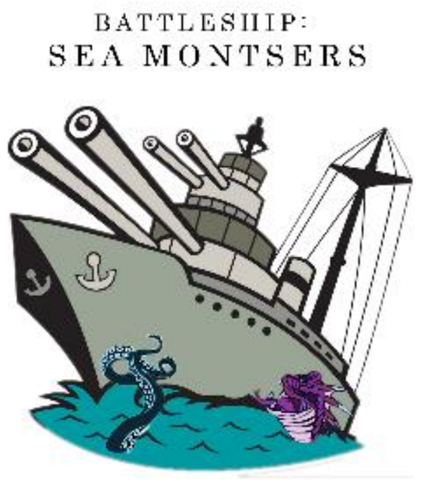

CS1810 Software Implementation ASSIGNMENT 3

GAME 4: BATTLESHIP VARIATION B

Usen-ita U. Asanga | 1921943 | Tutor: Mr. Alaa Marshan | Group Yellow12

Contents

Introduction

The assignment brief needed each member of the group to choose a variation out of the four given games. I chose game four – variation b.

the game is titled *Battleship*. The aim of the game is for the player to sink all five ships that are placed on a ten-by-ten square board.

- Aircraft carrier: 5 squares long
- Battleship: 4 squares long
- Submarine: 3 squares long
- Destroyer: 3 squares long
- Patrol Boat: 2 squares long

The tricky part is all ships are invisible. The variation adds two sea monsters, *Kraken* and *Cetus*. This report contains the algorithm of the game design.

Requirements Specification

FUNCTIONAL REQUIREMENTS

The functionalities for the game include.

1. Place the five type of ship of different lengths at random on 10 by 10 board.
2. place Kraken on random unused square
3. place Cetus on random unused square
4. Display empty board, current un-sunk ships and moves.
5. Player clicks the X, Y coordinates of shot.
6. Board updates by adding an 'M' on choses square, if it was a miss, an 'H', if it was a hit, a ‘K’, if Kaken is hit or a ‘C’, if Cetus is hit.
7. AI validates shot coordinates then display the appropriate message   according to the outcome of the shot:

   - My ship was hit!

   - You missed!

   - You sank my [ship type]!

   - Kraken hit!

   - Cetus hit!

- All ships revived!

8. AI displays current list of sunk, un-sunk ships and current moves.
9. Player sinks all ships.
10. Display final score.
11. Ask player question "PLAY AGAIN"
12. Player clicks "QUIT", go to GameOver page.
13. AI overlaps two or more ships, redo placement of ships in error.
14. AI place a ship(s) out of bounds, redo placement of ship(s) in error.
15. Click was a miss('M'), take away one from score.
16. Click was a hit('H'), add one to score.
17. Clicks hit('H') or miss('M') square, ignore click.
18. Kraken is hit('K'), score equals zero.
19. Cetus is hit('C'), AI redoes placement of un-suck ships.
20. Sinks Aircraft carrier, add ten to score.
21. Sinks Battleship, add eight to score.
22. Sinks Submarine, add six to score.
23. Sinks Destroyer, add six to score.
24. Sinks patrol Boat, add four to score.

NON-FUNCTIONAL REQUIREMENTS

These are how the game must behave for a good experience.

1. Be appealing and easy to use.
2. Have a time delay of less than three seconds before starting the battleship game.
3. Play sound effect on every ship that sinks.
4. Have background ocean ambient audio playing during game.
5. Update player's score and moves >0.5 seconds.

Additional Functions Explained

All additional functions above have a dark pink font color.

Functional:

- Player Moves: At the start, the player number of shots to win the game is counted. Moves increases per shoot. This motivates the player to decipher ways to finish game with less moves.

- Addition of ‘K’ to hit Kraken and ‘C’ to hit Cetus: This reduces ambiguity in identifying which shot hit the sea monsters.
- The messages, “Kraken hit!” and “Cetus hit” will let be aware of which monster has been hit.
- The message, “All ships revived!” will alert the user of the state of the ships.

Non-Functional:

- Have a time delay of less than three seconds before starting the battleship game: This will improve the feel of the game.

- Have background ocean ambient audio playing during game: This audio will make the user feel they are in are in the middle of the battle.

- Play sound effect on every ship that sink: This audio will improve the feel of the game by alerting the user that a ship has sunk.

