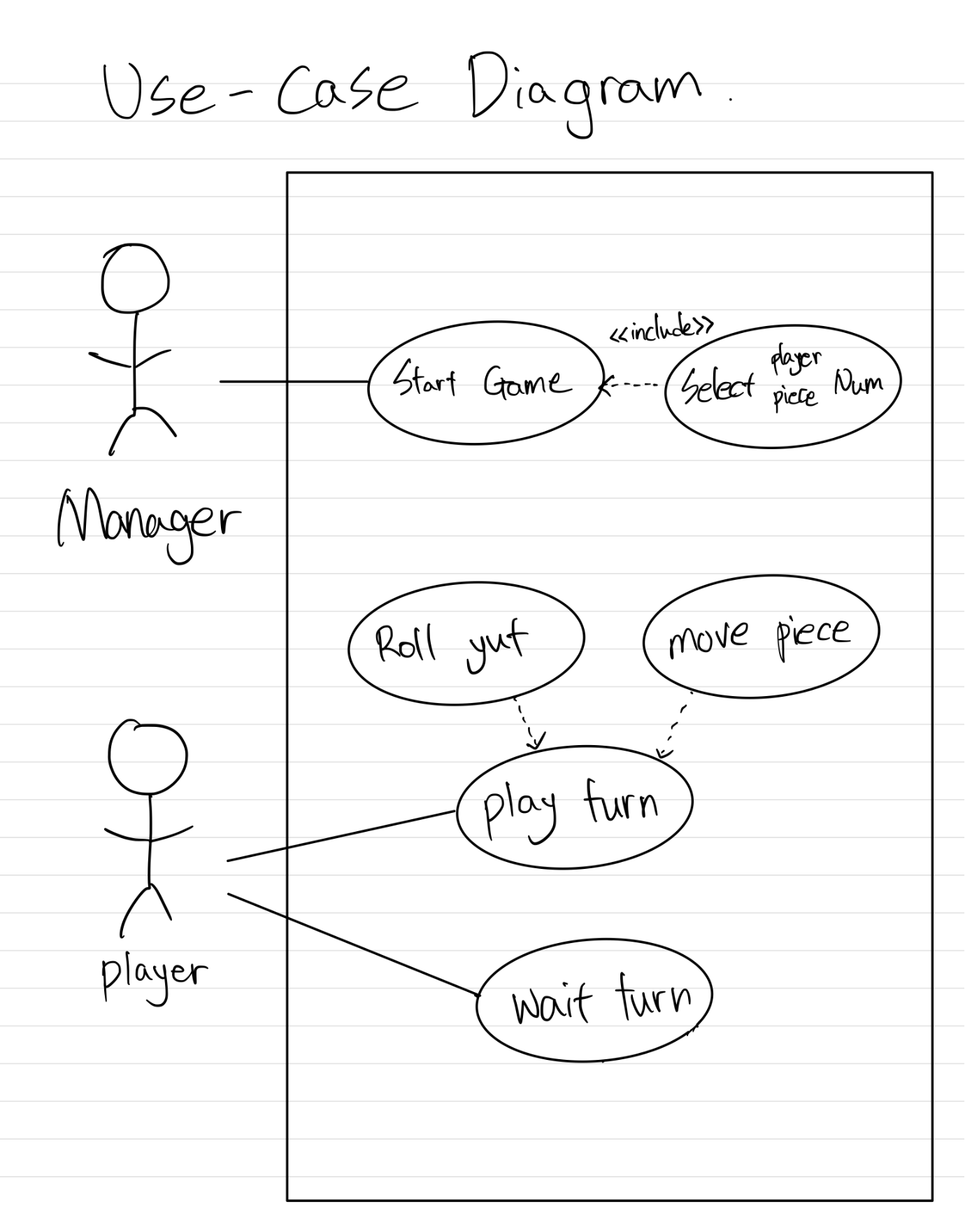
**Vision & MainFeauture**

실제 윷을 던지는 것과 같은 확률을 갖는, 마우스를 클릭을 통해 진행되는 간단한 윷놀이 게임

**Use-case Model**



**Use case description**

Scope: 윷놀이 프로그램

Level: user goal

Primary actor: Player

Stakeholders and interests:

* Player: 주어진 규칙 아래에서 자신의 piece를 최대한 빠르게 goal-in시킨다.
* Manager: 정해진 인원수와 말의 개수를 가지고 공평하게 게임이 시작할 수 있도록 초기화한다.

