```java
package com.example.template;

import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.fxml.FXML;
import javafx.scene.control.*;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
import javafx.scene.layout.GridPane;
import javafx.scene.text.Font;

import javax.swing.JOptionPane;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.PrintWriter;
import java.lang.reflect.Array;
import java.util.ArrayList;
import java.util.Objects;
import java.util.Scanner;

public class HelloController {
    @FXML
    public GridPane mainGridPane;
    @FXML
    public Button startButton;
    @FXML
    public Label attacksLeftLabel;
    @FXML
    public Label MainDisplayLabel;
    @FXML
    public Button turnDisplayButton;
    @FXML
    public Label currentTurnLabel;
    @FXML
    public Label creditsLabel;
    @FXML
    public CheckBox possibleOptionsCheckBox;


    //    @FXML
//    public ImageView tempImageView;
    Button[][] buttonArray = new Button[10][15];
    Territory[][] territoryArray = new Territory[10][15];
    private int numPlayers;
    private int turn;
    private ArrayList<Player> allPlayers = new ArrayList<>();
    private boolean editingMode = false;
    private int attacksLeft = 3;

    public void start() {
        mainGridPane.setVisible(true);
        startButton.setVisible(false);
        startButton.setDisable(true);
//        try{
//            FileInputStream input = new FileInputStream("src/main/resources/images/worldMap.jpg");
//            tempImageView.setImage(new Image(input));
//        }catch (FileNotFoundException variable){
//            System.out.println("file not found");
//        }
        for(int row =0;row<buttonArray.length;row++) {
            for(int column=0;column<buttonArray[0].length;column++) {
                territoryArray[row][column] = new Territory(row,column, new Player("-fx-background-color: LightBlue;"));
                buttonArray[row][column] = new Button("");
                buttonArray[row][column].setPrefHeight(100);
                buttonArray[row][column].setPrefWidth(100);
//                Font font = new Font(0);
//                buttonArray[row][column].setFont(font);
                buttonArray[row][column].setStyle("-
fx-background-radius: 0px; -fx-border-color: LightBlue; -fx-background-color: LightBlue;");
                mainGridPane.add(buttonArray[row][column],column,row);
            }
        }
        askForAmountOfPlayers();
        setUpPlayers();
        turn = (int) (Math.random()*numPlayers);
        playTurn();
        //printWorldBorders();
        EventHandler z = new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                int row = GridPane.getRowIndex((Button) event.getSource());
                int column = GridPane.getColumnIndex((Button) event.getSource());
            }
        };
        for(int row =0;row<buttonArray.length;row++) {
            for(int column=0;column<buttonArray[0].length;column++) {
                buttonArray[row][column].setOnAction(z);
            }
        }
    }

    public void playTurn() {
        setUpEditingMode();
    }

    public void setUpEditingMode() {
        MainDisplayLabel.setText("Editing Mode. Click on your territories to upgrade them.");
        currentTurnLabel.setText("Current Turn:");
        turnDisplayButton.setDisable(false);
        turnDisplayButton.setVisible(true);
        editingMode = true;
        turnDisplayButton.setStyle("-fx-background-radius: 0px;");
        turnDisplayButton.setStyle(allPlayers.get(turn%numPlayers).getColor());
    }

    public void onClickEndEditingMode() {
        setUpAttackingMode();
    }

    public void setUpAttackingMode() {
        editingMode = false;
        attacksLeft = 3;
        MainDisplayLabel.setText("Attack Mode. Click On The Surrounding Territories To Invade.");
        currentTurnLabel.setText("Current Turn:");
        turnDisplayButton.setDisable(false);
        turnDisplayButton.setVisible(true);
        turnDisplayButton.setStyle("-fx-background-radius: 0px;");
        turnDisplayButton.setStyle(allPlayers.get(turn%numPlayers).getColor());
        attacksLeftLabel.setText("Attacks Left: "+attacksLeft);
    }


    public void setUpPlayers() {
        for (int i =0;i<numPlayers;i++) {
            if (i==0) {
                allPlayers.add(new Player("temp",1,new Territory(0,0), "-fx-background-color: Blue;"));
                territoryArray[0][0].setControllingPlayer(allPlayers.get(allPlayers.size()-1));
            }
            else if (i==1) {
                allPlayers.add(new Player("temp",1,new Territory(0,14), "-fx-background-color: Red;"));
                territoryArray[0][14].
```

```java
setControllingPlayer(allPlayers.get(allPlayers.size()-1));
            }
            else if (i==2) {
                allPlayers.add(new Player("temp",1, new Territory(9,0), "-fx-background-color: Yellow;"));
                territoryArray[9][0].setControllingPlayer(allPlayers.get(allPlayers.size()-1));
            }
            else {
                allPlayers.add(new Player("temp",1, new Territory(9,14), "-fx-background-color: Green;"));
                territoryArray[9][14].setControllingPlayer(allPlayers.get(allPlayers.size()-1));
            }
        }
        updateMap();
    }

    public void updateMap() {
        for(int row =0;row<territoryArray.length;row++) {
            for(int column=0;column<territoryArray[0].length;column++) {
                buttonArray[row][column].setStyle(territoryArray[row][column].getControllingPlayer().getColor());
            }
        }
    }

    public void askForAmountOfPlayers() {
        boolean enteredProperly = false;
        while (!enteredProperly) {
            numPlayers = Integer.parseInt(JOptionPane.showInputDialog("how many people will be playing (2-4)"));
            if (numPlayers>1 && numPlayers<5) {
                enteredProperly = true;
            }
        }
    }

    public void printWorldBorders() {
        try{
            FileReader reader = new FileReader("src/main/resources/tempCoordinates.txt");
            Scanner in = new Scanner(reader);
            ArrayList<String> allCords = new ArrayList<>();
            while (in.hasNext()) {
                allCords.add(in.nextLine());
            }
            for (int i =0;i<allCords.size();i++) {
                int spot = 0;
                StringBuilder num = new StringBuilder();
                while (allCords.get(i).charAt(spot) != ',') {
                    num.append(allCords.get(i).charAt(spot));
                    spot++;
                }
                StringBuilder num1 = new StringBuilder();
                for (int j = spot+1;j<allCords.get(i).length();j++) {
                    num1.append(allCords.get(i).charAt(j));
                }
                buttonArray[Integer.parseInt(String.valueOf(num))][Integer.parseInt(String.valueOf(num1))].setStyle("-fx-border-color: Blue; -fx-background-color: Blue;");
            }
        }catch (FileNotFoundException variable){
            System.out.println("file not found");
        }
```

```java
    }

    public void addCoordinatesToTXT(int rowOfClick, int columnOfClick) {
        String outFile = "src/main/resources/tempCoordinates.txt";
        try{
            FileReader reader = new FileReader("src/main/resources/tempCoordinates.txt");
            Scanner in = new Scanner(reader);
            ArrayList<String> allCords = new ArrayList<>();
            while (in.hasNext()) {
                allCords.add(in.nextLine());
            }
            PrintWriter out = new PrintWriter(outFile);
            for (int i = 0;i<allCords.size();i++) {
                out.println(allCords.get(i));
            }
            out.println(rowOfClick +","+ columnOfClick);
            out.close();
        }catch (FileNotFoundException variable){
            System.out.println("file not found");
        }
    }
}
```