c0a24008ec / **ProjExD_Group04**

Code Pull requests Actions Projects Wiki Security Insights

main ProjExD_Group04 / **KOKATAIL.py**

c0a24008ec 追加機能:enemy,コンフリクト解消 fa8a220 · last week

441 lines (388 loc) · 17.5 KB

Code Blame Raw

```python
import os
import pygame as pg
import sys
from typing import List
import random

WHITE = (255, 255, 255)
BLACK = (0, 0, 0)
YELLOW = (255, 255, 0)

os.chdir(os.path.dirname(os.path.abspath(__file__)))

pg.init()  # pygameの初期化は必ず最初に行う

# 初期化後にフォント作成
WIDTH, HEIGHT = 1920, 1080
font = pg.font.SysFont("meiryo", 50)
small_font = pg.font.SysFont("meiryo", 36)

class TurnManager():
    def __init__(self):
        self.num = 1
        self.turn = "player"

    def turn_change(self):
        if self.turn == "player":
            self.turn = "enemy"
        elif self.turn == "enemy":
            self.turn = "player"
            self.num += 1

class Player():
    def __init__(self, HP:int, ATK:int, turn:TurnManager):
        self.former_hp = HP
        self.hp = self.former_hp
        self.former_atk = ATK
        self.atk = self.former_atk
        self.turn = turn

class CommandBoxManager():
    """
    コマンドボックスの位置計算と描画管理クラス
    """

    commands = ["こうげき", "アクション", "アイテム", "にげる"]

    def __init__(self, player:Player, turn:TurnManager, font: pg.font.Font) -> None:
        self.box_width = 265
        self.box_height = 80
        self.box_y = HEIGHT - 300
        self.hp_bar_width = 160
        self.hp_bar_height = 20
        self.hp_bar_margin_top = 10
        self.hp_bar_y = self.box_y + self.box_height + self.hp_bar_margin_top + 15
        self.commands = __class__.commands
        self.font = font
        self.former_hp = player.former_hp
        self.hp = player.hp
        self.turn = turn

    def get_command_boxes(self) -> List[pg.Rect]:
        """
        コマンドボックスのpygame.Rectリストを生成する。
        """
        self.spacing = 40
        self.total_width = len(self.commands) * self.box_width + (len(self.commands) - 1
        self.start_x = (WIDTH - self.total_width) // 2
        self.boxes = []
        for i in range(len(self.commands)):
            x = self.start_x + i * (self.box_width + self.spacing)
            self.boxes.append(pg.Rect(x, self.box_y, self.box_width, self.box_height))
        return self.boxes

    def draw(self, screen: pg.Surface, selected_index: int) -> None:
        """
        コマンドボックスを画面に描画する。選択中のコマンドは黄色で強調。
        """
        self.boxes = self.get_command_boxes()
        if self.turn.turn == "player":
            for i, rect in enumerate(self.boxes):
                self.color = YELLOW if i == selected_index else WHITE
                pg.draw.rect(screen, self.color, rect, 4)

                self.command_text = self.font.render(self.commands[i], True, WHITE)
                self.command_text_x = rect.x + (rect.width - self.command_text.get_width
                self.command_text_y = rect.y + (rect.height - self.command_text.get_heig
                screen.blit(self.command_text, (self.command_text_x, self.command_text_y

