

# 과제 보고서

## SE TERM PROJECT



20230066 이희재

20233394 이인표

20230431 민준영

20230880 남궁하람

20233048 다이마리 케리스

## ※ 목차

---

1. 프로젝트 개요
  2. 유스케이스 모델
  3. Operation Contract and SSD
  4. 설계 및 구현 리포트
  5. 테스트 리포트
  6. GitHub 프로젝트 리포트
- 

## 1. 프로젝트 개요

### 1. 프로젝트 소개

- 윷놀이 게임을 JAVA를 이용해 OOAD 설계로 구현하였다.
- UI를 변경해도 문제없이 실행할 수 있게 설계를 했다.

### 2. 요구사항

- 기능적 요구사항 (Functional Requirements)
  - 판 선택 기능
    - 사각형, 오각형, 육각형 중 하나를 고른다.
  - 플레이어 수 선택 기능
    - 플레이어 수를 2명에서 4명까지 선택할 수 있어야 한다.
  - 말의 수 선택 기능
    - 각 플레이어는 2개에서 5개의 말을 선택할 수 있어야 한다.
  - 윷 던지기 기능
    - 플레이어는 매 턴마다 윷을 던지고, 윷이나 모가 나올 경우 한 번 더 던질 수 있어야 한다.
  - 테스트 윷 던지기 기능
    - 윷을 던져 나오는 6가지 결과가 제대로 반영되지 테스트 기

능을 통해 확인하여야 한다.

- 잡기 기능
  - 상대팀의 말을 잡을 수 있으며, 잡은 말은 시작점으로 돌아가야 한다.
- 엮기 기능
  - 자신의 말에 있는 곳에 도착 시, 그 다음 턴부터 두 말은 함께 이동하게 된다.

○ 비기능적 요구사항 (Non-Functional Requirements)

- Usability 요구사항
  - 게임의 UI는 직관적이어야 하고, UI가 바뀌어도 사용자는 문제없이 플레이 할 수 있어야 한다.
- Supportability 요구사항
  - 시스템은 Windows 및 Mac OS에서 모두 실행되어야 한다.

### 3. 빌드 및 실행 방법

○ 빌드 방법

1. Main.java 프로젝트를 빌드한다.

○ 실행 방법

1. 판의 종류를 선택한다.
  - 사각형
  - 오각형
  - 육각형
2. 사용자 수와 말의 수를 설정한다.
3. 시작 버튼을 클릭하여 게임을 시작한다.
4. 우승자가 나오면 게임이 종료된다.

### 4. 게임 규칙

- 사용자는 2...4인, 말의 개수는 2...5개이다.

- 이후 윷을 던지고, 그 결과만큼 말을 선택해서 이동한다.
  - 백도 - -1 칸 / 도 - 1칸 / 개 - 2칸 / 걸 - 3칸 / 윷 - 4칸 / 모 - 5칸
  - 윷, 모일 경우, 한 번 더 굴린다.
- 말이 분기점에 정확히 도착했을 경우
  - 그 상황에서의 shortest path로 이동한다.
- 그렇지 않을 경우
  - 항상 직선으로 이동해야 한다.

## 5. 설계 및 기술 요소

- 이 게임은 OOAD 방식을 사용하여 class 기반으로 구현되었다.
- 게임은 Java로 구성되었고, MVC 구조를 적용하여 게임 진행과 UI를 분리하였다.
- 사용자와 말은 각각 Player, Horse 클래스로 구현되며, 게임판과 칸들은 각각 Board, Cell 클래스로 구현된다.

## 2. 유스케이스 모델

- Use Case UC1: Yut
  - Scope: Yut 게임 시스템
  - Level: User goal
  - Primary Actor : 플레이어
- Stakeholders and Interests:
  - Player: 윷을 굴려서 자신의 말을 원하는 방식으로 이동시키며, 자신의 말을 모두 완료 상태로 만들어 정상적으로 게임을 종료한다.
  - Other Players: 규칙에 따라 상대가 잡기/업기 기능을 사용할 수 있다.
  - system : 플레이어의 입력에 따라 올바르게 윷 결과 및 말 이동을 처리한다.
- Preconditions:
  - 게임이 시작된 상태이며, 플레이어의 턴이 되어 있다.
  - 윷 던지기 버튼이 활성화되어 있다.
- Success Guarantee (Postconditions):
  - 윷 결과가 저장된다.
  - 플레이어가 선택한 말이 해당 결과만큼 이동한다.
  - 윷/모/잡기/업기 등의 상황이 나올 경우 한 번 더 던진 후, 그 결과가 시스템에 저장되어야 한다.
  - 현재 말의 위치가 업데이트된다.
- Main Success Scenario (Basic Flow):
  1. 플레이어는 윷 던지기 버튼(랜덤)을 클릭한다.
  2. 윷 결과(도, 개, 걸, 윷, 모, 뺑도)가 저장된다.
    - 2-1. 윷, 모가 나온 경우, 추가 턴이 주어진다.
  3. 플레이어는 이동시킬 말을 선택한다.

4. 선택된 말을 윷 결과만큼 이동시킨다.

5. 다음 턴으로 넘어가거나, 추가 턴이 있으면 해당 플레이어의 턴이 계속된다.

- Extensions (Alternative Flows):

- 2a. 지정 윷 던지기를 선택한 경우:

- 1. 플레이어가 윷 종류를 직접 선택하므로, 테스트의 경우이다.

- 4a. 이동한 칸에 상대편 말이 있는 경우:

- 1. 잡기 기능 수행 후, 상대편의 말은 대기 상태로 돌아간다.

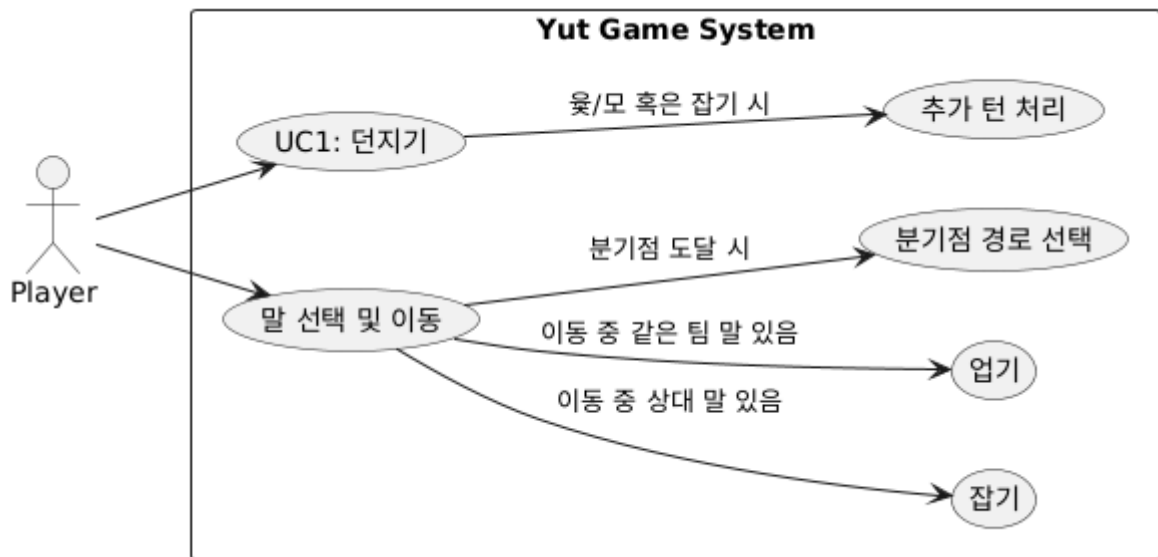
- 2. 상대방 말을 잡은 플레이어는 추가로 한 번 더 윷을 던질 수 있다.

- 4b. 이동한 칸에 같은 팀의 말이 있는 경우:

- 1. 업기 기능 수행 후, 업은 말과 함께 업힌 말도 이제부터 함께 이동한다.

- 4c. 분기점에 도달할 경우:

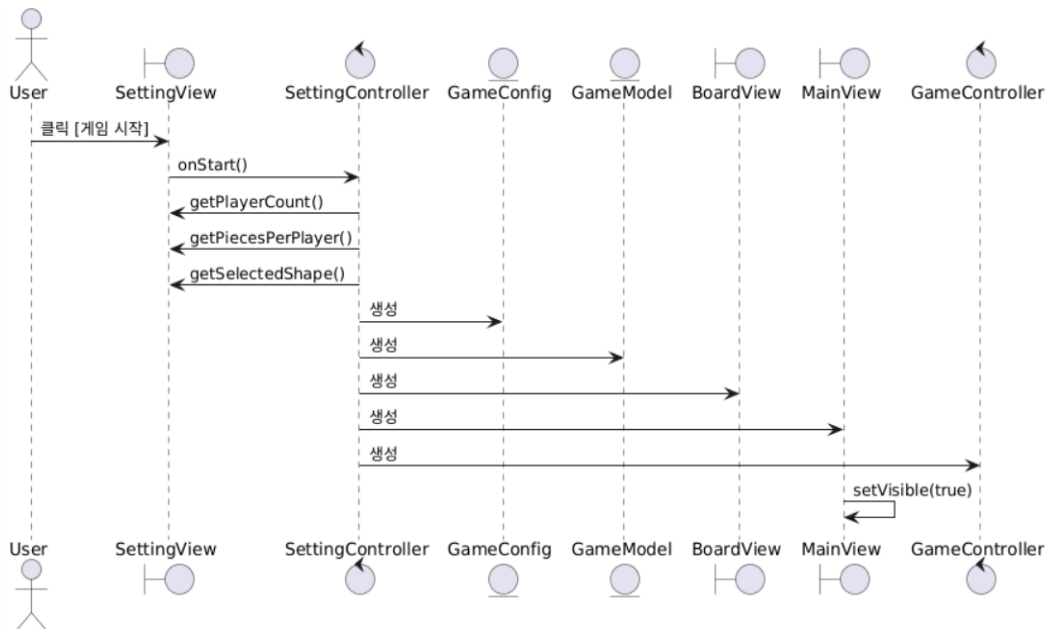
- 1. 그 상황에서의 shortest path로 경로를 변경한다.