Algorithm and User Interface Design

Differences

Design Algorithm	Java Code	Reasons for Change
Algorithm created for console interface.	A GUI interface	- Ease of use: average computer users can easily interact with the interface without having to type special keys for an action. - Looks more professional. - The interface has a better chance of being re-played by a user.
Algorithm design seemed to have only one class.	Code has three classes	- Code is more maintainable. - Code is more testable. - Code is more reusable.

Test Case

Test Log For Project: Rules Algorithm					
Purpose: Testing *showWindow*				**Date:** 15-02-2021	
Run Number: 2				**Carried Out By:** 1921943	
Action(Input)	**Expected Output**	**Observed Output**	**Pass/Fail**	**Reasons for Failure**	**Assigned To**
Test showWindow method runs	GUI display	GUI display	Pass	n/a	n/a

Test Log for Project: BattleshipSeaMonsters Algorithm					
Purpose: Testing funtional requirements				**Date:** 22-02-2021	
Run Number: 2				**Carried Out By:** student number; 1921943	
Action(Input)	**Expected Output**	**Observed Output**	**Pass/Fail**	**Reasons for Failure**	**Assigned To**
Patrol boat deployed	Patrol boat position in run 1 not equal to run 2	Patrol boat position in run 1 not equal to run 2	Pass	n/a	n/a
Submarine deployed	Submarine position in run 1 not equal to run 2	Submarine position in run 1 not equal to run 2	Pass	n/a	n/a
Destroyer deployed	Destroyer position in run 1 not equal to run 2	Destroyer position in run 1 not equal to run 2	Pass	n/a	n/a
Battleship deployed	Battleship position in run 1 not equal to run 2	Battleship position in run 1 not equal to run 2	Pass	n/a	n/a
Aircraft Carrier deployed	Aircraft Carrier position in run 1 not equal to run 2	Aircraft Carrier position in run 1 not equal to run 2	Pass	n/a	n/a
Kraken deployed	Kraken position in run 1 not equal to run 2	Kraken position in run 1 not equal to run 2	Pass	n/a	n/a
Cetus deployed	Cetus position in run 1 not equal to run 2	Cetus position in run 1 not equal to run 2	Pass	n/a	n/a
Board initialised	Board state empty	Board state empty	Pass	n/a	n/a

Moves display	intial moves equals to zero and is diplayed	intial moves equals to zero and is diplayed	Pass	n/a	n/a
Un-sunk ships display	Un-sunk ships displayed	Un-sunk ships display	Pass	n/a	n/a
click button on 10 x 10 board for shoot	Board state change based on button clicked	Board state change based on button clicked	Pass	n/a	n/a
button clicked has no ship, Kraken and Cetus	Board state change to 'M', "You missed!" is displayed, decrement of score and moves increment	Board state change to 'M', "You missed!" is displayed, decrement of score and moves increment	Pass	n/a	n/a
button clicked has a ship	Board state change to 'H', "My ship was hit!" is displayed, increment of score, and moves	Board state change to 'H', "My ship was hit!" is displayed, increment of score, and moves	Pass	n/a	n/a
button clicked has a Kraken	Board state change to 'K', "Kraken hit!" and 'X' is displayed, score multiped by 0 and increment of moves	Board state change to 'K', "Kraken hit!" and 'X is displayed, score multiped by 0 and increment of moves	Pass	n/a	n/a
button clicked has a Cetus	Board state change to 'C', "Cetus hit!" and 'X is displayed, Un-sunk ships redeployed and increment of moves	Board state change to 'C', "Cetus hit!" and 'X is displayed, Un-sunk ships redeployed and increment of moves	Pass	n/a	n/a
sunk patrol boat	"XX" and "You sank my Patrol boat!" is displayed, add four to score	"XX" and "You sank my Patrol boat!" is displayed, add four to score	Pass	n/a	n/a
sunk Submarine	"XXX" and "You sank my Submarine!" is displayed, add six to score	"XXX" and "You sank my Submarine!" is displayed, add six to score	Pass	n/a	n/a
sunk Destroyer	"XXX" and "You sank my Destroyer!" is displayed, add six to score	"XXX" and "You sank my Destroyer!" is displayed, add six to score	Pass	n/a	n/a
sunk Battleship	"XXXX" and "You sank my Battleship!" is displayed, add eight to score	"XXXX" and "You sank my Battleship!" is displayed, add eight to score	Pass	n/a	n/a

sunk Aircraft Carrier	"XXXXX" and "You sank my Aircraft Carrier!" is displayed, add ten to score	"XXXX" and "You sank my Aircraft Carrier!" is displayed, add ten to score	Pass	n/a	n/a
all ships sunk	Displayed final score in GameOver algorithm	Displayed final score in GameOver algorithm	Pass	n/a	n/a
Test "QUIT" button	Displayed final score in GameOver algorithm	Displayed final score in GameOver algorithm	Pass	n/a	n/a

