SE 4회차

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

1. Software Architecture

1.1 Architecture

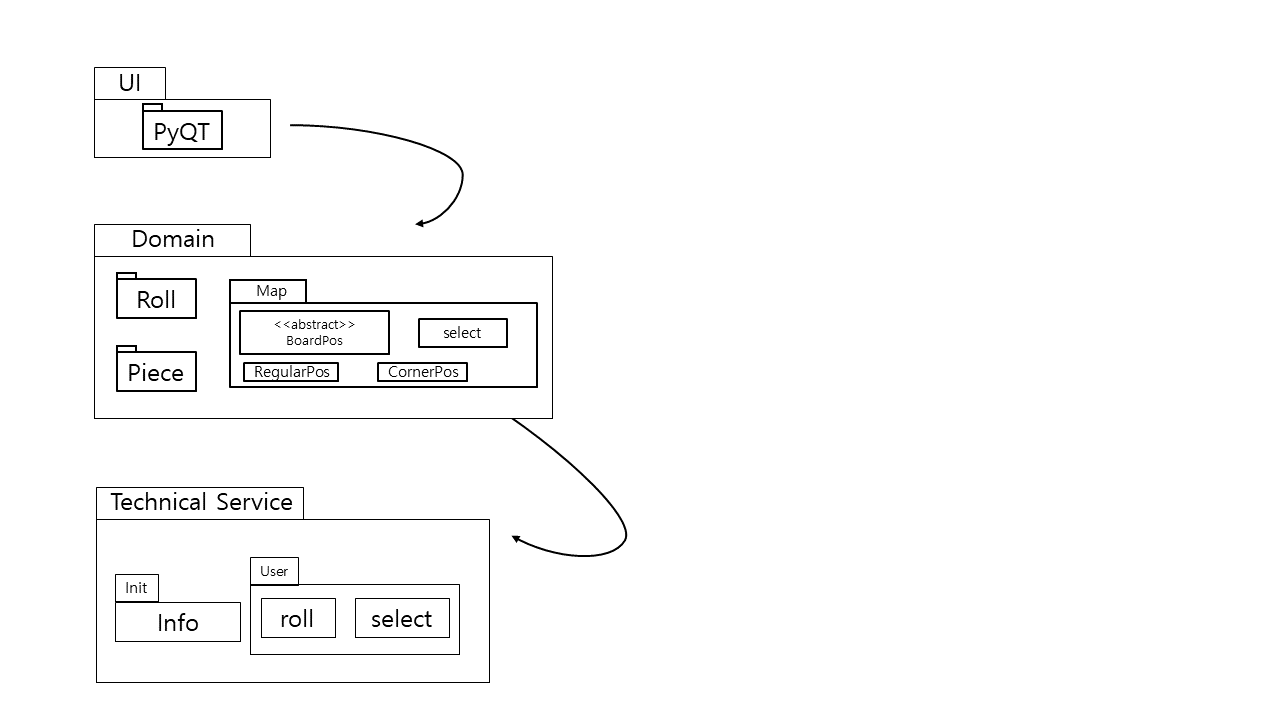
1.1.1 **The logical architecture**, which describes the system in terms of its conceptual organization

in layers, packages, major frameworks, classes, interfaces, and subsystems.

1.1.2 **The deployment architecture**, which describes the system in terms of the allocation of

processes to processing units, and the network configuration. // 얘는 수업 때 안다룸!