### 3. Operation Contract and SSD

- 시나리오 1
  - 게임 시작 (설정 → 실행)

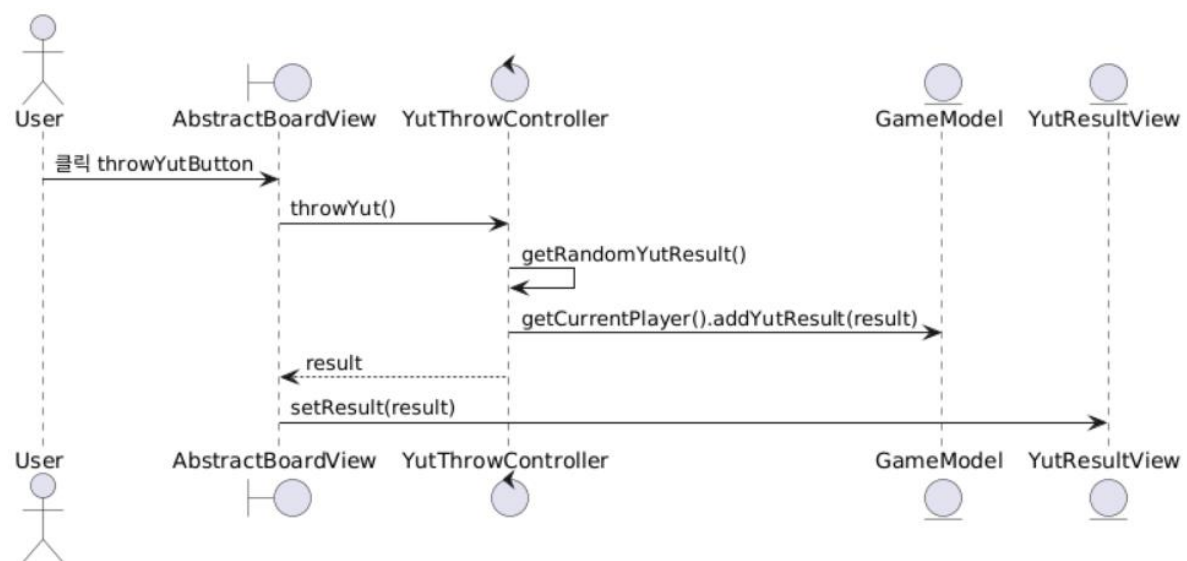
항목	내용
Operation Name	startGame
Responsibility	사용자로부터 입력받은 설정값을 기반으로 게임을 초기화하고 시작 화면을 띄운다.
Cross References	시나리오 1, "게임 시작" 버튼 클릭
Preconditions	SettingView가 활성화되어 있어야 하며, 사용자가 유효한 입력을 제공해야 한다.
Postconditions	1) GameConfig, GameModel, BoardView, MainView, GameController 인스턴스가 생성된다. 2) MainView가 사용자에게 표시된다.
Output	초기화된 게임 화면(MainView), 게임 데이터 모델(GameModel)



- 시나리오 2

- 윷 던지기 버튼 클릭 → 결과 출력

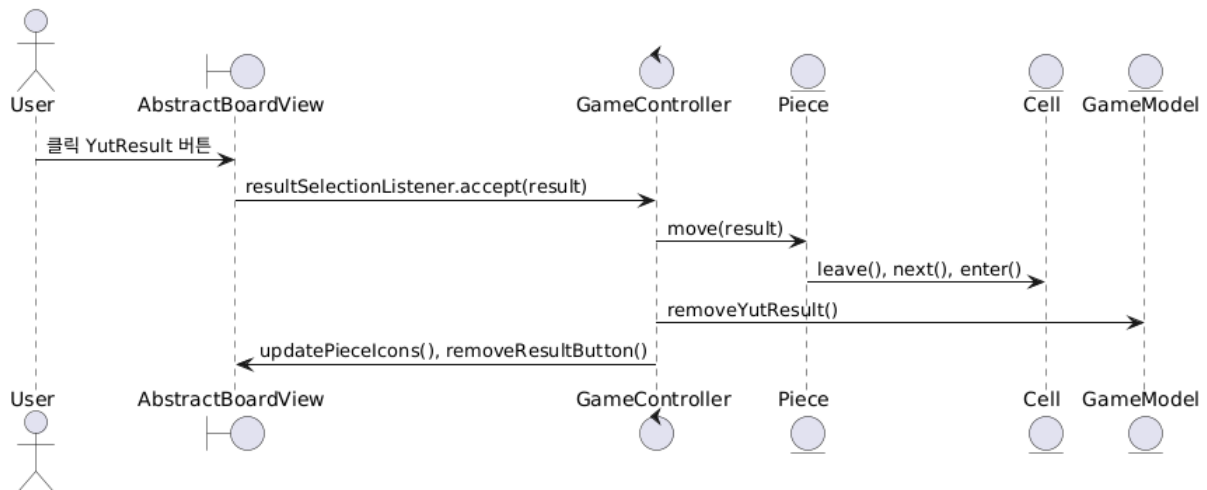
항목	내용
Operation Name	throwYut
Responsibility	현재 플레이어의 윷 던지기 결과를 생성하고 결과 화면에 출력한다.
Cross References	시나리오 2, throwYutButton 클릭
Preconditions	게임이 시작되어야 하며, 현재 플레이어가 윷 던지기를 할 차례여야 한다.
Postconditions	1) YutThrowResult가 생성되어 현재 플레이어의 history에 저장된다. 2) 결과가 YutResultView에 출력된다.
Output	YutThrowResult 객체





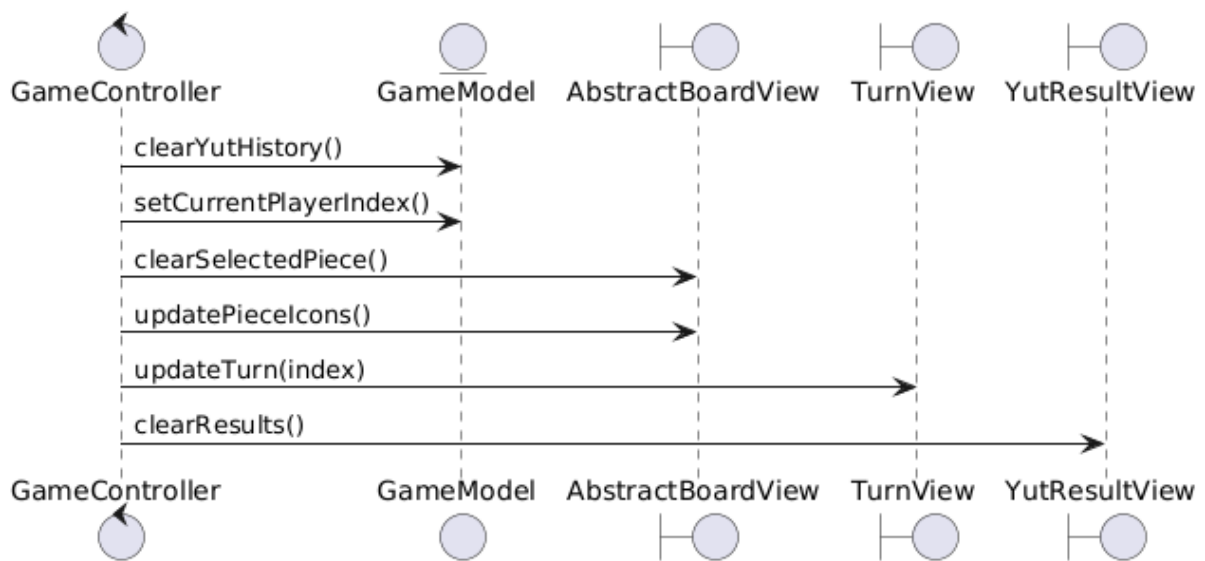
- 시나리오 3
  - 윷 결과 선택 → 말 이동

항목	내용
Operation Name	selectYutResult
Responsibility	사용자가 선택한 윷 결과에 따라 말 이동을 처리한다.
Cross References	시나리오 3, 윷 결과 버튼 클릭
Preconditions	유효한 윷 던지기 결과가 존재해야 하며, 이동 가능한 말이 선택되어야 한다.
Postconditions	1) 선택된 말이 이동된다. 2) 해당 결과가 사용 처리되어 버튼에서 제거된다.
Output	이동 후 말의 위치 및 게임 모델 갱신 상태



- 시나리오 4
  - 턴 전환

항목	내용
Operation Name	switchTurn
Responsibility	현재 플레이어의 턴이 종료되면 다음 플레이어로 턴을 넘기고 화면 정보를 업데이트한다.
Cross References	시나리오 4
Preconditions	현재 턴의 말 이동과 결과 처리가 완료되어야 한다.
Postconditions	1) 현재 플레이어 index가 다음으로 설정된다. 2) 관련 UI(TurnView, YutResultView 등)가 초기화된다.
Output	업데이트된 현재 턴 플레이어, 클리어된 결과 뷰



## 4. 설계 및 구현 리포트

### 1. System Architecture

- 이 프로젝트는 Java 언어와 Swing GUI 라이브러리와 FX를 활용하여 제작하였다.
- 전체 시스템은 MVC 패턴을 따른다.
- 객체지향적 설계를 하였다.
  - 각 객체는 각각의 Responsibility를 갖고 분리되어 있으며, 생성자와 캡슐화를 활용하였다.
- Polymorphism
  - Board가 다양한 형태(사각형, 오각형, 육각형)를 가질 수 있도록 하였다.
- UML 모델링 가능
  - 클래스 간의 관계(Association, Composition, Dependency, Inheritance), Multiplicity, Navigability가 명확하게 정의되어 있다.
- 업기/캡처 등 복잡 로직 분리
  - Cell.enter()와 Piece.move()에 윷놀이의 게임 규칙이 모듈화 되어있다.

