

Lab 8 Code/Output

Sol Ben-Ishay

Code:

FindorDie.java

```
FindOrDie.java

package FindOrDieGame;

import javax.swing.*;
import java.awt.*;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.util.ArrayList;
import java.util.Random;

public class FindOrDieGUI {

    FindOrDie game = new FindOrDie();
    ArrayList<JButton> cells = new ArrayList<>();

    FindOrDieGUI() {

        // Get the board size from the player
        while (!game.validBoardSize) {
            int input =
Integer.parseInt(JOptionPane.showInputDialog(null,"How big of a board (nxn)
do you want to play?"));
            if (((input%2) == 0)|| (input==1)) {
                JOptionPane.showMessageDialog(null,"The board size must be an
odd number greater than 1!");
            }
            else {
                game.boardDimension = input;
                game.numSquares = input*input;
                game.reqNumMinesAndSecondChances = game.numSquares/2;
                game.validBoardSize = true;
            }
        }

        game.buildBoard(game.boardDimension);

        // Set Default Close Operation
        JFrame frame = new JFrame("Find or Die");
        JPanel mainPanel = new JPanel();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Create the Clickable Cells
        for (int i = 0; i < game.numSquares; i++) {
            cells.add(new JButton()); // Add Cells/JButtons to the 'cells'
JButton ArrayList

```

```

        cells.get(i).setText("Square " + (i + 1));
        cells.get(i).setBackground(Color.GRAY);
        cells.get(i).setName("Square" + (i+1));

cells.get(i).setBorder(BorderFactory.createLineBorder(Color.BLACK));
    int cellNum = i;

    // Add Mouse/Click Listeners to Each of the Cells/JButtons
    cells.get(i).addMouseListener(new MouseAdapter() {
        @Override
        public void mouseClicked(MouseEvent e) {
            super.mouseClicked(e);
            // Actions dependent on which cell/JButton was clicked
            if (game.cellClicked.get(cellNum)) {
                JOptionPane.showMessageDialog(null,"Cell was already
clicked try a different one!");
            }
            else if (!game.cellClicked.get(cellNum)) {
                if (cellNum == game.prizeLocation) {
                    JOptionPane.showMessageDialog(null,"You clicked
the prize! You won!" + "\n" + "Game over!");
                    game.gameOver = true;
//                    System.exit(0);
                }
                else if (game.mineLocations.contains(cellNum)) {
                    JOptionPane.showMessageDialog(null,"You clicked a
mine! You lost!" + "\n" + "Game over!");
                    game.gameOver = true;
//                    System.exit(0);
                }
                else if
(game.secondChanceLocations.contains(cellNum)) {
                    int takeChance =
Integer.parseInt(JOptionPane.showInputDialog(null, ""
                    You clicked a second chance!
                    Would you like to roll the dice for a
chance to continue with a roll of 5 or 6?
                    Enter 0 for yes, 1 to end the
game:\s""));
                    while ((takeChance != 0) && (takeChance != 1)) {
                        takeChance =
Integer.parseInt(JOptionPane.showInputDialog(null,"Invalid choice! Please
enter 0 or 1: "));
                    }
                    if (takeChance == 0) {
                        Random dice = new Random();
                        int dieValue = dice.nextInt(6) + 1;
                        if (dieValue > 4) {
                            JOptionPane.showMessageDialog(null,"You
rolled a " + dieValue + "!\n" + "The second chance was worth it!");
                            // System.out.println("The second chance
was worth it!");
                        }
                        else {
                            JOptionPane.showMessageDialog(null,"You
rolled a " + dieValue + "!\n" + "You lost!" + "\n" + "Game over!");
                            game.gameOver = true;

