

보드게임 'YACHT DICE' 프로그래밍

(다전공 포트폴리오)

2020105326 응용물리학과 윤인희

1. 주제 선정 배경

유튜브에서 우연히 닌텐도 스위치의 게임 51 Worldwide Games에 수록된 보드게임 중 하나인 Yacht Dice를 플레이 하는 것을 보았습니다. 간단한 룰을 가진 게임이었지만, 그 사람의 운과 전략에 따라서 게임 결과가 크게 바뀌는 시스템을 가지고 있었고, 이에 저 뿐만 아니라 많은 사람들이 Yacht Dice라는 게임 자체에 흥미를 가지게 되었습니다. 그러나, 이 게임을 즐길 수 있는 방법은 매우 한정적이고 적지 않은 비용이 들어가는 방법들이었습니다. 이에 저는 제가 작년 2학기 교양과목에서 배운 C++지식을 활용해 직접 프로그래밍을 하여 이 게임을 구현해야겠다는 생각을 하게 되어 포트폴리오 주제를 위와 같이 선정하게 되었습니다.

2. 결과물 요약

https://github.com/mu08014/Yacht_Dice

개발한 프로그램은 github에 업로드 되어 오픈소스로 공유하고 있습니다.

프로젝트의 파일은 Visual Studio 2019로 생성하였습니다.

```
#include <iostream>
#include <iomanip>
#include <string>
#include <ctime>
using namespace std;

int InfoNum;
```

그림 1. 헤더파일 및 InfoNum 변수선언

이 결과물에서 사용한 헤더파일은 다음과 같습니다. 특별한 점은 'Yacht Dice' 는 주사위를 굴리는 방식으로 게임이 진행되므로 주사위를 구현하기 위해 <ctime>을 사용했다는 것입니다. 또한, InfoNum은 이 값에 따라 매뉴창에서 이동할 곳이 바뀌어서, 여러 함수에서 사용되기 때문에 global variable로 선언했습니다.

```

//메뉴창
void MenuInfo() {
    cout << "-----\n";
    cout << setfill(' ') << '|' << left << setw(38) << " 1. Game Information" << "\n";
    cout << setfill(' ') << '|' << left << setw(38) << " 2. The Method of Score Calculation" << "\n";
    cout << setfill(' ') << '|' << left << setw(38) << " 3. Start Game" << "\n";
    cout << setfill(' ') << '|' << left << setw(38) << " 4. Quit Game" << "\n";
    cout << "-----\n";
    cout << "원하는 실행 번호를 입력하세요 --> ";
    cin >> InfoNum;
    cout << "\n";
}

//메뉴
void Menu() {
    while (1) {
        MenuInfo();
        if (InfoNum == 1) {
            Menu1();
        }
        else if (InfoNum == 2) {
            Menu2();
        }
        else if (InfoNum == 3) {
            Menu3();
            break;
        }
        else if (InfoNum == 4) {
            break;
        }
        else {
            cout << "오류 : 1에서 4 사이의 정수값을 입력해주세요!\n";
        }
    }
}

```

그림 2. Menu 함수와 Menu창 함수

MenuInfo에서 InfoNum값을 입력받고, Menu함수에서 MenuInfo를 실행시킴으로써 받은 InfoNum값에 따라 다양한 메뉴로 이동할 수 있게 만들었습니다. 또한 게임을 직접 실행시키는 Menu3이 실행 된 후에는 게임이 종료되도록 했습니다.

Menu1과 Menu2에는 게임에 대한 전반적인 설명 및 게임 룰에 대해 적혀 있습니다. 모두 cout으로 문자를 출력하도록 만들었기 때문에 자세한 코드는 생략하도록 하겠습니다.

```

//주사위 발생
void RunDice(int& dice1, int& dice2, int& dice3, int& dice4, int& dice5) {
    srand(time(NULL));
    dice1 = (rand() % 6) + 1;
    dice2 = (rand() % 6) + 1;
    dice3 = (rand() % 6) + 1;
    dice4 = (rand() % 6) + 1;
    dice5 = (rand() % 6) + 1;
}