Test Log for Project: GameOver Algorithm					
Purpose: Testing *showWindow method*				**Date:** 15-02-2021	
Run Number: 1				**Carried Out By:** student number; 1921943	
Action(Input)	**Expected Output**	**Observed Output**	**Pass/Fail**	**Reasons for Failure**	**Assigned To**
Test showWindow method runs	GUI display	Error message: Cannot invoke "javax.swing.JLabel.setBounds(int, int, int, int)"	Fail	Score value in BattleshipSeaMonsters Algorithm is null	student number; 1921943
Test showWindow method runs when score value in BattleshipSeaMonsters algorithm is not null	GUI displayed score value	GUI displayed score value	Pass	n/a	n/a
Test "PLAY AGAIN" button	Display Rules algorithm GUI	Display Rules algorithm GUI	Pass	n/a	n/a

Source Code Listing

RULES CLASS

```
package Game;

import java.awt.BorderLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;

import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.Clip;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;

public class Rules {
        private static JButton Button;
        private static ImageIcon Icon1;
        private static JLabel myLabel;
        private static ImageIcon Icon2;
        private static JLabel myLabel2;
        private static ImageIcon LGO;

        public static void main(String[] args) {

                showWindow();

        }

        public static void showWindow() {

                Icon1 = new ImageIcon("RulesImages_01.jpg");
                Icon2 = new ImageIcon("RulesImages_02.jpg");
                myLabel = new JLabel(Icon1);
                myLabel.setSize(1280, 720);
                myLabel2 = new JLabel(Icon2);
                myLabel2.setSize(1280, 720);
                LGO = new ImageIcon("group12logo.jpg");
        // The application window.
        JPanel panel = new JPanel();
        JFrame frame = new JFrame("Battleship Sea Monsters: Rules");
        frame.setSize(1280, 720); // the (x,y) lengths
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // to make sure everything closes
        frame.setLocationRelativeTo(null);

        JScrollPane scroller = new JScrollPane(panel);
        scroller.setViewportView(panel);

        panel.setLayout(new GridLayout(2, 1, 70, 0));
        panel.setSize(1280, 720);
        panel.add(myLabel);
        panel.add(myLabel2);

        frame.setLocationRelativeTo(null);
        frame.add(scroller);
        frame.setIconImage(LGO.getImage());

        // the button
        Button = new JButton("START");
        Button.setBounds(140, 50, 300, 50);
        Button.setFont(new Font("Stencil", Font.BOLD, 30));
        Button.addActionListener(new ActionListener() {

                public void actionPerformed(ActionEvent arg0) {
                        File paper_crumble = new File("Battleship Audio/Paper crumble.wav");
                        PlaySound(paper_crumble);

                        frame.setVisible(false);
                        BattleshipSeaMonsters window = new BattleshipSeaMonsters();
                        window.showWindow();
                }
        });
        frame.add(BorderLayout.SOUTH, Button);
        frame.setResizable(false);

        frame.setVisible(true);
}

public static void PlaySound(File Sound) {
        try {
                Clip clip = AudioSystem.getClip();
                clip.open(AudioSystem.getAudioInputStream(Sound));
                clip.start();

                Thread.sleep(clip.getMicrosecondLength() / 1000);
        } catch (Exception e) {
                }
        }
}
```