```

```

//                                System.exit(0);
                                }
                                }
                                else {
                                    JOptionPane.showMessageDialog(null, "Game
quit!");
                                    game.gameOver = true;
//                                System.exit(0);
                                    }
                                    game.cellClicked.set(cellNum, true); // If cell
was clicked, set it's index in the cellClicked array to true
                                    cells.get(cellNum).setBackground(Color.RED); //
If cell was clicked and game can continue, i.e "Second Chance" change
background color of that cell to red
                                    cells.get(cellNum).setText("Clicked!"); // If
cell was clicked and game can continue, i.e "Second Chance" change text of
that cell to clicked
                                }
                            }
                        }
                    });

                    // Add the cells/JButtons to the main panel
                    mainPanel.add(cells.get(i));
                }

                // Layout
                frame.setSize(600,450);

                mainPanel.setSize(600, 450);
                mainPanel.setLayout(new java.awt.GridLayout(game.boardDimension,
game.boardDimension));
                mainPanel.setPreferredSize(new Dimension(400, 400));

                // Add Panel to Frame
                frame.add(mainPanel);

                // Set Visible
                frame.setVisible(true);
            }

            public static void main(String[] args) {
                FindOrDieGUI play = new FindOrDieGUI();
            }
        }
}

```

FindOrDieGUI.java

```

FindOrDieGUI.java

package FindOrDieGame;

import javax.swing.*.*;
import java.awt.*.*;

```

```

import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.util.ArrayList;
import java.util.Random;

public class FindOrDieGUI {

    FindOrDie game = new FindOrDie();
    ArrayList<JButton> cells = new ArrayList<>();

    FindOrDieGUI() {

        // Get the board size from the player
        while (!game.validBoardSize) {
            int input =
Integer.parseInt(JOptionPane.showInputDialog(null,"How big of a board (nxn)
do you want to play?"));
            if (((input%2) == 0) || (input==1)) {
                JOptionPane.showMessageDialog(null,"The board size must be an
odd number greater than 1!");
            }
            else {
                game.boardDimension = input;
                game.numSquares = input*input;
                game.reqNumMinesAndSecondChances = game.numSquares/2;
                game.validBoardSize = true;
            }
        }

        game.buildBoard(game.boardDimension);

        // Set Default Close Operation
        JFrame frame = new JFrame("Find or Die");
        JPanel mainPanel = new JPanel();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Create the Clickable Cells
        for (int i = 0; i < game.numSquares; i++) {
            cells.add(new JButton()); // Add Cells/JButtons to the 'cells'
JButton ArrayList
            cells.get(i).setText("Square " + (i + 1));
            cells.get(i).setBackground(Color.GRAY);
            cells.get(i).setName("Square" + (i+1));

cells.get(i).setBorder(BorderFactory.createLineBorder(Color.BLACK));
            int cellNum = i;

            // Add Mouse/Click Listeners to Each of the Cells/JButtons
            cells.get(i).addMouseListener(new MouseAdapter() {
                @Override
                public void mouseClicked(MouseEvent e) {
                    super.mouseClicked(e);
                    // Actions dependent on which cell/JButton was clicked
                    if (game.cellClicked.get(cellNum)) {
                        JOptionPane.showMessageDialog(null,"Cell was already
clicked try a different one!");
                    }
                }
            });
        }
    }
}

```

```

        else if (!game.cellClicked.get(cellNum)) {
            if (cellNum == game.prizeLocation) {
                JOptionPane.showMessageDialog(null, "You clicked
the prize! You won!" + "\n" + "Game over!");
                game.gameOver = true;
                System.exit(0);
            }
            else if (game.mineLocations.contains(cellNum)) {
                JOptionPane.showMessageDialog(null, "You clicked a
mine! You lost!" + "\n" + "Game over!");
                game.gameOver = true;
                System.exit(0);
            }
            else if
(game.secondChanceLocations.contains(cellNum)) {
                int takeChance =
Integer.parseInt(JOptionPane.showInputDialog(null, ""
                You clicked a second chance!
                Would you like to roll the dice for a
chance to continue with a roll of 5 or 6?
                Enter 0 for yes, 1 to end the
game:\s""));
                while ((takeChance != 0) && (takeChance != 1)) {
                    takeChance =
Integer.parseInt(JOptionPane.showInputDialog(null, "Invalid choice! Please
enter 0 or 1: "));
                }
                if (takeChance == 0) {
                    Random dice = new Random();
                    int dieValue = dice.nextInt(6) + 1;
                    if (dieValue > 4) {
                        JOptionPane.showMessageDialog(null, "You
rolled a " + dieValue + "!\n" + "The second chance was worth it!");
                    }
                    else {
                        JOptionPane.showMessageDialog(null, "You
rolled a " + dieValue + "!\n" + "You lost!" + "\n" + "Game over!");
                        game.gameOver = true;
                        System.exit(0);
                    }
                }
                else {
                    JOptionPane.showMessageDialog(null, "Game
quit!");
                    game.gameOver = true;
                    System.exit(0);
                }
                game.cellClicked.set(cellNum, true); // If cell
was clicked, set it's index in the cellClicked array to true
                cells.get(cellNum).setBackground(Color.RED); //
If cell was clicked and game can continue, i.e "Second Chance" change
background color of that cell to red
                cells.get(cellNum).setText("Clicked!"); // If
cell was clicked and game can continue, i.e "Second Chance" change text of
that cell to clicked
            }
        }
    }
}

