Rhythminator Part 2

CSE1102 Project 07, Spring 2016

Bryan Arnold

4/24/16

TA: Zigeng Wang

Section: 51

Instructor: Jeffrey A. Meunier

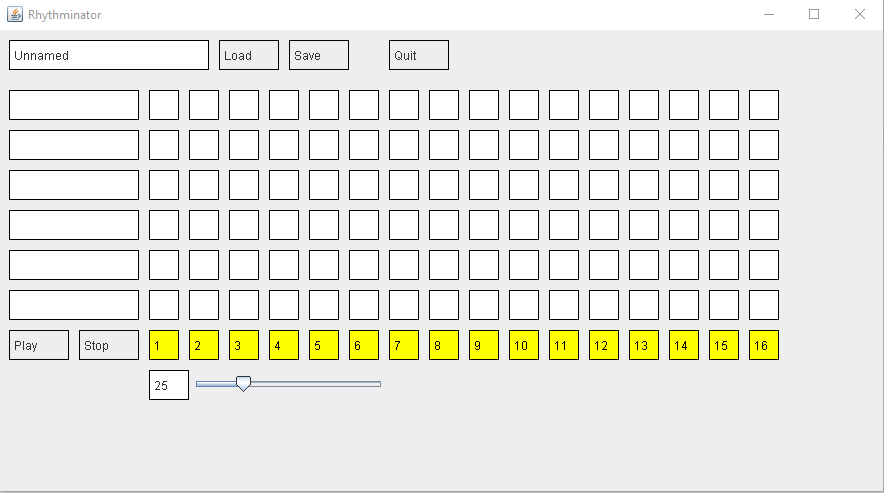
**Introduction:**

In this assignment you will write a Java program to display the graphical components of the Rhythminator project. You will start with a simple set of shapes that can be drawn on the screen, and gradually develop the GUI (Graphical User Interface) elements shown here. In this project you will only develop the view, which consists of the graphical elements. You will not attach it to the model, which is what you created in part 1. You will not even use the model in this project. This program will not play sounds. That will happen in the final project.

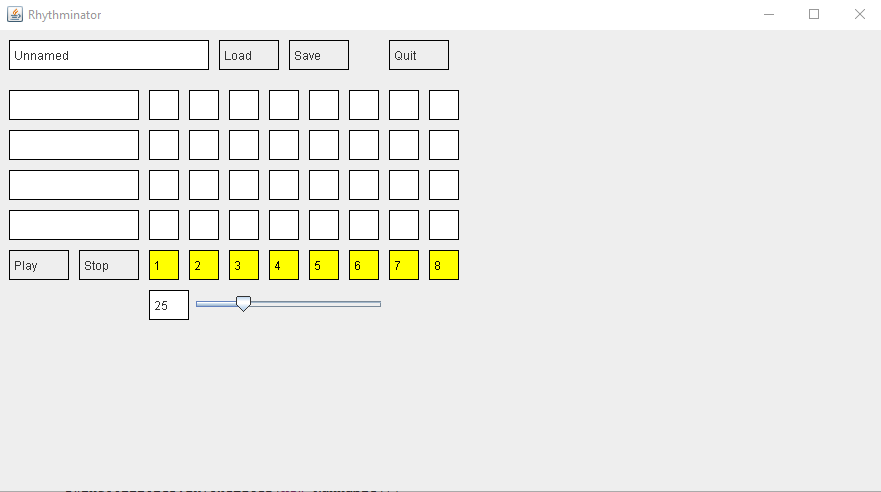
**Output:**

The output consists of a GUI displaying the tracks, beats, and options. Most of them do not actually do what they are intended for, which they will be later on, but they are all functional. Here are screenshots of track and beat variations:

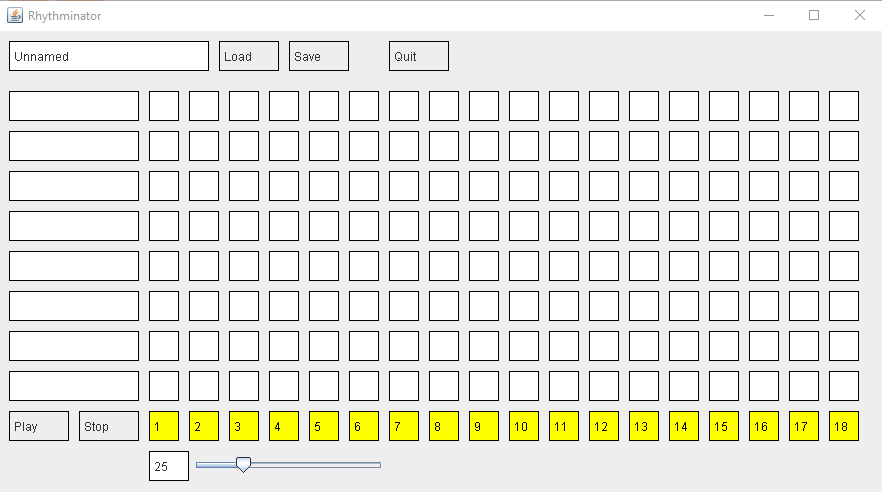
**When there are 6 tracks and 16 beats:**

****

**When there are 4 tracks and 8 beats:**

****

**When there are 8 tracks and 18 beats:**

****

**Lastly, when there are 5 tracks and 3 beats:**

****

There is no other output as of now, but the screen is interactive with the functions they will serve later on.

