Main Class:

Our main class is named CSEProject that extends JFrame and implements ActionListener and KeyListener.

We have a method CSEProject here that creates a JFrame. It has a setDefaultCloseOperation method that sets the exit application of the JFrame. The setBounds methods sets the bounds of this frame. A new JPanel is also created here with specific border set in contentPane.setBorder. We have a setContentPane property that sets the properties of contentPane. **contentPane.setLayout(null).** Finally the design method is called.

The design method is a void method that has all the buttons in the main menu. It has the texts in the buttons, the sizes , the positions , the fonts and the foreground of each button. It also has the actionPerformed methods that creates a new object of the specific class each button indicates. As soon as a new object is created , the first/main menu screen is disposed.

SinglePlayer class:

This is the class where the singleplayer game has been designed. This class extends JPanel and implements ActionListener and KeyListenter. The no argument constructor here has a Timer object that initializes both the initial delay and between-event delay to delay milliseconds. A frame is also created here with it’s own title, size, keys. The frame is visible but not resizable. A crosshair is created here as an object of a rectangle. The two ducks and a bomb is created as an array list of rectangles. A for loop is used to place 5 both kinds of ducks and 5 bombs placed randomly using the Math.random function. The repaint method is called and then timer is started.

The paint function takes the parameter Graphics g and the images for the background ,ducks ,bomb, crosshair are set. The g.strawString method sets the Score. The color and font for score is also set here.

The void method “keyPressed” sets what each keys do in our game for the motion of our crosshair. Every time a key is pressed the repaint method is called to paint the screen again. It also has for each loops for the array lists created for the ducks and bombs to show what happens when they collide with the crosshair. The score is incremented to a specific value when the crosshair collides with the ducks and decremented when the crosshair collides with the bomb.

The void method “keyReleased” method sets what happens after each key is pressed. Repaint method is called here again.

The void method “collision” takes in two Rectangle objects and returns the intersections between them.

Multiplayer class:

This is the same class as the SinglePlayer3 class with added functions to detect the motion for the second crosshair for the second player.