Criterion C: Development

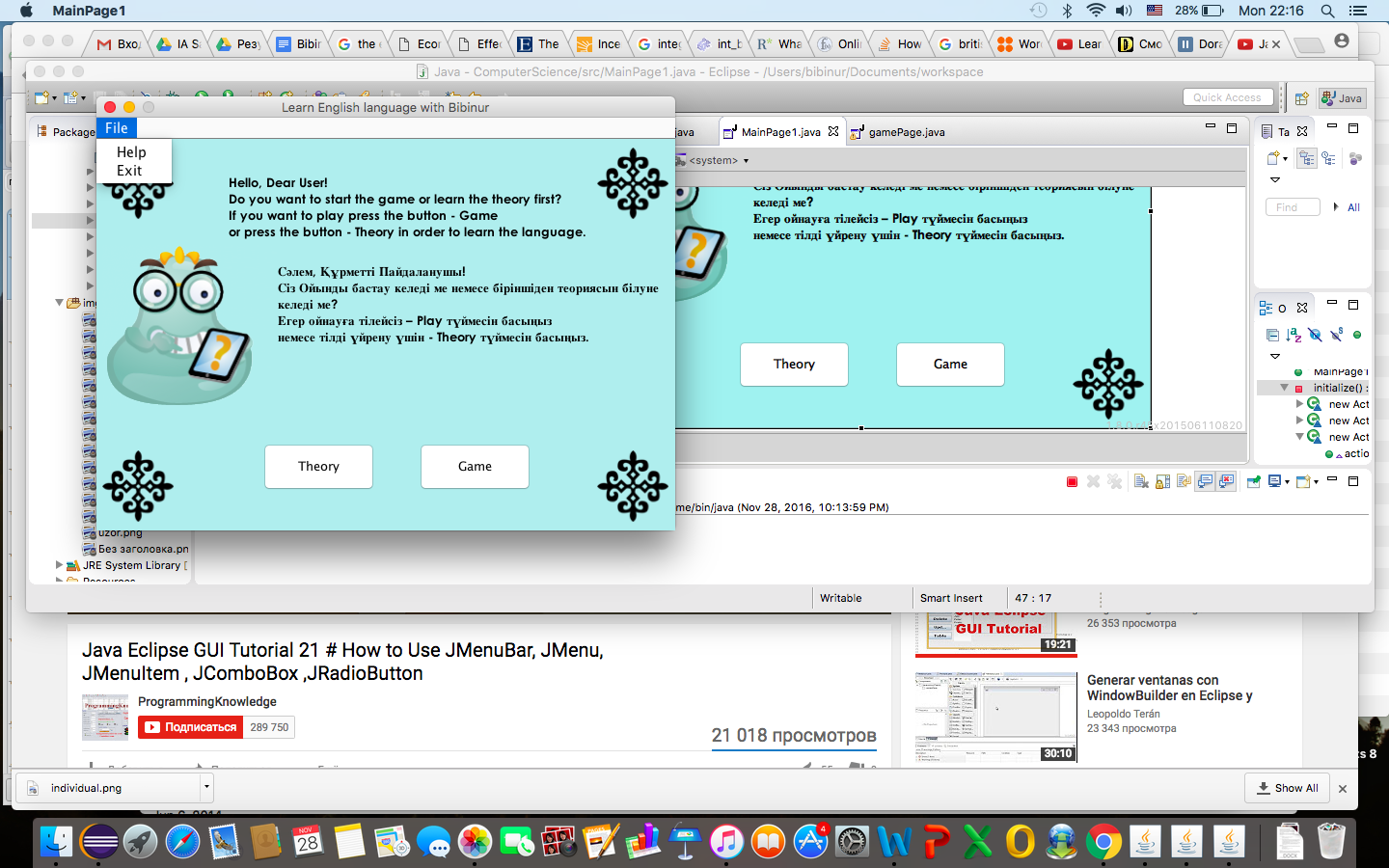
The techniques used in the application include:

* Graphical User Interface
* Global and local variables
* Inheritance
* Audio

**Graphical user interface**

The design of the program is the essential part because it will take the attention of the client and encourages her to continue to use it. The graphical user interface follows the requirements of the client. For instance, Tifanny's color, readable font, graphs, pictures and other tools were used.