NOTE: I could not make the window color white for whatever reason, but it's not a huge issue.

**Source Code:**

Note: Sound files included, wasn’t sure whether or include them in the submission.

**Default Package:**

**Rhythminator Class:**

**import** javax.swing.SwingUtilities;

**import** view.View;

**import** controller.Controller;

**import** model.Sound;

//Rhythminator Part 2

//CSE1102 Project 07, Spring 2016

//Bryan Arnold

//4/24/16

//TA: Zigeng Wang

//Section: 51

//Instructor: Jeffrey A. Meunier

**public** **class** Rhythminator {

**private** **static** **final** **int** ***NUM\_TRACKS*** = 6;

**private** **static** **final** **int** ***NUM\_BEATS*** = 16;

// I put all the interesting stuff in a separate method here so that you don't

// have to keep looking at the Swing threading stuff in the main method below.

**private** **static** **void** \_main(){

Sound.*scanSoundDir*();

Controller controller = **new** Controller();

@SuppressWarnings("unused")

View view = **new** View(controller, ***NUM\_TRACKS***, ***NUM\_BEATS***);

// we don't use the view variable yet, but we will

}

**public** **static** **void** main(String[] args){

// In order to use Swing graphics effectively, the program should be started

// running in a concurrent thread controlled by Swing. This is how you do it:

SwingUtilities.*invokeLater*(**new** Runnable(){

// the main part of a program that uses Swing graphics should appear inside this run method

@Override

**public** **void** run() {

*\_main*();

}

});

}

}

**Controller Package:**

**Controller Class:**

**package** controller;

**import** gui.Button;

**import** view.NoteSquare;

//Rhythminator Part 2

//CSE1102 Project 07, Spring 2016

//Bryan Arnold

//4/24/16

//TA: Zigeng Wang

//Section: 51

//Instructor: Jeffrey A. Meunier

**public** **class** Controller{

**public** Controller(){}

**public** **void** keyPressed(**int** keyCode){

//System.out.println("Controller.keyPressed " + keyCode);

}

**public** **void** keyReleased(**int** keyCode){

//System.out.println("Controller.keyReleased " + keyCode);

}

**public** **void** keyTyped(**char** keyChar){

//System.out.println("Controller.keyTyped '" + keyChar + "'");

}

**public** **void** soundNameSelected(**int** trackNumber, String soundName){

System.***out***.println("Sound " + soundName + " selected for track " + trackNumber);

}

**public** **void** buttonPressed(Button button) {

button.toString();

System.***out***.println("Button pressed");

}

**public** **void** sliderChange(String name, **int** value) {

System.***out***.println("Slider " + name + " changed to " + value);

}

**public** **void** noteSquareClicked(NoteSquare square){

System.***out***.println("NoteSquare clicked: " + square);

}

}

**Gui Package:**

**Box Class:**

**package** gui;

**import** java.awt.Color;

**import** java.awt.Graphics;

**import** java.awt.event.MouseEvent;

**import** java.awt.event.MouseListener;

**import** javax.swing.BorderFactory;

**import** javax.swing.JPanel;

//Rhythminator Part 2

//CSE1102 Project 07, Spring 2016

//Bryan Arnold

//4/24/16

//TA: Zigeng Wang

//Section: 51

//Instructor: Jeffrey A. Meunier

**public** **class** Box **extends** JPanel **implements** MouseListener {

**private** String \_text = **null**;

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** Box(){

**this**(**null**);

}

**public** Box(String text){

**this**.setBorder(BorderFactory.*createLineBorder*(Color.***black***));

**this**.addMouseListener(**this**);

\_text = text;

}

**public** String getText(){

**return** \_text;

}

@Override

**public** **void** paintComponent(Graphics g){

**super**.paintComponent(g);

**if**(\_text != **null**)

g.drawString(\_text, 5, 20);

}

@Override

**public** **void** mouseClicked(MouseEvent mev){}

@Override

**public** **void** mouseEntered(MouseEvent mev){}

@Override

**public** **void** mouseExited(MouseEvent mev){}

@Override

**public** **void** mousePressed(MouseEvent mev){}

@Override

**public** **void** mouseReleased(MouseEvent mev){}

**public** **void** setText(String text){

\_text = text;

**this**.repaint();

}

@Override

**public** String toString(){

**return** "Box(" + \_text + ")";

}

}

**Button Class:**

**package** gui;

**import** java.awt.event.MouseEvent;

**import** controller.Controller;

//Rhythminator Part 2

//CSE1102 Project 07, Spring 2016

//Bryan Arnold

//4/24/16

//TA: Zigeng Wang

//Section: 51

//Instructor: Jeffrey A. Meunier

**public** **class** Button **extends** Box {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** Controller \_controller;

**public** Button(Controller controller, String text){

**super**(text);

**this**.\_controller = controller;

}

@Override

**public** String toString(){

**return** "Button(" + getText() + ")";

}

@Override

**public** **void** mousePressed(MouseEvent mev){

**this**.\_controller.buttonPressed(**this**);

}

}

**Colors Class:**

**package** gui;

**import** java.awt.Color;

//Rhythminator Part 2

//CSE1102 Project 07, Spring 2016

//Bryan Arnold

//4/24/16

//TA: Zigeng Wang

//Section: 51

//Instructor: Jeffrey A. Meunier

/\*\*

\* These are color definitions for the GUI elements.

