

0404 BE 설계 회의

주제

- 좌표 방식 상의 - 시운 더 고민
- 도메인 구성
- ~~View~~랑 데이터 상호작용 방식 결정
- 백도 처리
 - 백도 선택 후 → 출발 안 될 말은 선택 가능한 말에 포함 안 되게
- 말 업는 거 처리 → <Piece> 자체를 옮기는 걸로
- 플레이어 플레이 순서 - Player에 id 넣어서 순서 부여할지, Game에 리스트로 넣기만 할지

Class 설계

클래스 이름	간략 설명
Game	게임 전체의 상태를 저장하고 관리하는 중심 클래스. → 이걸 통째로 View에 넘겨서 ViewBinding
Player	개별 사용자. 말 상태 저장 및 윷 던지기 수행.
Piece	게임 말 하나를 나타냄. 현재 위치 정보 등을 포함. 업기, 잡기 등 관련 로직에 사용됨.
Board	전체 게임판을 관리. 각 위치(Cell)에 있는 말들을 추적하며, 위치별 말 겹침/이동 등을 담당.
Cell	보드의 지점을 표현. 고유 id와 다음으로 이동 가능한 Cell id 목록을 포함해 path 예상에 사용
YutResult or Yut	윷 던지기 결과를 표현. 도, 개, 걸, 윷, 모, 백도 등과 추가 턴 여부를 담는다.

- ~~Game~~
 - ~~<Player> players~~; 플레이어 정보
 - ~~Board~~; 판 정보
 - ~~Yut~~; 윷 정보
- Player

- id ; 순서용 → 리스트에 순서 있으니까 상관 없나? 그래도 명시하는 게 좋나?
- Piece ; 보유 말 정보
- Piece
 - position ; 위치 정보
 - Player owner ; 소속 정보
- Board
 - <Cell> cells ; 지점 정보
- Cell
 - id ; 고유 아이디, 인덱스
 - <Cell> nextCell ; 경로 탐색을 위해서 이동 가능한 지점의 id 저장 → Cell? Int?
⇒ Linked-list 방식으로 최종 결정. Board가 일종의 'CellList'
 - type ; 유형 - enum(시작, 종료, 분기점)...그냥 id로 하는 것보다는 가독성이 좋을 것 같긴 한데
 - <Piece> withPieces ; 업은 말들
- Yut
 - resultList ; 결과 저장 - enum(뽕도, 도, 개, 걸, 윷, 모)
 - ~~remainingThrows ; 남은 던지기 횟수?~~

→ 말이 업었다 치면 개들을 옮겨야 될 텐데 그건 어떻게 인식할지

To do

- 좌표값 방식&지름길 사용 방식 결정