#### <Model>

- 게임 상태와 데이터를 관리하는 영역이다.
- 플레이어, 말(Piece), 보드(Board), 셀(Cell) 등 게임의 핵심 로직이 포함된다.

클래스	설명
GameModel	게임의 전체 상태 관리 (보드, 플레이어, 턴, 승리 여부 등)
Player	개별 플레이어 정보와 말(Piece) 상태 관리
Piece	실제 말 객체, 이동 및 업기/잡기 등의 로직 포함
Cell	보드의 각 칸을 나타내며 말의 위치, 경로 정보 포함
Board 및 하위 클래스	RectangleBoard, PentagonBoard, HexagonBoard 등 보드 구조 정의
YutThrowResult	윷 결과값을 Enum으로 정의 (도, 개, 걸, 윷, 모, 뱃도)
GameConfig	게임 초기 설정(말 수, 보드 모양, 인원 수 등) 담당

## <View>

- 사용자에게 화면을 보여주는 화면을 표현하는 역할을 담당한다.
  - 말의 위치, 현재 플레이어, 윷 결과 등을 시각적으로 사용자에게 전달한다.
- MainView를 중심으로 다양한 BoardView, ControlPanelView, TurnView, SettingView 등이 있다.
- 직접 Model을 수정하지 않고, 오직 Controller를 통해 간접적으로 작용한다.
- Swing 또는 JavaFX를 통한 GUI 구현을 가능하게 했다.
  - 코드 재사용의 비율을 높여 효율성을 추구했다.

## <Controller>

- 사용자 입력을 처리하고 Model의 메서드를 호출하여 상태를 변경한다.
- 변경된 Model의 상태를 View에 반영하도록 한다.
- 모델과 뷰 사이의 중재 역할이다.
- GameController를 중심으로 YutThrowController, PieceMovementController, SettingController 등이 있다.
  - 윷 던지기 입력 → YutThrowResult 생성 → 플레이어에 저장
  - 말 선택 및 이동 → Piece.move(steps) 호출
  - 턴 제어 및 승리 조건 확인 → GameModel.getWinner() 등 호출

## - MVC 요약

- Model은 게임의 상태와 로직을, View는 사용자와의 인터페이스를, Controller는 입력과 데이터 사이의 연결을 담당하도록 했다.
- UI 구현 방식 변경(Swing, JavaFX) 또는 게임 규칙 확장 시에도 Model을 수정하지 않고 View/Controller만 교체함으로써, 안정적인 구조를 유지할 수 있다.
- MVC 아키텍처에 따라 역할 분리를 통해 코드 유지 보수성과 독립성을 확보하였다.

## 2. UI 변경에 따른 코드 재사용 (Swing & JavaFX)

### a. UI ToolKit 변경

- 두 개의 UI ToolKit(Swing, JavaFX)를 사용할 수 있도록 개발하였다.
- Model과 Controller 패키지는 공통으로 하되, View 패키지만 별도로 구현하여 높은 코드 재사용률로 효율적인 설계가 가능하도록 하였다.

### b. 공통 코드

- Model (GameModel , Player, Piece, Board 등)의 경우, 완전히 동일한 클래스 구조를 Swing과 JavaFX 양쪽에서 사용하며 코드 재사용률을 높였다.
- Controller(GameController, YutThrowController 등)도 클래스 이름, 기능, 그리고 내부 로직은 거의 일치한다.
  - 단지 UI에 따라, 이벤트 처리 방식만 각 UI toolkit에 맞게 다르게 구현되었다.

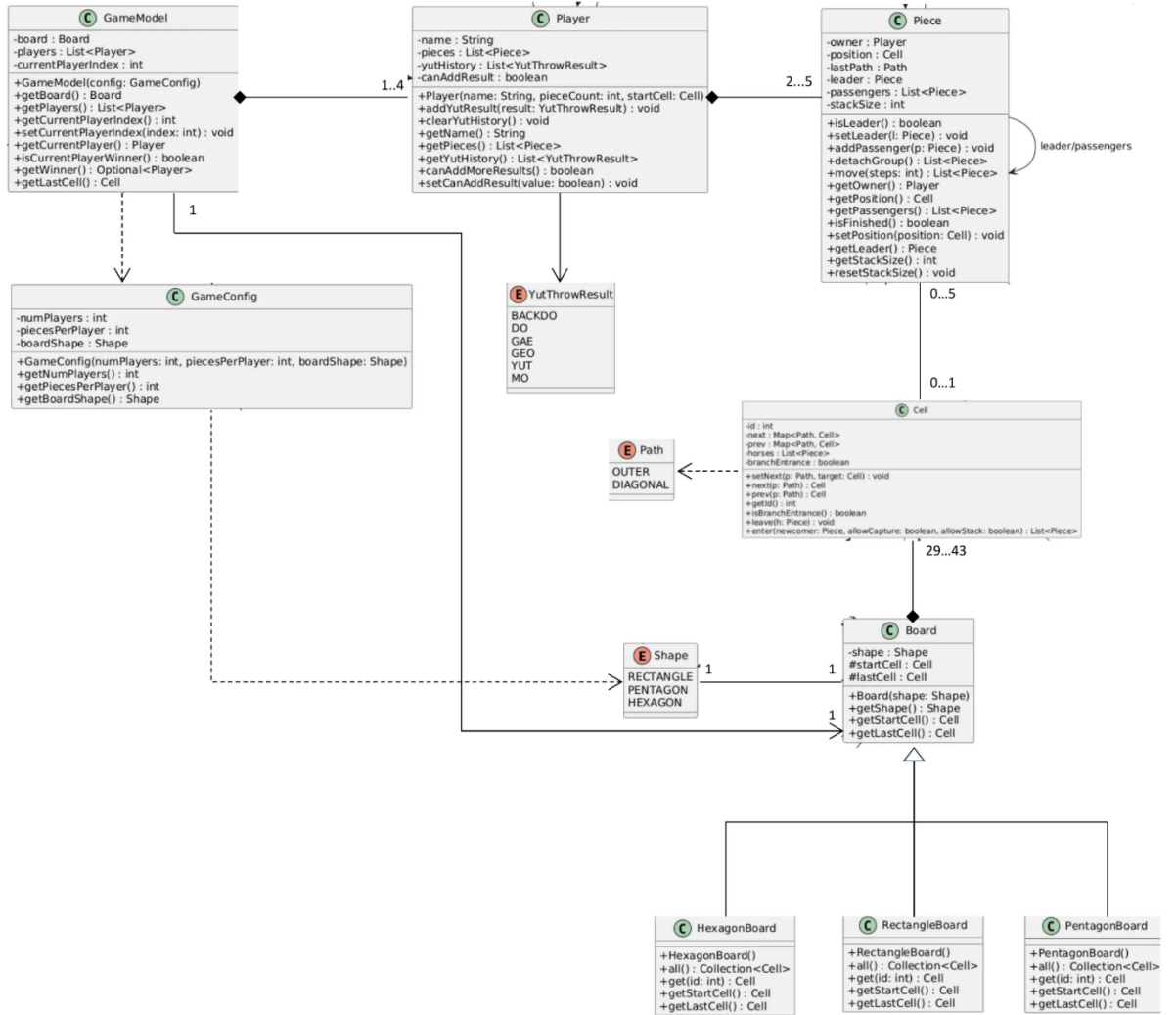
### c. 변경 부분

- 변경된 부분은 View 패키지에서 나타난다.
  - Swing에서는 JButton, JPanel, JFrame 등 Swing 컴포넌트를 사용하였다.
  - JavaFX에서는 Button, VBox, Stage 등 JavaFX의 컴포넌트를 사용하였다.
- View 클래스 구조(MainView, RectangleBoardView, SettingView 등)는 동일하게 유지되어, 유지보수성과 비교 테스트에 용이하다.