\* **@author** jeff

\*/

**public** **class** Colors{

**public** **static** **final** Color ***BUTTON\_FG*** = Color.***BLACK***;

**public** **static** **final** Color ***BUTTON\_BG*** = **new** Color(240, 240, 240);

**public** **static** **final** Color ***WINDOW\_BG*** = Color.***WHITE***;

}

**Dialog Class:**

**package** gui;

**import** javax.swing.JDialog;

**import** javax.swing.JOptionPane;

//Rhythminator Part 2

//CSE1102 Project 07, Spring 2016

//Bryan Arnold

//4/24/16

//TA: Zigeng Wang

//Section: 51

//Instructor: Jeffrey A. Meunier

**public** **class** Dialog{

**public** **static** **boolean** askYesNo(String title, String question){

**final** String YES = "Yes";

**final** String NO = "No";

JOptionPane pane = **new** JOptionPane(question);

Object[] options = **new** String[] {NO, YES};

pane.setOptions(options);

JDialog dialog = pane.createDialog(**null**, title);

dialog.setVisible(**true**);

Object obj = pane.getValue();

**if**(obj != **null** && obj.toString().equals(YES))

**return** **true**;

**return** **false**;

}

**public** **static** String getString(String title, String prompt, String initialValue){

Object value = JOptionPane.*showInputDialog*(**null**, prompt, title, JOptionPane.***PLAIN\_MESSAGE***, **null**, **null**, initialValue);

**if**(value != **null**)

**return** value.toString();

**return** **null**;

}

**public** **static** String selectOption(String title, String prompt, String[] options){

Object value = JOptionPane.*showInputDialog*(**null**, prompt, title, JOptionPane.***PLAIN\_MESSAGE***, **null**, options, **null**);

**if**(value != **null**)

**return** value.toString();

**return** **null**;

}

}

**NumberSlider Class:**

**package** gui;

**import** javax.swing.JPanel;

**import** javax.swing.JSlider;

**import** javax.swing.event.ChangeEvent;

**import** javax.swing.event.ChangeListener;

**import** controller.Controller;

//Rhythminator Part 2

//CSE1102 Project 07, Spring 2016

//Bryan Arnold

//4/24/16

//TA: Zigeng Wang

//Section: 51

//Instructor: Jeffrey A. Meunier

**public** **class** NumberSlider **extends** JPanel **implements** ChangeListener {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** Controller \_controller;

**private** String \_name;

**private** Box \_box;

**private** JSlider \_slider;

**public** NumberSlider(Controller controller, String name, **int** min, **int** max, **int** value){

**this**.setLayout(**null**);

**this**.\_name = name;

**this**.\_controller = controller;

Box box = **new** Box();

**this**.\_box = box;

box.setLocation(0, 0);

box.setSize(40, 30);

box.setText(Integer.*toString*(value));

box.setBackground(Colors.***WINDOW\_BG***);

add(box);

JSlider slider = **new** JSlider(JSlider.***HORIZONTAL***, min, max, value);

**this**.\_slider = slider;

slider.setLocation(box.getWidth(), 0);

slider.setSize(200, 30);

add(slider);

setSize(box.getWidth() + slider.getWidth(), 30);

slider.addChangeListener(**this**);

slider.setFocusable(**false**);

}

@Override

**public** **void** stateChanged(ChangeEvent arg0) {

\_box.setText(Integer.*toString*(\_slider.getValue()));

\_controller.sliderChange(\_name, \_slider.getValue());

}

}

**TextBox Class:**

**package** gui;

**import** java.awt.event.MouseEvent;

//Rhythminator Part 2

//CSE1102 Project 07, Spring 2016

//Bryan Arnold

//4/24/16

//TA: Zigeng Wang

//Section: 51

//Instructor: Jeffrey A. Meunier

**public** **class** TextBox **extends** Box{

**private** String \_dialogTitle;

**private** String \_dialogPrompt;

**private** **static** **final** **long** ***serialVersionUID*** = 1;

**public** TextBox(){

**this**("Click to enter text");

}

**public** TextBox(String text){

**this**(text, "Enter new text");

}

**public** TextBox(String text, String dialogPrompt){

**this**(text, dialogPrompt, "Question");

}

**public** TextBox(String text, String dialogPrompt, String dialogTitle){

**super**(text);

\_dialogPrompt = dialogPrompt;

\_dialogTitle = dialogTitle;

}

@Override

**public** **void** mousePressed(MouseEvent mev){

String initialValue = **this**.getText();