BATTLESHIPSEAMONSTERS CLASS

```
package Game;

import java.awt.Color;
import java.awt.Font;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.util.Random;

import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.Clip;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;

public class BattleshipSeaMonsters {
        private static JButton Button;
        private static ImageIcon BKG;
        private static JLabel myLabel;
        private static ImageIcon LGO;
        public static JLabel ScLabel;
        private static JLabel MLabel;
        private static JLabel label;
        private static JLabel Label2;
        //// battleship variables////
        private static Random random = new Random();
        static JButton[][] buttons = new JButton[10][10];
        private static int score = 0;
        private static int moves = 0;
        private static int X = 0;
        private static int Y = 0;
        private static int board_output;
        private static int[][] board = new int[10][10];
        private static int[][] ships = new int[10][10];
        private static int[] shipsLengths = { 2, 3, 3, 4, 5 };
        private static int[][] Kraken = new int[10][10];
        private static int[][] Cetus = new int[10][10];
        private static int[] Monsters = { 1, 2 };
```

```java
private static int kill = 0;
//// ships States////
private static JLabel ALabel;
private static JLabel BLabel;
private static JLabel SLabel;
private static JLabel DLabel;
private static JLabel PLabel;
//// monster States////
private static JLabel KLabel;
private static JLabel CLabel;

public static void main(String[] args) {
        showWindow();
}

public static void showWindow() {
        try {
                Thread.sleep(1000);
        } catch (InterruptedException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
        }

        // Audio files
        File backgroundNoise = new File("Battleship Audio/Ocean shore.wav");
        PlaySound(backgroundNoise);

        // Background and icon image
        BKG = new ImageIcon("GameBackground_03.png");
        myLabel = new JLabel(BKG);
        myLabel.setSize(1280, 720);
        LGO = new ImageIcon("group12logo.jpg");

        // The board
        JPanel button_panel = new JPanel();
        button_panel.setLayout(new GridLayout(10, 10));
        button_panel.setBackground(Color.BLACK);
        button_panel.setBounds(120, 110, 500, 400);

        // The application window.
        JFrame frame = new JFrame("Battleship Sea Monsters");
        frame.setSize(1280, 740); // the (x,y) lengths
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // to make sure everything closes
        frame.setLocationRelativeTo(null);
        frame.setLayout(null);
        frame.setIconImage(LGO.getImage());

        // the Quit button
```

```java
Button = new JButton("Quit");
Button.setBounds(940, 550, 300, 50);
Button.setFont(new Font("Stencil", Font.PLAIN, 30));
Button.addActionListener(new ActionListener() {

        public void actionPerformed(ActionEvent arg0) {
                File click = new File("Battleship Audio/Light switch.wav");
                PlaySound(click);

                frame.setVisible(false);
                GameOver window = new GameOver();
                window.showWindow();
        }
});

// Moves Label
JPanel Mcontainer = new JPanel();
Mcontainer.setBounds(240, 560, 90, 30);
MLabel = new JLabel("0");
MLabel.setBounds(500, 500, 100, 100);
MLabel.setFont(new Font("Stencil", Font.PLAIN, 30));
Mcontainer.add(MLabel);

// Score Label
JPanel Scontainer = new JPanel();
Scontainer.setBounds(240, 520, 90, 30);
ScLabel = new JLabel("0");
ScLabel.setBounds(500, 500, 100, 100);
ScLabel.setFont(new Font("Stencil", Font.PLAIN, 30));
Scontainer.add(ScLabel);

// The Comment label ("you missed!","My ships was hit!")
label = new JLabel("");
label.setBounds(440, 520, 270, 45); // the (x, y, width, height)
label.setFont(new Font("Stencil", Font.PLAIN, 30));
label.setForeground(Color.WHITE);

// The Comment label ("you sank my <ship type>")
Label2 = new JLabel("");
Label2.setBounds(440, 560, 540, 45); // the (x, y, width, height)
Label2.setFont(new Font("Stencil", Font.PLAIN, 15));
Label2.setForeground(Color.WHITE);

// killed ships
PLabel = new JLabel("");
PLabel.setBounds(820, 430, 60, 40);
PLabel.setFont(new Font("Stencil", Font.PLAIN, 40));
```

```java
DLabel = new JLabel("");
DLabel.setBounds(820, 365, 100, 40);
DLabel.setFont(new Font("Stencil", Font.PLAIN, 40));

SLabel = new JLabel("");
SLabel.setBounds(820, 300, 100, 40);
SLabel.setFont(new Font("Stencil", Font.PLAIN, 40));

BLabel = new JLabel("");
BLabel.setBounds(820, 240, 140, 40);
BLabel.setFont(new Font("Stencil", Font.PLAIN, 40));

ALabel = new JLabel("");
ALabel.setBounds(820, 160, 180, 40);
ALabel.setFont(new Font("Stencil", Font.PLAIN, 40));

// Killed Monsters
KLabel = new JLabel("");
KLabel.setBounds(1060, 190, 60, 40);
KLabel.setFont(new Font("Stencil", Font.PLAIN, 40));

CLabel = new JLabel("");
CLabel.setBounds(1060, 260, 60, 40);
CLabel.setFont(new Font("Stencil", Font.PLAIN, 40));

for (int i = 0; i < 10; i++) {
        for (int j = 0; j < 10; j++) {
                buttons[i][j] = new JButton();
                button_panel.add(buttons[i][j]);
                buttons[i][j].setFocusable(false);
                buttons[i][j].setBackground(Color.WHITE);
                buttons[i][j].setFont(new Font("Stencil", Font.PLAIN, 10));
                buttons[i][j].addActionListener(new ActionListener() {

                        public void actionPerformed(ActionEvent arg0) {

                                for (int i = 0; i < 10; i++) {
                                        for (int j = 0; j < 10; j++) {
                                                if (arg0.getSource() == buttons[i][j]) {
                                                        if (buttons[i][j].getText() == "") {
                                                                if ((board_output == 0 && ships[i][j] == 0)
                                                                                && (Kraken[i][j] == 0 && Cetus[i][j] == 0)) {

                                                                        buttons[i][j].setText("M");
                                                                        label.setText("You Missed!");
                                                                        score--;
                                                                        ScLabel.setText(String.valueOf(score));
                                                                        moves++;
        MLabel.setText(String.valueOf(moves));
} else if (board_output == 1 || ships[i][j] == 1) {

        buttons[i][j].setText("H");
        buttons[i][j].setFont(new Font("Stencil", Font.PLAIN, 20));
        label.setText("My Ship was hit!");
        int x = score;
        x++;
        kill++;
        if (kill == 5) {
                File killsound = new File("Battleship Audio/Rocks.wav");
                PlaySound(killsound);
                ALabel.setText("X X X X X");