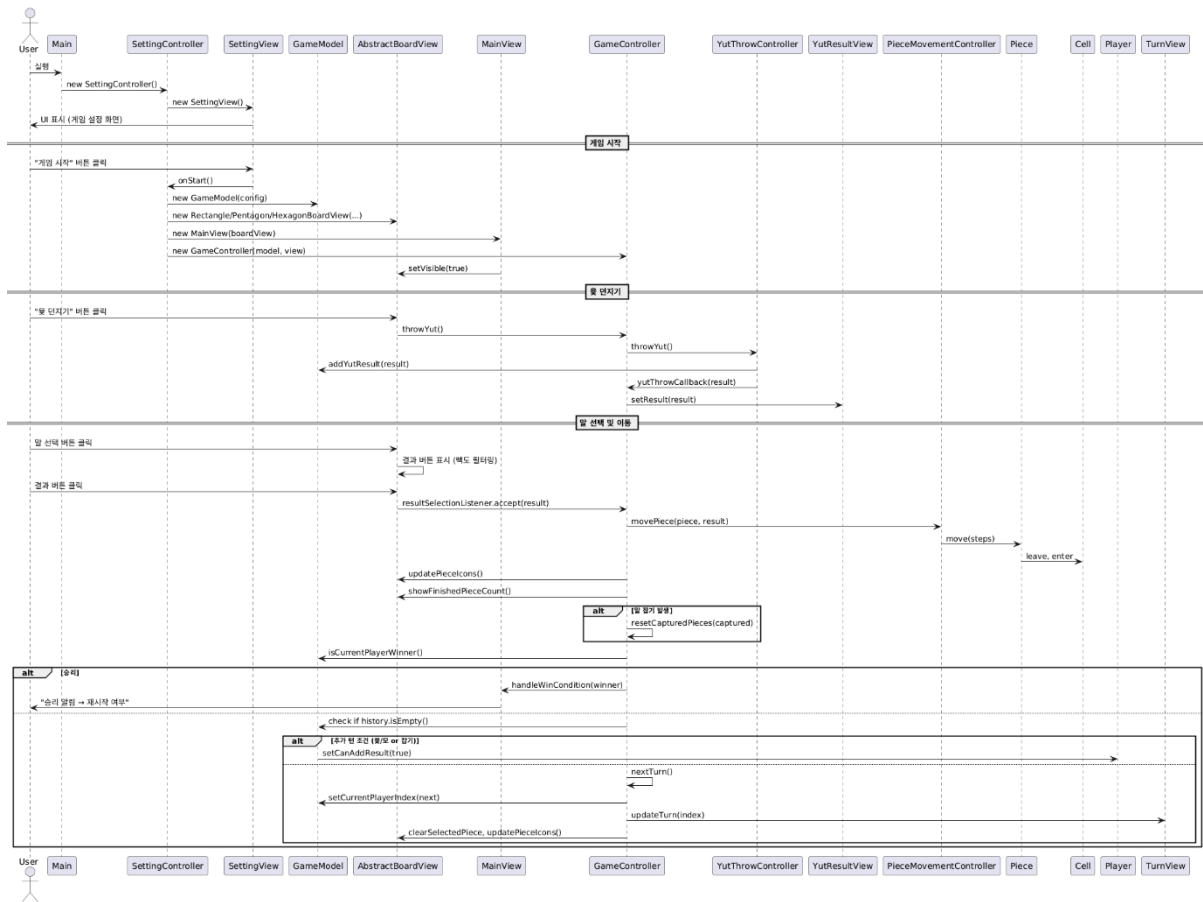
### d. 요약

- UI 관련 패키지인 view만 교체되고 전반적인 로직 재사용 가능하도록 하며, MVC 아키텍처의 이점을 활용했다.
- UI 변경 시 발생할 수 있는 중복 개발 및 오류를 최소화했다.

### 3. 클래스 다이어그램



#### 4. 시퀀스 다이어그램 (Swing 기준, FX도 메인 제외 유사함)



##### 1. 게임 시작

- 사용자가 게임을 실행하면 SettingView 를 통해 플레이어 수, 말 수, 판 모양을 선택한다.
- 설정 완료 후 SettingController 가 GameModel, GameController, MainView 를 초기화하며 게임이 시작된다.

##### 2. 윷 던지기

- 플레이어가 "윷 던지기" 버튼을 누르면 YutThrowController 가 랜덤 윷 결과를 생성한다.
- 해당 결과는 모델(GameModel)에 저장되며, 우측의 YutResultView 에 텍스트로 출력된다.

##### 3. 말 선택 및 이동

- 플레이어는 자신의 말을 선택하고 원하는 윷 결과 버튼을 클릭하여 이동을 지시한다.
- GameController 는 PieceMovementController 를 통해 해당 말을 이동시키고, Piece 클래스가 보드상 셀(Cell)을 따라 실제 위치를 변경한다.

#### 4. 잡기 및 업기

- 이동한 위치에 상대 말이 있을 경우 해당 말을 잡으며, 같은 편 말과는 업기를 통해 그룹을 이룬다.
- 잡힌 말은 출발점으로 되돌아간다.

#### 5. 턴 전환 또는 추가 턴

- 잡기를 했거나 뿔/모가 나온 경우, 플레이어는 추가로 턴을 진행할 수 있다.
- 그렇지 않으면 `nextTurn()`이 호출되어 다음 플레이어로 턴이 전환되며, UI도 이에 맞춰 갱신된다.

#### 6. 승리 판단

- 한 플레이어가 모든 말을 완주시켰을 경우, 게임은 종료되고 승리 알림과 재시작 여부를 묻는 다이얼로그가 출력된다.

### 4. Key Implementaion Details

#### • 게임 관리

- `GameModel`
  - 게임 전체의 상태를 관리하는 중심 클래스이다.
  - `GameConfig`를 기반으로 보드, 플레이어, 말을 초기화하고 현재 턴과 승자 여부를 판단한다.
  - `getWinner()`, `isCurrentPlayerWinner()` 등을 통해 게임 종료 조건을 체크한다.
  - `GameModel`은 `Board`, `Player`, `GameConfig`와는 직접 연결되고, `Cell`, `Piece`와는 간접적으로 연결된다.

#### • 게임 기본 값 설정

- `GameConfig`
- 플레이어 수, 말 개수, 보드 형태(사각형/오각형/육각형)를 저장한다.
- `GameModel`의 초기화에 사용되며, 생성자 주입 방식으로 설계된다.
- 보드 및 플레이어 생성을 설정값을 통해 제어할 수 있도록 구성된다.

#### • 보드 구성

- `Board`
  - `Board`는 `Shape`, `startCell`, `lastCell`을 필드로 가진다.
  - `RectangleBoard`, `PentagonBoard`, `HexagonBoard`는 각기 다른 셀 구조와 경로를 설정한 구현체이다.

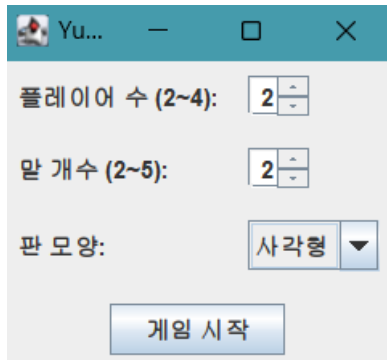


- 내부적으로 Map<Integer, Cell>을 통해 셀을 관리하고, linkOuterRing(), linkDiagonals()로 경로를 연결한다.
  - 보드 셀
    - Cell
      - 보드 위 각 칸을 표현하는 클래스이다. id, 다음/이전 셀 정보, 말을 담는 리스트 등을 포함한다.
      - enter()와 leave()를 통해 말의 진입/이탈을 처리하며, 잡기, 업기, 도착 등의 로직이 포함된다.
      - Cell.Path 를 이용해 OUTER/DIAGONAL 경로를 구분하며, 분기점 이동을 유연하게 처리한다.
  - 말
    - Piece
      - 플레이어에 속하며 현재 위치와 업기 여부, 그룹화 상태를 가지고 있다.
      - move(int steps)를 통해 이동하며, 분기점 경로 선택, 잡기, 도착, 업기 등 윷놀이 핵심 규칙을 수행한다.
      - 도착했을 경우, position == null 로 설정되어 도착 여부를 구분한다.
      - 말이 업혀 있을 경우 leader, passengers 를 통해 그룹으로 이동하고 해제된다.
      - 예외 상황으로, 업힌 말을 선택하면 move()에서 예외가 발생한다.
      - 그래서, GameController.setupResultButtonHandler()에서 리더를 자동 선택하도록 보완하였다.
  - 플레이어
    - Player
      - 이름, 보유 말 리스트, 윷 결과 히스토리를 저장한다.
      - 윷/모에 의한 추가 턴 여부는 내부 플래그로 판단한다.
      - 모든 말이 아직 출발하지 않은 상태에서 뺑도만 나올 경우 게임이 멈추는 문제가 발생했었다.
- YutThrowController 와 GameController 양쪽에서 이를 감지해 자동으로 턴을 넘기도록 처리해 해결하였다.
- 윷 던지기 결과
    - YutThrowResult

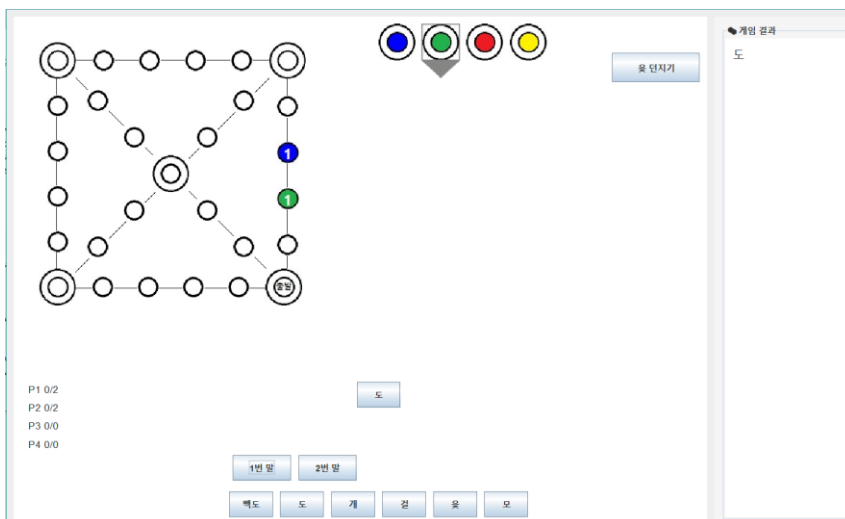
- 도/개/걸/웃/모/뺨도의 결과값을 갖는 enum 클래스이다.
- 각 결과는 이동 거리와 한글 표기를 가지며, toString()으로 출력할 수 있다.
- 웃/모 여부는 추가 턴 판단에 사용된다.
- 추가 턴은 마지막으로 움직인 말이 아니라, 마지막으로 던진 웃 결과가 웃/모일 경우에만 발생해야 하므로
- GameController.setupResultButtonHandler() 내부의 턴 종료 판단 조건을 이에 맞춰 수정하였다.
- 웃 던지기 컨트롤
  - YutThrowController
    - 웃을 던지고 결과를 게임에 반영하는 역할을 한다.
    - 모든 말이 아직 출발하지 않았고 뺨도만 나온 경우, 이동 가능한 말이 없어 게임이 멈추는 문제가 있었다.
    - 이를 감지하여 자동으로 턴을 넘기도록 하였다.
    - 이 로직을 랜덤 웃, 고정 웃 양쪽 모두에서 작동하게 하기 위해 중복으로 구현하였다.
    - 그러나 게임 도중 말이 이미 도착한 이후에 뺨도만 남았을 때는 작동하지 않아
  - GameController 에도 동일한 조건의 로직을 추가하여 처리하였다.
- 분기점 관리
  - Cell.Path
    - Cell 내부 enum 으로, OUTER 와 DIAGONAL 경로를 나타낸다.
    - 각 셀의 next, prev 는 Map<Cell.Path, Cell> 형태로 구성되어 있어서 경로 설정이 유연하다.
    - 분기점 셀(isBranchEntrance) 여부에 따라 대각선 이동이 가능하며, Piece.move() 내부에서 분기 방향을 선택하게 된다.

