

<휴먼 ICT 소프트웨어공학>

# 웃놀이

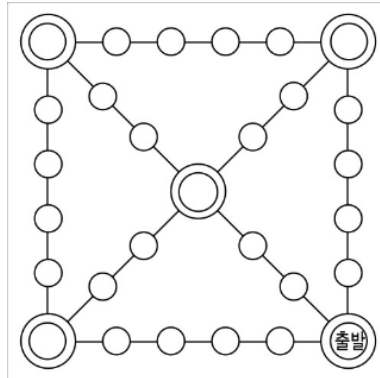
<https://github.com/kawoou/cau-se19-yutnori>

20131438 안정원

2019.06.07

## 1. 개요

- 윷놀이 게임을 OOAD를 적용하여 소프트웨어로 개발한다.
- 게임 시작 시 참여자의 명수(최소 2명, 최대 4명)와 게임 말 갯수(최소 2개, 최대 5개)를 지정할 수 있다.
- 표준 윷놀이 판은 다음 그림과 같게 하며, 각 참여자의 말의 현재 위치가 표시되어야 한다.



- 각 턴의 진행을 위해 <랜덤 윷 던지기> 버튼과 <지정 윷 던지기> 버튼이 표시된다.
  - <지정 윷 던지기> 버튼을 누르면 뺑도, 도, 개, 걸, 윷, 모 중 선택한 결과로 진행된다. (테스트용)
  - <랜덤 윷 던지기> 버튼을 누르면 랜덤한 윷의 결과가 나타난다.
- 사용자는 윷 던지기 결과를 적용할 게임 말을 선택할 수 있으며, 그에 따라 진행이 자동으로 되어야 한다.
- 게임 말을 엮는(grouping) 기능을 지원해야 한다.
- 다른 사용자의 말을 잡는 기능을 지원해야 한다.
- 게임 말들이 출발에서 시작해서 먼저 모든 말을 내보내는 팀이 게임에 승리하며, 이때 어느 팀이 승리했는지 표시한다.
- 한 게임이 끝났을 때 게임 재시작 혹은 종료가 가능해야 한다.

### 주요 사항:

- 본 프로젝트를 수행하면서 수업 시간에 다룬 OOAD 기법을 적극적으로 사용하며, 그 결과를 문서화해야 한다.
- MVC 아키텍처 패턴을 사용하여 UI와 모델을 분리하여 구현해야 한다.
  - MVVM 패턴을 사용해도 무방하다는 답변을 받았다.
- Extra point: 테스트 용이한 설계를 하여 JUnit으로 모델 테스트를 수행한다.

## 2. 프로젝트 소개

### 2.1. Object-Oriented Analysis and Design

먼저 OOAD 방법론을 사용하여 구현의 세부 내용을 구상하기 이전에 프로젝트의 전체적인 Feature 들을 UseCase Diagram 으로 작성했습니다.

그리고 구현 시에는 SOLID 원칙의 SRP(Single Responsibility Principle)를 지켜 하나의 클래스가 하나의 책임만을 수행하는데 집중하도록 하였습니다.

### 2.2. Domain Driven Design

도메인 내의 여러 업무 정의나 문제를 어떻게 모델로 잘 표현하고, 그것을 바탕으로 구현하기 쉽게 이해할 수 있는 구조로 정의되느냐가 DDD 의 핵심입니다. 이를 바탕으로 게임 내의 정보에 접근하는데 Entity 기반의 모델을 정의하는 것으로 시작했습니다.

또한 Layered Architecture 에 따라 클래스를 Entity 모델은 Data 레이어, 비즈니스 로직은 Service 레이어, Aggregate 에 접근하고 처리하는 Repository 레이어, 그리고 UI 레이어로 나누어 클래스들을 구성했습니다.

### 2.3. Micro-Service Architecture

DDD 를 설명할 때 항상 같이 나오는 MSA 는 각 Component 를 Service 라는 개념으로 정의합니다. 서비스는 Data 에서부터 비즈니스 로직까지 독립적으로 상호 컴포넌트간의 의존성이 없이 개발된 Component 로 그 기능을 외부로 제공합니다.

Service 레이어는 최대한 Stateless, Loosely Coupling, High Cohesion 한 서비스를 만들도록 했습니다.

