c0a24008ec 実装経過1 d31fc3a · 8 minutes ago

205 lines (165 loc) · 6.3 KB

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

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
font = pg.font.SysFont("meiryo", 50)
small_font = pg.font.SysFont("meiryo", 36)


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

    def __init__(
        self,
        commands: List[str],
        box_width: int,
        box_height: int,
        box_y: int,
        font: pg.font.Font,
    ) -> None:
        self.commands = commands
        self.box_width = box_width
        self.box_height = box_height
        self.box_y = box_y
        self.font = font

    def get_command_boxes(self) -> List[pg.Rect]:
```

```
37              コマンドボックスのpygame.Rectリストを生成する。
38              """
39              spacing = 40
40              total_width = len(self.commands) * self.box_width + (len(self.commands) - 1) * s
41              start_x = (WIDTH - total_width) // 2
42              boxes = []
43              for i in range(len(self.commands)):
44                  x = start_x + i * (self.box_width + spacing)
45                  boxes.append(pg.Rect(x, self.box_y, self.box_width, self.box_height))
46              return boxes
47
48          def draw(self, screen: pg.Surface, selected_index: int) -> None:
49              """
50              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。
51              """
52              boxes = self.get_command_boxes()
53              for i, rect in enumerate(boxes):
54                  color = YELLOW if i == selected_index else WHITE
55                  pg.draw.rect(screen, color, rect, 4)
56
57                  text = self.font.render(self.commands[i], True, WHITE)
58                  text_x = rect.x + (rect.width - text.get_width()) // 2
59                  text_y = rect.y + (rect.height - text.get_height()) // 2
60                  screen.blit(text, (text_x, text_y))
61
62
63  class Enemy():
64      """
65      敵に関するクラス
66      """
67      img0 = pg.image.load(f"photo/enemy1_bob_v2.gif")
68      img = pg.transform.rotozoom(img0,0,0.3)
69      def __init__(self):
70          self.image = __class__.img
71          self.rect = self.image.get_rect()
72          self.rect.centerx = WIDTH // 2
73          self.rect.centery = 300
74          self.tmr = 0
75
76      def update(self, screen: pg.Surface):
77          self.tmr += 1
78          if self.tmr >= 20:
79              if self.tmr % 80 == 20:
80                  self.rect.centerx += 20
81                  self.rect.centery += 10
82              elif self.tmr % 80 == 40:
83                  self.rect.centerx -= 20
84                  self.rect.centery -= 10
85              elif self.tmr % 80 == 60:
86                  self.rect.centerx -= 20
87                  self.rect.centery += 10
88              elif self.tmr % 80 == 0:
```

```
                    self.rect.centerx += 20
                    self.rect.centery -= 10
                screen.blit(self.image,self.rect)

    # class enemy_turn():
    #     delta = {  # 押下キーと移動量の辞書
    #         pg.K_UP: (0, -3),
    #         pg.K_DOWN: (0, +3),
    #         pg.K_LEFT: (-1, 3),
    #         pg.K_RIGHT: (+1, 3),
    #     }
    #     box_img = pg.Surface((30, 30))
    #     pg.draw.rect(box_img, (255, 255, 255), (10, 10), 10)
    #     box_img.set_colorkey((0, 0, 0))

    #     def __init__(self):

    #     def update():
    #         # 例: 画面下部に配置
    #         text_box_rect = pg.Rect(400, HEIGHT - 500, WIDTH - 800, 150)
    #         pg.draw.rect(screen, BLACK, text_box_rect)
    #         pg.draw.rect(screen, WHITE, text_box_rect, 4)


    # class enemy_atk():
    #     imgs = [pg.transform.rotozoom(f"photo/{i}.png",0,0.5) for i in range(0, 10)]

    #     def __init__(self, hp: int,atk: int):
    #         self.image = pg.transform.rotozoom(random.choice(__class__.imgs), 0, 0.8)
    #         self.rect = self.image.get_rect()

    # class TurnManager():
    #     def __init__():
    #         self.num = 0


    def main() -> None:
        """
        メインゲームループ
        """
        global screen, clock
        emy = Enemy()
        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)

        max_hp = 50
        current_hp = 50

        commands = ["こうげき", "アクション", "アイテム", "にげる"]
        selected_index = 0
```

enemy ProjExD_Group04 / KOKATAIL.py ↑ Top

Code Blame Raw

```
63    class Enemy():
76        def update(self, screen: pg.Surface):
148       hp_bar_margin_top = 10
149
150       command_manager = CommandBoxManager(commands, box_width, box_height, box_y, font)
151
152       running = True
153       while running:
154           for event in pg.event.get():
155               if event.type == pg.QUIT:
156                   running = False
157
158               elif event.type == pg.KEYDOWN:
159                   if event.key == pg.K_RIGHT:
160                       selected_index = (selected_index + 1) % len(commands)
161                   elif event.key == pg.K_LEFT:
162                       selected_index = (selected_index - 1) % len(commands)
163                   elif event.key == pg.K_RETURN:
164                       # アイテムメニューなどはなし。ここに処理を書きたい場合は追記
165                       pass
166
167           screen.fill(BLACK)
168
169           command_manager.draw(screen, selected_index)
170
171           boxes = command_manager.get_command_boxes()
172           center_x = (boxes[1].centerx + boxes[2].centerx) // 2
173           hp_bar_y = box_y + box_height + hp_bar_margin_top
174
175           # HPバー背景（黒）
176           pg.draw.rect(screen, BLACK, (center_x - hp_bar_width // 2, hp_bar_y, hp_bar_widt
177           # HPバー黄色部分（HPの割合に応じた幅）
178           hp_ratio = current_hp / max_hp
179           pg.draw.rect(screen, YELLOW, (center_x - hp_bar_width // 2, hp_bar_y, int(hp_bar
180           # HPバー枠（白）
181           pg.draw.rect(screen, WHITE, (center_x - hp_bar_width // 2, hp_bar_y, hp_bar_widt
182
183           # HPバー横にHP数値表示
184           hp_text = font.render(f"{current_hp} / {max_hp}", True, WHITE)
185           text_x = center_x - hp_bar_width // 2 + hp_bar_width + 10
186           text_y = hp_bar_y + (hp_bar_height - hp_text.get_height()) // 2
187           screen.blit(hp_text, (text_x, text_y))
188
189           emy.update(screen)
190
191           pg.display.update()
192           clock.tick(60)
```

```
193
194         pg.quit()
195         sys.exit()
196
197
198     if __name__ == "__main__":
199         WIDTH, HEIGHT = 1920, 1080
200
201         WHITE = (255, 255, 255)
202         BLACK = (0, 0, 0)
203         YELLOW = (255, 255, 0)
204
205         main()
```