## 5. UI 디자인

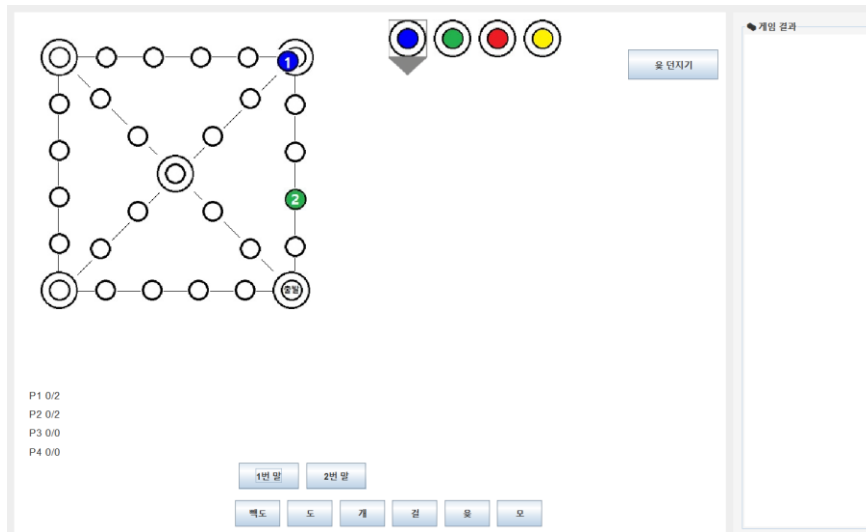
- Swing



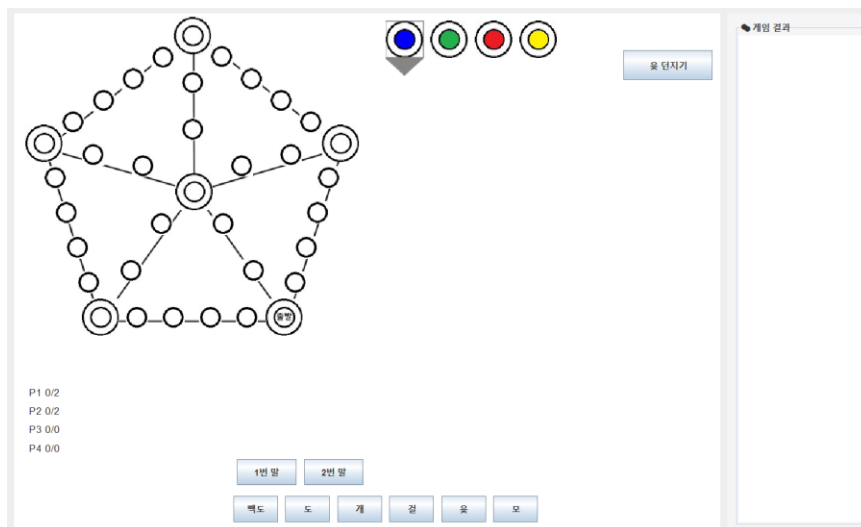
- 게임 시작 전 설정 화면
  - 플레이어 수 결정
  - 말 개수 결정
  - 판 모양 결정
  - 게임 시작 버튼



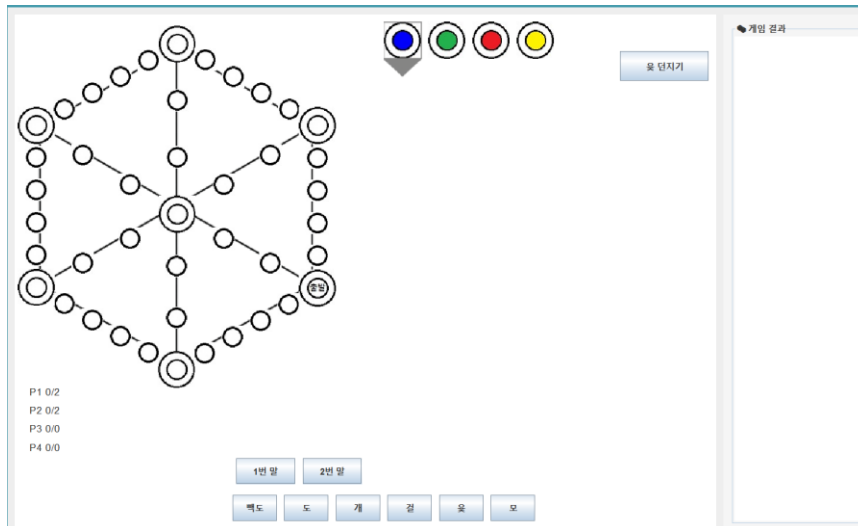
- 게임 화면
  - 좌측 상단에 판 위치
  - 중앙 상단에 색깔별로 플레이어 표시 및 화살표로 턴 표시
  - 우측 상단에 옷 던지기 버튼
  - 우측에 게임 결과(옷 던지기 결과 표시)
  - 하단에 말 선택 버튼, 테스트 옷 선택 버튼
  - 우측 하단에 랜덤 옷 던지기 버튼
  - 좌측 하단에 플레이어 별 완주한 말의 개수 표시
  - 옷 던지기 결과가 나오면 적용시킬 말 선택



- 적용시킬 말을 선택하면 적용 전에 판 위의 말에 보라색 사각형으로 표시  
➔ 던진 결과 버튼을 누르면 적용됨
- 같은 팀의 말을 업을 시 말 위에 업은 말의 개수 표시

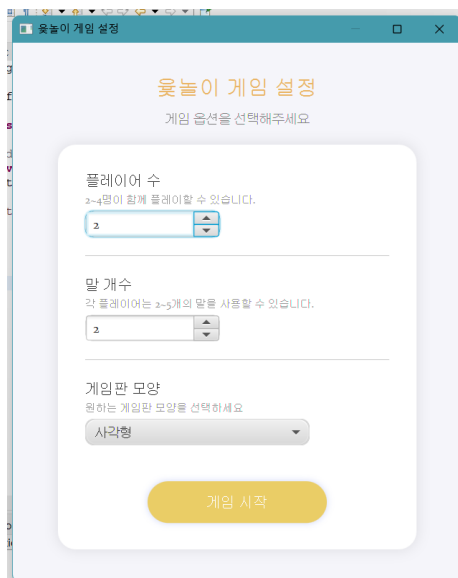


- 오각형판 게임 화면
  - 기능은 사각형 판과 동일

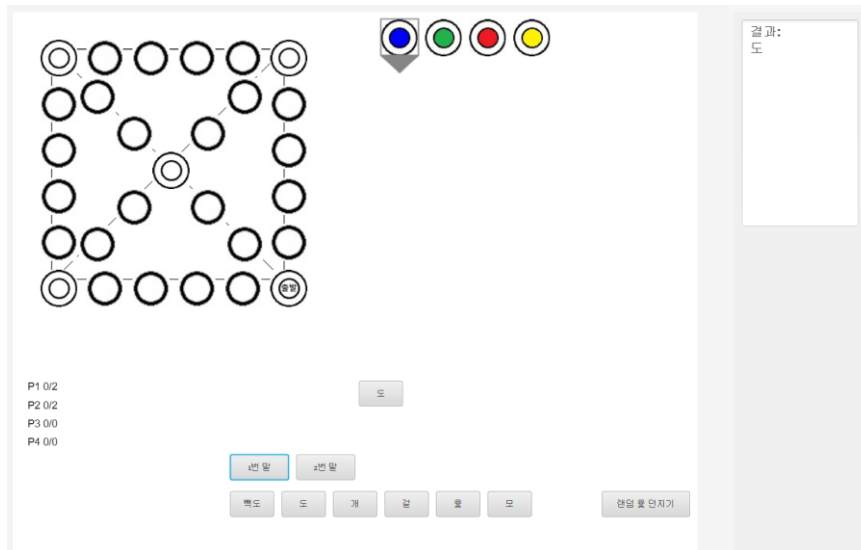


- 육각형판 게임 화면
  - 기능은 사각형 판과 동일

- JavaFX

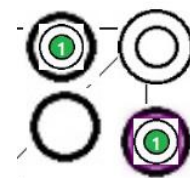
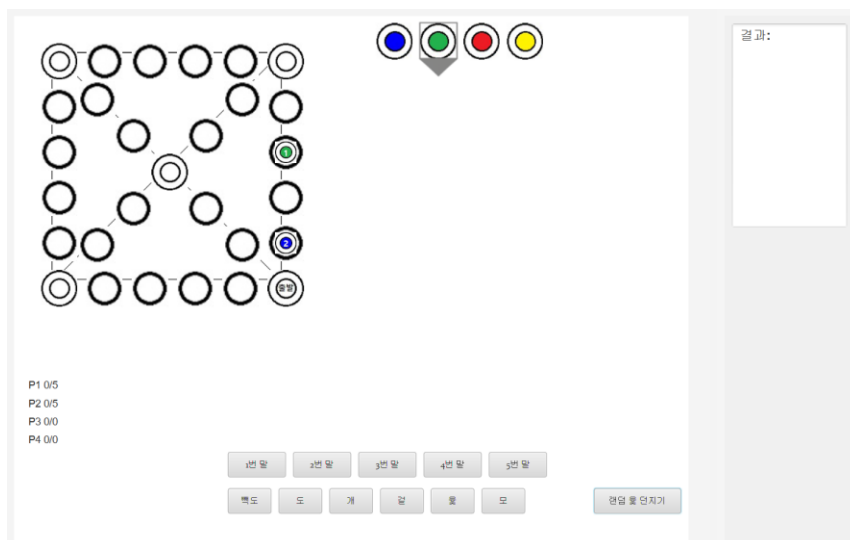


