

# 소프트웨어 프로젝트 Project

과 목 명 : 소프트웨어 프로젝트

과 제 명 : 볼링 게임 구현하기

사용언어 : 자바

학 과 : 심리학과

학 번 : 20141963

이 름 : 박승완

제 작 일 : 20190607

담당교수 : 이남규교수님

## ■ 볼링 게임

- 볼링은 한 게임당 10개의 프레임으로 이루어져 있다.
- 볼링 게임에 대해 조사하고 10개 프레임까지의 게임에 필요한 볼링 점수 계산 방법에 조사하고 기능을 구현한다.
- <http://cafe.daum.net/goodfreespace/XkXc/2099?q=%EB%B3%BC%EB%A7%81%20%EC%A0%90%EC%88%98%20%EA%B3%84%EC%82%B0%EB%B2%95>
- 선수의 구성은 2명으로 한다.
- 선수들의 점수 평균
  - 초보자(100), 중급(150), 상급(200)
  - Random number와 확률을 이용하여 스페이처리나 스트라이크 확률을 계산하여 평균에 근접하도록 나오게 한다.

## 게임의 규칙 :

기본적인 구동 방법은 GUI 좌측 부분에 적어두었다.

상세히 기술하자면,

1. GUI 상단의 사용자 이름과 난이도를 **각각** 선택하고 엔터 버튼을 **각각** 누른다.
  - 1-1. 하단의 내기를 적는 기능을 이용하고 fix하여 경기가 시작되면 내기 내용을 수정할 수 없게 한다.
2. GUI 우측의 player별 ROLL 버튼을 **차례로** 누른다.
  - 따라야할 규칙 : player1 , 2가 차례로 경기가 종료될 때까지 버튼을 눌러야 한다. Strike가 나와도 버튼은 눌러야 한다.
3. 주의사항 : 이 때 순서에 맞지 않는 버튼을 누르거나 여러 번 버튼을 누르는 행위는 삼가야 한다.
4. 버튼을 누르면 나오는 점수들을 확인하며 **볼링게임을 즐기면 된다.**

## 기능의 구현 :

크게 두 개의 클래스로 구성되어있다. **Game**과 **Modify**이다.

Game은 GUI와 게임의 진행을 담당한다.

Modify는 볼링게임의 기능적인 측면을 담당한다.

Game에 대해 상세히 보자면, 메인함수에서는 GUI를 실행하고 Modify의 main함수를 실행한다.

**GUI**는 전체적으로는 BorderLayout을 사용하여 North, South, Center, East, West에 각각 panel을 두었다.

North는 플레이어의 정보를 입력 받는데 사용되고 South는경기를 진행하기 전 내기를 정하여 경기 이후에 이를 이용하여 벌칙이나 상금을 받을 수 있다. West는 게임을 위한 가이드, East는 공을 던지는 버튼을 두었다. 그리고 Center는 점수 및 코멘트를 표시하는 칸으로 두었다.

Modify의 main함수는 GUI가 실행되면 자동으로 실행되는데 일정부분까지 실행되도록 lock이 걸려있다. 이 lock을 처음으로 푸는 것은 상단 panel에서 2player의 정보가 입력될 때이다.(enter 버튼)

그리고 우측의 던지기 버튼이 눌리면 각각에 Modify에 있는 해당되는 위치의 lock을 풀도록 설정 해두었다. 추가적으로 Game함수에서 몇 번째 공이 던져지는지를 나타내는 counter변수도 1씩 증가하도록 했다.

특히 가운데 panel은 우측 panel의 공을 던지는 버튼이 눌린 후 lock이 풀린 Modify에서 점수 계산이 실행되어 update한 점수 정보를 static변수를 통해 받아서 JTable에 update하는 식으로 GUI상에 출력한다.

**Modify에 대해서 상세히 알아보자면 Modify는 크게 메인함수와 점수를 구하는 함수(3개), 점수를 계산하는 함수(scoreCal)로 총 3개의 기능으로 구성이 되어있다.**

메인함수는 공을 던진 횟수를 나타내는 바깥 for문과 player1, 2를 구분하는 내부 for문으로 구성되어있다. 각각의 player차례일 때, 던진 공이 짝수 번째일 때, 홀수 번째일 때, 그 프레임이 strike일 때로 각각 if문을 통해 분기한다.

