



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



```
1  import os
2  import pygame as pg
3  import sys
4  from typing import List
5  import random
6
7  WHITE = (255, 255, 255)
8  BLACK = (0, 0, 0)
9  YELLOW = (255, 255, 0)
10
11  os.chdir(os.path.dirname(os.path.abspath(__file__)))
12
13  pg.init() # pygameの初期化は必ず最初に行う
14
15  # 初期化後にフォント作成
16  WIDTH, HEIGHT = 1920, 1080
17  font = pg.font.SysFont("meiryō", 50)
18  small_font = pg.font.SysFont("meiryō", 36)
19
20  class TurnManager():
21      def __init__(self):
22          self.num = 1
23          self.turn = "player"
24
25      def turn_change(self):
26          if self.turn == "player":
27              self.turn = "enemy"
28          elif self.turn == "enemy":
29              self.turn = "player"
30              self.num += 1
31
32  class Player():
33      def __init__(self, HP:int, ATK:int, turn:TurnManager):
34          self.former_hp = HP
35          self.hp = self.former_hp
36          self.former_atk = ATK
```

```
37         self.atk = self.former_atk
38         self.turn = turn
39
40     ✓ class CommandBoxManager():
41         """
42         コマンドボックスの位置計算と描画管理クラス
43         """
44
45         commands = ["こうげき", "アクション", "アイテム", "にげる"]
46
47     ✓ def __init__(self, player:Player, turn:TurnManager, font: pg.font.Font) -> None:
48         self.box_width = 265
49         self.box_height = 80
50         self.box_y = HEIGHT - 300
51         self.hp_bar_width = 160
52         self.hp_bar_height = 20
53         self.hp_bar_margin_top = 10
54         self.hp_bar_y = self.box_y + self.box_height + self.hp_bar_margin_top + 15
55         self.commands = __class__.commands
56         self.font = font
57         self.former_hp = player.former_hp
58         self.hp = player.hp
59         self.turn = turn
60
61     ✓ def get_command_boxes(self) -> List[pg.Rect]:
62         """
63         コマンドボックスのpygame.Rectリストを生成する。
64         """
65         self.spacing = 40
66         self.total_width = len(self.commands) * self.box_width + (len(self.commands) - 1) * self.spacing
67         self.start_x = (WIDTH - self.total_width) // 2
68         self.bboxes = []
69         for i in range(len(self.commands)):
70             x = self.start_x + i * (self.box_width + self.spacing)
71             self.bboxes.append(pg.Rect(x, self.box_y, self.box_width, self.box_height))
72         return self.bboxes
73
74     ✓ def draw(self, screen: pg.Surface, selected_index: int) -> None:
75         """
76         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。
77         """
78         self.bboxes = self.get_command_boxes()
79         if self.turn.turn == "player":
80             for i, rect in enumerate(self.bboxes):
81                 self.color = YELLOW if i == selected_index else WHITE
82                 pg.draw.rect(screen, self.color, rect, 4)
83
84                 self.command_text = self.font.render(self.commands[i], True, WHITE)
85                 self.command_text_x = rect.x + (rect.width - self.command_text.get_width()) // 2
86                 self.command_text_y = rect.y + (rect.height - self.command_text.get_height()) // 2
87                 screen.blit(self.command_text, (self.command_text_x, self.command_text_y))
88
```

```
89  ✓ def update(self, screen:pg.Surface):
90      self.bboxes = self.get_command_bboxes()
91      self.center_x = (self.bboxes[1].centerx + self.bboxes[2].centerx) // 2
92      if self.hp <= 0:
93          self.hp = 0
94      self.hp_ratio = self.hp / self.former_hp
95      self.hp_text = font.render(f"{self.hp} / {self.former_hp}", True, WHITE)
96      self.command_text_x = self.center_x - self.hp_bar_width // 2 + self.hp_bar_width
97      self.command_text_y = self.hp_bar_y + 5 + (self.hp_bar_height - self.hp_text.get_height() // 2)
98      # HPバー背景 (黒)
99      pg.draw.rect(screen, BLACK, (self.center_x - self.hp_bar_width // 2, self.hp_bar_y, self.hp_bar_width, self.hp_bar_height))
100     # HPバー黄色部分 (HPの割合に応じた幅)
101     pg.draw.rect(screen, YELLOW, (self.center_x - self.hp_bar_width // 2, self.hp_bar_y, self.hp_bar_width * self.hp_ratio, self.hp_bar_height))
102     # HPバー枠 (白)
103     pg.draw.rect(screen, WHITE, (self.center_x - self.hp_bar_width // 2, self.hp_bar_y, self.hp_bar_width, self.hp_bar_height))
104     # HPバー横にHP数値表示
105     screen.blit(self.hp_text, (self.command_text_x, self.command_text_y))
106
107  ✓ class EnemyBoxManager():
