

PokerThrone

一个RPG卡牌构筑冒险游戏的开发框架

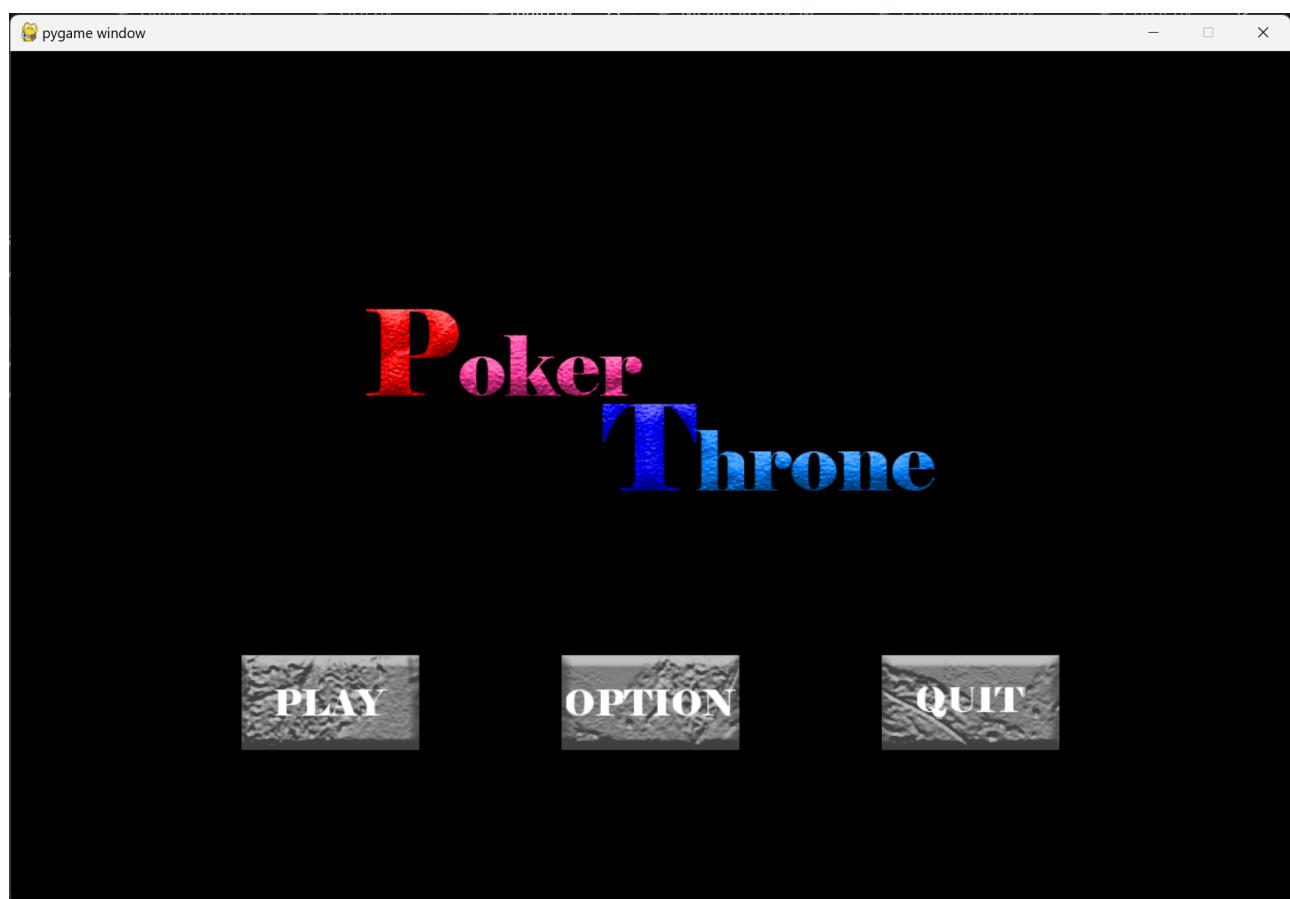
当前进度：已经完成游戏框架的整体开发，并完成一个初步的Demo