String text = Dialog.*getString*(\_dialogTitle, \_dialogPrompt, initialValue);

**if**(text != **null**)

setText(text);

}

}

**Window Class:**

**package** gui;

**import** java.awt.event.KeyEvent;

**import** java.awt.event.KeyListener;

**import** java.awt.event.WindowEvent;

**import** java.awt.event.WindowListener;

**import** javax.swing.JFrame;

//Rhythminator Part 2

//CSE1102 Project 07, Spring 2016

//Bryan Arnold

//4/24/16

//TA: Zigeng Wang

//Section: 51

//Instructor: Jeffrey A. Meunier

**public** **class** Window **extends** JFrame **implements** KeyListener, WindowListener{

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** Window(String name){

**super**(name);

**this**.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

**this**.setLayout(**null**); // layout = null means we want to place things on the screen using absolute coordinates

**this**.addWindowListener(**this**); // set this window to be its own window event listener

**this**.addKeyListener(**this**); // set this window to be its own key listener

**this**.setFocusable(**true**); // this is needed to allow key listener to capture events on this frame

**this**.setSize(400, 400); // subclasses will probably set their own size

**this**.setBackground(Colors.***WINDOW\_BG***); // sets the window's background color

}

/\*\*

\* Call this method to make the application window full-screen.

\* On a Mac this hides the title bar.

\*/

**public** **void** fullscreen(){

**this**.maximize();

**this**.setUndecorated(**true**);

}

// The key event and window event methods are not needed, but I put them in here so that you

// can see how you'd include them in this class.

@Override

**public** **void** keyPressed(KeyEvent kev){}

@Override

**public** **void** keyReleased(KeyEvent kev){}

@Override

**public** **void** keyTyped(KeyEvent kev){}

/\*\*

\* Call this method to maximize the application window.

\* On a Mac it's different from full-screen.

\*/

**public** **void** maximize(){

**this**.setSize(-1, -1);

}

@Override

**public** **void** windowActivated(**final** WindowEvent wev){}

@Override

**public** **void** windowClosed(**final** WindowEvent wev){}

@Override

**public** **void** windowClosing(**final** WindowEvent wev){

//System.exit(0); // not needed if JFrame.EXIT\_ON\_CLOSE is used in the constructor

}

@Override

**public** **void** windowDeactivated(**final** WindowEvent wev){}

@Override

**public** **void** windowDeiconified(**final** WindowEvent wev){}

@Override

**public** **void** windowIconified(**final** WindowEvent wev){}

@Override

**public** **void** windowOpened(**final** WindowEvent wev){}

}

**Model package:**

**Sound Class:**

**package** model;

**import** java.io.File;

**import** java.io.IOException;

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.Hashtable;

**import** java.util.Set;

**import** javax.sound.sampled.AudioInputStream;

**import** javax.sound.sampled.AudioSystem;

**import** javax.sound.sampled.Clip;

**import** javax.sound.sampled.LineUnavailableException;

**import** javax.sound.sampled.UnsupportedAudioFileException;

//Rhythminator Part 2

//CSE1102 Project 07, Spring 2016

//Bryan Arnold

//4/24/16

//TA: Zigeng Wang

//Section: 51

//Instructor: Jeffrey A. Meunier

/\*\*

\* This class represents sounds that can be played. The sounds are stored

\* in sound files on the disk, but get loaded into memory automatically.

\* **@author** jeff

\*/

**public** **class** Sound{

**private** String \_name;

**private** Clip \_clip;

**public** **static** **final** String ***SOUND\_DIR*** = "sounds";

**private** **static** Hashtable<String, File> *\_soundFiles* = **new** Hashtable<String, File>();

**private** **static** String[] *\_extensions* = {".aiff", ".wav"};

/\*\*

\* Creates a new sound from a sound file on disk. The sound file must be present

\* in the sounds directory.

\* **@param** soundName The base of the file name from which to create the sound.

\*/

**public** Sound(String soundName){

**this**(soundName, *\_soundFiles*.get(soundName));

}

/\*\*

\* Creates a new sound from a sound file. The file can be located anywhere on disk.

\* **@param** soundName What to call the sound.

\* **@param** soundFile The file that refers to the sound file on disk.

\*/

**public** Sound(String soundName, File soundFile){

**if**(soundFile == **null**)

**throw** **new** RuntimeException("there is no sound file having name " + soundName);

\_name = soundFile.getName();

**try**{

\_clip = AudioSystem.*getClip*();

AudioInputStream ais = AudioSystem.*getAudioInputStream*(soundFile);

\_clip.open(ais);

}

**catch**(LineUnavailableException exn){

**throw** **new** RuntimeException("Unable to open the line for audio output. Is another program using it?");

}

**catch**(UnsupportedAudioFileException exn){

System.***out***.println("Unsupported audio format for that file: " + soundFile.toString());

**throw** **new** RuntimeException("Unsupported audio format for that file: " + soundFile.toString());

}

**catch**(IOException exn){

System.***out***.println("Unable to open file: " + soundFile.toString());

**throw** **new** RuntimeException("Unable to open file: " + soundFile.toString());

}

}

/\*\*

\*

\* **@return** Array of all sound files that were found in the sound directory.

