

Bridge Game

소프트웨어공학 Term Project

2022

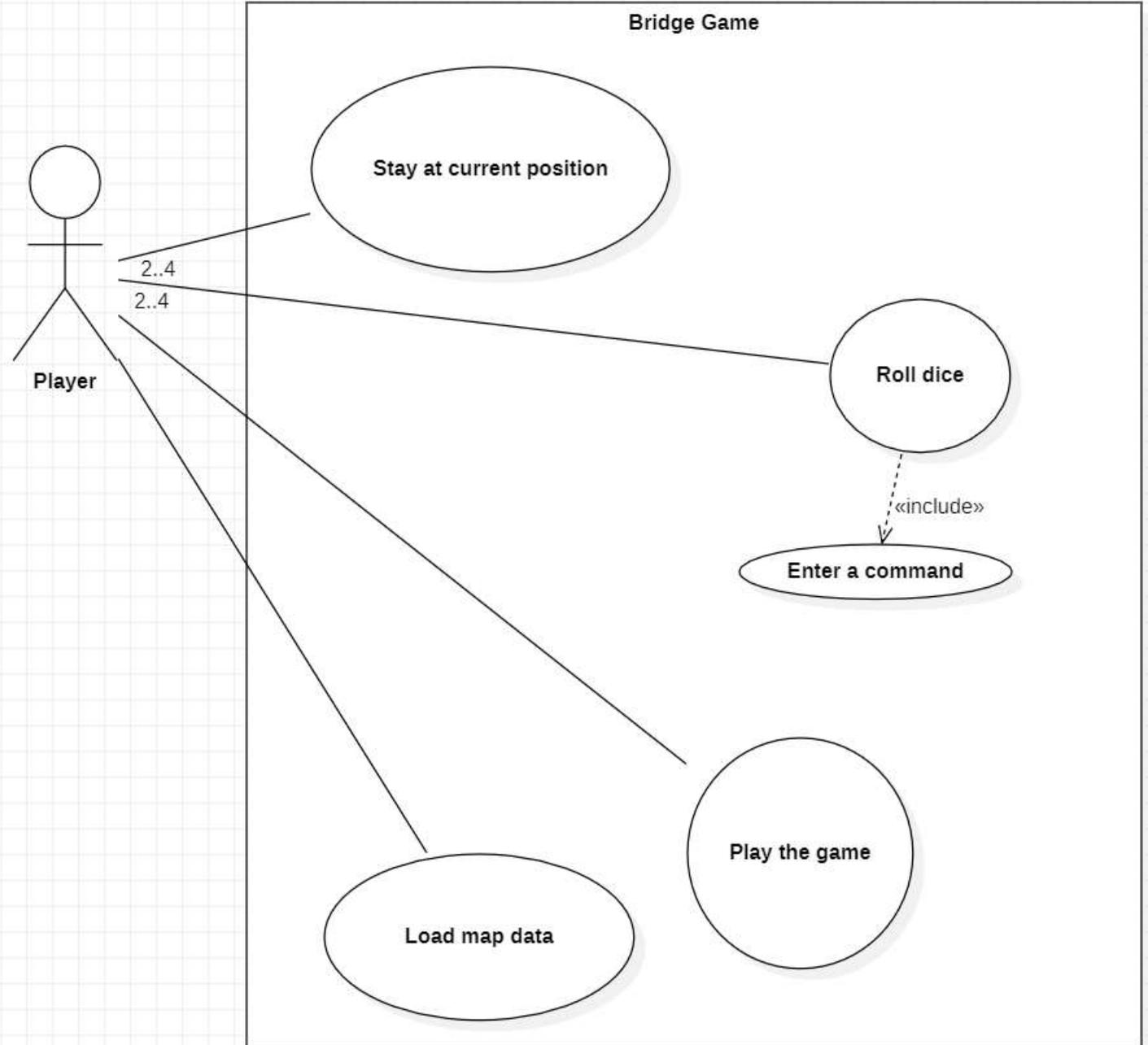
학번: 20183784
이름: 노현진

목차

1. 프로그램 시연
2. Use Case Diagram
3. Use Case 명세
4. Class Diagram
5. Sequence Diagram
6. MVC 패턴
7. 맵 크기 제약
8. 소스코드 설명
9. 테스트 결과
10. 프로그램 사용 방법

Use Case Diagram

다음의 Use Case Diagram은 플레이어가 시스템과 어떻게 상호작용할 수 있는지를 보여준다. 플레이어는 게임 시작 전 map data 파일을 load하거나 게임을 시작할 수 있으며, 또한 게임이 시작된 뒤에는 주사위를 굴러 이동하거나 Stay 기능을 통해 현재 위치에서 그대로 대기할 수 있다.



Use Case 명세: Roll dice

Actor: A Player

Summary: Player rolls dice and check if the player can move. If the player can move, then enter a command. If the command is valid, then player on the game moves according to the command. Otherwise, retry to enter a command.

Precondition: It's the player's turn at that time.

Success Guarantee: move the player's position successfully.

Scenarios:

1. Player decides to roll a dice at the time when it's player's turn.
2. Considering the value of dice and cards the player has, check if the player can move.
 - 2a. If the player can't move, then the player's turn is over.
3. If the player can move, then enter a command.
 - 3a. If the command is invalid, then repeat step 3.
4. If command is valid, then player on the game moves.
5. When player is moving, player can get a score if the player arrives at End or get an item.
6. If only one player is left on the game, then display scores of each player and exit the program.
7. Pass the turn to another active player.

Use Case 명세: Stay at current position

Actor: A Player

Summary: Player return a card and stay at current position.

Precondition: It's the player's turn at that time and the player has at least one card.

Success Guarantee: return a card and stay at current position successfully.

Scenarios:

1. Player decides to stay at current position at the time when it's player's turn.
2. Return a card the player has.
3. Stay at current position.
4. Pass the turn to another active player.

Use Case 명세: Load map data

Actor: A Player

Summary: Before the game begins, player can choose which map data to load.

Precondition: It's possible to load map data before the game begins.

Success Guarantee: Load map data successfully.

Scenarios:

1. Player decides to load specific map data.
2. Find the map data player wants to load.
3. If specific map data is chosen, then set the map data to load before the game begins.
 - 3a. If canceled, the don't change the map data to load.

Class Diagram

프로그램의 Class Diagram을 나타낸 것으로 각 클래스에 대해 설명하자면 다음과 같다.