Precondition: 윷놀이는 사전에 정해진 규칙만으로 진행된다. (플레이어는 명확하게 구분되고 검증된 존재이다).

Success guarantee(or post-condition): 한 플레이어의 말을 제외한 나머지 모든 플레이어의 말들은 goal-in되었고 완전히 goal-in한 플레이어 사이에는 rank가 정해졌다.

Main success scenario:

1. 관리자가 플레이어 수와 말의 수를 결정한다.
2. 윷을 던진다.
3. 던진 윷의 횟수만큼 말을 선택하여 움직인다.
4. 다시 자신의 차례가 올 때까지 기다린다.
5. 1~3을 자신의 모든 말이 goal in 할 때까지 반복한다.
6. 모든 말이 goal in 하면 ranking을 보여준다

Alternative flows:

1a. 윷이나 모가 나오면 한번 더 던진다.

2a. 출발점에서 빽도가 나오면 무시한다.

2b. 출발점의 한 칸 진행한 위치에서 빽도가 나오면 출발점으로 되돌아온다.

2c. 말이 분기점에 위치할 경우, 방향을 선택한다.

2d. 움직인 위치에 상대의 말이 있으면 말을 잡는다,

2e. 움직인 위치에 우리 팀의 말이 있으면 말을 업을 수 있다.

2f. 움직인 말이 결승점을 통과하면 goal-in한다.

3a. 상대방이 우리 팀의 말을 잡으면 말을 초기화한다.

**Operation Contract**

|  |  |
| --- | --- |
| Operation: | Inform turn(playerN: Player) |
| CrossReferences: | Use cases: wait turn |
| Preconditions: | play turn 중인 상대방의 turn이 끝나야 한다 |
| Postconditions: | - playerN instance와 System이 연결된다(association formed).  - System내 진행이 될 turn의 Player가 playerN으로 된다(attribute modification). |

|  |  |
| --- | --- |
| Operation: | select player & piece num() |
| CrossReferences: | Usecase:Start game |
| Preconditions: | 게임이 시작되기 전. |
| Postconditions: | - PlayerN instance를 선택한 만큼(2~4개)생성(instance creation)  - 각 playerN별 pieceN을 선택한 만큼 생성(instance creation)  - Piece는 playerN과 composition 관계(association formed) |

|  |  |
| --- | --- |
| Operation: | start game() |
| CrossReferences: | Usecase:Start game |
| Preconditions: | Player와 piece의 숫자가 결정되어있음. |
| Postconditions: | -Usecase:playturn 시작 |

|  |  |
| --- | --- |
| Operation: | Roll yut() |
| CrossReferences: | Usecase:Play Turn |
| Preconditions: | 플레이어의 순서가 결정된 후 |
| Postconditions: | -윷을 던져서 어떤 윷인지에 대한 결과를 저장  -Yut instance에 해당 윷을 저장한다(attribute modification)  -Usecase:move piece 시작 |

|  |  |
| --- | --- |
| Operation: | Move piece() |
| CrossReferences: | Usecase:Play Turn |
| Preconditions: | 윷이 던져진 후 |
| Postconditions: | -선택한 말을 알맞게 이동한다.  -Board instance에 해당 말의 이동 결과를 업데이트한다(attribute modification)  -Usecase:playturn 시작 |

**System Sequence Diagram**

