C++프로그래밍및실습

던전에서 살아남기

진척 보고서 #1

제출일자: 2024/11/17
제출자명: 이재륜
제출자학번: 213235

1. 프로젝트 목표 (16 pt)

1) 배경 및 필요성 (14 pt)

현재 수업 중에 만든 MUD 게임은 텍스트 위주로 구성되어 있어 시각적 재미가 부족하고 캐릭터나 아이템이 단순한 텍스트로만 표시됩니다. 이러한 요소는 게임의 몰입도를 떨어뜨리고 플레이어가 상상에 의존하여 즐기기에는 한계가 있습니다.

따라서 이 프로젝트에서는 ASCII 기반의 아트를 사용하여 캐릭터, 아이템, 몬스터 등을 시각적으로 표현하여 게임에 생동감을 더하여 플레이어가 더 직관적으로 게임에 몰입할 수 있도록 합니다. 또한 다양한 스토리 선택지와 전투 및 아이템 시스템을 통해 기존의 MUD 게임보다 즐길거리가 풍부한 경험을 제공하여 사용자가 게임으로 하여금 재미를 느낄 수 있게 합니다.

2) 프로젝트 목표

ASCII 아트 기반의 시각적 요소와 선택형 스토리 및 난이도 설정 시스템을 통해 플레이어가 몰입감 넘치는 모험을 경험하도록 하는 텍스트 기반 게임을 만드는 것을 목표로 합니다.

3) 차별점

기존 MUD 게임의 단순한 텍스트 기반에서 벗어나 ASCII 아트를 활용하여 시각적 흥미를 추가합니다. 더불어 플레이어가 던전 난이도를 선택할 수 있으며 난이도에 따라 몬스터의 강약이 변화하여 각기 다른 도전의 재미를 제공합니다.

2. 기능 계획

1) 기능 1: 스토리 진행 및 선택 시스템

- 텍스트와 ASCII 아트로 제공되는 스토리를 따라가며 플레이어가 선택지를 통해

모험을 진행하는 기능

(1) 세부 기능 1 (상황 묘사 및 선택지 제공)

- 미궁 속에서 발생하는 다양한 상황을 텍스트와 ASCII 아트를 통해 묘사하고 플레이어가 선택할 수 있는 여러 선택지를 제공합니다.

(2) 세부 기능 2 (선택에 따른 결과 제공)

- 플레이어가 선택한 옵션에 따라 스토리 전개가 달라지며 전투 또는 아이템 발견 등의 이벤트가 발생합니다.

2) 기능 2: 던전 난이도 선택 시스템

- 게임 시작 시 플레이어가 던전의 난이도를 선택할 수 있으며 선택한 난이도에 따라 등장하는 몬스터의 강약이 달라지는 기능

(1) 세부 기능 1 (난이도별 몬스터 설정)

- 쉬움, 보통, 어려움의 난이도를 선택할 수 있으며 난이도가 높아질수록 몬스터의 체력과 공격력이 증가합니다.

(2) 세부 기능 2 (선택에 따른 결과 제공)

- 플레이어가 선택한 옵션에 따라 스토리 전개가 달라지며 전투 또는 아이템 발견 등의 이벤트가 발생합니다.

3) 기능 3: 전투 시스템

- 플레이어와 적이 차례대로 공격하는 텍스트 기반 전투 기능

(1) 세부 기능 1 (플레이어와 적의 공격 및 체력 관리)

- ASCII 아트를 사용해 전투 중 캐릭터와 무기를 출력하고 플레이어와 적의 체력 및 공격력을 관리합니다.

(2) 세부 기능 2 (던전 난이도에 따른 전투 밸런스 조정)

- 선택한 던전 난이도에 따라 전투의 밸런스가 조정되어 플레이어가 도전을 즐길 수 있도록 설계합니다.

4) 기능 4: 아이템 및 보상 시스템

- 플레이어가 미궁에서 아이템을 발견하거나 전투에서 승리하면 보상을 획득하는 기능

(1) 랜덤 아이템 드랍

- 전투 승리 후 무작위로 회복 아이템 장비 등 다양한 아이템을 드랍하여 게임의 재미를 높입니다.

(2) 아이템 사용 및 효과 적용

- ASCII 아트로 아이템을 표현하고 플레이어가 아이템을 사용하여 체력을 회복하거나 전투 중 공격력을 높일 수 있도록 하는 등의 기능을 합니다.

