CS 106 project report.

***MAIN MENU (FlagMain.java):***

**Variables:**

* No variables.

**Constructor:**

* frameA (PartA)
* frameB (PartB)
* Wquit (WindowQuitter)
* modeOne (JButton)
* modeTwo (JButton)
* Panel1 (JPanel)
* Panel2 (JPanel)
* Panel3 (JPanel)
* infoText (JTextField)

**Methods:**

* ActionPerformed (void)
* Main

***PART A(PartA.java):***

**Variables:**

* streakCounter (int)
* scoreCounter (int)
* livesCounter(int)
* round (int)
* randNum (int)
* Countries (string[])
* flagCheck (boolean[])
* flagIcons (imageIcon[])
* flagLabels (JLabel[])
* scoreBorder (TitledBorder)
* streakBorder (TitledBorder)
* livesBorder (TitledBorder)
* resultText (JTextField)
* streakText (JTextField)
* livesText (JTextField)
* scoreText (JTextField)
* Panel1 (JPanel)
* Panel2 (JPanel)
* Panel3 (JPanel)
* Panel4 (JPanel)
* Panel5 (JPanel)
* dataBox (JComboBox)
* submitButton (JButton)

**Constructor:**

* PartA()

**Methods:**

* ActionPerformed
* flagPick

***PartB(PartB.java):***

**Variables:**

* streakCounter (int)
* scoreCounter (int)
* livesCounter(int)
* round (int)
* randFlagNum (int)
* randCountryNum (int)
* Countries (string[])
* flagCheck (boolean[])
* countryCheck (boolean[])
* repeatCheck (boolean)
* effects (Sounds)
* flagIcons (imageIcon[])
* flagLabels (JLabel[])
* flagPanels (JPanel[])
* flagButtons (JButton[])
* flagButtonGroup (ButtonGroup)
* scoreBorder (TitledBorder)
* streakBorder (TitledBorder)
* livesBorder (TitledBorder)
* resultText (JTextField)
* streakText (JTextField)
* livesText (JTextField)
* scoreText (JTextField)
* countriesText(JTextField)
* Panel1 (JPanel)
* Panel2 (JPanel)
* Panel3 (JPanel)
* Panel4 (JPanel)
* Panel5 (JPanel)

**Constructor:**

* PartB()

**Methods:**

* ActionPerformed (void)
* flagPick (void)
* countryPick (void)

***Sounds(Sounds.java)***

**Variables:**

* No Variables

**Constructor:**

* Default Constructor

**Methods:**

* correctEffect (void)
* incorrectEffect (void)
* MainSongEffect (void)
* ExtraLifeEffect (void)
* VictoryEffect (void)
* DefeatEffect (void)

***WindowQuitter (WindowQuitter.java)***

**Variables:**

* No Variables

**Constructor:**

* Default Constructor

**Methods:**

* windowClosing (void)

**FUNCTIONALITY**

***MAIN MENU (FlagMain.java):***

**Constructor:**

* Create frames A and B but keep them invisible
* Create a WindowQuitter object
* Create modeOne and modeTwo JButtons, add actionListeners for both.
* Create JPanels 1,2 and 3
* Set the title
* Create infoText JTextField, set its text and the horizontal alignment, make it non editable
* Add infoText to panel1, give a bevel effect
* Set panel2 layout to BorderLayout, add the two buttons west and east respectively
* Set panel3 layout to BorderLayout, add the two other panels north and south respectively
* Set the layout of the content pane to flowLayout, add panel3
* Set the dimensions of frame A and B, make them un-resizable, add the wquit WindowListener

**Action performed:**

* If modeOne button is pressed, make frameA visible and dispose frameB
* If modeTwo button is pressed, make frameB visible and dispose frameA

**Main:**

* Create frameSwitch frame
* Set dimensions
* Add wquit windowlistener
* Make it un-resizable
* Make it visible

***PartA(PartA.java):***

**Constructor:**

* Make sure that the order of the images is the order of the entries of the countries String Array
* Create 20 ImageIcons and fill them with the pictures
* Create 20 JLabels and fill them with the ImageIcons
* Create 3 titled borders
* Create 4 counters and initialise them
* Create the TextFields make them align to the center of the box and add the titled borders
* Give a bevel effect to resultText
* Create 5 panels
* Create a databox and fill it with the entries of the countries array
* Create Submit button and give it an ActionLIstener
* Set the layout of panel 2 to BorderLayout, give it a bevel effect and add the dataBox and the submit button, center and north respectively.
* Set the layout of panel 3 to flowLayout and add resultText
* Set the Layout of panel 4 to flowLayout and add the score, streak and lives textFields
* Set the layout of panel 5 to BorderLayout and fill it using the flagPick method
* Set the layout of panel 1 to BorderLayout, give it a bevel effect and add panel5 to the center, panel 3 south and panel 4 north.
* Set the layout of the content pane to BorderLayout and add panels 1 and 2, center and south respectively.

**Action performed:**

* When the submit button is pressed.
* Increase the round count
* Compare the text on the selected cell of the databox to the entry of the displayed flag in the countries array to see if the user is correct
* If he is correct the text changes to CORRECT and a sound plays. Also the selected item is removed from the combobox and the score goes up.
* If the user is wrong and has lives left, the text changes to wrong, a sound plays, streak is reset and a life is lost.
* if the user has no more lives, the game ends, the text changes to GAME OVER and the submit answer button becomes unusable. Also a sound plays.
* If the user reaches a score of 20 he has won the game. Text changes to CONGRATULATIONS and the victory sound plays. The submit button is unusable and the combobox only has one item that says YOU WIN!
* If the user answers thrice correctly in a row he gains a life. The text changes to EXTRA LIFE! and a sound plays.