游戏介绍：

世界上出现了大量的怪物，于是你要干掉他们

游戏流程与介绍

游戏主菜单

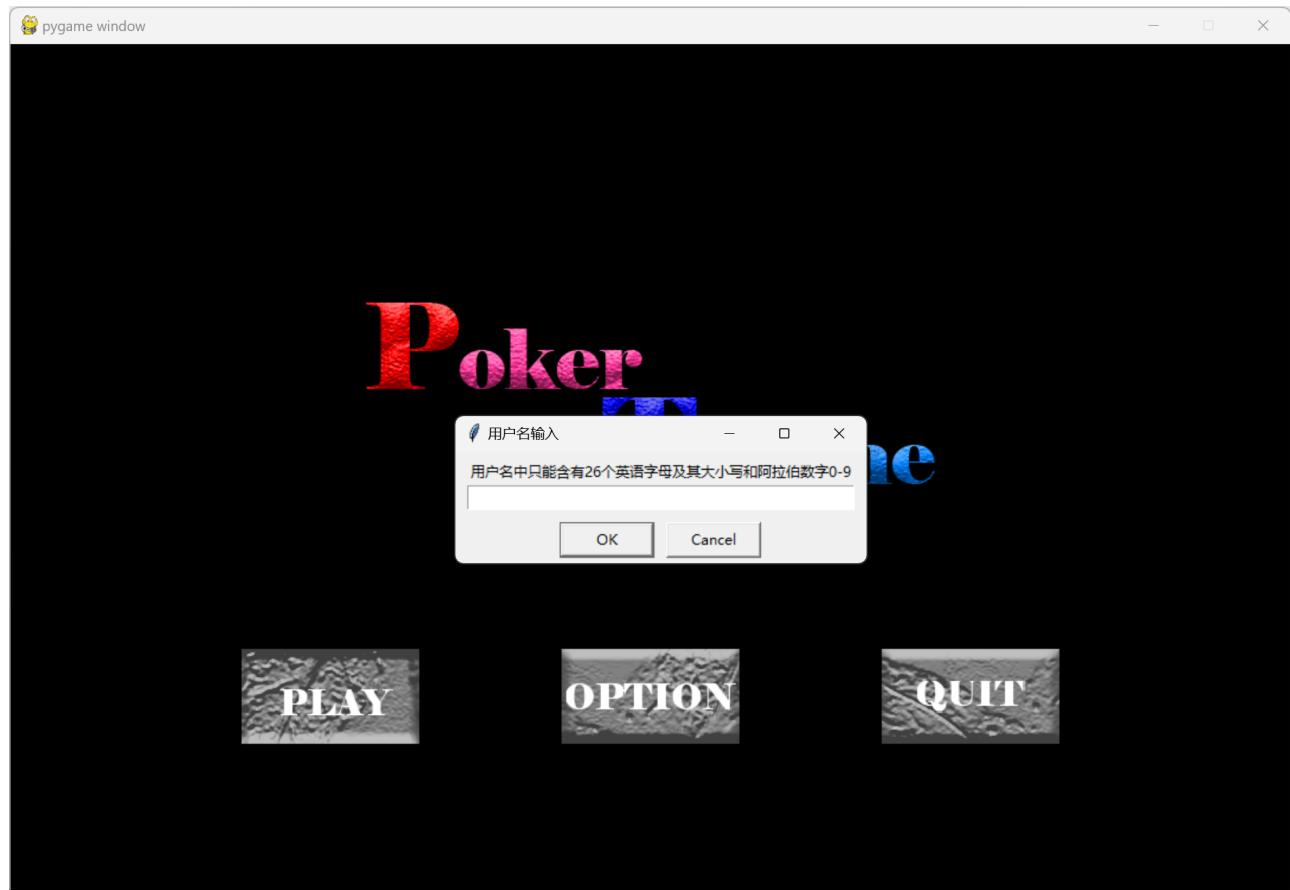


PLAY：开始游戏

