/\*

Daiwik Swaminathan

5.18.2018

WorldWar3.java

Period 6

\*/

//Imports

import java.awt.\*;import javax.swing.\*;import java.util.\*;import java.awt.event.\*;

import java.io.\*;

import java.applet.Applet; import java.applet.AudioClip; import java.net.URL;

import javax.swing.event.MenuEvent;import java.text.DateFormat;

import javax.swing.event.MenuListener;

import javax.swing.Timer;import java.util.Date;

import java.util.Calendar;import java.text.SimpleDateFormat;

import java.awt.geom.AffineTransform;

//Main class, has main method to launch the game

public class WorldWar3

{

JFrame frame;

int screenHeight, screenWidth;//The width and height of the computer screen

AudioClip clip, explosionSound;//Needed for sound

Background background;

//Main method, creates instance of the class which then

//calls the constructer method and calls run()

public static void main(String[] args)

{

WorldWar3 prog = new WorldWar3();

prog.run();

}

//Constructer method, initializes screenWidth and screenHeight

public WorldWar3()

{

//These are going to be the dimensions of the frame

screenWidth = 1366;//1366 or 1350

//^Gets the computer screen width^

screenHeight = 768;//768 or 740

//^Gets the computer screen height^

//System.out.println(screenWidth + " " +screenHeight);

}

//Run method to create frame, create background class and

//call method to play background music

public void run()

{

frame = new JFrame("World War 3");

frame.setLocation(0, 0);

frame.setSize(screenWidth, screenHeight);

frame.setResizable(true);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setFocusable(true);//Setting focusable so nothing is selected at the start

//Creating the background or main panel(Line of code below)

background = new Background(screenWidth, screenHeight);

frame.add(background);//Frame adds panel

frame.setVisible(true);

playBG();//Method to play background music

}

//Method is called to start the game over when the player wins or loses the game

public void remake()

{

frame.remove(background);

background = new Background(screenWidth, screenHeight);

frame.add(background);

frame.revalidate();

frame.repaint();

}

//Method which plays background music

public void playBG()

{

URL url = WorldWar3.class.getResource("music.wav");

clip = Applet.newAudioClip(url);

clip.loop();

URL url2 = WorldWar3.class.getResource("explosion.wav");

explosionSound = Applet.newAudioClip(url2);

}

//Background class or main panel of game

//Kept as inner class so variables from WorldWar3 class can be accessed here

class Background extends JPanel implements ActionListener, MenuListener, KeyListener

{

//Field variables to be accessed anywhere in class

Image background, conqueredFlag, notConqueredFlag;//Image of the background

Image plane, boat, aBomb, nBomb, aBombPlane, explosion;

Image endGameWin, instructionsImage, instructionsImage2, moneyLoseImage, peopleLoseImage, healthLoseImage,troopLoseImage;

int width, height;//Variables to get the width and height of screen

int gameOverCount;

//JButtons

JButton play, instructions, records, credits, settings, newGame, loadGame;

JButton easy, moderate, hard, next, sound, back, begin, continueButton;

JButton backFromGameSelectPanel, backFromGameOptionsPanel, soundEffects, music;

//JPanels that contain different buttons

JPanel startingGamePanel, gameSelectPanel, gameOptions, panelWithSettings;

JPanel panelWithStartingSpeech, panelWithEvent;

JLabel title, selectContinentLabel;//JLabel which is the title of the game

String temp="";

File file;//File to get records and if game should load previous game

Scanner scanner;//Scanner object for file io

//Booleans for whether it should disable load button and

//whether the game should be loaded or not(Line of code below)

boolean disableLoad, shouldLoad;

JPanel selectContinentPanel, fixBackPanel, fixBackPanel2;//Holds comboBox for selecting Continent and text above it

//Array for adding continents to combo box and for general use

String[] continents = {"North America","South America","Europe",

"Asia","Africa","Australia"};

JComboBox<String> selectContinentComboBox;//combo box for selecting continent

//continent will hold which continent user picks and mode is which mode is selected

String continent, mode, tempContinent, contToAttackEnemy;

JMenuBar jmb;//JMenu bar at the top with game functions

JMenu jm1, jm2, jm3, jm4, jm5, jm6, jm7, jm8, jm9, jm10, jm11;//JMenu items for JMenuBar

JLabel pauseLabel;//Indicates money and indicates the game is paused to user

//JMenuItems for the menus

JMenuItem quitAndSave, quitAndNoSave, soundFX, backMusic, resume;

JMenuItem[] startingMoneyArray = new JMenuItem[6];

JMenuItem[] continentPopulations = new JMenuItem[6];

JMenuItem[] troopPopulations = new JMenuItem[6];

JMenuItem[] warOptions = new JMenuItem[6];

int jetLevel=0, marineLevel=1, weaponLevel=1, bombLevel=0, nBombLevel=1, nationalDefenseLevel=0;

JMenuItem aircraftMenu = new JMenuItem("Level "+(jetLevel+1));

JMenuItem marineMenu = new JMenuItem("Level "+marineLevel);

//JMenuItem weaponMenu = new JMenuItem("Level "+weaponLevel);

JMenuItem aBombMenu = new JMenuItem("Level "+(bombLevel+1));

JMenuItem nBombMenu = new JMenuItem("Level "+nBombLevel);

JPanel planePanel, bombMenu, defenseMenu;

JMenuItem nationalDefense = new JMenuItem("Level "+(nationalDefenseLevel+1));

//ImageIcons for sound and pause

ImageIcon soundIcon, muteIcon, miniSoundIcon, miniMuteIcon, pauseIcon;

int whichPanel = 1;//Indicates which panel is being displayed

//Booleans whether sound effects and music should be on or not

boolean keepSFX = true, keepMusic = true;

boolean isPaused, startGame, startDayCounter;//Boolean for whether game is paused or not

int whichCont, daysElapsed;//Indicates which continent is chosen

//The starting money for each continent

//int[] startingMoney = {650, 75, 350, 450, 75, 50};

long[] startingMoney = {500000000, 75000000, 350000000, 450000000, 75000000, 250000000};

//The starting populations for each continent

//double[] populations = {600, 450, 800, 4.4, 1.4, 50};

long[] populations = {600000000, 450000000, 800000000, 4400000000l, 1400000000, 50000000};

//int[] startingTroops = {5, 3, 7, 10, 3, 1};4400000000

long[] startingTroops = {5000000, 3000000, 7000000, 10000000, 3000000, 1000000};

JTextArea startingSpeechPanel, eventPanel;

int testNumber, day, month, year;

Timer timer, timerForPlane, timerForABombs, timerForNukeEnemy, timerForPlaneEnemy;

String dateString;

int[] daysOfMonth = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

PrintWriter pw;

File outFile;

int nukesToSendEnemy, planesToSendEnemy;

boolean detectingOptions;

boolean foundFile, showFlag;

int countForDW;

int whichSubEvent, daysSinceLastEvent, daysToNextEvent;

//int[] flagLocations = {1};//Goes in normal continent order

String[] naturalDisasters = {"Tsunami", "Earthquake", "Volcano Eruption", "Tornado", "Flooding", "Disease", "Nuclear Bomb", "Atomic Bomb", "Declare War", "National Defense", "Threat", "EconomyB", "SendPlanes"};

String[][] eventArray = {naturalDisasters};

String event;

String[] tsunamiEvent = {"A devastating tsunami strikes ", "Collateral damage amounts to $", " in damage and ", " people have been killed."};

String[] earthquakeEvent = {"A massive earthquake devastates ", "Collateral damage amounts to $", " in damage and ", " people have been killed."};

String[] volcanoEvent = {"An unexpected volcano eruption thrashes ", "Collateral damage amounts to $", " in damage and ", " people have been killed."};

String[] tornadoEvent = {"An deadly tornado tears through ", "Collateral damage amounts to $", " in damage and ", " people have been killed."};

String[] floodingEvent = {"Severe flooding surprises ", "Collateral damage amounts to $", " in damage and ", " people have been killed."};

String[] diseaseEvent = {"A new serious disease is spreading in ", "$", " has been directed to research and ", " people have been killed."};

String[] nuclearBombEvent = {"", " nukes ", "Collateral damage amounts to $", " in damage and ", " people have been killed."};

String[] atomicBombEvent = {"", " detonates an atomic bomb in ", "Collateral damage amounts to $", " in damage and ", " people have been killed."};

String[] declareWarEvent = {" declares war on "};

String[] nationalDefenseEvent = {" has raised their national defense. Conquering ", " will now be much harder."};

String[] threatEvent = {" issues a threat to ", ". Rumors are spreading about what this could lead to..."};

String[] economyBoostEvent = {"'s economy sees in increase in production and general welfare as a result. ", " is added for military use."};

String[] allyEvent = {" allies with ", ". No fighting will take place between these two."};

boolean[] healthBars;

int numberOfNewsAllowed = 10;

int[] healthReduce, timesWarOptionsClicked;

boolean[][] checkLogicRecords;

JPanel[] warStatusPanels;

JLabel[] continentImages = new JLabel[6], aircraftLabels = new JLabel[6], marineLabels = new JLabel[6], troopCostLabels = new JLabel[6], moneyTotalLabels = new JLabel[6], aBombLabels = new JLabel[6], nBombLabels = new JLabel[6];

JButton[] confirm = new JButton[6];

JButton[] cancel = new JButton[6];

JTextField[] aircraftBoxes = new JTextField[6], marineBoxes = new JTextField[6], troopCostBoxes = new JTextField[6], moneyTotalBoxes = new JTextField[6], aBombBoxes = new JTextField[6], nBombBoxes = new JTextField[6];

JPanel[] leftSide = new JPanel[6];

JPanel[] rightSide = new JPanel[6];

JPanel[] eventLogs = new JPanel[numberOfNewsAllowed];

JMenuItem[] eventList = new JMenuItem[numberOfNewsAllowed];

int eventCount;

int[] currentPlanes = new int[6];

int[] currentBoats = new int[6];

int[] currentABombs = new int[6];

int[] currentNBombs = new int[6];

int planesToSend, aBombsToSend, nBombsToSend;

long[] currentTroopCost = new long[6];

long[] currentTotalCost = new long[6];

int planeSpace = 5000, boatSpace = 10000, aBombSpace = 500, nBombSpace = 1000;

JButton[] contButs = new JButton[numberOfNewsAllowed];

JTextArea[] logTexts = new JTextArea[numberOfNewsAllowed];

boolean shift;

int planeX=350, planeY=200;

String contToAttack;

int attackingCont;//use this

boolean sendThePlane, initializedGraphics, sendTheABomb, sendTheNBomb, sendTheNukeEnemy, sendThePlaneEnemy;

int tempX = 350;

JPanel holdPlaneUpgrade, holdBombUpgrade, holdDefenseUpgrade;

int[] planeXSpots, aBombXSpots, nukeXSpotsEnemy, planeXSpotsEnemy;

Timer[] planeTimers;

boolean firstTime=true;

boolean[] showPlanes;

boolean[] showABombs;

boolean[] showNBombs;

boolean[] dropABombs, dropNukesEnemy;

boolean[] showNukesEnemy;

boolean[] showPlanesEnemy;

Graphics2D g2d;

AffineTransform old;

int checkTimes;

double[] defenseCount = {0.2,0.0,0.1,0.1,0.0,0.1};

boolean timesChangedCont;

int contCount;

int[] aBombSizes, nukeSizesEnemy;

int size = 35;

boolean[] shouldABombExplode, shouldNukeExplode;

boolean setLater;

JLabel moneyForDefense, moneyForPlane, moneyForBomb;

JButton confirmDefense = new JButton("Confirm");

JButton confirmPlane = new JButton("Confirm");

JButton confirmBomb = new JButton("Confirm");

JButton cancelDefense;

JButton cancelPlane;

JButton cancelBomb;

boolean[][] canAdvance = new boolean[4][6];

boolean enterWar=true;

boolean[][] warCheck = new boolean[6][6];

boolean postIt;

boolean firstTimeEnemy=true;

boolean[][] allies = new boolean[6][6];

boolean[] makeBlue = new boolean[6];

boolean[] canUpdateInfo = new boolean[6];

boolean endGame;

JButton home, backFromFirstInstructions, goToSecondInstructions, backFromInstructions2;

boolean putInstructions, putInstructions2;

JLabel score;

boolean moneyGO, peopleGO, troopGO, healthGO, won;

int finalScore;

int[] bombSoundCounter, bombSoundCounterEnemy;

int xStopSpot, xStopSpotEnemy;

boolean[] showExplosion, showExplosionEnemy;

String[] testLeaderboard = {"A", "B", "C", "D"};

String[] leaderboardResults;

//JList<String> leaderboard;

JPanel leaderBoardPanel;

JButton backFromLeaderboard;

JPanel holdTheBack;

boolean resultsPresent;

boolean createFile = true;

String nameOfFile = "Test518.txt";

JTextField enterName;

JLabel easyOnes, moderateOnes, hardOnes;

//Main panel constructer to create

//all the components and panels and create ImageIcons used in the game

public Background(int widthInput, int heightInput)

{

easyOnes = new JLabel("Easy: North America and Asia", JLabel.CENTER);

easyOnes.setFont(new Font("Courier", Font.PLAIN, 15));

moderateOnes = new JLabel("Moderate: Europe and Australia", JLabel.CENTER);

moderateOnes.setFont(new Font("Courier", Font.PLAIN, 15));

hardOnes = new JLabel("Hard: South America and Africa", JLabel.CENTER);

hardOnes.setFont(new Font("Courier", Font.PLAIN, 15));

enterName = new JTextField("Enter Name Here");

enterName.setFont(new Font("Courier", Font.PLAIN, 20));

enterName.setEditable(true);

holdTheBack = new JPanel();

holdTheBack.setLayout(new FlowLayout(FlowLayout.LEFT));

holdTheBack.setOpaque(false);

backFromLeaderboard = new JButton("Back");

backFromLeaderboard.setFont(new Font("Courier", Font.PLAIN, 20));

backFromLeaderboard.addActionListener(this);

backFromLeaderboard.setPreferredSize(new Dimension(150, 50));

holdTheBack.add(backFromLeaderboard);

//leaderboard = new JList<>(leaderboardResults);

// leaderboard.setVisibleRowCount(4);

//leaderboard.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);

leaderBoardPanel = new JPanel();

leaderBoardPanel.setLayout(new GridLayout(3,3));

leaderBoardPanel.setOpaque(false);

for(int i=0; i<9; i++)

{

if(i==0) leaderBoardPanel.add(holdTheBack);

// else if(i==4) leaderBoardPanel.add(new JScrollPane(leaderboard));

else leaderBoardPanel.add(new JLabel(""));

}

score = new JLabel("Score: "+finalScore, JLabel.CENTER);

score.setFont(new Font("Courier", Font.PLAIN, 40));

score.setForeground(Color.RED);

frame.addKeyListener(this);

backFromFirstInstructions = new JButton("Back");

backFromFirstInstructions.setFont(new Font("Courier", Font.PLAIN, 20));

backFromFirstInstructions.addActionListener(this);

backFromInstructions2 = new JButton("Back");

backFromInstructions2.setFont(new Font("Courier", Font.PLAIN, 20));

backFromInstructions2.addActionListener(this);

goToSecondInstructions = new JButton("Next");

goToSecondInstructions.setFont(new Font("Courier", Font.PLAIN, 20));

goToSecondInstructions.addActionListener(this);

endGameWin = new ImageIcon("winTheGameScreen.png").getImage();

home = new JButton("Home");

home.setFont(new Font("Courier", Font.PLAIN, 20));

home.addActionListener(this);

for(int i=0; i<contButs.length; i++)

{

contButs[i] = new JButton("Continue");

contButs[i].setFont(new Font("Courier", Font.PLAIN, 20));

contButs[i].addActionListener(this);

logTexts[i] = new JTextArea();

logTexts[i].setFont(new Font("Courier", Font.PLAIN, 20));

logTexts[i].setLineWrap(true);

logTexts[i].setWrapStyleWord(true);

logTexts[i].setEditable(false);

}

warStatusPanels = new JPanel[6];

fixBackPanel = new JPanel();

fixBackPanel.setLayout(new FlowLayout(FlowLayout.LEFT));

fixBackPanel.setOpaque(false);

fixBackPanel2 = new JPanel();

fixBackPanel2.setLayout(new FlowLayout(FlowLayout.LEFT));

fixBackPanel2.setOpaque(false);

for(int i=0; i<warStatusPanels.length; i++){ warStatusPanels[i] = new JPanel(); warStatusPanels[i].setBorder(BorderFactory.createLineBorder(Color.BLACK, 3));}

for(int i=0; i<6; i++)

{

rightSide[i] = new JPanel();

leftSide[i] = new JPanel();

continentImages[i] = new JLabel("", JLabel.CENTER);

continentImages[i].setFont(new Font("Courier", Font.PLAIN, 15));

aircraftLabels[i] = new JLabel("Planes: ");

aircraftLabels[i].setFont(new Font("Courier", Font.PLAIN, 15));

marineLabels[i] = new JLabel("Submarines: ");

marineLabels[i].setFont(new Font("Courier", Font.PLAIN, 15));

troopCostLabels[i] = new JLabel("Troop Capacity: ");

troopCostLabels[i].setFont(new Font("Courier", Font.PLAIN, 10));

moneyTotalLabels[i] = new JLabel("Total Cost: ");

moneyTotalLabels[i].setFont(new Font("Courier", Font.PLAIN, 15));

aBombLabels[i] = new JLabel("Nukes: ");

aBombLabels[i].setFont(new Font("Courier", Font.PLAIN, 15));

nBombLabels[i] = new JLabel("Nuclear Bombs: ");

nBombLabels[i].setFont(new Font("Courier", Font.PLAIN, 12));

aircraftBoxes[i] = new JTextField();

aircraftBoxes[i].setFont(new Font("Courier", Font.PLAIN, 15));

aircraftBoxes[i].addKeyListener(this);

marineBoxes[i] = new JTextField();

marineBoxes[i].setFont(new Font("Courier", Font.PLAIN, 15));

marineBoxes[i].addKeyListener(this);

troopCostBoxes[i] = new JTextField();

troopCostBoxes[i].setFont(new Font("Courier", Font.PLAIN, 15));

troopCostBoxes[i].setEditable(false);

moneyTotalBoxes[i] = new JTextField();

moneyTotalBoxes[i].setFont(new Font("Courier", Font.PLAIN, 15));

moneyTotalBoxes[i].setEditable(false);

aBombBoxes[i] = new JTextField();

aBombBoxes[i].setFont(new Font("Courier", Font.PLAIN, 15));

aBombBoxes[i].addKeyListener(this);

nBombBoxes[i] = new JTextField();

nBombBoxes[i].setFont(new Font("Courier", Font.PLAIN, 15));

nBombBoxes[i].addKeyListener(this);

cancel[i] = new JButton("Cancel");

cancel[i].setFont(new Font("Courier", Font.PLAIN, 20));

cancel[i].addActionListener(this);

confirm[i] = new JButton("Confirm");

confirm[i].setFont(new Font("Courier", Font.PLAIN, 20));

confirm[i].addActionListener(this);

rightSide[i].setLayout(new GridLayout(5,2));//7 2

for(int j=0; j<leftSide.length; j++)

{

rightSide[i].add(aircraftLabels[i]);

rightSide[i].add(aircraftBoxes[i]);

//rightSide[i].add(marineLabels[i]);

//rightSide[i].add(marineBoxes[i]);

rightSide[i].add(aBombLabels[i]);

rightSide[i].add(aBombBoxes[i]);

//rightSide[i].add(nBombLabels[i]);

//rightSide[i].add(nBombBoxes[i]);

rightSide[i].add(troopCostLabels[i]);

rightSide[i].add(troopCostBoxes[i]);

rightSide[i].add(moneyTotalLabels[i]);

rightSide[i].add(moneyTotalBoxes[i]);

rightSide[i].add(cancel[i]);

rightSide[i].add(confirm[i]);

ImageIcon iconName = new ImageIcon(continents[i]+".png");

continentImages[i].setIcon(iconName);

leftSide[i].setLayout(new GridLayout(1,1));

leftSide[i].add(continentImages[i]);

warStatusPanels[i].setLayout(new GridLayout(1,2));

warStatusPanels[i].add(leftSide[i]);

warStatusPanels[i].add(rightSide[i]);

}

}

cancelDefense = new JButton("Cancel");

cancelPlane = new JButton("Cancel");

cancelBomb = new JButton("Cancel");

timesWarOptionsClicked = new int[6];

checkLogicRecords = new boolean[6][6];

healthReduce = new int[6];

for(int i=0; i<6; i++) if(i!=whichCont)healthReduce[i] = -90;

healthBars = new boolean[6];

Date today = Calendar.getInstance().getTime();

SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd-hh.mm.ss");

String folderName = formatter.format(today);

dateString = folderName.substring(0, 10);

getDateInfo();

setLayout(new GridLayout(3,3));//Sets grid layout 3 by 3

background = new ImageIcon("bg.png").getImage();//Creates background image

instructionsImage = new ImageIcon("InstructionsPanel.png").getImage();

instructionsImage2 = new ImageIcon("InstructionsPanels2.png").getImage();

conqueredFlag = new ImageIcon("blueFlag.png").getImage();

notConqueredFlag = new ImageIcon("redFlag.png").getImage();

plane = new ImageIcon("fighterJet.png").getImage();

aBombPlane = new ImageIcon("aBombPlane.png").getImage();

boat = new ImageIcon("submarine.png").getImage();

aBomb = new ImageIcon("atomicBomb.png").getImage();

nBomb = new ImageIcon("NuclearBomb.png").getImage();

explosion = new ImageIcon("explosionActual.png").getImage();

moneyLoseImage = new ImageIcon("MoneyLoseScreen.png").getImage();//CHANGABLE

troopLoseImage = new ImageIcon("TroopLoseScreen.png").getImage();//CHANGABLE

peopleLoseImage = new ImageIcon("PeopleLoseScreen.png").getImage();//CHANGABLE

healthLoseImage = new ImageIcon("HealthLoseScreen.png").getImage();//CHANGABLE

//imageString = aBomb.getWidth(null)+":"+aBomb.getHeight(null);

//System.out.println("blah: " + imageString);

width = widthInput;//Gets width of the screen from Main class

height = heightInput;//Gets height of the screen from Main class

play = new JButton("Play");

play.setFont(new Font("Courier", Font.PLAIN, 25));

play.addActionListener(this);

records = new JButton("Leaderboard");

records.setFont(new Font("Courier", Font.PLAIN, 25));

records.addActionListener(this);

credits = new JButton("Credits");

credits.setFont(new Font("Courier", Font.PLAIN, 25));

credits.addActionListener(this);

next = new JButton("Next");

next.setFont(new Font("Courier", Font.PLAIN, 25));

next.addActionListener(this);

instructions = new JButton("Instructions");

instructions.setFont(new Font("Courier", Font.PLAIN, 25));

instructions.addActionListener(this);

settings = new JButton("Settings");

settings.setFont(new Font("Courier", Font.PLAIN, 25));

settings.addActionListener(this);

title = new JLabel("World War 3");

title.setFont(new Font("Courier", Font.BOLD, 50));

title.setForeground(Color.RED);

title.setHorizontalAlignment(JLabel.CENTER);

startingGamePanel = new JPanel();//Creates panel with the four starting buttons

startingGamePanel.setLayout(new GridLayout(3, 1));//Sets 3 by 1 grid layout

startingGamePanel.setOpaque(false);

startingGamePanel.add(play);//Adds all the buttons to startingGamePanel

//startingGamePanel.add(records);

// startingGamePanel.add(credits);

startingGamePanel.add(instructions);

startingGamePanel.add(settings);

gameSelectPanel = new JPanel();

gameSelectPanel.setLayout(new GridLayout(5,5));//Sets 5 by 5 grid layout

gameSelectPanel.setOpaque(false);

newGame = new JButton("New Game");

newGame.setFont(new Font("Courier", Font.PLAIN, 25));

newGame.addActionListener(this);

loadGame = new JButton("Load Game");

loadGame.setFont(new Font("Courier", Font.PLAIN, 25));

loadGame.addActionListener(this);

backFromGameSelectPanel = new JButton("Back");

backFromGameSelectPanel.setFont(new Font("Courier", Font.PLAIN, 25));

backFromGameSelectPanel.addActionListener(this);

backFromGameOptionsPanel = new JButton("Back");

backFromGameOptionsPanel.setFont(new Font("Courier", Font.PLAIN, 25));

backFromGameOptionsPanel.addActionListener(this);

backFromGameOptionsPanel.setPreferredSize(new Dimension(150, 50));

backFromGameSelectPanel.setPreferredSize(new Dimension(150, 50));

fixBackPanel.add(backFromGameSelectPanel);

fixBackPanel2.add(backFromGameOptionsPanel);

soundEffects = new JButton("Sound Effects");

soundEffects.setFont(new Font("Courier", Font.PLAIN, 25));

soundEffects.setIcon(new ImageIcon("sound.png"));

soundEffects.setHorizontalTextPosition(SwingConstants.LEFT);

soundEffects.addActionListener(this);

music = new JButton("Music");

music.setFont(new Font("Courier", Font.PLAIN, 25));

music.setIcon(new ImageIcon("sound.png"));

music.setHorizontalTextPosition(SwingConstants.LEFT);

music.addActionListener(this);

holdPlaneUpgrade = new JPanel();

holdPlaneUpgrade.setLayout(new GridLayout(1,2));

holdBombUpgrade = new JPanel();

holdBombUpgrade.setLayout(new GridLayout(1,2));

holdDefenseUpgrade = new JPanel();

holdDefenseUpgrade.setLayout(new GridLayout(1,2));

planePanel = new JPanel();

planePanel.setLayout(new GridLayout(2,1));// moneyForPlane, moneyForBomb;

planePanel.setBorder(BorderFactory.createLineBorder(Color.BLACK, 3));

bombMenu = new JPanel();

bombMenu.setLayout(new GridLayout(2,1));

bombMenu.setBorder(BorderFactory.createLineBorder(Color.BLACK, 3));

defenseMenu = new JPanel();

defenseMenu.setLayout(new GridLayout(2,1));

defenseMenu.setBorder(BorderFactory.createLineBorder(Color.BLACK, 3));

confirmDefense.setFont(new Font("Courier", Font.PLAIN, 20));

confirmDefense.addActionListener(this);

confirmPlane.setFont(new Font("Courier", Font.PLAIN, 20));

confirmPlane.addActionListener(this);

confirmBomb.setFont(new Font("Courier", Font.PLAIN, 20));

confirmBomb.addActionListener(this);

cancelDefense.setFont(new Font("Courier", Font.PLAIN, 20));

cancelDefense.addActionListener(this);

cancelPlane.setFont(new Font("Courier", Font.PLAIN, 20));

cancelPlane.addActionListener(this);

cancelBomb.setFont(new Font("Courier", Font.PLAIN, 20));

cancelBomb.addActionListener(this);

moneyForDefense = new JLabel(""+(nationalDefenseLevel+1)\*10000000, JLabel.CENTER);

moneyForDefense.setIcon(new ImageIcon("money.jpg"));

moneyForDefense.setFont(new Font("Courier", Font.PLAIN, 20));

moneyForPlane = new JLabel(""+(jetLevel+1)\*10000000, JLabel.CENTER);

moneyForPlane.setIcon(new ImageIcon("money.jpg"));

moneyForPlane.setFont(new Font("Courier", Font.PLAIN, 20));

moneyForBomb = new JLabel(""+(bombLevel+1)\*10000000, JLabel.CENTER);

moneyForBomb.setIcon(new ImageIcon("money.jpg"));

moneyForBomb.setFont(new Font("Courier", Font.PLAIN, 20));

holdPlaneUpgrade.add(cancelPlane);

holdPlaneUpgrade.add(confirmPlane);

holdBombUpgrade.add(cancelBomb);

holdBombUpgrade.add(confirmBomb);

holdDefenseUpgrade.add(cancelDefense);

holdDefenseUpgrade.add(confirmDefense);

defenseMenu.add(moneyForDefense);

defenseMenu.add(holdDefenseUpgrade);

planePanel.add(moneyForPlane);

planePanel.add(holdPlaneUpgrade);

bombMenu.add(moneyForBomb);

bombMenu.add(holdBombUpgrade);

for(int i=0;i<25; i++)//Sets specific spots in the grid layout to have buttons

{

if(i==11) gameSelectPanel.add(newGame);

else if(i==13) gameSelectPanel.add(loadGame);

else if(i==0){gameSelectPanel.add(fixBackPanel);}

else gameSelectPanel.add(new JLabel(""));//Adds empty JLabel or space

}

gameOptions = new JPanel();

gameOptions.setLocation(0,0);

gameOptions.setSize(width, height);//Fills entire screen

gameOptions.setLayout(new GridLayout(7, 7));//Sets 7 by 7 grid layout

gameOptions.setOpaque(false);//Makes it tranparent

easy = new JButton("Easy");

easy.setFont(new Font("Courier", Font.PLAIN, 25));

easy.addActionListener(this);

moderate = new JButton("Moderate");

moderate.setFont(new Font("Courier", Font.PLAIN, 25));

moderate.addActionListener(this);

hard = new JButton("Hard");

hard.setFont(new Font("Courier", Font.PLAIN, 25));

hard.addActionListener(this);

for(int i=0; i<49; i++)//Sets specific spots in the grid layout to have buttons

{

if(i==22) gameOptions.add(easy);

else if(i==24) gameOptions.add(moderate);

else if(i==26) gameOptions.add(hard);

else if(i==0) gameOptions.add(fixBackPanel2);//Back button

else gameOptions.add(new JLabel(""));//Adds empty JLabel or space

}

for(int i=0; i<9; i++)

