package datastructures;

import java.awt.Component;

import java.io.File;

import java.io.FileInputStream;

import java.io.IOException;

import java.io.ObjectInputStream;

import gui.ATOGeneratorFrame;

import structures.Asset;

import swing.SingletonHolder;

import utilities.DebugUtility;

import utilities.FileImporter;

/\*\*

\* An Importer for ATO .PROJ files

\*/

public class ATOProjectImporter implements FileImporter {

@Override

public void doImport(File f) {

if (f != null) {

try {

System.err.println(f.getPath());

FileInputStream is = new FileInputStream(f.getAbsolutePath());

ObjectInputStream ois = new ObjectInputStream(is);

SingletonHolder.setInstanceOf(ATOData.class, ois.readObject());

System.err.println("ATO DATA LOADED");

ATOData data = (ATOData) SingletonHolder.getInstanceOf(ATOData.class);

for (Asset d : data) {

System.out.println(d);

}

((Component) SingletonHolder.getInstanceOf(ATOGeneratorFrame.class)).repaint();

((Component) SingletonHolder.getInstanceOf(ATOGeneratorFrame.class)).validate();

ois.close();

is.close();

String message = "Project loaded: \n" + "-- " + f.getAbsolutePath();

DebugUtility.debug(ATOData.class, message);

} catch (IOException e) {

DebugUtility.error(ATOData.class, "Error loading " + f.getName(), e);

} catch (ClassNotFoundException e) {

DebugUtility.error(ATOData.class, "Unable to load ATOData from " + f.getName());

}

}

}

}

package main;

import javax.swing.JFrame;

import gui.ATOGeneratorFrame;

import swing.SingletonHolder;

import utilities.Configuration;

import utilities.DebugUtility;

/\*\*

\* The main starting point of the ATO Generator.

\*/

public final class ATOGenerator {

private ATOGenerator() {

}

/\*\*

\* Starting point of ATO maker

\*

\* @param args

\*/

public static void main(String[] args) {

Configuration.setLookAndFeel(ATOGenerator.class);

((JFrame) SingletonHolder.getInstanceOf(ATOGeneratorFrame.class)).setVisible(true);

DebugUtility.debug(ATOGenerator.class, "Startup successful.");

}

}

package table;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import javax.swing.JButton;

import datastructures.ATOAsset;

import datastructures.ATOData;

import structures.ATOAssets;

import structures.ListOfAsset;

import swing.SingletonHolder;

import utilities.ADTTableModel;

/\*\*

\* ATO table model stores the data for ATO generator table

\*/

public class ATOTableModel extends ADTTableModel {

private static final long serialVersionUID = -9017185692217463087L;

private MouseListener listener = new MouseListener() {

@Override

public void mouseClicked(MouseEvent e) {

System.out.println("AR BUTTON PRESSED");

}

@Override

public void mousePressed(MouseEvent e) {

// nothing

}

@Override

public void mouseReleased(MouseEvent e) {

// nothing

}

@Override

public void mouseEntered(MouseEvent e) {

// nothing

}

@Override

public void mouseExited(MouseEvent e) {

// nothing

}

};

@Override

public void create() {

this.columnNames.add("MSN#");

this.columnNames.add("AMCMSN");

this.columnNames.add("PKGID");

this.columnNames.add("MSNCC");

this.columnNames.add("MSN");

this.columnNames.add("SECMSN");

this.columnNames.add("ALERT");

this.columnNames.add("DEPLOC");

this.columnNames.add("ARRLOC");

this.columnNames.add("#AC");

this.columnNames.add("ACTYPE");

this.columnNames.add("CALLSIGN");

this.columnNames.add("CONFIG");

this.columnNames.add("SECCONFIG");

this.columnNames.add("M1");

this.columnNames.add("M2");

this.columnNames.add("M3");

this.columnNames.add("AR");

this.items = (ListOfAsset) (SingletonHolder.getInstanceOf(ATOAssets.class));

}

@Override

public Class<?> getColumnClass(int col) {

if (col == 17) {

return JButton.class;

}

return super.getColumnClass(col);

}

@Override

public Object getValueAt(int rowIndex, int columnIndex) {

if (columnIndex == 17) {

JButton button = new JButton("PRESS");

button.addMouseListener(this.listener);

return button;