```

```

    });

    // Add the cells/JButtons to the main panel
    mainPanel.add(cells.get(i));
}

// Layout
frame.setSize(600,450);

mainPanel.setSize(600, 450);
mainPanel.setLayout(new java.awt.GridLayout(game.boardDimension,
game.boardDimension));
mainPanel.setPreferredSize(new Dimension(400, 400));

// Add Panel to Frame
frame.add(mainPanel);

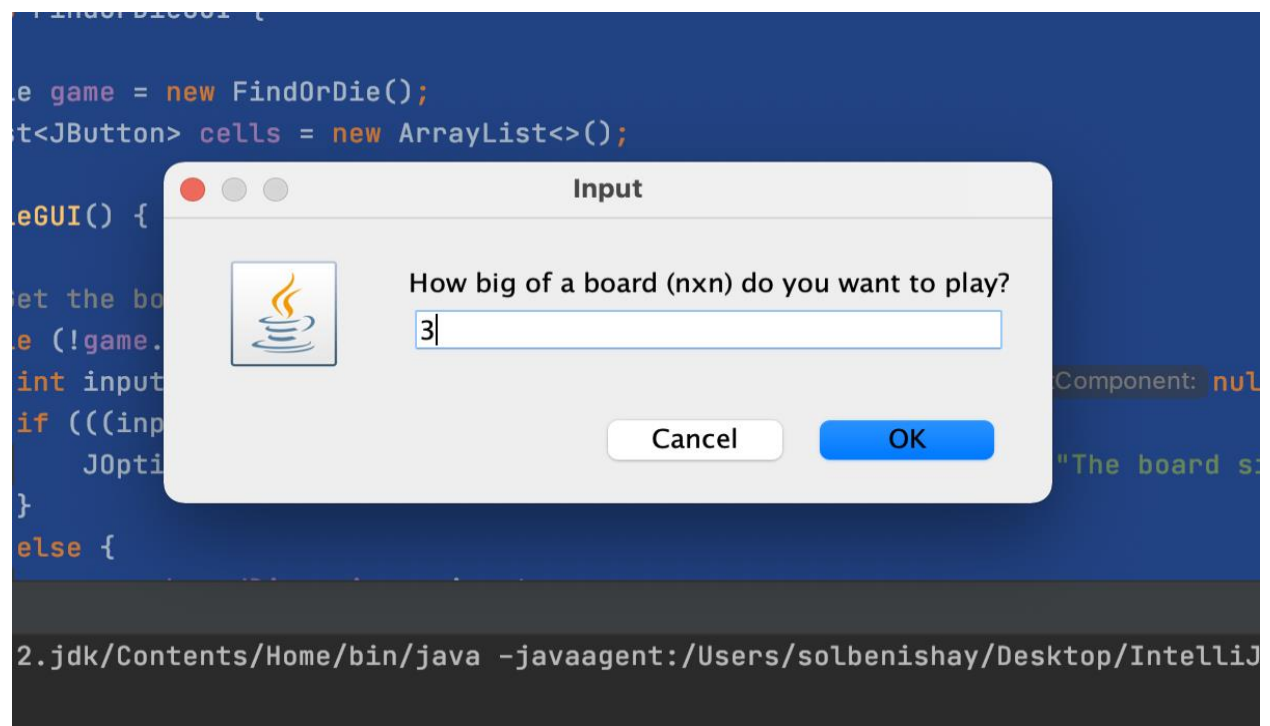
// Set Visible
frame.setVisible(true);
}

public static void main(String[] args) {
    FindOrDieGUI play = new FindOrDieGUI();
}
}

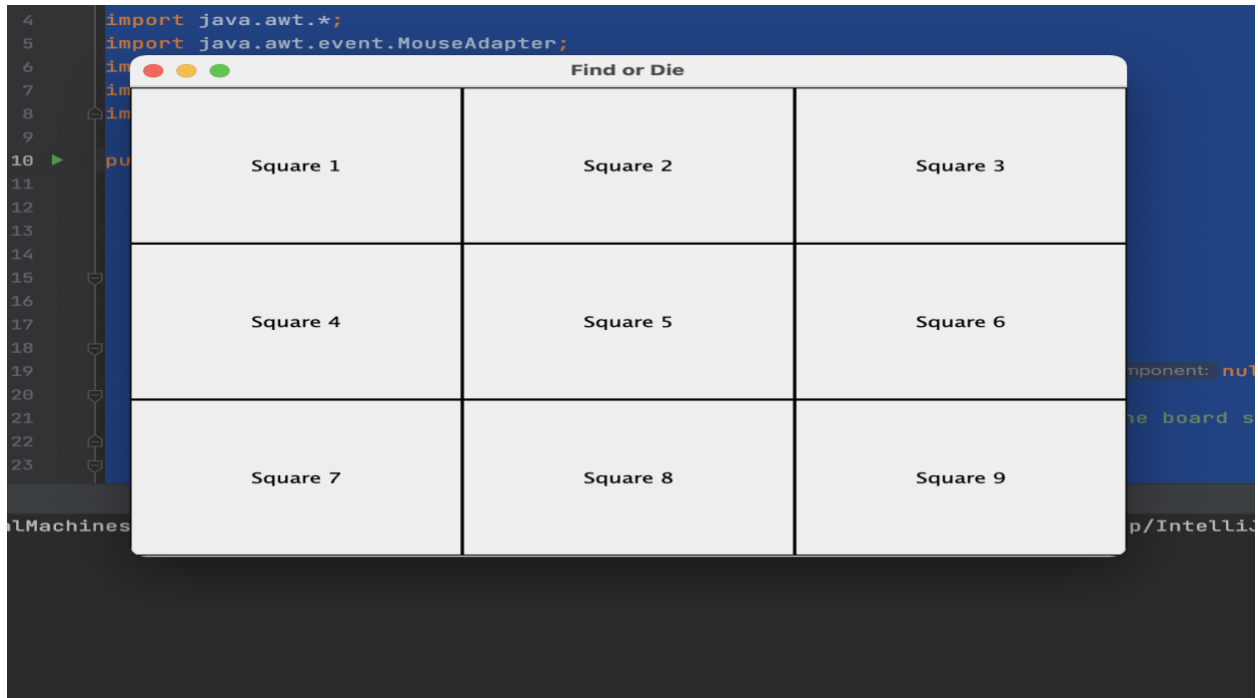
```