- 게임 시작 전 설정 화면
  - 플레이어 수 결정
  - 말 개수 결정
  - 판 모양 결정
  - 게임 시작 버튼

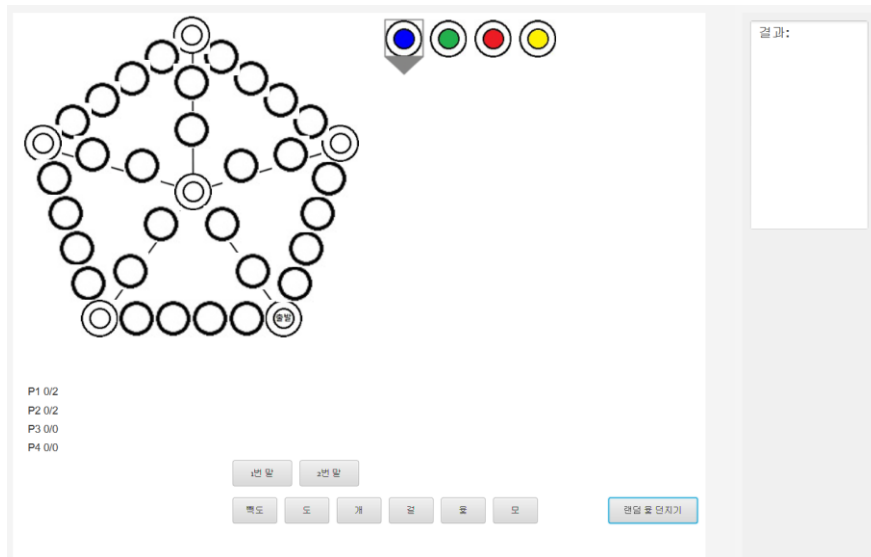


## • 게임 화면

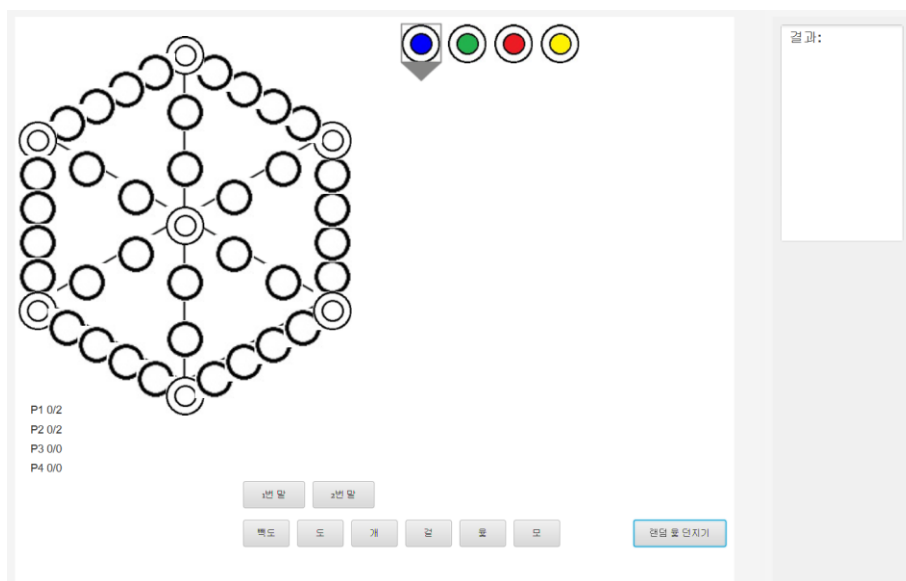
- 좌측 상단에 판 위치
- 중앙 상단에 색깔별로 플레이어 표시 및 화살표로 턴 표시
- 우측 상단에 윷 던지기 버튼
- 우측에 게임 결과(윷 던지기 결과 표시)
- 하단에 말 선택 버튼, 테스트 윷 선택 버튼
- 우측 하단에 랜덤 윷 던지기 버튼
- 좌측 하단에 플레이어 별 말의 현재 상황 표시
- 윷 던지기 결과가 나오면 적용시킬 말 선택



- 적용시킬 말을 선택하면 적용 전에 판 위의 말에 보라색 원으로 표시  
➔ 던진 결과 버튼을 누르면 적용됨
- 같은 팀의 말을 업을 시 말 위에 업은 말의 개수 표시



- 오각형판 게임 화면
  - 기능은 사각형 판과 동일



- 육각형판 게임 화면
  - 기능은 사각형 판과 동일

## 6. 구현 중 겪은 문제와 해결 방식

- 프로젝트 초반에는 단순히 작동하는 프로그램을 만드는 데 집중했고, Swing 을 기반으로 기능을 구현해 나갔다. 그래서, 이때는 MVC 구조는 형식적으로만 있었고, 로직과 UI 코드가 뒤섞여 있어 구조적인 문제점이 많았다.

이러한 방식을 사용했기에, 기능 하나를 추가할 때도 여러 컴포넌트를 동시에 수정해야 했고, 유지보수에 많은 시간과 노력이 들었다. 특히 JavaFX 로의 전환을 시도하면서 기존 구조로는 UI 변경이 거의 불가능하다는 문제가 생겼다.

이후 구조적 문제를 인식하고 클래스 다이어그램을 포함한 전체 설계를 재정비하였다. Model, View, Controller 의 책임을 명확히 분리하고 각 패키지가 독립적으로 동작하도록 구성하였다. 그 결과 기능 추가나 UI 변경이 훨씬 수월해졌고, JavaFX 로의 전환할 때도 Model 과 Controller 코드를 거의 그대로 재사용할 수 있었다.

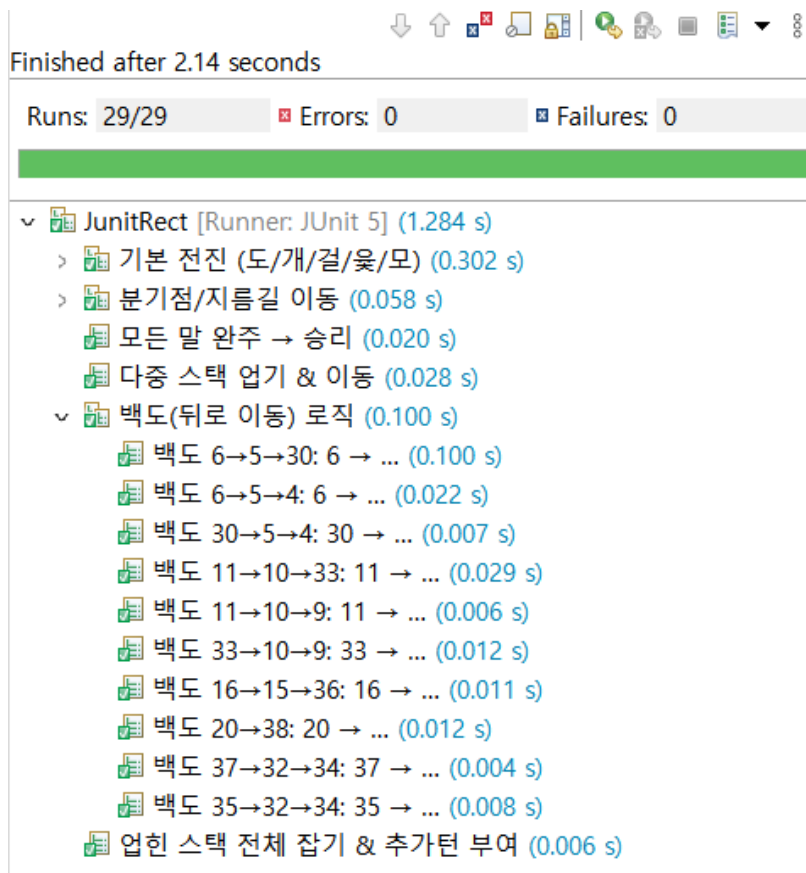
이번 프로젝트를 통해 단순히 작동하는 프로그램을 만드는 것을 넘어서, 유지보수성과 확장성을 고려한 구조적 설계가 왜 중요한지를 실감할 수 있었다. 시행착오도 많았지만, 그만큼 설계 역량을 키우고 개발자로서 성장할 수 있었던 값진 경험이었다.