```

그림 3. RunDice함수

주사위를 실행시키는 함수입니다. 계속해서 변화하는 시간값을 시드 값으로 받아서 5개의 주사위를 굴리는 것을 구현했습니다. rand함수를 통해 0~1 사이의 랜덤값이 출력되면 6으로 나눈 나머지로 0~5 사이의 정수가 나오고, 여기에 1을 더해 주사위의 숫자인 1~6의 값으로 나오게 각각의 변숫값을 설정했습니다.

```
//주사위 던진 결과 표시
void Repeat(string player) {
    int A = 0;
    int B = 0;
    int C = 0;
    int D = 0;
    int E = 0;
    int F = 0;
    int G = 0;
    int H = 0;
    int I = 0;
    int J = 0;
    int K = 0;
    int L = 0;
    int point = 0;
    int bonus = 0;
    int turn = 1;
    int Dice1, Dice2, Dice3, Dice4, Dice5;
    int play;
    while (1) {
        if (turn >= 13) {
            break;
        }
        else {
            int n = 1;
            while (1) {
                char decision;
                cout << player << "님 준비 되셨으면 1을 입력해 주세요!-> ";
                cin >> play;
                if (play == 1) {
                    cout << player << "님이 주사위를 던집니다. (" << turn << "번째 턴, " << n << "번째 주사위)";
                    Dice1 = rand() % 6 + 1;
                    Dice2 = rand() % 6 + 1;
                    Dice3 = rand() % 6 + 1;
                    Dice4 = rand() % 6 + 1;
                    Dice5 = rand() % 6 + 1;
                    A = Dice1; B = Dice2; C = Dice3; D = Dice4; E = Dice5;
                    point = A + B + C + D + E;
                    cout << "점수: " << point << endl;
                    n++;
                }
            }
            turn++;
        }
    }
}
```

그림 4.1. Repeat함수

다음은 주사위를 던진 결과를 표시하는 함수입니다. 이 게임에서는 나온 주사위의 값에 따라 기록한 족보를 만들어야 하는데, 각 족보는 한번밖에 사용하지 못하기 때문에 A~L의 변수를 선언해 족보를 사용하면 이 값이 1로 바뀌게 코딩해놨습니다. 변수 turn을 이용해 모든 족보가 채워졌을 때, 게임이 종료되게 설정했으며, 변수 play를 입력 받아 플레이어가 준비가 되었을 때 주사위를 굴리게 구현했습니다.

```

RunDice(Dice1, Dice2, Dice3, Dice4, Dice5);
cout << "결과 : " << Dice1 << " " << Dice2 << " " << Dice3 << " " << Dice4 << " " << Dice5 << "\n\n";
if (n >= 5) {
    break;
}
else {
    cout << "결정하시겠습니까? (Y or N) -> ";
    cin >> decision;
    cout << "\n";
    if (decision == 'Y') {
        break;
    }
    else {
        n = n + 1;
    }
}
}
else {
    cout << "1을 입력해주세요!\n";
}
}
Score(Dice1, Dice2, Dice3, Dice4, Dice5, point, A, B, C, D, E, F, G, H, I, J, K, L, player, bonus);
turn = turn + 1;
}
}
if (bonus >= 63) {
    point = point + 35;
    cout << "Aces ~ Sixes의 총 합이 63점이 넘었으므로 보너스 35점!\n";
}
cout << "게임종료! " << player << "님의 최종 점수는 " << point << "점 입니다.\n";
}
}

```

그림 4.2. Repeat함수

원래 게임에서는 5개의 주사위를 굴렸을 때, 플레이어가 직접 자신이 원하는 결과가 나온 주사위를 5개 중에 선택할 수 있지만, 저의 코딩실력 부족으로 그것 까지는 구현하지 못하고, 3번 주사위를 굴리는 대신, 5번으로 주사위 굴리는 횟수를 늘리는 방식으로 원래의 게임과 각 족보가 나올 확률을 비슷하게 하고자 했습니다. 또한, bonus 를을 구현하고자, bonus라는 변수를 새로 도입하여 Aces~Sixes의 합이 63이 넘으면 보너스 점수를 얻도록 하였습니다.

```

//3. Start Game
void Menu3() {
    string player;
    cout << "게임 시작!\n";
    cout << "플레이어의 이름을 지정해주세요 --> ";
    cin >> player;
    Repeat(player);
}

