

```

public class Game {

    public Game(){

    }

    //Playground erstellen und Schiffliste erstellen
    private List<Ship> shipList = new ArrayList<Ship>();
    private Playground playground = new Playground();

    public void intialGUI(){
        JFrame frame = new JFrame("Battleships");
        frame.setSize(2000,1000);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(new GridBagLayout());
        GridBagConstraints constraints = new GridBagConstraints();
        constraints.insets = new Insets(11,11,11,11);
        constraints.gridwidth = 1;
        constraints.gridheight = 1;
        for(int i = 0; i < 10; i++){
            constraints.gridx = i+1;
            constraints.gridy = 0;
            char letter = (char) (65+i);
            JLabel letterLabel = new JLabel(String.valueOf(letter));
            frame.getContentPane().add(letterLabel,constraints);
            constraints.gridx = 0;
            constraints.gridy = i+1;
            JLabel numberLabel = new JLabel(String.valueOf(i+1));
            frame.getContentPane().add(numberLabel,constraints);
            for (int j = 0; j < 10; j++){
                constraints.gridx = i+1;
                constraints.gridy = j+1;
                //Hier wird der Playground in einen Frame eingefügt
            }
            frame.getContentPane().add(playground.getPlayground()[i][j], constraints);
        }
        frame.setVisible(true);
    }

    public void placeShip(int x, int y){
        playground.getPlayground()[x][y].setBackground(Color.BLACK);
    }

    public void game(){
        for (int i = 0; i < playground.getPlayground().length; i++) {
            for (int j = 0; j < playground.getPlayground()[i].length; j++)
            {
                int finalI = i;
                int finalJ = j;
                playground.getPlayground()[i][j].addMouseListener(new
MouseAdapter() {
                    @Override
                    public void mouseClicked(MouseEvent e) {
                        placeShip(finalI, finalJ);
                    }
                });
            }
        }
    }
}

```

```

public class Ship {

    private boolean status;
    private int size;

    public Ship(boolean status, int size) {
        this.status = status;
        this.size = size;
    }
    public boolean isStatus() {
        return status;
    }
    public void setStatus(boolean status) {
        this.status = status;
    }
    public int getSize() {
        return size;
    }
    public void setSize(int size) {
        this.size = size;
    }
}

```

```

public class FieldButton extends JButton {
    public FieldButton() {
        super();
        this.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {

                System.out.println(getX());
                System.out.println(getY());
            }
        });
    }
}

```

```

public class MainMenu {
    public static void main(String[] args) {

        Game game = new Game();
        game.intialGUI();
        game.game();
    }
}

```