108  ✓     def __init__(self, player:Player, turn:TurnManager, command:CommandBoxManager, font:Font):
109         self.player = player
110         self.turn = turn
111         self.command = command
112         self.font = font
113         self.box_width = 1180
114         self.box_height = 280
115         self.enemybox_width = 280
116         self.enemybox_height = 280
117         self.start_x = (WIDTH - self.box_width) // 2
118         self.start_y = self.command.box_y - (self.box_height + 20)
119         self.comments_list = ["まずは練習だ",
120                                "だんだん難しくなっていくぜ?",
121                                "うまく避けろよ",
122                                "どこまで耐えられるかな?",
123                                "そろそろ本気だぜ?",
124                                "まだいけるか?",
125                                "そろそろ限界かな?",
126                                "おまえ強いな",
127                                "なんで生きてるの?",
128                                "疲れたな",
129                                "もはや怖いぞ?",
130                                "もう疲れたよ",
131                                "はよ〇ね",
132                                "." * self.turn.num
133                            ]
134         self.comment = self.comments_list[0] # 初期コメント
135         self.death_comment = "\"GameOver\" だな"
136         self.enemy_text = self.font.render(self.comment, True, WHITE)
137         self.enemy_text_x = self.start_x + 30
138         self.enemy_text_y = self.start_y + 30
139         self.enemybox_x = (WIDTH - self.enemybox_width) // 2
140         self.enemybox_y = self.command.box_y - (self.enemybox_height + 20)
```

```
141
142 ✓ def drawbox(self, screen:pg.Surface):
143     if self.turn.turn == "player":
144         pg.draw.rect(screen, WHITE, (self.start_x, self.start_y, self.box_width, sel
145     elif self.turn.turn == "enemy":
146         pg.draw.rect(screen, WHITE, (self.enemybox_x, self.enemybox_y, self.enemybox
147
148 ✓ def comments(self, screen: pg.Surface):
149     if self.command.hp > 0:
150         if self.turn.num < len(self.comments_list):
151             self.comment = self.comments_list[self.turn.num]
152         else:
153             self.comment = self.comments_list[-1]
154         self.enemy_text = self.font.render(self.comment, True, WHITE)
155         screen.blit(self.enemy_text, (self.enemy_text_x, self.enemy_text_y))
156     if self.command.hp == 0:
157         self.comment = self.death_comment
158         self.enemy_text = self.font.render(self.comment, True, WHITE)
159         screen.blit(self.enemy_text, (self.enemy_text_x, self.enemy_text_y))
160
161
162 ✓ class ItemMenu:
163     """
164     アイテムメニューの描画と操作管理クラス
165     """
166
167 ✓ def __init__(self, items: List[str], font: pg.font.Font, small_font: pg.font.Font)
168     self.items = items
169     self.font = font
170     self.small_font = small_font
171     self.selected_index = 0
172     self.is_open = False
173
174     self.menu_width = 800
175     self.menu_height = 400
176     self.menu_x = (WIDTH - self.menu_width) // 2
177     self.menu_y = (HEIGHT - self.menu_height) // 2
178     self.menu_rect = pg.Rect(self.menu_x, self.menu_y, self.menu_width, self.menu_he
179
180 ✓ def draw(self, screen: pg.Surface) -> None:
181     """
182     アイテムメニューを画面に描画する。
183     """
184     # 背景と枠
185     pg.draw.rect(screen, BLACK, self.menu_rect)
186     pg.draw.rect(screen, WHITE, self.menu_rect, 3)
187
188     # タイトル
189     title = self.font.render("アイテム", True, WHITE)
190     screen.blit(title, (self.menu_x + 20, self.menu_y + 10))
191
192     # アイテムリスト表示
```

```
193         for i, item in enumerate(self.items):
194             color = YELLOW if i == self.selected_index else WHITE
195             item_text = self.small_font.render(f"- {item}", True, color)
196             screen.blit(item_text, (self.menu_x + 40, self.menu_y + 70 + i * 50))
197
198
199     class Enemy():
200         """
201         敵に関するクラス
202         """
203         img0 = pg.image.load(f"photo/enemy1_bob_v2.gif")
204         img = pg.transform.rotozoom(img0, 0, 0.3)
205     def __init__(self, HP:int, ATK:int, command:CommandBoxManager, turn:TurnManager):
206         self.hp = HP
207         self.atk = ATK
208         self.image = __class__.img
209         self.rect = self.image.get_rect()
210         self.rect.centerx = WIDTH // 2
211         self.rect.centery = 300
212         self.command = command