**flagPick:**

* Creates a random number, relates it to an entry on the flagLabels array and ,after checking through a boolean array if the flag has

been displayed before.

* Adds the flag in the main panel.
* The visibility of the panel changes in order to allow for the change of the flags without any bugs.

***PartB(PartB.java):***

**Constructor:**

* Make sure that the order of the images is the order of the entries of the countries String Array
* FlagIcons is filled with the image files.
* FlagLabels is filled with FlagIcons, and given gray line borders.
* FlagButtons are created, given numeric action commands and added to flagButtonGroup.
* FlagPanels are created, given layouts and filled with all other objects.
* Create 3 titled borders
* Create 4 counters and initialise them
* Create the TextFields make them align to the center of the box and add the titled borders
* Give a bevel effect to resultText
* The text in countriesText is randomized by the CountryPick() method at the start of every game.
* Create 5 panels
* Create Submit button and give it an ActionLIstener
* Set the layout of panel 2 to BorderLayout, give it a bevel effect and add the countriesText and the submit button, center and east respectively.
* Set the layout of panel 3 to flowLayout and add resultText
* Set the Layout of panel 4 to flowLayout and add the score, streak and lives textFields
* Set the layout of panel 5 to BorderLayout and fill it using the flagPick method
* Set the layout of panel 1 to BorderLayout, give it a bevel effect and add panel5 to the center, panel 3 south and panel 4 north.
* Set the layout of the content pane to BorderLayout and add panels 1 and 2, center and south respectively.

**Action performed:**

* When the submit button is pressed.
* Increase the round count
* Compare the nuber of the selected radiobutton to the entries of the countries list to see if the user is correct
* If the user is correct the text changes to CORRECT!, a sound plays, the score and streak numbers go up, the flags are updated
* If the user has a score of 20 the flags are not updated
* If the user is wrong and has lives left, the text changes to wrong, a sound plays, streak is reset and a life is lost.
* if the user has no more lives, the game ends, the text changes to GAME OVER and the submit answer button becomes unusable. Also a sound plays.
* If the user reaches a score of 20 he has won the game. Text changes to CONGRATULATIONS and the victory sound plays. The submit button is unusable and the countriesText says YOU WIN!
* If the user answers thrice correctly in a row he gains a life. The text changes to EXTRA LIFE! and a sound plays.

**flagPick:**

Used to randomly pick and display flags. The correct flag is represented by randCountryNum, meaning the random number dictating the requested country.

There are two errors with this method. First is that it won't display the same amount of flags every round. Tried to circumvent this problem by making the program loop one extra time for every repeat flag it skips. This means that a flag will only be displayed if it is not already showing (.isShowing()), and that if it is showing it will decrease i, therefore extending the loop number. It will still display a random number of flags nevertheless.

Second problem lies within the display of the needed flag to proceed. In order for it to not be in the same spot every round it has been "tied" to the flagNum, a random number that changes with every loop. The needed flag is to be displayed the first time the condition is met and then never again through a boolean switch. As a contingency plan in the case that after all the loops the condition isn't met, an if statement was added that checked if the flag was not showing and if so it would display it in the final position. This caused the program to always display the needed flag in the final position no matter what. Afterwards, a statement was added inside the loop that aimed to increase the loop counter if the correct flag was not showing. This, for some reason, caused the menu frame not to appear. In the end i set the loop counter high enough to guarantee the display of the needed flag while limiting the amount of shown flags with an if statement.

**countryPick:**

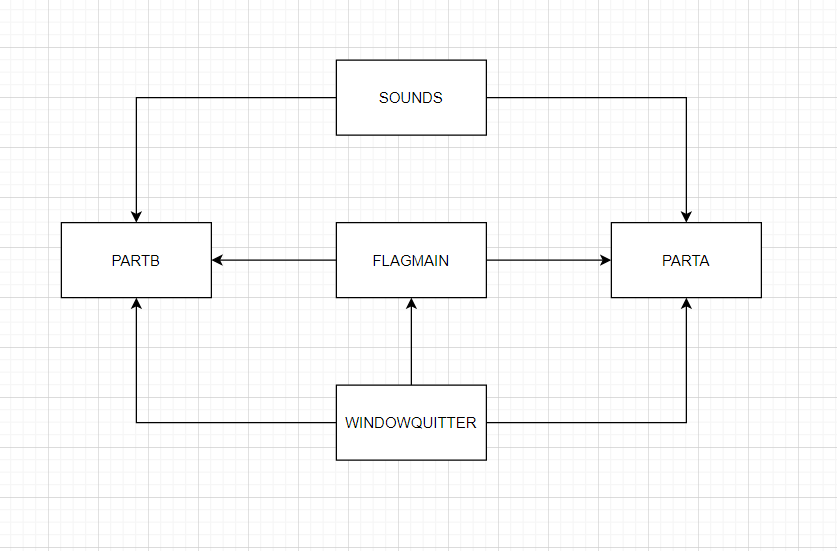
Picks a country to ask the user and makes sure it doesn't repeat.

***Sounds(Sounds.java):***

**correctEffect:**

This method is repeated for all sound effects. First a file object is created and correlated to the path of the sound file. This allows for the AudioImputStream to recognise the sound files. Afterwards a clip object is created to allow more control over the sound bit. The sound is imported to the clip and then "played". Trapping code is implemented for the cases of the system being unable to find the file or the path.

**Diagram:**

****