홀수 번째 던진 공(randScore\_odd)과 짝수 번째 던진 공(randScore\_even)은 각각 다른 점수를 구하는 함수를 호출한다. 이를 통해 나온 결과값을 scoreboardUpper에 업데이트하고 이를 이용한다.(점수 판 윗줄 == 각 공의 점수를 나타냄)

그리고 이 점수를 "점수를 계산하는 함수(scoreCal)"를 통해 scoreboard에 업데이트 한다.(점수 판 아랫줄 == 각 frame의 점수를 나타냄)

추가적으로 마지막 10번째 프레임은 Strike거나 spare라면 추가적인 공을 한 번 더 던지기에 이를 위한 점수를 구하는 함수를 따로 사용한다.(randScore\_last)

이제 **점수를 구하는 함수**를 알아보자.

이 함수는 player의 수준에 따른 점수를 확률을 이용하여 반환한다.

홀수 횟수의 공을 담당하는 함수는 스페어 처리를 신경 쓸 필요가 없다.

그러기에 하수의 경우에는 40%확률로 0~4점, 35%확률로 5~8점, 15%확률로 9점, 10%

확률로 10점을 내도록 설정하였다. 중수는 각각 15, 30, 40, 15. 고수는 10, 25, 17, 28로 설정하였다.

짝수 횟수의 공을 담당하는 함수는 스페어 처리를 신경 써야 한다. (이미 strike라면 이 과정은 넘어간다.)

이 함수의 결과는 무조건 오픈이거나 spare처리가 되어서 나옴을 유의하자.

10에서 이전 홀수 번째 던진 공의 결과를 뺀 rest\_score를 이용하여 spare처리를 하거나 안하거나에 대한 확률을 수준별로 나눈다.

하수는 75%의 확률로 spare처리에 실패하고, 25%의 확률로 성공한다. 중수는 각각 30, 70. 고수는 10, 90을 확률로 설정했다.

**Note** : 이렇게 설정한 함수들은 여러 번의 시도 끝에 교수님께서 제시하신 실력별 점수인 100, 150, 200점을 기준으로 하되 게임의 재미를 위해 어느정도 오차는 허용한다.

마지막 점수를 구하는 함수는 19, 20번째에 던진 공의 점수를 확인하고 이에 따라 점수를 낸다.

마지막으로 **점수를 계산하는 함수**를 알아보자. (**scoreCal**)

이 함수는 계산한 frame의 점수를 scoreboard에 업데이트 하는 함수이다.

파라미터로 몇 번째 공인지 (count), 그 점수가 얼마인지 (result), frame별 점수를 적을 array (scoreboard), 공 별 점수가 들어있는 array (scoreboardUpper), 각 frame이 strike인지 표시하는 bool array (is\_strike), 각 frame이 spare인지 표시하는 bool array (is\_spare)를 받는다.

한 프레임에서 최고 점수를 내는 경우는 "스트라이크가 연속적으로 3번"이 나왔을 경우이다. 이 경우에는 2, 3, 4 번째 프레임에서 strike가 나왔다고 가정하면 네 번째 프레임의 던지기가 완료되었을 때에 두 번째 프레임의 점수가 채워진다. 이를 고려하여 초기 3 개의 프레임(1~6번째 공)에 대해서 따로 점수를 계산해주는 **예외처리**를 해주었다. 추가적으로 마지막 프레임의 첫 번째, 두 번째 공에 대해서도 예외처리를 하여 이 결과에 따라 21 번째 공을 던질지 여부를 결정한다.

예외를 제외한 점수의 계산은 일반적으로 전전 프레임의 스트라이크 여부, 전 프레임의 스트라이크 여부, 전 프레임의 스페어여부를 각각 홀수 번째 공과 짝수 번째 공에 따라 분기하여 처리한다.

