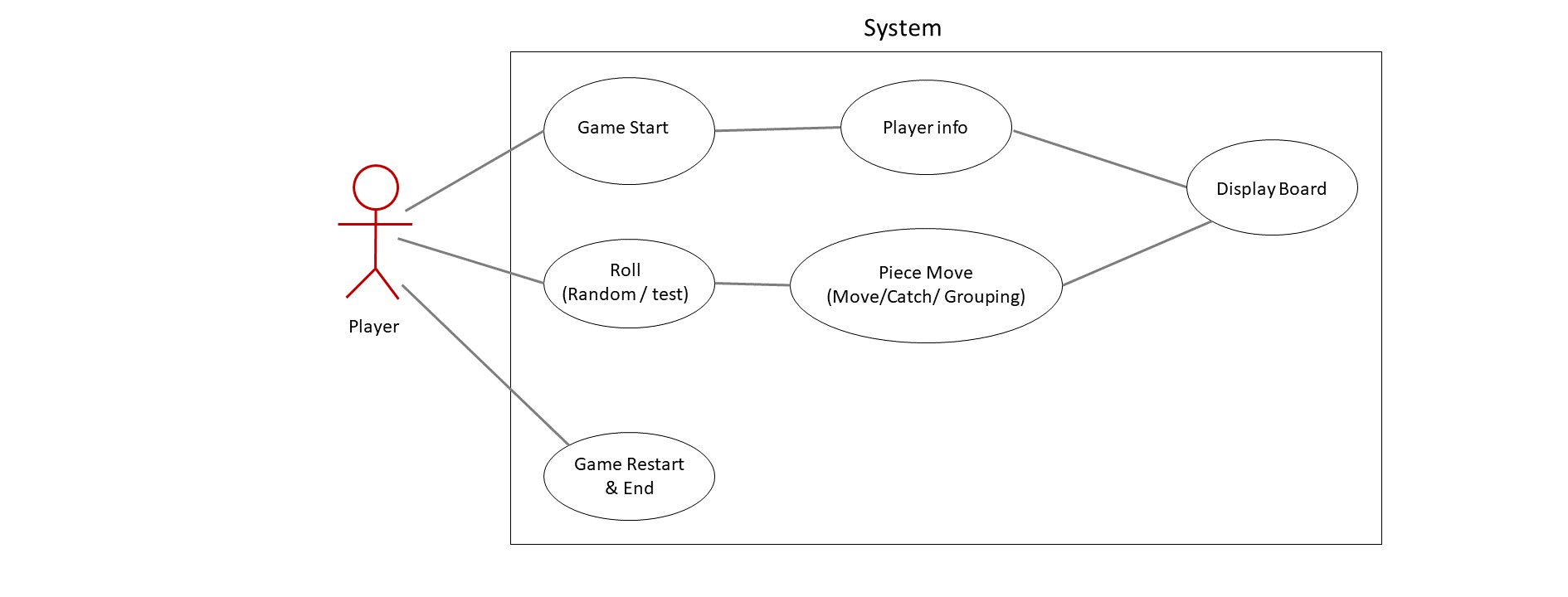
**[SE Team Project]**

**윷놀이**

**20174374 정여민** **20122776 양명철 20144753 선승엽 20156033 최은정**

**1. Use-case Diagram**



**2. Use-case Description**

**[UC1] 게임 시작 후 플레이어의 정보를 입력하는 Use-case**

**1.1 Preconditions** : X

**1.2 Main Flow**

게임 시작 시 플레이어의 명수(최소 2명, 최대 4명), 플레이어의 이름 및 게임 말 갯수(최소 2개, 최대 5개)를 입력한다.

**1.3 Subflows** : X

**1.4 Alternative Flows**:

[E1] 플레이어 수는 2에서 4사이의 정수. 플레이어가 정수를 입력하지 않거나 숫자가 2와 4사이가 아닌 경우 게임은 플레이어에게 다시 플레이어 수를 다시 입력하도록 요청한다.

[E2] 말의 갯수는 2에서 5사이의 정수. 플레이어가 정수를 입력하지 않거나 숫자가 2와 5사이가 아닌 경우 게임은 플레이어에게 다시 말의 갯수를 다시 입력하도록 요청한다.

[E3] 이름은 빈 문자열은 불가. 플레이어가 빈 문자열을 입력하면 게임은 플레이어에게 이름을 다시 입력하라고 요청한다.

[E4] 취소 버튼을 누르면 플레이어 정보 창이 닫히고 게임이 종료된다.

**[UC2] 말을 움직이는 Use-case**

**2.1 Preconditions** : UC2에서 플레이어의 순서가 정해져야 한다.

**2.2 Main Flow** : 게임은 차례로 진행되는 턴 기반이다. 플레이어 정보가 입력되면 첫번째 플레이어의 차례가 시작된다. 말의 움직임은 플레이어가 던진 윷의 결과에 따라 달라진다[UC7]. (도 → 한 칸, 개 → 두 칸, 걸 → 세 칸, 윷 → 네 칸, 모 → 다섯 칸 앞으로 움직인다.) 플레이어가 던진 윷의 결과에 따라 플레이어가 선택한 말이 이동하며 말의 이동이 끝나면 턴이 끝난다.

**2.3 Subflows** :

[S1] 플레이어는 새로운 말을 추가할 것인지(최대5개) 기존의 말을 움직일 것인지 정할 수 있다.

[S2] 플레이어가 움진인 말의 결과에 따라 다음 움직임 방향이 결정된다. (분기점들)[UC3]

[S3] 정해진 시간 내에 선택을 하지 않을 경우, 이전에 움직인 말을 움직인다.