{

if(i==1) add(title);

else if(i==4) add(startingGamePanel);

else add(new JLabel(""));//Adds empty JLabel or space

}

back = new JButton("Back");//JButton that goes back to previous screen

back.setFont(new Font("Courier", Font.PLAIN, 25));

back.addActionListener(this);

miniSoundIcon = new ImageIcon("miniSound.png");

miniMuteIcon = new ImageIcon("miniMute.png");

pauseIcon = new ImageIcon("pause.png");

panelWithSettings = new JPanel();//Panel that will have settings in title screen

panelWithSettings.setLayout(new GridLayout(3,1));//3 by 1 grid

panelWithSettings.setOpaque(false);//Makes it transparent

panelWithSettings.add(soundEffects);

panelWithSettings.add(music);

panelWithSettings.add(back);

selectContinentPanel = new JPanel();

selectContinentPanel.setLayout(new GridLayout(6, 1));//Sets 3 by 1 grid

selectContinentPanel.setBorder(BorderFactory.createLineBorder(Color.BLACK, 3));

selectContinentLabel = new JLabel("Select Your Continent", JLabel.CENTER);

selectContinentLabel.setFont(new Font("Courier", Font.PLAIN, 25));

selectContinentLabel.setForeground(Color.RED);//Changes its color to red

selectContinentPanel.add(selectContinentLabel);

//showModes.setBackground(selectContinentPanel.getBackground());

selectContinentComboBox = new JComboBox<String>();

selectContinentComboBox.setFont(new Font("Courier", Font.PLAIN, 20));

selectContinentComboBox.addActionListener(this);

for(int i=0; i<continents.length; i++)

{selectContinentComboBox.addItem(continents[i]);}

selectContinentPanel.add(easyOnes);

selectContinentPanel.add(moderateOnes);

selectContinentPanel.add(hardOnes);

selectContinentPanel.add(selectContinentComboBox);

selectContinentPanel.add(next);

pauseLabel = new JLabel("Paused", JLabel.CENTER);

pauseLabel.setFont(new Font("Courier", Font.BOLD, 50));

pauseLabel.setForeground(Color.RED);

jmb = new JMenuBar();

jmb.setLayout(new FlowLayout());

jm1 = new JMenu("");//Creates pause jmenu

jm1.setFont(new Font("Courier", Font.PLAIN, 15));

jm1.setIcon(pauseIcon);

jm1.addMenuListener(this);

jmb.add(jm1);//menubar adds quit menu

quitAndSave = new JMenuItem("Save and Quit");

quitAndSave.setFont(new Font("Courier", Font.PLAIN, 20));

quitAndNoSave = new JMenuItem("Quit");

quitAndNoSave.setFont(new Font("Courier", Font.PLAIN, 20));

quitAndNoSave.addActionListener(this);

quitAndSave.addActionListener(this);

resume = new JMenuItem("Resume");

resume.setFont(new Font("Courier", Font.PLAIN, 20));

resume.addActionListener(this);

jm1.add(resume);

//jm1.add(quitAndSave);//Adds them to the quit jmenu

jm1.add(quitAndNoSave);

jm2 = new JMenu("");

jm2.setIcon(miniSoundIcon);//Puts sound on icon

jmb.add(jm2);

soundFX = new JMenuItem("Sound Effects");

if(keepSFX)soundFX.setIcon(new ImageIcon("miniSound.png"));

else soundFX.setIcon(new ImageIcon("miniMute.png"));

soundFX.addActionListener(this);

soundFX.setHorizontalTextPosition(SwingConstants.LEFT);

soundFX.setFont(new Font("Courier", Font.PLAIN, 20));

backMusic = new JMenuItem("Music");

if(keepMusic)backMusic.setIcon(new ImageIcon("miniSound.png"));

else backMusic.setIcon(new ImageIcon("miniMute.png"));

backMusic.setHorizontalTextPosition(SwingConstants.LEFT);

backMusic.addActionListener(this);

backMusic.setFont(new Font("Courier", Font.PLAIN, 20));

jm2.add(soundFX);

jm2.add(backMusic);

jm3 = new JMenu(""+startingMoney[whichCont]);

jm3.setFont(new Font("Courier", Font.PLAIN, 17));

ImageIcon moneyIcon = new ImageIcon("moneyReal.png");

jm3.setIcon(moneyIcon);//Puts money icon image in the menubar

jmb.add(jm3);

jm4 = new JMenu(""+populations[0]);

jm4.setFont(new Font("Courier", Font.PLAIN, 17));

jm4.setIcon(new ImageIcon("population.png"));

jmb.add(jm4);

jm5 = new JMenu(""+startingTroops[0]);

jm5.setFont(new Font("Courier", Font.PLAIN, 17));

jm5.setIcon(new ImageIcon("troops.png"));

jmb.add(jm5);

jm6 = new JMenu("Declare War!");

jm6.setFont(new Font("Courier", Font.PLAIN, 17));

jm6.setIcon(new ImageIcon("declareWar.png"));

jmb.add(jm6);

jm7 = new JMenu("War Status");

jm7.setFont(new Font("Courier", Font.PLAIN, 17));

jm7.setIcon(new ImageIcon("warStatus.png"));

jm7.addActionListener(this);

jmb.add(jm7);

jm11 = new JMenu("Upgrade");

jm11.setFont(new Font("Courier", Font.PLAIN, 17));

jm11.setIcon(new ImageIcon("tank.png"));

jmb.add(jm11);

aircraftMenu.setFont(new Font("Courier", Font.PLAIN, 19));

aircraftMenu.setIcon(new ImageIcon("fighterJet.png"));

aircraftMenu.setHorizontalTextPosition(SwingConstants.LEFT);

/\*

marineMenu.setFont(new Font("Courier", Font.PLAIN, 19));

marineMenu.setIcon(new ImageIcon("submarine.png"));

marineMenu.setHorizontalTextPosition(SwingConstants.LEFT);

weaponMenu.setFont(new Font("Courier", Font.PLAIN, 19));

weaponMenu.setIcon(new ImageIcon("tankWeapon.png"));

weaponMenu.setHorizontalTextPosition(SwingConstants.LEFT);

\*/

aBombMenu.setFont(new Font("Courier", Font.PLAIN, 19));

aBombMenu.setIcon(new ImageIcon("atomicBomb.png"));

aBombMenu.setHorizontalTextPosition(SwingConstants.LEFT);

/\*

nBombMenu.setFont(new Font("Courier", Font.PLAIN, 19));

nBombMenu.setIcon(new ImageIcon("nuclearBomb.png"));

nBombMenu.setHorizontalTextPosition(SwingConstants.LEFT);

\*/

nationalDefense.setFont(new Font("Courier", Font.PLAIN, 19));

nationalDefense.setIcon(new ImageIcon("defense.png"));

nationalDefense.setHorizontalTextPosition(SwingConstants.LEFT);

aircraftMenu.addActionListener(this);

//marineMenu.addActionListener(this);

//weaponMenu.addActionListener(this);

aBombMenu.addActionListener(this);

//nBombMenu.addActionListener(this);

nationalDefense.addActionListener(this);

jm11.add(aircraftMenu);

//jm11.add(marineMenu);

//jm11.add(weaponMenu);

jm11.add(aBombMenu);

//jm11.add(nBombMenu);

jm11.add(nationalDefense);

jm8 = new JMenu("World News");

jm8.setFont(new Font("Courier", Font.PLAIN, 17));

jm8.setIcon(new ImageIcon("news.png"));

jmb.add(jm8);

jm9 = new JMenu("Days Past:"+daysElapsed);

jm9.setFont(new Font("Courier", Font.PLAIN, 17));

jm9.setIcon(new ImageIcon("clock.png"));

jmb.add(jm9);

jm10 = new JMenu(""+month+"/"+day+"/"+year);

jm10.setFont(new Font("Courier", Font.PLAIN, 17));

jmb.add(jm10);

panelWithStartingSpeech = new JPanel();

panelWithStartingSpeech.setLayout(new GridLayout(2,0));

panelWithStartingSpeech.setBackground(Color.WHITE);

panelWithStartingSpeech.setBorder(BorderFactory.createLineBorder(Color.BLACK, 3));

begin = new JButton("Begin");

begin.setFont(new Font("Courier", Font.PLAIN, 20));

begin.addActionListener(this);

startingSpeechPanel = new JTextArea();

startingSpeechPanel.setText("\nYou are the leader of " + continent + " and must drive you and your continent to the top. Conquer or be conquered!");

startingSpeechPanel.setFont(new Font("Courier", Font.PLAIN, 20));

startingSpeechPanel.setLineWrap(true);

startingSpeechPanel.setWrapStyleWord(true);

startingSpeechPanel.setEditable(false);

panelWithStartingSpeech.add(startingSpeechPanel);

panelWithStartingSpeech.add(begin);

eventPanel = new JTextArea();

eventPanel.setLineWrap(true);

eventPanel.setWrapStyleWord(true);

eventPanel.setEditable(false);

eventPanel.setFont(new Font("Courier", Font.PLAIN, 20));

continueButton = new JButton("Continue");

continueButton.setFont(new Font("Courier", Font.PLAIN, 20));

continueButton.addActionListener(this);

panelWithEvent = new JPanel();

panelWithEvent.setLayout(new GridLayout(2,1));

panelWithEvent.add(eventPanel);

panelWithEvent.add(continueButton);

panelWithEvent.setBorder(BorderFactory.createLineBorder(Color.BLACK, 3));

daysToNextEvent = (int)(Math.random()\*15+7);

timer = new Timer(250, this);//1000

timerForPlane = new Timer(5, new PlaneMover());

timerForABombs = new Timer(5, new ABombMover());

timerForNukeEnemy = new Timer(5, new EnemyNukeMover());

timerForPlaneEnemy = new Timer(5, new EnemyPlaneMover());

//timer.start();

//tryCatch();//Calls try catch method

}

//Method to deal with when user clicks on a JMenu(Currently for the pause JMenu)

public void menuSelected(MenuEvent e)

{

if(e.getSource()==jm1 && isPaused==false && startGame)

{

timer.stop();

startDayCounter = false;

removeAll();

setLayout(new GridLayout(3,3));//Sets 3 by 3 grid

for(int i=0; i<9; i++)

{

if(i==4) add(pauseLabel);//Adds pause label

else add(new JLabel(""));//Else adds empty label

}

revalidate();

repaint();

isPaused = true;

}

}

//This method is currently not used(Needed by implementation of MenuListener)

public void menuDeselected(MenuEvent e)

{

}

//This method is currently not used(Needed by implementation of MenuListener)

public void menuCanceled(MenuEvent e)

{

}

//Method to handle if user clicks new game and

//sets up gameOptionsPanel(Easy, Moderate, Hard)

//It is made to make code more efficient

public void setGameOptionsPanel()

{

removeAll();//removes all components from the panel

setLayout(new GridLayout(1,1));//Resets the grid to 1 by 1

add(gameOptions);

revalidate();

repaint();

}

public void keyPressed(KeyEvent e)

{

}

public void keyReleased(KeyEvent e)

{

}

public void keyTyped(KeyEvent e)

{

/\*

currentTroopCost[i] = currentPlanes[i]\*5000l+currentBoats[i]\*boatSpace+currentABombs[i]\*500l+currentNBombs[i]\*nBombSpace;

currentTroopCost[i] = currentPlanes[i]\*5000l+currentBoats[i]\*boatSpace+currentABombs[i]\*500l+currentNBombs[i]\*nBombSpace;

\*/

for(int i=0; i<aircraftBoxes.length; i++)

{

if(e.getSource()==aircraftBoxes[i])

{

String test = aircraftBoxes[i].getText()+e.getKeyChar();

if((int)(e.getKeyChar())==8) test = test.substring(0, test.length()-1);

try

{

if(test.equals("")==false)currentPlanes[i] = Integer.parseInt(test);

else currentPlanes[i] = 0;

currentTroopCost[i] = currentPlanes[i]\*5000l+currentBoats[i]\*boatSpace+currentABombs[i]\*500l+currentNBombs[i]\*nBombSpace;

troopCostBoxes[i].setText(""+currentTroopCost[i]);

currentTotalCost[i] = (currentPlanes[i]\*500000l+currentBoats[i]\*5000000+currentABombs[i]\*25000000l+currentNBombs[i]\*50000000);

moneyTotalBoxes[i].setText(""+currentTotalCost[i]);

if(currentTotalCost[i]>500000000){ moneyGO=true; repaint();}

}

catch (NumberFormatException error)

{

aircraftBoxes[i].setText(test.substring(0, test.length()-1));

currentTroopCost[i] = currentPlanes[i]\*5000l+currentBoats[i]\*boatSpace+currentABombs[i]\*500l+currentNBombs[i]\*nBombSpace;

troopCostBoxes[i].setText(""+currentTroopCost[i]);

currentTotalCost[i] = (currentPlanes[i]\*500000l+currentBoats[i]\*5000000+currentABombs[i]\*25000000l+currentNBombs[i]\*50000000);

moneyTotalBoxes[i].setText(""+currentTotalCost[i]);

if(currentTotalCost[i]>500000000) { moneyGO=true; repaint();}

return;

}

}

}

for(int i=0; i<marineBoxes.length; i++)

{

if(e.getSource()==marineBoxes[i])

{

String test = marineBoxes[i].getText()+e.getKeyChar();

if((int)(e.getKeyChar())==8) test = test.substring(0, test.length()-1);

try

{

if(test.equals("")==false)currentBoats[i] = Integer.parseInt(test);

else currentBoats[i] = 0;

currentTroopCost[i] = currentPlanes[i]\*5000l+currentBoats[i]\*boatSpace+currentABombs[i]\*500l+currentNBombs[i]\*nBombSpace;

troopCostBoxes[i].setText(""+currentTroopCost[i]);

currentTotalCost[i] = (currentPlanes[i]\*500000l+currentBoats[i]\*5000000+currentABombs[i]\*25000000l+currentNBombs[i]\*50000000);

moneyTotalBoxes[i].setText(""+currentTotalCost[i]);

}

catch (NumberFormatException error)

{

marineBoxes[i].setText(test.substring(0, test.length()-1));

currentTroopCost[i] = currentPlanes[i]\*5000l+currentBoats[i]\*boatSpace+currentABombs[i]\*500l+currentNBombs[i]\*nBombSpace;

troopCostBoxes[i].setText(""+currentTroopCost[i]);

currentTotalCost[i] = (currentPlanes[i]\*500000l+currentBoats[i]\*5000000+currentABombs[i]\*25000000l+currentNBombs[i]\*50000000);

moneyTotalBoxes[i].setText(""+currentTotalCost[i]);

return;

}

}

}

for(int i=0; i<aBombBoxes.length; i++)

{

if(e.getSource()==aBombBoxes[i])

{

String test = aBombBoxes[i].getText()+e.getKeyChar();

if((int)(e.getKeyChar())==8) test = test.substring(0, test.length()-1);

try

{

if(test.equals("")==false)currentABombs[i] = Integer.parseInt(test);

else currentABombs[i] = 0;

currentTroopCost[i] = currentPlanes[i]\*5000l+currentBoats[i]\*boatSpace+currentABombs[i]\*500l+currentNBombs[i]\*nBombSpace;

troopCostBoxes[i].setText(""+currentTroopCost[i]);

currentTotalCost[i] = (currentPlanes[i]\*500000l+currentBoats[i]\*5000000+currentABombs[i]\*25000000l+currentNBombs[i]\*50000000);

moneyTotalBoxes[i].setText(""+currentTotalCost[i]);

if(currentTotalCost[i]>500000000){ moneyGO=true; }

}

catch (NumberFormatException error)

{

aBombBoxes[i].setText(test.substring(0, test.length()-1));

currentTroopCost[i] = currentPlanes[i]\*5000l+currentBoats[i]\*boatSpace+currentABombs[i]\*500l+currentNBombs[i]\*nBombSpace;

troopCostBoxes[i].setText(""+currentTroopCost[i]);

currentTotalCost[i] = (currentPlanes[i]\*500000l+currentBoats[i]\*5000000+currentABombs[i]\*25000000l+currentNBombs[i]\*50000000);

moneyTotalBoxes[i].setText(""+currentTotalCost[i]);

if(currentTotalCost[i]>500000000){ moneyGO=true;}

return;

}

}

}

for(int i=0; i<nBombBoxes.length; i++)

{

if(e.getSource()==nBombBoxes[i])

{

String test = nBombBoxes[i].getText()+e.getKeyChar();

if((int)(e.getKeyChar())==8) test = test.substring(0, test.length()-1);

try

{

if(test.equals("")==false)currentNBombs[i] = Integer.parseInt(test);

else currentNBombs[i] = 0;

currentTroopCost[i] = currentPlanes[i]\*5000l+currentBoats[i]\*boatSpace+currentABombs[i]\*500l+currentNBombs[i]\*nBombSpace;

troopCostBoxes[i].setText(""+currentTroopCost[i]);

currentTotalCost[i] = (currentPlanes[i]\*500000l+currentBoats[i]\*5000000+currentABombs[i]\*25000000l+currentNBombs[i]\*50000000);

moneyTotalBoxes[i].setText(""+currentTotalCost[i]);

}

catch (NumberFormatException error)

{

nBombBoxes[i].setText(test.substring(0, test.length()-1));

currentTroopCost[i] = currentPlanes[i]\*5000l+currentBoats[i]\*boatSpace+currentABombs[i]\*500l+currentNBombs[i]\*nBombSpace;

troopCostBoxes[i].setText(""+currentTroopCost[i]);

currentTotalCost[i] = (currentPlanes[i]\*500000l+currentBoats[i]\*5000000+currentABombs[i]\*25000000l+currentNBombs[i]\*50000000);

moneyTotalBoxes[i].setText(""+currentTotalCost[i]);

return;

}

}

}

}

//Method to deal with when user clicks on components

//like buttons, comboBox for selecting continent, JMenuItems, etc

//Also makes the date in the game move forward (Used Timer object)

public void actionPerformed(ActionEvent e)

{

System.out.println(e.getActionCommand());

boolean testWin = true;

for(int i=0; i<makeBlue.length; i++)

{

if(i!=whichCont)

{

if(makeBlue[i]==false) testWin=false;

}

}

if(testWin) won=true;

//System.out.println(e.getActionCommand());

if(year%4==0) daysOfMonth[1] = 29;

else daysOfMonth[1] = 28;

//System.out.println("made it: "+e.getActionCommand());

if(e.getSource()==instructions)

{

//System.out.println("made it 2");

putInstructions = true;

removeAll();

setLayout(new GridLayout(12,12));

for(int i=0; i<144; i++)

{

if(i==0) add(backFromFirstInstructions);

else if(i==23) add(goToSecondInstructions);

else add(new JLabel(""));

}

revalidate();

repaint();

}

//System.out.println("hey " + e.getActionCommand());

if(e.getSource()==quitAndNoSave)

{

//System.out.println("working");

jmb.setVisible(false);

remake();

}

if(e.getSource()==goToSecondInstructions)

{

putInstructions=false;

putInstructions2=true;

removeAll();

setLayout(new GridLayout(12,12));

for(int i=0; i<144; i++)

{

if(i==0) add(backFromInstructions2);

else add(new JLabel(""));

}

revalidate();

repaint();

}

if(e.getSource()==backFromInstructions2)

{

putInstructions=true;

putInstructions2=false;

removeAll();

setLayout(new GridLayout(12,12));

for(int i=0; i<144; i++)

{

if(i==0) add(backFromFirstInstructions);

else if(i==23) add(goToSecondInstructions);

else add(new JLabel(""));

}

revalidate();

repaint();

}

if(e.getSource()==play)//If user clicks play

{

removeAll();//removes all components from the panel

setLayout(new GridLayout(3,3));//Resets the grid to 3 by 3

for(int i=0; i<9; i++)

{

if(i==0) add(fixBackPanel);

else if(i==4) add(selectContinentPanel);

else add(new JLabel(""));//Adds empty JLabel or space

}

if(disableLoad)loadGame.setEnabled(false);

revalidate();

repaint();

}

if(e.getSource()==newGame) setGameOptionsPanel();//When user clicks new game

//if(e.getSource()==loadGame) tryCatch();

if(e.getSource()==backFromGameOptionsPanel)

{

removeAll();//removes all components from the panel

setLayout(new GridLayout(1,1));//Resets the grid to 1 by 1

add(gameSelectPanel);

if(disableLoad)loadGame.setEnabled(false);

revalidate();

repaint();

}

if(e.getSource()==easy || e.getSource()==moderate || e.getSource()==hard)

{

removeAll();//removes all components from the panel

setLayout(new GridLayout(3,3));//Resets the grid to 3 by 3

for(int i=0; i<9; i++)

{

if(i==4) add(selectContinentPanel);

else add(new JLabel(""));//Adds empty JLabel or space

}

revalidate();

repaint();

if(e.getSource()==easy) mode = "Easy";

if(e.getSource()==moderate) mode = "Moderate";

if(e.getSource()==hard) mode = "Hard";

}

//Put own action method for this if block

if(e.getSource()==selectContinentComboBox && jm4!=null)

{

whichCont = selectContinentComboBox.getSelectedIndex();

if(whichCont!=0){timesChangedCont=true;contCount++;}

//System.out.println("ehllo");

/\*

jm3.removeAll();

jm4.removeAll();

jm5.removeAll();

jm6.removeAll();

\*/

//for(int i)

repaint();

revalidate();

jm3.setText(" "+startingMoney[whichCont]);

if(whichCont!=3 && whichCont!=4)jm4.setText(" "+populations[whichCont]);

else jm4.setText(" "+populations[whichCont]);

for(int i=0; i<6; i++)

{

if(i!=whichCont)

{

if(i!=3 && i!=4)

{

continentPopulations[i] = new JMenuItem(" "+continents[i]+": "+populations[i]);

continentPopulations[i].setIcon(new ImageIcon("population.png"));

continentPopulations[i].setFont(new Font("Courier", Font.PLAIN, 20));

jm4.add(continentPopulations[i]);

}

else

{

continentPopulations[i] = new JMenuItem(" "+continents[i]+": "+populations[i]);

continentPopulations[i].setIcon(new ImageIcon("population.png"));

continentPopulations[i].setFont(new Font("Courier", Font.PLAIN, 20));

jm4.add(continentPopulations[i]);

}

}

}

for(int i=0; i<6; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i] = new JMenuItem(" "+continents[i]+": "+startingMoney[i]);

startingMoneyArray[i].setIcon(new ImageIcon("money.jpg"));

startingMoneyArray[i].setFont(new Font("Courier", Font.PLAIN, 20));

jm3.add(startingMoneyArray[i]);

troopPopulations[i] = new JMenuItem(" "+continents[i]+": "+startingTroops[i]);

troopPopulations[i].setIcon(new ImageIcon("troops.png"));

troopPopulations[i].setFont(new Font("Courier", Font.PLAIN, 20));

jm5.add(troopPopulations[i]);

warOptions[i] = new JMenuItem(continents[i]);

warOptions[i].setIcon(new ImageIcon("declareWar.png"));

warOptions[i].setFont(new Font("Courier", Font.PLAIN, 20));

warOptions[i].addActionListener(this);

jm6.add(warOptions[i]);

}

}

}

//Put own action method for this if block

if(e.getSource()==next)//Handles when user chooses next

{

if(whichCont==0 || timesChangedCont==false)//Problem right here

{

jm3.removeAll();

jm4.removeAll();

jm5.removeAll();

jm6.removeAll();

repaint();

revalidate();

}

/\*

if(timesChangedCont)

{

for(int i=5; i<10; i++)

{

jm3.remove(i);

jm4.remove(i);

jm5.remove(i);

jm6.remove(i);

}

}

\*/

showFlag = true;

removeAll();//removes all components from the panel

frame.setJMenuBar(jmb);//Puts the menu bar in the frame

jmb.setVisible(true);

for(int i=0; i<6; i++)

{

if(i!=whichCont && whichCont==0)

{

if(i!=3 && i!=4)

{

continentPopulations[i] = new JMenuItem(" "+continents[i]+": "+populations[i]);

continentPopulations[i].setIcon(new ImageIcon("population.png"));

continentPopulations[i].setFont(new Font("Courier", Font.PLAIN, 20));

jm4.add(continentPopulations[i]);

}

else

{

continentPopulations[i] = new JMenuItem(" "+continents[i]+": "+populations[i]);

continentPopulations[i].setIcon(new ImageIcon("population.png"));

continentPopulations[i].setFont(new Font("Courier", Font.PLAIN, 20));

jm4.add(continentPopulations[i]);

}

}

}

for(int i=0; i<6; i++)

{

if(i!=whichCont && whichCont==0)

{

startingMoneyArray[i] = new JMenuItem(" "+continents[i]+": "+startingMoney[i]);

startingMoneyArray[i].setIcon(new ImageIcon("money.jpg"));

startingMoneyArray[i].setFont(new Font("Courier", Font.PLAIN, 20));

jm3.add(startingMoneyArray[i]);

troopPopulations[i] = new JMenuItem(" "+continents[i]+": "+startingTroops[i]);

troopPopulations[i].setIcon(new ImageIcon("troops.png"));

troopPopulations[i].setFont(new Font("Courier", Font.PLAIN, 20));

jm5.add(troopPopulations[i]);

warOptions[i] = new JMenuItem(continents[i]);

warOptions[i].setIcon(new ImageIcon("declareWar.png"));

warOptions[i].setFont(new Font("Courier", Font.PLAIN, 20));

warOptions[i].addActionListener(this);

jm6.add(warOptions[i]);

}

}

continent = continents[selectContinentComboBox.getSelectedIndex()];

startingSpeechPanel.setText("\nYou are the leader of " + continent + " and must drive you and your continent to the top. Conquer or be conquered!");

setLayout(new GridLayout(3,3));

for(int i=0; i<9; i++)

{

if(i==4) add(panelWithStartingSpeech);

else add(new JLabel(""));

}

revalidate();

repaint();

}

if(e.getSource()==settings)//Handles when user chooses settings in title screen

{

removeAll();//removes all components from the panel

setLayout(new GridLayout(3,3));

for(int i=0; i<9; i++)

{

if(i==4) add(panelWithSettings);

else add(new JLabel(""));

}

revalidate();

repaint();

whichPanel = 2;

}

if(e.getSource()==soundEffects)

{

if(keepSFX)

{

soundEffects.setText("Sound Effects");

keepSFX = false;

soundEffects.setIcon(new ImageIcon("mute.png"));

soundFX.setIcon(new ImageIcon("miniMute.png"));

}

else

{

soundEffects.setText("Sound Effects");

keepSFX = true;

soundEffects.setIcon(new ImageIcon("sound.png"));

soundFX.setIcon(new ImageIcon("miniSound.png"));

}

}

if(e.getSource()==music)

{

if(keepMusic)

{

clip.stop();//Stops the music

music.setText("Music");

keepMusic = false;

music.setIcon(new ImageIcon("mute.png"));

backMusic.setIcon(new ImageIcon("miniMute.png"));

}

else

{

clip.loop();//Loops the music again

music.setText("Music");

keepMusic = true;

music.setIcon(new ImageIcon("sound.png"));

backMusic.setIcon(new ImageIcon("miniSound.png"));

}

}

if(e.getSource()==back && whichPanel==2 || e.getSource()==backFromGameSelectPanel || e.getSource()==home || e.getSource()==backFromFirstInstructions || e.getSource()==backFromLeaderboard)

{

endGame=false;

moneyGO=false;

troopGO=false;

healthGO=false;

peopleGO=false;

resetToMainScreen();

if(e.getSource()==home) {remake();moneyGO=false;won=false;troopGO=false;peopleGO=false;healthGO=false;}

}

if(e.getSource()==soundFX)//If the user presses sound effects button

{

if(keepSFX)//If the sound effects are currently on

{

keepSFX = false;

soundFX.setIcon(new ImageIcon("miniMute.png"));//Sets icon to mute icon

}

else //If the sound effects is currently off

{

keepSFX = true;

soundFX.setIcon(new ImageIcon("miniSound.png"));//Sets icon to sound icon

}

}

if(e.getSource()==backMusic)//If the user presses music button

{

if(keepMusic)//If the music is currently playing

{

clip.stop();//Stops the music

keepMusic = false;

backMusic.setIcon(new ImageIcon("miniMute.png"));//Sets icon to mute icon

}

else //If the music is currently off

{

clip.loop();//Loops the music again

keepMusic = true;

backMusic.setIcon(new ImageIcon("miniSound.png"));

}

}

if(e.getSource()==resume && startGame)//If when the game is paused and the user presses resume

{

timer.start();

startDayCounter = true;

removeAll();

setLayout(new GridLayout(1,1));

revalidate();//This and the next line are needed for removing the component

repaint();

isPaused = false;

}

for(int i=0; i<6; i++)

{

if(e.getSource()==warOptions[i] && timesWarOptionsClicked[i]>0)

{

if(sendThePlane==false && sendTheABomb==false)contToAttack = warOptions[i].getText();

timer.stop();

removeAll();

setLayout(new GridLayout(3,3));

for(int j=0; j<9; j++)

{

if(j==4) add(warStatusPanels[i]);

else add(new JLabel(""));

}

revalidate();

repaint();

}

}

for(int i=0; i<6; i++)

{

if(e.getSource()==warOptions[i])

{

timesWarOptionsClicked[i]++;

jm6.remove(warOptions[i]);

if(warOptions[i].getText().equals("North America"))

{

healthBars[0] = true;

canAdvance[0][0] = true;

canAdvance[1][0] = true;

canAdvance[2][0] = true;

canAdvance[3][0] = true;

canAdvance[0][whichCont] = true;

canAdvance[1][whichCont] = true;

canAdvance[2][whichCont] = true;

canAdvance[3][whichCont] = true;

warCheck[0][whichCont] = true;

}

if(warOptions[i].getText().equals("South America"))

{

healthBars[1] = true;

canAdvance[0][1] = true;

canAdvance[1][1] = true;

canAdvance[2][1] = true;

canAdvance[3][1] = true;

canAdvance[0][whichCont] = true;

canAdvance[1][whichCont] = true;

canAdvance[2][whichCont] = true;

canAdvance[3][whichCont] = true;

warCheck[1][whichCont] = true;

}

if(warOptions[i].getText().equals("Europe"))

{

healthBars[2] = true;

canAdvance[0][2] = true;

canAdvance[1][2] = true;

canAdvance[2][2] = true;

canAdvance[3][2] = true;

canAdvance[0][whichCont] = true;

canAdvance[1][whichCont] = true;

canAdvance[2][whichCont] = true;

canAdvance[3][whichCont] = true;

warCheck[2][whichCont] = true;