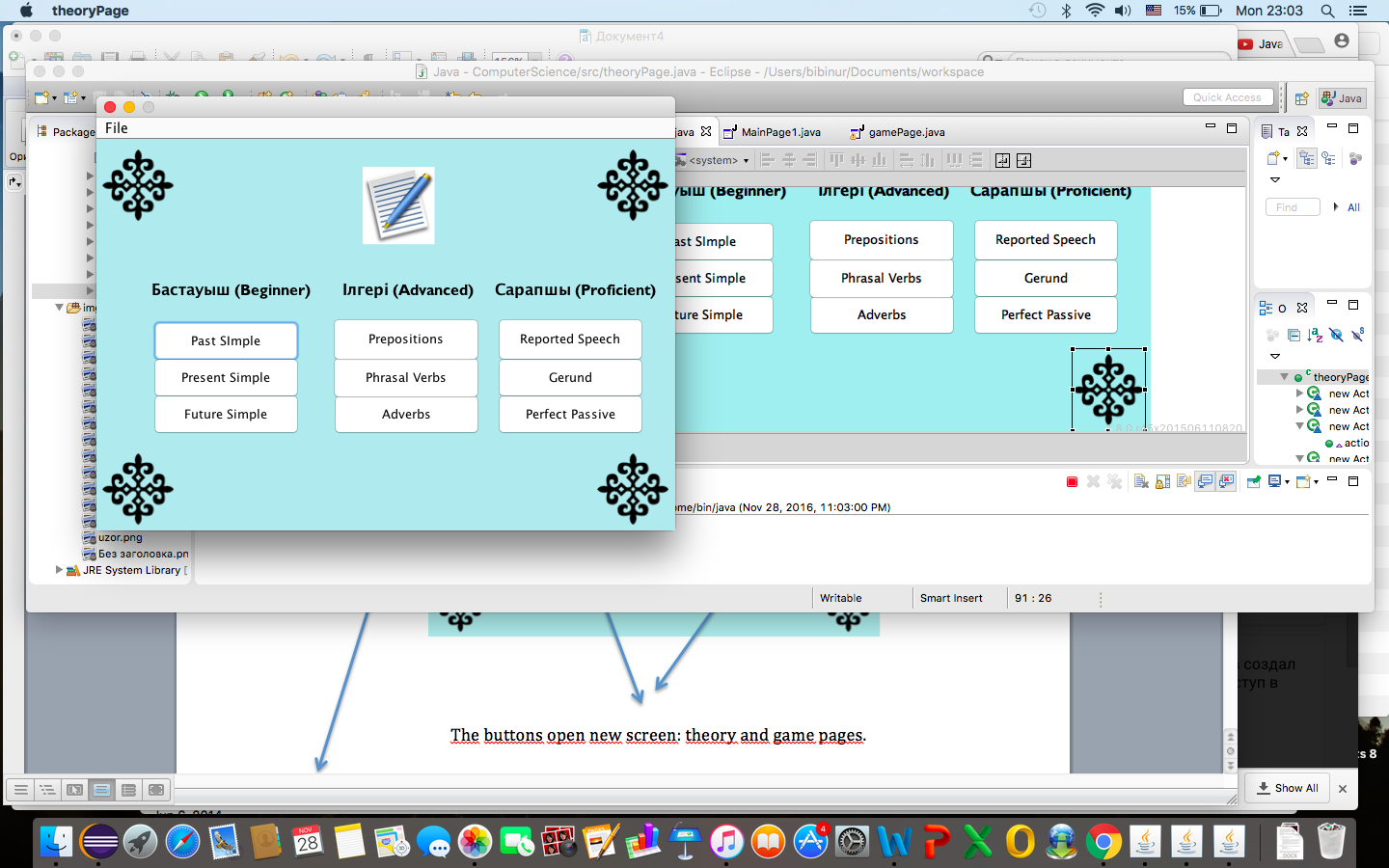
Main window



The menu bar consists of help file that shortly explains the purpose of the application and how does it work. The function exit closes the window.

The buttons open new screen: theory and game pages. The font is big, therefore allows the user read without any problem

