ProjExD_05 / tyari.py

c0b2200500 ジャンプ最終番 19 minutes ago

206 lines (166 loc) · 5.9 KB

Code Blame Raw

```python
import sys
import random
import pygame as pg

WIDTH = 1297
HEIGHT = 744



#完成版


def check_bound(obj: pg.Rect) -> tuple[bool, bool]:
    yoko, tate = True, True
    if obj.left < 0 or WIDTH < obj.right:  # 横方向のはみ出し判定
        yoko = False
    if obj.top < 0 or HEIGHT < obj.bottom:  # 縦方向のはみ出し判定
        tate = False
    return yoko, tate


class TYARI(pg.sprite.Sprite):

    def __init__(self,num: int ,xy:tuple[int,int]):
        img = pg.transform.rotozoom(pg.image.load(f"ex05/figs/tyari/{num}.png"), 0, 0.5)
        img = pg.transform.flip(img, True, False)  # デフォルトの自転車（右向き）
        self.imgs = {  # 0度から反時計回りに定義
            0: img,  # 右
            1: pg.transform.flip(img, True, False),
            2: pg.transform.rotozoom(img, 45, 1.0),  # 右上
            3: pg.transform.rotozoom(img, -45, 1.0),  # 右下

        }
        self.img = self.imgs[0]
        self.rct = self.img.get_rect()
        self.rct.center = xy
        self.on_floor = True
        self.acc = 1
        self.vel= -10
        self.y = HEIGHT * 0.85

    def change_img(self, num: int, screen: pg.Surface):
        self.img = self.imgs[num]
        screen.blit(self.img, self.rct)
```

```python

    def update(self, screen: pg.Surface):
        self.rct.move_ip(0,0)#自転車を描画
        screen.blit(self.img, self.rct)
        if self.on_floor:
            return
        self.vel += self.acc
        self.rct.y += self.vel
        if self.rct.y> HEIGHT * 0.85:
            self.rct.y= HEIGHT * 0.85
            self.vel = 0
            self.on_floor = True


    def jamp(self):#高さジャンプをするか決める

        if self.on_floor:
            self.on_floor=False
            self.vel = -30



    def events(self):
        for event in pg.event.get():
            if event.type == pg.KEYUP:
                TYARI.jamp()


class FLOOR:

    def __init__(self,floor_type):
        floor_img = pg.transform.rotozoom(pg.image.load(f"ex05/figs/floor.png"), 0, 1.0)

        self.img = floor_img
        self.map = [random.randint(0, 4) for i in range(200)]
        self.rct = [self.img.get_rect() for i in range(len(self.map))]

    def update(self, screen: pg.Surface,x):

        for i in range(len(self.map)):
            self.rct[i].move_ip(0,0)#地面を描画
            if  0< i * 66 - x + 66 < WIDTH:
                if self.map[i] != 0:
                    screen.blit(self.img, (i * 66 - x, HEIGHT-66))

    def check_bound(self,num):
        if self.map[(200+num) // 66]== 0:
            return 1
        else:
            return 0




class Coin(pg.sprite.Sprite):
    def __init__(self):
        """
        コイン画像を生成する
        """
```

```python
            super(Coin, self).__init__()
            #画像をリストに代入する
            self.imgs = list()
            for i in range(1,7):
                self.imgs.append(pg.transform.rotozoom(pg.image.load(f"ex05/coin01_gold01/{i}.png"),0,0.2))

            self.index = 0
            self.image = self.imgs[self.index]
            self.rect = self.image.get_rect()

        def update(self):
            if self.index >= len(self.imgs):
                self.index = 0

            self.image = self.imgs[self.index]
            self.index += 1

    class Score:
        """
        コインとチャリンコが接したときにスコアを表示するクラス
        1コイン  = 1ポイント
        """
        def __init__(self):
            self.font = pg.font.Font(None, 50)
            self.color = (0, 0, 255)
            self.score = 0
            self.image = self.font.render(f"Score: {self.score}", 0, self.color)
            self.rect = self.image.get_rect()
            self.rect.center = 100, HEIGHT-50

        def score_up(self, add):
            self.score += add

        def update(self, screen: pg.Surface):
            self.image = self.font.render(f"Score: {self.score}", 0, self.color)
            screen.blit(self.image, self.rect)


    def main():
        pg.display.set_caption("チャリ走DX")
        screen = pg.display.set_mode((WIDTH, HEIGHT))#スクリーンを描画
        clock  = pg.time.Clock()
        bg_img1 = pg.image.load("ex05/figs/bg.png")
        bg_img2 = pg.transform.flip(bg_img1, True ,False)
        bg_imgs = [bg_img1,bg_img2]
        bird = TYARI(1,(200,HEIGHT*0.85))#自転車を描画
        floor = FLOOR(1)
        reverse = False#反転
        tmr = 0
        bg = tmr
        x = tmr
        coin = Coin()
        coins = pg.sprite.Group()
        score = Score()
        tyaris = pg.sprite.Sprite()
        coin_group = pg.sprite.Group(coin)

        while True:
            for event in pg.event.get():
                if event.type == pg.QUIT: return
```

```python
        for i in range(-2,3):
            screen.blit(bg_imgs[i%2] , [-bg-i*WIDTH, 0])#背景を5枚描画

        if not reverse:#通常状態
          bg += 5
          x += 5
          if bg > WIDTH * 2:
              bg = 0
        else:#反転状態
          bg -= 5
          x -= 5
          if bg < -2*WIDTH:
              bg = 0
        if event.type == pg.KEYDOWN and event.key == pg.K_UP:
              bird.jamp()
        bird.update(screen)
        floor.update(screen,x)
        font = pg.font.Font(None,55)
        text = font.render(str(floor.check_bound(x)) , True , (255,255,255))
        screen.blit(text,[100,100])
        for coin in pg.sprite.groupcollide(coins, tyaris, True, True).keys():
            score.score_up(1)
        score.update(screen)
        if tmr % 3 == 1:
            coin_group.update()
        coin_group.draw(screen)
        pg.display.update()
        tmr += 1
        clock.tick(1000)
        for event in pg.event.get():
          if event.type == pg.KEYDOWN and event.key == pg.K_SPACE:#スペースで反転
              if reverse:
                reverse = False
                bird.change_img(0,screen)
              else:
                 reverse = True
                 bird.change_img(1,screen)
if __name__ == "__main__":
    pg.init()
    main()
    pg.quit()
    sys.exit()
```