## 2.4. Event-Driven Development

프로그램의 흐름이 Event 에 의해 결정되는 결정되는 Event-based 프로그래밍 방법론 중의 하나입니다. Stream 기반으로 동작하므로 Multi-thread 환경에서 성능상 이점을 구할 수 있으며 Immutable Type 을 사용하여 데이터를 전달하므로 Side-effect 를 최소화할 수 있습니다.

이를 위해 모든 클래스 간에 이벤트 전달은 RxSwift(ReactiveX)를 사용하여 Stream 을 구성했습니다.

## 2.5. Test Driven Development

윗놀이 프로젝트는 기획의 변동이 없이 고정되어 있으므로, 앞서 설명한 방법론과 Architecture 를 바탕으로 Interface 를 정의하고 테스트 코드를 우선 작성하여 Red-Green 방식의 개발을 진행했습니다.

테스트를 도와주는 프레임워크로는 Quick, Nimble 를 사용했습니다. 테스트 코드를 이해하기 쉽게 도와주고, 실패에 대한 메시지를 보기 쉽게 제공하기에 선택했습니다.

## 2.6. Dependency Injection

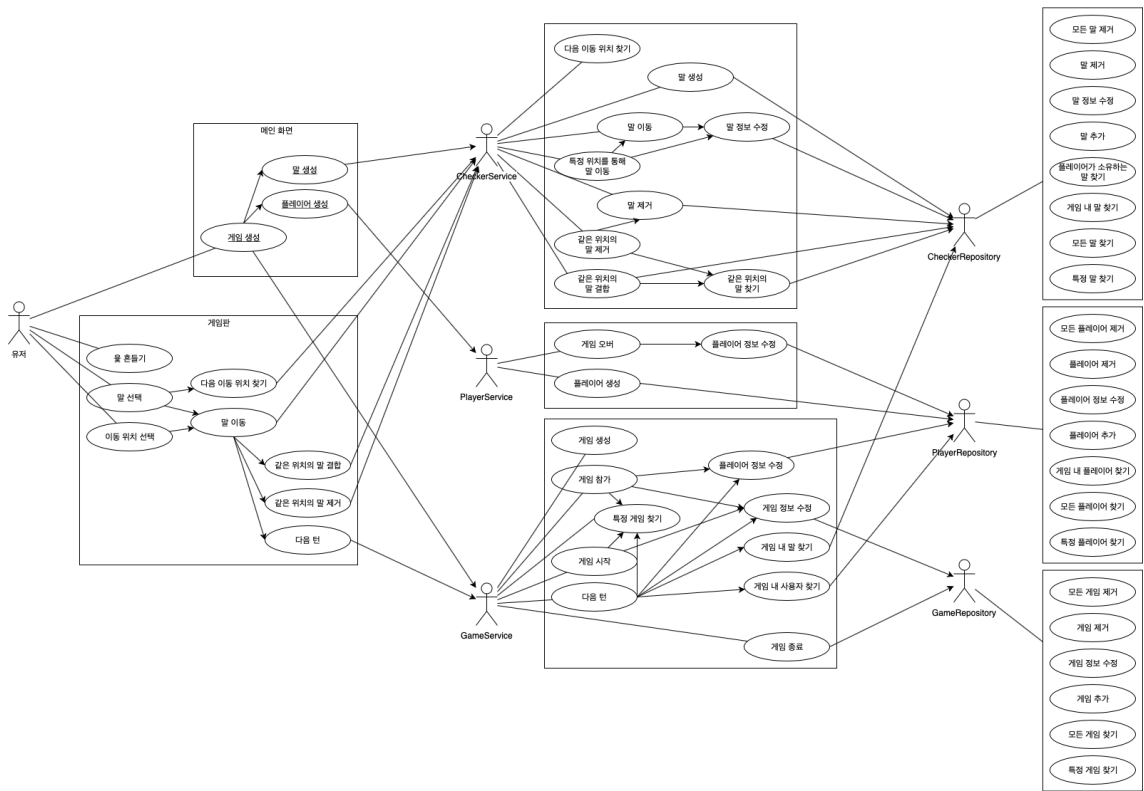
SOLID 원칙의 DIP(Dependency Inversion Principle)에 해당하는 내용을 해결하기 위해 사용한 방법으로 의존성을 분리하고 추상에 의존하도록 Service, Repository 레이어를 Interface 를 정의하고 생성자를 통해 주입하도록 설계하였습니다.

Swift 에서 DI 를 도와주는 프레임워크로 위해 Swinject, Deli 와 같은 라이브러리가 있으나 프로젝트 규모가 크지 않으므로 Dependency 클래스를 분리하는 것으로 해결하였습니다.

