "small")
        message_to_screen(str(score1), white, -200, -150, "small")
        message_to_screen("Player 2", white, -250, 150, "small")
        message_to_screen(str(score2), white, -200, 150, "small")
        pygame.draw.rect(screen, (255, 100, 100), paddle1)
        pygame.draw.rect(screen, (20, 20, 100), paddle2)
        pygame.draw.rect(screen, light_blue, goal1)
        pygame.draw.rect(screen, light_blue, goal2)
        screen.blit(img, (disc.x, disc.y))
        screen.blit(bluepadimg, (paddle1.x-5, paddle1.y-5))
        screen.blit(redpadimg, (paddle2.x-5, paddle2.y-5))
        pygame.draw.circle(screen, white, (screen.get_width()/2, screen.get_height()/2), screen.get_width()/
        pygame.draw.line(screen , white , divline1, divline2 ,5 )
        pygame.draw.line(screen, blue, (0,0), (screen.get_width()/2 - 5,0) ,5)
        pygame.draw.line(screen, blue, (0, screen.get_height()), (screen.get_width()/2 - 5, screen.get_heigh
        pygame.draw.line(screen, red, (screen.get_width()/2 + 5, 0), (screen.get_width(), 0), 5)
        pygame.draw.line(screen, red, (screen.get_width()/2 + 5, screen.get_height()), (screen.get_width(),
        pygame.draw.line(screen, blue, (0, 0), (0, screen.get_height()/2 - goalheight), 5)
        pygame.draw.line(screen, blue, (0,screen.get_height()/2 + goalheight), (0, screen.get_height()), 5)
        pygame.draw.line(screen, red, (screen.get_width(), 0), (screen.get_width(), screen.get_height()/2 -
        pygame.draw.line(screen, red, (screen.get_width(), screen.get_height()/2 + goalheight), (screen.get_
        pygame.display.update()
        clock.tick(50)

gameLoop()
```