213         self.turn = turn
214         self.tmr = 0
215
216     def update(self, screen: pg.Surface):
217         self.tmr += 1
218         if self.tmr >= 20 and self.command.hp > 0:
219             if self.tmr % 80 == 20:
220                 self.rect.centerx += 20
221                 self.rect.centery += 10
222             elif self.tmr % 80 == 40:
223                 self.rect.centerx -= 20
224                 self.rect.centery -= 10
225             elif self.tmr % 80 == 60:
226                 self.rect.centerx -= 20
227                 self.rect.centery += 10
228             elif self.tmr % 80 == 80:
229                 self.rect.centerx += 20
230                 self.rect.centery -= 10
231             elif self.command.hp == 0:
232                 self.rect.centerx = WIDTH // 2
233                 self.rect.centery = 300
234             screen.blit(self.image, self.rect)
235
236     class Heart():
237     def __init__(self, delta = {
238         pg.K_UP: (0, -1),
239         pg.K_DOWN: (0, +1),
240         pg.K_LEFT: (-1, 0),
241         pg.K_RIGHT: (+1, 0),
242     }):
243
244     def __init__(self, command:CommandBoxManager, enemy_box:EnemyBoxManager, turn:TurnMa
```

```

245         self.commannd = command
246         self.box = enemy_box
247         self.turn = turn
248         self.image = pg.transform.rotozoom(pg.image.load(f"photo/heart.png"), 0, 0.05)
249         self.rect = self.image.get_rect()
250         self.rect.centerx = self.box.enemybox_x + self.box.enemybox_width // 2
251         self.rect.centery = self.box.enemybox_y + self.box.enemybox_height // 2
252         self.speed = 6
253         self.state = "alive"
254
255     def update(self, key_lst:list[bool], screen: pg.Surface):
256         sum_mv = [0, 0]
257         if self.commannd.hp <= 0:
258             self.state = "death"
259         for k, mv in __class__.delta.items():
260             if key_lst[k]:
261                 sum_mv[0] += mv[0]
262                 sum_mv[1] += mv[1]
263         if self.state is "alive":
264             self.rect.move_ip(self.speed*sum_mv[0], self.speed*sum_mv[1])
265             if (self.rect.left < self.box.enemybox_x or self.rect.right > self.box.enemy
266                 self.rect.move_ip(-self.speed*sum_mv[0], -self.speed*sum_mv[1])
267             if self.rect.left < self.box.enemybox_x or self.rect.right > self.box.enemyt
268                 self.rect.move_ip(-self.speed*sum_mv[0], 0)
269             if self.rect.top < self.box.enemybox_y or self.rect.bottom > self.box.enemyt
270                 self.rect.move_ip(0, -self.speed*sum_mv[1])
271
272         if self.turn.turn == "enemy":
273             screen.blit(self.image, self.rect)
274
275     class Bomb():
276         kk_imgs = [pg.image.load(f"photo/{i}.png") for i in range(10)]
277
278     def __init__(self, enemy: Enemy, enemy_box: EnemyBoxManager, turn: TurnManager):
279         self.enemy = enemy
280         self.turn = turn
281         self.box = enemy_box
282         self.tmr = 0
283         self.image = pg.transform.rotozoom(random.choice(__class__.kk_imgs), 0, 0.6)
284         self.rect = self.image.get_rect()
285         self.rect.center = random.randint(self.box.enemybox_x + 10, self.box.enemybox_x
286         self.vx, self.vy = 0, 1
287         self.interval = random.randint(0, 10)
288         self.life = 600
289
290     @classmethod
291     def generate_bombs(cls, count: int, enemy: Enemy, box: EnemyBoxManager, turn: TurnManag
292         bombs = []
293         base_interval = 0
294         for _ in range(count):
295             bomb = Bomb(enemy, box, turn)
296

```

```
297         # ターン数に応じて最大インターバル値を短縮
298         max_interval = max(20, 60 - turn.num * 5) # 最大60 → 最小20秒まで
299         interval = base_interval + random.randint(15, max_interval)
300
301         bomb.interval = interval
302         base_interval = interval
303         bombs.append(bomb)
304     return bombs
305
306     def update(self, screen: pg.Surface):
307         if self.turn.turn == "enemy":
308             self.tmr += 1
309             if self.tmr > self.interval:
310                 if self.life > 0:
311                     screen.blit(self.image, self.rect)
312                     self.rect.move_ip(self.vx, self.vy)
313                     self.life -= 1
314
315     def main():
316         """
317         メインゲームループ
318         """
319         screen = pg.display.set_mode((WIDTH, HEIGHT))