1.2 Layered Architecture



2. Design Model

2.1 Yut Nori MVC

2.1.1 Model

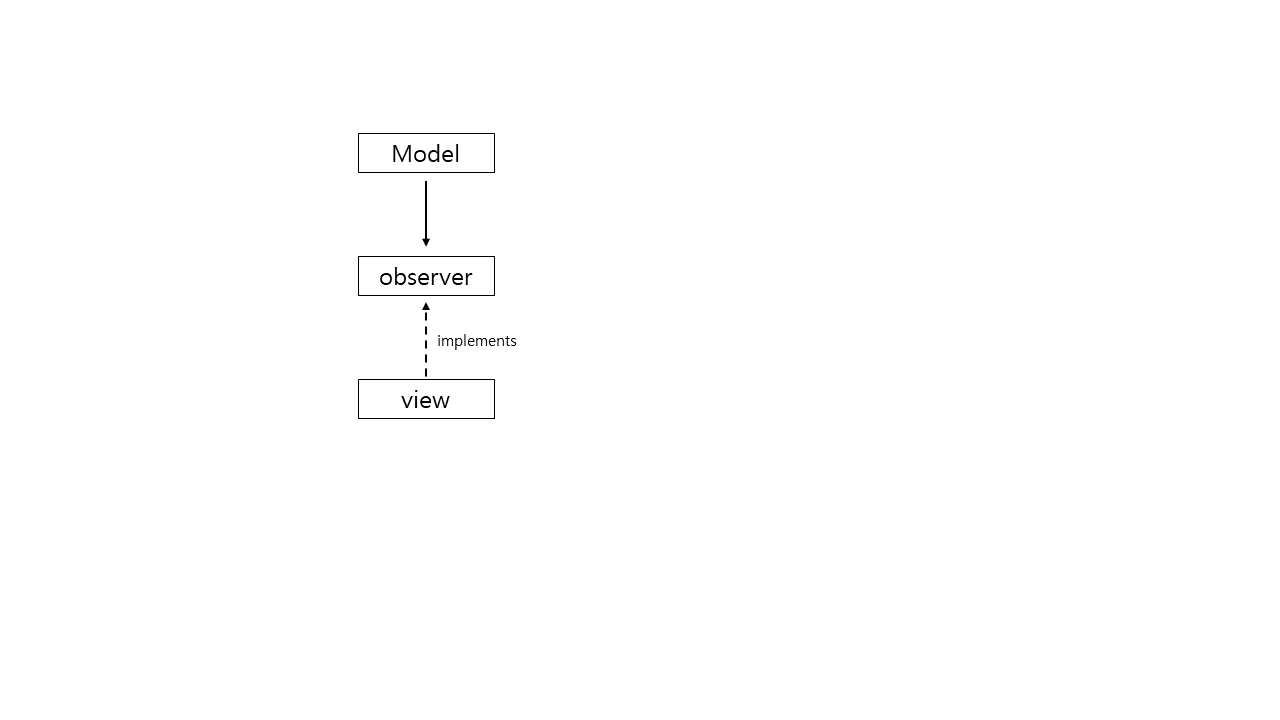
\* Yut Nori는 한국의 전통 게임 윷놀이를 플레이 할 수 있게 해준다.

\* Model은 전반적인 게임을 컨트롤 한다. (ex : rules, player info 등)

\* Model은 화면 크기를 알고 있어야한다. (view에 전달하기 위해)

\* Model은 GUI에 관련된 내용은 전혀 알지 못해야 한다. (Python- PyQT / JAVA- Swing)

\* Model은 Observable.



2.1.2 View

\* 현재 턴인 플레이어의 정보에 접근할 수 있어야 한다.

\* 윷놀이 결과(말의 움직임, 윷의 결과 등)를 화면에 나타내야 한다.

\* View는 Observer.

\* View는 Model에서 일어나는 상황을 보여준다.

2.1.3 Controller

\* Controller는 Model이 앞으로 무엇을 해야 하는지 알려준다.

\* Controller는 View가 화면을 refresh해야할 때를 알려준다.

\* Controller는 Model의 세부 동작 사항을 알지 못한다.

\* Controller는 View의 세부 동작 사항을 알지 못한다.

3. Operation Contract

|  |  |
| --- | --- |
| Operation | roll |
| Cross Reference | 윷 던지는 UseCase(UC6) |
| Preconditons | 1. 플레이어의 정보가 입력되고 게임이 시작된 상황이어야한다. |
| Postconditions | ∙ 플레이어가 던진 윷의 결과가 기록된다.  ∙ 플레이어는 윷 4개를 던졌다.  ∙ 플레이어가 던진 윷의 결과에 따라 말이 이동한다.  (도 : 1칸, 개 : 2칸, 걸 : 3칸, 윷 : 4칸, 모 : 5칸, 빽도 : 뒤로 1칸)  ∙ 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. |

|  |  |
| --- | --- |
| Operation | Info |
| Cross Reference | **게임 시작 후 플레이어의 정보를 입력하는 Use-case(UC1)** |
| Preconditons | 1. 게임 스타트 버튼이 눌린 상태여야한다. |
| Postconditions | ∙ 플레이어의 이름이 기록되었다.  ∙ 플레이어 명수가 기록되었다.  ∙ 플레이 할 말의 개수가 정해졌다.  ∙ 플레이어 순서가 정해졌다. |

|  |  |
| --- | --- |
| Operation | Move |
| Cross Reference | 보드판에서의 말의 움직임을 다루는 Use-Case(UC2, UC3) |
| Preconditons | 1. 윷이 던져진 이후여야 한다.  2. 말이 선택된 이후여야 한다. |
| Postconditions | ∙ 교차점이 아닌 곳에서는 말은 일직선으로 움직였다.  ∙ 교차점에 말이 걸려있다면, 말은 사선방향으로 움직였다. |

|  |  |
| --- | --- |
| Operation | Select( ) |
| Cross Reference | 보드판에서의 말을 추가할지, 기존의 말을 움직일지를 다루는 Use-Case(UC2) |
| Preconditons | 1. 윷이 던져진 이후여야 한다. |
| Postconditions | ∙ 플레이어가 새로운 말을 추가하기를 눌렀다면 시작 위치에 새로운 말이 하나 추가되고, 던진 윷의 결과는 새로운 말에 적용되었다.  ∙ 플레이어가 기존 말을 클릭하면 윷의 결과에 따라 보드 판 위에 이미 존재하는 선택된 말이 이동하였다. |