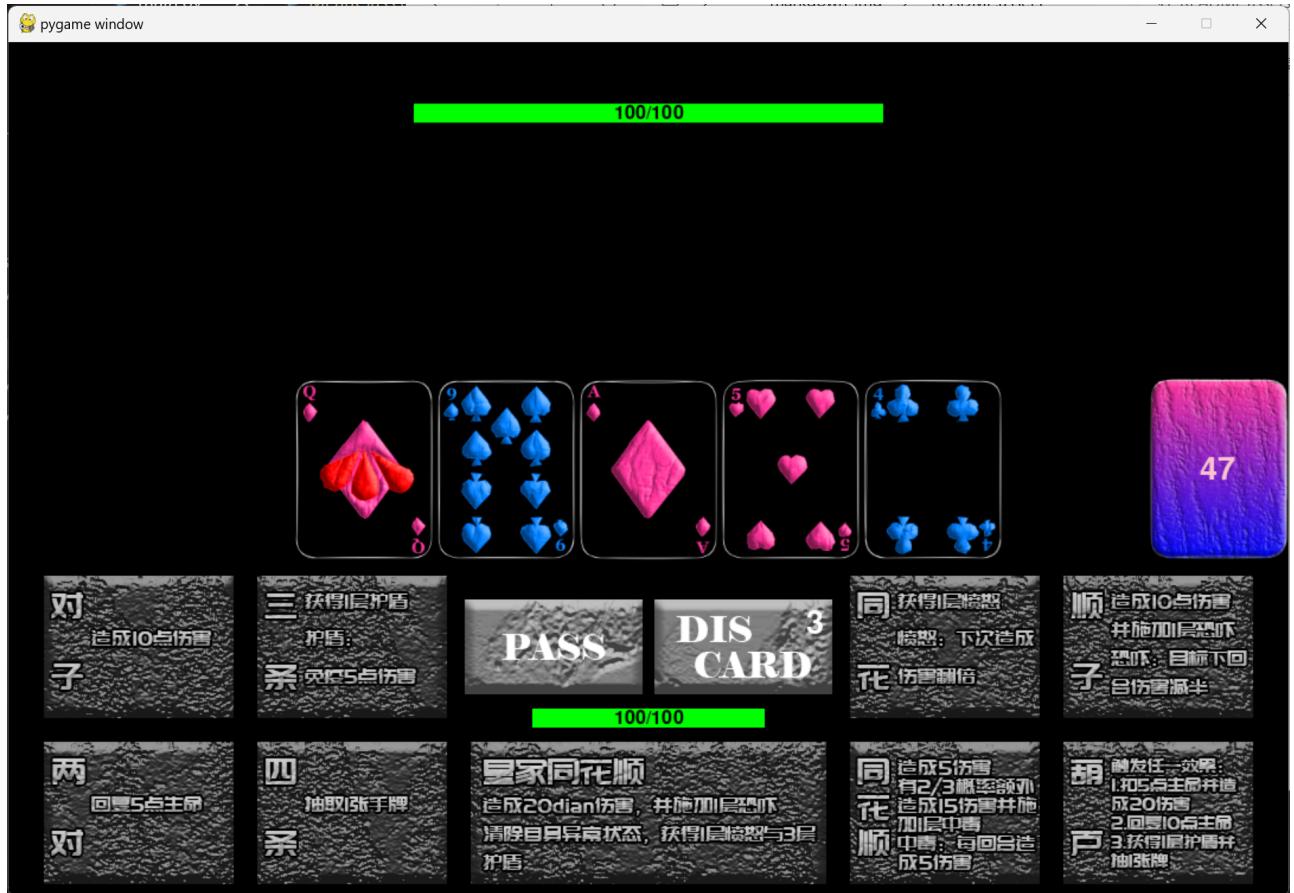
OPTION: (未开发)

QUIT: 退出游戏

为自己取一个独一无二的用户名吧! (后续排行榜开发, 该功能未美化)



