**소프트웨어 프로젝트 : Term Project**

****

20192293 최우형

**Term Project**

제출마감: 2020년 6월 14일(일)

제출방법: eclass 에서 제출

이번 학기에 학습한 JAVA 언어를 이용하여 개인별 프로젝트를 수행 하시오. 프로젝트 주제는 취미와 관련된 스포츠 게임이며 실재 스포츠를 하는 것과 유사하게 시나리오를 구성하고, 게임을 진행 및 결과를 보여주는 프로그램을 만드시오.

1. **프로젝트 주제 선정 : 아래 볼링 이나 골프 또는 동등 수준의 스포츠 게임 프로그램**

* **볼링 게임**

1. **과제 제출 파일**
   * Java 프로그램(.Java 파일)
     + 프로그램 작성 규칙에 맞게 작성(들여 쓰기, Comment 등등),
   * 프로그램 도큐먼트(.docx, .hwp, 또는 .pdf 파일등)
     + 게임의 규칙이나 기능 구현에 대한 상세 설명
     + 차별화 또는 강조하는 아이디어 제시
     + 프로그램 입력, 출력 결과 캡쳐 등등
2. **평가 방향**

* 과제 제출물의 완성도와 난이도로 A, B, C 로 평가

예) C: 완성도와 난이도가 낮은 경우

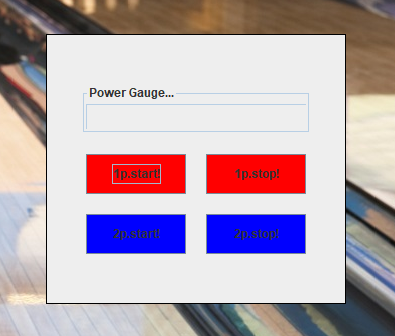
B: 평균수준, 완성도 또는 난이도가 높은 경우

A: 완성도와 난이도가 높은 경우

**프로그램 이름 : Bowling Adventure**

* + - **게임 소개 :** 게임의 규칙이나 기능 구현에 대한 상세 설명

**이 게임은 두 명이서 할 수 있는 게임 입니다. 빠르게 움직이는 파워 게이지를 조절하여 보다 많은 핀을 넘길 수 있도록 하십시오. 점수 체계는 변경된 점수 체계를 따릅니다.**

****

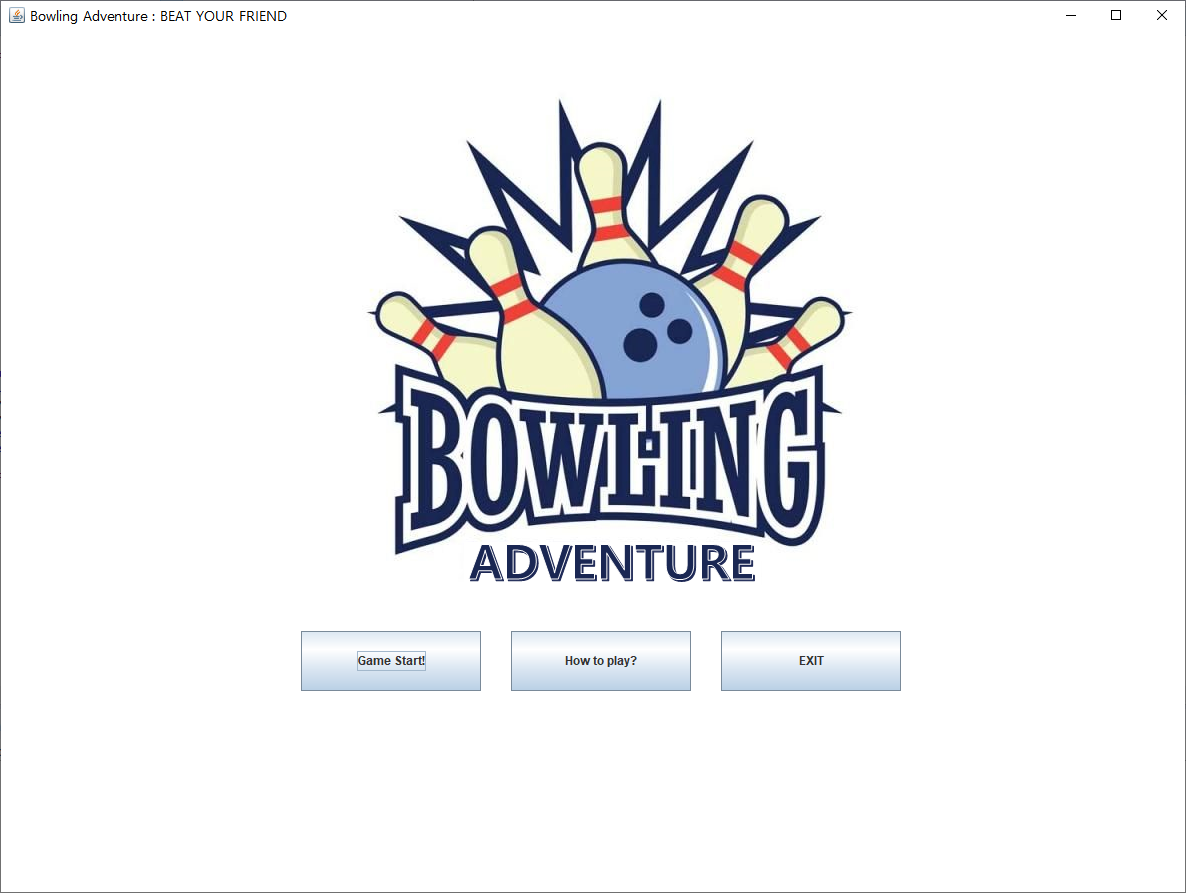
**플레이어별로 start, stop버튼이 있습니다. 스타트 버튼을 누르면 게이지가 변화합니다. Stop을 누르면 게이지가 멈추고, 게이지에 따라 볼링핀을 넘어뜨립니다.**