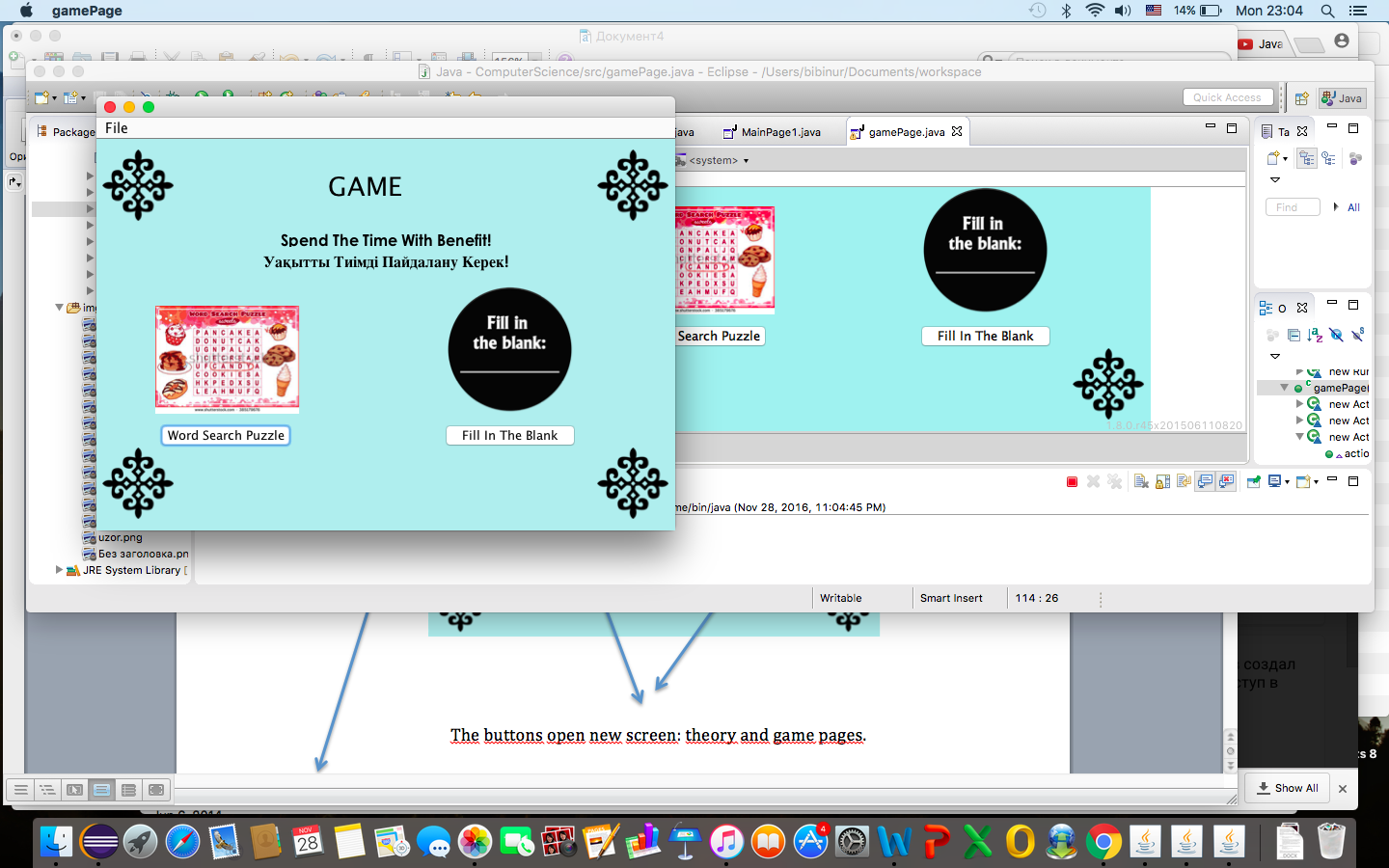
Theory page



The buttons open new window that contains the information of some English grammar rules and so on. The font is big, therefore allows the user read without any problem

The menu bar allows the client to exit the program, to open the Help file and to go to the Game Page after she finished the course.

Game page



The buttons open new window that contains the different types of games.

The font is big, therefore allows the user read without any problem

The menu bar allows the client to exit the program, open the Help File and to go to the Theory Page in order to revise some material.

This is the method that creates the interface of the program. All the other pages follow it: Tiffany’s color; the font size is 13; the Kazakh ornament’s pictures at the angles and menu bar can be met in the main page, help file, theory page and game page

**private** **void** initialize() {

frame = **new** JFrame("Learn English language with Bibinur");

frame.getContentPane().setBackground(**new** Color(175, 238, 238));

frame.setBounds(100, 100, 600, 450);

frame.setDefaultCloseOperation(JFrame.***DISPOSE\_ON\_CLOSE***);

frame.setResizable(**false**);

frame.getContentPane().setLayout(**null**);

JLabel lblNewLabel\_1 = **new** JLabel("");

lblNewLabel\_1.setBounds(6, 6, 98, 80);

Image img = **new** ImageIcon(**this**.getClass().getResource("/pp.png")).getImage();

The design: Kazakh ornaments in the corners of the application. It is used in every window.

lblNewLabel\_1.setIcon(**new** ImageIcon(img));

frame.getContentPane().add(lblNewLabel\_1);

JLabel label = **new** JLabel("");

label.setBounds(6, 315, 98, 90);

Image img1 = **new** ImageIcon(**this**.getClass().getResource("/pp.png")).getImage();

label.setIcon(**new** ImageIcon(img1));

frame.getContentPane().add(label);

JLabel label\_1 = **new** JLabel("");

label\_1.setBounds(519, 6, 81, 80);

Image img2 = **new** ImageIcon(**this**.getClass().getResource("/pp.png")).getImage();

label\_1.setIcon(**new** ImageIcon(img2));

frame.getContentPane().add(label\_1);

JLabel label\_2 = **new** JLabel("");

label\_2.setBounds(519, 315, 75, 90);