正式开始游戏



本游戏的核心玩法是配凑牌组来匹配技能，之后通过技能来攻击怪物（暂无立绘，当前只有一个血条）

当手中的五张牌满足牌型后，就释放技能

discard按钮可以重抽选中的牌：



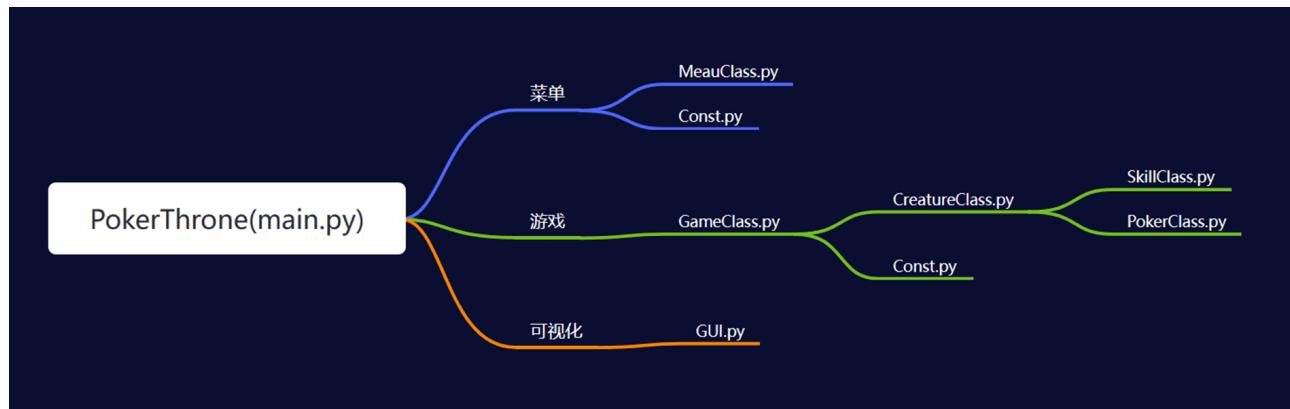
但要注意的是，discard的次数有限（在右上角显示）



如果无法重抽，就只能选择过牌或者释放技能

左侧为弃牌堆，右侧为牌堆，（点开查看暂未开发）

后续开发指南



添加新的英雄，请从`Hero`类继承，替换`hp`, `mp`, `skill_set`即可

添加新的怪物，请从`Monster`类继承，替换`hp`, `mp`, `skill_set`即可

添加新的技能，请从`Skill`类继承，替换`activate`方法即可

作业要求响应

综合运用Python内置对象、表达式、函数、分支、循环、序列，面向对象编程、字符串、正则表达式、文件IO、异常处理，解决某个问题或实现某个功能。

Python内置对象

```
class Monster(Creature):
    def __init__(self, pos:tuple[int,int]):
        super().__init__()
        self.pos=pos
        self.skill_set = {'Attack':SkillClass.Attack((0,0),0),
                          'Shield':SkillClass.Shield((0,0),0),
                          'Medicine':SkillClass.Medicine((0,0),0)}
        self.state = []          # 存状态token (如SHOCK, RAGE等)
        self.info = ''           # 放角色介绍之类的
```

表达式

```
class Skill(object):
    IMAGE_PATH='source/play.png'
    def __init__(self, pos:tuple[int,int], handtype_check_code:int): ...

    def update(self, handtype_code:list[int]):
        if(self.ui.is_selected()):
            return
        if any(self.handtype_check_code==code for code in handtype_code) :
            self.ui.enabled()
        else :      You, 2 weeks ago • update skills
            self.ui.disabled()
```

函数

```
class ImageSet(object):
    def __init__(self, ...):
    def render(self, surface:pygame.Surface): ...
    def set_pos(self, pos:Tuple[int,int], option=LEFTTOP): ...
    def change_status(self, value:int): ...
```

