

FIRST JOURNEY

START

INTRODUCTION

ผู้เล่นจะได้รับบทเป็นนักผจญภัยที่เดินทางไปจัดการกับมังกรที่เป็น Final boss โดย
ในระหว่างทางจะมีมอนสเตอร์ต่างๆปรากฏขึ้นแบบสลับ

MONSTERS



Goblin



Orc



Dire wolf



Slime

Final Boss



Dragon
king

HOW TO PLAY

STEP 1 Choose difficulty level

- 1.EASY
- 2.MEDIUM
- 3.HARD



STEP 2 ENTER PLAYER NAME

Player Name :

HP 100 | MANA 100 | POTION 3

=== Choose an action ===

1. Normal Attack (0 Mana, 15 dmg)
2. Fire Bolt (10 Mana, 20 dmg)
3. EXCALIBURR (70 Mana, 80 dmg)
4. Use Potion (+30 HP)

HOW TO PLAY

- ผู้เล่นสามารถเลือก action ได้ 1 อย่างต่อเทิร์นตาม option ที่มีให้เลือก
- เมื่อผู้เล่นเลือก action จะแสดงให้เห็นว่าผู้เล่นเลือก action ไหนบน terminal
- หากเลือกทำการโจมตี จะแสดงให้เห็นว่าผู้เล่นทำดาเมจใส่มอนสเตอร์ไปเท่าไร
- หากเลือกใช้ potion จะแสดงให้เห็น potion ที่เหลืออยู่และแสดงให้เห็นว่าได้รับHPเพิ่มเท่าไร

FINAL RESULT

- หากฝ่าย monster ชนะให้แสดงว่า “You were defeated by... Game Over.” และแสดง Final score ที่ได้รับ
- หากผู้เล่นสามารถเคลียร์มอนสเตอร์ไปได้จนถึง Final boss และสามารถชนะ Final boss ได้ให้แสดงว่า “You have defeated the Final Boss! Victory!!” และแสดง Final score ที่ได้รับ

Functional

1. เลือกระดับความยาก

- ผู้เล่นสามารถเลือกระดับความยากได้ 3 ระดับ: Easy, Medium, Hard
- ความยากส่งผลต่อ HP และพลังโจมตีของมอนสเตอร์

2. ระบบต่อสู้

- ผู้เล่นสามารถเลือกใช้การโจมตีปกติ, เวท Fire Bolt, หรือ EXCALIBURR ได้
- สามารถใช้โพชั่นเพิ่ม HP ระหว่างต่อสู้ได้
- การต่อสู้เป็นแบบเทิร์นเบส สลับการโจมตี

3. ระบบ HP และ Mana

- ตัวละครมี HP และ Mana ซึ่งลดลงตามการต่อสู้และใช้งานเวท
- Mana จะเพิ่มขึ้นในแต่ละเทิร์นโดยอัตโนมัติ

Functional

4. สกิลและเวทของผู้เล่น

- โจมตีปกติไม่ใช้mana
- Fire Bolt ใช้mana 10 และทำดาเมจ 20
- EXCALIBURR ใช้mana 70 และทำดาเมจ 80

5. มอนสเตอร์และบอส

- มอนสเตอร์มีการสุ่มชนิด HP และพลังโจมตีตามระดับความยาก
- ด้านสุดท้ายจะมีบอส "Dragon King" ซึ่งมี HP และพลังโจมตีสูงมาก

6. ระบบคะแนน

- ได้คะแนนเมื่อชนะมอนสเตอร์โดยคะแนนที่ได้คำนวณจากระดับความยากของศัตรู
- แสดงคะแนนรวมเมื่อจบเกม