```

그림 5. Menu3

위의 Repeat 함수를 실행시켜주는 함수가 Menu3입니다. Menu함수에 의해 실행되며 위 함수들에서 보였던, player의 값을 정해주는 함수입니다.

```

//족보 선택하기
void Score(int dice1, int dice2, int dice3, int dice4, int dice5, int& point, int& A, int& B, int& C, int& D, int& E, int& F, int& G, int& H, int&
int score;
while (1) {
    cout << "< 족보 표 > 사용 가능 족보 0, 사용 불가능 족보 1\n";
    cout << "Aces : " << A << "\n";
    cout << "Deuces : " << B << "\n";
    cout << "Threes : " << C << "\n";
    cout << "Fours : " << D << "\n";
    cout << "Fives : " << E << "\n";
    cout << "Sixes : " << F << "\n";
    cout << "choice : " << G << "\n";
    cout << "4 of a Kind : " << H << "\n";
    cout << "Full House : " << I << "\n";
    cout << "Little Straight : " << J << "\n";
    cout << "Big Straight : " << K << "\n";
    cout << "Yacht : " << L << "\n";
    cout << "점수를 얻을 족보를 선택하세요. (족보에 대한 정보를 얻으려면 0을 입력하세요. 괄호 안의 숫자 입력) -> ";
    cin >> score;
    if (score == 0) {
        Menu2();
    }
    else {
        if (score == 1) {
            if (A == 1) {
                cout << "\n이미 사용한 족보\n";
            }
            else {
                Aces(dice1, dice2, dice3, dice4, dice5, point, bonus);
                A = 1;
            }
        }
    }
}

```

그림 6. Score함수

플레이어에게 각 족보의 사용 여부를 알려주고, 주사위 값에 따라 플레이어가 사용할 족보를 결정할 수 있는 Score 함수입니다. Int&를 통해 위 함수에서 사용한 변수의 값이 실제로도 바뀌게 함으로써, 플레이어의 스코어가 계산될 수 있게 했습니다. 그림 6의 아래 코드 부분은 Aces~Yacht까지의 주사위 값에 따라 각 족보의 점수를 계산해 주는 함수들입니다. 그 중 하나인 Full House 함수를 소개하겠습니다.

```

//Full House
void Full(int dice1, int dice2, int dice3, int dice4, int dice5, int& point) {
    int dice[] = { dice1, dice2, dice3, dice4, dice5 };
    int a = 0, b = 0, c;
    for (int i = 1; i <= 6; i++) {
        if (count(dice, dice + 5, i) == 3) {
            a = a + 1;
        }
    }
    for (int i = 1; i <= 6; i++) {
        if (count(dice, dice + 5, i) == 2) {
            b = b + 1;
        }
    }
    if (a != 0 && b != 0) {
        c = dice1 + dice2 + dice3 + dice4 + dice5;
    }
    else {
        c = 0;
    }
    point = point + c;
}

```

그림 7. Full함수

족보 Full House를 구현한 Full함수는 결정한 5개의 주사위 값을 dice라는 변수의 Array로 받습니다. 이를 이용해 count함수를 사용하여 2 2 3 3 3 과 같이 각 주사위의 숫자가 겹치는 수가 2개와 3개일 때, a와 b의 값이 모두 1이되고, 이에 따라 c값이 주사위의 값을 모두 더한 값으로 변함으로써, Full House의 조건을 만족시킬 때만 주사위의 값을 모두 더한 값을 Score에 추가하는 함수입니다. 이와 같은 식으로 5의 숫자의 개수에 따라 점수가 결정되는 Fives 함수, 주사위의 수가 1 2 3 4 5가 나오면 30점을 추가하는 Little Straight, 5개의 주사위 숫자가 모두 같은 것이 나오면 50점을 추가하는 Yacht 등등, 총 12개의 족보 값을 함수로 구현하였습니다.

다음은 프로그램 실행 결과입니다.

```
-----  
1. Game Information  
2. The Method of Score Calculation  
3. Start Game  
4. Quit Game  
-----  
원하는 실행 번호를 입력하세요 -->
```

원하는 실행 번호를 입력하세요 --> 3

게임 시작!