3. 진척사항

1) 기능 구현

(1) 메인 입출력 함수 (2주차)

- 입출력

- **입력**: 플레이어의 선택 (0, 1, 2, 3 중 하나)
    - 0: 게임 종료
    - 1, 2, 3: 각각 첫 번째, 두 번째, 세 번째 길을 선택
- **출력**:
    - 게임 스토리 및 ASCII 아트 출력

    - 플레이어 선택에 따른 결과 메시지
    - 잘못된 입력 시 경고 메시지 출력
    - 0 입력 시 종료 메시지 출력

- 설명
    - 게임 내 이벤트 발생 및 사용자와의 상호작용을 위한 전체 입력과 출력의 흐름
    - ASCII 아트와 텍스트 기반의 선택지를 출력하여 게임 스토리를 진행.
    - 사용자로부터 길 선택 입력을 받고 이에 따라 이벤트가 발생.
    - 입력 유효성을 검사하고 잘못된 입력에 대해 재입력을 요청.
- 적용된 배운 내용
    - **반복문**: while 루프를 통해 사용자 입력 반복 처리.
    - **조건문**: 유효한 입력인지 확인하고 이에 따라 게임 진행 또는 오류 메시지 출력.
- 코드 스크린샷

```cpp
int main() {
    int choice;
    srand(time(0));
    // ASCII 아트와 소개 스토리를 출력합니다
    displayAsciiArt();
    displayIntro();

    // 올바른 선택이 입력될 때까지 반복
    while (true) {
        // 선택지를 보여주고 플레이어의 입력을 받음
        presentChoices();
        cout << "선택을 입력하세요 (0-3): ";
        cin >> choice;

        if (choice == 0) {
            displayDieArt();
            cout << "게임을 종료합니다. 철수는 당신이 끝까지 지켜봐주지 않아 고독사했습니다." << endl;
            break; // 0을 선택하면 게임 종료
        }

        // 선택이 유효한지 확인
        if (choice >= 1 && choice <= 3) {  // choice가 1, 2, 3 중 하나인지 확인
            processChoice(choice);         // 유효한 선택이면 해당 선택을 처리
        } else {
            cout << "잘못된 선택입니다. 다시 선택해 주세요.\n";  // 잘못된 입력일 때 경고 메시지 출력
        }
    }

    return 0;
}
```

(2) 플레이어의 선택에 따른 결과를 처리하는 함수 (3주차)

- 입출력

- **입력**: 플레이어의 선택값 (choice)과 랜덤 이벤트 결정값 (event).
    - choice: 1 (첫 번째 길), 2 (두 번째 길), 3 (세 번째 길).
    - event: 랜덤으로 결정되는 이벤트 번호 (0, 1, 2, 3 중 하나).
- **출력**:
    - 랜덤 이벤트에 따라 적의 ASCII 아트 출력 및 전투 메시지.
    - 보물상자 발견 시 보물상자 ASCII 아트와 아이템 획득 메시지.
    - 길에 따라 다른 적 또는 보물 이벤트 출력.

- 설명
    - 사용자가 선택한 길에 따라 무작위로 이벤트(몬스터와 전투, 보물상자 발견)가 발생합니다.
    - 각 길마다 이벤트 발생 확률이 다르며 세 번째 길에서는 고블린 전사, 궁수, 마법사와의 전투 또는 보물상자 발견이 동일 확률(25%)로 일어납니다.
    - 이벤트 발생 시 ASCII 아트를 출력하고 보물상자가 발견되면 dropItem() 함수를 호출합니다.
- 적용된 배운 내용
    - **랜덤 처리**: rand()와 % 연산을 활용해 이벤트를 무작위로 선택.
    - **조건문**: if-else를 사용해 선택된 길과 이벤트를 처리.
    - **함수 호출**: 이벤트에 따른 ASCII 아트 출력하는 함수나 보물상자 아이템 드랍 처리를 하는 함수 등을 함수 이름만으로 간단히 호출함.
- 코드 스크린샷

```cpp
// 플레이어의 선택에 따른 결과를 처리하는 함수
void processChoice(int choice) {
    int event = rand() % 4; // 0, 1, 2, 3 중 랜덤 이벤트 선택 -> 선택지가 4개인 3번째 길을 고려하여 수정
    if (choice == 1) {
        // 첫 번째 길: 미노타우르스, 멧돼지, 보물상자
        if (event == 0) {
            displayMinotaurArt();
            cout << "도끼를 든 미노타우르스가 돌진해옵니다!\n";
            cout << "전투를 시작합니다\n";
        } else if (event == 1) {
            displayTreasureArt();
            cout << "보물상자를 발견했습니다.\n";
            dropItem(1); // 첫 번째 길 아이템 드랍
        } else {
            displayBoarArt();//보물상자 나올확률이 25퍼가 되도록하였고 야생멧돼지가 나올확률이 50퍼가 되도록함
            cout << "야생 멧돼지가 돌진해옵니다!\n";
            cout << "전투를 시작합니다\n";
        }
    } else if (choice == 2) {
        // 두 번째 길: 해골전사, 마녀, 보물상자
        if (event == 0) {
            displayWitchArt();
            cout << "마녀가 당신을 노려봅니다!\n";
            cout << "전투를 시작합니다\n";
        } else if (event == 1) {
            displayTreasureArt();
            cout << "보물상자를 발견했습니다.\n";
            dropItem(2); // 두 번째 길 아이템 드랍
        } else {
            displaySkeletonWarriorArt();//해골전사가 나올확률이 25퍼가 되도록하였고 야생멧돼지가 나올확률이 50퍼가 되도록함
            cout << "해골전사가 나타났습니다!\n";
            cout << "전투를 시작합니다\n";
        }
    } else if (choice == 3) {
        // 세 번째 길: 고블린 전사, 궁수, 마법사, 보물상자
        if (event == 0) {
            displayGoblinWarriorArt();
            cout << "고블린 전사가 나타났습니다!\n";
            cout << "전투를 시작합니다\n";
        } else if (event == 1) {
            displayGoblinArcherArt();
            cout << "고블린 궁수가 당신을 겨누고 있습니다!\n";
            cout << "전투를 시작합니다\n";
        }  else if (event == 2) {
            displayGoblinMageArt();
            cout << "고블린 마법사가 주문을 외우고 있습니다!\n";
            cout << "전투를 시작합니다\n";
        } else {
            displayTreasureArt();//굳이 순서안바꿔도 각각 25퍼센트라 괜찮음
            cout << "보물상자를 발견했습니다!\n";
            dropItem(3); // 세 번째 길 아이템 드랍
        }
    }
}
```