## 5. 테스트 리포트

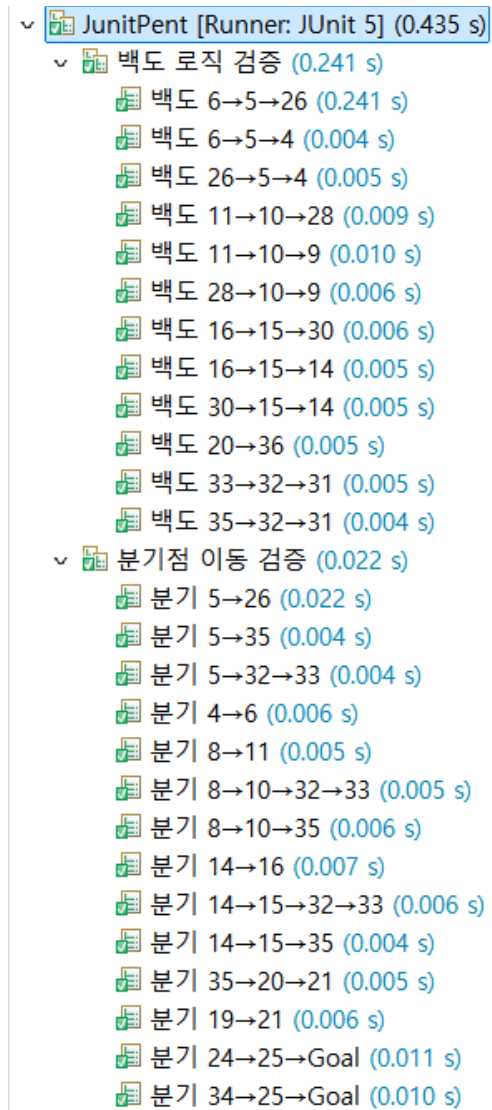
- 테스트 케이스
  - 기본 이동 케이스
    - 도, 개, 걸, 옷, 모를 통해 하는 기본적인 이동을 시험하기 위한 케이스였다.
    - 옷, 모일 경우 추가 턴이 발생하는지도 테스트하였다.
  - 분기점 case
    - 각 분기점마다 이동할 수 있는 경로를 테스트하였다.
  - 뱃지 case




- 백도로 인해 이동할 수 있는 경로를 테스트하였다.
      - 해당 플레이어의 모든 말이 0 번 cell 에 있거나 나갔을 경우의 백도는 컨트롤러에서 처리를 하기 때문에 이 Test 에서는 다루지 않았다.
    - 분기점에서의 백도가 나오는 경우도 포함하였다.
  - 잡기 case
    - 업혀 있는 말을 모두 잡는지 확인하였다.
    - 잡고 나서 한 번 더 던질 수 있는지 확인하였다.
  - 업기 case
    - 여러 말을 업을 수 있는지 테스트하였다.
    - 업힌 상태에서 이동이 잘 되는지 확인하였다.
    - 같은 팀일 경우 잘 업히는지 확인하였다.
  - 승리 case
    - 모든 말이 완주하면 승리가 되는지 확인하였다.
- 사각형 판 테스트



















- 오각형 판 테스트





- 육각형 판 테스트

✓  JunitHex [Runner: JUnit 5] (0.220 s)

✓  백도 이동 (0.051 s)

-  백도 6→5→31 (0.051 s)
-  백도 6→5→4 (0.006 s)
-  백도 31→5→4 (0.005 s)
-  백도 11→10→33 (0.007 s)
-  백도 11→10→9 (0.004 s)
-  백도 33→10→9 (0.007 s)
-  백도 16→15→35 (0.003 s)
-  백도 16→15→14 (0.002 s)
-  백도 35→15→14 (0.004 s)
-  백도 21→20→43 (0.005 s)
-  백도 21→20→19 (0.005 s)
-  백도 43→20→19 (0.004 s)
-  백도 26→25 (0.005 s)
-  백도 26→25→41 (0.002 s)
-  백도 40→37→38 (0.004 s)

✓  분기점 이동 (0.005 s)

-  분기 5→31 (0.005 s)
-  분기 5→42 (0.004 s)
-  분기 5→37→38 (0.005 s)
-  분기 4→6 (0.006 s)
-  분기 8→11 (0.003 s)
-  분기 8→10→37→38 (0.005 s)
-  분기 8→10→40 (0.003 s)
-  분기 14→16 (0.004 s)
-  분기 14→15→37→38 (0.004 s)
-  분기 14→15→40 (0.022 s)
-  분기 19→21 (0.003 s)
-  분기 19→20→37→38 (0.004 s)
-  분기 19→20→40 (0.004 s)
-  분기 40→25→26 (0.004 s)
-  분기 24→26 (0.005 s)
-  분기 29→30→Goal (0.009 s)
-  분기 39→30→Goal (0.003 s)

## 5. GitHub 리포트

- github 의 프로젝트 주소
  - Swing

<https://github.com/minjunyeong1/Yut/tree/main/dwdwd/src/yutgame>

- JavaFX

[https://github.com/minjunyeong1/Yut/tree/JavaFX\\_fixed/dwdwd/src/yutgame](https://github.com/minjunyeong1/Yut/tree/JavaFX_fixed/dwdwd/src/yutgame)

- project progress history 스크린샷
  - Swing

Commits on Jun 3, 2025		
클래스패스 추가 minjunyeong1 committed 17 hours ago	d58737a	
<b>JUnit test in HexagonBoard</b> ARanker authored 18 hours ago	Verified 015a849	
<b>JUnit test in PentagonBoard</b> ARanker authored 18 hours ago	Verified 5d93882	
<b>JUnit test in RectangleBoard</b> ARanker authored 19 hours ago	Verified 14ccbff	
Commits on Jun 2, 2025		
게임 종료 메시지 변경 및 필요없는 클래스 제거 minjunyeong1 committed 2 days ago	bd38843	
의미 없는 IF문 제거 minjunyeong1 committed 3 days ago	eaeef28	
Commits on Jun 1, 2025		
완주한 말 개수 표시 minjunyeong1 committed 3 days ago	4fa5a28	
Delete dwdwd/src/yutgame/model/GameController.java minjunyeong1 authored 3 days ago	Verified ff1dbab	
현재 플레이어의 완주한 말의 개수 출력(VIEW 연결 필요) ARanker authored 3 days ago	Verified 8e9b049	
현재 플레이어의 완주한 말의 개수 출력(VIEW 연결 필요) ARanker authored 3 days ago	Verified a640b25	
현재 플레이어의 완주한 말의 개수 출력(VIEW 연결 필요) ARanker authored 3 days ago	Verified 0241e66	

현재 플레이어의 완주한 말의 개수 출력(VIEW 연결 필요)	Verified	0246a53	<>
ARanker authored 3 days ago			
마지막 셀 숫자 오타 수정		d96eaaa	<>
minjunyeong1 committed 3 days ago			
랜덤 옷 백도 버그 + 백도를 수행 할 수 없을 때 튜닝하기 수정		f579861	<>
minjunyeong1 committed 4 days ago			
Commits on May 31, 2025			
옷 모 추가된 버그 개선		de486f5	<>
minjunyeong1 committed 4 days ago			
MVC 분리 승리판정 부분		d3ed84d	<>
minjunyeong1 committed 4 days ago			
1번셀 백도 버그, 말 없을 때 백도 튜닝할 기능, MVC 분리		ffc4f38	<>
minjunyeong1 committed 4 days ago			
Commits on May 30, 2025			
import swing 제거 이에 따른 GameController 수정 필요	Verified	6e239bf	<>
ARanker authored 5 days ago			
import swing 제거 이에 따른 GameController 수정 필요	Verified	6e2b61e	<>
ARanker authored 5 days ago			
import swing 삭제		20747ad	<>
ARanker committed 5 days ago			
골인한 말이 던진 결과를 소모하는 버그를 수정		b0f32f2	<>
ARanker committed 5 days ago			
GameController에서 승리조건판정 MVC 분리		280ebbc	<>
ARanker committed last week			
Commits on May 29, 2025			
추가된 오류 수정(옷이나 모를 던진 것이 아닌 이동하는 경우에도 추가된 발생하는 현상)	Verified	c31572f	<>
ARanker authored last week			
게임 재시작 누락 수정	Verified	0f7b868	<>
ARanker authored last week			
Commits on May 28, 2025			
코드 버그 수정 완료 스왑 최종 버전		01d7e0f	<>
minjunyeong1 committed last week			
버그 수정 완성본		3dd89db	<>
minjunyeong1 committed last week			
Commits on May 23, 2025			
Delete Project Overview.pdf	Verified	965c93a	<>
HaramNangung authored 2 weeks ago			
Add files via upload	Verified	5a06a9e	<>
CHARISHDAIMARI authored 2 weeks ago			
Add files via upload	Verified	0f087f1	<>
CHARISHDAIMARI authored 2 weeks ago			
use case 최종	Verified	21f8eaf	<>
HaramNangung authored 2 weeks ago			
Delete dwdwd/Project Overview.docx	Verified	a3ac29a	<>
HaramNangung authored 2 weeks ago			
project overview 최종본	Verified	ebc8c41	<>
HaramNangung authored 2 weeks ago			
Delete dwdwd/Project Overview.docx	Verified	0007358	<>
CHARISHDAIMARI authored 2 weeks ago			
Commits on May 22, 2025			
project overview 최종	Verified	98eca81	<>
HaramNangung authored 2 weeks ago			
Commits on May 22, 2025			
게임 승리 후 재시작이 안되는 현상 수정		bcffcaf	<>
ARanker committed 2 weeks ago			