**1p와 2p는 독립적이어서 순서는 정해져 있지 않지만, 번갈아 가면서 하는 것을 추천 드립니다.**

**10\* value / 100 : 처음 넘어뜨리는 볼링 핀의 개수**

**(나머지+4)\* value / 100 : 스트라이크가 아닐 경우, 두번째 넘어뜨리는 볼링 핀의 개수**

**(나머지+4) \* value / 100 이게 나머지보다 클 경우 -> 스페어 처리**

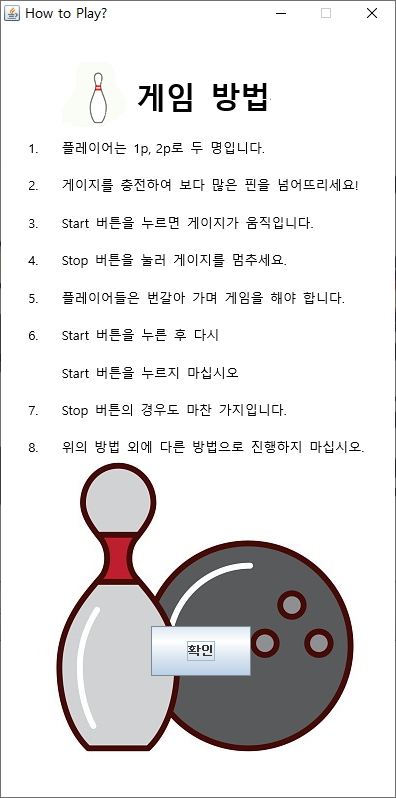
****

**첫페이지 : JFrame 에 JPanel을 넣어 구현, 배경화면을 삽입하고 3개의 버튼을 넣었음.**

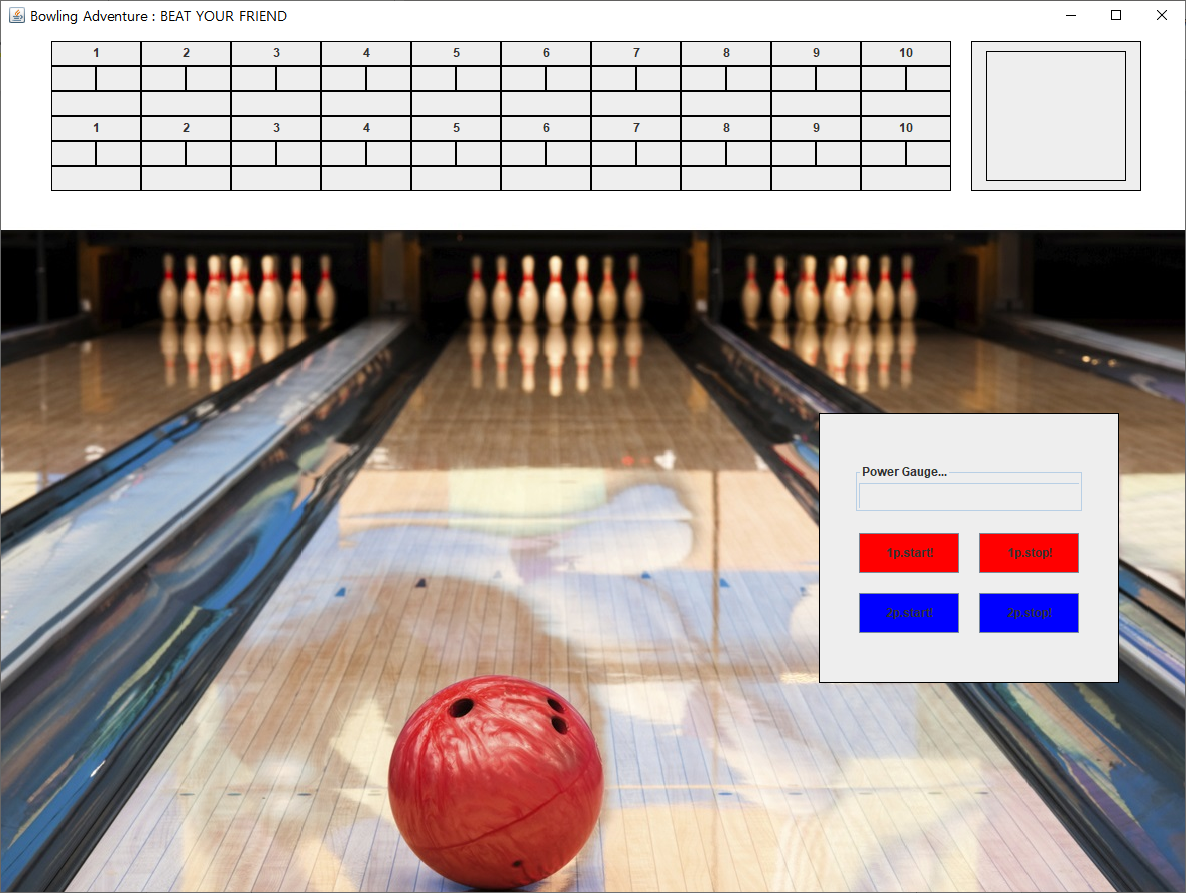
**Game start버튼을 누르면 이 패널은 보이지 않게 되고, 두번 째 페이지가 생성되어 프레임에 나타납니다.**