(3) 보물상자를 발견했을 때 아이템을 랜덤으로 드랍하는 함수 (5주차)

- 입출력

- **입력**: 함수 dropItem()이 호출될 때 전달된 path 값 (1, 2, 또는 3).

    - path == 1: 첫 번째 길
    - path == 2: 두 번째 길
    - path == 3: 세 번째 길
- **출력**: 보물상자에서 랜덤으로 선택된 아이템 이름이 출력됩니다.

- 설명

- 각 길에서 발견할 수 있는 고유한 보물상자 아이템을 랜덤으로 드랍합니다.
- 길에 따라 드랍 아이템 목록이 다르며, 목록 내에서 무작위로 선택됩니다.

- 적용된 배운 내용

- **배열**: 각 길의 고유 아이템 리스트를 배열로 저장.
- **랜덤 처리**: rand() % 배열 크기로 무작위 아이템 선택.
- **조건문**: if-else를 사용해 선택된 길에 따라 적절한 배열 사용.

- 코드 스크린샷

```
// 보물상자를 발견했을 때 아이템을 랜덤으로 드랍하는 함수
void dropItem(int path) {
    int itemIndex;

    if (path == 1) {
        // 첫 번째 길 아이템 리스트
        string items[] = {"피묻은 도끼", "가죽갑옷", "빨간포션"};
        itemIndex = rand() % 3;
        cout << "보물상자에서 " << items[itemIndex] << "을(를) 얻었습니다!" << endl;
    } else if (path == 2) {
        // 두 번째 길 아이템 리스트
        string items[] = {"날카로운 철검", "빨간포션", "천갑옷"};
        itemIndex = rand() % 3;
        cout << "보물상자에서 " << items[itemIndex] << "을(를) 얻었습니다!" << endl;
    } else if (path == 3) {
        // 세 번째 길 아이템 리스트
        string items[] = {"야만의 몽둥이", "빨간포션"};
        itemIndex = rand() % 2;
        cout << "보물상자에서 " << items[itemIndex] << "을(를) 얻었습니다!" << endl;
    }
}
```

(4) ASCII 아트 출력 및 스토리 출력 함축 함수 (2주차, 3주차)

- 입출력

  - **입력**:
    - 별도의 사용자 입력은 없음. 함수 호출 시 고정된 ASCII 아트와 스토리를 출력.
  - **출력**:
    - 특정 이벤트(예: 철수가 잠든 모습, 몬스터, 보물상자 등)에 따른 ASCII 아트와 메시지를 출력.
    - 스토리 소개를 통해 게임의 배경 설명과 플레이어의 몰입도 증가.

- 설명

  - 각 함수는 고유한 ASCII 아트를 출력하며, 게임 스토리와 이벤트를 시각적으로 표현.
  - 게임의 서사적 요소와 시각적 몰입감을 제공하기 위해 활용됨.
    - displayAsciiArt: 철수가 책상에서 잠든 모습을 ASCII 아트로 표현.
    - displayMinotaurArt, displayBoarArt, displaySkeletonWarriorArt 등: 적의 모습을 ASCII 아트로 출력.
    - displayTreasureArt: 보물상자 발견 시 아트 출력.
    - displayDieArt: 게임 종료 시 사망 메시지를 출력.
    - displayIntro: 게임 시작 시 배경 스토리를 출력.