}

if(warOptions[i].getText().equals("Asia"))

{

healthBars[3] = true;

canAdvance[0][3] = true;

canAdvance[1][3] = true;

canAdvance[2][3] = true;

canAdvance[3][3] = true;

canAdvance[0][whichCont] = true;

canAdvance[1][whichCont] = true;

canAdvance[2][whichCont] = true;

canAdvance[3][whichCont] = true;

warCheck[3][whichCont] = true;

}

if(warOptions[i].getText().equals("Africa"))

{

healthBars[4] = true;

canAdvance[0][4] = true;

canAdvance[1][4] = true;

canAdvance[2][4] = true;

canAdvance[3][4] = true;

canAdvance[0][whichCont] = true;

canAdvance[1][whichCont] = true;

canAdvance[2][whichCont] = true;

canAdvance[3][whichCont] = true;

warCheck[4][whichCont] = true;

}

if(warOptions[i].getText().equals("Australia"))

{

healthBars[5] = true;

canAdvance[0][5] = true;

canAdvance[1][5] = true;

canAdvance[2][5] = true;

canAdvance[3][5] = true;

canAdvance[0][whichCont] = true;

canAdvance[1][whichCont] = true;

canAdvance[2][whichCont] = true;

canAdvance[3][whichCont] = true;

warCheck[5][whichCont] = true;

}

jm7.add(warOptions[i]);

repaint();

}

}

if(e.getSource()==nationalDefense)

{

timer.stop();

nationalDefenseLevel++;

removeAll();

setLayout(new GridLayout(3,3));

for(int i=0; i<9; i++)

{

if(i==4) add(defenseMenu);

else add(new JLabel(""));

}

revalidate();

repaint();

}

if(e.getSource()==confirmDefense)

{

timer.start();

if(healthReduce[whichCont]<=-10)healthReduce[whichCont] += 5;

else healthReduce[whichCont]=0;

startingMoney[whichCont] -= 10000000\*(nationalDefenseLevel);

jm3.setText(""+startingMoney[whichCont]);

defenseCount[whichCont] += 0.1;

//nationalDefenseLevel++;

nationalDefense.setText("Level "+(nationalDefenseLevel+1));

removeAll();

setLayout(new GridLayout(1,1));

revalidate();

repaint();

moneyForDefense.setText(""+10000000\*(nationalDefenseLevel+1));

}

if(e.getSource()==cancelPlane || e.getSource()==cancelBomb || e.getSource()==cancelDefense)

{

timer.start();

removeAll();

setLayout(new GridLayout(1,1));

revalidate();

repaint();

}

if(e.getSource()==aircraftMenu)

{

timer.stop();

jetLevel++;

removeAll();

setLayout(new GridLayout(3,3));

for(int i=0; i<9; i++)

{

if(i==4) add(planePanel);

else add(new JLabel(""));

}

revalidate();

repaint();

}

if(e.getSource()==confirmPlane)

{

timer.start();

startingMoney[whichCont] -= 10000000\*(jetLevel);

jm3.setText(""+startingMoney[whichCont]);

for(int i=0; i<6; i++) if(i!=whichCont)defenseCount[i] -= 0.1;

aircraftMenu.setText("Level "+(jetLevel+1));

removeAll();

setLayout(new GridLayout(1,1));

revalidate();

repaint();

moneyForPlane.setText(""+10000000\*(jetLevel+1));

}

if(e.getSource()==aBombMenu)

{

timer.stop();

bombLevel++;

removeAll();

setLayout(new GridLayout(3,3));

for(int i=0; i<9; i++)

{

if(i==4) add(bombMenu);

else add(new JLabel(""));

}

revalidate();

repaint();

}

if(e.getSource()==confirmBomb)

{

timer.start();

startingMoney[whichCont] -= 10000000\*(bombLevel);

jm3.setText(""+startingMoney[whichCont]);

for(int i=0; i<6; i++) if(i!=whichCont)defenseCount[i] -= 0.01;

aBombMenu.setText("Level "+(bombLevel+1));

removeAll();

setLayout(new GridLayout(1,1));

revalidate();

repaint();

moneyForBomb.setText(""+10000000\*(bombLevel+1));

}

if(e.getSource()==begin)

{

timer.start();

startGame = true;

removeAll();

setLayout(new GridLayout(3,3));

revalidate();//This and the next line are needed for removing the component

repaint();

startDayCounter=true;

}

for(int i=0; i<contButs.length; i++)

{

if(e.getSource()==contButs[i])

{

removeAll();

setLayout(new GridLayout(1,1));

revalidate();

repaint();

timer.start();

for(int j=0; j<6; j++) confirm[j].setEnabled(true);

}

}

if(e.getSource()==continueButton)

{

removeAll();

revalidate();

repaint();

daysToNextEvent = (int)(Math.random()\*15+7);

jm1.addMenuListener(this);

timer.start();

for(int i=0; i<6; i++) confirm[i].setEnabled(true);

}

for(int i=0; i<6; i++)

{

if(e.getSource()==cancel[i])

{

timer.start();

removeAll();

setLayout(new GridLayout(1,1));

revalidate();

repaint();

}

}

for(int i=0; i<confirm.length; i++)

{

if(e.getSource()==confirm[i] && moneyGO==false)//put string var here

{

//System.out.println("about to leave");

removeAll();

setLayout(new GridLayout(1,1));

revalidate();

repaint();

planesToSend = currentPlanes[i];

aBombsToSend = currentABombs[i];

if(planesToSend>0)sendThePlane = true;

if(aBombsToSend>0)sendTheABomb = true;

if(sendThePlane && sendTheABomb){sendThePlane=false;setLater=true;}

if(planesToSend>0)planeXSpots = new int[planesToSend];

if(planesToSend>0)showPlanes = new boolean[planesToSend];

if(planesToSend>0)for(int j=0; j<showPlanes.length; j++) showPlanes[j] = true;

if(aBombsToSend>0)aBombXSpots = new int[aBombsToSend];

if(aBombsToSend>0)

{

aBombSizes = new int[aBombsToSend];

showExplosion = new boolean[aBombsToSend];

bombSoundCounter = new int[aBombsToSend];

for(int j=0; j<aBombSizes.length; j++) aBombSizes[j] = size;

}

if(aBombsToSend>0)showABombs = new boolean[aBombsToSend];

if(aBombsToSend>0)dropABombs = new boolean[aBombsToSend];

if(aBombsToSend>0)for(int j=0; j<showABombs.length; j++) showABombs[j] = true;

if(aBombsToSend>0) shouldABombExplode = new boolean[aBombsToSend];

startingMoney[whichCont] -= currentTotalCost[i];

startingTroops[whichCont] -= currentTroopCost[i];

jm3.setText(""+startingMoney[whichCont]);

jm5.setText(""+startingTroops[whichCont]);

if(startingMoney[whichCont]<=0) {moneyGO=true;endGame=true;}

if(startingTroops[whichCont]<=0) {troopGO=true;endGame=true;}

timer.start();

revalidate();

repaint();

//System.out.println("about to leave");

}

else if(moneyGO){endGame=true;repaint();}

}

//System.out.println("left");

for(int i=0; i<eventList.length; i++)

{

if(e.getSource()==eventList[i])

{

timer.stop();

removeAll();

setLayout(new GridLayout(3,3));

for(int j=0; j<9; j++)

{

if(j==4) add(eventLogs[i]);

else add(new JLabel(""));

}

revalidate();

repaint();

}

}

if(sendTheNukeEnemy==false && sendTheABomb==false && sendThePlaneEnemy==false && sendThePlane==false)for(int i=0; i<6; i++) confirm[i].setEnabled(true);

if(e.getSource()==timer) {

if(startDayCounter){daysElapsed++;day++;}

if(day>daysOfMonth[month-1]) {day=1; month++;}

if(month>12) {month=1;year++;}

if(jm9!=null && startDayCounter)jm9.setText("Days Past:"+daysElapsed);

if(jm10!=null && startDayCounter)jm10.setText(""+month+"/"+day+"/"+year);

daysSinceLastEvent++;

//System.out.println("South America: " + startingMoney[1]);

//System.out.println("here are the stats: " + daysSinceLastEvent + " and " + daysToNextEvent + " and boolean is " + sendTheNukeEnemy);

if(daysSinceLastEvent>=daysToNextEvent && sendTheNukeEnemy==false && sendTheABomb==false && sendThePlaneEnemy==false)//daysElapsed%10==0 && daysElapsed!=0

{

//System.out.println("dude be in here");

daysSinceLastEvent=0;

timer.stop();

jm1.removeMenuListener(this);

setLayout(new GridLayout(3,3));

getRandomEvent();

if(sendTheNukeEnemy==false && sendTheABomb==false && sendThePlaneEnemy==false)

{

for(int i=0; i<9; i++)

{

if(i==4) {add(panelWithEvent);}

else add(new JLabel(""));

}

revalidate();

repaint();

}

else timer.start();

}

else if(sendTheNukeEnemy || sendTheABomb || sendThePlaneEnemy || sendThePlane)

{

for(int i=0; i<6; i++) confirm[i].setEnabled(false);

}

}

}

//This method generates random events in the game

//Most of the logic is complete

public void getRandomEvent()

{

//System.out.println("event");

int randomContinent = (int)(Math.random()\*6);

int randomContinent2 = (int)(Math.random()\*6);

while(continents[randomContinent].equals(continents[randomContinent2])) randomContinent2 = (int)(Math.random()\*6);

int moneyCost = (int)(Math.random()\*10000000+1000000);

int peopleDamage = (int)(Math.random()\*100000+500);

whichSubEvent = (int)(Math.random()\*naturalDisasters.length);//length

while(postIt==false)

{

whichSubEvent = (int)(Math.random()\*naturalDisasters.length);

while(whichSubEvent==7) whichSubEvent = (int)(Math.random()\*naturalDisasters.length);//length

moneyCost = (int)(Math.random()\*10000000+1000000);

peopleDamage = (int)(Math.random()\*10000+500);

//System.out.println("bro: "+whichSubEvent);

if(whichSubEvent==5) {moneyCost\*=2; peopleDamage\*=15;}

if(whichSubEvent==6) {moneyCost\*=10; peopleDamage\*=50;}

if(whichSubEvent==8 || whichSubEvent==10 || whichSubEvent==13) {moneyCost=0; peopleDamage=0; }

if(whichSubEvent==9) {moneyCost=0; peopleDamage=0;}

if(whichSubEvent==11) {moneyCost = (int)(Math.random()\*10000000+1000000); peopleDamage=0;}

if(whichSubEvent<6 || whichSubEvent==11)

{

while(canUpdateInfo[randomContinent])randomContinent = (int)(Math.random()\*6);

while(canUpdateInfo[randomContinent2])randomContinent2 = (int)(Math.random()\*6);

}

else if(whichSubEvent>=6 && whichSubEvent<=10)

{

while(randomContinent==whichCont || randomContinent==randomContinent2 || canUpdateInfo[randomContinent]) randomContinent = (int)(Math.random()\*6);

while(randomContinent==whichCont || randomContinent==randomContinent2 || canUpdateInfo[randomContinent2]) randomContinent2 = (int)(Math.random()\*6);

}

else if(whichSubEvent>11)

{

while(randomContinent==whichCont || randomContinent==randomContinent2 || canUpdateInfo[randomContinent]) randomContinent = (int)(Math.random()\*6);

while(randomContinent==whichCont || randomContinent==randomContinent2 || canUpdateInfo[randomContinent2]) randomContinent2 = (int)(Math.random()\*6);

}

if(whichSubEvent==0 && Math.random()>0.80){event = tsunamiEvent[0] + continents[randomContinent] + ". " + tsunamiEvent[1] + moneyCost + tsunamiEvent[2] + peopleDamage + tsunamiEvent[3];postIt=true;defenseCount[randomContinent]-=0.15;}

else if(whichSubEvent==1 && Math.random()>0.80){event = earthquakeEvent[0] + continents[randomContinent] + ". " + earthquakeEvent[1] + moneyCost + earthquakeEvent[2] + peopleDamage + earthquakeEvent[3];postIt=true;defenseCount[randomContinent]-=0.15;}

else if(whichSubEvent==2 && Math.random()>0.80){event = volcanoEvent[0] + continents[randomContinent] + ". " + volcanoEvent[1] + moneyCost + volcanoEvent[2] + peopleDamage + volcanoEvent[3];postIt=true;defenseCount[randomContinent]-=0.15;}

else if(whichSubEvent==3 && Math.random()>0.80){event = tornadoEvent[0] + continents[randomContinent] + ". " + tornadoEvent[1] + moneyCost + tornadoEvent[2] + peopleDamage + tornadoEvent[3];postIt=true;defenseCount[randomContinent]-=0.15;}

else if(whichSubEvent==4 && Math.random()>0.80){event = floodingEvent[0] + continents[randomContinent] + ". " + floodingEvent[1] + moneyCost + floodingEvent[2] + peopleDamage + floodingEvent[3];postIt=true;defenseCount[randomContinent]-=0.15;}

else if(whichSubEvent==5 && Math.random()>0.80){event = diseaseEvent[0] + continents[randomContinent] + ". " + diseaseEvent[1] + moneyCost + diseaseEvent[2] + peopleDamage + diseaseEvent[3];postIt=true;defenseCount[randomContinent]-=0.15;}

else if(whichSubEvent==11 && Math.random()>0.75){event = continents[randomContinent] + economyBoostEvent[0] + "$"+Math.abs(moneyCost) + economyBoostEvent[1];postIt=true;}

// else if(whichSubEvent==13 && Math.random()>0.25)

//{

// event = continents[randomContinent] + allyEvent[0] + continents[randomContinent2] + allyEvent[1]; postIt=true;

// allies[randomContinent][randomContinent2]=true;

// allies[randomContinent2][randomContinent]=true;

//}

else if(whichSubEvent==9 && canAdvance[0][randomContinent])

{

if(canAdvance[1][randomContinent]==false)

{

//while(allies[randomContinent][randomContinent2]==true && allies[randomContinent2][randomContinent]==true && randomContinent==whichCont && randomContinent2==whichCont) randomContinent = (int)(Math.random()\*6);

postIt=true;

event = continents[randomContinent] + nationalDefenseEvent[0] + continents[randomContinent] + nationalDefenseEvent[1];

canAdvance[1][randomContinent] = true;

for(int i=0; i<warCheck[randomContinent].length; i++)

{

if(warCheck[randomContinent][i]) canAdvance[1][i] = true;

}

for(int i=0; i<6; i++)

{

for(int j=0; j<6; j++)

{

if(allies[i][j])

{

//if(i==randomContinent && j==randomContinent2) {warCheck[i][j] = true;warCheck[j][i] = true;canAdvance[1][i] = true;canAdvance[1][j] = true;}

//if(j==randomContinent && i==randomContinent2) {warCheck[i][j] = true;warCheck[j][i] = true;canAdvance[1][i] = true;canAdvance[1][j] = true;}

}

}

}

}

}

else if(whichSubEvent==8 && canAdvance[1][randomContinent])

{

// while(allies[randomContinent][randomContinent2]==true && allies[randomContinent2][randomContinent]==true && randomContinent==whichCont && randomContinent2==whichCont) randomContinent = (int)(Math.random()\*6);

if(canAdvance[2][randomContinent]==false)// || canAdvance[2][randomContinent2]==false

{

postIt=true;

//for(int i=0; i<warCheck[randomContinent].length; i++)

//{

int generate = (int)(Math.random()\*6);

while(warCheck[randomContinent][generate]==false || canUpdateInfo[generate]) generate = (int)(Math.random()\*6);

//}

randomContinent2=generate;

event = continents[randomContinent] + declareWarEvent[0] + continents[randomContinent2]+"!";

canAdvance[2][randomContinent] = true;

for(int i=0; i<warCheck[randomContinent].length; i++)

{

if(warCheck[randomContinent][i]) canAdvance[2][i] = true;

}

for(int i=0; i<6; i++)

{

for(int j=0; j<6; j++)

{

if(allies[i][j])

{

// if(i==randomContinent && j==randomContinent2) {warCheck[i][j] = true;warCheck[j][i] = true;canAdvance[2][i] = true;canAdvance[2][j] = true;}

// if(j==randomContinent && i==randomContinent2) {warCheck[i][j] = true;warCheck[j][i] = true;canAdvance[2][i] = true;canAdvance[2][j] = true;}

}

}

}

}

}

else if(whichSubEvent==12 && canAdvance[2][randomContinent] && startingMoney[randomContinent]>500000)//6 canAdvance[2][randomContinent]

{

//while(allies[randomContinent][randomContinent2]==true && allies[randomContinent2][randomContinent]==true && randomContinent==whichCont && randomContinent2==whichCont) randomContinent = (int)(Math.random()\*6);

int generate = (int)(Math.random()\*6);

while(warCheck[randomContinent][generate]==false || canUpdateInfo[generate]) generate = (int)(Math.random()\*6);

randomContinent2=generate;

postIt=true;

planesToSendEnemy = 5;

if(planesToSendEnemy>0)sendThePlaneEnemy = true;

//if(sendThePlane && sendTheABomb){sendThePlane=false;setLater=true;}

if(planesToSendEnemy>0)planeXSpotsEnemy = new int[planesToSendEnemy];

if(planesToSendEnemy>0)showPlanesEnemy = new boolean[planesToSendEnemy];

if(planesToSendEnemy>0)for(int j=0; j<showPlanesEnemy.length; j++) showPlanesEnemy[j] = true;

tempContinent = continents[randomContinent];

attackingCont = randomContinent;

contToAttackEnemy = continents[randomContinent2];;

startingMoney[randomContinent] -= planesToSendEnemy\*500000;

startingTroops[randomContinent] -= planesToSendEnemy\*5000;

repaint();

canAdvance[3][randomContinent] = true;

for(int i=0; i<warCheck[randomContinent].length; i++)

{

if(warCheck[randomContinent][i]) canAdvance[3][i] = true;

}

for(int i=0; i<6; i++)

{

for(int j=0; j<6; j++)

{

if(allies[i][j])

{

//if(i==randomContinent && j==randomContinent2) {warCheck[i][j] = true;warCheck[j][i] = true;canAdvance[0][i] = true;canAdvance[3][j] = true;}

//if(j==randomContinent && i==randomContinent2) {warCheck[i][j] = true;warCheck[j][i] = true;canAdvance[0][i] = true;canAdvance[3][j] = true;}

}

}

}

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

troopPopulations[i].setText(" "+continents[i]+": "+startingTroops[i]);

}

else

{

jm3.setText(""+startingMoney[i]);

jm4.setText(""+populations[i]);

jm5.setText(""+startingTroops[i]);

}

}

}

else if(whichSubEvent==6 && canAdvance[3][randomContinent] && startingMoney[randomContinent]>30000000)//6 canAdvance[2][randomContinent]

{

//while(allies[randomContinent][randomContinent2]==true && allies[randomContinent2][randomContinent]==true && randomContinent==whichCont && randomContinent2==whichCont) randomContinent = (int)(Math.random()\*6);

int generate = (int)(Math.random()\*6);

while(warCheck[randomContinent][generate]==false || canUpdateInfo[generate]) generate = (int)(Math.random()\*6);

//}

randomContinent2=generate;

postIt=true;

if(randomContinent==randomContinent2) event = nuclearBombEvent[0] + continents[randomContinent] + " accidentally"+nuclearBombEvent[1] + continents[randomContinent2] + "! " + nuclearBombEvent[2]+ moneyCost + nuclearBombEvent[3] + peopleDamage + nuclearBombEvent[4];

else event = nuclearBombEvent[0] + continents[randomContinent] + nuclearBombEvent[1] + continents[randomContinent2] + "! " + nuclearBombEvent[2]+ moneyCost + nuclearBombEvent[3] + peopleDamage + nuclearBombEvent[4];

nukesToSendEnemy = 1;

// if(planesToSend>0)sendThePlane = true;

if(nukesToSendEnemy>0)sendTheNukeEnemy = true;

// if(sendThePlane && sendTheABomb){sendThePlane=false;setLater=true;}

// if(planesToSend>0)planeXSpotsEnemy = new int[planesToSend];

// if(planesToSend>0)showPlanes = new boolean[planesToSend];

//if(planesToSend>0)for(int j=0; j<showPlanes.length; j++) showPlanes[j] = true;

if(nukesToSendEnemy>0)nukeXSpotsEnemy = new int[nukesToSendEnemy];//RIGHT HERE

if(nukesToSendEnemy>0)

{

nukeSizesEnemy = new int[nukesToSendEnemy];

showExplosionEnemy = new boolean[nukesToSendEnemy];

bombSoundCounterEnemy = new int[nukesToSendEnemy];//apply to buke ene class

for(int j=0; j<nukeSizesEnemy.length; j++) nukeSizesEnemy[j] = size;

}

if(nukesToSendEnemy>0)showNukesEnemy = new boolean[nukesToSendEnemy];

if(nukesToSendEnemy>0)dropNukesEnemy = new boolean[nukesToSendEnemy];

if(nukesToSendEnemy>0)for(int j=0; j<showNukesEnemy.length; j++) showNukesEnemy[j] = true;

if(nukesToSendEnemy>0) shouldNukeExplode = new boolean[nukesToSendEnemy];

tempContinent = continents[randomContinent];

attackingCont = randomContinent;

contToAttackEnemy = continents[randomContinent2];

startingMoney[randomContinent] -= 25000000;

startingTroops[randomContinent] -= 1000;

repaint();

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

troopPopulations[i].setText(" "+continents[i]+": "+startingTroops[i]);

}

else

{

jm3.setText(""+startingMoney[i]);

jm4.setText(""+populations[i]);

jm5.setText(""+startingTroops[i]);

}

}

}

else if(whichSubEvent==10 || enterWar)

{

if(whichSubEvent!=10){moneyCost=0; peopleDamage=0;}

while(randomContinent==whichCont || randomContinent==randomContinent2) randomContinent = (int)(Math.random()\*6);

//System.out.println(canAdvance[0][randomContinent] + " and " + canAdvance[0][randomContinent2]);

if(canAdvance[0][randomContinent]==false || canAdvance[0][randomContinent2]==false)

{

//for(int i=randomContinent; i<6; i++)

//while(allies[randomContinent][randomContinent2]==true && allies[randomContinent2][randomContinent]==true && randomContinent==whichCont && randomContinent2==whichCont){ randomContinent = (int)(Math.random()\*6);System.out.println("here");}

//System.out.println("here");

postIt=true;

event = continents[randomContinent] + threatEvent[0] + continents[randomContinent2] + threatEvent[1];

canAdvance[0][randomContinent] = true;

canAdvance[0][randomContinent2] = true;

warCheck[randomContinent][randomContinent2] = true;

warCheck[randomContinent2][randomContinent] = true;

for(int i=0; i<6; i++)

{

for(int j=0; j<6; j++)

{

if(allies[i][j])

{

// System.out.println("hi, i is: " + i + " and j is: " + j);

// if(i==randomContinent && j==randomContinent2) {warCheck[i][j] = true;warCheck[j][i] = true;canAdvance[0][i] = true;canAdvance[0][j] = true;}

// if(j==randomContinent && i==randomContinent2) {warCheck[i][j] = true;warCheck[j][i] = true;canAdvance[0][i] = true;canAdvance[0][j] = true;}

}

}

}

}

}

else postIt=false;

}

//System.out.println(whichSubEvent + " and postIt is " + postIt);

postIt=false;

eventPanel.setText(event);

if(shift)

{

for(int i=0; i<eventLogs.length; i++)

{

if(i<eventLogs.length-1)

{

eventLogs[i].remove(logTexts[i]);

eventLogs[i].remove(contButs[i]);

eventLogs[i] = new JPanel();

eventLogs[i].setLayout(new GridLayout(2,1));

logTexts[i].setText(logTexts[i+1].getText());

eventLogs[i].add(logTexts[i]);

eventLogs[i].add(contButs[i]);

eventLogs[i].setBorder(BorderFactory.createLineBorder(Color.BLACK, 3));

eventList[i]=eventList[i+1];

revalidate();

repaint();

}

}

}

logTexts[eventCount].setText(event);

eventLogs[eventCount] = new JPanel();

eventLogs[eventCount].setLayout(new GridLayout(2,1));

eventLogs[eventCount].add(logTexts[eventCount]);

eventLogs[eventCount].add(contButs[eventCount]);

eventLogs[eventCount].setBorder(BorderFactory.createLineBorder(Color.BLACK, 3));

eventList[eventCount] = new JMenuItem(""+month+"/"+day+"/"+year);

eventList[eventCount].setFont(new Font("Courier", Font.PLAIN, 20));

eventList[eventCount].addActionListener(this);

jm8.add(eventList[eventCount]);

if(jm8.getItemCount()>eventLogs.length) {jm8.remove(0);}

eventCount++;

if(eventCount==eventLogs.length) {eventCount = eventLogs.length-1;shift=true;}

if(whichSubEvent<6 || whichSubEvent>7)

{

if(whichSubEvent!=11 && whichSubEvent<8)

{

startingMoney[randomContinent] -= moneyCost;

populations[randomContinent] -= peopleDamage;

int toRemove = moneyCost%10;

int toRemove2 = peopleDamage%10;

//System.out.println((toRemove+toRemove2));

healthReduce[randomContinent] -= toRemove+toRemove2;

if(healthReduce[whichCont]<=-97) endGame=true;

if(startingMoney[whichCont]<=0) endGame=true;

if(populations[whichCont]<=0) endGame=true;

if(startingTroops[whichCont]<=0) endGame=true;

if(startingMoney[whichCont]<=0) endGame=true;

}

else if(whichSubEvent==11)

{

startingMoney[randomContinent] -= moneyCost;

int toRemove = (Math.abs(moneyCost)%10)\*2;

if(healthReduce[randomContinent]<=-10)healthReduce[randomContinent] += toRemove;

else healthReduce[randomContinent]=0;

if(healthReduce[randomContinent]>0) healthReduce[randomContinent] = 0;

defenseCount[randomContinent] += 0.4;

}

else if(whichSubEvent==9)

{

int toRemove = 10;

if(healthReduce[randomContinent]<=-10)healthReduce[randomContinent] += toRemove;

else healthReduce[randomContinent]=0;

if(healthReduce[randomContinent]>0) healthReduce[randomContinent] = 0;

startingMoney[randomContinent] -= 500000;

}

}

else if(whichSubEvent<8 && whichSubEvent>5)

{

if(whichSubEvent!=6)

{

startingMoney[randomContinent2] -= moneyCost;

populations[randomContinent2] -= peopleDamage;

int toRemove = moneyCost%25;

int toRemove2 = peopleDamage%25;

//System.out.println((toRemove+toRemove2));

healthReduce[randomContinent2] -= toRemove+toRemove2;

if(healthReduce[whichCont]<=-97) endGame=true;

if(startingMoney[whichCont]<=0) endGame=true;

if(populations[whichCont]<=0) endGame=true;

if(startingTroops[whichCont]<=0) endGame=true;

}

}

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

}

else

{

jm3.setText(""+startingMoney[i]);

jm4.setText(""+populations[i]);

}

}

}

//This method is made to recreate the main screen so that this

//code does not have to be repeated over and over again in actionPerformed()

public void resetToMainScreen()

{

putInstructions=false;

removeAll();//removes all components from the panel

setLayout(new GridLayout(3,3));//Sets grid layout 3 by 3

for(int i=0; i<9; i++)

{

if(i==1) add(title);

else if(i==4) add(startingGamePanel);

else add(new JLabel(""));//Adds empty JLabel or space

}

revalidate();

repaint();

}

//Method which finds out what the current date(month, day, year) and initializes their respective variables

//This is needed for the JMenu in the JMenuBar which will display the date based of the current time the user plays the game

public void getDateInfo()

{

year = Integer.parseInt(dateString.substring(0, dateString.indexOf('-')));

int check = Integer.parseInt(dateString.substring(dateString.indexOf('-')+1, dateString.indexOf('-')+2));

if(check==0)month = Integer.parseInt(dateString.substring(dateString.indexOf('-')+2, dateString.lastIndexOf('-')));

else month = Integer.parseInt(dateString.substring(dateString.indexOf('-')+1, dateString.lastIndexOf('-')));

day = Integer.parseInt(dateString.substring(dateString.lastIndexOf('-')+1));

}

public void calculateScore()

{

finalScore = 50000;

int toAdd1 = 0;

int toAdd2 = 0;

int toAdd3 = 0;

int toAdd4 = daysElapsed;

int toAdd5 = Math.abs(healthReduce[whichCont]);

for(int i=0; i<6; i++)

{

toAdd1+=populations[i]/10000l;//l maybe

toAdd2+=startingMoney[i]/10000l;//l maybe

toAdd3+=startingTroops[i]/10000l;//l maybe

}

finalScore = toAdd1+toAdd2+toAdd3;

finalScore += toAdd5\*100;

finalScore -= toAdd4\*100l;

score.setText("Score: "+finalScore);

score.setFont(new Font("Courier", Font.BOLD, 20));

}

//Paint component method to draw the earth background and make graphics quality better

//Draws flags showing which continents are conquered and which are not

//Draws health meters for continents

public void paintComponent(Graphics g)

{

super.paintComponent(g);//Calls super paintComponent

//Makes any graphic drawing sharp(not pixely)

((Graphics2D)g).setRenderingHint(RenderingHints.KEY\_ANTIALIASING, RenderingHints.VALUE\_ANTIALIAS\_ON);

//Makes any image sharp(not pixely)

((Graphics2D)g).setRenderingHint(RenderingHints.KEY\_RENDERING, RenderingHints.VALUE\_RENDER\_QUALITY);

g.drawImage(background, 0, 0, width, height, null);

//System.out.println("instructions is: " + putInstructions);

if(putInstructions) g.drawImage(instructionsImage, 0, 0, width, height, null);

if(putInstructions2) g.drawImage(instructionsImage2, 0, 0, width, height, null);

if(endGame)

{

removeAll();

jmb.setVisible(false);