                Label2.setText("You sank my Aircraft Crarrier!");
                frame.setVisible(false);
                GameOver window = new GameOver();
                window.showWindow();
                score = x + 10;
        } else if (kill == 4) {
                File killsound = new File("Battleship Audio/Rocks.wav");
                PlaySound(killsound);
                BLabel.setText("X X X X");
                Label2.setText("You sank my Battleship!");
                score = x + 8;
        } else if (kill == 3) {
                File killsound = new File("Battleship Audio/Rocks.wav");
                PlaySound(killsound);
                DLabel.setText("X X X");
                Label2.setText("You sank my Destroyer!");
                score = x + 6;
        } else if (kill == 2) {
                File killsound = new File("Battleship Audio/Rocks.wav");
                PlaySound(killsound);
                SLabel.setText("X X X");
                Label2.setText("You sank my Submarine!");
                score = x + 6;
        } else if (kill == 1) {
                File killsound = new File("Battleship Audio/Rocks.wav");
                PlaySound(killsound);
                PLabel.setText("X X");
                Label2.setText("You sank my Patrol Boat!");
                score = x + 4;
        }
        ScLabel.setText(String.valueOf(score));
        moves++;
        MLabel.setText(String.valueOf(moves));

} else if (board_output == -1 || Kraken[i][j] == 1) {
```

```
                                                                        buttons[i][j].setText("K");
                                                                        label.setText("Kraken hit!");
                                                                        KLabel.setText("X");
                                                                        int x = score;
                                                                        score = x * 0;
                                                                        ScLabel.setText(String.valueOf(score));
                                                                        moves++;
                                                                        MLabel.setText(String.valueOf(moves));
                                                                } else if (board_output == -2 || Cetus[i][j] == 1) {

                                                                        buttons[i][j].setText("C");
                                                                        label.setText("Cetus hit!");
                                                                        Label2.setText("All ships revived!");
                                                                        CLabel.setText("X");
                                                                        int x = kill;
                                                                        kill = x * 0;
                                                                        ALabel.setText("");
                                                                        BLabel.setText("");
                                                                        DLabel.setText("");
                                                                        SLabel.setText("");
                                                                        PLabel.setText("");
                                                                        // #method for ship placement//
                                                                        Formships(X, Y);
                                                                        moves++;
                                                                        MLabel.setText(String.valueOf(moves));

                                                                }

                                                        }
                                                }
                                        }
                                }

                        }
                });
                board[i][j] = -3;
                buttons[i][j].setText(String.valueOf(board[i][j]));

        }
}

////// constructing the frame/////
frame.add(label); // comments label
frame.add(Label2);// kill comment
frame.add(ALabel); // aircraft carrier state label
frame.add(BLabel); // Battleship state label
frame.add(DLabel); // Destroyer state label
```

```
        frame.add(SLabel); // Submarine state label
        frame.add(PLabel); // Patrol boat state label
        frame.add(KLabel); // Kraken state label
        frame.add(CLabel); // Cetus state label
        frame.add(Mcontainer); // moves panel
        frame.add(Scontainer); // score panel
        frame.add(button_panel); // battleship board
        frame.add(Button); // Quit button
        frame.add(myLabel); // game background image
        frame.setResizable(false);
        frame.setVisible(true);

        initboard(button_panel);

        Formships(X, Y);

        FormMonsters(X, Y);

}

public static void PlaySound(File Sound) {
        try {
                Clip clip = AudioSystem.getClip();
                clip.open(AudioSystem.getAudioInputStream(Sound));
                clip.start();

        } catch (Exception e) {

        }
}

public static void initboard(JPanel button_panel) {
        for (int i = 0; i < 10; i++) {
                for (int j = 0; j < 10; j++) {
                        if (board[i][j] == -3) {
                                buttons[i][j].setText("");
                        } else if (board[i][j] == -2) {
                                buttons[i][j].setText("C");
                                board_output = -2;
                                Cetus[i][j] = 1;
                        } else if (board[i][j] == -1) {
                                buttons[i][j].setText("K");
                                board_output = -1;
                                Kraken[i][j] = 1;
                        } else if (board[i][j] == 0) {
                                buttons[i][j].setText("M");
                                board_output = 0;
                                ships[i][j] = 0;
```

```
					Kraken[i][j] = 0;
					Cetus[i][j] = 0;
				} else if (board[i][j] == 1) {
					buttons[i][j].setText("H");
					board_output = 1;
					ships[i][j] = 1;
				}

			}
		}
	}

	public static void Formships(int X, int Y) {

		int shipLength = shipsLengths[4];

		for (int i = 0; i <= shipLength;) {
			X = random.nextInt(10);
			Y = random.nextInt(10);

			if ((X >= 0 && X < 10) && (Y >= 0 && Y < 10) && (board[X][Y] == -3)) {
				ships[X][Y] = 1;
				i++;
			}
		}
	}

	public static void FormMonsters(int X, int Y) {
		int MonstersLength = Monsters[1];

		for (int i = 0; i < MonstersLength;) {
			X = random.nextInt(10);
			Y = random.nextInt(10);

			if ((X >= 0 && X < 10) && (Y >= 0 && Y < 10) && (board[X][Y] == -3)) {

				if (i == 0) {
					Kraken[X][Y] = 1;
				} else if (i == 1) {
					Cetus[X][Y] = 1;
				}

				i++;
			}
		}
	}

}
```