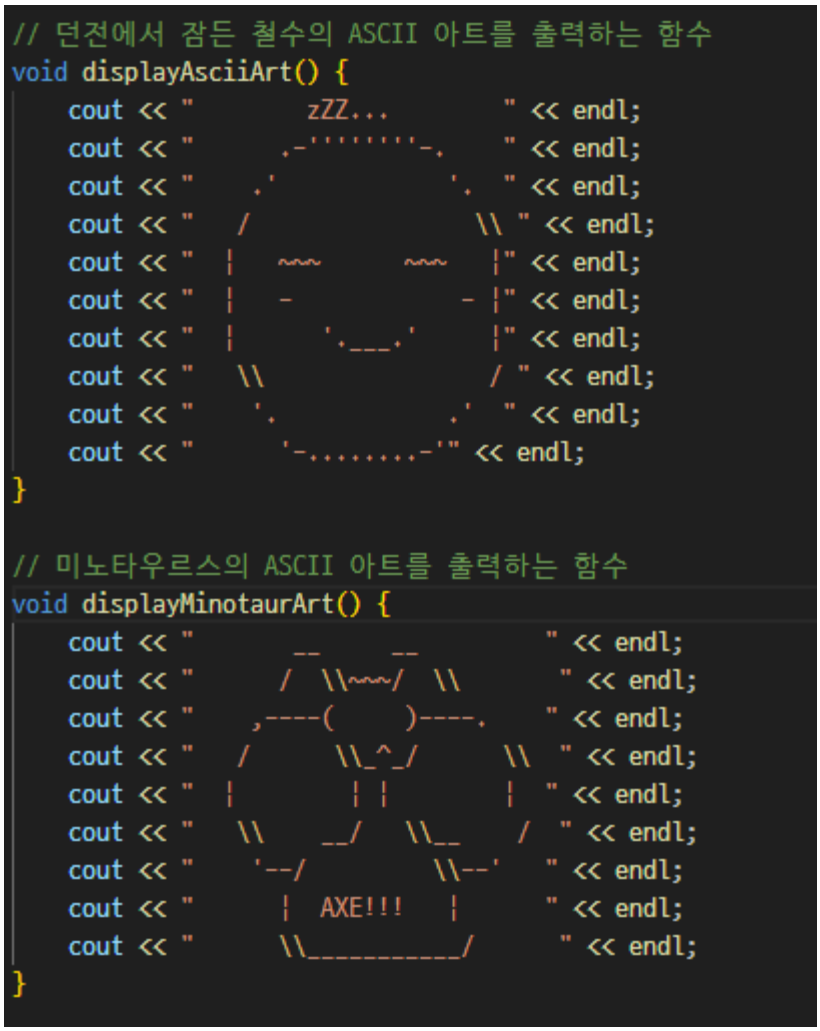
- 적용된 배운 내용

  - **함수**:
    - 각 ASCII 아트를 별도의 함수로 분리하여 유지보수성과 가독성을 향상.
    - 필요할 때마다 호출 가능.
  - **출력 제어**:

- cout을 활용한 텍스트 출력.
- 줄바꿈(endl)과 공백 등을 활용해 아트를 정확한 위치에 배치.

- **코드 구조화**:
  - 여러 적, 이벤트, 스토리를 각각의 함수로 분리하여 코드 중복 제거.
  - 스토리텔링 요소를 효율적으로 구현.