7. ระบบเล่นซ้ำ

- หลังจบเกม ผู้เล่นสามารถเลือกได้ว่าจะเล่นใหม่หรือจบเกม

Non-Functional

1. ประสิทธิภาพของโค้ด

- โค้ดสามารถรันได้อย่างราบรื่น ไม่มีข้อผิดพลาด
- ไม่มีการหน่วงเวลาในเกมเพลย์

2. ความเข้าใจง่าย

- โค้ดมีโครงสร้างชัดเจน อ่านง่าย
- การตั้งชื่อคลาส ฟังก์ชัน และตัวแปรบ่งบอกหน้าที่ชัดเจน เช่น Player, Monster, attack()

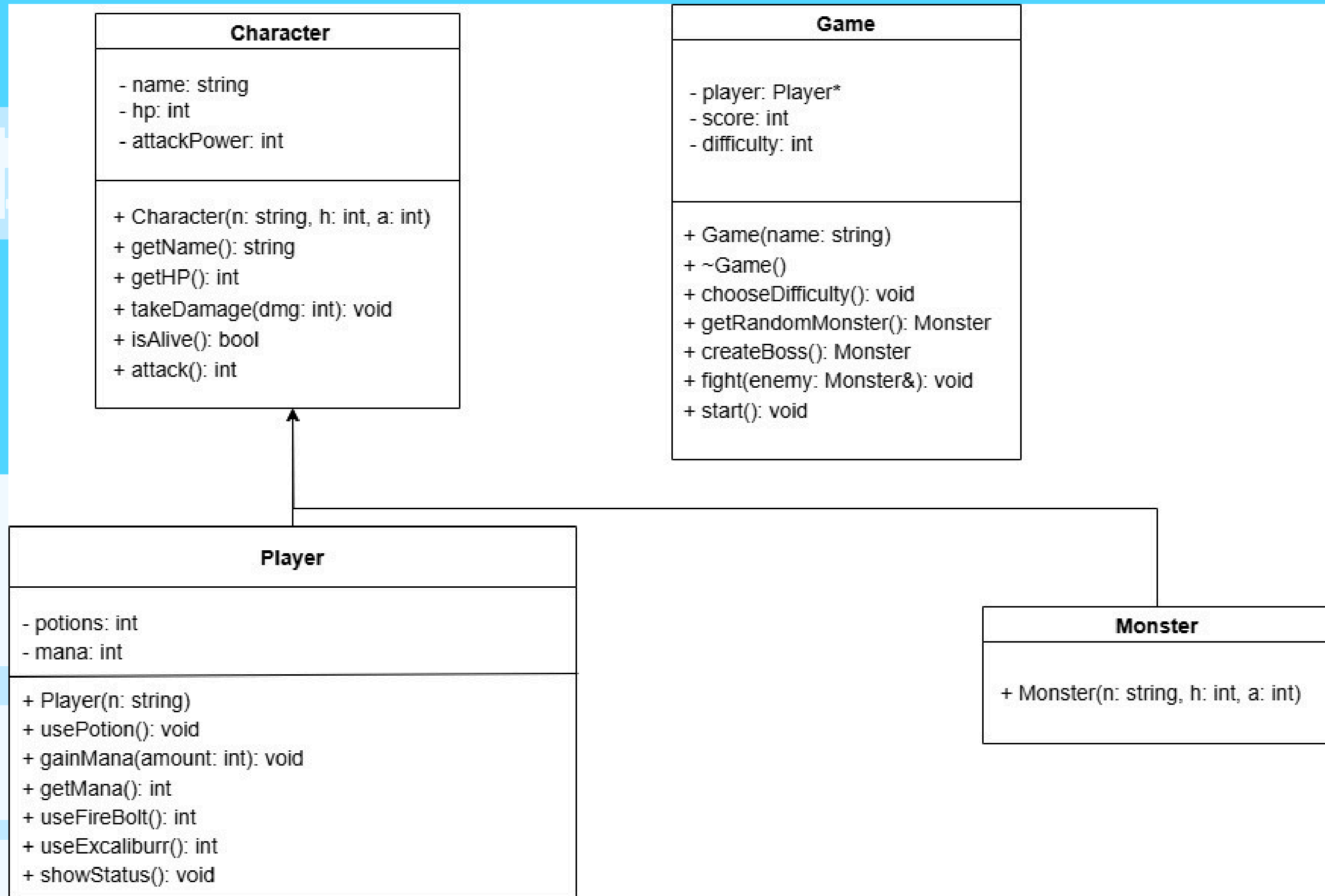
3. การพัฒนาในอนาคต

- โครงสร้างโปรแกรมรองรับการเพิ่มระบบใหม่ เช่น เพิ่มคลาสตัวละคร, เพิ่มสกิล, เพิ่มด่านได้ง่าย
- สามารถเพิ่มมอนสเตอร์หรือบอสใหม่ได้โดยไม่ต้องยุ่งยาก

4. การโต้ตอบกับผู้เล่น

- มีข้อความอธิบายทุกตัวเลือก เช่น เมื่อเลือกกระหว่างเวท, โจมตี, ใช้ยา
- แสดงสถานะ HP, Mana, และจำนวน Potion ทุกเทิร์น

Flow Chart



Class example

```
class Character {
protected:
    string name;
    int hp;
    int attackPower;
public:
    Character(string n, int h, int a) : name(n), hp(h), attackPower(a) {}
    virtual ~Character() {}

    string getName() { return name; }
    int getHP() { return hp; }
    void takeDamage(int dmg) { hp -= dmg; if (hp < 0) hp = 0; }
    bool isAlive() { return hp > 0; }
    virtual int attack() { return attackPower; }
};
```

Class example

```
class Player : public Character {
private:
    int potions;
    int mana;
public:
    Player(string n) : Character(n, 100, 20), potions(5), mana(100) {}

    void usePotion() {
        if (potions > 0) {
            hp += 30;
            if (hp > 100) hp = 100;
            potions--;
            cout << name << " uses a potion! (+30 HP, Potions left: " << potions << ")\n";
        } else {
            cout << "No potions left!\n";
        }
    }
}
```

Function

```
void chooseDifficulty() {  
    cout << "Choose difficulty level:\n1. Easy\n2. Medium\n3. Hard\n> ";  
    cin >> difficulty;  
    if (difficulty < 1 || difficulty > 3) difficulty = 2;  
}  
  
Monster getRandomMonster() {  
    vector<string> names = {"Slime", "Goblin", "Orc", "Wolf"};  
    string mName = names[rand() % names.size()];  
    int mHP = 40 + (rand() % 20) + difficulty * 5;  
    int mAtk = 6 + difficulty * 2;  
    return Monster(mName, mHP, mAtk);  
}  
  
Monster createBoss() {  
    return Monster("Dragon King", 400, 20);  
}
```

