Component Design and Design Tradeoffs

The software is broken down into 7 main components: Map creation, Map display, Striking, Strike hit confirmation, Aircraft generation, Radar, and Map change. These components all work together and so are defined as static functions.

1. Map creation

Map creation is designed to initialize all elements to a number indicating that it is unknown to the user. It is initialized in a for loop that inherits another for loop. The main for loop sets each row to a number and the inherited one sets each column to the same number.

1. Map display

Map display works like the creation but instead of initializing the elements of the array, it includes if statements to determine what should be displayed if the array elements are equal to determined numbers.

1. Striking

Striking is a unique function because it works with the GUI. The way it is designed is it will convert the row and column input into integers and store them into a row and column element in a strike array.

1. Strike Hit Confirmation

Strike hit takes what was entered into the strike array and checks that with the aircraft array. If there is a match then the strikeHit variable is incremented.

1. Aircraft Generation

Aircrafts are generated randomly on the map. If duplicate aircrafts are found on the map the aircraft is regenerated until it finds a unique location to be stored.

1. Radar

The radar comes up after the player strikes. The way it works is if the player strikes a target, the radar checks every possible row and column where that target is located to determine if any other aircrafts are located there. This is a way to guide the player in difficulty mode as the number of attempts are limited.

1. Map Change

Map change determines whether an aircraft is hit or not and will set it to a corresponding number.

Interface Design and Design Tradeoffs

The interface is designed using the GUI components. Labels are used to mark the row and column as well as the icons used on the map. The interface is comprised of a main panel, a bottom panel, an input panel, and a radar panel. The radar and main panel need to be revalidated after striking. The bottom and input panel is for the player to use to play the game. The main components used for the GUI are JFrame, JPanel, JLabel, JMenu, and JMenuItem. The JFrames include: GUI, and frame(for pop ups), The JPanels include: biggerPanel, bottomPanel, inputPanel, and radarPanel. The JLabels include: rowRadar, colRadar, attemptsLabel, strikeHitLabel, row and column.

1. JFrame

The main frame is extended in the constructor and set inside. It makes up the basis for the entire game. Another frame called “frame” is for the JOptionPane dialog which will notify the player depending on what difficulty the player selects also for introducing the player to the game.

1. JPanel

The panels are used to add the labels. The panels layouts are in both grid and border layout design. The grid layout is for the map and the border layout is for the main panel as well as the bottom panel. The different layouts help place the components of the game in convenient places.

1. JLabel

The labels are for giving the player visual cues. The labels guide the player as well as dictate the game. Majority of the game is in labels which are set by icons. Sounds are also added to give a more appealing effect to the player when playing the game.

1. JMenu

The menu bar is for the user to look at main options in the game or to set the difficulty.

1. JMenuItem

These are items added to the menu bar and have distinct functions for each one.

Code Design and Design Tradeoffs

The code design is still being worked on and will most likely be separated into specific classes for each functionality. This will help make the game easier to debug. There are 8 main functions in the code design that should have their own class: GUI, createMap, showMap, createAircrafts, radar, strike, strikeHit, and changeMap.

1. GUI

The GUI is the constructor and initializes all components of the game. It implements ActionListener as well as KeyListener. ActionEvents are fired when the player selects an option. These ActionEvents are listed under an actionPerformed function. Each component of the GUI must have an ActionListener to be able to execute what it is required to do.

1. createMap

The code is designed to use a for loop in a for loop to initialize the map.

1. showMap

Show Map displays icons for values 0, 1 and -1. -1 is set as an unknown gif file. 0 is set as a miss gif file. 1 is set as a hit gif icon.

1. createAircrafts

Create aircrafts uses the random package and generates each aircraft depending on the number of aircraft into a random number 1-10 for row and column.

1. radar

Radar is determined by line and the line is determined by how many aircrafts there are for row and column. If there is an aircraft in that line for row then it increments row, if there is an aircraft in that line for column then it increments column.

1. strike

Striking determines what to put into each element of the strike array based on the user input. Striking can be done from clicking on a JButton or using the Enter key.

1. strikeHit

This confirms whether what the user enters is equal to the aircraft generated. It is a Boolean function that returns true if there is a match or false otherwise.

1. changeMap

changeMap is dependent on strikeHit because if true it will set the value of the map array to 1, if false it sets it to 0.