- 코드 스크린샷



```
// 던전에서 잠든 철수의 ASCII 아트를 출력하는 함수
void displayAsciiArt() {
    cout << "        zZZ,,,         " << endl;
    cout << "      .-''''''''-.     " << endl;
    cout << "    .'            '.   " << endl;
    cout << "   /                \\ " << endl;
    cout << "  |   ~~~~      ~~~~   |" << endl;
    cout << "  |   -            -   |" << endl;
    cout << "  |      '.___.'      |" << endl;
    cout << "   \\                / " << endl;
    cout << "    '.            .'  " << endl;
    cout << "      '-........-'" << endl;
}

// 미노타우르스의 ASCII 아트를 출력하는 함수
void displayMinotaurArt() {
    cout << "        __      __          " << endl;
    cout << "      /  \\~~~/  \\        " << endl;
    cout << "    ,----(     )----.     " << endl;
    cout << "   /      \\_^_/      \\   " << endl;
    cout << "  |         | |         |  " << endl;
    cout << "   \\    __/   \\__    /   " << endl;
    cout << "    '--/         \\--'    " << endl;
    cout << "      |  AXE!!!   |      " << endl;
    cout << "      \\___________/      " << endl;
}
```

```cpp
// 멧돼지의 ASCII 아트를 출력하는 함수
void displayBoarArt() {
    cout << "      /\\   ___   /\\     " << endl;
    cout << "     //\\\\ (o o) //\\\\    " << endl;
    cout << "    ||   \\\\_^_//   ||    " << endl;
    cout << "     \\\\    '-'    //     " << endl;
    cout << "      '-........-'       " << endl;
}

// 보물상자의 ASCII 아트를 출력하는 함수
void displayTreasureArt() {
    cout << "       __________        " << endl;
    cout << "     /          \\       " << endl;
    cout << "    / $$$$$$$$$  \\      " << endl;
    cout << "   | $$$$$$$$$$ |      " << endl;
    cout << "   | $$$$$$$$$$ |      " << endl;
    cout << "    \\_____[T]____/      " << endl;
}

// 해골전사의 ASCII 아트를 출력하는 함수
void displaySkeletonWarriorArt() {
    cout << "      .-.           " << endl;
    cout << "     (o.o)          " << endl;
    cout << "      |=|           " << endl;
    cout << "     __|__          " << endl;
    cout << "   //.=|=.\\\\        " << endl;
    cout << "  // .=|=. \\\\       " << endl;
    cout << "  \\\\ .=|=. //       " << endl;
    cout << "   \\\\(_=_)//        " << endl;
    cout << "    (:| |:)         " << endl;
    cout << "     || ||          " << endl;
    cout << "     () ()          " << endl;
}

// 마녀의 ASCII 아트를 출력하는 함수
void displayWitchArt() {
    cout << "        ______         " << endl;
    cout << "     .-'      `-.      " << endl;
    cout << "    /    .--.    \\     " << endl;
    cout << "   /  _.-'    '-._\\    " << endl;
    cout << "   | (_ o      o _) |  " << endl;
    cout << "    \\   '._.--._.'  /  " << endl;
    cout << "     '-.          .-'   " << endl;
    cout << "        `--------`      " << endl;
}

// 고블린 전사의 ASCII 아트를 출력하는 함수
void displayGoblinWarriorArt() {
    cout << "     .----.      " << endl;
    cout << "    /  o o \\     " << endl;
    cout << "   |    ^    |    " << endl;
    cout << "   |  \\___/  |    " << endl;
    cout << "    \\_______/     " << endl;
    cout << "     //   \\\\      " << endl;
    cout << "    ||     ||     " << endl;
}


// 고블린 궁수의 ASCII 아트를 출력하는 함수
void displayGoblinArcherArt() {
    cout << "     .----.      " << endl;
    cout << "    /  o o \\     " << endl;
    cout << "   |    ^    |    " << endl;
    cout << "   |  \\___/  |    " << endl;
    cout << "    \\\--| |--/     " << endl;
    cout << "       | |        " << endl;
}
```

```
// 고블린 마법사의 ASCII 아트를 출력하는 함수
void displayGoblinMageArt() {
    cout << "     .----.     " << endl;
    cout << "    /  o o \\    " << endl;
    cout << "   |    ^    |   " << endl;
    cout << "   |  \\___/  |   " << endl;
    cout << "    \\_-====-_/    " << endl;
    cout << "     /     \\     " << endl;
}

//사망 메시지 출력 함수
void displayDieArt() {
    cout << "[illegible]" << endl;
    cout << "[illegible]" << endl;
    cout << "[illegible]" << endl;
    cout << "[illegible]" << endl;
    cout << "[illegible]" << endl;
    cout << "[illegible]" << endl;
    cout << "[illegible]" << endl;
    cout << "[illegible]" << endl;
    cout << "[illegible]" << endl;
    cout << "[illegible]" << endl;
    cout << "[illegible]" << endl;
    cout << "[illegible]" << endl;
    cout << "[illegible]" << endl;
    cout << "[illegible]" << endl;
}

// 소개 스토리를 출력하는 함수
void displayIntro() {
    cout << "[illegible]" << endl;
    cout << "[illegible]" << endl;
    cout << "[illegible]" << endl;
    cout << "[illegible]" << endl;
    cout << "[illegible]" << endl;
    cout << "[illegible]" << endl;
    cout << "[illegible]" << endl;
    cout << "[illegible]" << endl;
    cout << "[illegible]" << endl;
    cout << "[illegible]" << endl;
    cout << "[illegible]" << endl;
    cout << "[illegible]" << endl;
    cout << "[illegible]" << endl;
    cout << "[illegible]" << endl;
    cout << "한밤중, 코딩에 집중하던 철수는 졸음에 못 이겨 그만 책상에 고꾸라져 잠이 들고 말았습니다.\n";
    cout << "얼마나 시간이 흘렀을까, 깨달아 보니 낯선 돌바닥 위. 주변은 차갑고 축축한 공기로 가득 찼습니다.\n";
    cout << "눈앞엔 고대 유적 같은 돌기둥들이 우뚝 서 있고, 어둠 속에서 흐릿한 불빛이 깜박입니다.\n";
    cout << "'여긴... 어디지?' 철수는 아찔한 느낌에 사로잡혔습니다. 뒤를 돌아보니 더는 돌아갈 길이 없었습니다.\n";
    cout << "갑자기 멀리서 낮게 울리는 소리-짐승의 숨소리 같은 것이 들려옵니다.\n";
    cout << "“살아남으려면 뭔가를 찾아야 해...” 철수는 두근거리는 가슴을 안고 주변을 살피기 시작했습니다.\n";
    cout << "손을 뻗어 벽을 더듬는 순간, 날카로운 금속의 감촉이 느껴졌습니다.\n";
    cout << "철수는 작은 칼을 발견하고 움켜쥐었습니다. 이제, 이 던전에서 생존을 위한 첫 발걸음을 내디딥니다.\n\n";
    cout << "앞을 보니 세 가지 길이 있습니다.\n\n";
}

// 플레이어에게 선택지를 보여주는 함수
void presentChoices() {
    cout << "\n어떻게 할까요?\n";
    cout << "1. 첫번째 길로 간다\n";
    cout << "2. 두번째 길로 간다\n";
    cout << "3. 세번째 길로 간다\n";
    cout << "0. 게임종료\n";
}
```