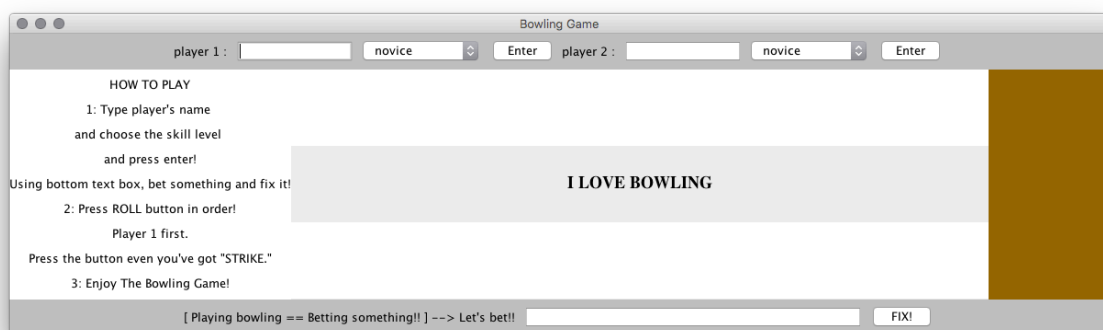
예를 들자면 9 번째 공의 점수를 계산할 때에 3 번째 프레임의 스트라이크, 4 번째 프레임의 스트라이크, 스페어 여부를 파악하여 3 번째 프레임이 스트라이크이고 4 번째 프레임이 스트라이크라면 3 번째 프레임의 점수를 계산하여 업데이트 해주고 스트라이크 여부를 FALSE로 만든다. 그리고 4, 와 5 번째 프레임은 아직 계산할 수 없기에 다음 공으로 넘어간다.

이와 같은 방식으로 구성되어 동작하는 프로그램이다.

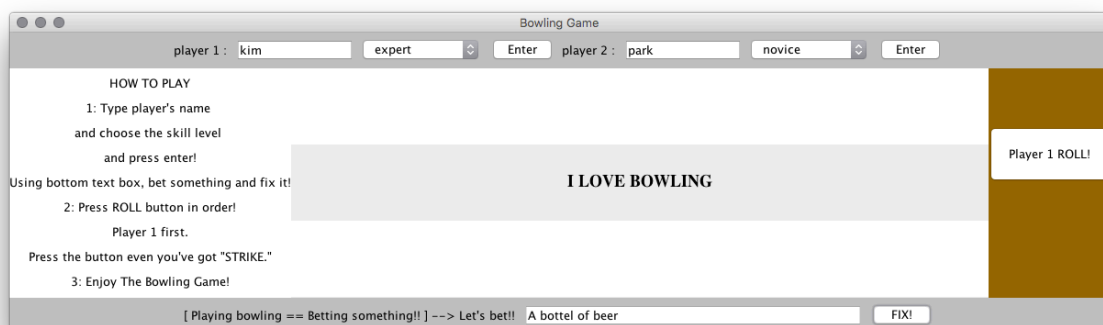
## 아이디어 :

1. GUI : 이 프로그램은 GUI를 이용하여 구동되므로 사용자들이 편리하게 이용할 수 있는 장점이 있다. 추가적으로 던진 공에 대한 점수, strike나 spare 여부를 글로 표현해주어 사용자들이 보다 쉽게 자신의 점수를 알 수 있도록 한다.
2. 진행적인 요소 : 사용자의 이름을 단순히 player 1, 2로 두는게 아닌 임의로 설정이 가능하게 하여 실제로 두 사람이 경기를 진행하는 느낌을 받을 수 있도록 하였다. 영어를 사용하여 한글을 모르는 외국인도 게임을 할 수 있도록 제작했다.
3. 흥미로운 요소 : 실제로 제작자는 볼링을 할 때마다 맥주 한 병, 게임비 등의 내기를 걸었고 게임을 했다. 그렇지만 이런 내기 내용이 경기가 진행되거나, 끝났을 때에 바뀌어서 흐지부지 되는 경우를 많이 겪기도 했다. 이러한 실제 경험을 바탕으로 프로그램 하단에 내기를 설정하고 경기가 시작되면 이를 수정할 수 없게 하는 기능을 추가했다.