\*/

**public** **static** String[] getAllSoundFileNames(){

Set<String> nameSet = *\_soundFiles*.keySet();

String[] names = **new** String[nameSet.size()];

nameSet.toArray(names);

Arrays.*sort*(names);

**return** names;

}

/\*\*

\*

\* **@return** The name of this sound instance.

\*/

**public** String getName(){

**return** \_name;

}

/\*\*

\* Plays the sound. Make sure your speakers are turned up.

\*/

**public** **void** play(){

\_clip.setFramePosition(0); // necessary for re-playing an already-played clip

\_clip.start();

}

/\*\*

\* Scans the default sound directory for sound files.

\*/

**public** **static** **void** scanSoundDir(){

*scanSoundDir*(***SOUND\_DIR***);

}

/\*\*

\* Scans the specified sound directory for sound files.

\* **@param** dirName

\*/

**public** **static** **void** scanSoundDir(String dirName){

ArrayList<String> soundFileNames = **new** ArrayList<String>();

File directory = **new** File(dirName);

File[] files = directory.listFiles();

**for**(File file : files){

String fileName = file.getName();

**for**(String ext : *\_extensions*){

**if**(fileName.endsWith(ext)){

soundFileNames.add(fileName);

String shortName = file.getName();

shortName = shortName.substring(0, shortName.lastIndexOf('.'));

*\_soundFiles*.put(shortName, file);

**break**;

}

}

}

**int** count = *\_soundFiles*.size();

System.***out***.println("Sound.scanSoundDir class found " + count + " sound files in directory '" + ***SOUND\_DIR*** + "'");

*\_soundFiles*.put("(none)", **new** File(""));

}

@Override

**public** String toString(){

**return** "Sound(" + \_name + ")";

}

}

**View Package:**

**BeatNumbers Class:**

**package** view;

**import** javax.swing.JPanel;

//Rhythminator Part 2

//CSE1102 Project 07, Spring 2016

//Bryan Arnold

//4/24/16

//TA: Zigeng Wang

//Section: 51

//Instructor: Jeffrey A. Meunier

**public** **class** BeatNumbers **extends** JPanel {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** BeatNumberSquare[] \_beats;

**public** BeatNumbers(**int** beat){

**this**.setLayout(**null**);

**this**.\_beats = **new** BeatNumberSquare[beat];

**int** n = 0;

**for**(**int** i = 0; i < beat; i++){

BeatNumberSquare beats = **new** BeatNumberSquare(i + 1);

**this**.add(beats);

\_beats[i] = beats;

beats.setLocation(n, 0);

**if** (i < beat - 1){

n += NoteSquare.***SIZE*** + Tracks.***GAP\_SIZE***;

}

}

**this**.setSize((beat \* NoteSquare.***SIZE***) + (beat \* Tracks.***GAP\_SIZE***), NoteSquare.***SIZE***);

}

**public** **void** setBeatNumber(**int** beat){

**for**(**int** i = 0; i < **this**.\_beats.length; i++){

**if**(\_beats[i].getState() == **true**){

\_beats[i].setState(**false**);

}

}

\_beats[beat].setState(**true**);

}

**public** **void** clear(){

**for**(**int** i = 0; i < **this**.\_beats.length; i++){

**if**(\_beats[i].getState() == **true**){

\_beats[i].setState(**false**);

}

}

}

}

**BeatNumberSquare Class:**

**package** view;

**import** gui.Box;

//Rhythminator Part 2

//CSE1102 Project 07, Spring 2016

//Bryan Arnold

//4/24/16

//TA: Zigeng Wang

//Section: 51

//Instructor: Jeffrey A. Meunier

**public** **class** BeatNumberSquare **extends** Box {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** **int** \_beat;

**private** **boolean** \_state;

**public** BeatNumberSquare(**int** beat){

**super**();

**this**.\_beat = beat;

**this**.\_state = **false**;

setText(Integer.*toString*(beat));

setForeground(Colors.***BEATNUMBER\_OFF\_FG***);

setBackground(Colors.***BEATNUMBER\_OFF\_BG***);

setSize(NoteSquare.***SIZE***, NoteSquare.***SIZE***);

}

**public** **boolean** getState(){

**return** **this**.\_state;

}

**public** **int** getBeat(){

**return** **this**.\_beat;

}

**public** **void** setState(**boolean** state) {

**this**.\_state = state;

**if**(**this**.\_state == **true**){

setForeground(Colors.***BEATNUMBER\_ON\_FG***);

setBackground(Colors.***BEATNUMBER\_ON\_BG***);

} **else** **if**(**this**.\_state == **false**){

setForeground(Colors.***BEATNUMBER\_OFF\_FG***);

setBackground(Colors.***BEATNUMBER\_OFF\_BG***);

}

invalidate();

}

}

**Colors Class:**

**package** view;

**import** java.awt.Color;

/\*\*

\* These are the color definition for the view elements.

