

QT 프로젝트 (Knife Hit)

목차

1. 코드 설명

2. 소개

캡슐화

```
class Player {  
private:  
    QString name;  
    int level;  
    int score;  
public:  
    Player(const QString& name, int level, int score);  
  
    QString getName() const { return name; }  
    int getLevel() const { return level; }  
    int getScore() const { return score; }
```

Player클래스에서 name, level, score같은 멤버 변수를 private로 선언

- getName()
 - getLevel()
 - getScore()
- > 값을 읽기 위해 선언

추상화

```
// 파일에서 기존 Player 정보를 불러옴  
std::vector<Player> players = Player::loadFile();  
  
// 파일에 업데이트된 랭킹을 저장  
Player::saveRankingFile(players);
```

- saveRankingFile
- loadFile

-> 파일 입출력 관련 로직이 구현 되어있다.

But. 함수를 호출하는 쪽에서는 동작만 추상적으로 사용할 수 있도록 하였다.

STL

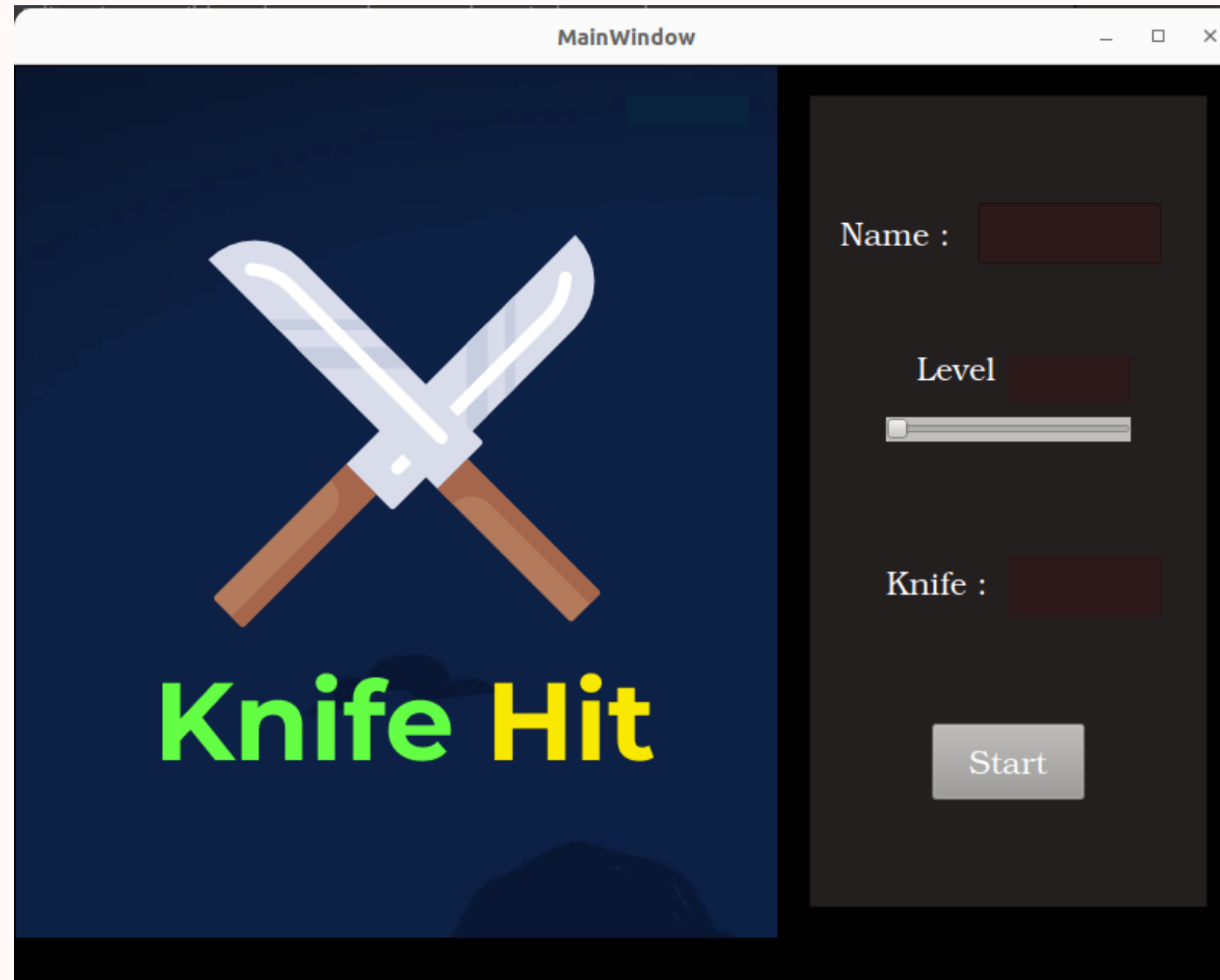
```
// 파일에서 기존 Player 정보를 불러옴
std::vector<Player> players = Player::loadFile();

// player 객체를 players 벡터에 추가
players.push_back(player);

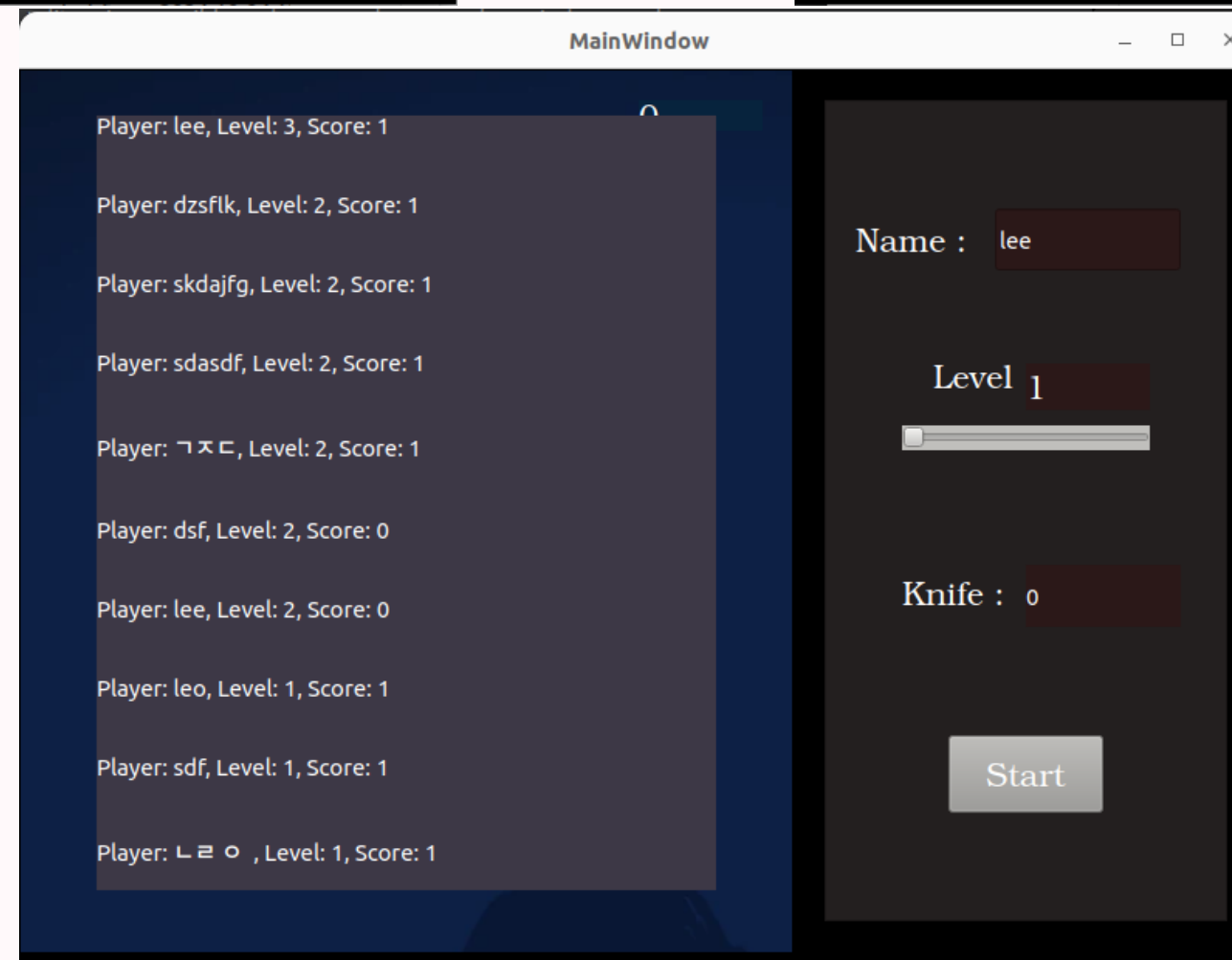
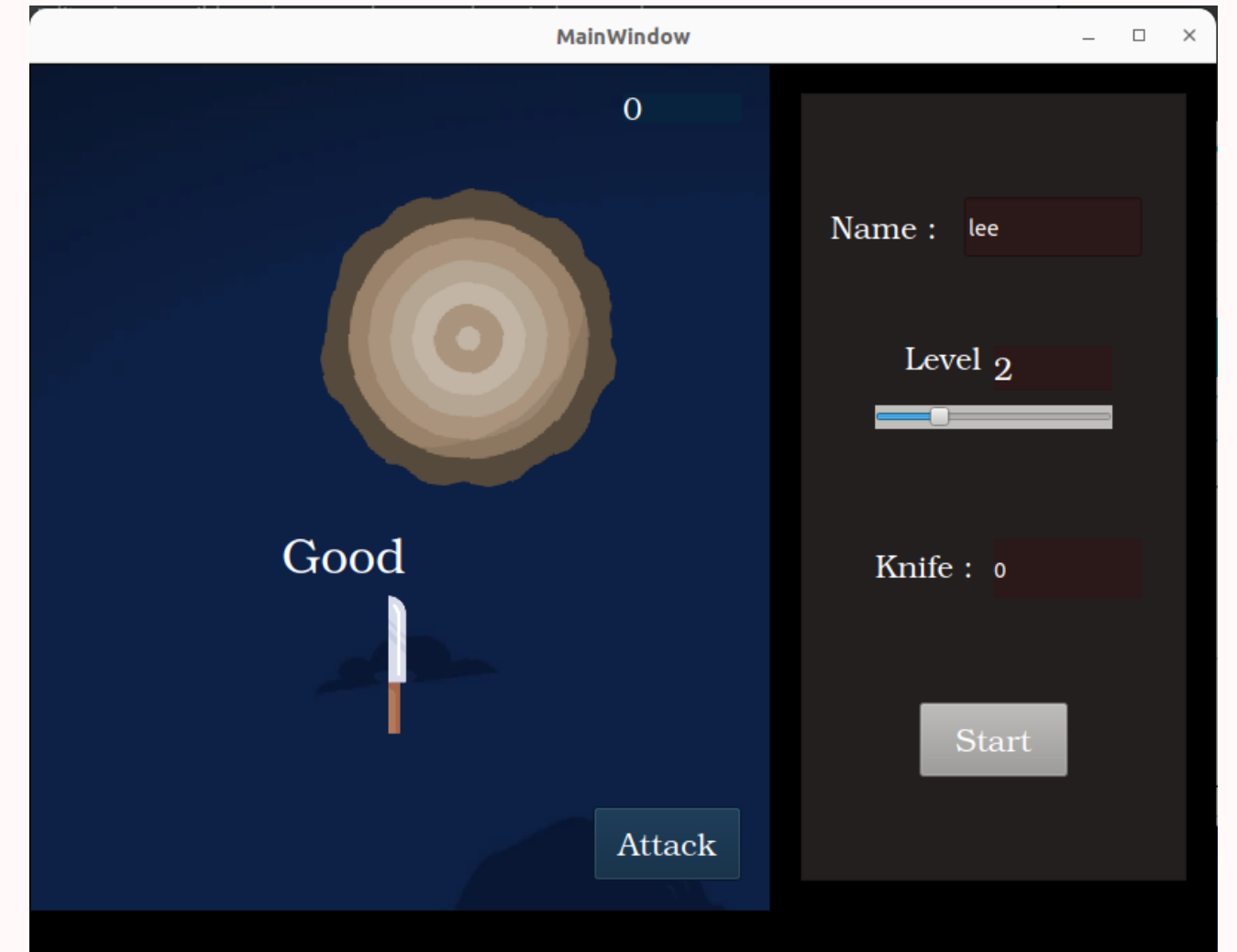
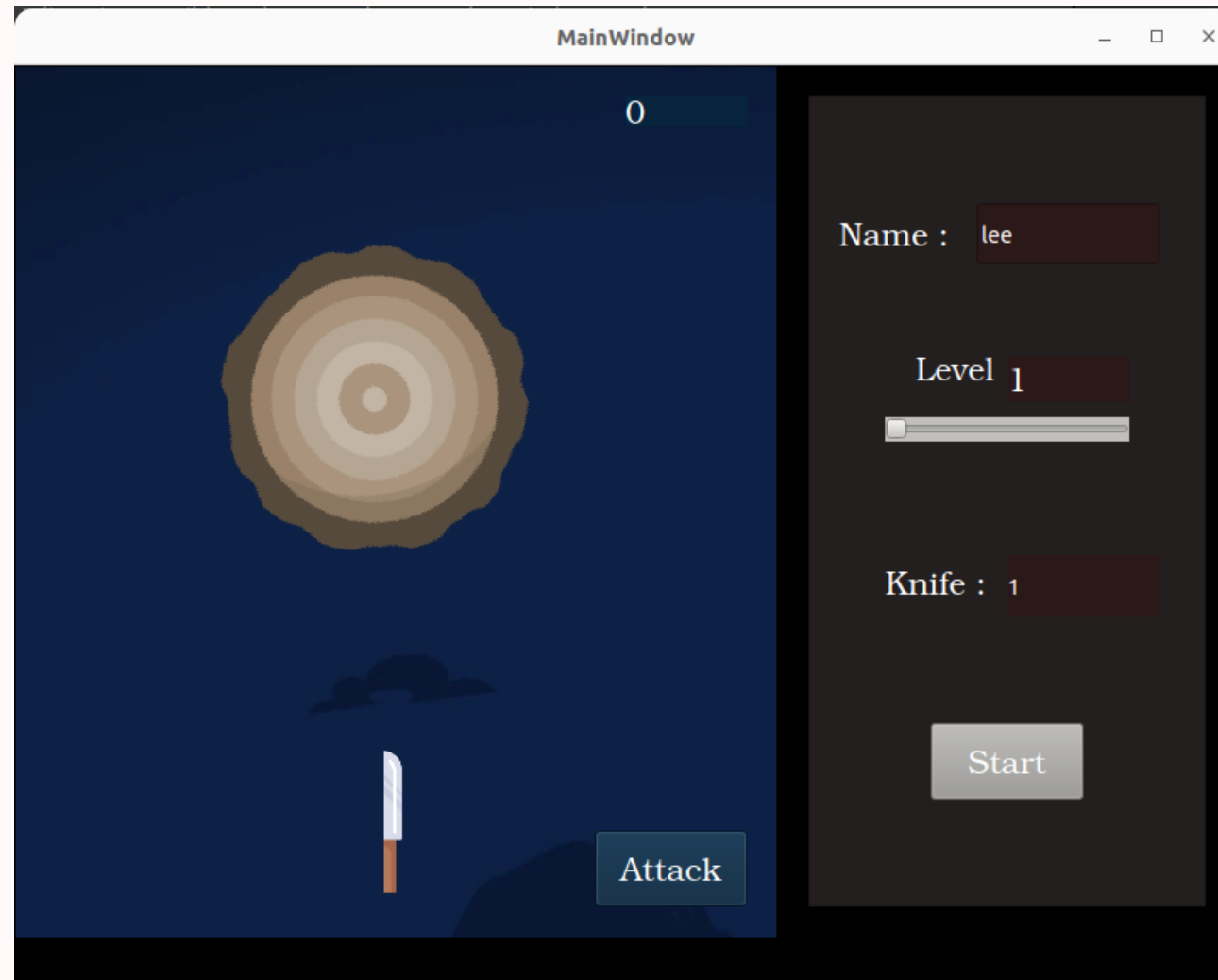
// 내림차순으로 랭킹을 정렬 (Level 우선, 그 다음 Score)
std::sort(players.begin(), players.end(), [](const Player& a, const Player& b) {
    if (a.getLevel() == b.getLevel()) {
        return a.getScore() > b.getScore(); // 점수가 같으면 Score로 비교
    }
    return a.getLevel() > b.getLevel(); // Level 우선 비교
});
```

- vector
 - sort
- > 랭킹을 효율적으로 관리할 수 있다.

소개



소개



감사합니다.