```

```python
    def update(self, screen:pg.Surface):
        self.boxes = self.get_command_boxes()
        self.center_x = (self.boxes[1].centerx + self.boxes[2].centerx) // 2
        if self.hp <= 0:
            self.hp = 0
        self.hp_ratio = self.hp / self.former_hp
        self.hp_text = font.render(f"{self.hp} / {self.former_hp}", True, WHITE)
        self.command_text_x = self.center_x - self.hp_bar_width // 2 + self.hp_bar_widt
        self.command_text_y = self.hp_bar_y + 5 + (self.hp_bar_height - self.hp_text.ge
        # HPバー背景（黒）
        pg.draw.rect(screen, BLACK, (self.center_x - self.hp_bar_width // 2, self.hp_bar
        # HPバー黄色部分（HPの割合に応じた幅）
        pg.draw.rect(screen, YELLOW, (self.center_x - self.hp_bar_width // 2, self.hp_ba
        # HPバー枠（白）
        pg.draw.rect(screen, WHITE, (self.center_x - self.hp_bar_width // 2, self.hp_bar
        # HPバー横にHP数値表示
        screen.blit(self.hp_text, (self.command_text_x, self.command_text_y))

class EnemyBoxManager():
    def __init__(self, player:Player, turn:TurnManager, command:CommandBoxManager, font
        self.player = player
        self.turn = turn
        self.command = command
        self.font = font
        self.box_width = 1180
        self.box_height = 280
        self.enemybox_width = 280
        self.enemybox_height = 280
        self.start_x = (WIDTH - self.box_width) // 2
        self.start_y = self.command.box_y - (self.box_height + 20)
        self.comments_list = ["まずは練習だ",
                              "だんだん難しくなっていくぜ？",
                              "うまく避けろよ",
                              "どこまで耐えられるかな？",
                              "そろそろ本気だぜ？",
                              "まだいけるか？",
                              "そろそろ限界かな？",
                              "おまえ強いな",
                              "なんで生きてるの？",
                              "疲れたな",
                              "もはや怖いぞ？",
                              "もう疲れたよ",
                              "はよ〇ね",
                              "." * self.turn.num
                              ]
        self.comment = self.comments_list[0]  # 初期コメント
        self.death_comment = "\"GameOver\" だな"
        self.enemy_text = self.font.render(self.comment, True, WHITE)
        self.enemy_text_x = self.start_x + 30
        self.enemy_text_y = self.start_y + 30
        self.enemybox_x = (WIDTH - self.enemybox_width) // 2
        self.enemybox_y = self.command.box_y - (self.enemybox_height + 20)

    def drawbox(self, screen:pg.Surface):
        if self.turn.turn == "player":
            pg.draw.rect(screen, WHITE, (self.start_x, self.start_y, self.box_width, sel
        elif self.turn.turn == "enemy":
            pg.draw.rect(screen, WHITE, (self.enemybox_x, self.enemybox_y, self.enemybox

    def comments(self, screen: pg.Surface):
        if self.command.hp > 0:
            if self.turn.num < len(self.comments_list):
                self.comment = self.comments_list[self.turn.num]
            else:
                self.comment = self.comments_list[-1]
            self.enemy_text = self.font.render(self.comment, True, WHITE)
            screen.blit(self.enemy_text, (self.enemy_text_x, self.enemy_text_y))
        if self.command.hp == 0:
            self.comment = self.death_comment
            self.enemy_text = self.font.render(self.comment, True, WHITE)
            screen.blit(self.enemy_text, (self.enemy_text_x, self.enemy_text_y))


class ItemMenu:
    """
    アイテムメニューの描画と操作管理クラス
    """

    def __init__(self, items: List[str], font: pg.font.Font, small_font: pg.font.Font) -
        self.items = items
        self.font = font
        self.small_font = small_font
        self.selected_index = 0
        self.is_open = False

        self.menu_width = 800
        self.menu_height = 400
        self.menu_x = (WIDTH - self.menu_width) // 2
        self.menu_y = (HEIGHT - self.menu_height) // 2
        self.menu_rect = pg.Rect(self.menu_x, self.menu_y, self.menu_width, self.menu_he

    def draw(self, screen: pg.Surface) -> None:
        """
        アイテムメニューを画面に描画する。
        """
        # 背景と枠
        pg.draw.rect(screen, BLACK, self.menu_rect)