## 2.7. MVVM Architecture

MVC 와 MVP 의 단점은 데이터가 양방향으로 흐르게 된다는 것입니다. 그러므로 단방향으로 데이터가 흐르는 MVVM Architecture 의 변형인 Event-Driven 한 RxMVVM 를 사용했습니다.

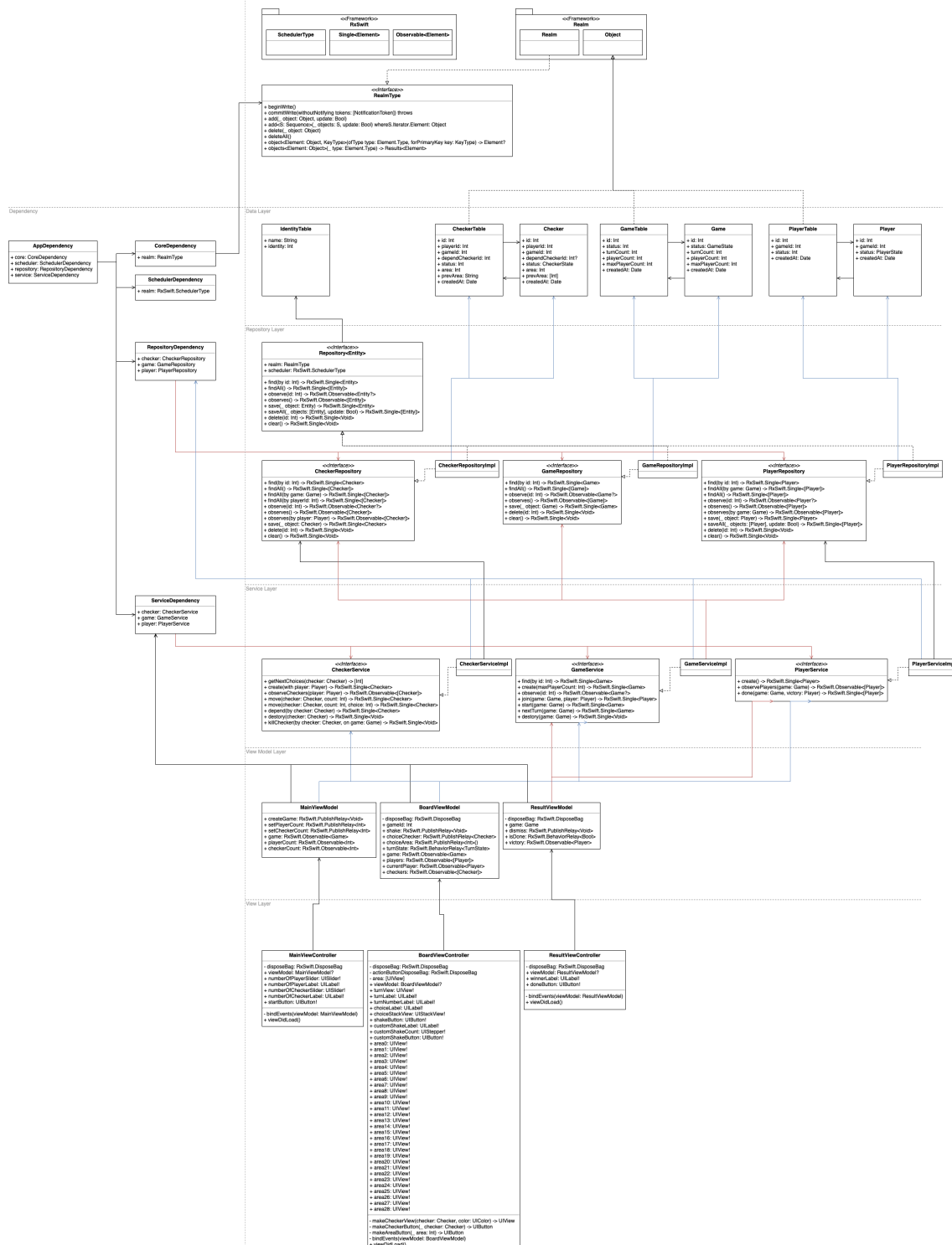