**How to play버튼을 누르면 게임 방법 설명 창이 나타납닏.**

**Exit 버튼을 누르면 프로그램을 종료합니다.**

****

**게임 방법 창 : 새로운 JFrame 을 생성하여 구현, 확인 버튼을 넣어놓았으며 확인 버튼을 누르면 dispose()함수를 통해 게임 방법 창을 종료함**

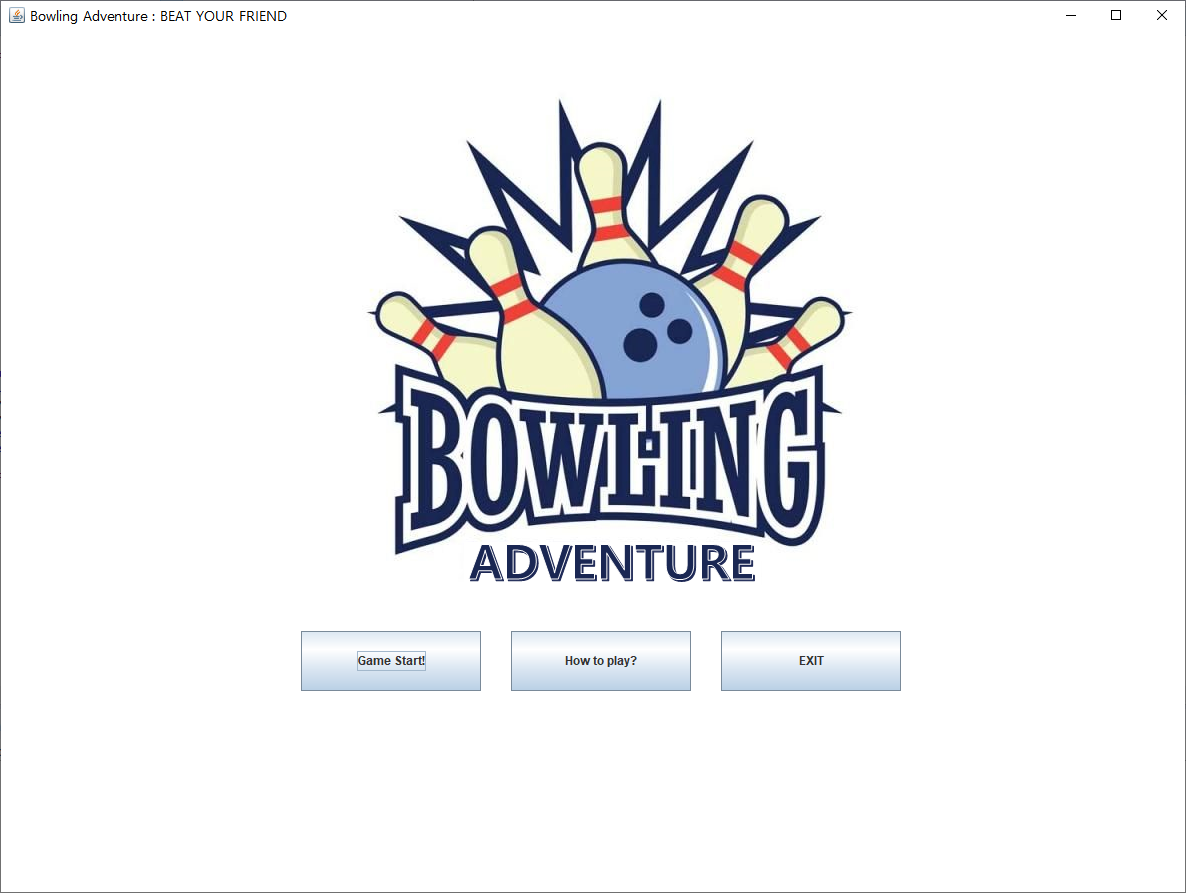