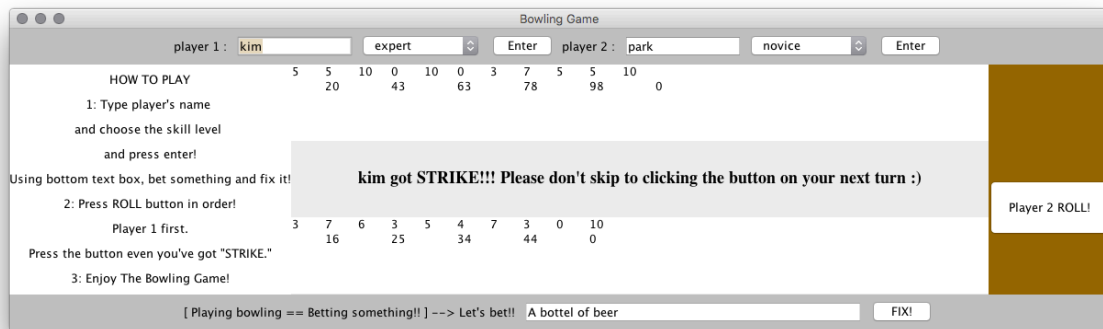
## 결과 캡처 :



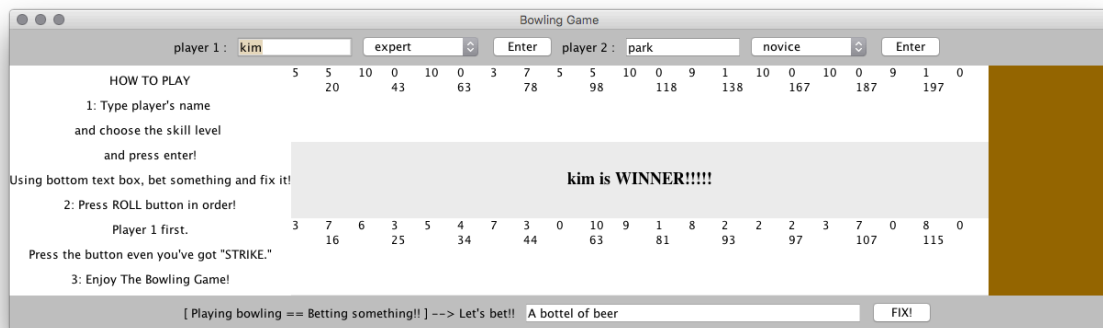
### <실행 초기 화면>



<이름, 실력, 내기를 기입한 후 - player2의 enter를 누르면 우측 player 1 roll 버튼이 보임>



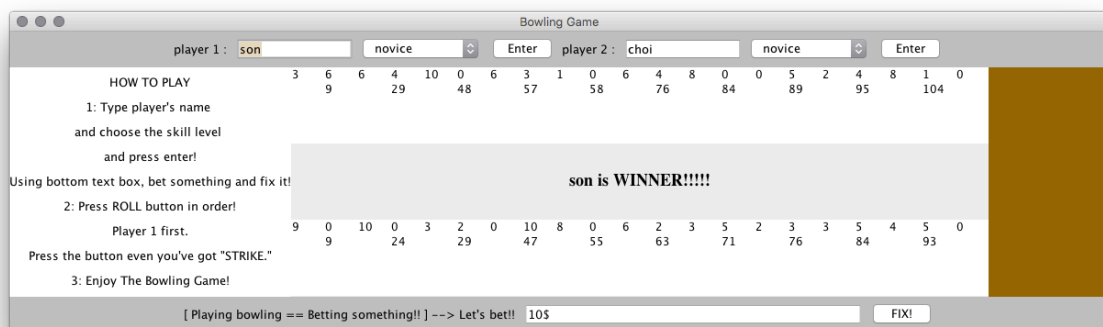
<실행 중반 - kim이 strike를 쳤을 경우.>