}

return ((ATOAsset) ((ATOData) SingletonHolder.getInstanceOf(ATOData.class)).get(rowIndex)).getItems()

.get(columnIndex);

}

@Override

public void setValueAt(Object aValue, int rowIndex, int columnIndex) {

ATOAsset chosen = (ATOAsset) ((ATOData) SingletonHolder.getInstanceOf(ATOData.class)).get(rowIndex);

chosen.getItems().set(columnIndex, aValue);

ATOData.checkAddNew();

}

@Override

public void addNew() {

ATOData.addNew();

}

@Override

public int getRowCount() {

return ((ATOData) SingletonHolder.getInstanceOf(ATOData.class)).size();

}

}

package table;

import java.awt.Component;

import java.awt.Dimension;

import java.awt.Font;

import javax.swing.JButton;

import javax.swing.JTable;

import javax.swing.table.DefaultTableCellRenderer;

import javax.swing.table.TableModel;

import swing.Borders;

import swing.SingletonHolder;

import utilities.Fonts;

/\*\*

\* A rundown JTable to display rundown information

\*/

public class ATOTable extends JTable {

private static final long serialVersionUID = 1267395829859061144L;

private static ATOTable instance = new ATOTable();

/\*\*

\* the previously selected row

\*/

int prevRow;

/\*\*

\* the previously selected column

\*/

int prevCol;

/\*\*

\* Singleton implementation

\*

\* @return - single instance

\*/

public static ATOTable getInstance() {

return instance;

}

/\*\*

\* @param col

\*/

public static void setPrevCol(int col) {

instance.prevCol = col;

}

/\*\*

\* @param row

\*/

public static void setPrevRow(int row) {

instance.prevRow = row;

}

/\*\*

\* @return previously selected column

\*/

public static int getPrevCol() {

return instance.prevCol;

}

/\*\*

\* @return previously selected row

\*/

public static int getPrevRow() {

return instance.prevRow;

}

private ATOTable() {

// set the data

this.setModel((TableModel) SingletonHolder.getInstanceOf(ATOTableModel.class));

this.setDefaultRenderer(JButton.class, new DefaultTableCellRenderer() {

private static final long serialVersionUID = 1L;

@Override

public Component getTableCellRendererComponent(JTable table, Object value, boolean isSelected,

boolean hasFocus, int row, int column) {

return (Component) value;

}

});

this.setDefaultRenderer(String.class, new DefaultTableCellRenderer() {

/\*\*

\* Serialization info

\*/

private static final long serialVersionUID = 15484447;

@Override

public Component getTableCellRendererComponent(JTable table, Object value, boolean isSelected,

boolean hasFocus, int row, int column) {

Component c = super.getTableCellRendererComponent(table, value, isSelected, hasFocus, row, column);

if (!isSelected) {

c.setBackground(getBackground());

}

return c;

}

});

// font

this.getTableHeader().setFont(Fonts.serifBold);

this.getTableHeader().setBorder(Borders.BLACK);

// sizing

this.setFillsViewportHeight(true);

this.setBounds(10, 200, 500, 250);

// reordering and column selection not allowed

this.setColumnSelectionAllowed(false);

this.getTableHeader().setReorderingAllowed(false);

// this.setFont(Fonts.serif);

this.setFont(new Font("Verdana", 1, 20));

// column header click events

this.setSurrendersFocusOnKeystroke(true);

this.setFocusTraversalKeysEnabled(false);

this.setRowHeight(40 + 10);

this.setPreferredScrollableViewportSize(new Dimension(523, 233));

ATOTableCellEditor editor = new ATOTableCellEditor();

for (int x = 0; x < getModel().getColumnCount() - 1; x++) {

getColumnModel().getColumn(x).setCellEditor(editor);

}

}

}

package gui;

import swing.BaseFrame;

/\*\*

\* A frame to add a player's AR information

\*/

public class ARInfoFrame extends BaseFrame {

/\*\* Serialization information \*/

private static final long serialVersionUID = -6283637369374037729L;

@Override

public void create() {

this.setSize(400, 400);

}

}

package gui;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.border.EmptyBorder;

import javax.swing.filechooser.FileNameExtensionFilter;

import datastructures.ATOData;

import datastructures.ATOProjectImporter;

import swing.ADTLabel;

import swing.ActionButton;

import swing.BasePanel;

import swing.SingletonHolder;

import utilities.Configuration;

import utilities.FileChooser;

/\*\*

\* The panel containing all of the ATO functional buttons.