Card: 카드

Player: 플레이어

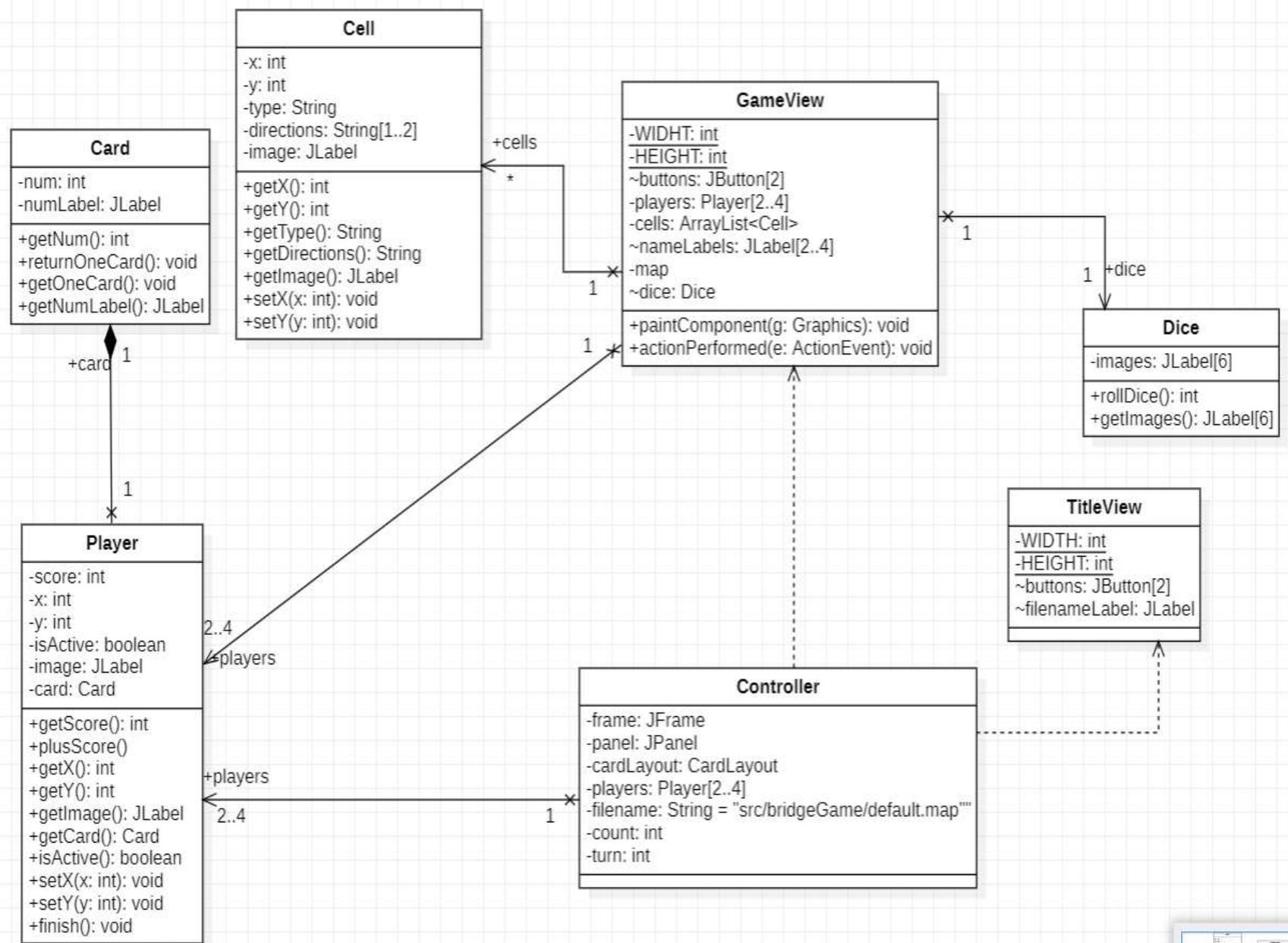
Cell: 맵의 셀 하나에 해당

GameView: 게임이 진행되는 화면에 해당

Dice: 주사위

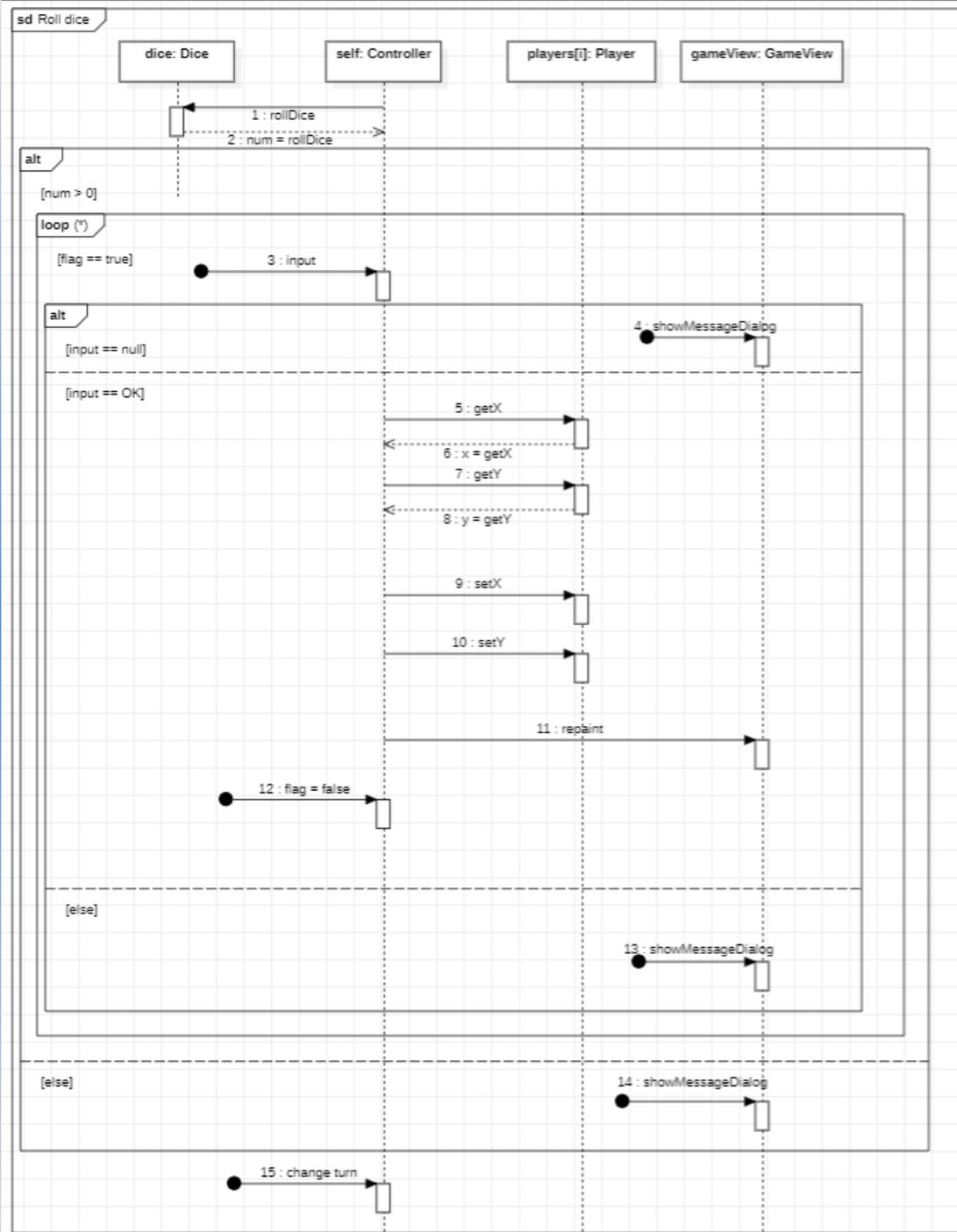
TitleView: 게임 시작 화면에 해당

Controller: MVC 패턴에서 controller에 해당하며 게임의 로직을 담당한다.



Sequence Diagram

다음 Sequence Diagram은 주사위를 굴러 이동할 때 objects와 시스템 사이의 진행 과정을 보여준다. 처음에 rollDice() 메소드를 호출하여 주사위를 굴린다. 그런 다음 그 주사위 값에 따라 플레이어로부터 Input 값을 입력 받을지를 결정한다. Input 값을 입력 받았다면 그 입력 받은 문자열을 검사하여 유효한 문자열인지 확인한다. 만약 유효하지 않다면 다시 입력 받는다. 유효하다면 게임 상의 플레이어를 입력 받은 값에 따라 이동시킨다. 이 때 gameView의 repaint() 메소드를 호출하여 플레이어의 변경된 위치를 표시하도록 한다. 이 모든 과정이 끝나면 마지막으로 다른 플레이어에게 턴을 넘긴다.



MVC 패턴

MVC 패턴을 적용하여 UI와 응용 로직(logic)을 분리하였습니다. 이에 따라 UI가 변경되더라도 로직은 변경없이 재사용이 가능하도록 프로그램이 구현되었습니다.