3. UseCase Diagram



<Figure 1. UseCase Diagram>

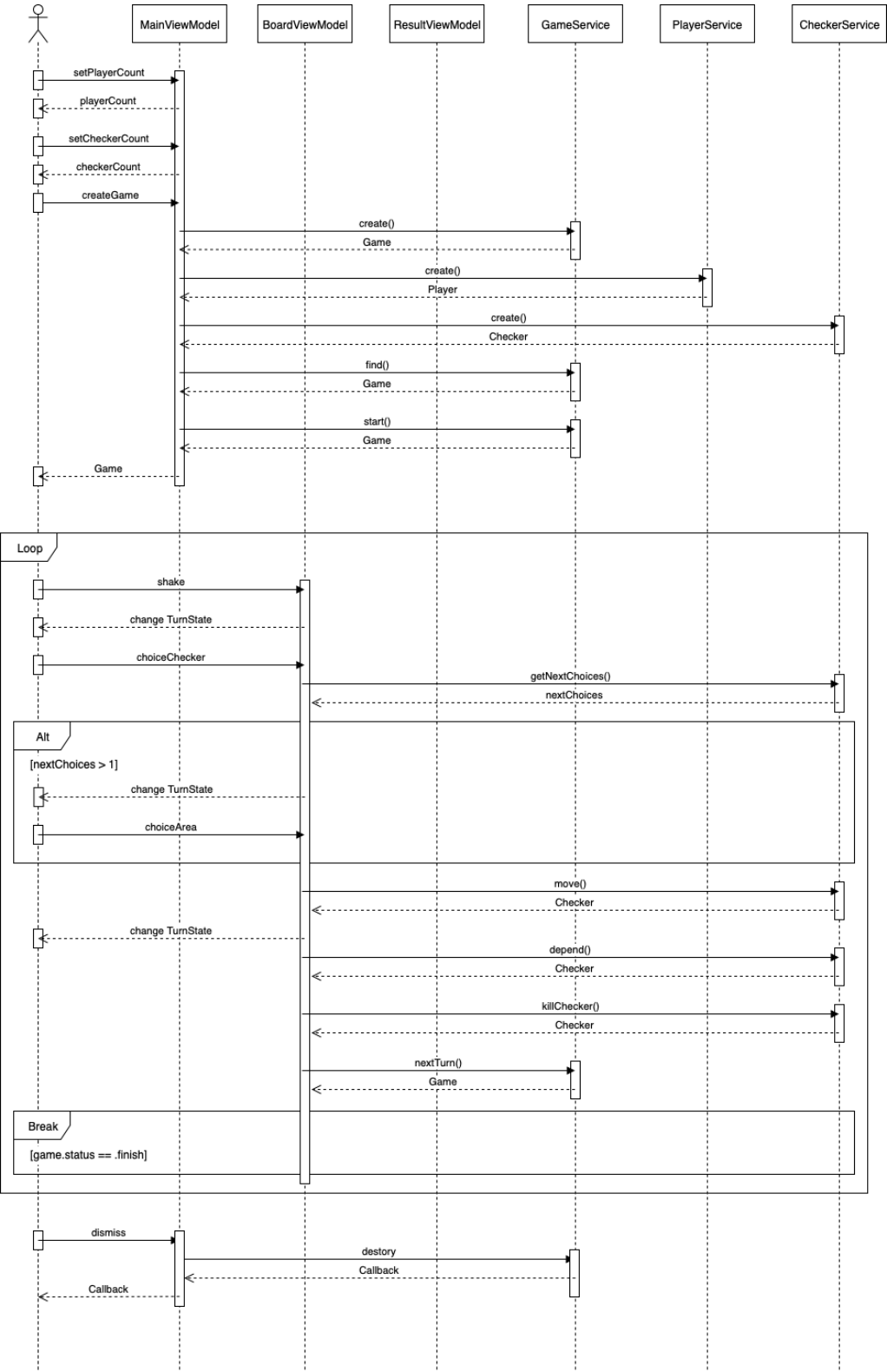
DDD 에 따라 Service, Repository 를 나누어 각각의 책임자를 둔 모습입니다.

