Term Project - Bowling

소프트웨어프로젝트 05분반



학과: 경영학과

학번: 20134789

이름: 김진관

**1. 개요**

2020년 1학기 소프트웨어 학부 소프트웨어 프로젝트 텀프로젝트 주제로 볼링 게임을 선택하고 JAVA로 프로그래한다. 실제 스포츠를 하는 것과 유사하게 시나리오를 구성하고, 진행 및 결과를 보여준다.

Github: <https://github.com/kkole3897/Bowling-Game-with-Java>

**주요 특징**

- 볼링의 일반적인 규칙과 점수 계산 방식을 그대로 적용한다..  
- 2명의 플레이어가 10프레임을 돌아가면서 진행한다.  
- 점수판을 실시간으로 업데이트하면서 보여준다.  
- 플레이어의 등급을 나눠 등급마다 얻는 점수의 차이를 준다.

**규칙**

규칙과 관련된 사항은 Github Readme.md에 자세히 기록

**2. 계획**

- 요구사항 도출 및 분석을 통해 필요한 기능을 파악  
- 분석 결과를 이용해서 필요한 다이어그램을 그려보면서 모델링하고 소프트웨어 구성을 파악  
- 도출된 모델을 기반으로 코딩  
- 테스트

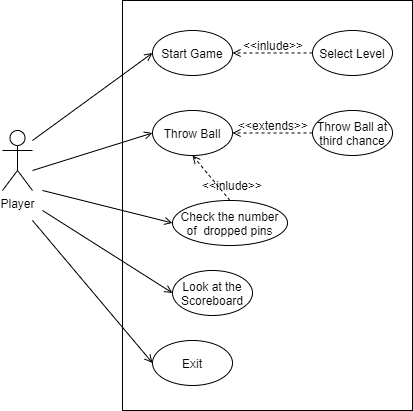
추가: 여건이 된다면 GUI도 적용

**3. 요구사항**

**3-1. 요구사항 도출**

|  |  |
| --- | --- |
| 기능 요구사항 | 1. 플레이어는 플레이어 등급을 정한다.  2. 플레이어가 볼링공을 던진다.  3. 1~9프레임까지는 최대 2번의 기회가 주어진다.  4. 10프레임은 최대 3번의 기회가 주어진다.  5. 10프레임의 첫번째 스트라이크를 치거나 2번째에 스페어 처리를 한 경우 3번째 기회까지 주어진다.  6. 해당 프레임의 기본점수는 해당 프레임에 쓰러뜨린 총 핀의 수이다.  7. 스페어 처리를 한 경우 바로 다음에 쓰러뜨린 핀의 수만큼 보너스 점수를 얻는다.  8. 스트라이크를 한 경우 다음과 다다음에 쓰러뜨린 핀의 합만큼 보너스 점수를 얻는다.  9. 모든 프레임이 끝나면 승자를 결정한다. |
| 비기능 요구사항 | 1. 자바로 구현한다.  2. 플레이어 등급은 초급, 중급, 상급으로 한다.  3. 초급은 평균 100점, 중급은 150점, 상급은 200점을 기준으로 한다.  4. 점수판은 공을 던질 때마다 업데이트한다.  5. 파울로 점수를 얻지 못하는 경우는 없다. |

**3-2. UseCase Diagram**



**3-3. UseCase Scenario**

|  |  |  |
| --- | --- | --- |
| 유스케이스명 | | Start Game |
| 개요 | | 게임 설정을 마친 후 플레이어가 볼링 게임을 시작한다. |
| 관련 액터 | | Player |
| 선행 조건 | |  |
| 후행 조건 | | Player가 공을 던지기 전 대기 상태로 진입한다. |
| 흐름 | 기본 흐름 | 1. Player1이 등급을 입력한다.  2. Player2가 등급을 입력한다.  3. 게임을 시작한다. |
| 대체 흐름 | 1a, 2a. 시스템에 등록되지 않은 등급을 입력한다.  1a1, 2a1. 등급을 다시 안내하고 다시 입력하도록 한다.  3a. 등급이 정해지지 않은 Player가 있다.  3a1. 먼저 등급을 입력하도록 한다. |
| 기타 요구사항 | | 1. Player는 2명이다.  2. 등급은 초급, 중급, 상급 3단계로 구분한다. |

|  |  |  |
| --- | --- | --- |
| 유스케이스명 | | Throw Ball |
| 개요 | | Player가 자신 턴에 볼링공을 던진다. |
| 관련 액터 | | Player |
| 선행 조건 | | Start Game을 완료한 후 대기 상태 |
| 후행 조건 | | 턴을 종료하고, 해당 턴에 쓰러뜨린 핀의 개수와 프레임 종료 상태를 반환한다. |
| 흐름 | 기본 흐름 | 1. Player가 초구를 던진다.  2. Player가 2구를 던진다. |
| 대체 흐름 | 1a. 스트라이크를 친 경우  1a1. 1~9프레임에서는 Player 턴을 종료한다.  1a2. 10프레임인 경우 기본 흐름 2를 하고 3번째 기회가 주어진다.  2a. 10프레임이고 스페어를 처리한 경우 3번째 기회가 주어진다. |
| 기타 요구사항 | | 프레임 종료 상태는 진행중, 오픈, 스트라이크, 스페어로 나뉜다. |

|  |  |  |
| --- | --- | --- |
| 유스케이스명 | | Check the number of dropped pins |
| 개요 | | Player가 공을 던진 후 쓰러뜨린 핀의 개수를 확인한다. |
| 관련 액터 | | Player |
| 선행 조건 | | Player가 공을 던졌다. |
| 후행 조건 | | 1. 이전 프레임들의 종료 상태와 현재 쓰러뜨린 핀의 수에 따라 시스템은 기본점수와 보너스를 계산한다.  2. Player의 시도 횟수와 남은 핀의 수에 따라 진행 여부를 결정한다. |
| 흐름 | 기본 흐름 | 1. 쓰러뜨린 핀의 개수를 알려준다. |
| 대체 흐름 |  |
| 기타 요구사항 | |  |

|  |  |  |
| --- | --- | --- |
| 유스케이스명 | | Look at the scoreboard |
| 개요 | | 현재 모든 Player의 점수를 확인한다. |
| 관련 액터 | | Player |
| 사전 조건 | | 게임이 정상 시작되었다. |
| 사후 조건 | | 점수판 형태로 모든 Player의 현재 점수와 이전 기록들을 출력한다. |
| 흐름 | 기본 흐름 | 1. 이전 기록들과 보너스 점수까지 계산이 완료된 점수까지 출력한다. |
| 대체 흐름 |  |
| 기타 요구사항 | | 1. 따로 요청이 없더라도 점수판은 상시 화면에 띄워놓는다. |

|  |  |  |
| --- | --- | --- |
| 유스케이스명 | | Exit |
| 개요 | | 프로그램을 종료한다. |
| 관련 액터 | | Player |
| 사전 조건 | | 프로그램이 실행되었다. |
| 사후 조건 | | 프로그램을 종료한다. |
| 흐름 | 기본 흐름 | 1. Player가 종료를 요청한다.  2. 종료 여부를 되묻는다.  3. 종료한다. |
| 대체 흐름 | 2a. Player가 종료를 원하지 않으면 다시 게임을 재개한다. |
| 기타 요구사항 | |  |

**4. 모델링**

**4-1. Class Diagram**