\* **@author** jeff

\*/

**public** **class** Colors

{

**public** **static** **final** Color ***BEATNUMBER\_OFF\_FG*** = Color.***BLACK***;

**public** **static** **final** Color ***BEATNUMBER\_OFF\_BG*** = Color.***YELLOW***;

**public** **static** **final** Color ***BEATNUMBER\_ON\_FG*** = Color.***BLACK***;

**public** **static** **final** Color ***BEATNUMBER\_ON\_BG*** = Color.***GREEN***;

**public** **static** **final** Color ***NOTESQUARE\_OFF*** = Color.***WHITE***;

**public** **static** **final** Color ***NOTESQUARE\_ON*** = Color.***LIGHT\_GRAY***;

}

**ControlButtons Class:**

**package** view;

**import** javax.swing.JPanel;

**import** controller.Controller;

**import** gui.Button;

//Rhythminator Part 2

//CSE1102 Project 07, Spring 2016

//Bryan Arnold

//4/24/16

//TA: Zigeng Wang

//Section: 51

//Instructor: Jeffrey A. Meunier

**public** **class** ControlButtons **extends** JPanel {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** ControlButtons(Controller controller){

setLayout(**null**);

Button play = **new** Button(controller, "Play");

Button stop = **new** Button(controller, "Stop");

add(play);

add(stop);

play.setSize(60, 30);

stop.setSize(60, 30);

play.setLocation(0, 0);

stop.setLocation(70, 0);

setSize(130, 30);

}

}

**Header Class:**

**package** view;

**import** javax.swing.JPanel;

**import** controller.Controller;

**import** gui.Button;

**import** gui.TextBox;

//Rhythminator Part 2

//CSE1102 Project 07, Spring 2016

//Bryan Arnold

//4/24/16

//TA: Zigeng Wang

//Section: 51

//Instructor: Jeffrey A. Meunier

**public** **class** Header **extends** JPanel{

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** TextBox \_text;

**public** Header(Controller controller){

setLayout(**null**);

TextBox text = **new** TextBox("Unnamed", "Enter a new name: ", "Change name");

**this**.\_text = text;

text.setSize(200, 30);

text.setLocation(0, 0);

text.setBackground(Colors.***NOTESQUARE\_OFF***);

Button load = **new** Button(controller, "Load");

load.setSize(60, 30);

load.setLocation(210, 0);

Button save = **new** Button(controller, "Save");

save.setSize(60, 30);

save.setLocation(280, 0);

Button quit = **new** Button(controller, "Quit");

quit.setSize(60, 30);

quit.setLocation(380, 0);

**this**.setSize(440, 30);

**this**.add(text);

**this**.add(load);

**this**.add(save);

**this**.add(quit);

}

**public** String getFile(){

String text = **this**.\_text.getText();

**return** text;

}

}

**NoteSquare Class:**

**package** view;

**import** javax.swing.JPanel;

**import** controller.Controller;

**import** gui.Button;

**import** gui.TextBox;

//Rhythminator Part 2

//CSE1102 Project 07, Spring 2016

//Bryan Arnold

//4/24/16

//TA: Zigeng Wang

//Section: 51

//Instructor: Jeffrey A. Meunier

**public** **class** Header **extends** JPanel{

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** TextBox \_text;

**public** Header(Controller controller){

setLayout(**null**);

TextBox text = **new** TextBox("Unnamed", "Enter a new name: ", "Change name");

**this**.\_text = text;

text.setSize(200, 30);

text.setLocation(0, 0);

text.setBackground(Colors.***NOTESQUARE\_OFF***);

Button load = **new** Button(controller, "Load");

load.setSize(60, 30);

load.setLocation(210, 0);

Button save = **new** Button(controller, "Save");

save.setSize(60, 30);

save.setLocation(280, 0);

Button quit = **new** Button(controller, "Quit");

quit.setSize(60, 30);

quit.setLocation(380, 0);

**this**.setSize(440, 30);

**this**.add(text);

**this**.add(load);

**this**.add(save);

**this**.add(quit);

}

**public** String getFile(){

String text = **this**.\_text.getText();

**return** text;

}

}

**SoundBank Class:**

**package** view;

**import** javax.swing.JPanel;

**import** controller.Controller;

//Rhythminator Part 2

//CSE1102 Project 07, Spring 2016

//Bryan Arnold

//4/24/16

//TA: Zigeng Wang

//Section: 51

//Instructor: Jeffrey A. Meunier

**public** **class** SoundBank **extends** JPanel{

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

SoundNameBox[] \_soundName;

**public** SoundBank(Controller controller, **int** tracks){

**this**.setLayout(**null**);

SoundNameBox[] \_soundName = **new** SoundNameBox[tracks];

**int** n = 0;

**for**(**int** i = 0; i < tracks; i++){

\_soundName[i] = **new** SoundNameBox(controller, i);

**this**.add(\_soundName[i]);

\_soundName[i].setBackground(Colors.***NOTESQUARE\_OFF***);

\_soundName[i].setLocation(0, n);

**if**(i < tracks - 1){

n = n + Tracks.***GAP\_SIZE*** + SoundNameBox.***HEIGHT***;

