The purpose of this software is to create a robot simulation game that we can use keypressed comands to control the robot to reach to homebase before battery is all depleted. Everytime when the robot move or stay, battery will deplete. There are several key elements can be found in the code structure. The design structure and the UML diagram will be covered as follow.

**Design of the simulation:**

When we look at the UML diagram, we know that we can see that there are two main types of ArenaEntity here. One is arena\_mobile\_entity, the other one is arena\_Immobile\_entity. arena\_mobile\_entity is for entities that can move around, such as robot and homebase, while arena\_Immobile\_entity is for the entities that stay at the same position for the whole time. Examples of arena\_Immobile\_entity are Obstacle and RechargeStation. Therefore, we can see from the UML diagram that Robot and HomeBase both inherited from the arena\_mobile\_entity class while Obstacle and RechargeStation both inherited from the arena\_Immobile\_entity. Although HomeBase is not inherited from Robot class, they have similar constructor and functions. They both are able to use functions from RobotMotionBehavor and RobotMotionHandler classes to handle movement of the entity. Besides, Robot is the only entity that have battery. Therefore, only Robot could has access to the RobotBattery class.

EventBaseClass is also one of the base class in this project. Its derived classes are EventCommand, EventCollision, and EventRecharge. All these classes are being called and used its functions when Arena classes need to check for any collision, keypressed commands or recharge events before they can call out other functions. We can also see that when our program starts, GraphicArenaViewer is the first class that being called (at the bottom of the UML diagram). It will create all the GUI parts by drawing out the Robot, HomeBase, Obstacles, RechargeStation, and all the button. It also provide the functionality of the keypressed command. Then it will call Arena for further functions.

One last thing about the UML diagram, a arrow from Arena to HomeBase is missing because I have used up 60/60 complexity in the lucidchart website, and it stops me from adding more lines in the diagram.