**2.4 Alternative Flows** :

[E1] 추가된 말의 개수가 최대치일 때 더 이상 말을 추가할 수 없다는 문구가 뜬다.

[E2] 이전에 움직인 말이 보드에 없을 시, 새로운 말을 추가한다.

[E3] 이전에 움직인 말이 없을 시, 보드에 있는 말이 움직인다.

**[UC3] 교차점에서의 플레이어의 움직임에 대한 Use-case**

**3.1 Preconditions** :

1. 플레이어의 차례.

2. 플레이어가 윷을 던진 후여야 한다.

**3.2 Main Flow** : 플레이어의 기존 위치가 모서리인 경우, 플레이어는 사각형 테두리가 아닌 대각선 방향으로 말을 움직인다.

**3.3 Subflows** : X

**3.4 Alternative Flows** : X

**[UC4] 같은 팀의 말을 업는(Grouping) Use-case**

**4.1 Preconditions** :

1. 플레이어의 차례.

2. 플레이어가 윷을 던진 후여야 한다.

**4.2 Main Flow** : 플레이어가 움직인 위치에 같은 팀의 말이 이미 있다면, 해당 말을 업을(grouping) 수 있다.

**4.3 Subflows** : X

**4.4 Alternative Flows** : X

**[UC5] 다른 팀의 게임의 말을 잡는 Use-case**

**5.1 Preconditions** :

1. 플레이어의 차례.

2. 플레이어가 윷을 던진 후여야 한다.

**5.2 Main Flow** : 플레이어가 움직인 위치에 다른 팀의 말이 이미 있다면, 해당 말을 잡을 수 있다.

**5.3 Subflows** :

[S1] 잡힌 말은 처음부터 다시 출발해야 한다.

[S2] 상대의 말을 잡으면 한 번 더 윷을 던진다.

**5.4 Alternative Flows** : X

**[UC6] 윷을 던지는 Use-case**

**6.1 Preconditions** :

1. 플레이어의 차례.

**6.2 Main Flow** : 각 플레어는 자신의 차례에 윷 4개를 한번에 던진다.

**6.3 Subflows** :

[S1] 테스트용 버튼을 누르면 빽도/도/개/걸/윷/모를 선택할 수 있고 선택한대로 윷이 던져진다.

[S2] 랜덤을 누르면 윷이 랜덤으로 던져진다.

[S3] 윷/모가 나온 경우는 윷을 한 번 더 던진다.

[S4] 정해진 시간 내에 윷을 던지지 않으면 자동으로 윷이 던져진다.

**6.4 Alternative Flows** : X

**[UC7] 게임 종료 Use-case**

**7.1 Preconditions** :

1. 플레이어의 차례.

2. 플레이어가 윷을 던진 후여야 한다.

**7.2 Main Flow** : 출발점에서 시작하여 한 바퀴 돌아 상대편 말보다 먼저 모든 말이 모두 도착하면 게임이 종료된다.

**7.3 Subflows** :

[S1] 이긴 팀의 이름이 화면에 표시된다.

[S2] 게임 재시작 및 종료를 선택할 수 있다.

**7.4 Alternative Flows** : X

**3. System Sequence Diagram**

**4. Operation Contract – Use Case를 자세하게 나눠놓은 상태여서 더 쓸게 없..**

|  |  |
| --- | --- |
| Operation | 말의 이동 위치 |
| Cross Reference | Use case : 말 움직이기 |
| Preconditons | 1. 플레이어가 윷을 던진 상황이어야 한다. |
| Postconditions | ∙ 플레이어가 던진 윷의 결과가 기록된다.  ∙ 플레이어는 보드에 말을 추가할 것인지,  기존의 말을 움직일 것인지 정했다.  ∙ 움직인 칸에 같은 팀의 말이 있다면 업힌다.  ∙ 움직인 칸에 상대 팀의 말이 있었다면 잡았다.  ∙ 플레이어가 던진 윷의 결과가 ‘윷/모’이거나 플레이어가 상대팀의 말을 잡은 경우 플레이어는 한 번 더 윷을 던진 상황이다. |

**1. 비전**

1.1 Short Introduction to the system

이 프로그램은 한국의 전통놀이인 윷놀이를 바탕으로 제작되었다. 게임을 하는 방법은 비교적 쉬우나 게임에서 이기기 위해서는 주어진 규칙을 익혀 최대한으로 활용해야 한다. 게임 시작 전 최소 2명에서 최대 4명의 유저를 설정할 수 있으며, 말판 위에서 움직이는 말은 최소 2개에서 5개까지 설정할 수 있다. 순서를 정하기 위한 윷을 던져서 유저의 게임 순서를 정하게 되며 중복되지 않도록 한다. 유저 설정 이후에 윷을 던지면서 게임이 본격적으로 시작된다. 한 턴에는 제한된 시간이 있고, 이 시간 내에 랜덤 던지기 버튼을 눌러 윷을 던져야 한다. 윷을 던진 후에는 도,개,걸,윷,모,빽도 중 하나의 결과를 확인하고 새로운 말을 추가하거나 기존의 말을 이동한다. 말이 이동하여 상대방의 말을 잡으면 한 번 더 기회가 주어지고 잡힌 말은 출발점으로 이동해 처음부터 다시 시작한다. 자신의 말을 업을 경우 말을 묶어 같이 이동시킨다. 또한, 윷이나 모가 나온 경우에는 윷을 한 번 더 던질 기회가 주어지고 모든 말이 이동하고 윷을 던질 기회가 없다면 다음 유저에게 턴이 넘어간다. 한 유저의 모든 말이 출발하여 말판을 한 바퀴 돌아 다시 출발점으로 도착한 경우에 승자가 정해지고 게임이 종료된다.

1.2 Main Features