Function

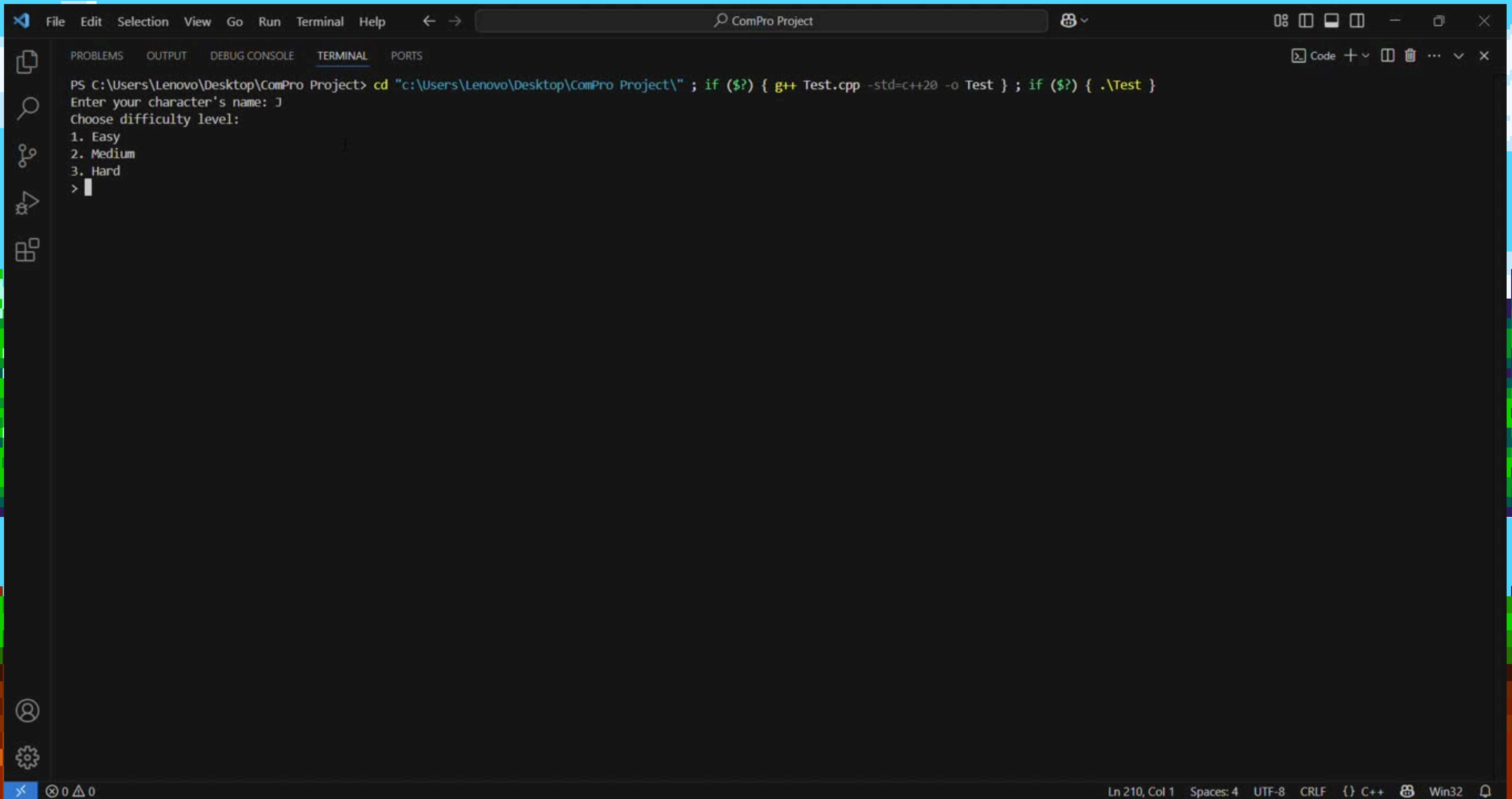
```
Player(string n) : Character(n, 100, 20), potions(5), mana(100) {}

void usePotion() {
    if (potions > 0) {
        hp += 30;
        if (hp > 100) hp = 100;
        potions--;
        cout << name << " uses a potion! (+30 HP, Potions left: " << potions << ")\n";
    } else {
        cout << "No potions left!\n";
    }
}

void gainMana(int amount) {
    mana += amount;
    if (mana > 100) mana = 100;
}

int getMana() { return mana; }
```

PROGRAM DEMO



```
File Edit Selection View Go Run Terminal Help ComPro Project
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Lenovo\Desktop\ComPro Project> cd "c:\Users\Lenovo\Desktop\ComPro Project\" ; if ($?) { g++ Test.cpp -std=c++20 -o Test } ; if ($?) { .\Test }
Enter your character's name: J
Choose difficulty level:
1. Easy
2. Medium
3. Hard
> |
```

Ln 210, Col 1 Spaces: 4 UTF-8 CRLF {} C++ Win32

YEAY! GAME IS OVER!

THANK YOU