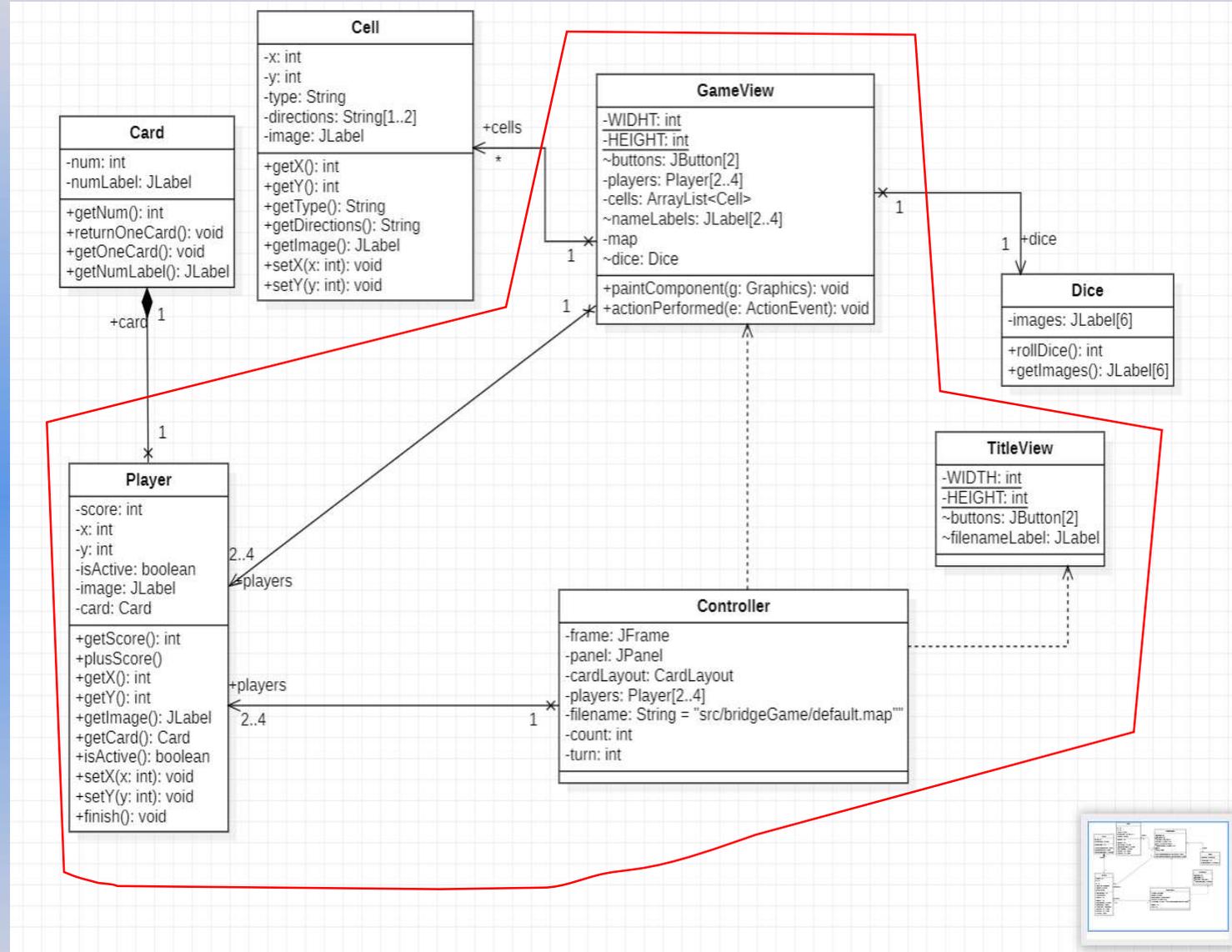
Model: Player 클래스

View: TitleView 클래스, GameView 클래스

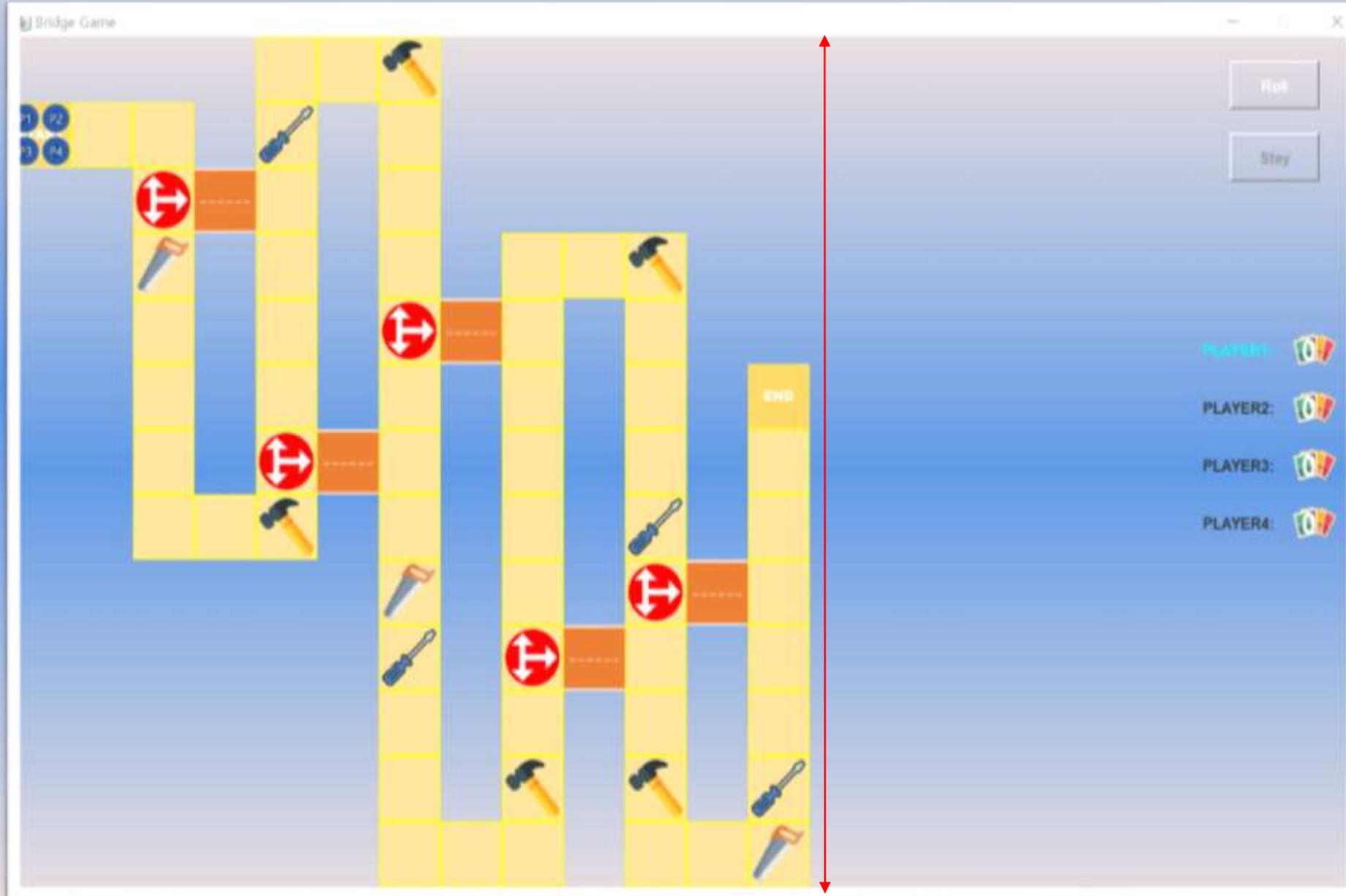
Controller: Controller 클래스

Scenario:

1. User가 주사위를 굴러 플레이어의 위치를 이동하려고 시도함.
2. Controller가 User로부터 Input 값을 처리함.
3. Controller가 Model에 해당하는 Player의 위치 값을 변경
4. 변경된 플레이어의 위치가 GameView에 전달되어 변경된 위치가 화면에 표시됨.
5. View가 User의 상호작용을 기다림.