}

}

**this**.setSize(130, (tracks \* Tracks.***GAP\_SIZE***) + (tracks \* SoundNameBox.***HEIGHT***));

}

**public** **void** setSoundName(**int** track, String name){

**this**.\_soundName[track].setText(name);

}

}

**SoundChooser Class:**

**package** view;

**import** java.awt.BorderLayout;

**import** java.awt.GridLayout;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.awt.event.KeyEvent;

**import** java.awt.event.KeyListener;

**import** java.awt.event.MouseEvent;

**import** java.awt.event.MouseListener;

**import** javax.swing.JButton;

**import** javax.swing.JDialog;

**import** javax.swing.JList;

**import** javax.swing.JPanel;

**import** javax.swing.JScrollPane;

**import** model.Sound;

**import** controller.Controller;

//Rhythminator Part 2

//CSE1102 Project 07, Spring 2016

//Bryan Arnold

//4/24/16

//TA: Zigeng Wang

//Section: 51

//Instructor: Jeffrey A. Meunier

**public** **class** SoundChooser **extends** JDialog **implements** ActionListener, KeyListener, MouseListener {

**private** JButton \_play, \_ok, \_cancel;

@SuppressWarnings("rawtypes")

**private** JList \_soundNames;

**private** Controller \_controller;

**private** **int** \_trackNumber;

**private** **static** SoundChooser *\_theSoundChooser* = **null**;

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** **static** **void** open(Controller controller, **int** trackNumber, String initialValue){

**if**(*\_theSoundChooser* == **null**)

*\_theSoundChooser* = **new** SoundChooser(controller, trackNumber);

**else**

*\_theSoundChooser*.\_trackNumber = trackNumber;

*\_theSoundChooser*.\_setInitialValue(initialValue);

*\_theSoundChooser*.setVisible(**true**);

}

@SuppressWarnings({ "unchecked", "rawtypes" })

**private** SoundChooser(Controller controller, **int** trackNumber){

**this**.setTitle("Sound chooser");

**this**.setDefaultCloseOperation(JDialog.***HIDE\_ON\_CLOSE***);

**this**.setModal(**true**);

\_controller = controller;

\_trackNumber = trackNumber;

**this**.setLayout(**new** BorderLayout());

String[] soundNames = Sound.*getAllSoundFileNames*();

\_soundNames = **new** JList(soundNames);

\_soundNames.addKeyListener(**this**);

\_soundNames.addMouseListener(**this**);

JScrollPane scrollPane = **new** JScrollPane(\_soundNames);

**this**.add(scrollPane, BorderLayout.***CENTER***);

JPanel buttonPanel = **new** JPanel();

buttonPanel.setLayout(**new** GridLayout(1, 3));

\_ok = **new** JButton("Use this sound");

\_play = **new** JButton("Play");

\_cancel = **new** JButton("Cancel");

\_play.addActionListener(**this**);

\_ok.addActionListener(**this**);

\_cancel.addActionListener(**this**);

buttonPanel.add(\_ok);

buttonPanel.add(\_play);

buttonPanel.add(\_cancel);

**this**.add(buttonPanel, BorderLayout.***SOUTH***);

**this**.setSize(400, 400);

**this**.setLocationRelativeTo(**null**);

}

@Override

**public** **void** actionPerformed(ActionEvent aev){

Object source = aev.getSource();

**if**(source == \_play)

\_play();

**else** **if**(source == \_ok)

\_select();

**else**

\_cancel();

}

@Override

**public** **void** keyPressed(KeyEvent kev){

**int** keyCode = kev.getKeyCode();

**switch**(keyCode){

**case** 10: // enter

\_select();

**break**;

**case** 27: // escape

\_cancel();

**break**;

**case** 32: // space

\_play();

**break**;

}

}

@Override

**public** **void** keyReleased(KeyEvent kev){}

@Override

**public** **void** keyTyped(KeyEvent kev){}

**private** **void** \_cancel(){

**this**.setVisible(**false**);

}

**private** **void** \_play(){

String soundName = (String)\_soundNames.getSelectedValue();

**if**(soundName != **null** && !soundName.equals("(none)")){

Sound s = **new** Sound(soundName);

s.play();

}

}

@Override

**public** **void** mouseClicked(MouseEvent mev){

**if**(mev.getButton() == MouseEvent.***BUTTON1*** && mev.getClickCount() == 2)

\_select();

}

@Override

**public** **void** mouseEntered(MouseEvent mev){}

@Override

**public** **void** mouseExited(MouseEvent mev){}

@Override

**public** **void** mousePressed(MouseEvent mev){}

@Override

**public** **void** mouseReleased(MouseEvent mev){}

**private** **void** \_select(){

**if**(\_controller != **null**){

String soundName = (String)\_soundNames.getSelectedValue();

\_controller.soundNameSelected(\_trackNumber, soundName);

}

**this**.setVisible(**false**);

}

**private** **void** \_setInitialValue(String initialValue){

**if**(initialValue == **null** || initialValue.length() == 0)

\_soundNames.setSelectedIndex(0);

