|  |  |
| --- | --- |
| 4. UIDrawer | 오른쪽 부분, 버튼터치를 관리한다 |
| 5. MapDrawer | 어떤 타일을 선택했는지 알 수 있다 |
| -. GameView | 게임 뷰 |
| 7. Builder | 건물들을 빌드하는데 필요한 SQL 처리를 한다 |
| 8. WorldMap | 월드 맵에 대한 SQL 처리를 한다.  전쟁, 다른 타일로 진영 늘리기 |
| 9. Building | View와 Function에 대한 처리를 한다 |
| 6. AppController | 핵심 개체  게임 내에서 타일과 UI에 대한 버튼 처리의 모든 반응을 처리한다 |
| 10. MainMenu | 메인 메뉴를 처리한다. AppController랑 동급이다 |
| 1. ViewData | UI 뷰에 들어가게 될 아이콘 파일 이름과 숫자값이 들어간다 |
| 2. ButtonData | 버튼에 들어갈 이미지 파일 이름과 버튼 리스너가 들어간다 |
| 2-1. ButtonListener | 버튼을 터치할 때, 어떤 처리를 할지 매개변수로 버튼 이름을 입력받아 처리를 하게 한다. |
| 3. TileData | 타일에 어떤 텍스쳐를 칠해야하는지에 대한 데이터를 가지고 있다. |
| 11. TempMessage | static instance 개체. 간단한 텍스트와 확인 취소 창이 나올 때 이를 사용한다 |
| 12. TempDialog | 특정행동을 할 때, 정말로 할 것인지 다시 물어봐주는 다이얼로그 |
| 13. CampaignApp | AppController랑 동급이다.  캠페인을 실행할 때 다룬다 |

1. ViewData

|  |  |
| --- | --- |
| - String filename | 뷰에 있어서 왼쪽에 들어가게될 아이콘용 이미지 파일 이름 |
| - int data | 뷰는 주로 개수를 나타내므로 몇 개인지에 대해 integer값이면 충분히 나타낼 수 있음 |
| + get/set | String filename, int data에 대한 각각 getter, setter와 생성자함수가 들어간다 |

2. ButtonData

|  |  |
| --- | --- |
| - ButtonListener listener | 터치를 한 경우 어떤 처리를 해야하는지에 대한 함수 처리 |
| - String filename | 어떤 이미지 파일이 들어갈 지 꺼내 놓는다 |
| - String label | 디비에 존재하는 라벨 이름을 따온다 |
| - int id | id |

2-1. ButtonListener

|  |  |
| --- | --- |
| + void onTouch(String label); | 터치를 한 경우 어떤 처리를 해야하는지에 대해 처리한다 |

3. TileData

|  |  |
| --- | --- |
| - int x,y | 해당 타일이 어느 좌표에 위치한지를 나타낸다 |
| - String imageFilename | 이미지 파일 이름 |
| + get/set | String filename, int data에 대한 각각 getter, setter와 생성자함수가 들어간다 |

4. UIDrawer

|  |  |
| --- | --- |
| + void draw() | 버튼과 view를 포함한 UI를 그린다  (버튼과 view정보는 어디서 가져오지?) |
| + void touch(float x, float y) | AppController에 터치를 한 버튼의 라벨 데이터를 반환해준다 |
| + void update() | DB 데이터를 가져와서 동기화한다 |
| + void update(ButtonData[] btn,  ViewData[] views) | 버튼과 뷰데이터를 매개변수로 받아 처리한다 |

5. MapDrawer

|  |  |
| --- | --- |
| + void draw() | 타일 데이터를 그려준다 |
| + void touch(float x, float y) | AppController에 어떤 타일 부분을 터치한건지 타일의 x,y 좌표를 반환해준다. |
| + void update() | DB 데이터를 가져와서 동기화한다 |

6. AppController

|  |  |
| --- | --- |
| - boolean isWorldMap | true일 경우 현재 월드맵임. false일 경우 로컬 맵 |
| - int mapId | 월드맵이 아닐 경우 어느 로컬맵에 위치하는지에 대한 값이다 |
| + void touch(ButtonData data) | 특정 버튼을 눌렀을 경우 이 함수가 호출 됨 |
| + void touch(int[] data,  int building\_id) | 특정 타일을 터치했을 때 이 함수가 호출 됨  building\_id = -1이면 공백이란 얘기 |

7. Builder

|  |  |
| --- | --- |
| + ButtonData[] imageFilenames() | 빌드 가능한 모든 건물에 대한 이미지 데이터를 반환합니다 |
| + boolean build(int id, ButtonData data,  int map\_id) | 빌딩버튼 들중 유저가 선택한 버튼을 id로 받습니다.  id는 building id입니다  리소스 조건이 부합가능한지를 반환값으로 주고 만족하는 경우 미리 감산 시켜놓습니다. |

8. WorldMap

|  |  |
| --- | --- |
| + TileData[] getTiles(int map\_id) | 타일 데이터를 반환받는다  map\_id = 0 : 월드맵  나머지 : 로컬맵 |
| + void war(int x, int y) | 월드맵을 기준으로 특정 타일에 존재하는 지역에 전쟁을 건다. |

9. Building

|  |  |
| --- | --- |
| + void getBuildingData(int building\_id  ViewData[] views, ButtonData[] btns) | 매개변수는 길이 2의 배열이며, 이 배열에 대입함으로써 값을 넘겨주는 방식이다 |
| + void nextTurn() | 소유하고 있는 모든 빌딩이 일어날 수 있는 턴당 작용하는 패시브를 발동시켜준다 |
| + boolean execute(int building\_id) | 해당 빌딩의 기능을 수행한다.  단순히 버튼을 터치후 바로 이곳으로 이동하므로 리소스 조건 등을 확인해봐야 한다 |

10. MainMenu

AppController랑 동급이라고 보면 됨

|  |  |
| --- | --- |
| + void draw() | 메뉴를 그려준다 |
| + void touch(x : float, y : float) | 어떤 버튼을 누르냐에 따라 다르게 처리되도록 함. |

11. TempMessage

|  |  |
| --- | --- |
| + void showMsg(String msg) | 메시지를 표시한다. 캠페인을 위해 주로 쓰인다 |

12. TempDialog

|  |  |
| --- | --- |
| + void showDialog(String msg,  String ok) | 메시지를 표시한다. 캠페인을 위해 주로 쓰인다 |

13. CampaignApp : AppController

|  |  |
| --- | --- |
| + void draw() | 메뉴를 그려준다 |
| + void touch(ButtonData btn) | 어떤 버튼을 누르냐에 따라 다르게 처리되도록 함. |