맵 크기 제약



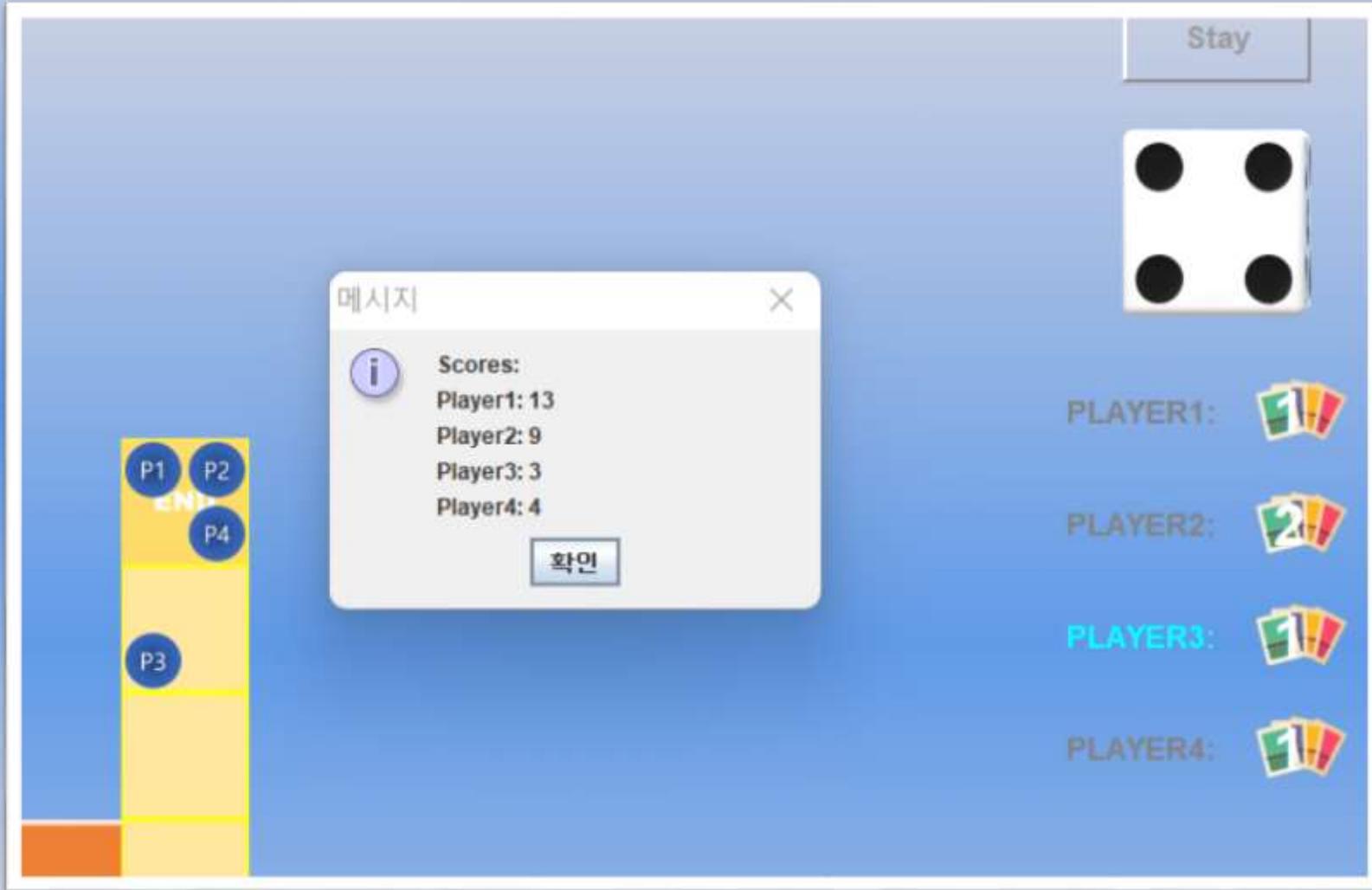
제출된 프로그램은 load 할 맵 크기에 제약 사항이 존재합니다. 게임 특성 상 맵 크기가 지나치게 크면 화면에 전부 표시할 수 없으므로 화면 밖으로 벗어나지 않는 크기의 맵만 load하는 것이 가능합니다. 예시로 왼쪽의 맵 크기는 load할 수 있는 맵 중 세로 길이가 가장 긴 맵입니다. 이보다 세로 길이가 긴 맵은 load할 수 없습니다.

소스코드 설명

1. `Main.java`: 프로그램의 시작 지점
2. `TitleView.java`: 게임의 시작 화면으로 `map data`을 읽거나, 또는 게임을 시작할 수 있다.
3. `GameView.java`: 게임 화면으로 각 플레이어의 위치, 카드의 수, 맵 화면 등을 표시한다.
4. `Controller.java`: 게임의 로직을 담당하는 소스코드
5. `Card.java`: 각 플레이어가 갖고 있는 카드에 해당한다.
6. `Cell.java`: `map`을 화면에 표시할 때 사용하는 클래스로 하나의 셀에 해당한다.
7. `Dice.java`: 주사위에 해당하는 클래스
8. `Player.java`: 게임을 진행하는 플레이어에 해당하는 클래스로 게임을 시작할 때 그 수가 결정된다.

각 파일의 소스코드는 제출된 `java` 파일을 참조해주시기 바랍니다.

테스트 결과



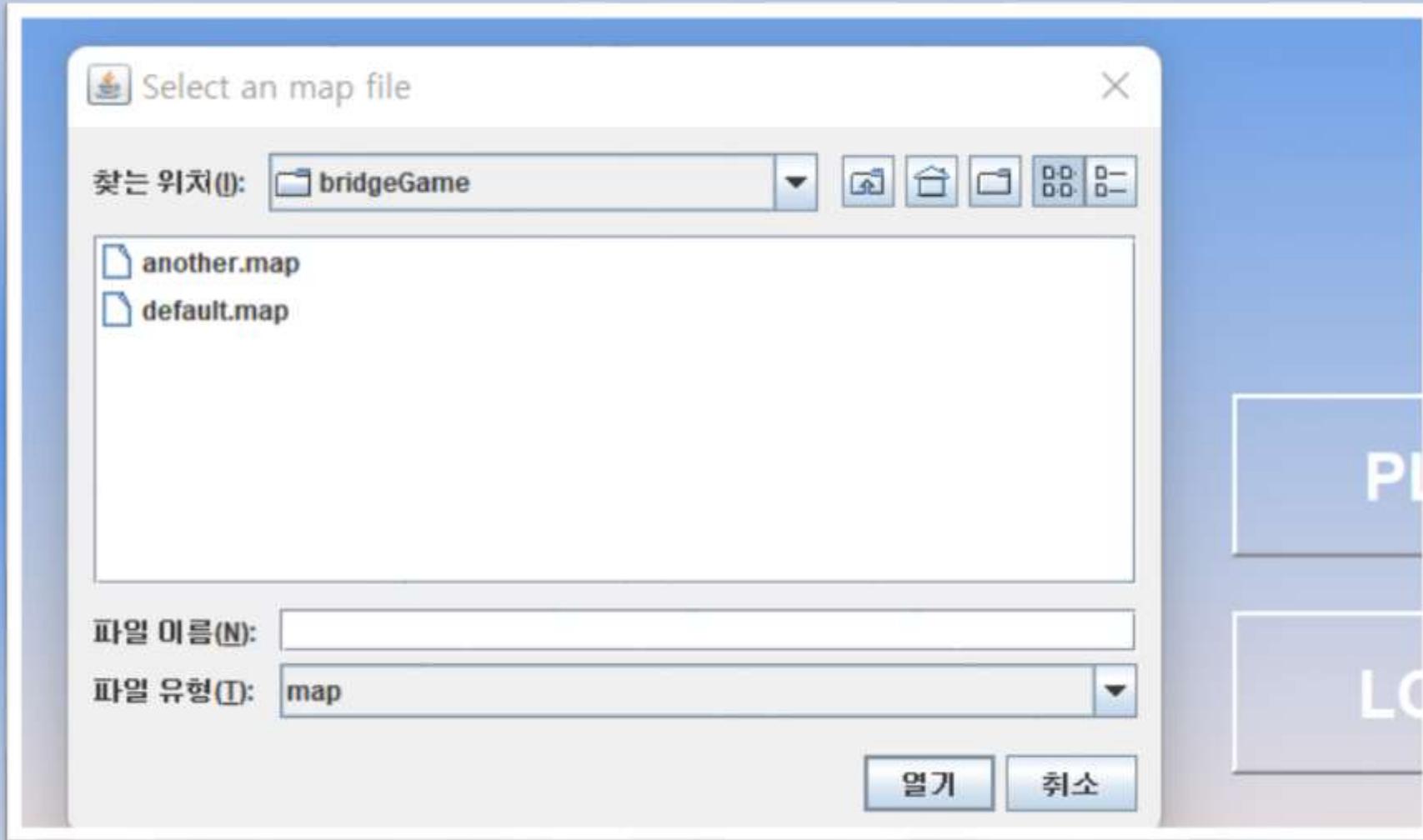
게임이 종료되었을 때 다음과 같은 결과가 표시된다.