//for(int i=0; i<healthBars.length; i++) healthBars[i]=false;

if(moneyGO) g.drawImage(moneyLoseImage, 0, 0, width, height, null);

if(peopleGO) g.drawImage(peopleLoseImage, 0, 0, width, height, null);

if(troopGO) g.drawImage(troopLoseImage, 0, 0, width, height, null);//Change these to real ones also not - 50;

if(healthGO) g.drawImage(healthLoseImage, 0, 0, width, height, null);

System.out.println("won is " + won);

if(won) g.drawImage(endGameWin, 0, 0, width, height, null);

showFlag=false;//Timer timer, timerForPlane, timerForABombs, timerForNukeEnemy, timerForPlaneEnemy;

//if(timer.isRunning())

//timer.stop();

if(timerForPlane.isRunning())timerForPlane.stop();

if(timerForABombs.isRunning())timerForABombs.stop();

if(timerForNukeEnemy.isRunning())timerForNukeEnemy.stop();

if(timerForPlaneEnemy.isRunning())timerForPlaneEnemy.stop();

gameOverCount++;

//createFile = false;

setLayout(new GridLayout(7,7));

if(won)calculateScore();

for(int i=0; i<49; i++)

{

if(i==31) add(score);

else if(i==38) add(home);

else add(new JLabel(""));

}

revalidate();

repaint();

//return;

}

//if(endGame) return;

if(showFlag)

{

if(healthBars[0]==true)

{

g.setColor(Color.BLACK);

g.fillRect(250, 230, 100, 5);

g.setColor(Color.BLUE);

g.fillRect(251, 231, 97, 3);

g.setColor(Color.RED);

g.fillRect(251, 231, 97+healthReduce[0], 3);

}

//Other continents

if(healthBars[1]==true)

{

g.setColor(Color.BLACK);

g.fillRect(400, 430, 100, 5);

g.setColor(Color.BLUE);

g.fillRect(401, 431, 97, 3);

g.setColor(Color.RED);

g.fillRect(401, 431, 97+healthReduce[1], 3);

}

if(healthBars[2]==true)

{

g.setColor(Color.BLACK);

g.fillRect(700, 180, 100, 5);

g.setColor(Color.BLUE);

g.fillRect(701, 181, 97, 3);

g.setColor(Color.RED);

g.fillRect(701, 181, 97+healthReduce[2], 3);

}

if(healthBars[3]==true)

{

g.setColor(Color.BLACK);

g.fillRect(950, 180, 100, 5);

g.setColor(Color.BLUE);

g.fillRect(951, 181, 97, 3);

g.setColor(Color.RED);

g.fillRect(951, 181, 97+healthReduce[3], 3);

}

if(healthBars[4]==true)

{

g.setColor(Color.BLACK);

g.fillRect(675, 330, 100, 5);

g.setColor(Color.BLUE);

g.fillRect(676, 331, 97, 3);

g.setColor(Color.RED);

g.fillRect(676, 331, 97+healthReduce[4], 3);

}

if(healthBars[5]==true)

{

g.setColor(Color.BLACK);

g.fillRect(1125, 505, 100, 5);

g.setColor(Color.BLUE);

g.fillRect(1126, 506, 97, 3);

g.setColor(Color.RED);

g.fillRect(1126, 506, 97+healthReduce[5], 3);

}

/\*

g.drawImage(conqueredFlag, 300, 200, 25, 25, null);//North America

g.drawImage(notConqueredFlag, 450, 400, 25, 25, null);//South America

g.drawImage(notConqueredFlag, 725, 300, 25, 25, null);//Europe

g.drawImage(notConqueredFlag, 750, 150, 25, 25, null);//Asia

g.drawImage(notConqueredFlag, 1000, 150, 25, 25, null);//Africa

g.drawImage(notConqueredFlag, 1175, 475, 25, 25, null);//Australia

\*/

if(continent.equals("North America"))

{

g.drawImage(conqueredFlag, 300, 200, 25, 25, null);//North America

if(makeBlue[1]==false)g.drawImage(notConqueredFlag, 450, 400, 25, 25, null);//South America

else g.drawImage(conqueredFlag, 450, 400, 25, 25, null);

if(makeBlue[2]==false)g.drawImage(notConqueredFlag, 750, 150, 25, 25, null);//Europe

else g.drawImage(conqueredFlag, 750, 150, 25, 25, null);

if(makeBlue[3]==false)g.drawImage(notConqueredFlag, 1000, 150, 25, 25, null);//Asia

else g.drawImage(conqueredFlag, 1000, 150, 25, 25, null);

if(makeBlue[4]==false)g.drawImage(notConqueredFlag, 725, 300, 25, 25, null);//Africa

else g.drawImage(conqueredFlag, 725, 300, 25, 25, null);

if(makeBlue[5]==false)g.drawImage(notConqueredFlag, 1175, 475, 25, 25, null);//Australia

else g.drawImage(conqueredFlag, 1175, 475, 25, 25, null);

g.setColor(Color.BLACK);

g.fillRect(250, 230, 100, 5);

g.setColor(Color.RED);

g.fillRect(251, 231, 97, 3);

g.setColor(Color.BLUE);

g.fillRect(251, 231, 97+healthReduce[0], 3);

}

//africa is 725

//europe is 750

if(continent.equals("South America"))

{

if(makeBlue[0]==false)g.drawImage(notConqueredFlag, 300, 200, 25, 25, null);

else g.drawImage(conqueredFlag, 300, 200, 25, 25, null);//North America

g.drawImage(conqueredFlag, 450, 400, 25, 25, null);

if(makeBlue[2]==false)g.drawImage(notConqueredFlag, 750, 150, 25, 25, null);

else g.drawImage(conqueredFlag, 750, 150, 25, 25, null);

if(makeBlue[3]==false)g.drawImage(notConqueredFlag, 1000, 150, 25, 25, null);

else g.drawImage(conqueredFlag, 1000, 150, 25, 25, null);

if(makeBlue[4]==false)g.drawImage(notConqueredFlag, 725, 300, 25, 25, null);

else g.drawImage(conqueredFlag, 725, 300, 25, 25, null);

if(makeBlue[5]==false)g.drawImage(notConqueredFlag, 1175, 475, 25, 25, null);

else g.drawImage(conqueredFlag, 1175, 475, 25, 25, null);

g.setColor(Color.BLACK);

g.fillRect(400, 430, 100, 5);

g.setColor(Color.RED);

g.fillRect(401, 431, 97, 3);

g.setColor(Color.BLUE);

g.fillRect(401, 431, 97+healthReduce[1], 3);

}

if(continent.equals("Europe"))

{

if(makeBlue[0]==false)g.drawImage(notConqueredFlag, 300, 200, 25, 25, null);

else g.drawImage(conqueredFlag, 300, 200, 25, 25, null);//North America

if(makeBlue[1]==false)g.drawImage(notConqueredFlag, 450, 400, 25, 25, null);//South America

else g.drawImage(conqueredFlag, 450, 400, 25, 25, null);

g.drawImage(conqueredFlag, 750, 150, 25, 25, null);

if(makeBlue[3]==false)g.drawImage(notConqueredFlag, 1000, 150, 25, 25, null);

else g.drawImage(conqueredFlag, 1000, 150, 25, 25, null);

if(makeBlue[4]==false)g.drawImage(notConqueredFlag, 750, 300, 25, 25, null);

else g.drawImage(conqueredFlag, 725, 300, 25, 25, null);

if(makeBlue[5]==false)g.drawImage(notConqueredFlag, 1175, 475, 25, 25, null);

else g.drawImage(conqueredFlag, 1175, 475, 25, 25, null);

g.setColor(Color.BLACK);

g.fillRect(700, 180, 100, 5);

g.setColor(Color.RED);

g.fillRect(701, 181, 97, 3);

g.setColor(Color.BLUE);

g.fillRect(701, 181, 97+healthReduce[2], 3);

}

if(continent.equals("Asia"))

{

if(makeBlue[0]==false)g.drawImage(notConqueredFlag, 300, 200, 25, 25, null);

else g.drawImage(conqueredFlag, 300, 200, 25, 25, null);//North America

if(makeBlue[1]==false)g.drawImage(notConqueredFlag, 450, 400, 25, 25, null);//South America

else g.drawImage(conqueredFlag, 450, 400, 25, 25, null);

if(makeBlue[2]==false)g.drawImage(notConqueredFlag, 750, 150, 25, 25, null);

else g.drawImage(conqueredFlag, 750, 150, 25, 25, null);

g.drawImage(conqueredFlag, 1000, 150, 25, 25, null);

if(makeBlue[4]==false)g.drawImage(notConqueredFlag, 750, 300, 25, 25, null);

else g.drawImage(conqueredFlag, 725, 300, 25, 25, null);

if(makeBlue[5]==false)g.drawImage(notConqueredFlag, 1175, 475, 25, 25, null);

else g.drawImage(conqueredFlag, 1175, 475, 25, 25, null);

g.setColor(Color.BLACK);

g.fillRect(950, 180, 100, 5);

g.setColor(Color.RED);

g.fillRect(951, 181, 97, 3);

g.setColor(Color.BLUE);

g.fillRect(951, 181, 97+healthReduce[3], 3);

}

if(continent.equals("Africa"))

{

if(makeBlue[0]==false)g.drawImage(notConqueredFlag, 300, 200, 25, 25, null);

else g.drawImage(conqueredFlag, 300, 200, 25, 25, null);//North America

if(makeBlue[1]==false)g.drawImage(notConqueredFlag, 450, 400, 25, 25, null);//South America

else g.drawImage(conqueredFlag, 450, 400, 25, 25, null);

if(makeBlue[2]==false)g.drawImage(notConqueredFlag, 750, 150, 25, 25, null);

else g.drawImage(conqueredFlag, 750, 150, 25, 25, null);

if(makeBlue[3]==false)g.drawImage(notConqueredFlag, 1000, 150, 25, 25, null);

else g.drawImage(conqueredFlag, 1000, 150, 25, 25, null);

g.drawImage(conqueredFlag, 725, 300, 25, 25, null);

if(makeBlue[5]==false)g.drawImage(notConqueredFlag, 1175, 475, 25, 25, null);

else g.drawImage(conqueredFlag, 1175, 475, 25, 25, null);

g.setColor(Color.BLACK);

g.fillRect(675, 330, 100, 5);

g.setColor(Color.RED);

g.fillRect(676, 331, 97, 3);

g.setColor(Color.BLUE);

g.fillRect(676, 331, 97+healthReduce[4], 3);

}

if(continent.equals("Australia"))

{

if(makeBlue[0]==false)g.drawImage(notConqueredFlag, 300, 200, 25, 25, null);

else g.drawImage(conqueredFlag, 300, 200, 25, 25, null);//North America

if(makeBlue[1]==false)g.drawImage(notConqueredFlag, 450, 400, 25, 25, null);//South America

else g.drawImage(conqueredFlag, 450, 400, 25, 25, null);

if(makeBlue[2]==false)g.drawImage(notConqueredFlag, 750, 150, 25, 25, null);

else g.drawImage(conqueredFlag, 750, 150, 25, 25, null);

if(makeBlue[3]==false)g.drawImage(notConqueredFlag, 1000, 150, 25, 25, null);

else g.drawImage(conqueredFlag, 1000, 150, 25, 25, null);

if(makeBlue[4]==false)g.drawImage(notConqueredFlag, 725, 300, 25, 25, null);

else g.drawImage(conqueredFlag, 725, 300, 25, 25, null);

g.drawImage(conqueredFlag, 1175, 475, 25, 25, null);

g.setColor(Color.BLACK);

g.fillRect(1125, 505, 100, 5);

g.setColor(Color.RED);

g.fillRect(1126, 506, 97, 3);

g.setColor(Color.BLUE);

g.fillRect(1126, 506, 97+healthReduce[5], 3);

}

if(sendThePlane)

{

timerForPlane.start();

if(continent.equals("North America"))

{

if(firstTime)

{

for(int i=0; i<planeXSpots.length; i++)

{

//if(contToAttack.equals("South America") || contToAttack.equals("Europe") || contToAttack.equals("Asia"))

planeXSpots[i]=350;//\*

}

//if(contToAttack.equals("South America") || contToAttack.equals("Europe") || contToAttack.equals("Asia"))

planeY=200;//\*

firstTime=false;

}

g2d = (Graphics2D)g;

old = g2d.getTransform();

if(contToAttack.equals("South America"))

{

g2d.translate(375, -200); //\*

g2d.rotate(Math.toRadians(60));//\*

}

if(contToAttack.equals("Europe") || contToAttack.equals("Asia"))

{

g2d.translate(-15, 40); //\*

g2d.rotate(Math.toRadians(-5));//\*

}

if(contToAttack.equals("Africa"))

{

g2d.translate(45, -45); //\*

g2d.rotate(Math.toRadians(10));//\*

}

if(contToAttack.equals("Australia"))

{

g2d.translate(80, -80); //\*

g2d.rotate(Math.toRadians(17));//\*

}

for(int i=0; i<planesToSend; i++)//350

{

if(showPlanes[i])

{

g2d.drawImage(plane, planeXSpots[i], planeY, null);

//g2d.drawImage(aBomb, planeXSpots[i], planeY, null);

}

//draw the bombs here

}

g2d.setTransform(old);

}

if(continent.equals("South America"))

{

if(firstTime)

{

for(int i=0; i<planeXSpots.length; i++)

{

//if(contToAttack.equals("South America") || contToAttack.equals("Europe") || contToAttack.equals("Asia"))

planeXSpots[i]=100;//\*

}

//if(contToAttack.equals("South America") || contToAttack.equals("Europe") || contToAttack.equals("Asia"))

planeY=300;//\*

firstTime=false;

}

g2d = (Graphics2D)g;

old = g2d.getTransform();

if(contToAttack.equals("North America"))

{

g2d.translate(200, 650); //\*

g2d.rotate(Math.toRadians(-120));//\*

}

if(contToAttack.equals("Europe"))

{

g2d.translate(175, 150); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-30));//\*

}

if(contToAttack.equals("Asia")||contToAttack.equals("Africa"))

{

g2d.translate(225, 100); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-15));//\*

}

if(contToAttack.equals("Australia"))

{

g2d.translate(325, 75); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(5));//\*

}

for(int i=0; i<planesToSend; i++)//350

{

if(showPlanes[i])g2d.drawImage(plane, planeXSpots[i], planeY, null);

}

g2d.setTransform(old);

}

if(continent.equals("Europe"))

{

if(firstTime)

{

for(int i=0; i<planeXSpots.length; i++)

{

//if(contToAttack.equals("South America") || contToAttack.equals("Europe") || contToAttack.equals("Asia"))

planeXSpots[i]=100;//\*

}

//if(contToAttack.equals("South America") || contToAttack.equals("Europe") || contToAttack.equals("Asia"))

planeY=300;//\*

firstTime=false;

}

g2d = (Graphics2D)g;

old = g2d.getTransform();

if(contToAttack.equals("North America"))

{

g2d.translate(900, 500); //\*

g2d.rotate(Math.toRadians(-185));//\*

}

if(contToAttack.equals("South America"))

{

g2d.translate(1050, 325); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-225));//\*

}

if(contToAttack.equals("Asia"))

{

g2d.translate(625, -150); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(0));//\*

}

if(contToAttack.equals("Africa"))

{

g2d.translate(1075, 75); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(90));//\*

}

if(contToAttack.equals("Australia"))

{

g2d.translate(900, -150); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(40));//\*

}

for(int i=0; i<planesToSend; i++)//350

{

if(showPlanes[i])g2d.drawImage(plane, planeXSpots[i], planeY, null);

}

g2d.setTransform(old);

}

if(continent.equals("Asia"))

{

if(firstTime)

{

for(int i=0; i<planeXSpots.length; i++)

{

//if(contToAttack.equals("South America") || contToAttack.equals("Europe") || contToAttack.equals("Asia"))

planeXSpots[i]=100;//\*

}

//if(contToAttack.equals("South America") || contToAttack.equals("Europe") || contToAttack.equals("Asia"))

planeY=300;//\*

firstTime=false;

}

g2d = (Graphics2D)g;

old = g2d.getTransform();

if(contToAttack.equals("North America"))

{

g2d.translate(1100, 500); //\*

g2d.rotate(Math.toRadians(-185));//\*

}

if(contToAttack.equals("South America"))

{

g2d.translate(1200, 450); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-200));//\*

}

if(contToAttack.equals("Europe"))

{

g2d.translate(1100, 500); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-180));//\*

}

if(contToAttack.equals("Africa"))

{

g2d.translate(1300, 300); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-225));//\*

}

if(contToAttack.equals("Australia"))

{

g2d.translate(1350, -25); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(75));//\*

}

for(int i=0; i<planesToSend; i++)//350

{

if(showPlanes[i])g2d.drawImage(plane, planeXSpots[i], planeY, null);

}

g2d.setTransform(old);

}

if(continent.equals("Africa"))

{

if(firstTime)

{

for(int i=0; i<planeXSpots.length; i++)

{

//if(contToAttack.equals("South America") || contToAttack.equals("Europe") || contToAttack.equals("Asia"))

planeXSpots[i]=100;//\*

}

//if(contToAttack.equals("South America") || contToAttack.equals("Europe") || contToAttack.equals("Asia"))

planeY=300;//\*

firstTime=false;

}

g2d = (Graphics2D)g;

old = g2d.getTransform();

if(contToAttack.equals("North America"))

{

g2d.translate(700,650); //\*

g2d.rotate(Math.toRadians(-160));//\*

}

if(contToAttack.equals("South America"))

{

g2d.translate(900, 600); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-200));//\*

}

if(contToAttack.equals("Europe"))

{

g2d.translate(450, 500); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-90));//\*

}

if(contToAttack.equals("Asia"))

{

g2d.translate(450, 200); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-45));//\*

}

if(contToAttack.equals("Australia"))

{

g2d.translate(850, 0); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(25));//\*

}

for(int i=0; i<planesToSend; i++)//350

{

if(showPlanes[i])g2d.drawImage(plane, planeXSpots[i], planeY, null);

}

g2d.setTransform(old);

}

if(continent.equals("Australia"))

{

if(firstTime)

{

for(int i=0; i<planeXSpots.length; i++)

{

//if(contToAttack.equals("South America") || contToAttack.equals("Europe") || contToAttack.equals("Asia"))

planeXSpots[i]=100;//\*

}

//if(contToAttack.equals("South America") || contToAttack.equals("Europe") || contToAttack.equals("Asia"))

planeY=300;//\*

firstTime=false;

}

g2d = (Graphics2D)g;

old = g2d.getTransform();

if(contToAttack.equals("North America")||contToAttack.equals("Africa"))

{

g2d.translate(1200,825); //\*

g2d.rotate(Math.toRadians(-160));//\*

}

if(contToAttack.equals("South America"))

{

g2d.translate(1200, 825); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-170));//\*

}

if(contToAttack.equals("Europe"))

{

g2d.translate(1100, 825); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-140));//\*

}

if(contToAttack.equals("Asia"))

{

g2d.translate(900, 700); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-110));//\*

}

for(int i=0; i<planesToSend; i++)//350

{

if(showPlanes[i])g2d.drawImage(plane, planeXSpots[i], planeY, null);

}

g2d.setTransform(old);

}

}

if(sendTheABomb)

{

timerForABombs.start();

if(continent.equals("North America"))

{

if(firstTime)

{

for(int i=0; i<aBombXSpots.length; i++)

{

//if(contToAttack.equals("South America") || contToAttack.equals("Europe") || contToAttack.equals("Asia"))

aBombXSpots[i]=350;//\*

}

//if(contToAttack.equals("South America") || contToAttack.equals("Europe") || contToAttack.equals("Asia"))

planeY=200;//\*

firstTime=false;

}

g2d = (Graphics2D)g;

old = g2d.getTransform();

if(contToAttack.equals("South America"))

{

g2d.translate(375, -200); //\*

g2d.rotate(Math.toRadians(60));//\*

}

if(contToAttack.equals("Europe") || contToAttack.equals("Asia"))

{

g2d.translate(-15, 40); //\*

g2d.rotate(Math.toRadians(-5));//\*

}

if(contToAttack.equals("Africa"))

{

g2d.translate(45, -45); //\*

g2d.rotate(Math.toRadians(10));//\*

}

if(contToAttack.equals("Australia"))

{

g2d.translate(80, -80); //\*

g2d.rotate(Math.toRadians(17));//\*

}

for(int i=0; i<aBombsToSend; i++)//350

{

if(showABombs[i])

{

g2d.drawImage(aBombPlane, aBombXSpots[i], planeY, null);

}

else

{

aBombSizes[i]--;

if(aBombSizes[i]<=0)

{

aBombSizes[i]=0;

shouldABombExplode[i] = true;

if(showExplosion[i] && bombSoundCounter[i]<5)

{

//System.out.println("i is " + i);

g.drawImage(explosion, aBombXSpots[i]-50, planeY, null);

bombSoundCounter[i]++;

if(bombSoundCounter[i]==5)showExplosion[i]=false;

}

//System.out.println("i is " + i +" and "+bombSoundCounter[i]);

//if(dropABombs[i] && bombSoundCounter[i]<2) {g.drawImage(explosion, aBombXSpots[i], planeY, null);bombSoundCounter[i]++;}

}

//if(shouldABombExplode[i]) g2d.drawImage(explosion, aBombXSpots[i], planeY, 25, 25, null);

g2d.drawImage(aBomb, aBombXSpots[i], planeY, aBombSizes[i], aBombSizes[i], null);//aBombSizes[i], aBombSizes[i], null);

}

}

g2d.setTransform(old);

}

if(continent.equals("South America"))

{

if(firstTime)

{

for(int i=0; i<aBombXSpots.length; i++)

{

//if(contToAttack.equals("South America") || contToAttack.equals("Europe") || contToAttack.equals("Asia"))

aBombXSpots[i]=100;//\*

}

//if(contToAttack.equals("South America") || contToAttack.equals("Europe") || contToAttack.equals("Asia"))

planeY=300;//\*

firstTime=false;

}

g2d = (Graphics2D)g;

old = g2d.getTransform();

if(contToAttack.equals("North America"))

{

g2d.translate(200, 650); //\*

g2d.rotate(Math.toRadians(-120));//\*

}

if(contToAttack.equals("Europe"))

{

g2d.translate(175, 150); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-30));//\*

}

if(contToAttack.equals("Asia")||contToAttack.equals("Africa"))

{

g2d.translate(225, 100); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-15));//\*

}

if(contToAttack.equals("Australia"))

{

g2d.translate(325, 75); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(5));//\*

}

for(int i=0; i<aBombsToSend; i++)//350

{

if(showABombs[i])g2d.drawImage(aBombPlane, aBombXSpots[i], planeY, null);

else

{

aBombSizes[i]--;

if(aBombSizes[i]<=0)

{

aBombSizes[i]=0;

shouldABombExplode[i] = true;

if(showExplosion[i] && bombSoundCounter[i]<5)

{

//System.out.println("i is " + i);

g.drawImage(explosion, aBombXSpots[i]-50, planeY, null);

bombSoundCounter[i]++;

if(bombSoundCounter[i]==5)showExplosion[i]=false;

}

//System.out.println("i is " + i +" and "+bombSoundCounter[i]);

//if(dropABombs[i] && bombSoundCounter[i]<2) {g.drawImage(explosion, aBombXSpots[i], planeY, null);bombSoundCounter[i]++;}

}

g2d.drawImage(aBomb, aBombXSpots[i], planeY, aBombSizes[i], aBombSizes[i], null);//aBombSizes[i], aBombSizes[i], null);

}

}

g2d.setTransform(old);

}

if(continent.equals("Europe"))

{

if(firstTime)

{

for(int i=0; i<aBombXSpots.length; i++)

{

//if(contToAttack.equals("South America") || contToAttack.equals("Europe") || contToAttack.equals("Asia"))

aBombXSpots[i]=100;//\*

}

//if(contToAttack.equals("South America") || contToAttack.equals("Europe") || contToAttack.equals("Asia"))

planeY=300;//\*

firstTime=false;

}

g2d = (Graphics2D)g;

old = g2d.getTransform();

if(contToAttack.equals("North America"))

{

g2d.translate(900, 500); //\*

g2d.rotate(Math.toRadians(-185));//\*

}

if(contToAttack.equals("South America"))

{

g2d.translate(1050, 325); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-225));//\*

}

if(contToAttack.equals("Asia"))

{

g2d.translate(625, -150); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(0));//\*

}

if(contToAttack.equals("Africa"))

{

g2d.translate(1075, 75); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(90));//\*

}

if(contToAttack.equals("Australia"))

{

g2d.translate(900, -150); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(40));//\*

}

for(int i=0; i<aBombsToSend; i++)//350

{

if(showABombs[i])g2d.drawImage(aBombPlane, aBombXSpots[i], planeY, null);

else

{

aBombSizes[i]--;

if(aBombSizes[i]<=0)

{

aBombSizes[i]=0;

shouldABombExplode[i] = true;

if(showExplosion[i] && bombSoundCounter[i]<5)

{

// System.out.println("i is " + i);

g.drawImage(explosion, aBombXSpots[i]-50, planeY, null);

bombSoundCounter[i]++;

if(bombSoundCounter[i]==5)showExplosion[i]=false;

}

//System.out.println("i is " + i +" and "+bombSoundCounter[i]);

//if(dropABombs[i] && bombSoundCounter[i]<2) {g.drawImage(explosion, aBombXSpots[i], planeY, null);bombSoundCounter[i]++;}

}

g2d.drawImage(aBomb, aBombXSpots[i], planeY, aBombSizes[i], aBombSizes[i], null);//aBombSizes[i], aBombSizes[i], null);

}

}

g2d.setTransform(old);

}

if(continent.equals("Asia"))

{

if(firstTime)

{

for(int i=0; i<aBombXSpots.length; i++)

{

//if(contToAttack.equals("South America") || contToAttack.equals("Europe") || contToAttack.equals("Asia"))

aBombXSpots[i]=100;//\*

}

//if(contToAttack.equals("South America") || contToAttack.equals("Europe") || contToAttack.equals("Asia"))

planeY=300;//\*

firstTime=false;

}

g2d = (Graphics2D)g;

old = g2d.getTransform();

if(contToAttack.equals("North America"))

{

g2d.translate(1100, 500); //\*

g2d.rotate(Math.toRadians(-185));//\*

}

if(contToAttack.equals("South America"))

{

g2d.translate(1200, 450); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-200));//\*

}

if(contToAttack.equals("Europe"))

{

g2d.translate(1100, 500); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-180));//\*

}

if(contToAttack.equals("Africa"))

{

g2d.translate(1300, 300); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-225));//\*

}

if(contToAttack.equals("Australia"))

{

g2d.translate(1350, -25); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(75));//\*

}

for(int i=0; i<aBombsToSend; i++)//350

{

if(showABombs[i])g2d.drawImage(aBombPlane, aBombXSpots[i], planeY, null);

else

{

aBombSizes[i]--;

if(aBombSizes[i]<=0)

{

aBombSizes[i]=0;

shouldABombExplode[i] = true;

if(showExplosion[i] && bombSoundCounter[i]<5)

{

//System.out.println("i is " + i);

g.drawImage(explosion, aBombXSpots[i]-50, planeY, null);

bombSoundCounter[i]++;

if(bombSoundCounter[i]==5)showExplosion[i]=false;

}

//System.out.println("i is " + i +" and "+bombSoundCounter[i]);

//if(dropABombs[i] && bombSoundCounter[i]<2) {g.drawImage(explosion, aBombXSpots[i], planeY, null);bombSoundCounter[i]++;}

}

g2d.drawImage(aBomb, aBombXSpots[i], planeY, aBombSizes[i], aBombSizes[i], null);//aBombSizes[i], aBombSizes[i], null);

}

}

g2d.setTransform(old);

}

if(continent.equals("Africa"))

{

if(firstTime)

{

for(int i=0; i<aBombXSpots.length; i++)

{

//if(contToAttack.equals("South America") || contToAttack.equals("Europe") || contToAttack.equals("Asia"))

aBombXSpots[i]=100;//\*

}

//if(contToAttack.equals("South America") || contToAttack.equals("Europe") || contToAttack.equals("Asia"))

planeY=300;//\*

firstTime=false;

}

g2d = (Graphics2D)g;

old = g2d.getTransform();

if(contToAttack.equals("North America"))

{

g2d.translate(700,650); //\*

g2d.rotate(Math.toRadians(-160));//\*

}

if(contToAttack.equals("South America"))

{

g2d.translate(900, 600); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-200));//\*

}

if(contToAttack.equals("Europe"))

{

g2d.translate(450, 500); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-90));//\*

}

if(contToAttack.equals("Asia"))

{

g2d.translate(450, 200); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-45));//\*

}

if(contToAttack.equals("Australia"))

{

g2d.translate(850, 0); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(25));//\*

}

for(int i=0; i<aBombsToSend; i++)//350

{

if(showABombs[i])g2d.drawImage(aBombPlane, aBombXSpots[i], planeY, null);

else

{

aBombSizes[i]--;

if(aBombSizes[i]<=0)

{

aBombSizes[i]=0;

shouldABombExplode[i] = true;

if(showExplosion[i] && bombSoundCounter[i]<5)

{

//System.out.println("i is " + i);

g.drawImage(explosion, aBombXSpots[i]-50, planeY, null);

bombSoundCounter[i]++;

if(bombSoundCounter[i]==5)showExplosion[i]=false;

}

//System.out.println("i is " + i +" and "+bombSoundCounter[i]);

//if(dropABombs[i] && bombSoundCounter[i]<2) {g.drawImage(explosion, aBombXSpots[i], planeY, null);bombSoundCounter[i]++;}

}

g2d.drawImage(aBomb, aBombXSpots[i], planeY, aBombSizes[i], aBombSizes[i], null);//aBombSizes[i], aBombSizes[i], null);

}

}

g2d.setTransform(old);

}

if(continent.equals("Australia"))

{

if(firstTime)

{

for(int i=0; i<aBombXSpots.length; i++)