分支

```

if GameEvent.POISONED in host.state:          # Poisoned 处理
    ret = host.lose_HP(5)
    if ret == GameEvent.LOSE_HP_OVER:
        return 0
    host.state.remove(GameEvent.POISONED)
    host.state.append(GameEvent.POISONED2)
elif GameEvent.POISONED2 in host.state:         # 持续三回合的中毒效果，引入另外两个表示中毒的event
    ret = host.lose_HP(5)
    if ret == GameEvent.LOSE_HP_OVER:
        return 0
    host.state.remove(GameEvent.POISONED2)
    host.state.append(GameEvent.POISONED3)
elif GameEvent.POISONED3 in host.state:
    ret = host.lose_HP(5)
    if ret == GameEvent.LOSE_HP_OVER:
        return 0
    host.state.remove(GameEvent.POISONED3)

```

循环

```

class Game(object):
    def __init__(self, screen: pygame.Surface): ...
    def game_init(self): ...
    def game_run(self): ...
    def game_quit(self): ...
    def game_save(self): ...
    def game_load(self): ...
    def game(self):
        self.game_init()
        self.game_load()
        game_running=1
        while(game_running):
            game_running=self.game_run()
            self.game_save()
        self.game_quit()
        pass

```

序列

```

#英雄技能包
self.skill_set =[SkillClass.Attack((pos[0]-2*w_gap-w_fix,pos[1]+h_gap+h_fix),PokerClass.ONE_PAIR),
                 SkillClass.Shield((pos[0]-w_gap-w_fix,pos[1]+h_gap+h_fix),PokerClass.THREE_AKIND),
                 SkillClass.Medicine((pos[0]-2*w_gap-w_fix,pos[1]+2*h_gap+h_fix),PokerClass.TWO_PAIRS),
                 SkillClass.Getcard((pos[0]-w_gap-w_fix,pos[1]+2*h_gap+h_fix),PokerClass.FOUR_AKIND),
                 SkillClass.Rage((pos[0]+w_gap+w_fix,pos[1]+h_gap+h_fix),PokerClass.FLUSH),
                 SkillClass.Shockwave((pos[0]+2*w_gap+w_fix,pos[1]+h_gap+h_fix),PokerClass.STRAIGHT),
                 SkillClass.Poison((pos[0]+w_gap+w_fix,pos[1]+2*h_gap+h_fix),PokerClass.STRAIGHT_FLUSH),
                 SkillClass.TrickBag((pos[0]+2*w_gap+w_fix,pos[1]+2*h_gap+h_fix),PokerClass.FULL_HOUSE),
                 SkillClass.Ultimate((pos[0],pos[1]+2*h_gap+h_fix),PokerClass.ROYAL_FLUSH)
]

```

面向对象编程

```
You, 2 weeks ago | 1 author (You)
15 > class ImageSet(object): ...
109
You, 2 weeks ago | 1 author (You)
110 > class Button(ImageSet): ...
164
▶ You, 2 weeks ago | 1 author (You)
165 > class Rod(Button): ...
```

字符串，正则表达式

```
# 输入字符串
while window:
    if not wrong_input:
        entry_str = simpledialog.askstring(title='用户名输入',
                                             prompt='用户名中只能含有26个英文字母及其大小写和阿拉伯数字0-9')
    else:
        entry_str = simpledialog.askstring(title='用户名输入',
                                             prompt='格式错误！请重新输入！\n'
                                             '用户名中只能含有26个英文字母及其大小写和阿拉伯数字0-9')
```

文件IO

```
#按照规则读取文件

if mode == PATH_IN:
    if type(image_file)==str :
        if os.path.exists(image_file) :
            self.image = pygame.image.load(image_file)
        else:
            raise Exception('No such file: '+image_file)
    else:
        raise Exception("image_file is not a image path")
elif mode == FILE_IN :...
elif mode == LIST_IN :...
else:
    raise Exception('No such mode')
```

异常处理

```
try:  
    str = re.search(r'[A-Z0-9a-z]+', entry_str)  
except TypeError:  
    return QUIT # Press Cancel  
  
try:  
    str.group() # In case of matching nothing  
except AttributeError:  
    wrong_input = True  
    continue
```