

UseCase Description

UC1. 게임 설정

항목	내용
Use Case 이름	게임 설정
액터	사용자 (Player)
목적	플레이어 수와 말 수를 선택하여 게임 시작 조건을 설정한다.
사전조건	시스템이 초기화 상태이며 설정 화면이 활성화되어 있어야 한다.

기본 흐름

1. 사용자는 FirstPage에서 플레이어 수(2~4명)를 선택한다.
2. 사용자는 말 수(2~5개)를 선택한다.
3. 사용자는 "시작" 버튼을 클릭한다.
4. Controller (`GameController`)가 `PlayConfig` 의 값을 읽어 게임을 초기화한다.
5. `GameState` 와 `YutBoard` 가 초기화되며, 게임이 시작된다.
 - **예외 흐름:**선택 없이 시작 버튼을 누르면 기본값(2명, 2말)으로 진행된다.
 - **사후 조건:** 게임 상태 초기화 완료. 게임 보드가 사용자에게 표시됨.

UC2. 윷 던지기

항목	내용
Use Case 이름	윷 던지기
액터	사용자 (Player)
목적	자신의 차례에 윷을 던져 이동할 수를 결정한다.
사전조건	게임이 시작되어 있고, 사용자의 차례이며 윷 던지기 버튼이 활성화되어야 한다.

기본 흐름

1. 사용자가 "윷 던지기" 버튼을 클릭한다.
2. Controller는 `Yut.throwing()` 을 호출하여 윷 결과를 생성한다.
3. 생성된 결과는 `GameState` 에 저장된다.
4. Controller는 View(`YutBoard`)에 윷 결과 표시를 요청한다.

- **예외 흐름:** 게임이 종료된 상태거나 차례가 아닌 경우 버튼은 비활성화되어 입력 무시됨.
- **사후 조건:** 이동할 수(도, 개, ...)가 결정되고 사용자에게 표시됨. 다음 단계(말 이동)로 전환됨.

UC3. 말 선택 및 이동

항목	내용
Use Case 이름	말 이동
액터	사용자 (Player)
목적	윷 결과에 따라 이동시킬 말을 선택하고 해당 말을 이동시킨다.
사전조건	윷을 던진 후 결과가 존재해야 하며, 말이 존재해야 한다.

기본 흐름

1. 사용자가 보드 위에서 말을 선택한다.
2. Controller는 해당 위치에 사용자의 말이 있는지 확인한다.
3. `Player.move(x, y, result)` 를 호출하여 말을 이동시킨다.
4. 이동 후 업기(`checkUpda`) 또는 골인(`checkPiecein`) 여부를 판단한다.
5. Controller는 View에 말 위치 변경을 요청한다.
 - **예외 흐름:** 선택된 위치에 사용자의 말이 없으면 메시지를 띄우고 입력을 무시한다.
 - **사후 조건:** 선택된 말이 윷 결과만큼 이동됨. 상황에 따라 다시 던지거나 턴 종료됨.

UC4. 말 업기

항목	내용
Use Case 이름	말 업기
액터	시스템
목적	같은 위치에 있는 아군 말을 하나의 말로 묶어 함께 이동시킨다.
사전조건	동일 플레이어의 말 두 개 이상이 동일 위치에 도착

기본 흐름

1. `Player.move()` 이후 `checkUpda()` 를 호출하여 같은 위치에 있는 말을 탐색
2. 존재하면 기존 말을 삭제하고 대표 말에 point 수를 추가

- 예외 흐름: 없음.
- 사후 조건: 말이 병합되며 하나의 말로 움직인다. point 값이 증가한다.

UC5. 말 잡기

항목	내용
Use Case 이름	말 잡기
액터	시스템
목적	이동된 말이 상대방 말과 위치가 겹칠 경우 상대 말을 제거한다.
사전조건	이동 후 다른 플레이어의 말과 동일 위치 도달

기본 흐름

1. 이동이 완료된 후 `checkCatch(x, y)` 호출
2. 같은 위치에 있는 상대 말이 있다면 해당 말을 제거
3. 잡힌 말의 point만큼 상대방 말 수를 복구
4. View에 "말을 잡았다" 메시지 출력
 - 예외 흐름: 잡을 수 있는 말이 없으면 아무 작업도 하지 않음.
 - 사후 조건: 상대방 말이 제거되며, 잡은 플레이어는 한 번 더 턴을 갖는다.

UC6. 승리 판정

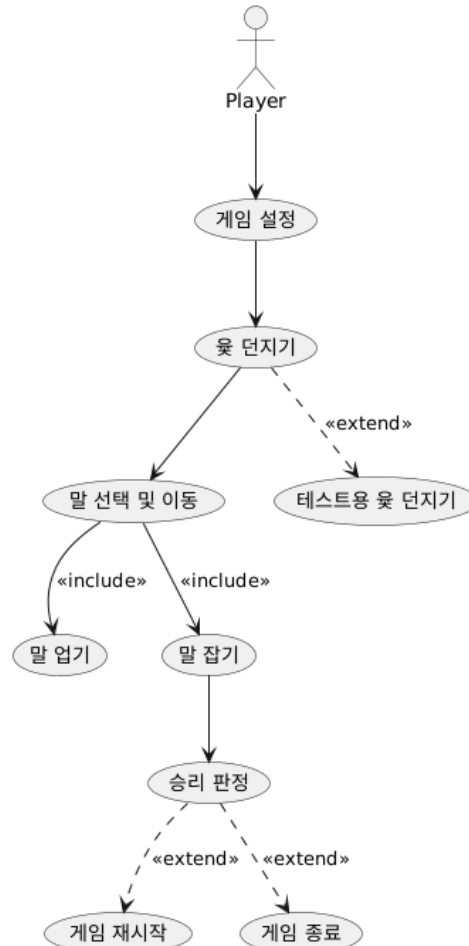
항목	내용
Use Case 이름	승리 판정
액터	시스템
목적	한 플레이어가 모든 말을 완주시키면 승리 상태를 표시한다.
사전조건	<code>Player.point == pieceNum</code> 일 경우

기본 흐름

1. 말이 골인될 때마다 `checkPiecein()` 으로 point 증가
2. Controller가 `GameState` 를 통해 point 상태 확인
3. 특정 플레이어의 point가 총 말 수와 같으면 승리 처리
4. View에 승리 메시지 출력 및 버튼 비활성화
 - 예외 흐름: 없음

- 사후 조건: 게임 종료, 입력 비활성화. 재시작 UI가 활성화될 수 있음.

유스케이스 간 관계도



관계 설명

관계 유형	설명
UC1 → UC2	게임 설정 이후 윷 던지기가 가능한 상태로 전이됨
UC2 → UC3	윷을 던지면 말 선택 및 이동을 진행함 (필수 연계)
UC3 → UC4 , UC3 → UC5	말 이동 시 업기나 잡기가 조건에 따라 포함됨 (<<include>>)
UC5 → UC6	말 잡기 이후, 점수 증가로 인해 승리 조건이 될 수 있음
UC6 → UC7 , UC6 → UC8	게임 승리 시 유저는 재시작 또는 종료를 선택할 수 있음 (<<extend>>)
UC2 → UC11	테스트 상황에서는 UC11: 테스트용 윷 던지기 를 대신 사용 (<<extend>>)