Our group's approach to implementing the game was to create a prototype of the final product and improve the prototype over time to reach the desired spot. Our final product contains multiple adjustments compared to the original design. We decided that the option to change the skin and difficulty of the game is not needed since it does not add much to the original version of the game. Additionally, we decided that the option to watch a tutorial of the game on Youtube is not necessary since the game is pretty straightforward and does not require a lot of instructions to play. As a result, our group thinks that it is not necessary to separate the game screen and the menu since the menu would only contain the option to choose the map. Furthermore, we made changes to the leaderboard system such that highscores would be automatically saved to the leaderboard rather than require adding manually. In our original design, the BonusReward can spawn an infinite amount of time, randomly throughout the game. However, in the final product, we decided that the Bonus Reward will only spawn once since it gives us control over the maximum score that the players can get. Our group also decided to make the character able to move faster than the enemy in the final product because it makes the game less frustrating to play.

The final product is much more complex compared to the original UML diagram that we created. We decided to not include some of the classes such as Exit or Entrance since there is not much to do with those classes. However, for the Enemy class, we had to create a lot of additional methods to help with the breadth first search algorithm which was not reflected in the original design. For the Character class, we added multiple methods that manage how the character interacts with other objects in the game. Additionally, we used KeyListener to give control to the character, which was not reflected in the original UML diagram. The Maze class in the original design is renamed to Game and the GameStage in the original design is renamed to GameScreen. The GameScreen class is an extended version of GameStage that is in charge of displaying the game information to the user interface.

Each member of the group is responsible for a class from Character, Enemy, Reward and Punishment. Further additions to the game such as LeaderBoard, Settings and PowerUps are also divided among the members to implement. For the GUI design, our group chooses to use the Java Swing library since some of our members already had experience using it. Additionally, there are a lot of resources related to Java Swing that are available online which makes the library easier to use. To enhance the quality of the code, we provided comments which explains the thinking process in addition to adding JavaDocs comments. We also have group members looking at each others' codes to make sure that the code contains no mistakes. One of the biggest challenges that our group faced is getting used to the softwares that we haven't used before such as Maven. Due to being short 2 members for the duration of the phase, we used a more agile approach where we would think of features and implement them regardless of who was working on a certain part of the game. This allowed us to implement more features in a shorter amount of time, something that was important for us during this phase due to the limitations on the amount of people working.