## 4. Class Diagram



<Figure 2. Class Diagram>

## 5. Sequence Diagram



<Figure 3. 사용자 입장에서의 Sequence Diagram>



## 6. Test Documetation

대상	절차	결과
에러메시지	CommonError 의 error code 가 정상적인가?	OK
	RepositoryError 의 error code 가 정상적인가?	OK
	CheckerServiceError 의 error code 가 정상적인가?	OK
	GameServiceError 의 error code 가 정상적인가?	OK
Repository	존재하지 않는 오브젝트를 save() 했을 때, 결과의 id 는 새롭게 발급되었는가?	OK
	존재하지 않는 오브젝트를 save() 했을 때, Entity 가 Realm 에 정상적으로 저장되었는가?	OK
	존재하지 않는 오브젝트를 save() 한 후 존재하는 오브젝트를 save() 했을 때, 변경된 Entity 의 id 가 변경되지 않았는가?	OK
	존재하지 않는 오브젝트를 save() 한 후 존재하는 오브젝트를 save() 했을 때, Entity 가 Realm 에 정상적으로 저장되었는가?	OK
	존재하지 않는 오브젝트를 save() 한 후 존재하는 오브젝트를 delete() 했을 때, Entity 를 Realm 에서 발견할 수 없는가?	OK
	존재하지 않는 오브젝트를 save() 한 후 존재하지 않는 오브젝트를 delete() 했을 때, 에러가 발생하는가?	OK
	존재하지 않는 오브젝트를 save() 한 후 clear() 했을 때, 어떠한 Entity 도 Realm 에서 찾을 수 없는가?	OK
	존재하지 않는 오브젝트를 save() 한 후 존재하는 오브젝트를 find(by:) 했을 때, 결과가 nil 이 아닌가?	OK
	존재하지 않는 오브젝트를 save() 한 후 findAll() 했을 때, 결과가 1 개인가?	OK
	존재하지 않는 오브젝트를 save() 한 후 존재하는 오브젝트를 observe(by:) 하고 값을 변경했을 때, 결과의 변경 내역이 정상적인가?	OK
	존재하지 않는 오브젝트를 save() 한 후 존재하는 오브젝트를 observe(by:) 하고 값을 지웠을 때, 결과의 변경 내역에 nil 이 정상적으로 포함되는가?	OK
	존재하지 않는 오브젝트를 save() 한 후 observeAll() 하고 값을 추가했을 때, 결과의 개수가 늘어나는게 기록되었는가?	OK
	존재하지 않는 오브젝트를 save() 한 후 observeAll() 하고 값을 지웠을 때, 결과의 개수가 줄어드는게 기록되었는가?	OK

	오브젝트를 추가하지 않고 find(by:) 했을 때, 결과가 nil 인가?	OK
	오브젝트를 추가하지 않고 findAll() 했을 때, 결과의 수가 0 개인가?	OK
CheckerRepository	존재하지 않는 오브젝트를 save() 한 후 현재 게임에 존재하는 말을 찾을 때, 값이 1 개 인가?	OK
	존재하지 않는 오브젝트를 save() 한 후 특정 플레이어가 소유하는 말을 찾을 때, 값이 1 개인가?	OK
	존재하지 않는 오브젝트를 save() 한 후 특정 플레이어가 소유하는 말을 observe 받고 말을 생성할 때, 말의 개수가 1 개에서 2 개로 늘어났는가?	OK
	존재하지 않는 오브젝트를 save() 한 후 특정 플레이어가 소유하는 말을 observe 받고 말을 지웠을 때, 말의 개수가 1 개에서 0 개로 줄어들었는가?	OK
PlayerRepository	존재하지 않는 오브젝트를 save() 한 후 현재 게임에 참가한 플레이어를 찾을 때, 값이 1 개인가?	OK
	존재하지 않는 오브젝트를 save() 한 후 현재 게임에 참가한 플레이어를 observe 받고 플레이어를 생성할 때, 플레이어의 개수가 1 개에서 2 개로 바뀌었는가?	OK
	존재하지 않는 오브젝트를 save() 한 후 현재 게임에 참가한 플레이어를 observe 받고 플레이어를 지웠을 때, 플레이어의 개수가 1 개에서 0 개로 바뀌었는가?	OK
CheckerService	바로 다음 칸이 1 개 뿐이 위치에서 getNextChoices()를 불렀을 때, 다음 이동 지역이 정상적으로 나오는가?	OK
	선택이 필요한 위치에서 getNextChoices()를 불렀을 때, 다음 이동 지역이 정상적으로 나오는가?	OK
	이전 위치에 따라 달라지는 위치에서 getNextChoices()를 불렀을 때, 다음 이동 지역이 정상적으로 나오는가?	OK
	완료된 상태의 말을 getNextChoices()를 불렀을 때, 다음 이동 지역이 안나오는가?	OK
	다른 말에 얹혀있는 말을 getNextChoices()를 불렀을 때, 다음 이동 지역이 안나오는가?	OK
	말을 생성했을 때, CheckerRepository 의 생성 메소드가 정상적으로 불리는가?	OK
	말을 observe 받을 때, CheckerRepository 의 observe 메소드가 정상적으로 불리는가?	OK
	말이 뒤로 이동할 때, 새로운 말의 위치가 정상적으로 계산되는가?	OK