GAMEOVER CLASS

```
package Game;

import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;

public class GameOver {
        private static JButton Button2;
        private static ImageIcon BKG;
        private static ImageIcon LGO;
        private static JLabel myLabel;
        private static JLabel SLabel2;

        public static void main(String[] args) {
                showWindow();
        }

        @SuppressWarnings("removal")
        public static void showWindow() {
                try {
                        Thread.sleep(1100);
                } catch (InterruptedException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }
                BKG = new ImageIcon("QuiteBackground.jpg");
                myLabel = new JLabel(BKG, JLabel.CENTER);
                myLabel.setSize(1280, 720);
                LGO = new ImageIcon("group12logo.jpg");

                // The application window.
                JFrame frame = new JFrame("Battleship Sea Monsters: GameOver");
                frame.setSize(1280, 720); // the (x,y) lengths
                frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // to make sure everything closes
                frame.setLocationRelativeTo(null);
                frame.setLayout(null);

                // Score Label
        JPanel Scontainer = new JPanel();
        Scontainer.setBounds(430, 210, 410, 250);
        SLabel2 = new JLabel();
        SLabel2 = BattleshipSeaMonsters.ScLabel;
        SLabel2.setBounds(500, 500, 350, 250);
        SLabel2.setFont(new Font("Stencil", Font.PLAIN, 300));
        Scontainer.add(SLabel2);

        // the button2
        Button2 = new JButton("Play Again");
        Button2.setBounds(470, 530, 300, 50);
        Button2.setFont(new Font("Stencil", Font.PLAIN, 30));
        Button2.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent arg0) {
                frame.setVisible(false);
                Rules window = new Rules();
                window.showWindow();
            }
        });

        ////// constructing the frame/////
        frame.setIconImage(LGO.getImage());
        frame.add(Button2);
        frame.add(Scontainer);
        frame.add(myLabel);
        frame.setResizable(false);
        frame.setVisible(true);
    }
}
```