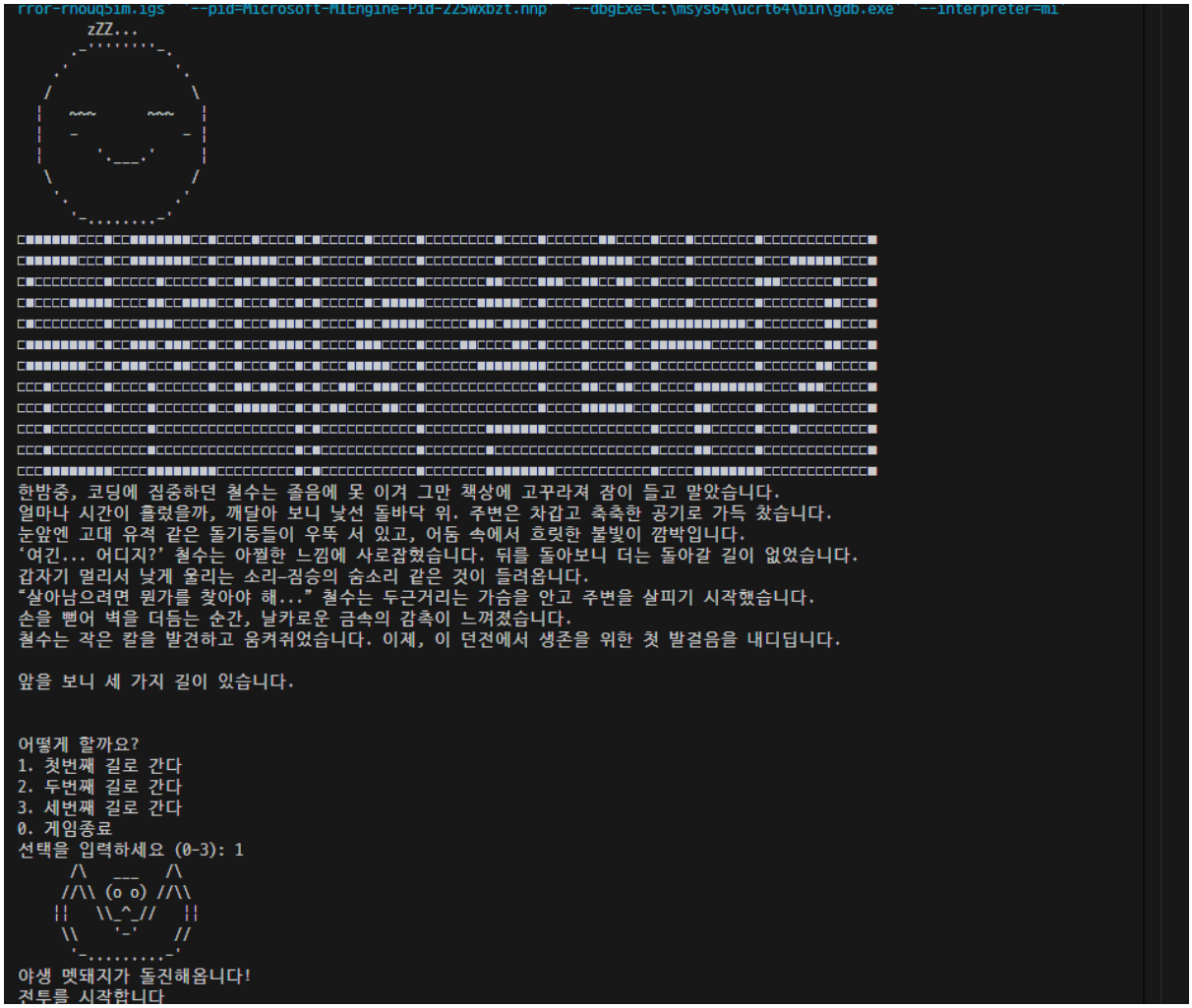
2) 테스트 결과

(1) 메인 입출력 함수

- 설명

  - ASCII 아트와 소개 스토리 출력.
  - 사용자 입력을 받아 선택을 처리하거나 종료.
  - 유효하지 않은 입력값에 대한 오류 처리.
  - 전체적인 흐름을 한눈에 보여줌
  - 코드의 본체