**else**

\_soundNames.setSelectedValue(initialValue, **true**);

}

}

**SoundNameBox Class:**

**package** view;

**import** java.awt.event.MouseEvent;

**import** controller.Controller;

**import** gui.Box;

//Rhythminator Part 2

//CSE1102 Project 07, Spring 2016

//Bryan Arnold

//4/24/16

//TA: Zigeng Wang

//Section: 51

//Instructor: Jeffrey A. Meunier

**public** **class** SoundNameBox **extends** Box{

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** **static** **final** **int** ***WIDTH*** = 130;

**public** **static** **final** **int** ***HEIGHT*** = 30;

**private** Controller \_controller;

**private** **int** \_track;

**public** SoundNameBox(Controller controller, **int** track){

**super**();

**this**.\_track = track;

**this**.\_controller = controller;

setSize(***WIDTH***, ***HEIGHT***);

}

@Override

**public** **void** mousePressed(MouseEvent mev){

String text = getText();

SoundChooser.*open*(\_controller, \_track, text);

}

}

**Tracks Class:**

**package** view;

**import** javax.swing.JPanel;

**import** controller.Controller;

//Rhythminator Part 2

//CSE1102 Project 07, Spring 2016

//Bryan Arnold

//4/24/16

//TA: Zigeng Wang

//Section: 51

//Instructor: Jeffrey A. Meunier

**public** **class** Tracks **extends** JPanel{

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** **static** **final** **int** ***GAP\_SIZE*** = 10;

**public** Tracks(Controller controller, **int** tracks, **int** beats){

**this**.setLayout(**null**);

NoteSquare [][] squares = **new** NoteSquare[tracks][beats];

**int** n = 0;

**int** m = 0;

**for** (**int** i = 0; i < tracks; i++){

**for** (**int** j = 0; j < beats; j++){

squares[i][j] = **new** NoteSquare(controller, i, j);

**this**.add(squares[i][j]);

squares[i][j].setLocation(n, m);

**if** (j < beats - 1){

n += NoteSquare.***SIZE*** + ***GAP\_SIZE***;

} **else** {

m += NoteSquare.***SIZE*** + ***GAP\_SIZE***;

n = 0;

}

}

}

setSize((beats \* NoteSquare.***SIZE***) + (beats \* ***GAP\_SIZE***), ((tracks \* NoteSquare.***SIZE***) + (tracks \* ***GAP\_SIZE***)));

}

}

**View Class:**

**package** view;

**import** gui.NumberSlider;

**import** controller.Controller;

//Rhythminator Part 2

//CSE1102 Project 07, Spring 2016

//Bryan Arnold

//4/24/16

//TA: Zigeng Wang

//Section: 51

//Instructor: Jeffrey A. Meunier

**public** **class** View {

**public** View(Controller controller, **int** numTracks, **int** numBeats){

Window window = **new** Window(controller, "Rhythminator");

window.setSize(900, 500);

Header head = **new** Header(controller);

head.setLocation(10, 10);

window.add(head);

SoundBank sounds = **new** SoundBank(controller, numTracks);

sounds.setLocation(10, 60);

window.add(sounds);

Tracks tracks = **new** Tracks(controller, numTracks, numBeats);

tracks.setLocation(150, 60);

window.add(tracks);

ControlButtons controls = **new** ControlButtons(controller);

controls.setLocation(10, numTracks \* NoteSquare.***SIZE*** + 30 + numTracks \* Tracks.***GAP\_SIZE*** + 30);

window.add(controls);

BeatNumbers beatNum = **new** BeatNumbers(numBeats);

beatNum.setLocation(150, numTracks \* NoteSquare.***SIZE*** + 30 + numTracks \* Tracks.***GAP\_SIZE*** + 30);

window.add(beatNum);

NumberSlider slider = **new** NumberSlider(controller, **null**, 0, 100, 25);

slider.setLocation(150, numTracks \* NoteSquare.***SIZE*** + 30 + numTracks \* Tracks.***GAP\_SIZE*** + 70);

window.add(slider);

window.setVisible(**true**);

}

}

**Window Class:**

**package** view;

**import** java.awt.event.KeyEvent;

**import** controller.Controller;

//Rhythminator Part 2

//CSE1102 Project 07, Spring 2016

//Bryan Arnold

//4/24/16

//TA: Zigeng Wang

//Section: 51

//Instructor: Jeffrey A. Meunier

/\*\*

\* The view.Window extends the gui.Window by notifying the controller

\* when a key event takes place.

\* **@author** jeff

\*/

**public** **class** Window **extends** gui.Window{

**private** Controller \_controller;

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** Window(Controller controller, String name){

**super**(name);

\_controller = controller;

}

@Override

**public** **void** keyPressed(KeyEvent kev){

\_controller.keyPressed(kev.getKeyCode());

}

@Override

**public** **void** keyReleased(KeyEvent kev){

\_controller.keyReleased(kev.getKeyCode());

}

@Override

**public** **void** keyTyped(KeyEvent kev){

//System.out.println("Window kev = " + kev);

\_controller.keyTyped(kev.getKeyChar());

}

}