플레이어의 이름을 지정해주세요 --> 윤인희

윤인희님 준비 되셨으면 1을 입력해 주세요!--> 1

윤인희님이 주사위를 던집니다. (1번째 턴, 1번째 주사위)

결과 : 6 4 5 5 1

결정하시겠습니까? (Y or N) -> N

윤인희님 준비 되셨으면 1을 입력해 주세요!--> 1

윤인희님이 주사위를 던집니다. (1번째 턴, 2번째 주사위)

결과 : 1 6 2 6 1

결정하시겠습니까? (Y or N) -> N

윤인희님 준비 되셨으면 1을 입력해 주세요!--> 1

윤인희님이 주사위를 던집니다. (1번째 턴, 3번째 주사위)

결과 : 5 5 6 6 1

결정하시겠습니까? (Y or N) -> N

윤인희님 준비 되셨으면 1을 입력해 주세요!--> 1

윤인희님이 주사위를 던집니다. (1번째 턴, 4번째 주사위)

윤인희님 준비 되셨으면 1을 입력해 주세요!-> 1
윤인희님이 주사위를 던집니다. (1번째 던, 4번째 주사위)
결과 : 6 5 5 1 5

결정하시겠습니까? (Y or N) -> Y

< 족보 표 > 사용 가능 족보 0, 사용 불가능 족보 1

Aces : 0
Deuces : 0
Threes : 0
Fours : 0
Fives : 0
Sixes : 0
choice : 0
4 of a Kind : 0
Full House : 0
Little Straight : 0
Big Straight : 0
Yacht : 0

점수를 얻을 족보를 선택하세요. (족보에 대한 정보를 얻으려면 0을 입력하세요. 괄호 안의 숫자 입력) -> 5
윤인희님의 현재 점수는 15입니다.

윤인희님 준비 되셨으면 1을 입력해 주세요!->

윤인희님 준비 되셨으면 1을 입력해 주세요!-> 1
윤인희님이 주사위를 던집니다. (12번째 던, 5번째 주사위)
결과 : 1 1 6 2 5

< 족보 표 > 사용 가능 족보 0, 사용 불가능 족보 1

Aces : 1
Deuces : 1
Threes : 1
Fours : 1
Fives : 1
Sixes : 1
choice : 1
4 of a Kind : 1
Full House : 1
Little Straight : 0
Big Straight : 1
Yacht : 1

점수를 얻을 족보를 선택하세요. (족보에 대한 정보를 얻으려면 0을 입력하세요. 괄호 안의 숫자 입력) -> 1

이미 사용한 족보

< 족보 표 > 사용 가능 족보 0, 사용 불가능 족보 1

Aces : 1
Deuces : 1
Threes : 1
Fours : 1

< 족보 표 > 사용 가능 족보 0, 사용 불가능 족보 1

Aces : 1
Deuces : 1
Threes : 1
Fours : 1
Fives : 1
Sixes : 1
choice : 1
4 of a Kind : 1
Full House : 1
Little Straight : 0
Big Straight : 1
Yacht : 1

점수를 얻을 족보를 선택하세요. (족보에 대한 정보를 얻으려면 0을 입력하세요. 괄호 안의 숫자 입력) -> 10
윤인희님의 현재 점수는 82입니다.

게임종료! 윤인희님의 최종 점수는 82점 입니다.

C:\Users\#82105\Desktop#경희대학교 응용물리학과#다전공 준비#Yacht_Dice#Debug#Yacht_Dice.exe(프로세스 25756개)이(가)되었습니다(코드: 0개).
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...

그림 8. Yacht_Dice 실행 결과

3. 기대 효과 및 향후 계획

완벽하게 Yacht_Dice의 게임 룰을 그대로 이식하지 못해 원래 게임의 완벽한 대응이 될 수는 없겠지만, 나름 저만의 규칙대로 게임을 재미있게 구현한 것 같습니다. 무엇보다 제가 스스로 만들었기 때문에 무료로 언제든지 플레이할 수 있다는 점에서 만족스럽습니다.

이 게임을 더욱 발전시키기 위해서는 두 가지를 해야 한다고 생각합니다. 첫 번째는, 주사위를 굴릴 때나, 점수판에 점수를 기록할 때 시각적으로 직관적인 모션을 넣는 것입니다. 아직 움직임을 구현할 수 있는 프로그램을 짜는 방법을 모르기 때문에 굉장히 딱딱한 형식의 게임이 되었다고 생각합니다. 두 번째는, 플레이어의 수를 직접 설정할 수 있게 하여 게임 내부에서 플레이어들이 번갈아가면서 주사위를 던질 수 있게 하는 것입니다. 사실 여기에 대한 시도가 있기는 했으나 저의 능력의 한계로 실패하였습니다. 나중에 더욱 코딩 실력을 키워 다시 시도해 보고 싶습니다.