프로그램 사용 방법 - LOAD



프로그램이 시작되면
default.map 파일을 디
폴트로 load한다

프로그램 사용 방법 - LOAD



LOAD 버튼을 누르면
다음과 같이 확장자가
map인 파일을 Load 할
수 있다

프로그램 사용 방법 - LOAD



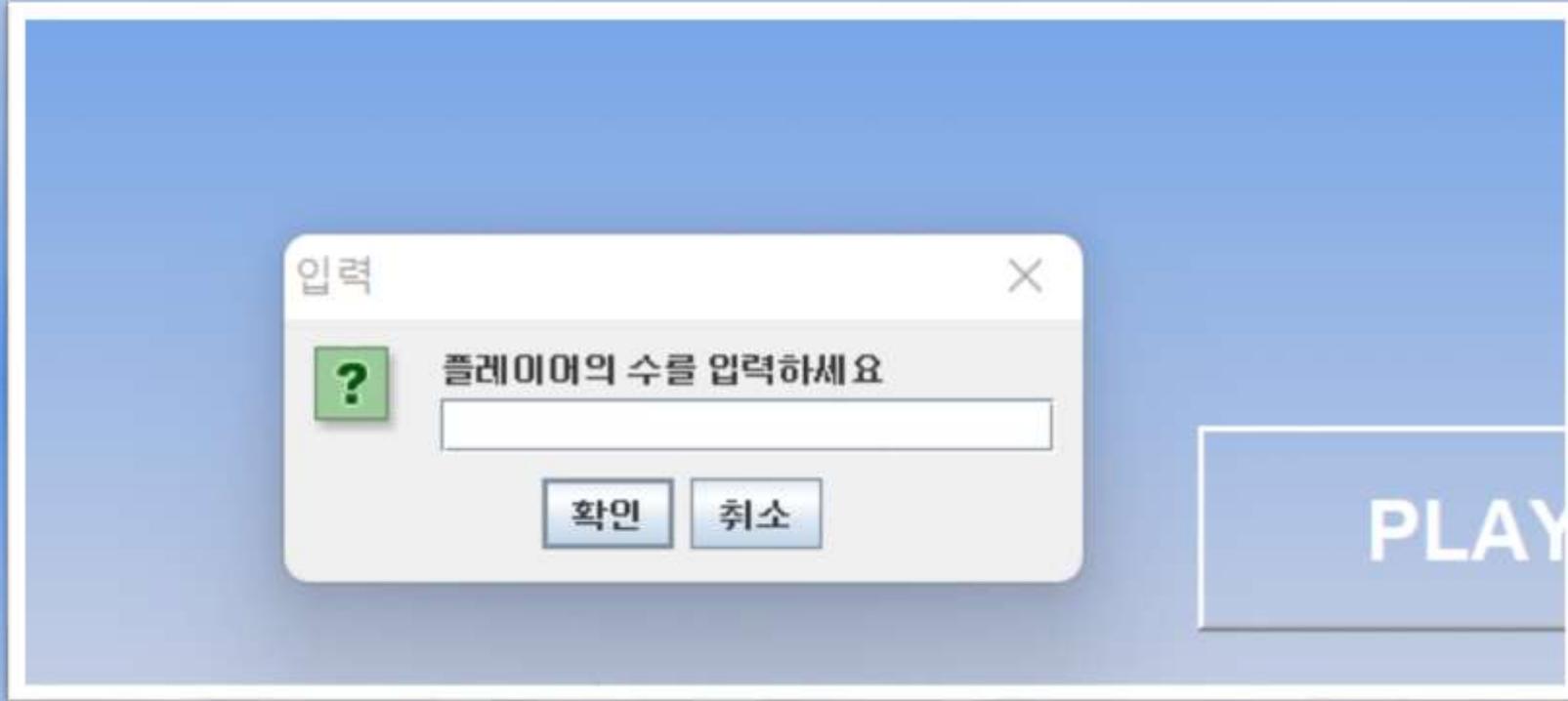
Map 파일을 load 했을 경우 다음과 같이 화면에 load할 파일 이름을 표시한다.

프로그램 사용 방법 - 플레이어 수 입력



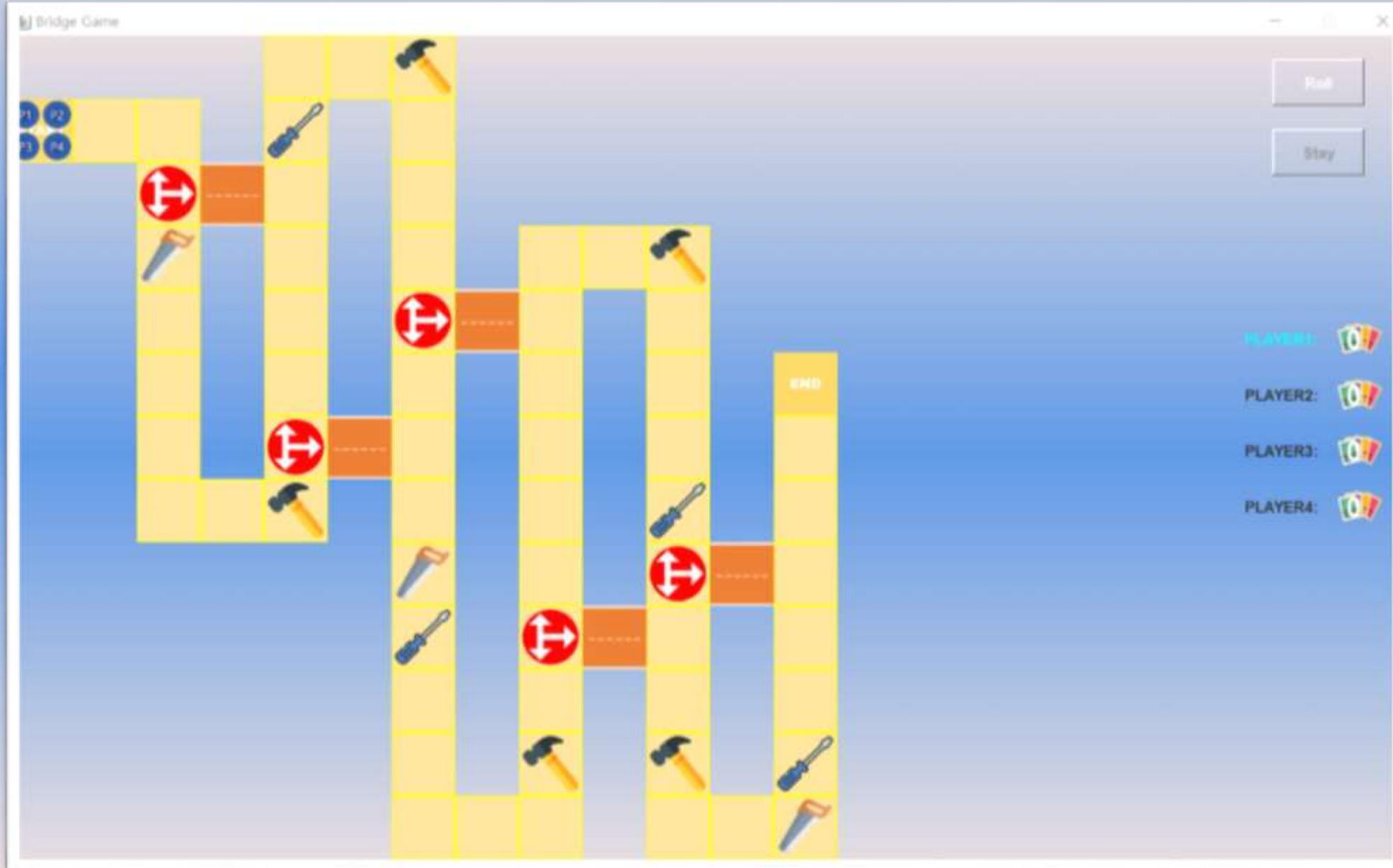
PLAY 버튼을 누르면
load할 map 파일을 갖
고서 게임을 시작할 수
있다.

프로그램 사용 방법 - 플레이어 수 입력



PLAY 버튼을 누르면 다음과 같이 플레이어의 수를 입력 받는다. 입력 받는 값은 2, 3, 4 중 하나로 한정되며 다른 것을 입력 받았을 경우 게임이 시작되지 않는다.