	말이 뒤로 이동할 때, CheckerRepository 의 저장 메소드가 정상적으로 불리는가?	OK
	만약 0 번 째 위치에서 말이 뒤로 이동할 때, 새로운 말의 위치는 0 이고 상태는 idle 이 되는가?	OK
	말이 앞으로 이동할 때, 새로운 말의 위치가 정상적으로 계산되는가?	OK
	말이 앞으로 이동할 때, CheckerRepository 의 저장 메소드가 정상적으로 불리는가?	OK
	만약 마지막 위치의 말이 앞으로 이동할 때, 새로운 말의 위치는 0 이고 상태는 done 이 되는가?	OK
	말의 이동 위치를 선택할 때, 새로운 말의 위치는 선택된 방향으로 이동하는가?	OK
	말의 이동 위치를 선택할 때, CheckerRepository 의 저장 메소드가 정상적으로 불리는가?	OK
	만약 마지막 위치의 말의 이동 위치를 선택할 때, 새로운 말의 위치는 0 이고 상태는 done 이 되는가?	OK
	말을 업을 때, CheckerRepository 의 저장 메소드가 정상적으로 불리는가?	OK
	말을 지울 때, CheckerRepository 의 삭제 메소드가 정상적으로 불리는가?	OK
	게임 내의 같은 위치의 말을 제거할 때, CheckerRepository 의 저장 메소드가 정상적으로 불리는가?	OK
GameService	특정 게임을 가져올 때, GameRepository 의 찾기 메소드가 정상적으로 불리는가?	OK
	게임을 생성할 때, 지정한 최대 플레이어 수가 Game Entity 에 적용되어있는가?	OK
	게임을 생성할 때, GameRepository 의 저장 메소드가 정상적으로 불리는가?	OK
	특정 게임을 observe 받을 때, GameRepository 의 observe 메소드가 정상적으로 불리는가?	OK
	게임에 플레이어가 참가하는데 성공했을 때, GameRepository 의 저장 메소드가 정상적으로 불리는가?	OK
	게임에 플레이어가 참가하는데 성공했을 때, Player 의 gameId 는 게임의 id 와 같은가?	OK
	게임에 플레이어가 참가하는데 실패했을 때, GameRepository 의 저장 메소드가 정상적으로 불리는가?	OK
	게임에 플레이어가 참가하는데 실패했을 때, 결과가 nil 이 맞	OK

	는가?	
	게임을 시작했을 때, 게임의 상태가 starting 으로 변경되는가?	OK
	게임을 시작했을 때, 게임의 턴 수가 0 이 맞는가?	OK
	게임을 시작했을 때, GameRepository 의 저장 메소드가 정상적으로 불렀는가?	OK
	게임의 턴을 다음으로 넘기면 게임 오버일 때, PlayerRepository 의 저장 메소드가 정상적으로 불렀는가?	OK
	게임의 턴을 다음으로 넘기면 게임 오버일 때, GameRepository 의 저장 메소드가 정상적으로 불렀는가?	OK
	게임의 턴을 다음으로 넘기면 게임 오버일 때, 결과의 상태는 finished 가 되는가?	OK
	게임의 턴을 다음으로 넘겼을 때, PlayerRepository 의 저장 메소드가 불리지 않는가?	OK
	게임의 턴을 다음으로 넘겼을 때, GameRepository 의 저장 메소드가 불리는가?	OK
	게임의 턴을 다음으로 넘겼을 때, 턴 수가 오르는가?	OK
	게임을 종료했을 때, GameRepository 의 삭제 메소드가 정상적으로 불리는가?	OK
PlayerService	플레이어를 생성했을 때, PlayerRepository 의 생성 메소드가 정상적으로 불리는가?	OK
	플레이어를 observe 받을 때, PlayerRepository 의 observe 메소드가 정상적으로 불리는가?	OK
	특정 플레이어가 승리하여 게임이 종료될 때, 결과의 플레이어들 중 승리자는 1 명인가?	OK