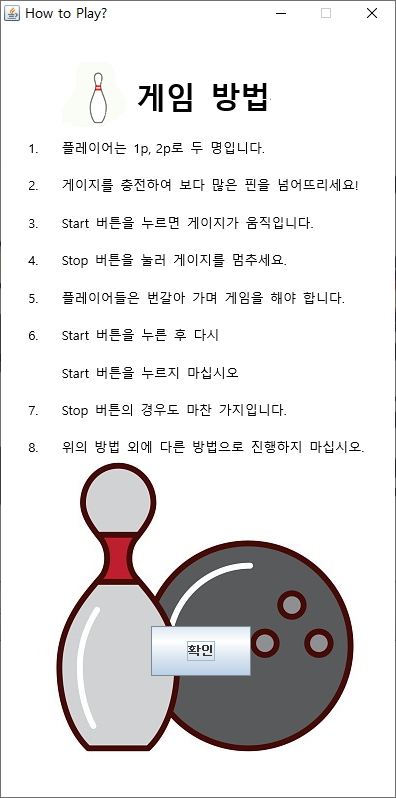
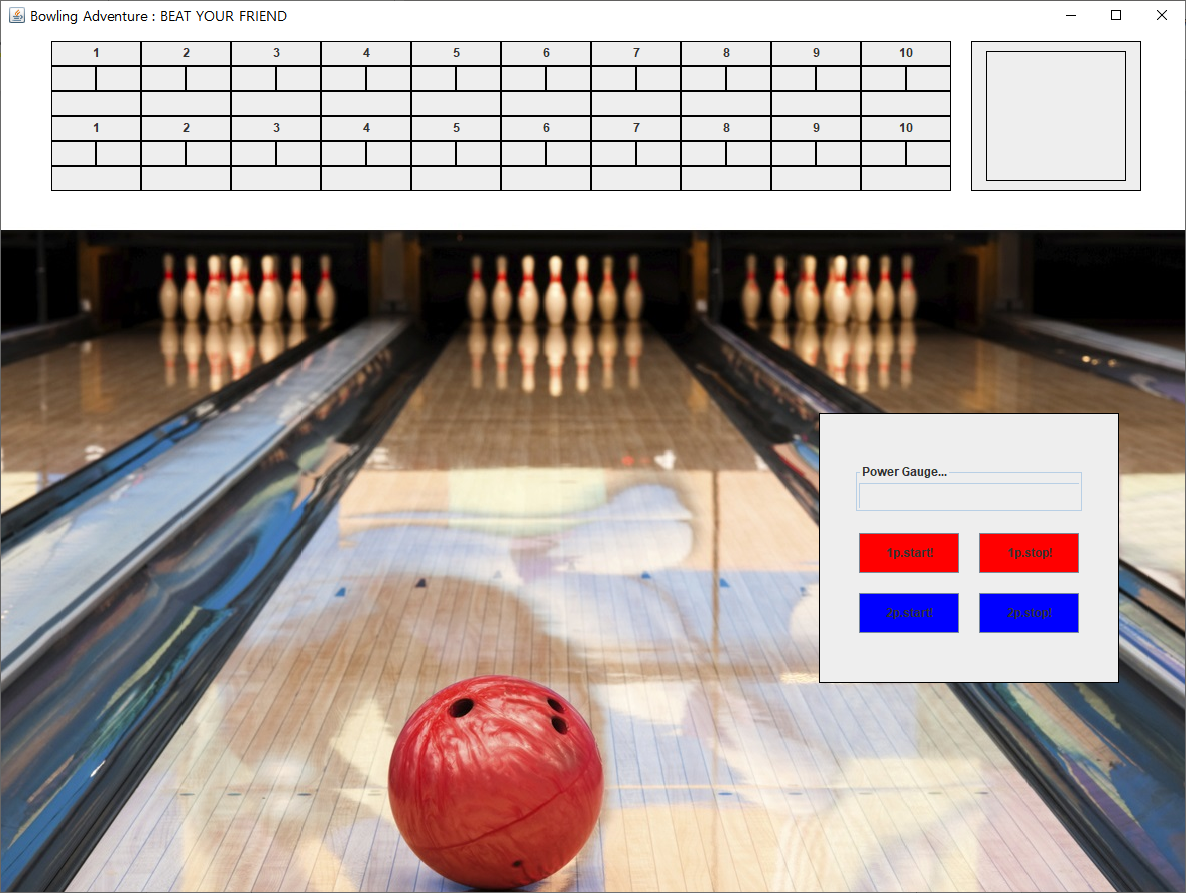
**게임 플레이 창 : JPanel을 통해 큰 패널을 만들고 ScorePanel 에 1,6 의 gridlayout을 설정하고 girdlayout (10,1), (20,1)로 이루어진 6개의 패널을 더 만들어 add해줍니다. 추후 거기에 JLabel 을 넣어 점수를 표시합니다.**

