UML Diagrams

UML is Unified Modeling Language. The UML is an international industry standard graphical notation for describing software analysis and design. (Thomas, 1997). The way we use in UML Diagram is Class Diagrams. Class diagrams are used in both the process of analysis and the design. During the analysis and design phase, we detailed implementation information, including class name, the methods and attribute of the classes, and the relationships among classes.

1. The goal of UML is using a picture to replace thousands of words. The most important goal is defining some general purpose modelling language which all modellers can use and also it needs to be made simple to understand and use. (Tutorialspoint,2016)

2. A conceptual model can be defined as a model consisting of concepts and their relationships. The conceptual model is the first step before drawing a UML diagram. It helps to understand entities in the real world and how they interact with each other. Because UML describes real-time systems, it is important to develop a conceptual model and then proceed step-by-step. The UML conceptual model can be learned by learning the following three main elements: UML building blocks, rules for connecting building blocks, UML Common Mechanism. (Tutorialspoint,2016)

3. UML play an important role in defining different perspective of a system, including design, implementation, process and deployment. The centre is the UML Diagram which connects all these four. A UML Diagram shows the functionality of the system.

4. Design a system consist of classes, interface and collaboration. Implementation defines the components together to show a complete system. The process defined the flow of the system and deployment represents the physical nodes of the system that forms the hardware.

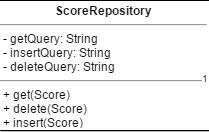
5. According to use cases definition, there should be two sides in the system: server and client side. Client side is separated with the model, controller. The model has main attributes of the uses cases. View side is the screen and results that are shown to the actors in the game. For the server side, controllers are used for keep tracking of movement, state, position and the services of the match. In addition to these databases should keep the records of the game.

6. The top compartment shows the class's name. The middle compartment lists the class's attributes. The bottom compartment lists the class's operations. When drawing a class element on a class diagram, you must use the top compartment, and the bottom two compartments are optional. (The bottom two would be unnecessary on a diagram depicting a higher level of detail in which the purpose is to show only the relationship between the classifiers.) (IBM, 2016)

UML class is represented by the diagram shown below:

* The top section is used to name the class
* The second one is used to show the attributes of the class
* The third section is used to describe the operations performed by the class.

Class



Operations

Attribution

Name

Public

Private

4.1 Client Side



In the client side, match communicator, move communicator and state communicator extend communicator base, and they both aggregate one bot. Main realize one bot and map aggregate bot.

4.2 Server-side Controllers



In the Server-side Controllers, there are four controllers. The first controller is score controller which create score board model. The second controller is game controller. The character model and tile model aggregate game state model which created by state service. Visibility service package imply IVisibilityService in interface and aggregate state service package. State service and movement service imply IStateService and IMovementService respectively in interface and aggregate game controller together. In the part of player controller, authentication service implies IAuthenticationService in interface and aggregate player controller which create login model. In the process of Match controller, match service implies IMatchService in interface and create match status and match result model, and then aggregate match controller at last.

4.3 Server-Side Models & Database



In the Server-Side Models & Database, one tile type aggregate one Tile and large number of tile aggregate one map. One player and one point aggregate one character, one character and one match state aggregate one match, and then varies of matches aggregate one match list which imply IMatchList in interface. In other side, database connect score repository and player repository. The score repository and player repository imply IScoreRepositpry and IPlayerRepository respectively. Finally, the score repository and player repository extend database repository.

Reference

Thomas, R. (1997). Introduction to the Unified Modeling Language. *Technology of Object-Oriented Languages and Systems - Tools*(Vol.126, pp.354). IEEE Computer Society.

<https://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep04/bell/>

https://www.tutorialspoint.com/uml/uml\_overview.htm