

프로그램 개발 설계서-1

[참고] 교과서 181

1. 프로그램명	원카드
----------	-----

2. 요구사항 분석

<기존 프로그램 분석> 또는 / <현재 상황>	<필요한 주요 기능 및 개선 사항>
<p>실제 카드가 필요해서 시간과 장소 제한</p> <p>특수 카드가 없다는 한계</p> <p>게임 규칙이 단순</p> <p>룰 설명이 없음</p>	<p>장소에 상관없이 게임 가능</p> <p>폭탄 카드로 게임 규칙 추가</p> <p>게임 시간 표시 기능으로 시간 측정 가능</p> <p>룰 설명 기능</p> <p>카드 랜덤으로 나누는 기능</p> <p>승패 메시지 출력 기능</p> <p>잘못 누르면 순서 넘어가는 기능</p>

<입력 설계>	<입력 데이터 예시>
<p>닉네임 입력</p> <p>게임 시작 한다는 입력 기능</p> <p>널 카드 입력기능</p> <p>폭탄 카드 사용 입력 기능</p>	<p>당신의 이름을 입력하세요:</p> <p>게임을 시작하려면 Enter를 누르세요 ▼</p> <p>널 카드 번호를 입력하거나 'q'를 눌러 카드 받기:</p>

<출력 설계>	<출력 데이터 예시>
---------	-------------

<p>룰 출력</p> <p>게임 시작 준비 상태 출력</p> <p>바닥에 놓인 시작 카드 출력</p> <p>봇이 내고 받은 카드 출력</p> <p>현재 플레이어 카드, 받은 카드 출력</p> <p>폭탄 카드 사용 출력</p> <p>승패 메시지 출력</p> <p>게임 시간 출력</p>	<p>📖 게임 규칙 안내 📖</p> <ol style="list-style-type: none"> 1. 플레이어와 컴퓨터는 랜덤 카드 5장씩 받습니다. 2. 테이블에는 랜덤 카드 1장이 놓입니다. 3. 플레이어는 숫자나 무늬가 테이블 카드와 같으면 카드 여러 장을 낼 수 있습니다. 4. 낼 카드가 없으면 'q'를 눌러 카드 1장을 받습니다. 5. 폭탄 카드는 20% 확률로 등장하며, 내면 상대가 카드 2장을 받습니다. <p>게임을 시작하려면 Enter를 누르세요 ▼</p> <p>테이블 카드: 9 ♥ 💣 e님의 카드:</p> <ol style="list-style-type: none"> 1. 1 ♠ 2. 2 ♠ 3. 7 ♥ 4. 8 ♦ 5. 12 ♠ <p>넌 카드: 8 ♥ 💣 💣 봇이 카드 2장을 받았습니다!</p> <p>🤖 봇 차례... 봇이 낸 카드: 7 ♣</p> <p>카드 받음: 4 ♠</p> <p>😞 봇 승리! 다음에 다시 도전하세요! 🕒 총 플레이 시간: 144.94초</p>
--	--

<해결 방법>	<해결 방법 예시>
---------	------------

<p>1 플레이어와 봇에게 각각 5장씩 카드 분배</p> <p>2 시작 카드 한장 바닥에 둬</p> <p>3 낼 수 있는 카드는 바닥 카드랑 숫자나 색이 같은거</p> <p>4 잘못 입력하면 메시지 출력후 차례 넘어감</p> <p>5 승리자 패배자 출력</p>	<p>1 card_num이라는 변수는 1부터 12까지의 숫자를 랜덤으로 추출하고 card_shape라는 변수는 위에 shapes라는 리스트 안에 있는 무늬(모양) 4개 중에서 하나를 추출 card_bomb이라는 변수는 20% 확률로 나눔</p> <p>2 table_num이라는 변수 랜덤 숫자(1~12) table_shape라는 변수 랜덤 모양 table_bomb라는 변수 20% 확률로 폭탄 table_card라는 변수는 위에서 했던 변수 3개를 차례대로 출력</p> <p>3 if choice == "q" 는 만약 낼 카드가 없으면 choice라는 변수에 "q"를 입력 card_num, shape, bomb 각각 1~12, shapes리스트 안에 있는 모양, 20%확률로 폭탄 카드가 나눔 if selected_card[2] == 'bomb': # 낸 카드가 폭탄이면 for _ in range(2): # 봇이 카드 2장 받음 <pre> card_num = random.randint(1, 12) card_shape = random.choice(shapes) card_bomb = 'bomb' if random.random() < 0.2 else " bot_hand.append((card_num, card_shape, card_bomb)) print("💣 봇이 카드 2장을 받았습니다!") </pre> <p>4 card_index = int(choice)를 사용 하여 정수형으로 변환함(왜냐하면 위에서 input)으로 문자로 받았기 때문 if 0 <= card_index < len(player_hand): # 인덱스 번호 0 ~ 낼 수 있는 인덱스 selected_card = player_hand[card_index] # 해당 카드 선택 if selected_card[0] == table_card[0] or selected_card[1] == table_card[1]: 테이블 카드와 내가 고른 카드의 숫자나 모양이 같은지 확인 <pre> table_card = selected_card # 테이블 카드 업데이트 (이유: 플레이어나 봇이 낸 카드가 테이블 카드로 바뀌기 때문 player_hand.pop(card_index) # 낸 카드는 덱에서 제거 else: print("❌ 이 카드는 낼 수 없습니다.") # 규칙 위반 else: print("❌ 이 카드는 낼 수 없습니다.") # 인덱스 범위 초과 위에 조건들에 맞지 않으면 이것들 중 맞지 않은것을 출력 </pre> <p>5 # 플레이어의 손에 카드가 없으면 승리 if len(player_hand) == 0: print(f"\n🎉 {player_name}님 승리! 축하합니다!")</p> </p></p>
---	---

```
🎉")
    game_over = True
    break # 게임 종료
# 봇이 카드 다 냈으면 봇 승리
if len(bot_hand) == 0:
    print("\n😞 봇 승리! 다음에 다시 도전하세요!")
    game_over = True
    break # 게임 종료
```