**GamePanel은 우측 상단에 있는 패널로서, 패널 안에 JLabel 이 있고, 거기에는 투구할 때마다 투구의 결과 ex) 넘어뜨린 핀 개수, 스트라이크, 스페어 여부등을 표시해 줍니다.**

**ControlPanel 에는 게임 조작을 위한 버튼들과 progressbar를 넣어주었습니다. Progressbar는 게이지를 표현하기 위해 추가 되었습니다. timerThread를 통해 progressbar 의 수치를 0~100까지 왔다 갔다 하게 만들고 stop버튼을 눌렀을 때 멈춤과 동시에 수치를 받아옵니다.**

* + - 차별화 또는 강조하는 아이디어 제시

****

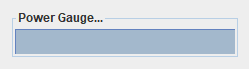
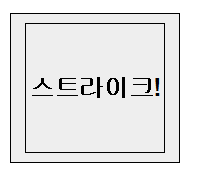
**** 

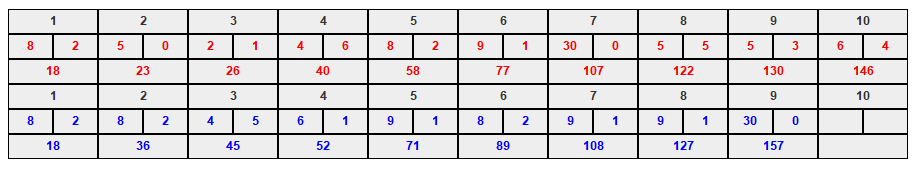
위 화면은 차례대로, 메인화면, 방법 설명화면, 게임 플레이 화면입니다.

GUI를 통해 게임을 만들었습니다. 확률로서 점수만 표시 되는 것이 아니라 게임 플레이어들이 어떻게 하느냐에 따라 점수가 달라집니다. 스트라이크와 스페어가 나오는 빈도수 : 본인이 직접 플레이 해보면서 알고리즘을 만들었고, 게임의 난이도와 점수 부여 방식은 적절한 것으로 판단됩니다. 플레이어의 실력과 집중력에 따라 다르겠지만 보통 적당한 노력을 들이었을때 150점 정도가 나옵니다. 승패의 요인은 스트라이크를 얼마나 많이 하느냐 인데, 스트라이크를 치기는 쉽지 않습니다.

