

1) How will the board be drawn?

We use matrix to draw the board. Every cell in the matrix is a Cell object. Every time the system draws the board, it will update the state of each cell. For example, if a cell is selected by user, it shows miss or hit. We accomplish this by using the function `getIsSelected()` to see whether a cell has been selected. If a cell is selected, the system calls `getIsTank()` to see whether the user hits or misses a tank. If a cell is not selected, the cell's state remains unknown. We will use for loop to check every cell before the updated board is drawn.

If user loses, the system will draw the real board showing all the tanks. For each tank cell, we simply make it selected by calling `setIsSelected()`, so that it will show an 'X' on the map. For each cell that is not part of a tank, we need to check if it has been selected or not by using `getIsSelected()`. If the cell has not been selected, it will become an empty cell on the map. If a cell has been selected, it will be shown a dot.

2) How is the user's move handled?

In the GameBoard class, we have an ArrayList to store 100 Cell objects; hence, we can change user input into an index number of each Cell in the ArrayList by translating each letter into integer, such as a to 0, b to 1 and so on. Then we translate user input into integer by with special calculation. For example, when user types in c8, it indicates the cell object at index 27 ($2 * 10 + 8 - 1 = 27$) in the ArrayList. Once we know which Cell object is targeted, we use `setIsSelected()` to set it as selected and check if it is a tank cell to indicate whether user hits or misses. The tank containing this targeted cell will use this information to return the number of undamaged cells.

If user enters the wrong type or invalid input, we will use some while and if statements to check and loop until user enters the right form.

3) How is the damage each existing tank causes onto user updated?

The Enemy class has an ArrayList of Tank objects and knows the number of working tanks. To update damages each working tank causes onto user, the system calls `getDamages()` function from Enemy class to get the damage of each tank. This method will return the number of undamaged tank cells for each tank if the tank is not fully damaged. Each Tank object knows how many undamaged cells it has by calling `isSelected()` for its ArrayList of Cell objects. Finally, the `displayShotResult()` from GameBoard class can display how much damage each tank is made to the user.