\*/

public class ATOButtonPanel extends BasePanel {

private static final long serialVersionUID = 6980336047696920906L;

/\*\* button to load ATO project \*/

public ActionButton loadBtn;

/\*\* button to generate the ATO from current data \*/

public ActionButton genBtn;

/\*\* button to load ATO project \*/

public ActionButton saveBtn;

/\*\*

\* ActionListener to be added to the buttons

\*/

public ActionListener atoButtonListener;

/\*\*

\* Listen to button events on the ATOButtonPanel

\*/

class ATOButtonListener implements ActionListener {

/\*\* save the panel for button sources \*/

ATOButtonPanel pnl;

/\*\*

\* Constructor

\*

\* @param panel the panel to listen to

\*/

ATOButtonListener(ATOButtonPanel panel) {

this.pnl = panel;

}

@Override

public void actionPerformed(ActionEvent e) {

if (e.getSource().equals(this.pnl.saveBtn)) {

((ATOData) SingletonHolder.getInstanceOf(ATOData.class)).save();

} else if (e.getSource().equals(this.pnl.loadBtn)) {

FileChooser.selectAndLoadFile("Select an ATO Proj", new FileNameExtensionFilter("ATO Proj", "proj"),

Configuration.getInstance().getATOProjFileLoc(), new ATOProjectImporter());

((ATOGeneratorFrame) SingletonHolder.getInstanceOf(ATOGeneratorFrame.class)).repaint();

} else if (e.getSource().equals(this.pnl.genBtn)) {

((ATOData) SingletonHolder.getInstanceOf(ATOData.class)).output();

}

}

}

@Override

public void create() {

this.atoButtonListener = new ATOButtonListener(this);

setLayout(new GridLayout(1, 5, 20, 20));

setBorder(new EmptyBorder(20, 20, 20, 20));

this.saveBtn = new ActionButton("Save", this.atoButtonListener);

this.loadBtn = new ActionButton("Load", this.atoButtonListener);

this.genBtn = new ActionButton("Generate", this.atoButtonListener);

add(new ADTLabel());

add(this.saveBtn);

add(this.loadBtn);

add(this.genBtn);

add(new ADTLabel());

}

}

package gui;

import java.awt.Component;

import swing.BaseFrame;

import swing.SingletonHolder;

import utilities.DebugUtility;

import utilities.ImageLibrary;

/\*\*

\* A class to generate valid, formatted ATOs in USMTF00.txt format.

\*

\*/

public class ATOGeneratorFrame extends BaseFrame {

private static final long serialVersionUID = 6718631382663500003L;

@Override

public void create() {

this.setSize(1600, 600);

this.setLocationRelativeTo(null);

this.setTitle("ATO Generator");

this.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

DebugUtility.trace(ATOGeneratorFrame.class, "Frame loaded");

this.setIconImage(ImageLibrary.getImage("./AF-Roundel"));

DebugUtility.trace(ATOGeneratorFrame.class, "IconImage loaded/set");

this.add((Component) SingletonHolder.getInstanceOf(ATOPanel.class));

DebugUtility.trace(ATOGeneratorFrame.class, "MainPanel added");

}

}

package gui;

import java.awt.BorderLayout;

import java.awt.Component;

import javax.swing.JScrollPane;

import javax.swing.border.EmptyBorder;

import swing.BasePanel;

import swing.SingletonHolder;

import table.ATOTable;

/\*\*

\* The main panel for ATO table and information

\*/

public class ATOPanel extends BasePanel {

private static final long serialVersionUID = -6930365855924400678L;

@Override

public void create() {

// establish the main content pane (layout the way we want)

this.setBorder(new EmptyBorder(20, 20, 20, 20));

this.setLayout(new BorderLayout());

// Create the scroll pane and add the table to it.

JScrollPane scrollPane = new JScrollPane(ATOTable.getInstance());

scrollPane.setBounds(12, 205, 523, 226);

// Add the scroll pane and button panel to the content.

this.add(scrollPane, BorderLayout.CENTER);

this.add((Component) SingletonHolder.getInstanceOf(ATOButtonPanel.class), BorderLayout.NORTH);

}

}