{

//if(contToAttack.equals("South America") || contToAttack.equals("Europe") || contToAttack.equals("Asia"))

aBombXSpots[i]=100;//\*

}

//if(contToAttack.equals("South America") || contToAttack.equals("Europe") || contToAttack.equals("Asia"))

planeY=300;//\*

firstTime=false;

}

g2d = (Graphics2D)g;

old = g2d.getTransform();

if(contToAttack.equals("North America")||contToAttack.equals("Africa"))

{

g2d.translate(1200,825); //\*

g2d.rotate(Math.toRadians(-160));//\*

}

if(contToAttack.equals("South America"))

{

g2d.translate(1200, 825); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-170));//\*

}

if(contToAttack.equals("Europe"))

{

g2d.translate(1100, 825); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-140));//\*

}

if(contToAttack.equals("Asia"))

{

g2d.translate(900, 700); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-110));//\*

}

for(int i=0; i<aBombsToSend; i++)//350

{

if(showABombs[i])g2d.drawImage(aBombPlane, aBombXSpots[i], planeY, null);

else

{

aBombSizes[i]--;

if(aBombSizes[i]<=0)

{

aBombSizes[i]=0;

shouldABombExplode[i] = true;

if(showExplosion[i] && bombSoundCounter[i]<5)

{

//System.out.println("i is " + i);

g.drawImage(explosion, aBombXSpots[i]-50, planeY, null);

bombSoundCounter[i]++;

if(bombSoundCounter[i]==5)showExplosion[i]=false;

}

//System.out.println("i is " + i +" and "+bombSoundCounter[i]);

//if(dropABombs[i] && bombSoundCounter[i]<2) {g.drawImage(explosion, aBombXSpots[i], planeY, null);bombSoundCounter[i]++;}

}

g2d.drawImage(aBomb, aBombXSpots[i], planeY, aBombSizes[i], aBombSizes[i], null);//aBombSizes[i], aBombSizes[i], null);

}

}

g2d.setTransform(old);

}

}

else if(setLater) {sendThePlane=true;setLater=false;}

if(sendTheNukeEnemy && sendThePlane==false && sendTheABomb==false && sendThePlaneEnemy==false)

{

timerForNukeEnemy.start();

if(tempContinent.equals("North America"))

{

if(firstTimeEnemy)

{

for(int i=0; i<nukeXSpotsEnemy.length; i++)

{

//if(contToAttackEnemy.equals("South America") || contToAttackEnemy.equals("Europe") || contToAttackEnemy.equals("Asia"))

nukeXSpotsEnemy[i]=350;//\*

}

//if(contToAttackEnemy.equals("South America") || contToAttackEnemy.equals("Europe") || contToAttackEnemy.equals("Asia"))

planeY=200;//\*

firstTimeEnemy=false;

}

g2d = (Graphics2D)g;

old = g2d.getTransform();

if(contToAttackEnemy.equals("South America"))

{

g2d.translate(375, -200); //\*

g2d.rotate(Math.toRadians(60));//\*

}

if(contToAttackEnemy.equals("Europe") || contToAttackEnemy.equals("Asia"))

{

g2d.translate(-15, 40); //\*

g2d.rotate(Math.toRadians(-5));//\*

}

if(contToAttackEnemy.equals("Africa"))

{

g2d.translate(45, -45); //\*

g2d.rotate(Math.toRadians(10));//\*

}

if(contToAttackEnemy.equals("Australia"))

{

g2d.translate(80, -80); //\*

g2d.rotate(Math.toRadians(17));//\*

}

/\*

if(aBombSizes[i]<=0)

{

aBombSizes[i]=0;

shouldABombExplode[i] = true;

if(showExplosion[i] && bombSoundCounter[i]<5)

{

System.out.println("i is " + i);

g.drawImage(explosion, aBombXSpots[i]-50, planeY, null);

bombSoundCounter[i]++;

if(bombSoundCounter[i]==5)showExplosion[i]=false;

}

//System.out.println("i is " + i +" and "+bombSoundCounter[i]);

//if(dropABombs[i] && bombSoundCounter[i]<2) {g.drawImage(explosion, aBombXSpots[i], planeY, null);bombSoundCounter[i]++;}

}

\*/

for(int i=0; i<nukesToSendEnemy; i++)//350

{

if(showNukesEnemy[i])

{

g2d.drawImage(aBombPlane, nukeXSpotsEnemy[i], planeY, null);

}

else

{

nukeSizesEnemy[i]--;

if(nukeSizesEnemy[i]<=0)

{

nukeSizesEnemy[i]=0;

shouldNukeExplode[i] = true;

if(showExplosionEnemy[i] && bombSoundCounterEnemy[i]<5)

{

//System.out.println("i is " + i);

g.drawImage(explosion, nukeXSpotsEnemy[i]-50, planeY, null);

bombSoundCounterEnemy[i]++;

if(bombSoundCounterEnemy[i]==5)showExplosionEnemy[i]=false;

}

//System.out.println("i is " + i +" and "+bombSoundCounter[i]);

//if(dropABombs[i] && bombSoundCounter[i]<2) {g.drawImage(explosion, aBombXSpots[i], planeY, null);bombSoundCounter[i]++;}

}

//if(shouldNukeExplode[i]) g2d.drawImage(explosion, nukeXSpotsEnemy[i], planeY, 25, 25, null);

g2d.drawImage(aBomb, nukeXSpotsEnemy[i], planeY, nukeSizesEnemy[i], nukeSizesEnemy[i], null);//nukeSizesEnemy[i], nukeSizesEnemy[i], null);

}

}

g2d.setTransform(old);

}

if(tempContinent.equals("South America"))

{

if(firstTimeEnemy)

{

for(int i=0; i<nukeXSpotsEnemy.length; i++)

{

//if(contToAttackEnemy.equals("South America") || contToAttackEnemy.equals("Europe") || contToAttackEnemy.equals("Asia"))

nukeXSpotsEnemy[i]=100;//\*

}

//if(contToAttackEnemy.equals("South America") || contToAttackEnemy.equals("Europe") || contToAttackEnemy.equals("Asia"))

planeY=300;//\*

firstTimeEnemy=false;

}

g2d = (Graphics2D)g;

old = g2d.getTransform();

if(contToAttackEnemy.equals("North America"))

{

g2d.translate(200, 650); //\*

g2d.rotate(Math.toRadians(-120));//\*

}

if(contToAttackEnemy.equals("Europe"))

{

g2d.translate(175, 150); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-30));//\*

}

if(contToAttackEnemy.equals("Asia")||contToAttackEnemy.equals("Africa"))

{

g2d.translate(225, 100); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-15));//\*

}

if(contToAttackEnemy.equals("Australia"))

{

g2d.translate(325, 75); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(5));//\*

}

for(int i=0; i<nukesToSendEnemy; i++)//350

{

if(showNukesEnemy[i])g2d.drawImage(aBombPlane, nukeXSpotsEnemy[i], planeY, null);

else

{

nukeSizesEnemy[i]--;

if(nukeSizesEnemy[i]<=0)

{

nukeSizesEnemy[i]=0;

shouldNukeExplode[i] = true;

if(showExplosionEnemy[i] && bombSoundCounterEnemy[i]<5)

{

//System.out.println("i is " + i);

g.drawImage(explosion, nukeXSpotsEnemy[i]-50, planeY, null);

bombSoundCounterEnemy[i]++;

if(bombSoundCounterEnemy[i]==5)showExplosionEnemy[i]=false;

}

//System.out.println("i is " + i +" and "+bombSoundCounter[i]);

//if(dropABombs[i] && bombSoundCounter[i]<2) {g.drawImage(explosion, aBombXSpots[i], planeY, null);bombSoundCounter[i]++;}

}

g2d.drawImage(aBomb, nukeXSpotsEnemy[i], planeY, nukeSizesEnemy[i], nukeSizesEnemy[i], null);//nukeSizesEnemy[i], nukeSizesEnemy[i], null);

}

}

g2d.setTransform(old);

}

if(tempContinent.equals("Europe"))

{

if(firstTimeEnemy)

{

for(int i=0; i<nukeXSpotsEnemy.length; i++)

{

//if(contToAttackEnemy.equals("South America") || contToAttackEnemy.equals("Europe") || contToAttackEnemy.equals("Asia"))

nukeXSpotsEnemy[i]=100;//\*

}

//if(contToAttackEnemy.equals("South America") || contToAttackEnemy.equals("Europe") || contToAttackEnemy.equals("Asia"))

planeY=300;//\*

firstTimeEnemy=false;

}

g2d = (Graphics2D)g;

old = g2d.getTransform();

if(contToAttackEnemy.equals("North America"))

{

g2d.translate(900, 500); //\*

g2d.rotate(Math.toRadians(-185));//\*

}

if(contToAttackEnemy.equals("South America"))

{

g2d.translate(1050, 325); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-225));//\*

}

if(contToAttackEnemy.equals("Asia"))

{

g2d.translate(625, -150); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(0));//\*

}

if(contToAttackEnemy.equals("Africa"))

{

g2d.translate(1075, 75); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(90));//\*

}

if(contToAttackEnemy.equals("Australia"))

{

g2d.translate(900, -150); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(40));//\*

}

for(int i=0; i<nukesToSendEnemy; i++)//350

{

if(showNukesEnemy[i])g2d.drawImage(aBombPlane, nukeXSpotsEnemy[i], planeY, null);

else

{

nukeSizesEnemy[i]--;

if(nukeSizesEnemy[i]<=0)

{

nukeSizesEnemy[i]=0;

shouldNukeExplode[i] = true;

if(showExplosionEnemy[i] && bombSoundCounterEnemy[i]<5)

{

//System.out.println("i is " + i);

g.drawImage(explosion, nukeXSpotsEnemy[i]-50, planeY, null);

bombSoundCounterEnemy[i]++;

if(bombSoundCounterEnemy[i]==5)showExplosionEnemy[i]=false;

}

//System.out.println("i is " + i +" and "+bombSoundCounter[i]);

//if(dropABombs[i] && bombSoundCounter[i]<2) {g.drawImage(explosion, aBombXSpots[i], planeY, null);bombSoundCounter[i]++;}

}

g2d.drawImage(aBomb, nukeXSpotsEnemy[i], planeY, nukeSizesEnemy[i], nukeSizesEnemy[i], null);//nukeSizesEnemy[i], nukeSizesEnemy[i], null);

}

}

g2d.setTransform(old);

}

if(tempContinent.equals("Asia"))

{

if(firstTimeEnemy)

{

for(int i=0; i<nukeXSpotsEnemy.length; i++)

{

//if(contToAttackEnemy.equals("South America") || contToAttackEnemy.equals("Europe") || contToAttackEnemy.equals("Asia"))

nukeXSpotsEnemy[i]=100;//\*

}

//if(contToAttackEnemy.equals("South America") || contToAttackEnemy.equals("Europe") || contToAttackEnemy.equals("Asia"))

planeY=300;//\*

firstTimeEnemy=false;

}

g2d = (Graphics2D)g;

old = g2d.getTransform();

if(contToAttackEnemy.equals("North America"))

{

g2d.translate(1100, 500); //\*

g2d.rotate(Math.toRadians(-185));//\*

}

if(contToAttackEnemy.equals("South America"))

{

g2d.translate(1200, 450); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-200));//\*

}

if(contToAttackEnemy.equals("Europe"))

{

g2d.translate(1100, 500); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-180));//\*

}

if(contToAttackEnemy.equals("Africa"))

{

g2d.translate(1300, 300); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-225));//\*

}

if(contToAttackEnemy.equals("Australia"))

{

g2d.translate(1350, -25); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(75));//\*

}

for(int i=0; i<nukesToSendEnemy; i++)//350

{

if(showNukesEnemy[i])g2d.drawImage(aBombPlane, nukeXSpotsEnemy[i], planeY, null);

else

{

nukeSizesEnemy[i]--;

if(nukeSizesEnemy[i]<=0)

{

nukeSizesEnemy[i]=0;

shouldNukeExplode[i] = true;

if(showExplosionEnemy[i] && bombSoundCounterEnemy[i]<5)

{

// System.out.println("i is " + i);

g.drawImage(explosion, nukeXSpotsEnemy[i]-50, planeY, null);

bombSoundCounterEnemy[i]++;

if(bombSoundCounterEnemy[i]==5)showExplosionEnemy[i]=false;

}

//System.out.println("i is " + i +" and "+bombSoundCounter[i]);

//if(dropABombs[i] && bombSoundCounter[i]<2) {g.drawImage(explosion, aBombXSpots[i], planeY, null);bombSoundCounter[i]++;}

}

g2d.drawImage(aBomb, nukeXSpotsEnemy[i], planeY, nukeSizesEnemy[i], nukeSizesEnemy[i], null);//nukeSizesEnemy[i], nukeSizesEnemy[i], null);

}

}

g2d.setTransform(old);

}

if(tempContinent.equals("Africa"))

{

if(firstTimeEnemy)

{

for(int i=0; i<nukeXSpotsEnemy.length; i++)

{

//if(contToAttackEnemy.equals("South America") || contToAttackEnemy.equals("Europe") || contToAttackEnemy.equals("Asia"))

nukeXSpotsEnemy[i]=100;//\*

}

//if(contToAttackEnemy.equals("South America") || contToAttackEnemy.equals("Europe") || contToAttackEnemy.equals("Asia"))

planeY=300;//\*

firstTimeEnemy=false;

}

g2d = (Graphics2D)g;

old = g2d.getTransform();

if(contToAttackEnemy.equals("North America"))

{

g2d.translate(700,650); //\*

g2d.rotate(Math.toRadians(-160));//\*

}

if(contToAttackEnemy.equals("South America"))

{

g2d.translate(900, 600); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-200));//\*

}

if(contToAttackEnemy.equals("Europe"))

{

g2d.translate(450, 500); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-90));//\*

}

if(contToAttackEnemy.equals("Asia"))

{

g2d.translate(450, 200); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-45));//\*

}

if(contToAttackEnemy.equals("Australia"))

{

g2d.translate(850, 0); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(25));//\*

}

for(int i=0; i<nukesToSendEnemy; i++)//350

{

if(showNukesEnemy[i])g2d.drawImage(aBombPlane, nukeXSpotsEnemy[i], planeY, null);

else

{

nukeSizesEnemy[i]--;

if(nukeSizesEnemy[i]<=0)

{

nukeSizesEnemy[i]=0;

shouldNukeExplode[i] = true;

if(showExplosionEnemy[i] && bombSoundCounterEnemy[i]<5)

{

//System.out.println("i is " + i);

g.drawImage(explosion, nukeXSpotsEnemy[i]-50, planeY, null);

bombSoundCounterEnemy[i]++;

if(bombSoundCounterEnemy[i]==5)showExplosionEnemy[i]=false;

}

//System.out.println("i is " + i +" and "+bombSoundCounter[i]);

//if(dropABombs[i] && bombSoundCounter[i]<2) {g.drawImage(explosion, aBombXSpots[i], planeY, null);bombSoundCounter[i]++;}

}

g2d.drawImage(aBomb, nukeXSpotsEnemy[i], planeY, nukeSizesEnemy[i], nukeSizesEnemy[i], null);//nukeSizesEnemy[i], nukeSizesEnemy[i], null);

}

}

g2d.setTransform(old);

}

if(tempContinent.equals("Australia"))

{

if(firstTimeEnemy)

{

for(int i=0; i<nukeXSpotsEnemy.length; i++)

{

//if(contToAttackEnemy.equals("South America") || contToAttackEnemy.equals("Europe") || contToAttackEnemy.equals("Asia"))

nukeXSpotsEnemy[i]=100;//\*

}

//if(contToAttackEnemy.equals("South America") || contToAttackEnemy.equals("Europe") || contToAttackEnemy.equals("Asia"))

planeY=300;//\*

firstTimeEnemy=false;

}

g2d = (Graphics2D)g;

old = g2d.getTransform();

if(contToAttackEnemy.equals("North America")||contToAttackEnemy.equals("Africa"))

{

g2d.translate(1200,825); //\*

g2d.rotate(Math.toRadians(-160));//\*

}

if(contToAttackEnemy.equals("South America"))

{

g2d.translate(1200, 825); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-170));//\*

}

if(contToAttackEnemy.equals("Europe"))

{

g2d.translate(1100, 825); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-140));//\*

}

if(contToAttackEnemy.equals("Asia"))

{

g2d.translate(900, 700); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-110));//\*

}

for(int i=0; i<nukesToSendEnemy; i++)//350

{

if(showNukesEnemy[i])g2d.drawImage(aBombPlane, nukeXSpotsEnemy[i], planeY, null);

else

{

nukeSizesEnemy[i]--;

if(nukeSizesEnemy[i]<=0)

{

nukeSizesEnemy[i]=0;

shouldNukeExplode[i] = true;

if(showExplosionEnemy[i] && bombSoundCounterEnemy[i]<5)

{

// System.out.println("in enemy nuke place and is is " + i);

g.drawImage(explosion, nukeXSpotsEnemy[i]-50, planeY, null);

bombSoundCounterEnemy[i]++;

if(bombSoundCounterEnemy[i]==5)showExplosionEnemy[i]=false;

}

//System.out.println("i is " + i +" and "+bombSoundCounter[i]);

//if(dropABombs[i] && bombSoundCounter[i]<2) {g.drawImage(explosion, aBombXSpots[i], planeY, null);bombSoundCounter[i]++;}

}

g2d.drawImage(aBomb, nukeXSpotsEnemy[i], planeY, nukeSizesEnemy[i], nukeSizesEnemy[i], null);//nukeSizesEnemy[i], nukeSizesEnemy[i], null);

}

}

g2d.setTransform(old);

}

}

if(sendThePlaneEnemy && sendTheNukeEnemy==false && sendThePlane==false && sendTheABomb==false)

{

timerForPlaneEnemy.start();

if(tempContinent.equals("North America"))

{

if(firstTimeEnemy)

{

for(int i=0; i<planeXSpotsEnemy.length; i++)

{

//if(contToAttackEnemy.equals("South America") || contToAttackEnemy.equals("Europe") || contToAttackEnemy.equals("Asia"))

planeXSpotsEnemy[i]=350;//\*

}

//if(contToAttackEnemy.equals("South America") || contToAttackEnemy.equals("Europe") || contToAttackEnemy.equals("Asia"))

planeY=200;//\*

firstTimeEnemy=false;

}

g2d = (Graphics2D)g;

old = g2d.getTransform();

if(contToAttackEnemy.equals("South America"))

{

g2d.translate(375, -200); //\*

g2d.rotate(Math.toRadians(60));//\*

}

if(contToAttackEnemy.equals("Europe") || contToAttackEnemy.equals("Asia"))

{

g2d.translate(-15, 40); //\*

g2d.rotate(Math.toRadians(-5));//\*

}

if(contToAttackEnemy.equals("Africa"))

{

g2d.translate(45, -45); //\*

g2d.rotate(Math.toRadians(10));//\*

}

if(contToAttackEnemy.equals("Australia"))

{

g2d.translate(80, -80); //\*

g2d.rotate(Math.toRadians(17));//\*

}

for(int i=0; i<planesToSendEnemy; i++)//350

{

if(showPlanesEnemy[i])

{

g2d.drawImage(plane, planeXSpotsEnemy[i], planeY, null);

//g2d.drawImage(aBomb, planeXSpotsEnemy[i], planeY, null);

}

//draw the bombs here

}

g2d.setTransform(old);

}

if(tempContinent.equals("South America"))

{

if(firstTimeEnemy)

{

for(int i=0; i<planeXSpotsEnemy.length; i++)

{

//if(contToAttackEnemy.equals("South America") || contToAttackEnemy.equals("Europe") || contToAttackEnemy.equals("Asia"))

planeXSpotsEnemy[i]=100;//\*

}

//if(contToAttackEnemy.equals("South America") || contToAttackEnemy.equals("Europe") || contToAttackEnemy.equals("Asia"))

planeY=300;//\*

firstTimeEnemy=false;

}

g2d = (Graphics2D)g;

old = g2d.getTransform();

if(contToAttackEnemy.equals("North America"))

{

g2d.translate(200, 650); //\*

g2d.rotate(Math.toRadians(-120));//\*

}

if(contToAttackEnemy.equals("Europe"))

{

g2d.translate(175, 150); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-30));//\*

}

if(contToAttackEnemy.equals("Asia")||contToAttackEnemy.equals("Africa"))

{

g2d.translate(225, 100); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-15));//\*

}

if(contToAttackEnemy.equals("Australia"))

{

g2d.translate(325, 75); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(5));//\*

}

for(int i=0; i<planesToSendEnemy; i++)//350

{

if(showPlanesEnemy[i])g2d.drawImage(plane, planeXSpotsEnemy[i], planeY, null);

}

g2d.setTransform(old);

}

if(tempContinent.equals("Europe"))

{

if(firstTimeEnemy)

{

for(int i=0; i<planeXSpotsEnemy.length; i++)

{

//if(contToAttackEnemy.equals("South America") || contToAttackEnemy.equals("Europe") || contToAttackEnemy.equals("Asia"))

planeXSpotsEnemy[i]=100;//\*

}

//if(contToAttackEnemy.equals("South America") || contToAttackEnemy.equals("Europe") || contToAttackEnemy.equals("Asia"))

planeY=300;//\*

firstTimeEnemy=false;

}

g2d = (Graphics2D)g;

old = g2d.getTransform();

if(contToAttackEnemy.equals("North America"))

{

g2d.translate(900, 500); //\*

g2d.rotate(Math.toRadians(-185));//\*

}

if(contToAttackEnemy.equals("South America"))

{

g2d.translate(1050, 325); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-225));//\*

}

if(contToAttackEnemy.equals("Asia"))

{

g2d.translate(625, -150); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(0));//\*

}

if(contToAttackEnemy.equals("Africa"))

{

g2d.translate(1075, 75); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(90));//\*

}

if(contToAttackEnemy.equals("Australia"))

{

g2d.translate(900, -150); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(40));//\*

}

for(int i=0; i<planesToSendEnemy; i++)//350

{

if(showPlanesEnemy[i])g2d.drawImage(plane, planeXSpotsEnemy[i], planeY, null);

}

g2d.setTransform(old);

}

if(tempContinent.equals("Asia"))

{

if(firstTimeEnemy)

{

for(int i=0; i<planeXSpotsEnemy.length; i++)

{

//if(contToAttackEnemy.equals("South America") || contToAttackEnemy.equals("Europe") || contToAttackEnemy.equals("Asia"))

planeXSpotsEnemy[i]=100;//\*

}

//if(contToAttackEnemy.equals("South America") || contToAttackEnemy.equals("Europe") || contToAttackEnemy.equals("Asia"))

planeY=300;//\*

firstTimeEnemy=false;

}

g2d = (Graphics2D)g;

old = g2d.getTransform();

if(contToAttackEnemy.equals("North America"))

{

g2d.translate(1100, 500); //\*

g2d.rotate(Math.toRadians(-185));//\*

}

if(contToAttackEnemy.equals("South America"))

{

g2d.translate(1200, 450); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-200));//\*

}

if(contToAttackEnemy.equals("Europe"))

{

g2d.translate(1100, 500); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-180));//\*

}

if(contToAttackEnemy.equals("Africa"))

{

g2d.translate(1300, 300); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-225));//\*

}

if(contToAttackEnemy.equals("Australia"))

{

g2d.translate(1350, -25); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(75));//\*

}

for(int i=0; i<planesToSendEnemy; i++)//350

{

if(showPlanesEnemy[i])g2d.drawImage(plane, planeXSpotsEnemy[i], planeY, null);

}

g2d.setTransform(old);

}

if(tempContinent.equals("Africa"))

{

if(firstTimeEnemy)

{

for(int i=0; i<planeXSpotsEnemy.length; i++)

{

//if(contToAttackEnemy.equals("South America") || contToAttackEnemy.equals("Europe") || contToAttackEnemy.equals("Asia"))

planeXSpotsEnemy[i]=100;//\*

}

//if(contToAttackEnemy.equals("South America") || contToAttackEnemy.equals("Europe") || contToAttackEnemy.equals("Asia"))

planeY=300;//\*

firstTimeEnemy=false;

}

g2d = (Graphics2D)g;

old = g2d.getTransform();

if(contToAttackEnemy.equals("North America"))

{

g2d.translate(700,650); //\*

g2d.rotate(Math.toRadians(-160));//\*

}

if(contToAttackEnemy.equals("South America"))

{

g2d.translate(900, 600); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-200));//\*

}

if(contToAttackEnemy.equals("Europe"))

{

g2d.translate(450, 500); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-90));//\*

}

if(contToAttackEnemy.equals("Asia"))

{

g2d.translate(450, 200); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-45));//\*

}

if(contToAttackEnemy.equals("Australia"))

{

g2d.translate(850, 0); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(25));//\*

}

for(int i=0; i<planesToSendEnemy; i++)//350

{

if(showPlanesEnemy[i])g2d.drawImage(plane, planeXSpotsEnemy[i], planeY, null);

}

g2d.setTransform(old);

}

if(tempContinent.equals("Australia"))

{

if(firstTimeEnemy)

{

for(int i=0; i<planeXSpotsEnemy.length; i++)

{

//if(contToAttackEnemy.equals("South America") || contToAttackEnemy.equals("Europe") || contToAttackEnemy.equals("Asia"))

planeXSpotsEnemy[i]=100;//\*

}

//if(contToAttackEnemy.equals("South America") || contToAttackEnemy.equals("Europe") || contToAttackEnemy.equals("Asia"))

planeY=300;//\*

firstTimeEnemy=false;

}

g2d = (Graphics2D)g;

old = g2d.getTransform();

if(contToAttackEnemy.equals("North America")||contToAttackEnemy.equals("Africa"))

{

g2d.translate(1200,825); //\*

g2d.rotate(Math.toRadians(-160));//\*

}

if(contToAttackEnemy.equals("South America"))

{

g2d.translate(1200, 825); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-170));//\*

}

if(contToAttackEnemy.equals("Europe"))

{

g2d.translate(1100, 825); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-140));//\*

}

if(contToAttackEnemy.equals("Asia"))

{

g2d.translate(900, 700); //\*//\*175,150 and -30

g2d.rotate(Math.toRadians(-110));//\*

}

for(int i=0; i<planesToSendEnemy; i++)//350

{

if(showPlanesEnemy[i])g2d.drawImage(plane, planeXSpotsEnemy[i], planeY, null);

}

g2d.setTransform(old);

}

}

}

}

class PlaneMover implements ActionListener

{

public void actionPerformed(ActionEvent e)

{

//System.out.println("plane moving");

if(continent.equals("North America"))

{

planeXSpots[0]++;

for(int i=1; i<planeXSpots.length; i++)

{

if(planeXSpots[i-1]-planeXSpots[i]>=50) planeXSpots[i]++;

}

for(int i=0; i<planeXSpots.length; i++)

{

if(contToAttack.equals("South America")) if(planeXSpots[i]>=550) showPlanes[i] = false;//\*

if(contToAttack.equals("Europe")) if(planeXSpots[i]>=750) showPlanes[i] = false;//\*

if(contToAttack.equals("Asia")) if(planeXSpots[i]>=1000) showPlanes[i] = false;//\*

if(contToAttack.equals("Africa")) if(planeXSpots[i]>=700) showPlanes[i] = false;//\*

if(contToAttack.equals("Australia")) if(planeXSpots[i]>=1200) showPlanes[i] = false;//\*

}

boolean checkAll = false;

for(int i=0; i<showPlanes.length; i++)

{

if(showPlanes[i]==true) checkAll = true;

}

if(checkAll==false)

{

timerForPlane.stop();

sendThePlane=false;

firstTime = true;

int indexOfContAttack = 0;

double successChance = Math.random();

for(int i=0; i<continents.length; i++)

{

if(continents[i].equals(contToAttack)) indexOfContAttack=i;

}

if(successChance<0.9-defenseCount[indexOfContAttack]) //\*

{

if(planesToSend>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttack+" were successful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttack+" were successful in their mission.");//\*

healthReduce[indexOfContAttack] -= planesToSend+(jetLevel\*2);

int moneyCost = (int)(Math.random()\*100000+10000);

int peopleDamage = ((int)(Math.random()\*1000+100));

startingMoney[indexOfContAttack] -= moneyCost;

populations[indexOfContAttack] -= peopleDamage;

if(healthReduce[indexOfContAttack]<=-97)

{

canUpdateInfo[indexOfContAttack]=true;

makeBlue[indexOfContAttack]=true;

if(startingMoney[indexOfContAttack]>=0)startingMoney[whichCont] += startingMoney[indexOfContAttack]/2;

if(startingTroops[indexOfContAttack]>=0)startingTroops[whichCont] += startingTroops[indexOfContAttack]/2;

startingTroops[indexOfContAttack]=0;

startingMoney[indexOfContAttack] = 0;

for(int i=0; i<6; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

troopPopulations[i].setText(" "+continents[i]+": "+startingTroops[i]);

}

else {jm3.setText(""+startingMoney[i]);jm5.setText(""+startingTroops[i]);}

}

}

}

else

{

if(planesToSend>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttack+" were unsuccessful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttack+" were unsuccessful in their mission.");//\*

}

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

}

else

{

jm3.setText(""+startingMoney[i]);

jm4.setText(""+populations[i]);

}

}

removeAll();

setLayout(new GridLayout(3,3));

timer.stop();

for(int i=0; i<9; i++)

{

if(i==4) add(panelWithEvent);

else add(new JLabel(""));

}

revalidate();

repaint();

}

}

if(continent.equals("South America"))

{

planeXSpots[0]++;

for(int i=1; i<planeXSpots.length; i++)

{

if(planeXSpots[i-1]-planeXSpots[i]>=50) planeXSpots[i]++;

}

for(int i=0; i<planeXSpots.length; i++)

{

if(contToAttack.equals("North America")) if(planeXSpots[i]>=300) showPlanes[i] = false;//\*

if(contToAttack.equals("Europe")) if(planeXSpots[i]>=450) showPlanes[i] = false;//\*

if(contToAttack.equals("Asia")) if(planeXSpots[i]>=700) showPlanes[i] = false;//\*

if(contToAttack.equals("Africa")) if(planeXSpots[i]>=425) showPlanes[i] = false;//\*

if(contToAttack.equals("Australia")) if(planeXSpots[i]>=850) showPlanes[i] = false;//\*