320         pg.display.set_caption("コマンド選択画面 + HPバー + HP表示")
321         clock = pg.time.Clock()
322         pg.mouse.set_visible(False)
323
324         turn = TurnManager()
325         player = Player(50, 5, turn)
326         command_manager = CommandBoxManager(player, turn, font)
327         enemy_manager = EnemyBoxManager(player, turn, command_manager, font)
328         enemy = Enemy(50, 3, command_manager, turn)
329         heart = Heart(command_manager, enemy_manager, turn)
330         bombs_num = 15 + turn.num * 5
331         bombs = Bomb.generate_bombs(bombs_num, enemy, enemy_manager, turn)
332         selected_index = 0
333         show_comment = False
334         tmr = 0
335         cnt = None
336         item_menu = ItemMenu(items, font, small_font)
337
338         while True:
339             screen.fill(BLACK)
340             key_lst = pg.key.get_pressed()
341             for event in pg.event.get():
342                 if event.type == pg.QUIT:
343                     return
344                 elif event.type == pg.K_q: #Qキーで終了
345                     return
346                 elif event.type == pg.KEYDOWN:
347                     if item_menu.is_open:
348                         if event.key == pg.K_ESCAPE:
```

```

349         item_menu.is_open = False
350     elif event.key == pg.K_DOWN:
351         item_menu.selected_index = (item_menu.selected_index + 1) % len(
352     elif event.key == pg.K_UP:
353         item_menu.selected_index = (item_menu.selected_index - 1) % len(
354     elif event.key == pg.K_RETURN:
355         # アイテム使用時のHP回復処理
356         if command_manager.hp < command_manager.former_hp:
357             command_manager.hp += 10
358             if command_manager.hp > command_manager.former_hp:
359                 command_manager.hp = command_manager.former_hp
360         item_menu.is_open = False
361     else:
362         if event.key == pg.K_RIGHT:
363             selected_index = (selected_index + 1) % len(CommandBoxManager.co
364     elif event.key == pg.K_LEFT:
365             selected_index = (selected_index - 1) % len(CommandBoxManager.co
366     elif event.key == pg.K_RETURN:
367         if CommandBoxManager.commands[selected_index] == "アイテム":
368             item_menu.is_open = True
369             item_menu.selected_index = 0
370         else: #いったんエンターキーでターンチェンジ
371             if turn.turn == "player":
372                 if show_comment == False:
373                     show_comment = True
374                 else:
375                     show_comment = False
376                     turn.turn_change()
377             else:
378                 turn.turn_change()
379
380 if command_manager.hp == 0 and cnt is None:
381     cnt = tmr
382
383 if cnt is not None and tmr - cnt >= 60:
384     turn.turn = "player"
385     show_comment = True
386     if event.type == pg.KEYDOWN:
387         if event.key == pg.K_RETURN:
388             return
389
390 if show_comment == True:
391     enemy_manager.comments(screen)
392
393 for bomb in bombs:
394     if bomb.rect.bottom >= enemy_manager.enemybox_y + enemy_manager.enemybox_hei
395         bombs.remove(bomb)
396     elif heart.rect.colliderect(bomb.rect):
397         bombs.remove(bomb)
398         command_manager.hp -= enemy.atk
399     else:
400         bomb.update(screen)

```



```
401
402     if command_manager.hp > 0:
403         command_manager.draw(screen, selected_index)
404
405     enemy_manager.drawbox(screen)
406     enemy.update(screen)
407     command_manager.update(screen) # HPバーの描画と更新
408     heart.update(key_lst, screen)
409
410     if turn.turn == "enemy" and all(bomb.life <= 0 for bomb in bombs):
411         turn.turn_change()
412         bombs_num = 15 + turn.num * 5
413         bombs = Bomb.generate_bombs(bombs_num, enemy, enemy_manager, turn)
414
415     if item_menu.is_open:
416         item_menu.draw(screen)
417
418     pg.display.update()
419     clock.tick(60)
420     tmr += 1
421
422 if __name__ == "__main__":
423     WIDTH, HEIGHT = 1920, 1080
424
425     WHITE = (255, 255, 255)
426     BLACK = (0, 0, 0)
427     YELLOW = (255, 255, 0)
428
429     items = [
430         "こうかとんのから揚げ",
431         "こうかとんのつくね",
432         "こうかとんのぼんじり",
433         "こうかとんのもも串",
434         "こうかとんの皮串",
435         "こうかとんだったもの",
436     ]
437
438     pg.init()
439     main()
440     pg.quit()
441     sys.exit()
```