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

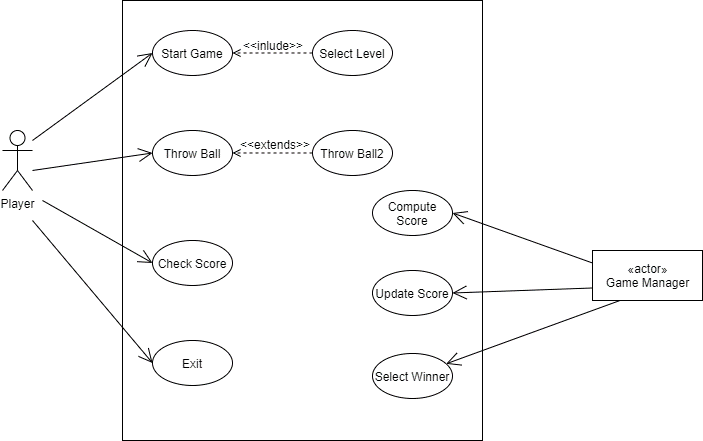
추가: 여건이 된다면 GUI도 적용

**3. 요구사항**

**3-1. 요구사항 도출**

|  |  |
| --- | --- |
| 기능 요구사항 | 1. 플레이어는 플레이어 등급을 정한다.  2. 플레이어가 볼링공을 던진다.  3. 볼링공이 볼링핀을 쓰러뜨린다.  4. 쓰러뜨린 볼링핀에 따라 점수를 계산한다.  5. 점수를 점수판에 표시한다.  6. 모든 프레임이 끝나면 승자를 결정한다. |
| 비기능 요구사항 | 1. 자바로 구현한다.  2. 플레이어 등급은 초급, 중급, 상급으로 한다.  3. 플레이어 등급에 따라 핀을 쓰러뜨릴 확률이 달라야 한다.  4. 점수판은 공을 던질 때마다 업데이트한다.  5. 파울로 점수를 얻지 못하는 경우는 없게 한다. |

**3-2. UseCase Diagram**



**3-3. UseCase Scenario**

|  |  |  |
| --- | --- | --- |
| 유스케이스명 | | Start Game |
| 개요 | | 게임 설정을 마친 후 플레이어가 볼링 게임을 시작한다. |
| 주액터 | | Player |
| 사전 조건 | |  |
| 사후 조건 | | Player가 공을 던지기 전 대기 상태로 진입한다. |
| 흐름 | 정상 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. |
| 주액터 | | Player |
| 사전 조건 | | Start Game을 완료한 후 대기 상태 |
| 사후 조건 | | 쓰러진 핀 수가 정해진다. |
| 흐름 | 정상 흐름 | 1. Player가 공을 던진다.  2. 쓰러진 핀의 수를 보여주고 저장한다. |
| 대체 흐름 |  |
| 기타 요구사항 | |  |

|  |  |  |
| --- | --- | --- |
| 유스케이스명 | | Compute Score |
| 개요 | | 쓰러진 핀 수에 따라 점수를 계산한다. |
| 주액터 | | Game Manager |
| 사전 조건 | | 쓰러진 핀의 수가 반환된 상태 |
| 사후 조건 | | 점수가 반환된다. |
| 흐름 | 정상 흐름 | 1. 쓰러진 핀의 수에 따라 점수를 계산한다.  2. 계산된 점수를 반환한다. |
| 대체 흐름 |  |
| 기타 요구사항 | |  |

|  |  |  |
| --- | --- | --- |
| 유스케이스명 | | Update Score |
| 개요 | | 점수판에 점수를 업데이트한다. |
| 주액터 | | Game Manager |
| 사전 조건 | | 점수 계산 완료 |
| 사후 조건 | | 1. 현재 플레이어가 스페어 처리를 시도한다.  2. 다음 플레이어에게 턴을 넘긴다.  3. 승자를 선정한다. |
| 흐름 | 정상 흐름 | 1. 계산된 점수를 점수판에 기록한다. |
| 대체 흐름 |  |
| 기타 요구사항 | |  |

|  |  |  |
| --- | --- | --- |
| 유스케이스명 | | Throw Ball2 |
| 개요 | | Player가 스페어 처리를 시도한다. |
| 주액터 | | Player |
| 사전 조건 | | 첫번째 시도에서 핀이 남았고, 점수 기록이 끝났다. |
| 사후 조건 | | 핀이 쓰러진 개수가 정해진다. |
| 흐름 | 정상 흐름 | 1. Player가 공을 던진다.  2. 쓰러진 핀의 수를 보여주고 저장한다. |
| 대체 흐름 |  |
| 기타 요구사항 | |  |

|  |  |  |
| --- | --- | --- |
| 유스케이스명 | | Select Winner |
| 개요 | | 승자를 선정한다. |
| 주액터 | | Gamer Manager |
| 사전 조건 | | 두 명의 플레이어 모두 10프레임까지 진행했고, 점수판에 모든 점수가 기록되었다. |
| 사후 조건 | | 승자를 표시한다. |
| 흐름 | 정상 흐름 | 1. Player1과 Player2의 마지막 점수를 비교한다.  2. 점수가 더 높은 Player를 표시한다. |
| 대체 흐름 |  |
| 기타 요구사항 | | 무승부가 나올 수도 있다. |

|  |  |  |
| --- | --- | --- |
| 유스케이스명 | | Check Score |
| 개요 | | Player가 점수판을 확인한다. |
| 주액터 | | Player |
| 사전 조건 | |  |
| 사후 조건 | | 점수판을 보여준다. |
| 흐름 | 정상 흐름 | 모든 Player의 점수를 보여준다. |
| 대체 흐름 |  |
| 기타 요구사항 | | 별도의 요청이 없어도 점수판은 계속해서 띄워놓는다. |

|  |  |  |
| --- | --- | --- |
| 유스케이스명 | | Exit |
| 개요 | | 프로그램을 종료한다. |
| 주액터 | | Player |
| 사전 조건 | |  |
| 사후 조건 | | 프로그램을 종료한다. |
| 흐름 | 정상 흐름 | 1. 플레이어가 게임 종료를 요청한다.  2. 게임 종료 여부를 다시 한번 더 요청한다. |
| 대체 흐름 | 2a1. 게임 종료를 결정하면 프로그램을 종료한다.  2a2. 게임 종료를 번복하면 게임을 다시 진행한다. |
| 기타 요구사항 | | 어떤 상황에서도 게임을 종료할 수 있다. |