}

boolean checkAll = false;

for(int i=0; i<showPlanes.length; i++)

{

if(showPlanes[i]==true) checkAll = true;

}

if(checkAll==false)

{

timerForPlane.stop();

sendThePlane=false;

firstTime = true;

int indexOfContAttack = 0;

double successChance = Math.random();

for(int i=0; i<continents.length; i++)

{

if(continents[i].equals(contToAttack)) indexOfContAttack=i;

}

if(successChance<0.9-defenseCount[indexOfContAttack]) //\*

{

if(planesToSend>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttack+" were successful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttack+" were successful in their mission.");//\*

healthReduce[indexOfContAttack] -= planesToSend+(jetLevel\*2);

int moneyCost = (int)(Math.random()\*100000+10000);

int peopleDamage = ((int)(Math.random()\*1000+100));

startingMoney[indexOfContAttack] -= moneyCost;

populations[indexOfContAttack] -= peopleDamage;

if(healthReduce[indexOfContAttack]<=-97)

{

canUpdateInfo[indexOfContAttack]=true;

makeBlue[indexOfContAttack]=true;

if(startingMoney[indexOfContAttack]>=0)startingMoney[whichCont] += startingMoney[indexOfContAttack]/2;

if(startingTroops[indexOfContAttack]>=0)startingTroops[whichCont] += startingTroops[indexOfContAttack]/2;

startingTroops[indexOfContAttack]=0;

startingMoney[indexOfContAttack] = 0;

for(int i=0; i<6; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

troopPopulations[i].setText(" "+continents[i]+": "+startingMoney[i]);

}

else {jm3.setText(""+startingMoney[i]);jm5.setText(""+startingTroops[i]);}

}

}

}

else

{

if(planesToSend>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttack+" were unsuccessful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttack+" were unsuccessful in their mission.");//\*

}

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

}

else

{

jm3.setText(""+startingMoney[i]);

jm4.setText(""+populations[i]);

}

}

removeAll();

setLayout(new GridLayout(3,3));

timer.stop();

for(int i=0; i<9; i++)

{

if(i==4) add(panelWithEvent);

else add(new JLabel(""));

}

revalidate();

repaint();

}

}

if(continent.equals("Europe"))

{

planeXSpots[0]++;

for(int i=1; i<planeXSpots.length; i++)

{

if(planeXSpots[i-1]-planeXSpots[i]>=50) planeXSpots[i]++;

}

for(int i=0; i<planeXSpots.length; i++)

{

if(contToAttack.equals("North America")) if(planeXSpots[i]>=575) showPlanes[i] = false;//\*

if(contToAttack.equals("South America")) if(planeXSpots[i]>=500) showPlanes[i] = false;//\*

if(contToAttack.equals("Asia")) if(planeXSpots[i]>=350) showPlanes[i] = false;//\*

if(contToAttack.equals("Africa")) if(planeXSpots[i]>=250) showPlanes[i] = false;//\*

if(contToAttack.equals("Australia")) if(planeXSpots[i]>=600) showPlanes[i] = false;//\*

}

boolean checkAll = false;

for(int i=0; i<showPlanes.length; i++)

{

if(showPlanes[i]==true) checkAll = true;

}

if(checkAll==false)

{

timerForPlane.stop();

sendThePlane=false;

firstTime = true;

int indexOfContAttack = 0;

double successChance = Math.random();

for(int i=0; i<continents.length; i++)

{

if(continents[i].equals(contToAttack)) indexOfContAttack=i;

}

if(successChance<0.9-defenseCount[indexOfContAttack]) //\*

{

if(planesToSend>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttack+" were successful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttack+" were successful in their mission.");//\*

healthReduce[indexOfContAttack] -= planesToSend+(jetLevel\*2);

int moneyCost = (int)(Math.random()\*100000+10000);

int peopleDamage = ((int)(Math.random()\*1000+100));

startingMoney[indexOfContAttack] -= moneyCost;

populations[indexOfContAttack] -= peopleDamage;

if(healthReduce[indexOfContAttack]<=-97)

{

canUpdateInfo[indexOfContAttack]=true;

makeBlue[indexOfContAttack]=true;

if(startingMoney[indexOfContAttack]>=0)startingMoney[whichCont] += startingMoney[indexOfContAttack]/2;

if(startingTroops[indexOfContAttack]>=0)startingTroops[whichCont] += startingTroops[indexOfContAttack]/2;

startingTroops[indexOfContAttack]=0;

startingMoney[indexOfContAttack] = 0;

for(int i=0; i<6; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

troopPopulations[i].setText(" "+continents[i]+": "+startingMoney[i]);

}

else {jm3.setText(""+startingMoney[i]);jm5.setText(""+startingTroops[i]);}

}

}

}

else

{

if(planesToSend>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttack+" were unsuccessful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttack+" were unsuccessful in their mission.");//\*

}

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

}

else

{

jm3.setText(""+startingMoney[i]);

jm4.setText(""+populations[i]);

}

}

removeAll();

setLayout(new GridLayout(3,3));

timer.stop();

for(int i=0; i<9; i++)

{

if(i==4) add(panelWithEvent);

else add(new JLabel(""));

}

revalidate();

repaint();

}

}

if(continent.equals("Asia"))

{

planeXSpots[0]++;

for(int i=1; i<planeXSpots.length; i++)

{

if(planeXSpots[i-1]-planeXSpots[i]>=50) planeXSpots[i]++;

}

for(int i=0; i<planeXSpots.length; i++)

{

if(contToAttack.equals("North America")) if(planeXSpots[i]>=700) showPlanes[i] = false;//\*

if(contToAttack.equals("South America")) if(planeXSpots[i]>=700) showPlanes[i] = false;//\*

if(contToAttack.equals("Europe")) if(planeXSpots[i]>=350) showPlanes[i] = false;//\*

if(contToAttack.equals("Africa")) if(planeXSpots[i]>=400) showPlanes[i] = false;//\*

if(contToAttack.equals("Australia")) if(planeXSpots[i]>=400) showPlanes[i] = false;//\*

}

boolean checkAll = false;

for(int i=0; i<showPlanes.length; i++)

{

if(showPlanes[i]==true) checkAll = true;

}

if(checkAll==false)

{

timerForPlane.stop();

sendThePlane=false;

firstTime = true;

int indexOfContAttack = 0;

double successChance = Math.random();

for(int i=0; i<continents.length; i++)

{

if(continents[i].equals(contToAttack)) indexOfContAttack=i;

}

if(successChance<0.9-defenseCount[indexOfContAttack]) //\*

{

if(planesToSend>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttack+" were successful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttack+" were successful in their mission.");//\*

healthReduce[indexOfContAttack] -= planesToSend+(jetLevel\*2);

int moneyCost = (int)(Math.random()\*100000+10000);

int peopleDamage = ((int)(Math.random()\*1000+100));

startingMoney[indexOfContAttack] -= moneyCost;

populations[indexOfContAttack] -= peopleDamage;

if(healthReduce[indexOfContAttack]<=-97)

{

canUpdateInfo[indexOfContAttack]=true;

makeBlue[indexOfContAttack]=true;

if(startingMoney[indexOfContAttack]>=0)startingMoney[whichCont] += startingMoney[indexOfContAttack]/2;

if(startingTroops[indexOfContAttack]>=0)startingTroops[whichCont] += startingTroops[indexOfContAttack]/2;

startingTroops[indexOfContAttack]=0;

startingMoney[indexOfContAttack] = 0;

for(int i=0; i<6; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

troopPopulations[i].setText(" "+continents[i]+": "+startingMoney[i]);

}

else {jm3.setText(""+startingMoney[i]);jm5.setText(""+startingTroops[i]);}

}

}

}

else

{

if(planesToSend>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttack+" were unsuccessful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttack+" were unsuccessful in their mission.");//\*

}

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

}

else

{

jm3.setText(""+startingMoney[i]);

jm4.setText(""+populations[i]);

}

}

removeAll();

setLayout(new GridLayout(3,3));

timer.stop();

for(int i=0; i<9; i++)

{

if(i==4) add(panelWithEvent);

else add(new JLabel(""));

}

revalidate();

repaint();

}

}

if(continent.equals("Africa"))

{

planeXSpots[0]++;

for(int i=1; i<planeXSpots.length; i++)

{

if(planeXSpots[i-1]-planeXSpots[i]>=50) planeXSpots[i]++;

}

for(int i=0; i<planeXSpots.length; i++)

{

if(contToAttack.equals("North America")) if(planeXSpots[i]>=500) showPlanes[i] = false;//\*

if(contToAttack.equals("South America")) if(planeXSpots[i]>=300) showPlanes[i] = false;//\*

if(contToAttack.equals("Europe")) if(planeXSpots[i]>=325) showPlanes[i] = false;//\*

if(contToAttack.equals("Asia")) if(planeXSpots[i]>=400) showPlanes[i] = false;//\*

if(contToAttack.equals("Australia")) if(planeXSpots[i]>=450) showPlanes[i] = false;//\*

}

boolean checkAll = false;

for(int i=0; i<showPlanes.length; i++)

{

if(showPlanes[i]==true) checkAll = true;

}

if(checkAll==false)

{

timerForPlane.stop();

sendThePlane=false;

firstTime = true;

int indexOfContAttack = 0;

double successChance = Math.random();

for(int i=0; i<continents.length; i++)

{

if(continents[i].equals(contToAttack)) indexOfContAttack=i;

}

if(successChance<0.9-defenseCount[indexOfContAttack]) //\*

{

if(planesToSend>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttack+" were successful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttack+" were successful in their mission.");//\*

healthReduce[indexOfContAttack] -= planesToSend+(jetLevel\*2);

int moneyCost = (int)(Math.random()\*100000+10000);

int peopleDamage = ((int)(Math.random()\*1000+100));

startingMoney[indexOfContAttack] -= moneyCost;

populations[indexOfContAttack] -= peopleDamage;

if(healthReduce[indexOfContAttack]<=-97)

{

canUpdateInfo[indexOfContAttack]=true;

makeBlue[indexOfContAttack]=true;

if(startingMoney[indexOfContAttack]>=0)startingMoney[whichCont] += startingMoney[indexOfContAttack]/2;

if(startingTroops[indexOfContAttack]>=0)startingTroops[whichCont] += startingTroops[indexOfContAttack]/2;

startingTroops[indexOfContAttack]=0;

startingMoney[indexOfContAttack] = 0;

for(int i=0; i<6; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

troopPopulations[i].setText(" "+continents[i]+": "+startingMoney[i]);

}

else {jm3.setText(""+startingMoney[i]);jm5.setText(""+startingTroops[i]);}

}

}

}

else

{

if(planesToSend>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttack+" were unsuccessful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttack+" were unsuccessful in their mission.");//\*

}

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

}

else

{

jm3.setText(""+startingMoney[i]);

jm4.setText(""+populations[i]);

}

}

removeAll();

setLayout(new GridLayout(3,3));

timer.stop();

for(int i=0; i<9; i++)

{

if(i==4) add(panelWithEvent);

else add(new JLabel(""));

}

revalidate();

repaint();

}

}

if(continent.equals("Australia"))

{

planeXSpots[0]++;

for(int i=1; i<planeXSpots.length; i++)

{

if(planeXSpots[i-1]-planeXSpots[i]>=50) planeXSpots[i]++;

}

for(int i=0; i<planeXSpots.length; i++)

{

if(contToAttack.equals("North America")) if(planeXSpots[i]>=1000) showPlanes[i] = false;//\*

if(contToAttack.equals("South America")) if(planeXSpots[i]>=800) showPlanes[i] = false;//\*

if(contToAttack.equals("Europe")) if(planeXSpots[i]>=650) showPlanes[i] = false;//\*

if(contToAttack.equals("Asia")) if(planeXSpots[i]>=400) showPlanes[i] = false;//\*

if(contToAttack.equals("Africa")) if(planeXSpots[i]>=600) showPlanes[i] = false;//\*

}

boolean checkAll = false;

for(int i=0; i<showPlanes.length; i++)

{

if(showPlanes[i]==true) checkAll = true;

}

if(checkAll==false)

{

timerForPlane.stop();

sendThePlane=false;

firstTime = true;

int indexOfContAttack = 0;

double successChance = Math.random();

for(int i=0; i<continents.length; i++)

{

if(continents[i].equals(contToAttack)) indexOfContAttack=i;

}

if(successChance<0.9-defenseCount[indexOfContAttack]) //\*

{

if(planesToSend>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttack+" were successful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttack+" were successful in their mission.");//\*

healthReduce[indexOfContAttack] -= planesToSend+(jetLevel\*2);

int moneyCost = (int)(Math.random()\*100000+10000);

int peopleDamage = ((int)(Math.random()\*1000+100));

startingMoney[indexOfContAttack] -= moneyCost;

populations[indexOfContAttack] -= peopleDamage;

if(healthReduce[indexOfContAttack]<=-97)

{

canUpdateInfo[indexOfContAttack]=true;

makeBlue[indexOfContAttack]=true;

if(startingMoney[indexOfContAttack]>=0)startingMoney[whichCont] += startingMoney[indexOfContAttack]/2;

if(startingTroops[indexOfContAttack]>=0)startingTroops[whichCont] += startingTroops[indexOfContAttack]/2;

startingTroops[indexOfContAttack]=0;

startingMoney[indexOfContAttack] = 0;

for(int i=0; i<6; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

troopPopulations[i].setText(" "+continents[i]+": "+startingMoney[i]);

}

else {jm3.setText(""+startingMoney[i]);jm5.setText(""+startingTroops[i]);}

}

}

}

else

{

if(planesToSend>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttack+" were unsuccessful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttack+" were unsuccessful in their mission.");//\*

}

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

}

else

{

jm3.setText(""+startingMoney[i]);

jm4.setText(""+populations[i]);

}

}

removeAll();

setLayout(new GridLayout(3,3));

timer.stop();

for(int i=0; i<9; i++)

{

if(i==4) add(panelWithEvent);

else add(new JLabel(""));

}

revalidate();

repaint();

}

}

repaint();

if(healthReduce[whichCont]<=-97) {endGame=true;gameOverCount++;healthGO=true;}

if(startingMoney[whichCont]<=0) {endGame=true;moneyGO=true;gameOverCount++;}

if(populations[whichCont]<=0) {endGame=true;gameOverCount++;peopleGO=true;}

if(startingTroops[whichCont]<=0) {endGame=true;gameOverCount++;troopGO=true;}

boolean testWin = true;

for(int i=0; i<makeBlue.length; i++)

{

if(i!=whichCont)

{

if(makeBlue[i]==false) testWin=false;

}

}

if(testWin) won=true;

}

}

class ABombMover implements ActionListener

{

public void actionPerformed(ActionEvent e)

{

//System.out.println("bomb moving");

if(continent.equals("North America"))

{

aBombXSpots[0]++;

for(int i=1; i<aBombXSpots.length; i++)

{

if(aBombXSpots[i-1]-aBombXSpots[i]>=50) aBombXSpots[i]++;

}

for(int i=0; i<aBombXSpots.length; i++)

{

if(contToAttack.equals("South America")) if(aBombXSpots[i]>=550){ showABombs[i] = false; }

if(contToAttack.equals("Europe")) if(aBombXSpots[i]>=750) { showABombs[i] = false; }

if(contToAttack.equals("Asia")) if(aBombXSpots[i]>=1000) { showABombs[i] = false; }

if(contToAttack.equals("Africa")) if(aBombXSpots[i]>=700) { showABombs[i] = false;}

if(contToAttack.equals("Australia")) if(aBombXSpots[i]>=1200) { showABombs[i] = false; }

}

for(int i=0; i<aBombXSpots.length; i++)

{//+50 || aBombXSpots[aBombXSpots.length-1]==550

//use drop a bombs or another array to play sounds and then make checkAll check that array to do this also in enemy nukes

if(contToAttack.equals("South America")) if(aBombXSpots[i]==550+50){dropABombs[i] = true;showExplosion[i]=true;}//or the last index is equal to the x coordinate

if(contToAttack.equals("Europe")) if(aBombXSpots[i]==750+50) {dropABombs[i] = true;showExplosion[i]=true;}

if(contToAttack.equals("Asia")) if(aBombXSpots[i]==1000+50) {dropABombs[i] = true;showExplosion[i]=true;}

if(contToAttack.equals("Africa")) if(aBombXSpots[i]==700+50){dropABombs[i] = true;showExplosion[i]=true;}

if(contToAttack.equals("Australia")) if(aBombXSpots[i]==1200+50) {dropABombs[i] = true;showExplosion[i]=true;}

}

for(int i=0; i<dropABombs.length; i++)

{

if(dropABombs[i] && bombSoundCounter[i]==0 && keepSFX) {explosionSound.play();bombSoundCounter[i]++;}

}

boolean checkAll = false;

for(int i=0; i<showABombs.length; i++)

{

if(showABombs[i]==true || aBombSizes[i]>0 || dropABombs[i]==false || showExplosion[i]) checkAll = true;

}

if(checkAll==false)

{

//for(int i=0; i<dropABombs.length; i++) dropABombs[i]=true;

timerForABombs.stop();

sendTheABomb=false;

firstTime = true;

int indexOfContAttack = 0;

double successChance = Math.random();

for(int i=0; i<continents.length; i++)

{

if(continents[i].equals(contToAttack)) indexOfContAttack=i;

}

if(true) //\*

{

if(aBombsToSend>1)eventPanel.setText("The nukes dropped in "+contToAttack+" were highly destructive.");//\*

else eventPanel.setText("The nuke dropped in "+contToAttack+" was highly destructive.");//\*

healthReduce[indexOfContAttack] -= aBombsToSend\*10+(bombLevel\*2);

//System.out.println(healthReduce[indexOfContAttack]);

if(healthReduce[indexOfContAttack]<=-97)

{

//System.out.println("dude");

canUpdateInfo[indexOfContAttack]=true;

makeBlue[indexOfContAttack]=true;

if(startingMoney[indexOfContAttack]>=0)startingMoney[whichCont] += startingMoney[indexOfContAttack]/2;

if(startingTroops[indexOfContAttack]>=0)startingTroops[whichCont] += startingTroops[indexOfContAttack]/2;

startingTroops[indexOfContAttack]=0;

startingMoney[indexOfContAttack] = 0;

// System.out.println(startingMoney[indexOfContAttack]);//???

for(int i=0; i<6; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

troopPopulations[i].setText(" "+continents[i]+": "+startingTroops[i]);

}

else {jm3.setText(""+startingMoney[i]);jm5.setText(""+startingTroops[i]);}

}

//revalidate();

repaint();

}

}

else

{

if(aBombsToSend>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttack+" were unsuccessful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttack+" were unsuccessful in their mission.");//\*

}

int moneyCost = (int)(Math.random()\*10000000+1000000);

int peopleDamage = ((int)(Math.random()\*1000000+500))\*10;

if(startingMoney[indexOfContAttack]>0)startingMoney[indexOfContAttack] -= moneyCost;

populations[indexOfContAttack] -= peopleDamage;

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

}

}

removeAll();

setLayout(new GridLayout(3,3));

timer.stop();

for(int i=0; i<9; i++)

{

if(i==4) add(panelWithEvent);

else add(new JLabel(""));

}

revalidate();

repaint();

}

}

if(continent.equals("South America"))

{

aBombXSpots[0]++;

for(int i=1; i<aBombXSpots.length; i++)

{

if(aBombXSpots[i-1]-aBombXSpots[i]>=50) aBombXSpots[i]++;

}

for(int i=0; i<aBombXSpots.length; i++)

{

if(contToAttack.equals("North America")) if(aBombXSpots[i]>=300) { showABombs[i] = false; }

if(contToAttack.equals("Europe")) if(aBombXSpots[i]>=450) { showABombs[i] = false; }

if(contToAttack.equals("Asia")) if(aBombXSpots[i]>=700) { showABombs[i] = false; }

if(contToAttack.equals("Africa")) if(aBombXSpots[i]>=425) { showABombs[i] = false; }

if(contToAttack.equals("Australia")) if(aBombXSpots[i]>=850) { showABombs[i] = false; }//showExplosion[i]=true;

}

for(int i=0; i<aBombXSpots.length; i++)

{

if(contToAttack.equals("North America")) if(aBombXSpots[i]==300+50) {dropABombs[i] = true;showExplosion[i]=true;}//{dropABombs[i] = true;} for all

if(contToAttack.equals("Europe")) if(aBombXSpots[i]==450+50) {dropABombs[i] = true;showExplosion[i]=true;}

if(contToAttack.equals("Asia")) if(aBombXSpots[i]==700+50) {dropABombs[i] = true;showExplosion[i]=true;}

if(contToAttack.equals("Africa")) if(aBombXSpots[i]==425+50) {dropABombs[i] = true;showExplosion[i]=true;}

if(contToAttack.equals("Australia")) if(aBombXSpots[i]==850+50) {dropABombs[i] = true;showExplosion[i]=true;}

}

for(int i=0; i<dropABombs.length; i++)

{

if(dropABombs[i] && bombSoundCounter[i]==0 && keepSFX) {explosionSound.play();bombSoundCounter[i]++;}

}

boolean checkAll = false;

for(int i=0; i<showABombs.length; i++)

{

if(showABombs[i]==true || aBombSizes[i]>0 || dropABombs[i]==false || showExplosion[i]) checkAll = true;

}

if(checkAll==false)

{

timerForABombs.stop();

sendTheABomb=false;

firstTime = true;

int indexOfContAttack = 0;

double successChance = Math.random();

for(int i=0; i<continents.length; i++)

{

if(continents[i].equals(contToAttack)) indexOfContAttack=i;

}

if(true) //\*

{

if(aBombsToSend>1)eventPanel.setText("The nukes dropped in "+contToAttack+" were highly destructive.");//\*

else eventPanel.setText("The nuke dropped in "+contToAttack+" was highly destructive.");//\*

healthReduce[indexOfContAttack] -= aBombsToSend\*10+(bombLevel\*2);

if(healthReduce[indexOfContAttack]<=-97)

{

canUpdateInfo[indexOfContAttack]=true;

makeBlue[indexOfContAttack]=true;

if(startingMoney[indexOfContAttack]>=0)startingMoney[whichCont] += startingMoney[indexOfContAttack]/2;

if(startingTroops[indexOfContAttack]>=0)startingTroops[whichCont] += startingTroops[indexOfContAttack]/2;

startingTroops[indexOfContAttack]=0;

startingMoney[indexOfContAttack] = 0;

for(int i=0; i<6; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

troopPopulations[i].setText(" "+continents[i]+": "+startingTroops[i]);

}

else {jm3.setText(""+startingMoney[i]);jm5.setText(""+startingTroops[i]);}

}

}

}

else

{

if(aBombsToSend>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttack+" were unsuccessful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttack+" were unsuccessful in their mission.");//\*

}

int moneyCost = (int)(Math.random()\*10000000+1000000);

int peopleDamage = ((int)(Math.random()\*1000000+500))\*10;

if(startingMoney[indexOfContAttack]>0)startingMoney[indexOfContAttack] -= moneyCost;

populations[indexOfContAttack] -= peopleDamage;

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

}

}

removeAll();

setLayout(new GridLayout(3,3));

timer.stop();

for(int i=0; i<9; i++)

{

if(i==4) add(panelWithEvent);

else add(new JLabel(""));

}

revalidate();

repaint();

}

}

if(continent.equals("Europe")) //xStopSpot=600; for all things

{

aBombXSpots[0]++;

for(int i=1; i<aBombXSpots.length; i++)

{

if(aBombXSpots[i-1]-aBombXSpots[i]>=50) aBombXSpots[i]++;

}

for(int i=0; i<aBombXSpots.length; i++)

{

if(contToAttack.equals("North America")) if(aBombXSpots[i]>=575) { showABombs[i] = false; }

if(contToAttack.equals("South America")) if(aBombXSpots[i]>=500) { showABombs[i] = false; }

if(contToAttack.equals("Asia")) if(aBombXSpots[i]>=350) { showABombs[i] = false; }

if(contToAttack.equals("Africa")) if(aBombXSpots[i]>=250) { showABombs[i] = false; }

if(contToAttack.equals("Australia")) if(aBombXSpots[i]>=600) { showABombs[i] = false;}

}

for(int i=0; i<aBombXSpots.length; i++)

{

if(contToAttack.equals("North America")) if(aBombXSpots[i]==575+50) {dropABombs[i] = true;showExplosion[i]=true;}

if(contToAttack.equals("South America")) if(aBombXSpots[i]==500+50) {dropABombs[i] = true;showExplosion[i]=true;}

if(contToAttack.equals("Asia")) if(aBombXSpots[i]==350+50) {dropABombs[i] = true;showExplosion[i]=true;}

if(contToAttack.equals("Africa")) if(aBombXSpots[i]==250+50) {dropABombs[i] = true;showExplosion[i]=true;}

if(contToAttack.equals("Australia")) if(aBombXSpots[i]==600+50) {dropABombs[i] = true;showExplosion[i]=true;}

}

for(int i=0; i<dropABombs.length; i++)

{

if(dropABombs[i] && bombSoundCounter[i]==0 && keepSFX) {explosionSound.play();bombSoundCounter[i]++;}

}

boolean checkAll = false;

for(int i=0; i<showABombs.length; i++)

{

if(showABombs[i]==true || aBombSizes[i]>0 || dropABombs[i]==false || showExplosion[i]) checkAll = true;

}

if(checkAll==false)

{

timerForABombs.stop();

sendTheABomb=false;

firstTime = true;

int indexOfContAttack = 0;

double successChance = Math.random();

for(int i=0; i<continents.length; i++)

{

if(continents[i].equals(contToAttack)) indexOfContAttack=i;

}

if(true) //\*

{

if(aBombsToSend>1)eventPanel.setText("The nukes dropped in "+contToAttack+" were highly destructive.");//\*

else eventPanel.setText("The nuke dropped in "+contToAttack+" was highly destructive.");//\*

healthReduce[indexOfContAttack] -= aBombsToSend\*10+(bombLevel\*2);

if(healthReduce[indexOfContAttack]<=-97)

{

canUpdateInfo[indexOfContAttack]=true;

makeBlue[indexOfContAttack]=true;

if(startingMoney[indexOfContAttack]>=0)startingMoney[whichCont] += startingMoney[indexOfContAttack]/2;

if(startingTroops[indexOfContAttack]>=0)startingTroops[whichCont] += startingTroops[indexOfContAttack]/2;

startingTroops[indexOfContAttack]=0;

startingMoney[indexOfContAttack] = 0;

for(int i=0; i<6; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

troopPopulations[i].setText(" "+continents[i]+": "+startingTroops[i]);

}

else {jm3.setText(""+startingMoney[i]);jm5.setText(""+startingTroops[i]);}

}

}

}

else

{

if(aBombsToSend>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttack+" were unsuccessful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttack+" were unsuccessful in their mission.");//\*

}

int moneyCost = (int)(Math.random()\*10000000+1000000);

int peopleDamage = ((int)(Math.random()\*1000000+500))\*10;

if(startingMoney[indexOfContAttack]>0)startingMoney[indexOfContAttack] -= moneyCost;

populations[indexOfContAttack] -= peopleDamage;

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

}

}

removeAll();

setLayout(new GridLayout(3,3));

timer.stop();

for(int i=0; i<9; i++)

{

if(i==4) add(panelWithEvent);

else add(new JLabel(""));

}

revalidate();

repaint();

}

}

if(continent.equals("Asia"))

{

aBombXSpots[0]++;

for(int i=1; i<aBombXSpots.length; i++)

{

if(aBombXSpots[i-1]-aBombXSpots[i]>=50) aBombXSpots[i]++;

}

for(int i=0; i<aBombXSpots.length; i++)

{

if(contToAttack.equals("North America")) if(aBombXSpots[i]>=700) { showABombs[i] = false; }

if(contToAttack.equals("South America")) if(aBombXSpots[i]>=700) { showABombs[i] = false; }

if(contToAttack.equals("Europe")) if(aBombXSpots[i]>=350) { showABombs[i] = false; }

if(contToAttack.equals("Africa")) if(aBombXSpots[i]>=400) { showABombs[i] = false; }

if(contToAttack.equals("Australia")) if(aBombXSpots[i]>=400) { showABombs[i] = false; }

}

for(int i=0; i<aBombXSpots.length; i++)

{

if(contToAttack.equals("North America")) if(aBombXSpots[i]==700+50) {dropABombs[i] = true;showExplosion[i]=true;}

if(contToAttack.equals("South America")) if(aBombXSpots[i]==700+50) {dropABombs[i] = true;showExplosion[i]=true;}

if(contToAttack.equals("Europe")) if(aBombXSpots[i]==350+50){dropABombs[i] = true;showExplosion[i]=true;}

if(contToAttack.equals("Africa")) if(aBombXSpots[i]==400+50) {dropABombs[i] = true;showExplosion[i]=true;}

if(contToAttack.equals("Australia")) if(aBombXSpots[i]==400+50){ dropABombs[i] = true;showExplosion[i]=true;}

}

for(int i=0; i<dropABombs.length; i++)

{

if(dropABombs[i] && bombSoundCounter[i]==0 && keepSFX) {explosionSound.play();bombSoundCounter[i]++;}

}

boolean checkAll = false;

for(int i=0; i<showABombs.length; i++)

{

if(showABombs[i]==true || aBombSizes[i]>0 || dropABombs[i]==false || showExplosion[i]) checkAll = true;

}

if(checkAll==false)

{

timerForABombs.stop();

sendTheABomb=false;

firstTime = true;

int indexOfContAttack = 0;

double successChance = Math.random();

for(int i=0; i<continents.length; i++)

{

if(continents[i].equals(contToAttack)) indexOfContAttack=i;

}

if(true) //\*

{

if(aBombsToSend>1)eventPanel.setText("The nukes dropped in "+contToAttack+" were highly destructive.");//\*

else eventPanel.setText("The nuke dropped in "+contToAttack+" was highly destructive.");//\*

healthReduce[indexOfContAttack] -= aBombsToSend\*10+(bombLevel\*2);

if(healthReduce[indexOfContAttack]<=-97)