Image img3 = **new** ImageIcon(**this**.getClass().getResource("/pp.png")).getImage();

label\_2.setIcon(**new** ImageIcon(img3));

frame.getContentPane().add(label\_2);

JLabel lblNewLabel\_2 = **new** JLabel("");

lblNewLabel\_2.setBounds(6, 105, 181, 180);

Image img4 = **new** ImageIcon(**this**.getClass().getResource("/individual.png")).getImage();

lblNewLabel\_2.setIcon(**new** ImageIcon(img4));

frame.getContentPane().add(lblNewLabel\_2);

JMenuBar menuBar = **new** JMenuBar();

Menu bar that allows opening new window or closing the program

frame.setJMenuBar(menuBar);

JMenu mnNewMenu = **new** JMenu("File");

menuBar.add(mnNewMenu);

JMenuItem mntmNewMenuItem = **new** JMenuItem("Exit");

mntmNewMenuItem.addActionListener(**new** ActionListener() {

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

System.*exit*(JFrame.***EXIT\_ON\_CLOSE***);

}

});

JMenuItem mntmHelp = **new** JMenuItem("Help");

mntmHelp.addActionListener(**new** ActionListener() {

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

helpFile help=**new** helpFile();

help.*help*();

}

});

mnNewMenu.add(mntmHelp);

mnNewMenu.add(mntmNewMenuItem);

}

}

**Variables**

I used local variables in the classes and methods to keep track of different variables and to use them in the application. For instance, to calculate the score of the “Fill In The Blank” game, the variable scores was used.

Moreover, I used a lot of labels and buttons.

**int** scores= 0;

**int** ind=comboBox\_1.getSelectedIndex();

**if**(ind==0)

{

scores=scores+1;

System.***out***.println("Scores"+scores);

}

**else** {

}

JButton btnNewButton\_1 = **new** JButton("Theory");

btnNewButton\_1.setBackground(Color.***CYAN***);

btnNewButton\_1.addActionListener(**new** ActionListener() {

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

theoryPage theory=**new** theoryPage();

theory.setVisible(**true**);

}

});

btnNewButton\_1.setBounds(172, 315, 117, 51);

frame.getContentPane().add(btnNewButton\_1);

**Inheritance**

"Extends Thread" means inheriting all the functions of the Thread class. It is vey useful because less time and resource are needed to create a thread. Threads share their parent process data and code and the intercommunication is easies that process communication.

**private** **class** PlayThread **extends** Thread

{

**byte** tempBuffer[] = **new** **byte**[10000];

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

{

**try**

{

sourceDataLine.open(audioFormat);

sourceDataLine.start();

**int** cnt;

**while**((cnt = audioInputStream.read(

tempBuffer, 0, tempBuffer.length)) != 1

&& stopPlayback == **false**)

{

**if**(cnt >0)

{

sourceDataLine.write(tempBuffer, 0, cnt);

}

}

sourceDataLine.drain();

sourceDataLine.close();

stopBtn.setEnabled(**false**);

playBtn.setEnabled(**true**);

stopPlayback = **false**;

}

**catch** (Exception e)

{

e.printStackTrace();

System.*exit*(0);

}

}

}

**Audio**

The audio was used in order in order to help the user with pronunciation. If the user presses button “Play”, the audio starts. If the button “Stop” is pressed, the audio is stopped. The audio will always begin from the start.

**private** **void** playAudio()

{

The directory.

**try**

{

File soundFile = **new** File("/Users/bibinur/Downloads/Dialog1.wav");

audioInputStream = AudioSystem.*getAudioInputStream*(soundFile);

audioFormat = audioInputStream.getFormat();

System.***out***.println(audioFormat);

DataLine.Info dataLineInfo = **new** DataLine.Info(SourceDataLine.**class**, audioFormat);

sourceDataLine = (SourceDataLine)AudioSystem.*getLine*(dataLineInfo);

**new** PlayThread().start();

}

**catch**(Exception e)

{

e.printStackTrace();

System.*exit*(0);

}

}

**private** **class** PlayThread **extends** Thread

{

**byte** tempBuffer[] = **new** **byte**[10000];

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

{

**try**

{

sourceDataLine.open(audioFormat);

sourceDataLine.start();

**int** cnt;

**while**((cnt = audioInputStream.read(

tempBuffer, 0, tempBuffer.length)) != 1

&& stopPlayback == **false**)

{

**if**(cnt >0)

{

sourceDataLine.write(tempBuffer, 0, cnt);

}

}

sourceDataLine.drain();

sourceDataLine.close();

stopBtn.setEnabled(**false**);

playBtn.setEnabled(**true**);

stopPlayback = **false**;

}

**catch** (Exception e)

{

e.printStackTrace();

System.*exit*(0);

}

}

}