- 테스트 결과 스크린샷



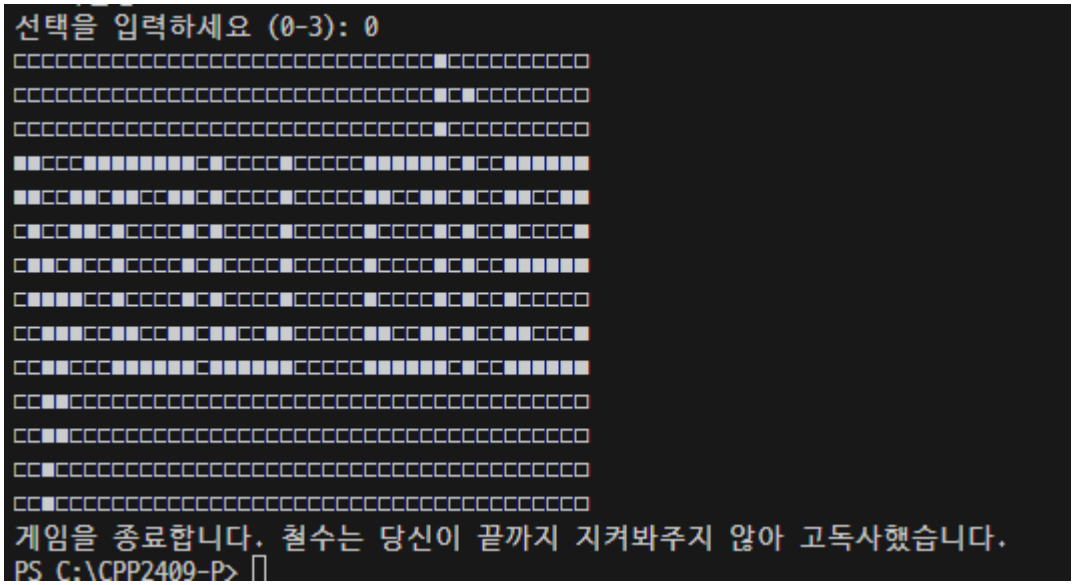
```
어떻게 할까요?
1. 첫번째 길로 간다
2. 두번째 길로 간다
3. 세번째 길로 간다
0. 게임종료
선택을 입력하세요 (0-3): 3
     .----.
    /  o o \
   |    ^   |
   |  \___/  |
    \_-===-_/
     /    \
고블린 마법사가 주문을 외우고 있습니다!
전투를 시작합니다

어떻게 할까요?
1. 첫번째 길로 간다
2. 두번째 길로 간다
3. 세번째 길로 간다
0. 게임종료
선택을 입력하세요 (0-3): 3

      __________
     /          \
    / $$$$$$$$$$  \
   | $$$$$$$$$$$$ |
   | $$$$$$$$$$$$ |
    \_____[T]____/
보물상자를 발견했습니다!
보물상자에서 빨간포션을(를) 얻었습니다!

어떻게 할까요?
1. 첫번째 길로 간다
2. 두번째 길로 간다
3. 세번째 길로 간다
0. 게임종료
선택을 입력하세요 (0-3): 0
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□■□□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□■□■□□□□□□□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□■□□□□□□□□□□□
■■□□□■■■■■■■□■□□□□■□□□□□■■■■■■□■□□□■■■■■■■
■■□□□■■□■■□□■■□■□□□□■□□□□□■■□□■■□■□□■■□□■■■
□■□□□■■□■□□□□■□■□□□□■□□□□□■□□□□■□■□□□■□□□□■
□■■□■□□□■□□□□■□■□□□□■□□□□□■□□□□■□■□□□■■■■■■■
□■■■■□□□■□□□□■□■□□□□■□□□□□■□□□□■□■□□□■□□□□□
□□■■■□□□■■□□□■■□■□□□■■□□□□□■■□□□■■□■□□□■■□□□■
□□■■□□□□■■■■■■□■■■■■■□□□□□■■■■■■□■□□□■■■■■■■
□□■■□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□■■□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□■□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□■□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
게임을 종료합니다. 철수는 당신이 끝까지 지켜봐주지 않아 고독사했습니다.
```

(2) 플레이어의 선택에 따른 결과를 처리하는 함수

- 설명

  - 플레이어의 선택에 따라 랜덤 이벤트(몬스터, 보물상자)를 처리.