{

canUpdateInfo[indexOfContAttack]=true;

makeBlue[indexOfContAttack]=true;

if(startingMoney[indexOfContAttack]>=0)startingMoney[whichCont] += startingMoney[indexOfContAttack]/2;

if(startingTroops[indexOfContAttack]>=0)startingTroops[whichCont] += startingTroops[indexOfContAttack]/2;

startingTroops[indexOfContAttack]=0;

startingMoney[indexOfContAttack] = 0;

for(int i=0; i<6; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

troopPopulations[i].setText(" "+continents[i]+": "+startingTroops[i]);

}

else {jm3.setText(""+startingMoney[i]);jm5.setText(""+startingTroops[i]);}

}

}

}

else

{

if(aBombsToSend>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttack+" were unsuccessful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttack+" were unsuccessful in their mission.");//\*

}

int moneyCost = (int)(Math.random()\*10000000+1000000);

int peopleDamage = ((int)(Math.random()\*1000000+500))\*10;

if(startingMoney[indexOfContAttack]>0)startingMoney[indexOfContAttack] -= moneyCost;

populations[indexOfContAttack] -= peopleDamage;

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

}

}

removeAll();

setLayout(new GridLayout(3,3));

timer.stop();

for(int i=0; i<9; i++)

{

if(i==4) add(panelWithEvent);

else add(new JLabel(""));

}

revalidate();

repaint();

}

}

if(continent.equals("Africa"))

{

aBombXSpots[0]++;

for(int i=1; i<aBombXSpots.length; i++)

{

if(aBombXSpots[i-1]-aBombXSpots[i]>=50) aBombXSpots[i]++;

}

for(int i=0; i<aBombXSpots.length; i++)

{

if(contToAttack.equals("North America")) if(aBombXSpots[i]>=500) { showABombs[i] = false;}

if(contToAttack.equals("South America")) if(aBombXSpots[i]>=300) { showABombs[i] = false; ;}

if(contToAttack.equals("Europe")) if(aBombXSpots[i]>=325) { showABombs[i] = false; }

if(contToAttack.equals("Asia")) if(aBombXSpots[i]>=400) { showABombs[i] = false; }

if(contToAttack.equals("Australia")) if(aBombXSpots[i]>=450) { showABombs[i] = false; }

}

for(int i=0; i<aBombXSpots.length; i++)

{

if(contToAttack.equals("North America")) if(aBombXSpots[i]==500+50) {dropABombs[i] = true;showExplosion[i]=true;}

if(contToAttack.equals("South America")) if(aBombXSpots[i]==300+50) {dropABombs[i] = true;showExplosion[i]=true;}

if(contToAttack.equals("Europe")) if(aBombXSpots[i]==325+50) {dropABombs[i] = true;showExplosion[i]=true;}

if(contToAttack.equals("Asia")) if(aBombXSpots[i]==400+50) {dropABombs[i] = true;showExplosion[i]=true;}

if(contToAttack.equals("Australia")) if(aBombXSpots[i]==450+50){ dropABombs[i] = true;showExplosion[i]=true;}

}

for(int i=0; i<dropABombs.length; i++)

{

if(dropABombs[i] && bombSoundCounter[i]==0 && keepSFX) {explosionSound.play();bombSoundCounter[i]++;}

}

boolean checkAll = false;

for(int i=0; i<showABombs.length; i++)

{

if(showABombs[i]==true || aBombSizes[i]>0 || dropABombs[i]==false || showExplosion[i]) checkAll = true;

}

if(checkAll==false)

{

timerForABombs.stop();

sendTheABomb=false;

firstTime = true;

int indexOfContAttack = 0;

double successChance = Math.random();

for(int i=0; i<continents.length; i++)

{

if(continents[i].equals(contToAttack)) indexOfContAttack=i;

}

if(true) //\*

{

if(aBombsToSend>1)eventPanel.setText("The nukes dropped in "+contToAttack+" were highly destructive.");//\*

else eventPanel.setText("The nuke dropped in "+contToAttack+" was highly destructive.");//\*

healthReduce[indexOfContAttack] -= aBombsToSend\*10+(bombLevel\*2);

if(healthReduce[indexOfContAttack]<=-97)

{

canUpdateInfo[indexOfContAttack]=true;

makeBlue[indexOfContAttack]=true;

if(startingMoney[indexOfContAttack]>=0)startingMoney[whichCont] += startingMoney[indexOfContAttack]/2;

if(startingTroops[indexOfContAttack]>=0)startingTroops[whichCont] += startingTroops[indexOfContAttack]/2;

startingTroops[indexOfContAttack]=0;

startingMoney[indexOfContAttack] = 0;

for(int i=0; i<6; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

troopPopulations[i].setText(" "+continents[i]+": "+startingTroops[i]);

}

else {jm3.setText(""+startingMoney[i]);jm5.setText(""+startingTroops[i]);}

}

}

}

else

{

if(aBombsToSend>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttack+" were unsuccessful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttack+" were unsuccessful in their mission.");//\*

}

int moneyCost = (int)(Math.random()\*10000000+1000000);

int peopleDamage = ((int)(Math.random()\*1000000+500))\*10;

if(startingMoney[indexOfContAttack]>0)startingMoney[indexOfContAttack] -= moneyCost;

populations[indexOfContAttack] -= peopleDamage;

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

}

}

removeAll();

setLayout(new GridLayout(3,3));

timer.stop();

for(int i=0; i<9; i++)

{

if(i==4) add(panelWithEvent);

else add(new JLabel(""));

}

revalidate();

repaint();

}

}

if(continent.equals("Australia"))

{

aBombXSpots[0]++;

for(int i=1; i<aBombXSpots.length; i++)

{

if(aBombXSpots[i-1]-aBombXSpots[i]>=50) aBombXSpots[i]++;

}

for(int i=0; i<aBombXSpots.length; i++)

{

if(contToAttack.equals("North America")) if(aBombXSpots[i]>=1000) { showABombs[i] = false; }

if(contToAttack.equals("South America")) if(aBombXSpots[i]>=800) { showABombs[i] = false; }

if(contToAttack.equals("Europe")) if(aBombXSpots[i]>=650) { showABombs[i] = false; }

if(contToAttack.equals("Asia")) if(aBombXSpots[i]>=400) { showABombs[i] = false; }

if(contToAttack.equals("Africa")) if(aBombXSpots[i]>=600) { showABombs[i] = false; }

}

for(int i=0; i<aBombXSpots.length; i++)

{

if(contToAttack.equals("North America")) if(aBombXSpots[i]==1000+50) {dropABombs[i] = true;showExplosion[i]=true;}

if(contToAttack.equals("South America")) if(aBombXSpots[i]==800+50) {dropABombs[i] = true;showExplosion[i]=true;}

if(contToAttack.equals("Europe")) if(aBombXSpots[i]==650+50) {dropABombs[i] = true;showExplosion[i]=true;}

if(contToAttack.equals("Asia")) if(aBombXSpots[i]==400+50) {dropABombs[i] = true;showExplosion[i]=true;}

if(contToAttack.equals("Africa")) if(aBombXSpots[i]==600+50) {dropABombs[i] = true;showExplosion[i]=true;}

}

for(int i=0; i<dropABombs.length; i++)

{

if(dropABombs[i] && bombSoundCounter[i]==0 && keepSFX) {explosionSound.play();bombSoundCounter[i]++;}

}

boolean checkAll = false;

for(int i=0; i<showABombs.length; i++)

{

if(showABombs[i]==true || aBombSizes[i]>0 || dropABombs[i]==false || showExplosion[i]) checkAll = true;

}

if(checkAll==false)

{

timerForABombs.stop();

sendTheABomb=false;

firstTime = true;

int indexOfContAttack = 0;

//double successChance = Math.random();

for(int i=0; i<continents.length; i++)

{

if(continents[i].equals(contToAttack)) indexOfContAttack=i;

}

if(true) //\*

{

if(aBombsToSend>1)eventPanel.setText("The nukes dropped in "+contToAttack+" were highly destructive.");//\*

else eventPanel.setText("The nuke dropped in "+contToAttack+" was highly destructive.");//\*

healthReduce[indexOfContAttack] -= aBombsToSend\*10+(bombLevel\*2);

if(healthReduce[indexOfContAttack]<=-97)

{

canUpdateInfo[indexOfContAttack]=true;

makeBlue[indexOfContAttack]=true;

if(startingMoney[indexOfContAttack]>=0)startingMoney[whichCont] += startingMoney[indexOfContAttack]/2;

if(startingTroops[indexOfContAttack]>=0)startingTroops[whichCont] += startingTroops[indexOfContAttack]/2;

startingTroops[indexOfContAttack]=0;

startingMoney[indexOfContAttack] = 0;

for(int i=0; i<6; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

troopPopulations[i].setText(" "+continents[i]+": "+startingTroops[i]);

}

else {jm3.setText(""+startingMoney[i]);jm5.setText(""+startingTroops[i]);}

}

}

}

else

{

if(aBombsToSend>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttack+" were unsuccessful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttack+" were unsuccessful in their mission.");//\*

}

int moneyCost = (int)(Math.random()\*10000000+1000000);

int peopleDamage = ((int)(Math.random()\*1000000+500))\*10;

if(startingMoney[indexOfContAttack]>0)startingMoney[indexOfContAttack] -= moneyCost;

populations[indexOfContAttack] -= peopleDamage;

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

}

}

removeAll();

setLayout(new GridLayout(3,3));

timer.stop();

for(int i=0; i<9; i++)

{

if(i==4) add(panelWithEvent);

else add(new JLabel(""));

}

revalidate();

repaint();

}

}

repaint();

if(healthReduce[whichCont]<=-97) {endGame=true;gameOverCount++;healthGO=true;}

if(startingMoney[whichCont]<=0) {endGame=true;moneyGO=true;gameOverCount++;}

if(populations[whichCont]<=0) {endGame=true;gameOverCount++;peopleGO=true;}

if(startingTroops[whichCont]<=0) {endGame=true;gameOverCount++;troopGO=true;}

boolean testWin = true;

for(int i=0; i<makeBlue.length; i++)

{

if(i!=whichCont)

{

if(makeBlue[i]==false) testWin=false;

}

}

if(testWin) won=true;

}

}

class EnemyNukeMover implements ActionListener

{

public void actionPerformed(ActionEvent e)

{

//System.out.println("enemy nuke moving");

if(tempContinent.equals("North America"))

{

nukeXSpotsEnemy[0]++;

for(int i=1; i<nukeXSpotsEnemy.length; i++)

{

if(nukeXSpotsEnemy[i-1]-nukeXSpotsEnemy[i]>=50) nukeXSpotsEnemy[i]++;

}

for(int i=0; i<nukeXSpotsEnemy.length; i++)

{

//System.out.println(i);

if(contToAttackEnemy.equals("South America")) if(nukeXSpotsEnemy[i]>=550){ showNukesEnemy[i] = false; }

if(contToAttackEnemy.equals("Europe")) if(nukeXSpotsEnemy[i]>=750) { showNukesEnemy[i] = false; }

if(contToAttackEnemy.equals("Asia")) if(nukeXSpotsEnemy[i]>=1000) { showNukesEnemy[i] = false; }

if(contToAttackEnemy.equals("Africa")) if(nukeXSpotsEnemy[i]>=700) { showNukesEnemy[i] = false; }

if(contToAttackEnemy.equals("Australia")) if(nukeXSpotsEnemy[i]>=1200) { showNukesEnemy[i] = false; }

}

for(int i=0; i<nukeXSpotsEnemy.length; i++)

{

//System.out.println(i);

if(contToAttackEnemy.equals("South America")) if(nukeXSpotsEnemy[i]==550+50){dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

if(contToAttackEnemy.equals("Europe")) if(nukeXSpotsEnemy[i]==750+50){dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

if(contToAttackEnemy.equals("Asia")) if(nukeXSpotsEnemy[i]==1000+50){dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

if(contToAttackEnemy.equals("Africa")) if(nukeXSpotsEnemy[i]==700+50){dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

if(contToAttackEnemy.equals("Australia")) if(nukeXSpotsEnemy[i]==1200+50) {dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

}

for(int i=0; i<dropNukesEnemy.length; i++)

{

if(dropNukesEnemy[i] && bombSoundCounterEnemy[i]==0) {explosionSound.play();bombSoundCounterEnemy[i]++;}

}

boolean checkAll = false;

for(int i=0; i<showNukesEnemy.length; i++)

{

if(showNukesEnemy[i]==true || nukeSizesEnemy[i]>0 || dropNukesEnemy[i]==false || showExplosionEnemy[i]) checkAll = true;

}

if(checkAll==false)

{

timerForNukeEnemy.stop();

sendTheNukeEnemy=false;

firstTimeEnemy = true;

int indexOfContAttack = 0;

double successChance = Math.random();

for(int i=0; i<continents.length; i++)

{

if(continents[i].equals(contToAttackEnemy)) indexOfContAttack=i;

}

if(true) //\*

{

//if(nukesToSendEnemy>1)eventPanel.setText("The nukes dropped in "+contToAttackEnemy+" were highly destructive.");//\*

//else eventPanel.setText("The nuke dropped in "+contToAttackEnemy+" was highly destructive.");//\*

healthReduce[indexOfContAttack] -= nukesToSendEnemy\*15;

if(healthReduce[indexOfContAttack]<=-97)

{

//System.out.println("dude");

canUpdateInfo[indexOfContAttack]=true;

makeBlue[indexOfContAttack]=true;

if(startingMoney[indexOfContAttack]>=0)startingMoney[attackingCont] += startingMoney[indexOfContAttack];

if(startingTroops[indexOfContAttack]>=0)startingTroops[attackingCont] += startingTroops[indexOfContAttack];

startingTroops[indexOfContAttack]=0;

startingMoney[indexOfContAttack] = 0;

// System.out.println(startingMoney[indexOfContAttack]);//???

for(int i=0; i<6; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

troopPopulations[i].setText(" "+continents[i]+": "+startingTroops[i]);

}

else {jm3.setText(""+startingMoney[i]);jm5.setText(""+startingTroops[i]);}

}

//revalidate();

repaint();

}

}

else

{

//if(nukesToSendEnemy>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttackEnemy+" were unsuccessful in their mission.");//\*

//else eventPanel.setText("The troops deployed by jet sent to "+contToAttackEnemy+" were unsuccessful in their mission.");//\*

}

int moneyCost = (int)(Math.random()\*10000000+1000000);

int peopleDamage = ((int)(Math.random()\*1000000+500))\*10;

startingMoney[indexOfContAttack] -= moneyCost;

populations[indexOfContAttack] -= peopleDamage;

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

jm3.setText(""+startingMoney[i]);

jm4.setText(""+populations[i]);

}

}

timer.stop();

removeAll();

setLayout(new GridLayout(3,3));

for(int i=0; i<9; i++)

{

if(i==4) {add(panelWithEvent);}

else add(new JLabel(""));

}

revalidate();

repaint();

}

}

if(tempContinent.equals("South America"))

{

nukeXSpotsEnemy[0]++;

for(int i=1; i<nukeXSpotsEnemy.length; i++)

{

if(nukeXSpotsEnemy[i-1]-nukeXSpotsEnemy[i]>=50) nukeXSpotsEnemy[i]++;

}

for(int i=0; i<nukeXSpotsEnemy.length; i++)

{

if(contToAttackEnemy.equals("North America")) if(nukeXSpotsEnemy[i]>=300) { showNukesEnemy[i] = false; }

if(contToAttackEnemy.equals("Europe")) if(nukeXSpotsEnemy[i]>=450) { showNukesEnemy[i] = false; }

if(contToAttackEnemy.equals("Asia")) if(nukeXSpotsEnemy[i]>=700) { showNukesEnemy[i] = false; }

if(contToAttackEnemy.equals("Africa")) if(nukeXSpotsEnemy[i]>=425) { showNukesEnemy[i] = false; }

if(contToAttackEnemy.equals("Australia")) if(nukeXSpotsEnemy[i]>=850) { showNukesEnemy[i] = false; }

}

for(int i=0; i<nukeXSpotsEnemy.length; i++)

{

if(contToAttackEnemy.equals("North America")) if(nukeXSpotsEnemy[i]==300+50) {dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

if(contToAttackEnemy.equals("Europe")) if(nukeXSpotsEnemy[i]==450+50) {dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

if(contToAttackEnemy.equals("Asia")) if(nukeXSpotsEnemy[i]==700+50) {dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

if(contToAttackEnemy.equals("Africa")) if(nukeXSpotsEnemy[i]==425+50) {dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

if(contToAttackEnemy.equals("Australia")) if(nukeXSpotsEnemy[i]==850+50){dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

}

boolean checkAll = false;

for(int i=0; i<dropNukesEnemy.length; i++)

{

if(dropNukesEnemy[i] && bombSoundCounterEnemy[i]==0) {explosionSound.play();bombSoundCounterEnemy[i]++;}

}

for(int i=0; i<showNukesEnemy.length; i++)

{

if(showNukesEnemy[i]==true || nukeSizesEnemy[i]>0 || dropNukesEnemy[i]==false || showExplosionEnemy[i]) checkAll = true;

}

if(checkAll==false)

{

timerForNukeEnemy.stop();

sendTheNukeEnemy=false;

firstTimeEnemy = true;

int indexOfContAttack = 0;

double successChance = Math.random();

for(int i=0; i<continents.length; i++)

{

if(continents[i].equals(contToAttackEnemy)) indexOfContAttack=i;

}

if(true) //\*

{

//if(nukesToSendEnemy>1)eventPanel.setText("The nukes dropped in "+contToAttackEnemy+" were highly destructive.");//\*

//else eventPanel.setText("The nuke dropped in "+contToAttackEnemy+" was highly destructive.");//\*

healthReduce[indexOfContAttack] -= nukesToSendEnemy\*15;

if(healthReduce[indexOfContAttack]<=-97)

{

//System.out.println("dude");

canUpdateInfo[indexOfContAttack]=true;

makeBlue[indexOfContAttack]=true;

if(startingMoney[indexOfContAttack]>=0)startingMoney[attackingCont] += startingMoney[indexOfContAttack];

if(startingTroops[indexOfContAttack]>=0)startingTroops[attackingCont] += startingTroops[indexOfContAttack];

startingTroops[indexOfContAttack]=0;

startingMoney[indexOfContAttack] = 0;

// System.out.println(startingMoney[indexOfContAttack]);//???

for(int i=0; i<6; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

troopPopulations[i].setText(" "+continents[i]+": "+startingTroops[i]);

}

else {jm3.setText(""+startingMoney[i]);jm5.setText(""+startingTroops[i]);}

}

//revalidate();

repaint();

}

}

else

{

//if(nukesToSendEnemy>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttackEnemy+" were unsuccessful in their mission.");//\*

//else eventPanel.setText("The troops deployed by jet sent to "+contToAttackEnemy+" were unsuccessful in their mission.");//\*

}

//System.out.println("hey");

int moneyCost = (int)(Math.random()\*10000000+1000000);

int peopleDamage = ((int)(Math.random()\*1000000+500))\*10;

startingMoney[indexOfContAttack] -= moneyCost;

populations[indexOfContAttack] -= peopleDamage;

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

}

else

{

jm3.setText(""+startingMoney[i]);

jm4.setText(""+populations[i]);

}

}

timer.stop();

removeAll();

setLayout(new GridLayout(3,3));

for(int i=0; i<9; i++)

{

if(i==4) {add(panelWithEvent);}

else add(new JLabel(""));

}

revalidate();

repaint();

}

}

if(tempContinent.equals("Europe"))

{

nukeXSpotsEnemy[0]++;

for(int i=1; i<nukeXSpotsEnemy.length; i++)

{

if(nukeXSpotsEnemy[i-1]-nukeXSpotsEnemy[i]>=50) nukeXSpotsEnemy[i]++;

}

for(int i=0; i<nukeXSpotsEnemy.length; i++)

{

if(contToAttackEnemy.equals("North America")) if(nukeXSpotsEnemy[i]>=575) { showNukesEnemy[i] = false; }

if(contToAttackEnemy.equals("South America")) if(nukeXSpotsEnemy[i]>=500) { showNukesEnemy[i] = false; }

if(contToAttackEnemy.equals("Asia")) if(nukeXSpotsEnemy[i]>=350) { showNukesEnemy[i] = false; }

if(contToAttackEnemy.equals("Africa")) if(nukeXSpotsEnemy[i]>=250) { showNukesEnemy[i] = false;}

if(contToAttackEnemy.equals("Australia")) if(nukeXSpotsEnemy[i]>=600) { showNukesEnemy[i] = false;}

}

for(int i=0; i<nukeXSpotsEnemy.length; i++)

{

if(contToAttackEnemy.equals("North America")) if(nukeXSpotsEnemy[i]==575+50){dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

if(contToAttackEnemy.equals("South America")) if(nukeXSpotsEnemy[i]==500+50) {dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

if(contToAttackEnemy.equals("Asia")) if(nukeXSpotsEnemy[i]==350+50) {dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

if(contToAttackEnemy.equals("Africa")) if(nukeXSpotsEnemy[i]==250+50) {dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

if(contToAttackEnemy.equals("Australia")) if(nukeXSpotsEnemy[i]==600+50) {dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

}

boolean checkAll = false;

for(int i=0; i<dropNukesEnemy.length; i++)

{

if(dropNukesEnemy[i] && bombSoundCounterEnemy[i]==0) {explosionSound.play();bombSoundCounterEnemy[i]++;}

}

for(int i=0; i<showNukesEnemy.length; i++)

{

if(showNukesEnemy[i]==true || nukeSizesEnemy[i]>0 || dropNukesEnemy[i]==false || showExplosionEnemy[i]) checkAll = true;

}

if(checkAll==false)

{

timerForNukeEnemy.stop();

sendTheNukeEnemy=false;

firstTimeEnemy = true;

int indexOfContAttack = 0;

double successChance = Math.random();

for(int i=0; i<continents.length; i++)

{

if(continents[i].equals(contToAttackEnemy)) indexOfContAttack=i;

}

if(true) //\*

{

//if(nukesToSendEnemy>1)eventPanel.setText("The nukes dropped in "+contToAttackEnemy+" were highly destructive.");//\*

//else eventPanel.setText("The nuke dropped in "+contToAttackEnemy+" was highly destructive.");//\*

healthReduce[indexOfContAttack] -= nukesToSendEnemy\*15;

if(healthReduce[indexOfContAttack]<=-97)

{

//System.out.println("dude");

canUpdateInfo[indexOfContAttack]=true;

makeBlue[indexOfContAttack]=true;

if(startingMoney[indexOfContAttack]>=0)startingMoney[attackingCont] += startingMoney[indexOfContAttack];

if(startingTroops[indexOfContAttack]>=0)startingTroops[attackingCont] += startingTroops[indexOfContAttack];

startingTroops[indexOfContAttack]=0;

startingMoney[indexOfContAttack] = 0;

// System.out.println(startingMoney[indexOfContAttack]);//???

for(int i=0; i<6; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

troopPopulations[i].setText(" "+continents[i]+": "+startingTroops[i]);

}

else {jm3.setText(""+startingMoney[i]);jm5.setText(""+startingTroops[i]);}

}

//revalidate();

repaint();

}

}

else

{

//if(nukesToSendEnemy>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttackEnemy+" were unsuccessful in their mission.");//\*

//else eventPanel.setText("The troops deployed by jet sent to "+contToAttackEnemy+" were unsuccessful in their mission.");//\*

}

int moneyCost = (int)(Math.random()\*10000000+1000000);

int peopleDamage = ((int)(Math.random()\*1000000+500))\*10;

startingMoney[indexOfContAttack] -= moneyCost;

populations[indexOfContAttack] -= peopleDamage;

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

}

else

{

jm3.setText(""+startingMoney[i]);

jm4.setText(""+populations[i]);

}

}

timer.stop();

removeAll();

setLayout(new GridLayout(3,3));

for(int i=0; i<9; i++)

{

if(i==4) {add(panelWithEvent);}

else add(new JLabel(""));

}

revalidate();

repaint();

}

}

if(tempContinent.equals("Asia"))

{

nukeXSpotsEnemy[0]++;

for(int i=1; i<nukeXSpotsEnemy.length; i++)

{

if(nukeXSpotsEnemy[i-1]-nukeXSpotsEnemy[i]>=50) nukeXSpotsEnemy[i]++;

}

for(int i=0; i<nukeXSpotsEnemy.length; i++)

{

if(contToAttackEnemy.equals("North America")) if(nukeXSpotsEnemy[i]>=700) { showNukesEnemy[i] = false; }

if(contToAttackEnemy.equals("South America")) if(nukeXSpotsEnemy[i]>=700) { showNukesEnemy[i] = false; }

if(contToAttackEnemy.equals("Europe")) if(nukeXSpotsEnemy[i]>=350) { showNukesEnemy[i] = false; }

if(contToAttackEnemy.equals("Africa")) if(nukeXSpotsEnemy[i]>=400) { showNukesEnemy[i] = false; }

if(contToAttackEnemy.equals("Australia")) if(nukeXSpotsEnemy[i]>=400) { showNukesEnemy[i] = false; }

}

for(int i=0; i<nukeXSpotsEnemy.length; i++)

{

if(contToAttackEnemy.equals("North America")) if(nukeXSpotsEnemy[i]==700+50) {dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

if(contToAttackEnemy.equals("South America")) if(nukeXSpotsEnemy[i]==700+50) {dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

if(contToAttackEnemy.equals("Europe")) if(nukeXSpotsEnemy[i]==350+50) {dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

if(contToAttackEnemy.equals("Africa")) if(nukeXSpotsEnemy[i]==400+50){dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

if(contToAttackEnemy.equals("Australia")) if(nukeXSpotsEnemy[i]==400+50){dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

}

boolean checkAll = false;

for(int i=0; i<dropNukesEnemy.length; i++)

{

if(dropNukesEnemy[i] && bombSoundCounterEnemy[i]==0) {explosionSound.play();bombSoundCounterEnemy[i]++;}

}

for(int i=0; i<showNukesEnemy.length; i++)

{

if(showNukesEnemy[i]==true || nukeSizesEnemy[i]>0 || dropNukesEnemy[i]==false || showExplosionEnemy[i]) checkAll = true;

}

if(checkAll==false)

{

timerForNukeEnemy.stop();

sendTheNukeEnemy=false;

firstTimeEnemy = true;

int indexOfContAttack = 0;

double successChance = Math.random();

for(int i=0; i<continents.length; i++)

{

if(continents[i].equals(contToAttackEnemy)) indexOfContAttack=i;

}

if(true) //\*

{

//if(nukesToSendEnemy>1)eventPanel.setText("The nukes dropped in "+contToAttackEnemy+" were highly destructive.");//\*

//else eventPanel.setText("The nuke dropped in "+contToAttackEnemy+" was highly destructive.");//\*

healthReduce[indexOfContAttack] -= nukesToSendEnemy\*15;

if(healthReduce[indexOfContAttack]<=-97)

{

//System.out.println("dude");

canUpdateInfo[indexOfContAttack]=true;

makeBlue[indexOfContAttack]=true;

if(startingMoney[indexOfContAttack]>=0)startingMoney[attackingCont] += startingMoney[indexOfContAttack];

if(startingTroops[indexOfContAttack]>=0)startingTroops[attackingCont] += startingTroops[indexOfContAttack];

startingTroops[indexOfContAttack]=0;

startingMoney[indexOfContAttack] = 0;

// System.out.println(startingMoney[indexOfContAttack]);//???

for(int i=0; i<6; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

troopPopulations[i].setText(" "+continents[i]+": "+startingTroops[i]);

}

else {jm3.setText(""+startingMoney[i]);jm5.setText(""+startingTroops[i]);}

}

//revalidate();

repaint();

}

}

else

{

//if(nukesToSendEnemy>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttackEnemy+" were unsuccessful in their mission.");//\*

//else eventPanel.setText("The troops deployed by jet sent to "+contToAttackEnemy+" were unsuccessful in their mission.");//\*

}

int moneyCost = (int)(Math.random()\*10000000+1000000);

int peopleDamage = ((int)(Math.random()\*1000000+500))\*10;

startingMoney[indexOfContAttack] -= moneyCost;

populations[indexOfContAttack] -= peopleDamage;

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

}

else

{

jm3.setText(""+startingMoney[i]);

jm4.setText(""+populations[i]);

}

}

timer.stop();

removeAll();

setLayout(new GridLayout(3,3));

for(int i=0; i<9; i++)

{

if(i==4) {add(panelWithEvent);}

else add(new JLabel(""));

}

revalidate();

repaint();

}

}

if(tempContinent.equals("Africa"))

{

nukeXSpotsEnemy[0]++;

for(int i=1; i<nukeXSpotsEnemy.length; i++)

{

if(nukeXSpotsEnemy[i-1]-nukeXSpotsEnemy[i]>=50) nukeXSpotsEnemy[i]++;

}

for(int i=0; i<nukeXSpotsEnemy.length; i++)

{

if(contToAttackEnemy.equals("North America")) if(nukeXSpotsEnemy[i]>=500) { showNukesEnemy[i] = false; }

if(contToAttackEnemy.equals("South America")) if(nukeXSpotsEnemy[i]>=300) { showNukesEnemy[i] = false; }

if(contToAttackEnemy.equals("Europe")) if(nukeXSpotsEnemy[i]>=325) { showNukesEnemy[i] = false;}

if(contToAttackEnemy.equals("Asia")) if(nukeXSpotsEnemy[i]>=400) { showNukesEnemy[i] = false; }

if(contToAttackEnemy.equals("Australia")) if(nukeXSpotsEnemy[i]>=450) { showNukesEnemy[i] = false; }

}

for(int i=0; i<nukeXSpotsEnemy.length; i++)