        pg.draw.rect(screen, WHITE, self.menu_rect, 3)

        # タイトル
        title = self.font.render("アイテム", True, WHITE)
        screen.blit(title, (self.menu_x + 20, self.menu_y + 10))

        # アイテムリスト表示
        for i, item in enumerate(self.items):
            color = YELLOW if i == self.selected_index else WHITE
            item_text = self.small_font.render(f"- {item}", True, color)
            screen.blit(item_text, (self.menu_x + 40, self.menu_y + 70 + i * 50))


class Enemy():
    """
    敵に関するクラス
    """
    img0 = pg.image.load(f"photo/enemy1_bob_v2.gif")
    img = pg.transform.rotozoom(img0,0,0.3)
    def __init__(self, HP:int, ATK:int, command:CommandBoxManager, turn:TurnManager):
        self.hp = HP
        self.atk = ATK
        self.image = __class__.img
        self.rect = self.image.get_rect()
        self.rect.centerx = WIDTH // 2
        self.rect.centery = 300
        self.command = command
        self.turn = turn
        self.tmr = 0

    def update(self, screen: pg.Surface):
        self.tmr += 1
        if self.tmr >= 20 and self.command.hp > 0:
            if self.tmr % 80 == 20:
                self.rect.centerx += 20
                self.rect.centery += 10
            elif self.tmr % 80 == 40:
                self.rect.centerx -= 20
                self.rect.centery -= 10
            elif self.tmr % 80 == 60:
                self.rect.centerx -= 20
                self.rect.centery += 10
            elif self.tmr % 80 == 0:
                self.rect.centerx += 20
                self.rect.centery -= 10
        elif self.command.hp == 0:
            self.rect.centerx = WIDTH // 2
            self.rect.centery = 300
        screen.blit(self.image,self.rect)

class Heart():
    delta = {
        pg.K_UP: (0, -1),
        pg.K_DOWN: (0, +1),
        pg.K_LEFT: (-1, 0),
        pg.K_RIGHT: (+1, 0),
    }

    def __init__(self, command:CommandBoxManager, enemy_box:EnemyBoxManager, turn:TurnMa
        self.commannd = command
        self.box = enemy_box
        self.turn = turn
        self.image = pg.transform.rotozoom(pg.image.load(f"photo/heart.png"), 0, 0.05)
        self.rect = self.image.get_rect()
        self.rect.centerx = self.box.enemybox_x + self.box.enemybox_width // 2
        self.rect.centery = self.box.enemybox_y + self.box.enemybox_height // 2
        self.speed = 6
        self.state = "alive"

    def update(self, key_lst:list[bool], screen: pg.Surface):
        sum_mv = [0, 0]
        if self.commannd.hp <= 0:
            self.state = "death"
        for k, mv in __class__.delta.items():
            if key_lst[k]:
                sum_mv[0] += mv[0]
                sum_mv[1] += mv[1]
        if self.state is "alive":
            self.rect.move_ip(self.speed*sum_mv[0], self.speed*sum_mv[1])
            if (self.rect.left < self.box.enemybox_x or self.rect.right > self.box.enemy
                self.rect.move_ip(-self.speed*sum_mv[0], -self.speed*sum_mv[1])
            if self.rect.left < self.box.enemybox_x or self.rect.right > self.box.enemyb
                self.rect.move_ip(-self.speed*sum_mv[0], 0)
            if self.rect.top < self.box.enemybox_y or self.rect.bottom > self.box.enemyb
                self.rect.move_ip(0, -self.speed*sum_mv[1])

        if self.turn.turn == "enemy":
            screen.blit(self.image, self.rect)

class Bomb():
    kk_imgs = [pg.image.load(f"photo/{i}.png") for i in range(10)]

    def __init__(self, enemy: Enemy, enemy_box: EnemyBoxManager, turn: TurnManager):
        self.enemy = enemy
        self.turn = turn
        self.box = enemy_box
        self.tmr = 0
        self.image = pg.transform.rotozoom(random.choice(__class__.kk_imgs), 0, 0.6)
        self.rect = self.image.get_rect()
        self.rect.center = random.randint(self.box.enemybox_x + 10, self.box.enemybox_x
        self.vx, self.vy = 0, 1
        self.interval = random.randint(0, 10)
        self.life = 600

    @classmethod
    def generate_bombs(cls, count: int, enemy:Enemy, box:EnemyBoxManager, turn:TurnManag
        bombs = []
        base_interval = 0
        for _ in range(count):
            bomb = Bomb(enemy, box, turn)