- 테스트 결과 스크린샷

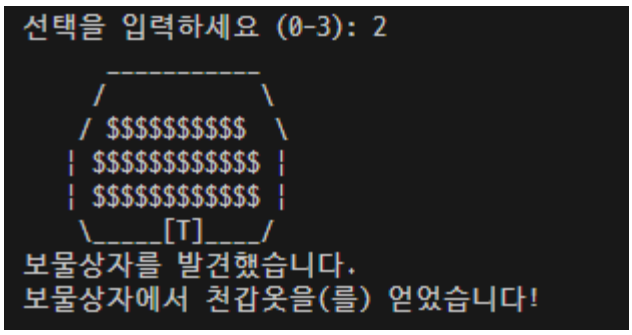


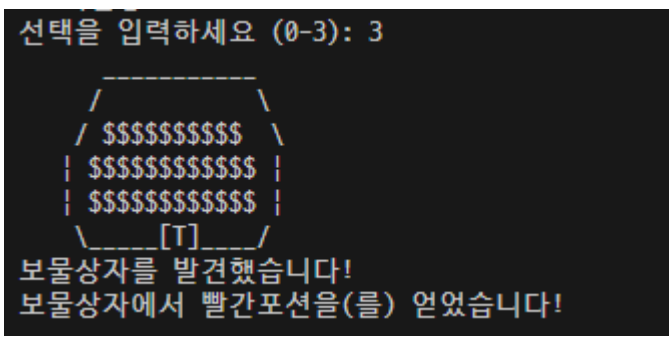
(3) 보물상자를 발견했을 때 아이템을 랜덤으로 드랍하는 함수

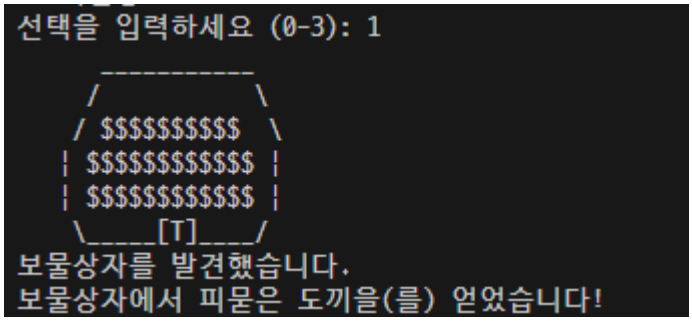
- 설명

  - 각 길에 따라 적절한 아이템 리스트에서 랜덤하게 드랍.

- 테스트 결과 스크린샷







(4) ASCII 아트 출력 및 스토리 출력 함축 함수

- 설명

  - 각 상황에 맞는 ASCII 아트 및 스토리 출력.

- 테스트 결과 스크린샷

4. 계획 대비 변경 사항

1) 기능2의 세부 기능1

- 이전 : 쉬움,보통,어려움으로 단순히 난이도를 유저가 선택하게끔하려고함

- 이후 : 길1,길2,길3로 나눠서 길마다 등장 몬스터나, 보물상자에서 나오는 아이템이 다르도록 조정함

- 사유 : 난이도를 알려줘서 플레이하면 재미가 없을 것 같아서이다.

2) 기능3의 개발 일정 조정

- 이전 : 기능2 이후 기능3을 개발하려고함

- 이후 : 기능2와 3을 같이 개발하려고함

- 사유 : 난이도별 몬스터의 체력과 공격력을 조절하는건 전투 시스템을 구축하면서 생각해봐야할 문제이기 때문이다. 던전에 있는 길마다의 전투 밸런스를 조정한다거나 하는 등 함께 고려해야할 사항이 많기 때문에 다음 진척보고서까지 함께 고려하여 한번에 개발을 진행하려고한다.

3) 기능4의 세부기능2 개발 일정 조정

- 이전 : 랜덤 아이템 드랍을 6주차부터 개발하려고했음

- 이후 : 미리 개발 완료를 하였음

- 사유 : 스토리 진행부분인 2,3주차 부분에서 아이템을 드랍하는 내용을 만드는 도중에 어차피 나중에 랜덤으로 드랍할건데 지금 구현하면 편할 것 같아서 미리 했다.

5. 프로젝트 일정

<table>
<tr><th colspan="2">업무</th><th>11/3</th><th>11/10</th><th>11/17</th><th>11/24</th><th>12/1</th><th>12/8</th><th>12/15</th><th>12/22</th></tr>
<tr><td colspan="2">제안서 작성</td><td>완료</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>기능 1</td><td>세부 기능1</td><td></td><td colspan="2">완료</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>세부 기능2</td><td></td><td colspan="2">완료</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>기능 2</td><td>세부 기능1</td><td></td><td></td><td></td><td colspan="2">진행중</td><td></td><td></td><td></td></tr>
<tr><td></td><td>세부 기능2</td><td></td><td></td><td></td><td colspan="2">진행중</td><td></td><td></td><td></td></tr>
<tr><td>기능 3</td><td>세부 기능1</td><td></td><td></td><td></td><td colspan="2">진행중</td><td></td><td></td><td></td></tr>
<tr><td></td><td>세부 기능2</td><td></td><td></td><td></td><td colspan="2">진행중</td><td></td><td></td><td></td></tr>
<tr><td>기능 4</td><td>세부 기능1</td><td></td><td></td><td>완료</td><td colspan="4"></td></tr>
<tr><td></td><td>세부 기능2</td><td></td><td></td><td></td><td></td><td colspan="3">----------
----------></td></tr>
<tr><td>전체 테스트 및 수정</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="2">----------
----------></td></tr>
</table>