프로그램 사용 방법 - 게임 화면

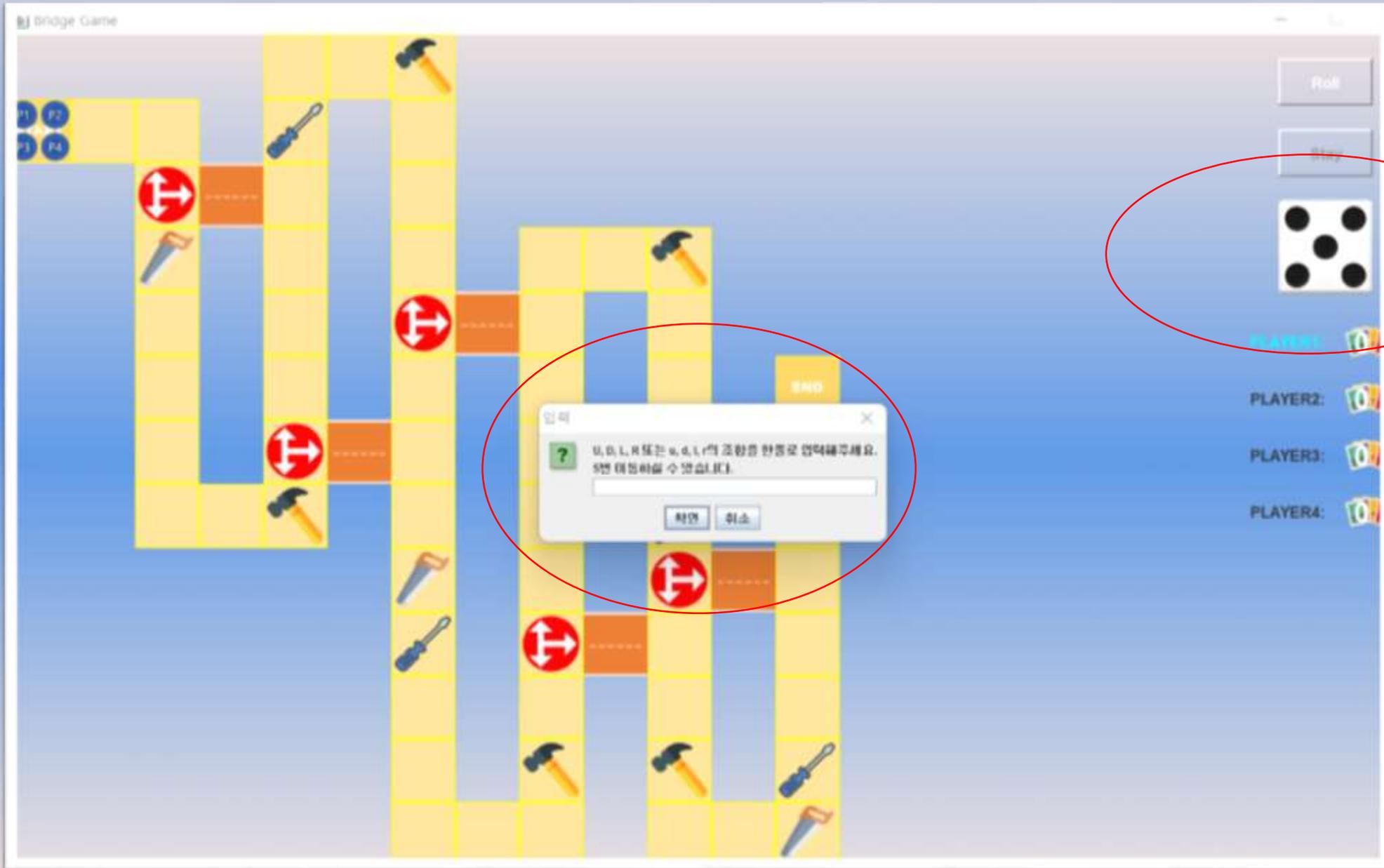


게임이 시작되었을 때의 화면. 각 플레이어는 **Start** 지점에 위치하며 오른쪽의 **PLAYER1** 처럼 현재 플레이어의 턴을 표시한다.

또한 각 플레이어의 카드의 수도 표시한다.

또한 **Roll** 버튼과 **Stay** 버튼 중 하나를 선택해 게임을 진행할 수 있다. 카드의 수가 0일 경우 **Stay** 기능은 동작하지 않는다.

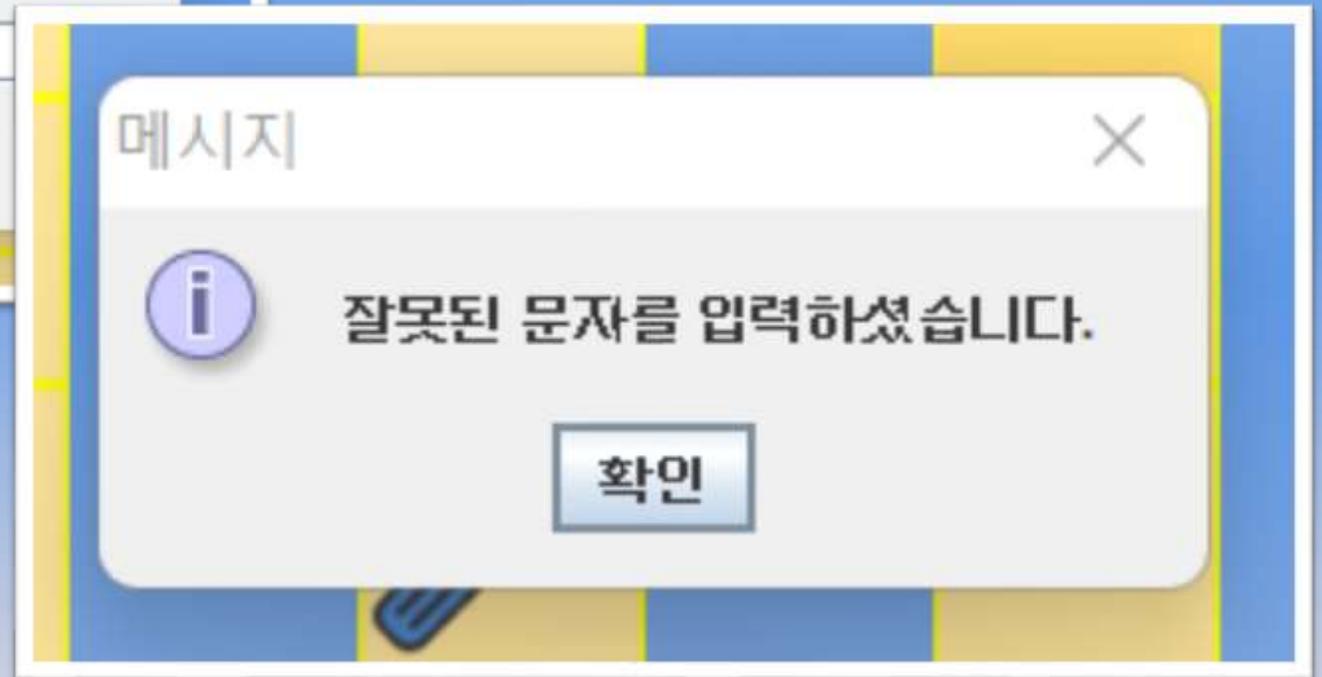
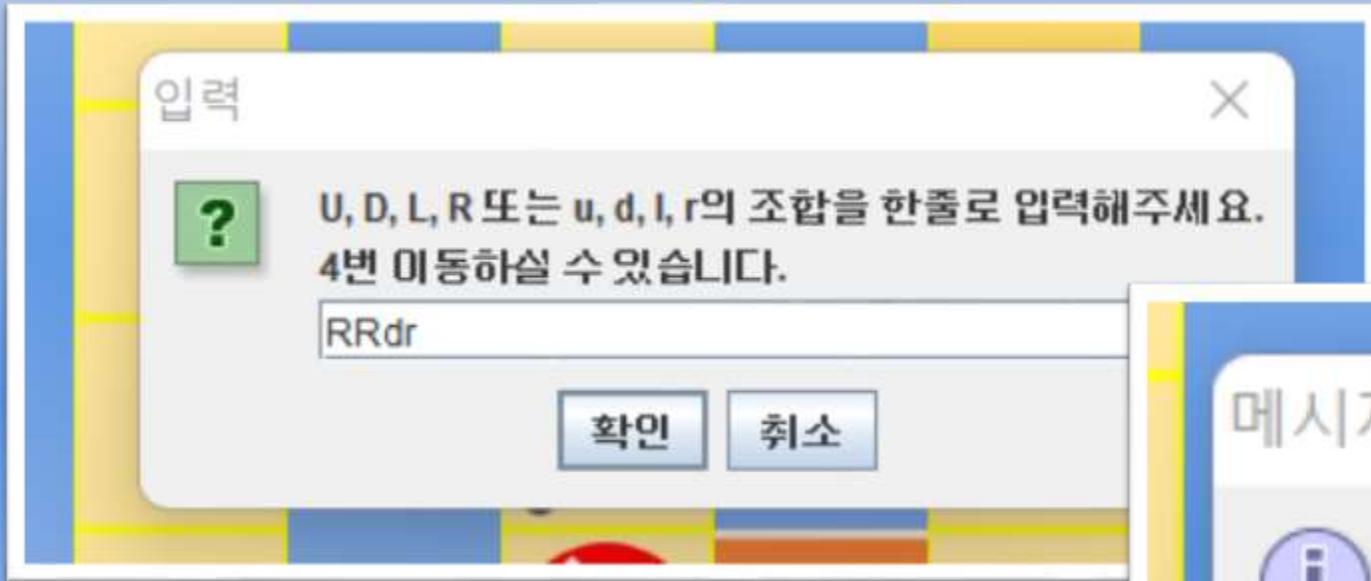
프로그램 사용 방법 - Roll & 입력