```

```python
            # ターン数に応じて最大インターバル値を短縮
            max_interval = max(20, 60 - turn.num * 5)  # 最大60 → 最小20秒まで
            interval = base_interval + random.randint(15, max_interval)

            bomb.interval = interval
            base_interval = interval
            bombs.append(bomb)
        return bombs

    def update(self, screen: pg.Surface):
        if self.turn.turn == "enemy":
            self.tmr += 1
            if self.tmr > self.interval:
                if self.life > 0:
                    screen.blit(self.image, self.rect)
                    self.rect.move_ip(self.vx, self.vy)
                    self.life -= 1

def main():
    """
    メインゲームループ
    """
    screen = pg.display.set_mode((WIDTH, HEIGHT))
    pg.display.set_caption("コマンド選択画面 + HPバー + HP表示")
    clock = pg.time.Clock()
    pg.mouse.set_visible(False)

    turn = TurnManager()
    player = Player(50, 5, turn)
    command_manager = CommandBoxManager(player, turn, font)
    enemy_manager = EnemyBoxManager(player, turn, command_manager, font)
    enemy = Enemy(50, 3, command_manager, turn)
    heart = Heart(command_manager, enemy_manager, turn)
    bombs_num = 15 + turn.num * 5
    bombs = Bomb.generate_bombs(bombs_num, enemy, enemy_manager, turn)
    selected_index = 0
    show_comment = False
    tmr = 0
    cnt = None
    item_menu = ItemMenu(items, font, small_font)

    while True:
        screen.fill(BLACK)
        key_lst = pg.key.get_pressed()
        for event in pg.event.get():
            if event.type == pg.QUIT:
                return
            elif event.type == pg.K_q: #Qキーで終了
                return
            elif event.type == pg.KEYDOWN:
                if item_menu.is_open:
                    if event.key == pg.K_ESCAPE:
                        item_menu.is_open = False
                    elif event.key == pg.K_DOWN:
                        item_menu.selected_index = (item_menu.selected_index + 1) % len(
                    elif event.key == pg.K_UP:
                        item_menu.selected_index = (item_menu.selected_index - 1) % len(
                    elif event.key == pg.K_RETURN:
                        # アイテム使用時のHP回復処理
                        if command_manager.hp < command_manager.former_hp:
                            command_manager.hp += 10
                            if command_manager.hp > command_manager.former_hp:
                                command_manager.hp = command_manager.former_hp
                        item_menu.is_open = False
                else:
                    if event.key == pg.K_RIGHT:
                        selected_index = (selected_index + 1) % len(CommandBoxManager.co
                    elif event.key == pg.K_LEFT:
                        selected_index = (selected_index - 1) % len(CommandBoxManager.co
                    elif event.key == pg.K_RETURN:
                        if CommandBoxManager.commands[selected_index] == "アイテム":
                            item_menu.is_open = True
                            item_menu.selected_index = 0
                        else: #いったんエンターキーでターンチェンジ
                            if turn.turn == "player":
                                if show_comment == False:
                                    show_comment = True
                                else:
                                    show_comment = False
                                    turn.turn_change()
                            else:
                                turn.turn_change()

        if command_manager.hp == 0 and cnt is None:
            cnt = tmr

        if cnt is not None and tmr - cnt >= 60:
            turn.turn = "player"
            show_comment = True
            if event.type == pg.KEYDOWN:
                if event.key == pg.K_RETURN:
                    return

        if show_comment == True:
            enemy_manager.comments(screen)

        for bomb in bombs:
            if bomb.rect.bottom >= enemy_manager.enemybox_y + enemy_manager.enemybox_hei
                bombs.remove(bomb)
            elif heart.rect.colliderect(bomb.rect):
                bombs.remove(bomb)
                command_manager.hp -= enemy.atk
            else:
                bomb.update(screen)

        if command_manager.hp > 0:
            command_manager.draw(screen, selected_index)

        enemy_manager.drawbox(screen)
        enemy.update(screen)
        command_manager.update(screen) # HPバーの描画と更新
        heart.update(key_lst, screen)

        if turn.turn == "enemy" and all(bomb.life <= 0 for bomb in bombs):
            turn.turn_change()
            bombs_num = 15 + turn.num * 5
            bombs = Bomb.generate_bombs(bombs_num, enemy, enemy_manager, turn)

        if item_menu.is_open:
            item_menu.draw(screen)

        pg.display.update()
        clock.tick(60)
        tmr += 1

if __name__ == "__main__":
    WIDTH, HEIGHT = 1920, 1080

    WHITE = (255, 255, 255)
    BLACK = (0, 0, 0)
    YELLOW = (255, 255, 0)

    items = [
        "こうかとんのから揚げ",
        "こうかとんのつくね",
        "こうかとんのぼんじり",
        "こうかとんのもも串",
        "こうかとんの皮串",
        "こうかとんだったもの",
    ]

    pg.init()
    main()
    pg.quit()
    sys.exit()
```