

# Squid Game in C++

C++로 오징어 게임 구현하기



# Index

1. **Squid Game 설명**
2. **Squid Game 에 사용된 OOP/C++**
3. **구현된 Squid Game 간단 설명**
4. **아쉬운 점 / 개선해야 할 점**

# 1. Squid Game

## 드라마에서의 Squid Game

1단계: 무궁화 꽃이 피었습니다

2단계: 설탕 뽑기

3단계: 줄다리기

4단계: 구슬치기

5단계: 징검다리 건너기

6단계: 오징어 게임

+ 가위바위보



## 2. Squid Game 에 사용된 OOP

### 1. 추상 class - Game과 Player

```
class Game
{
protected:
    std::string gameName;
    std::list<Player*> players;
    virtual void printGameName();
public:
    Game(std::string name) :gameName(name) {};

    virtual ~Game()=0;
    virtual void join(Player* player) = 0;
    virtual void play() = 0;
    void printAlivePlayers();
    int getGameType(); // marble game 에서 사용
    std::list<Player*> getAlivePlayers() { return players; };
    std::string getGameName() { return gameName; };
    //Player* operator[](int index) { return players[index]; };
};
```

# 2. Squid Game 에 사용된 OOP

## 2. Game과 Player를 상속받는 class들

```
class PlayerMarbleGame : public Player
{
public:
    std::pair< int, int> player1_marble; //Odd Or Even 뿐만 아니라 Hole In One 게임에서 동일한 매커니즘
    을 사용하므로 정의 class를 marbleGame으로 이동 필요
    std::pair< int, int> player2_marble;
    PlayerMarbleGame(const Player& player) : Player(player) { playing = true; };
    virtual bool act2() { return false; }; // 플레이어 2명에서 하는 게임이라, player1, player2의 act를
    각각 정의해주기 위해 act2()함수를 순수 가상함수로 만들어줌
    virtual void checkMarble() = 0;
    void CompareOddorEven(const std::pair<int, int>& player1_marble, const std::pair<int, int>&
    player2_marble);
    bool check_OddOrEven = false;
    bool check_num = false;
    void dyingMessage();
};
```

마찬가지로 Game 및 Player를 상속받는 다양한 class 존재

## 2. Squid Game 에 사용된 OOP

### 3. 캡슐화 (Encapsulation), 동적 바인딩(dynamic binding)

```
class Player
{
public:
    Player(int number);
    Player(int number, int agility, int fearlessness);
    ~Player() {};
    virtual bool hasEnoughMarbles() { return false; };
    virtual bool act() { return true; };
    virtual bool act2() { return true; }; //marble game 에서 사용
    virtual void checkMarble() { }; // marble game에서 사용
    virtual bool isPlaying() { return playing; };
    void aliveMessage() { printStatus(); std::cout << " is alive." << std::endl; };
    virtual void dyingMessage() { printStatus(); std::cout << " died." << std::endl; };
    virtual void printStatus() { std::cout << "Player #" << number << "(" << agility << "," <<
    fearlessness << ")"; };
protected:
    unsigned int number;
    unsigned int agility;
    unsigned int fearlessness;
    bool playing = true;
    static std::default_random_engine random_engine;
    static std::uniform_int_distribution<unsigned int> ability_range;

    static std::uniform_real_distribution<float> possibility;
};
```

멤버 변수와 멤버 함수들을 protected 및 public으로 구분하여 외부에서의 접근을 제어하고 데이터와 함수를 하나로 묶어 관리하고 virtual 키워드를 사용하여 동적 바인딩을 지원하며, 런타임에 적절한 함수가 호출되도록 함.



## 2. Squid Game 에 사용된 C++

### 4. new 와 delete를 사용한 동적 할당

```
std::list<Player*> players;
for (int i = 0; i < 456; ++i)
{
    players.push_back(new Player(i + 1));
}

std::vector<Game*> games;
games.push_back(new RedLightGreenLight(20));
games.push_back(new RPS());
games.push_back(new MarbleGame());

// ...

for (auto game : games)
{
    delete game;
}
```

# 4. 구현된 Squid Game 설명

플레이어들은 개인 능력을 갖고 있음

Player::agility : [0:100]

Player::fearlessness : [0:100]

Ex) The red light green light game

- 게임 내에서, 지정된 경기장의 크기는 1000

  - const unsigned int RedLightGreenLight::distance

- 플레이어에게 20턴을 부여

  - const unsigned int RedLightGreenLight::turn

플레이어들은 매 턴마다 전진

플레이어의 이동 거리는 플레이어의 agility와 fearlessness에 따라 다름

이동 거리 = agility + random distance + fearlessness 따른 추가 거리

- random distance: [0:10]

- 두려움에 따른 추가 거리

  - » 플레이어의 높은 두려움 수치는 추가 보너스 거리를 얻을 확률을 높임.

[todayoneul/Squid-Game \(github.com\)](https://github.com/todayoneul/Squid-Game)



## 5. 아쉬운 점 / 개선해야 할 점

1. 저번에 진행했던 Snake Game처럼 콘솔창을 제어하는 기능을 넣고 싶었음. -> players에 1명만 남을 때까지 게임을 진행 한 후, 콘솔창 모두 지우고 우승자 축하 창 띄우기
2. 구슬 게임을 구현하기 위해 약간 스파게티 코드가 된 감이 없지 않아 있음.
3. 오징어 게임에 나오는 설탕 뽑기, 줄다리기, 징검다리 건너기 게임은 구현 할 수 있을 것 같아 이 게임들을 추가해보고 싶음.
4. GUI를 활용하는 방법을 공부하는 것도 재밌을 것 같음.

# THANK YOU

C++로 Squid Game 게임 만들기  
-이규환

구현 소스코드 : [todayoneul/Squid-Game \(github.com\)](https://github.com/todayoneul/Squid-Game)