Roll 버튼을 눌렀을 때의 화면. 오른쪽에 주사위의 값을 표시하며 플레이어로부터 이동 커맨드를 입력받는다.

프로그램 사용 방법 - Roll & 입력 오류

만약 잘못된 문자를 입력할 경우 오류 메시지를 출력한 후 다시 입력 받는다.

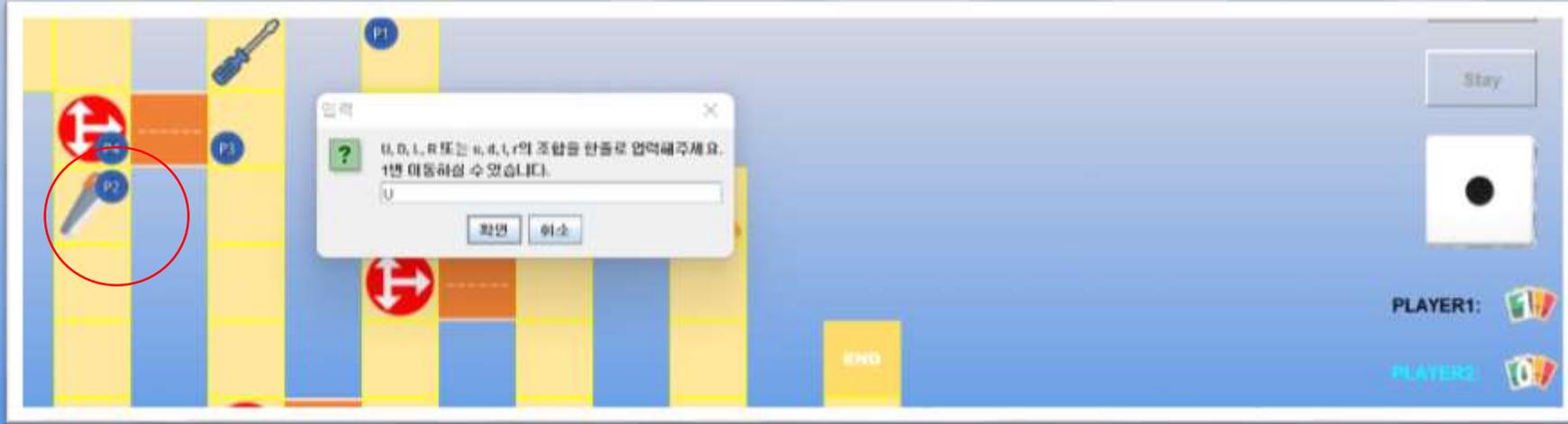


프로그램 사용 방법 - Roll & 입력 가능 횟수

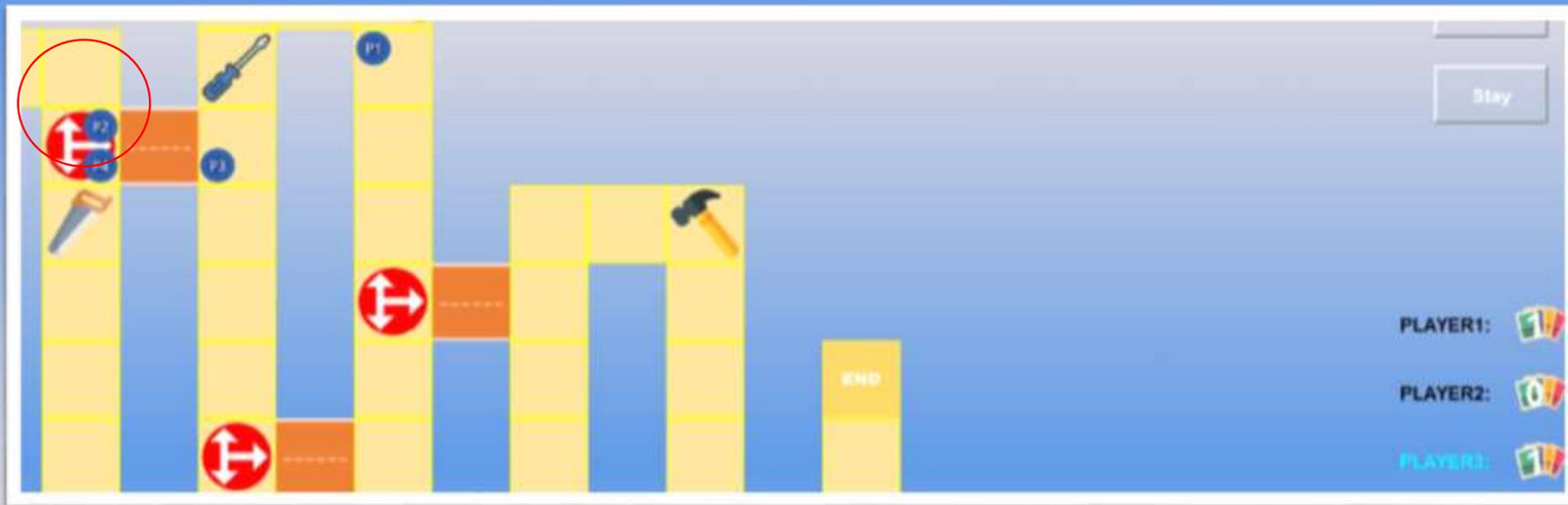


또한 플레이어가 이동할 수 있는 횟수는 (주사위의 값 - 플레이어가 갖고 있는 카드의 수)가 된다. 만약 이 값이 0 이하이면 이동 가능 횟수는 0이 된다.