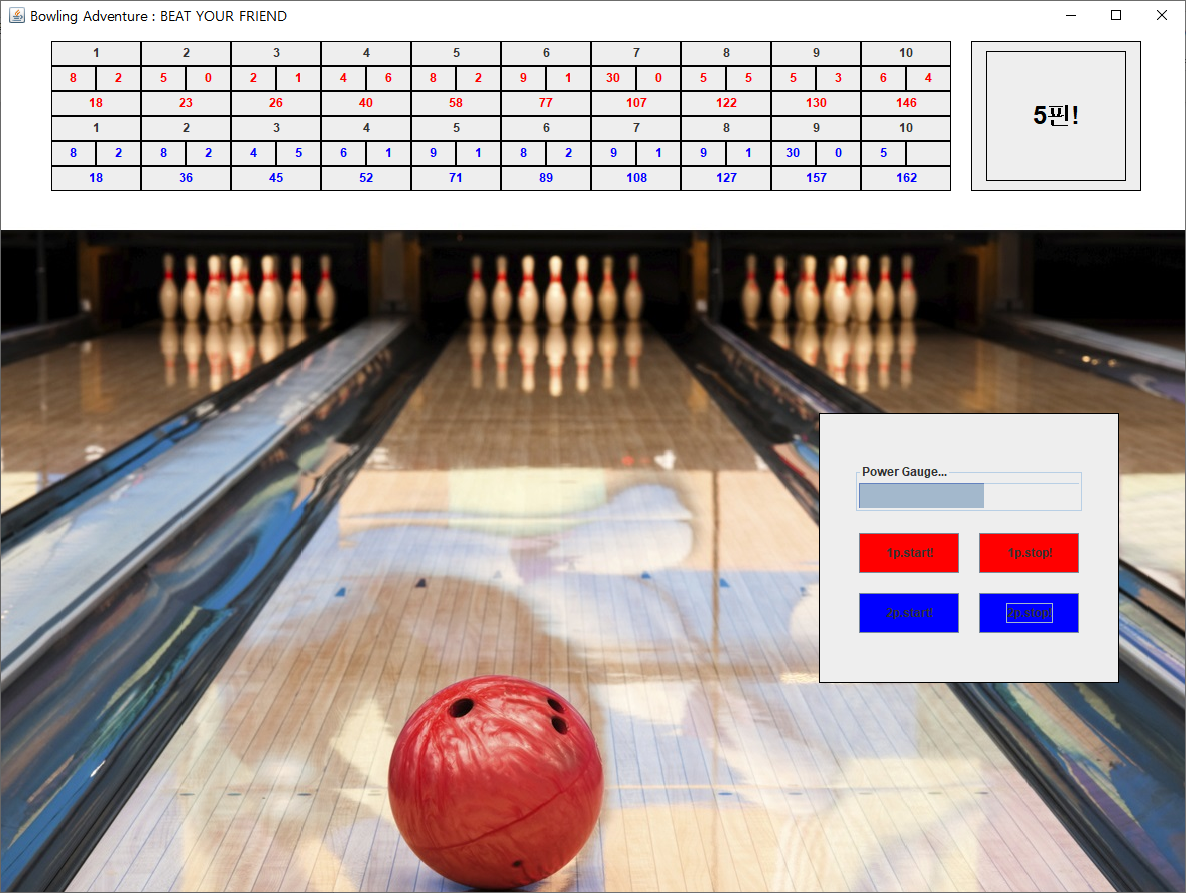
게임화면과 게임 설명 창에 이미지를 그려넣어 보기에 좋게 만들었습니다.

* + - 프로그램 입력, 출력 결과 캡쳐 등등

** 스트라이크를 쳤을 때의 파워 케이지와 점수 알림입니다.**

****

**1p 10번째 프레임까지 게임을 진행한 결과입니다. 1p는 빨간 글씨, 2p는 파란 글씨로 나타내었습니다.**



**전체적인 모습입니다.**

**여러 클래스들을 만들어 여러 객체들을 생성하고, 그 객체들이 정보를 잘 전달받고 상호작용하는 프로그램을 만들고 싶었습니다. 각 객체들이 독립적으로 잘 존재하여 어느 클래스의 소스를 수정하더라도 다른 클래스의 소스에는 영향이 가지 않도록 잘 짜여진 것을 모듈화가 잘 되었다고 하는 것으로 알고 있습니다. 계획은 있었지만, 막상 프로그램을 짜면서 깊어지는 객체들의 관계에 정보를 어떤 방식으로 넘겨주고 어떤 방식으로 전달 받을지, 고민하는 시간이 많았습니다. 필드에 어떤 배열들과 변수들을 놓고, 어떤 메소드들을 만들고, 생성자에는 어떤 파라미터를 받아야 하는지,,, 나중에 다른 프로젝트에서 새 프로그램을 만들 때에는 보다 모듈화가 잘 되어있는 체계적인 프로그램을 만들 수 있지 않을까 생각해 봅니다.**