Commits on May 22, 2025		
상대 말 잡았을 때 추가 된 수정 ARanker committed 2 weeks ago	1ea8b55	<>
게임 승리 시 재시작/종료 여부 (오타 수정) ARanker authored 2 weeks ago	Verified 72ba678	<>
게임 승리시 재시작/종료 여부 선택 ARanker authored 2 weeks ago	Verified 27213c6	<>
Commits on May 21, 2025		
백드, UI 수정 minjunyeong1 committed 2 weeks ago	79d1427	<>
테스트 안함 거의 완성본 minjunyeong1 committed 2 weeks ago	559dff3	<>
Commits on May 20, 2025		
움직이는 버튼 버그 수정 minjunyeong1 committed 2 weeks ago	288b48c	<>
nextTurn의 책임 controller->model / 추가 된 부여 정리 ARanker authored 2 weeks ago	Verified 7Feb02e	<>
nextTurn의 책임 controller->model / 추가 된 부여 정리 ARanker authored 2 weeks ago	Verified a058505	<>
Add files via upload ARanker authored 2 weeks ago	Verified e350e33	<>
MVC 분리 및 데드리 추가 minjunyeong1 committed 2 weeks ago	6120e4b	<>
Commits on May 18, 2025		
중간 과제 minjunyeong1 committed 3 weeks ago	7897c44	<>
Commits on Apr 30, 2025		
Add Test minjunyeong1 committed on Apr 30	288b5e4	<>
Add files via upload minjunyeong1 authored on Apr 30	Verified 83cc78e	<>
Delete dwdwd/프로젝트 개요.docx minjunyeong1 authored on Apr 30	Verified ae977f5	<>
Delete dwdwd/테스트 리포트.docx minjunyeong1 authored on Apr 30	Verified c774b99	<>
Delete dwdwd/설계 및 구현.docx minjunyeong1 authored on Apr 30	Verified ac4ab66	<>
Add files via upload minjunyeong1 authored on Apr 30	Verified 43b9ed0	<>
Delete dwdwd/docx minjunyeong1 authored on Apr 30	Verified 90e9576	<>
Create docx minjunyeong1 authored on Apr 30	Verified e826213	<>
Commits on Apr 4, 2025		
rename minju committed on Apr 4	5d79d71	<>
swing_ui code minju committed on Apr 4	461c545	<>
pushtest minju committed on Apr 4	3e6ad89	<>
Commits on Mar 9, 2025		
javafx test minju committed on Mar 9	abb8b81	<>
Commits on Mar 8, 2025		
test minju committed on Mar 8	16c6544	<>
Initial commit minjunyeong1 authored on Mar 8	Verified 69db2b4	<>

## ○ FX

### Commits on Jun 2, 2025

승리 메시지 수정, 필요없는 클래스 삭제 minjunyeong1 committed 2 days ago	7b7d185		<>
의미 없는 If문 제거 minjunyeong1 committed 3 days ago	028e49d		<>

### Commits on Jun 1, 2025

나간 말 표시 minjunyeong1 committed 3 days ago	5a305fa		<>
마지막 셀 숫자 오타 minjunyeong1 committed 3 days ago	76fc587		<>
랜덤 옷 던지기 움직일 수 있는 말 없으면 백도 턴 넘기기 minjunyeong1 committed 4 days ago	6f49735		<>

### Commits on May 31, 2025

UI 개선 minjunyeong1 committed 4 days ago	5f1390a		<>
옷 모 추가된 버그 수정 minjunyeong1 committed 4 days ago	7a27540		<>
이전 커밋 메시지 추가용 JavaFX에서 수정한 스윙 부분의 컨트롤러 병합 완료 minjunyeong1 committed 4 days ago	d09a308		<>
이전 JavaFX 브랜치 오류로 새로 생성 minjunyeong1 committed 4 days ago	9a8cf04		<>
MVC 분리 승리판정 부분 minjunyeong1 committed 4 days ago	d3ed84d		<>
1번셀 백도 버그, 말 없을 때 백도 턴넘김 기능, MVC 분리 minjunyeong1 committed 4 days ago	ffc4f38		<>

### Commits on May 30, 2025

import swing 제거 이에 따른 GameController 수정 필요 ARanker authored 5 days ago	Verified	6e239bf		<>
import swing 제거 이에 따른 GameController 수정 필요 ARanker authored 5 days ago	Verified	6e2b61e		<>
import swing 삭제 ARanker committed 5 days ago		20747ad		<>
공인한 말이 던진 결과를 소모하는 버그를 수정 ARanker committed last week		b0f32f2		<>
GameController에서 승리조건판정 MVC 분리 ARanker committed last week		28bebbc		<>

### Commits on May 29, 2025

추가된 오류 수정(옷이나 모를 던진 것이 아닌 이동하는 경우에도 추가턴이 발생하는 현상) ARanker authored last week	Verified	c31572f		<>
게임 재시작 누락 수정 ARanker authored last week	Verified	0f7b868		<>

### Commits on May 28, 2025

코드 버그 수정 완료 스윙 최종 버전 minjunyeong1 committed last week	01d7e0f		<>
버그 수정 완성본 minjunyeong1 committed last week	3dd89db		<>

Commits on May 23, 2025

<b>Delete Project Overview.pdf</b> HaramNamgung authored 2 weeks ago	Verified	965c93a		
<b>Add files via upload</b> CHARISHDAIMARI authored 2 weeks ago	Verified	5a96a9e		
<b>Add files via upload</b> CHARISHDAIMARI authored 2 weeks ago	Verified	0f807f1		
<b>use case 최종</b> HaramNamgung authored 2 weeks ago	Verified	21f8eaf		
<b>Delete dwdwd/Project Overview.docx</b> HaramNamgung authored 2 weeks ago	Verified	a3ac29a		
<b>project overview 최종본</b> HaramNamgung authored 2 weeks ago	Verified	ebc8c41		
<b>Delete dwdwd/Project Overview.docx</b> CHARISHDAIMARI authored 2 weeks ago	Verified	0007358		
<b>project overview 최종</b> HaramNamgung authored 2 weeks ago	Verified	98eca81		

Commits on May 22, 2025

게임 승리 후 재시작이 안되는 현상 수정 ARanker committed 2 weeks ago	bcaffcaf		
상대 말 잡았을 때 추가 턴 수정 ARanker committed 2 weeks ago	1ea0b55		
게임 승리 시 재시작/종료 여부 (오타 수정) ARanker authored 2 weeks ago	Verified 72ba670		
게임 승리시 재시작/종료 여부 선택 ARanker authored 2 weeks ago	Verified 27213c6		

Commits on May 21, 2025

백도, UI 수정 minjunyeong1 committed 2 weeks ago	79d1427		
테스트 안함 거의 완성본 minjunyeong1 committed 2 weeks ago	559dff3		

Commits on May 20, 2025

움직이는 버튼 버그 수정 minjunyeong1 committed 2 weeks ago	288b48c		
---	---------	--	--

Commits on May 20, 2025

<b>nextTurn의 책임 controller-&gt;model / 추가 턴 부여 정리</b> ARanker authored 2 weeks ago	Verified 7feb02e		
<b>nextTurn의 책임 controller-&gt;model / 추가 턴 부여 정리</b> ARanker authored 2 weeks ago	Verified a058505		
<b>Add files via upload</b> ARanker authored 2 weeks ago	Verified e350e33		
<b>MVC 분리 및 데두리 추가</b> minjunyeong1 committed 2 weeks ago	6120e4b		

















Commits on May 18, 2025

중간 과제 minjunyeong1 committed 3 weeks ago	7897c44		
---	---------	--	--

Commits on Apr 30, 2025

<b>Add Test</b> minjunyeong1 committed on Apr 30	280b5e4		
<b>Add files via upload</b> minjunyeong1 authored on Apr 30	Verified 83cc78e		
<b>Delete dwdwd/프로젝트 개요.docx</b> minjunyeong1 authored on Apr 30	Verified ae977f5		
<b>Delete dwdwd/테스트 리포트.docx</b> minjunyeong1 authored on Apr 30	Verified c774b99		
<b>Delete dwdwd/설계 및 구현.docx</b> minjunyeong1 authored on Apr 30	Verified ac4ab66		
<b>Add files via upload</b> minjunyeong1 authored on Apr 30	Verified 43b9ed0		



Delete dwdwd/docx  minjunyeong1 authored on Apr 30	<span>Verified</span> 90e9576  <>
Create docx  minjunyeong1 authored on Apr 30	<span>Verified</span> e826213  <>
Commits on Apr 4, 2025	
rename  minju committed on Apr 4	5d79d71  <>
swing_ui code  minju committed on Apr 4	461c545  <>
pushtest  minju committed on Apr 4	3e6ad89  <>
Commits on Mar 9, 2025	
javafx test  minju committed on Mar 9	abb8b81  <>
Commits on Mar 8, 2025	
test  minju committed on Mar 8	16c6544  <>
Initial commit  minjunyeong1 authored on Mar 8	<span>Verified</span> 69db2b4  <>

- 팀원별 기여도

- 민준영

- 프로젝트 실행 환경 설정 체크
    - MVC 패턴 기반 세부 기능 구현 (오각형/육각형 판 - Swing 버전 View 및 Controller 영역)
    - Swing 에서 JavaFX 로의 전환 작업 담당
    - 전환 과정에서 발생한 일부 오류 수정
    - View, Controller 전반적인 코드 작성
    - Model 일부 구현
    - 버그 수정 및 코드 안정화
    - 뺑도 특수 케이스 구현
    - GameController 클래스 설계 논의 및 구현

- 이인표

- Model 구현
    - Controller 일부 구현
    - 주요 버그 수정 담당
    - Junit case 분류 및 코드 작성
    - GameController 클래스 설계 논의 구현

- 남궁하람

- 전체 디자인 및 구현 설계
    - 전체 보고서 작성
    - 프로젝트 개요 작성
    - 유스 케이스 텍스트 작성 및 다이어그램 제작
    - 설계 및 구현 전략 문서 작성

- 클래스 다이어그램 제작
- 영상 제작
- 테스트 리포트 작성
- Junit case 분류
- PPT 제작
- 발표
- 다이마리 케리스
  - 전체 디자인 및 구현 설계
  - 전체 보고서 작성
  - 프로젝트 개요 작성
  - 유스 케이스 텍스트 작성 및 다이어그램 제작
  - 설계 및 구현 전략 문서 작성
    - 클래스 다이어그램 제작
  - 영상 제작
  - 테스트 리포트 작성
  - PPT 제작
  - 발표
- 이희재
  - MVC 패키지 구조 및 클래스 구분 후 생성
  - 팝업창, UI 수정
  - Swing -> JavaFX 전환 중 발생한 오류 수정
  - GameControlle 클래스 설계 논의 및 구현
- 공동작업
  - Git history 분석 및 정리
  - UML 설계부터 클래스 구현까지의 전체적인 협업
  - 프로젝트 시나리오 기획