Output:

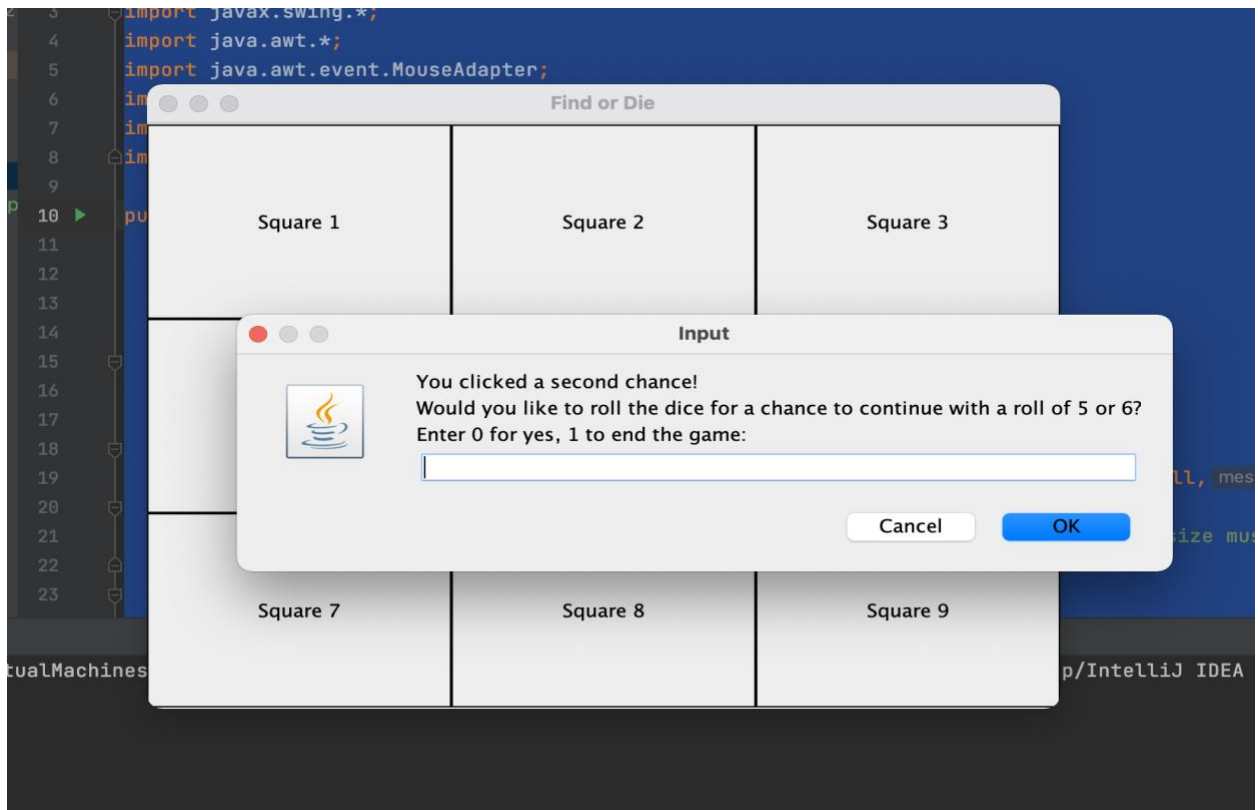
1 . Set Board Size



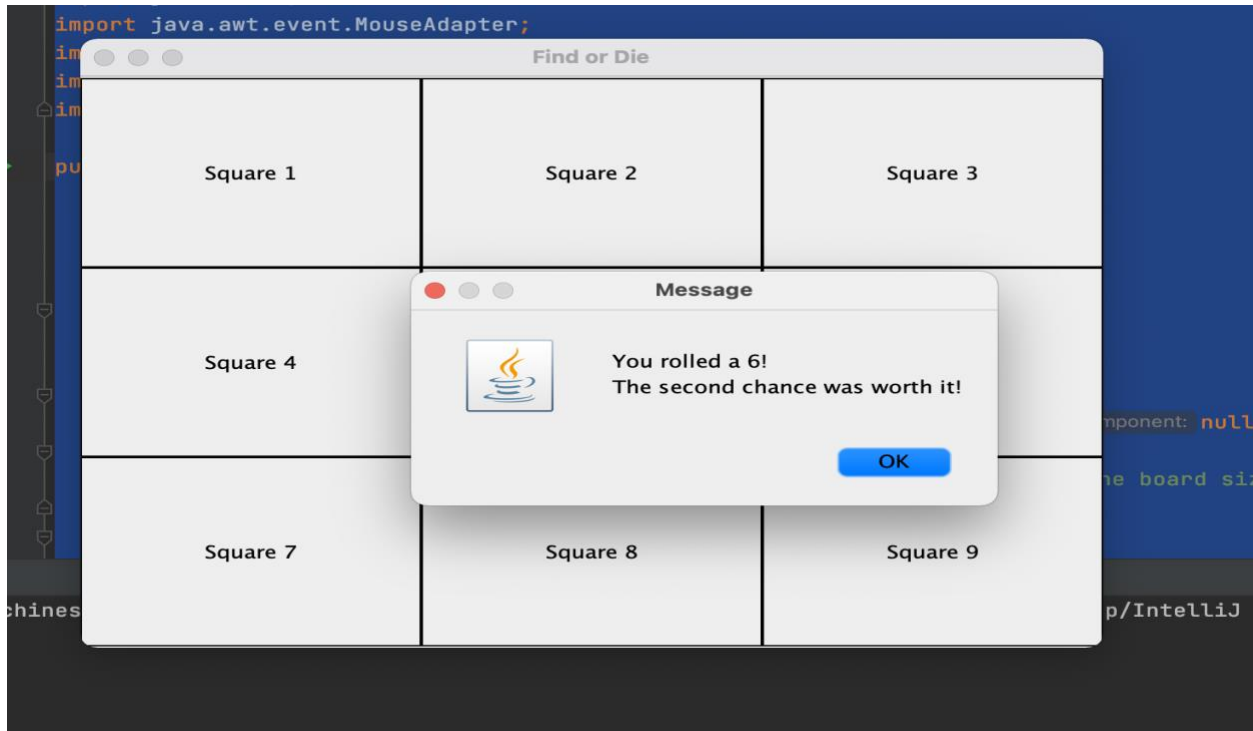
2. Base Board