프로그램 사용 방법 - Roll & 뒤로 이동



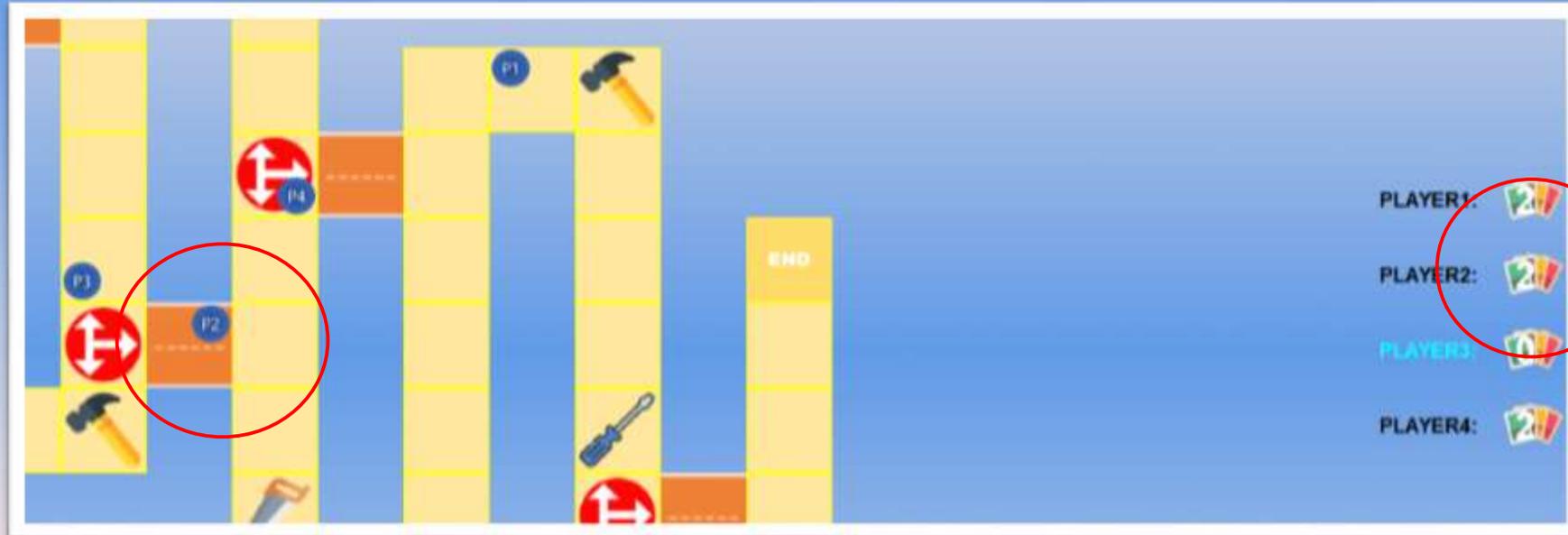
아직 아무도 End 지점에 도달하지 못했을 경우 모든 플레이어는 뒤로 이동이 가능하다.



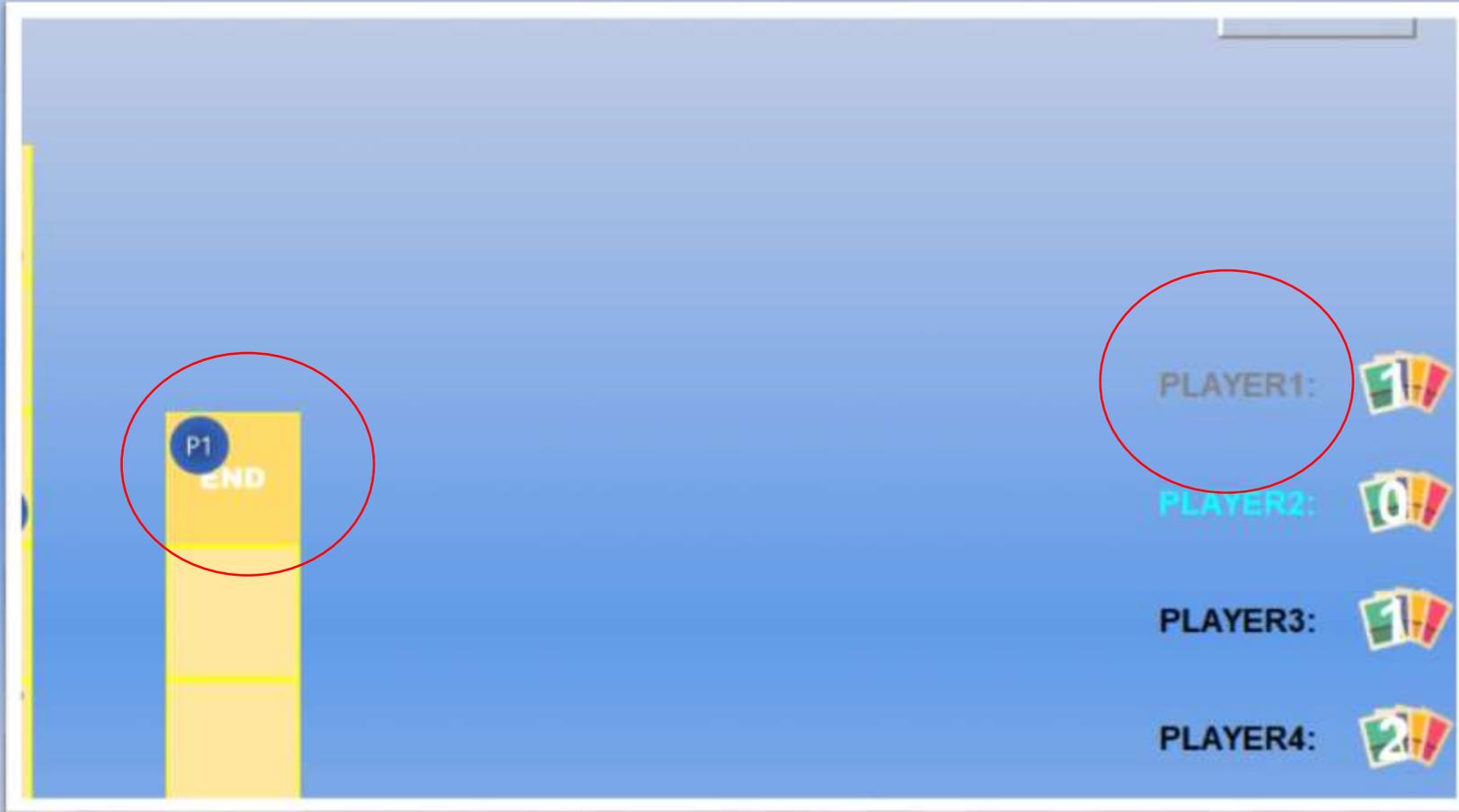
프로그램 사용 방법 - 다리 건너기



다리를 건너 플레이어는 다리 카드 1장을 받는다. 플레이어가 이동하여 다리 위로 올라가게 되면 카드 1장을 받는다.

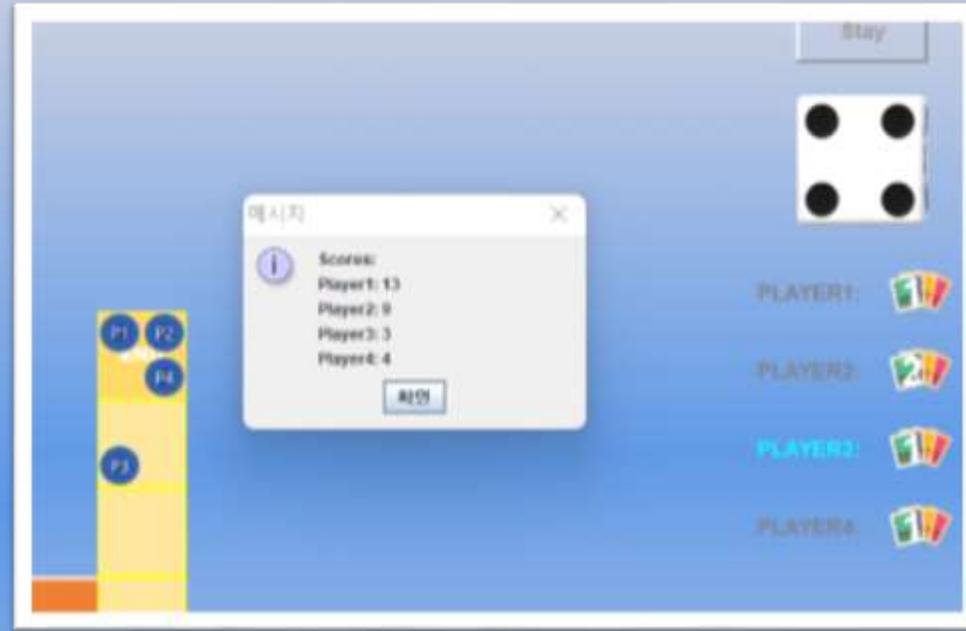


프로그램 사용 방법 - End 지점 도달



End 지점에 도달한 플레이어는 더 이상 게임에 참여할 수 없으며 화면 오른쪽처럼 **PLAYER1**의 색깔을 통해 게임을 마쳤다는 것을 나타낸다.

프로그램 사용 방법 - 게임 종료 후 점수 표시



한 명의 플레이어를 제외하고 모든 플레이어가 End 지점에 도달하면 게임은 종료되고 화면에 각 플레이어의 점수를 표시한다.