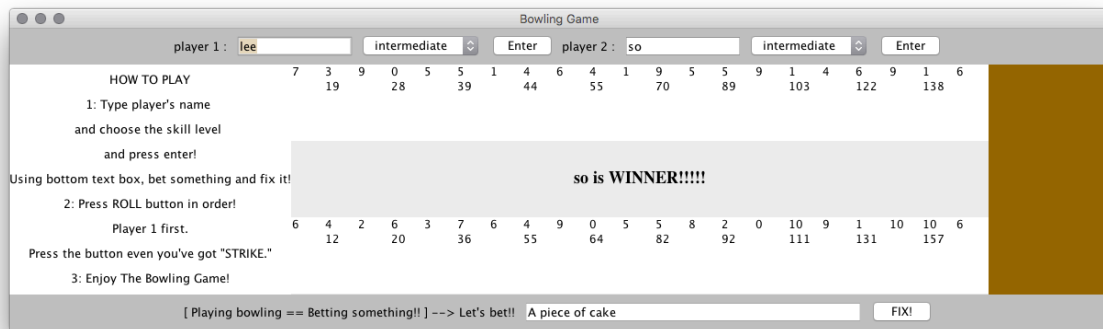
<게임 완료>

Expert – 197점, novice – 115점

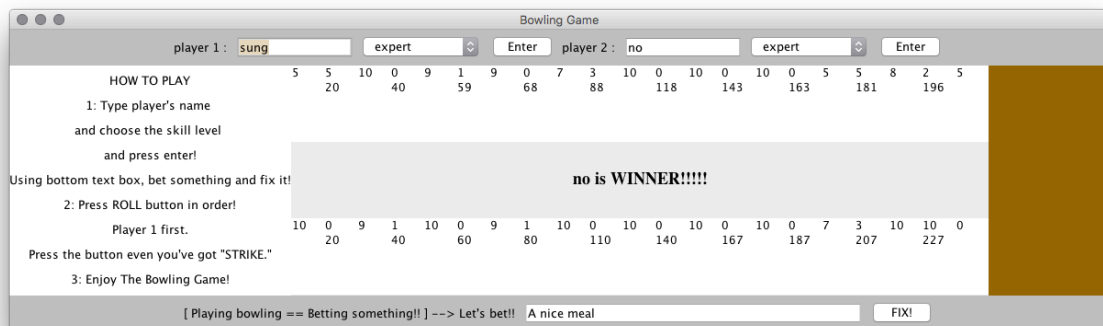
다른 실력으로 경기 진행 시의 결과



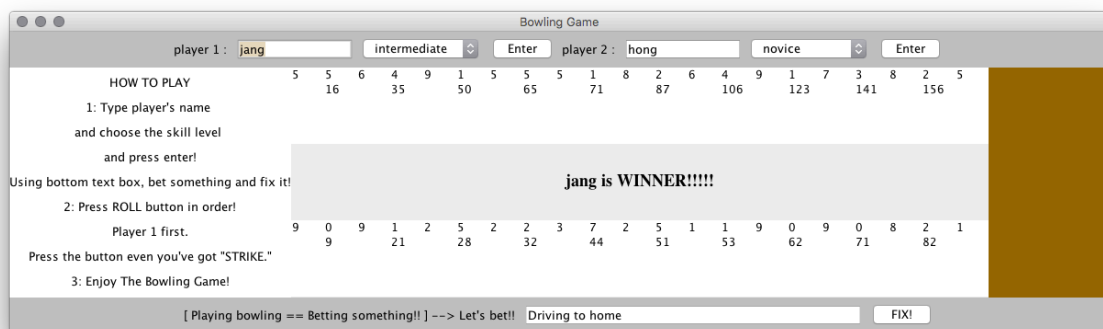
<하수들끼리의 경기 (104, 93)>



<중수들끼리의 경기 (138, 157)>



<고수들끼리의 경기 (196, 227)>



<중수와 하수의 경기 (156, 82)>

감사합니다.