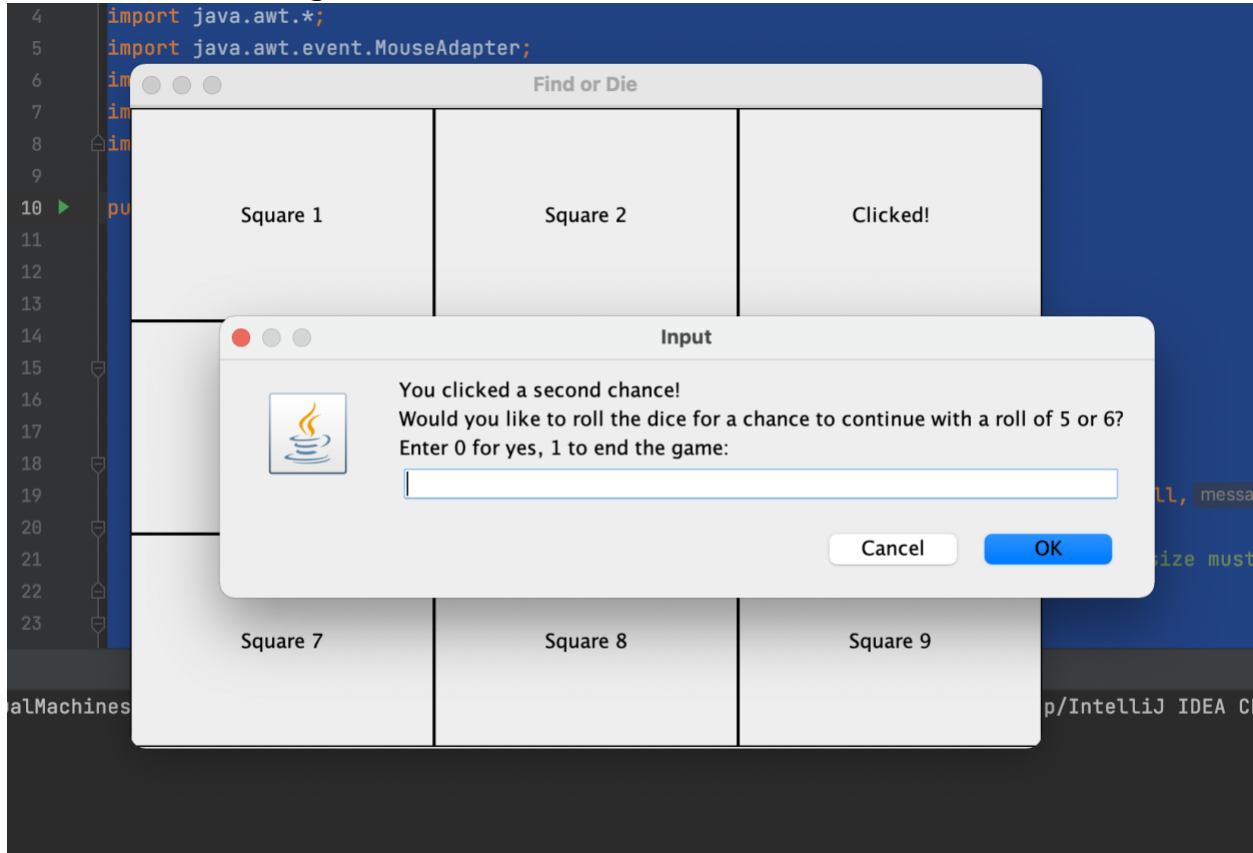
3. Second Chance Selected



4. Second Chance Successful



5. Second Chance Again



6. Second Chance Unsuccessful. Game over!