{

if(contToAttackEnemy.equals("North America")) if(nukeXSpotsEnemy[i]==500+50) {dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

if(contToAttackEnemy.equals("South America")) if(nukeXSpotsEnemy[i]==300+50) {dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

if(contToAttackEnemy.equals("Europe")) if(nukeXSpotsEnemy[i]==325+50) {dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

if(contToAttackEnemy.equals("Asia")) if(nukeXSpotsEnemy[i]==400+50) {dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

if(contToAttackEnemy.equals("Australia")) if(nukeXSpotsEnemy[i]==450+50){dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

}

boolean checkAll = false;

for(int i=0; i<dropNukesEnemy.length; i++)

{

if(dropNukesEnemy[i] && bombSoundCounterEnemy[i]==0) {explosionSound.play();bombSoundCounterEnemy[i]++;}

}

for(int i=0; i<showNukesEnemy.length; i++)

{

if(showNukesEnemy[i]==true || nukeSizesEnemy[i]>0 || dropNukesEnemy[i]==false || showExplosionEnemy[i]) checkAll = true;

}

if(checkAll==false)

{

timerForNukeEnemy.stop();

sendTheNukeEnemy=false;

firstTimeEnemy = true;

int indexOfContAttack = 0;

double successChance = Math.random();

for(int i=0; i<continents.length; i++)

{

if(continents[i].equals(contToAttackEnemy)) indexOfContAttack=i;

}

if(true) //\*

{

//if(nukesToSendEnemy>1)eventPanel.setText("The nukes dropped in "+contToAttackEnemy+" were highly destructive.");//\*

//else eventPanel.setText("The nuke dropped in "+contToAttackEnemy+" was highly destructive.");//\*

healthReduce[indexOfContAttack] -= nukesToSendEnemy\*15;

if(healthReduce[indexOfContAttack]<=-97)

{

//System.out.println("dude");

canUpdateInfo[indexOfContAttack]=true;

makeBlue[indexOfContAttack]=true;

if(startingMoney[indexOfContAttack]>=0)startingMoney[attackingCont] += startingMoney[indexOfContAttack];

if(startingTroops[indexOfContAttack]>=0)startingTroops[attackingCont] += startingTroops[indexOfContAttack];

startingTroops[indexOfContAttack]=0;

startingMoney[indexOfContAttack] = 0;

// System.out.println(startingMoney[indexOfContAttack]);//???

for(int i=0; i<6; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

troopPopulations[i].setText(" "+continents[i]+": "+startingTroops[i]);

}

else {jm3.setText(""+startingMoney[i]);jm5.setText(""+startingTroops[i]);}

}

//revalidate();

repaint();

}

}

else

{

//if(nukesToSendEnemy>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttackEnemy+" were unsuccessful in their mission.");//\*

//else eventPanel.setText("The troops deployed by jet sent to "+contToAttackEnemy+" were unsuccessful in their mission.");//\*

}

int moneyCost = (int)(Math.random()\*10000000+1000000);

int peopleDamage = ((int)(Math.random()\*1000000+500))\*10;

startingMoney[indexOfContAttack] -= moneyCost;

populations[indexOfContAttack] -= peopleDamage;

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

}

else

{

jm3.setText(""+startingMoney[i]);

jm4.setText(""+populations[i]);

}

}

timer.stop();

removeAll();

setLayout(new GridLayout(3,3));

for(int i=0; i<9; i++)

{

if(i==4) {add(panelWithEvent);}

else add(new JLabel(""));

}

revalidate();

repaint();

}

}

if(tempContinent.equals("Australia"))

{

nukeXSpotsEnemy[0]++;

for(int i=1; i<nukeXSpotsEnemy.length; i++)

{

if(nukeXSpotsEnemy[i-1]-nukeXSpotsEnemy[i]>=50) nukeXSpotsEnemy[i]++;

}

for(int i=0; i<nukeXSpotsEnemy.length; i++)

{

if(contToAttackEnemy.equals("North America")) if(nukeXSpotsEnemy[i]>=1000) { showNukesEnemy[i] = false; }

if(contToAttackEnemy.equals("South America")) if(nukeXSpotsEnemy[i]>=800) { showNukesEnemy[i] = false; }

if(contToAttackEnemy.equals("Europe")) if(nukeXSpotsEnemy[i]>=650) { showNukesEnemy[i] = false;}

if(contToAttackEnemy.equals("Asia")) if(nukeXSpotsEnemy[i]>=400) { showNukesEnemy[i] = false; }

if(contToAttackEnemy.equals("Africa")) if(nukeXSpotsEnemy[i]>=600) { showNukesEnemy[i] = false; }

}

for(int i=0; i<nukeXSpotsEnemy.length; i++)

{

if(contToAttackEnemy.equals("North America")) if(nukeXSpotsEnemy[i]==1000+50) {dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

if(contToAttackEnemy.equals("South America")) if(nukeXSpotsEnemy[i]==800+50) {dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

if(contToAttackEnemy.equals("Europe")) if(nukeXSpotsEnemy[i]==650+50) {dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

if(contToAttackEnemy.equals("Asia")) if(nukeXSpotsEnemy[i]==400+50) {dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

if(contToAttackEnemy.equals("Africa")) if(nukeXSpotsEnemy[i]==600+50) {dropNukesEnemy[i]=true;showExplosionEnemy[i]=true;}

}

boolean checkAll = false;

for(int i=0; i<dropNukesEnemy.length; i++)

{

if(dropNukesEnemy[i] && bombSoundCounterEnemy[i]==0) {explosionSound.play();bombSoundCounterEnemy[i]++;}

}

for(int i=0; i<showNukesEnemy.length; i++)

{

if(showNukesEnemy[i]==true || nukeSizesEnemy[i]>0 || dropNukesEnemy[i]==false || showExplosionEnemy[i]) checkAll = true;

}

if(checkAll==false)

{

timerForNukeEnemy.stop();

sendTheNukeEnemy=false;

firstTimeEnemy = true;

int indexOfContAttack = 0;

//double successChance = Math.random();

for(int i=0; i<continents.length; i++)

{

if(continents[i].equals(contToAttackEnemy)) indexOfContAttack=i;

}

if(true) //\*

{

//if(nukesToSendEnemy>1)eventPanel.setText("The nukes dropped in "+contToAttackEnemy+" were highly destructive.");//\*

//else eventPanel.setText("The nuke dropped in "+contToAttackEnemy+" was highly destructive.");//\*

healthReduce[indexOfContAttack] -= nukesToSendEnemy\*15;

if(healthReduce[indexOfContAttack]<=-97)

{

//System.out.println("dude");

canUpdateInfo[indexOfContAttack]=true;

makeBlue[indexOfContAttack]=true;

if(startingMoney[indexOfContAttack]>=0)startingMoney[attackingCont] += startingMoney[indexOfContAttack];

if(startingTroops[indexOfContAttack]>=0)startingTroops[attackingCont] += startingTroops[indexOfContAttack];

startingTroops[indexOfContAttack]=0;

startingMoney[indexOfContAttack] = 0;

// System.out.println(startingMoney[indexOfContAttack]);//???

for(int i=0; i<6; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

troopPopulations[i].setText(" "+continents[i]+": "+startingTroops[i]);

}

else {jm3.setText(""+startingMoney[i]);jm5.setText(""+startingTroops[i]);}

}

//revalidate();

repaint();

}

}

else

{

//if(nukesToSendEnemy>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttackEnemy+" were unsuccessful in their mission.");//\*

//else eventPanel.setText("The troops deployed by jet sent to "+contToAttackEnemy+" were unsuccessful in their mission.");//\*

}

int moneyCost = (int)(Math.random()\*10000000+1000000);

int peopleDamage = ((int)(Math.random()\*1000000+500))\*10;

startingMoney[indexOfContAttack] -= moneyCost;

populations[indexOfContAttack] -= peopleDamage;

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

}

else

{

jm3.setText(""+startingMoney[i]);

jm4.setText(""+populations[i]);

}

}

timer.stop();

removeAll();

setLayout(new GridLayout(3,3));

for(int i=0; i<9; i++)

{

if(i==4) {add(panelWithEvent);}

else add(new JLabel(""));

}

revalidate();

repaint();

}

}

repaint();

if(healthReduce[whichCont]<=-97) {endGame=true;gameOverCount++;healthGO=true;}

if(startingMoney[whichCont]<=0) {endGame=true;moneyGO=true;gameOverCount++;}

if(populations[whichCont]<=0) {endGame=true;gameOverCount++;peopleGO=true;}

if(startingTroops[whichCont]<=0) {endGame=true;gameOverCount++;troopGO=true;}

boolean testWin = true;

for(int i=0; i<makeBlue.length; i++)

{

if(i!=whichCont)

{

if(makeBlue[i]==false) testWin=false;

}

}

if(testWin) won=true;

}

}

class EnemyPlaneMover implements ActionListener

{

public void actionPerformed(ActionEvent e)

{

//System.out.println("enemy plane moving");

if(tempContinent.equals("North America"))

{

planeXSpotsEnemy[0]++;

for(int i=1; i<planeXSpotsEnemy.length; i++)

{

if(planeXSpotsEnemy[i-1]-planeXSpotsEnemy[i]>=50) planeXSpotsEnemy[i]++;

}

for(int i=0; i<planeXSpotsEnemy.length; i++)

{

if(contToAttackEnemy.equals("South America")) if(planeXSpotsEnemy[i]>=550) showPlanesEnemy[i] = false;//\*

if(contToAttackEnemy.equals("Europe")) if(planeXSpotsEnemy[i]>=750) showPlanesEnemy[i] = false;//\*

if(contToAttackEnemy.equals("Asia")) if(planeXSpotsEnemy[i]>=1000) showPlanesEnemy[i] = false;//\*

if(contToAttackEnemy.equals("Africa")) if(planeXSpotsEnemy[i]>=700) showPlanesEnemy[i] = false;//\*

if(contToAttackEnemy.equals("Australia")) if(planeXSpotsEnemy[i]>=1200) showPlanesEnemy[i] = false;//\*

}

boolean checkAll = false;

for(int i=0; i<showPlanesEnemy.length; i++)

{

if(showPlanesEnemy[i]==true) checkAll = true;

}

if(checkAll==false)

{

timerForPlaneEnemy.stop();

sendThePlaneEnemy=false;

firstTimeEnemy = true;

int indexOfContAttack = 0;

double successChance = Math.random();

for(int i=0; i<continents.length; i++)

{

if(continents[i].equals(contToAttackEnemy)) indexOfContAttack=i;

}

if(successChance<0.75-defenseCount[indexOfContAttack]) //\*

{

if(planesToSendEnemy>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttackEnemy+" were successful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttackEnemy+" were successful in their mission.");//\*

healthReduce[indexOfContAttack] -= planesToSendEnemy;

int moneyCost = (int)(Math.random()\*100000+10000);

int peopleDamage = ((int)(Math.random()\*1000+100));

startingMoney[indexOfContAttack] -= moneyCost;

populations[indexOfContAttack] -= peopleDamage;

if(healthReduce[indexOfContAttack]<=-97)

{

// System.out.println("dude");

canUpdateInfo[indexOfContAttack]=true;

makeBlue[indexOfContAttack]=true;

if(startingMoney[indexOfContAttack]>=0)startingMoney[attackingCont] += startingMoney[indexOfContAttack];

if(startingTroops[indexOfContAttack]>=0)startingTroops[attackingCont] += startingTroops[indexOfContAttack];

startingTroops[indexOfContAttack]=0;

startingMoney[indexOfContAttack] = 0;

// System.out.println(startingMoney[indexOfContAttack]);//???

for(int i=0; i<6; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

troopPopulations[i].setText(" "+continents[i]+": "+startingTroops[i]);

//continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

}

else {jm3.setText(""+startingMoney[i]);jm5.setText(""+startingTroops[i]);jm4.setText(""+populations[i]);}

}

//revalidate();

repaint();

}

}

else

{

if(planesToSendEnemy>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttackEnemy+" were unsuccessful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttackEnemy+" were unsuccessful in their mission.");//\*

}

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

}

else

{

jm3.setText(""+startingMoney[i]);

jm4.setText(""+populations[i]);

}

}

removeAll();

setLayout(new GridLayout(3,3));

timer.stop();

for(int i=0; i<9; i++)

{

if(i==4) add(panelWithEvent);

else add(new JLabel(""));

}

revalidate();

repaint();

}

}

if(tempContinent.equals("South America"))

{

planeXSpotsEnemy[0]++;

for(int i=1; i<planeXSpotsEnemy.length; i++)

{

if(planeXSpotsEnemy[i-1]-planeXSpotsEnemy[i]>=50) planeXSpotsEnemy[i]++;

}

for(int i=0; i<planeXSpotsEnemy.length; i++)

{

if(contToAttackEnemy.equals("North America")) if(planeXSpotsEnemy[i]>=300) showPlanesEnemy[i] = false;//\*

if(contToAttackEnemy.equals("Europe")) if(planeXSpotsEnemy[i]>=450) showPlanesEnemy[i] = false;//\*

if(contToAttackEnemy.equals("Asia")) if(planeXSpotsEnemy[i]>=700) showPlanesEnemy[i] = false;//\*

if(contToAttackEnemy.equals("Africa")) if(planeXSpotsEnemy[i]>=425) showPlanesEnemy[i] = false;//\*

if(contToAttackEnemy.equals("Australia")) if(planeXSpotsEnemy[i]>=850) showPlanesEnemy[i] = false;//\*

}

boolean checkAll = false;

for(int i=0; i<showPlanesEnemy.length; i++)

{

if(showPlanesEnemy[i]==true) checkAll = true;

}

if(checkAll==false)

{

timerForPlaneEnemy.stop();

sendThePlaneEnemy=false;

firstTimeEnemy = true;

int indexOfContAttack = 0;

double successChance = Math.random();

for(int i=0; i<continents.length; i++)

{

if(continents[i].equals(contToAttackEnemy)) indexOfContAttack=i;

}

if(successChance<0.75-defenseCount[indexOfContAttack]) //\*

{

if(planesToSendEnemy>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttackEnemy+" were successful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttackEnemy+" were successful in their mission.");//\*

healthReduce[indexOfContAttack] -= planesToSendEnemy;

int moneyCost = (int)(Math.random()\*100000+10000);

int peopleDamage = ((int)(Math.random()\*1000+100));

startingMoney[indexOfContAttack] -= moneyCost;

populations[indexOfContAttack] -= peopleDamage;

if(healthReduce[indexOfContAttack]<=-97)

{

//System.out.println("dude");

canUpdateInfo[indexOfContAttack]=true;

makeBlue[indexOfContAttack]=true;

if(startingMoney[indexOfContAttack]>=0)startingMoney[attackingCont] += startingMoney[indexOfContAttack];

if(startingTroops[indexOfContAttack]>=0)startingTroops[attackingCont] += startingTroops[indexOfContAttack];

startingTroops[indexOfContAttack]=0;

startingMoney[indexOfContAttack] = 0;

// System.out.println(startingMoney[indexOfContAttack]);//???

for(int i=0; i<6; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

troopPopulations[i].setText(" "+continents[i]+": "+startingTroops[i]);

}

else {jm3.setText(""+startingMoney[i]);jm5.setText(""+startingTroops[i]);}

}

//revalidate();

repaint();

}

}

else

{

if(planesToSendEnemy>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttackEnemy+" were unsuccessful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttackEnemy+" were unsuccessful in their mission.");//\*

}

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

}

else

{

jm3.setText(""+startingMoney[i]);

jm4.setText(""+populations[i]);

}

}

removeAll();

setLayout(new GridLayout(3,3));

timer.stop();

for(int i=0; i<9; i++)

{

if(i==4) add(panelWithEvent);

else add(new JLabel(""));

}

revalidate();

repaint();

}

}

if(tempContinent.equals("Europe"))

{

planeXSpotsEnemy[0]++;

for(int i=1; i<planeXSpotsEnemy.length; i++)

{

if(planeXSpotsEnemy[i-1]-planeXSpotsEnemy[i]>=50) planeXSpotsEnemy[i]++;

}

for(int i=0; i<planeXSpotsEnemy.length; i++)

{

if(contToAttackEnemy.equals("North America")) if(planeXSpotsEnemy[i]>=575) showPlanesEnemy[i] = false;//\*

if(contToAttackEnemy.equals("South America")) if(planeXSpotsEnemy[i]>=500) showPlanesEnemy[i] = false;//\*

if(contToAttackEnemy.equals("Asia")) if(planeXSpotsEnemy[i]>=350) showPlanesEnemy[i] = false;//\*

if(contToAttackEnemy.equals("Africa")) if(planeXSpotsEnemy[i]>=250) showPlanesEnemy[i] = false;//\*

if(contToAttackEnemy.equals("Australia")) if(planeXSpotsEnemy[i]>=600) showPlanesEnemy[i] = false;//\*

}

boolean checkAll = false;

for(int i=0; i<showPlanesEnemy.length; i++)

{

if(showPlanesEnemy[i]==true) checkAll = true;

}

if(checkAll==false)

{

timerForPlaneEnemy.stop();

sendThePlaneEnemy=false;

firstTimeEnemy = true;

int indexOfContAttack = 0;

double successChance = Math.random();

for(int i=0; i<continents.length; i++)

{

if(continents[i].equals(contToAttackEnemy)) indexOfContAttack=i;

}

if(successChance<0.75-defenseCount[indexOfContAttack]) //\*

{

if(planesToSendEnemy>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttackEnemy+" were successful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttackEnemy+" were successful in their mission.");//\*

healthReduce[indexOfContAttack] -= planesToSendEnemy;

int moneyCost = (int)(Math.random()\*100000+10000);

int peopleDamage = ((int)(Math.random()\*1000+100));

startingMoney[indexOfContAttack] -= moneyCost;

populations[indexOfContAttack] -= peopleDamage;

if(healthReduce[indexOfContAttack]<=-97)

{

//System.out.println("dude");

canUpdateInfo[indexOfContAttack]=true;

makeBlue[indexOfContAttack]=true;

if(startingMoney[indexOfContAttack]>=0)startingMoney[attackingCont] += startingMoney[indexOfContAttack];

if(startingTroops[indexOfContAttack]>=0)startingTroops[attackingCont] += startingTroops[indexOfContAttack];

startingTroops[indexOfContAttack]=0;

startingMoney[indexOfContAttack] = 0;

// System.out.println(startingMoney[indexOfContAttack]);//???

for(int i=0; i<6; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

troopPopulations[i].setText(" "+continents[i]+": "+startingTroops[i]);

}

else {jm3.setText(""+startingMoney[i]);jm5.setText(""+startingTroops[i]);}

}

//revalidate();

repaint();

}

}

else

{

if(planesToSendEnemy>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttackEnemy+" were unsuccessful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttackEnemy+" were unsuccessful in their mission.");//\*

}

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

}

else

{

jm3.setText(""+startingMoney[i]);

jm4.setText(""+populations[i]);

}

}

removeAll();

setLayout(new GridLayout(3,3));

timer.stop();

for(int i=0; i<9; i++)

{

if(i==4) add(panelWithEvent);

else add(new JLabel(""));

}

revalidate();

repaint();

}

}

if(tempContinent.equals("Asia"))

{

planeXSpotsEnemy[0]++;

for(int i=1; i<planeXSpotsEnemy.length; i++)

{

if(planeXSpotsEnemy[i-1]-planeXSpotsEnemy[i]>=50) planeXSpotsEnemy[i]++;

}

for(int i=0; i<planeXSpotsEnemy.length; i++)

{

if(contToAttackEnemy.equals("North America")) if(planeXSpotsEnemy[i]>=700) showPlanesEnemy[i] = false;//\*

if(contToAttackEnemy.equals("South America")) if(planeXSpotsEnemy[i]>=700) showPlanesEnemy[i] = false;//\*

if(contToAttackEnemy.equals("Europe")) if(planeXSpotsEnemy[i]>=350) showPlanesEnemy[i] = false;//\*

if(contToAttackEnemy.equals("Africa")) if(planeXSpotsEnemy[i]>=400) showPlanesEnemy[i] = false;//\*

if(contToAttackEnemy.equals("Australia")) if(planeXSpotsEnemy[i]>=400) showPlanesEnemy[i] = false;//\*

}

boolean checkAll = false;

for(int i=0; i<showPlanesEnemy.length; i++)

{

if(showPlanesEnemy[i]==true) checkAll = true;

}

if(checkAll==false)

{

timerForPlaneEnemy.stop();

sendThePlaneEnemy=false;

firstTimeEnemy = true;

int indexOfContAttack = 0;

double successChance = Math.random();

for(int i=0; i<continents.length; i++)

{

if(continents[i].equals(contToAttackEnemy)) indexOfContAttack=i;

}

if(successChance<0.75-defenseCount[indexOfContAttack]) //\*

{

if(planesToSendEnemy>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttackEnemy+" were successful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttackEnemy+" were successful in their mission.");//\*

healthReduce[indexOfContAttack] -= planesToSendEnemy;

int moneyCost = (int)(Math.random()\*100000+10000);

int peopleDamage = ((int)(Math.random()\*1000+100));

startingMoney[indexOfContAttack] -= moneyCost;

populations[indexOfContAttack] -= peopleDamage;

if(healthReduce[indexOfContAttack]<=-97)

{

//System.out.println("dude");

canUpdateInfo[indexOfContAttack]=true;

makeBlue[indexOfContAttack]=true;

if(startingMoney[indexOfContAttack]>=0)startingMoney[attackingCont] += startingMoney[indexOfContAttack];

if(startingTroops[indexOfContAttack]>=0)startingTroops[attackingCont] += startingTroops[indexOfContAttack];

startingTroops[indexOfContAttack]=0;

startingMoney[indexOfContAttack] = 0;

// System.out.println(startingMoney[indexOfContAttack]);//???

for(int i=0; i<6; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

troopPopulations[i].setText(" "+continents[i]+": "+startingTroops[i]);

}

else {jm3.setText(""+startingMoney[i]);jm5.setText(""+startingTroops[i]);}

}

//revalidate();

repaint();

}

}

else

{

if(planesToSendEnemy>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttackEnemy+" were unsuccessful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttackEnemy+" were unsuccessful in their mission.");//\*

}

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

}

else

{

jm3.setText(""+startingMoney[i]);

jm4.setText(""+populations[i]);

}

}

removeAll();

setLayout(new GridLayout(3,3));

timer.stop();

for(int i=0; i<9; i++)

{

if(i==4) add(panelWithEvent);

else add(new JLabel(""));

}

revalidate();

repaint();

}

}

if(tempContinent.equals("Africa"))

{

planeXSpotsEnemy[0]++;

for(int i=1; i<planeXSpotsEnemy.length; i++)

{

if(planeXSpotsEnemy[i-1]-planeXSpotsEnemy[i]>=50) planeXSpotsEnemy[i]++;

}

for(int i=0; i<planeXSpotsEnemy.length; i++)

{

if(contToAttackEnemy.equals("North America")) if(planeXSpotsEnemy[i]>=500) showPlanesEnemy[i] = false;//\*

if(contToAttackEnemy.equals("South America")) if(planeXSpotsEnemy[i]>=300) showPlanesEnemy[i] = false;//\*

if(contToAttackEnemy.equals("Europe")) if(planeXSpotsEnemy[i]>=325) showPlanesEnemy[i] = false;//\*

if(contToAttackEnemy.equals("Asia")) if(planeXSpotsEnemy[i]>=400) showPlanesEnemy[i] = false;//\*

if(contToAttackEnemy.equals("Australia")) if(planeXSpotsEnemy[i]>=450) showPlanesEnemy[i] = false;//\*

}

boolean checkAll = false;

for(int i=0; i<showPlanesEnemy.length; i++)

{

if(showPlanesEnemy[i]==true) checkAll = true;

}

if(checkAll==false)

{

timerForPlaneEnemy.stop();

sendThePlaneEnemy=false;

firstTimeEnemy = true;

int indexOfContAttack = 0;

double successChance = Math.random();

for(int i=0; i<continents.length; i++)

{

if(continents[i].equals(contToAttackEnemy)) indexOfContAttack=i;

}

if(successChance<0.75-defenseCount[indexOfContAttack]) //\*

{

if(planesToSendEnemy>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttackEnemy+" were successful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttackEnemy+" were successful in their mission.");//\*

healthReduce[indexOfContAttack] -= planesToSendEnemy;

int moneyCost = (int)(Math.random()\*100000+10000);

int peopleDamage = ((int)(Math.random()\*1000+100));

startingMoney[indexOfContAttack] -= moneyCost;

populations[indexOfContAttack] -= peopleDamage;

if(healthReduce[indexOfContAttack]<=-97)

{

//System.out.println("dude");

canUpdateInfo[indexOfContAttack]=true;

makeBlue[indexOfContAttack]=true;

if(startingMoney[indexOfContAttack]>=0)startingMoney[attackingCont] += startingMoney[indexOfContAttack];

if(startingTroops[indexOfContAttack]>=0)startingTroops[attackingCont] += startingTroops[indexOfContAttack];

startingTroops[indexOfContAttack]=0;

startingMoney[indexOfContAttack] = 0;

// System.out.println(startingMoney[indexOfContAttack]);//???

for(int i=0; i<6; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

troopPopulations[i].setText(" "+continents[i]+": "+startingTroops[i]);

}

else {jm3.setText(""+startingMoney[i]);jm5.setText(""+startingTroops[i]);}

}

//revalidate();

repaint();

}

}

else

{

if(planesToSendEnemy>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttackEnemy+" were unsuccessful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttackEnemy+" were unsuccessful in their mission.");//\*

}

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

//continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

}

else

{

jm3.setText(""+startingMoney[i]);

jm4.setText(""+populations[i]);

}

}

removeAll();

setLayout(new GridLayout(3,3));

timer.stop();

for(int i=0; i<9; i++)

{

if(i==4) add(panelWithEvent);

else add(new JLabel(""));

}

revalidate();

repaint();

}

}

if(tempContinent.equals("Australia"))

{

planeXSpotsEnemy[0]++;

for(int i=1; i<planeXSpotsEnemy.length; i++)

{

if(planeXSpotsEnemy[i-1]-planeXSpotsEnemy[i]>=50) planeXSpotsEnemy[i]++;

}

for(int i=0; i<planeXSpotsEnemy.length; i++)

{

if(contToAttackEnemy.equals("North America")) if(planeXSpotsEnemy[i]>=1000) showPlanesEnemy[i] = false;//\*

if(contToAttackEnemy.equals("South America")) if(planeXSpotsEnemy[i]>=800) showPlanesEnemy[i] = false;//\*

if(contToAttackEnemy.equals("Europe")) if(planeXSpotsEnemy[i]>=650) showPlanesEnemy[i] = false;//\*

if(contToAttackEnemy.equals("Asia")) if(planeXSpotsEnemy[i]>=400) showPlanesEnemy[i] = false;//\*

if(contToAttackEnemy.equals("Africa")) if(planeXSpotsEnemy[i]>=600) showPlanesEnemy[i] = false;//\*

}

boolean checkAll = false;

for(int i=0; i<showPlanesEnemy.length; i++)

{

if(showPlanesEnemy[i]==true) checkAll = true;

}

if(checkAll==false)

{

timerForPlaneEnemy.stop();

sendThePlaneEnemy=false;

firstTimeEnemy = true;

int indexOfContAttack = 0;

double successChance = Math.random();

for(int i=0; i<continents.length; i++)

{

if(continents[i].equals(contToAttackEnemy)) indexOfContAttack=i;

}

if(successChance<0.75-defenseCount[indexOfContAttack]) //\*

{

if(planesToSendEnemy>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttackEnemy+" were successful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttackEnemy+" were successful in their mission.");//\*

healthReduce[indexOfContAttack] -= planesToSendEnemy;

int moneyCost = (int)(Math.random()\*100000+10000);

int peopleDamage = ((int)(Math.random()\*1000+100));

startingMoney[indexOfContAttack] -= moneyCost;

populations[indexOfContAttack] -= peopleDamage;

if(healthReduce[indexOfContAttack]<=-97)

{

//System.out.println("dude");

canUpdateInfo[indexOfContAttack]=true;

makeBlue[indexOfContAttack]=true;

if(startingMoney[indexOfContAttack]>=0)startingMoney[attackingCont] += startingMoney[indexOfContAttack];

if(startingTroops[indexOfContAttack]>=0)startingTroops[attackingCont] += startingTroops[indexOfContAttack];

startingTroops[indexOfContAttack]=0;

startingMoney[indexOfContAttack] = 0;

// System.out.println(startingMoney[indexOfContAttack]);//???

for(int i=0; i<6; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

troopPopulations[i].setText(" "+continents[i]+": "+startingTroops[i]);

}

else {jm3.setText(""+startingMoney[i]);jm5.setText(""+startingTroops[i]);}

}

//revalidate();

repaint();

}

}

else

{

if(planesToSendEnemy>1)eventPanel.setText("The troops deployed by jets sent to "+contToAttackEnemy+" were unsuccessful in their mission.");//\*

else eventPanel.setText("The troops deployed by jet sent to "+contToAttackEnemy+" were unsuccessful in their mission.");//\*

}

for(int i=0; i<continentPopulations.length; i++)

{

if(i!=whichCont)

{

startingMoneyArray[i].setText(" "+continents[i]+": "+startingMoney[i]);

continentPopulations[i].setText(" "+continents[i]+": "+populations[i]);

}

else

{

jm3.setText(""+startingMoney[i]);

jm4.setText(""+populations[i]);

}

}

removeAll();

setLayout(new GridLayout(3,3));

timer.stop();

for(int i=0; i<9; i++)

{

if(i==4) add(panelWithEvent);

else add(new JLabel(""));

}

revalidate();

repaint();

}

}

repaint();

if(healthReduce[whichCont]<=-97) {endGame=true;gameOverCount++;healthGO=true;}

if(startingMoney[whichCont]<=0) {endGame=true;moneyGO=true;gameOverCount++;}

if(populations[whichCont]<=0) {endGame=true;gameOverCount++;peopleGO=true;}

if(startingTroops[whichCont]<=0) {endGame=true;gameOverCount++;troopGO=true;}

boolean testWin = true;

for(int i=0; i<makeBlue.length; i++)

{

if(i!=whichCont)

{

if(makeBlue[i]==false) testWin=false;

}

}

if(testWin) won=true;

}

}

}

}