Java Graphic Assessment - Quiz Creator

Usage

1. Make sure the folder structure is like

```
- folder
   - Images
      - background.jpg
      - confirm.png
      - createQ.png
      - leave.png
      - next.png
      - reset.png
      - start.png
      - submit.png
   - Answer.java
   - GameStarts.java
   - Integrate.java
   - Question.java
   - QuestionCreate.java
   - QuestionList.java
   - SelectMenu.java
   - TopMenu.java
```

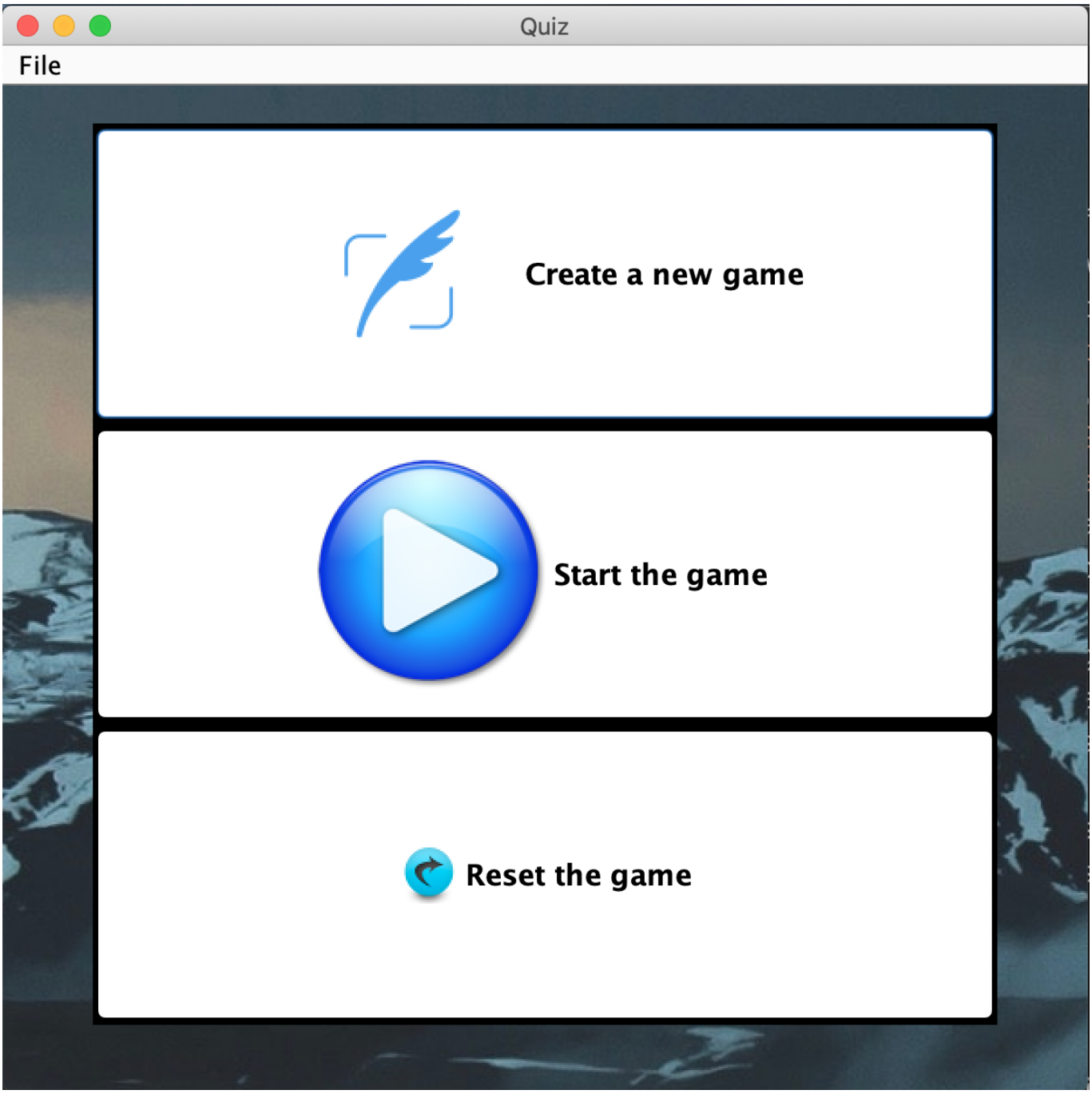
2. Directly compile Integrate.java.

1. Game Design

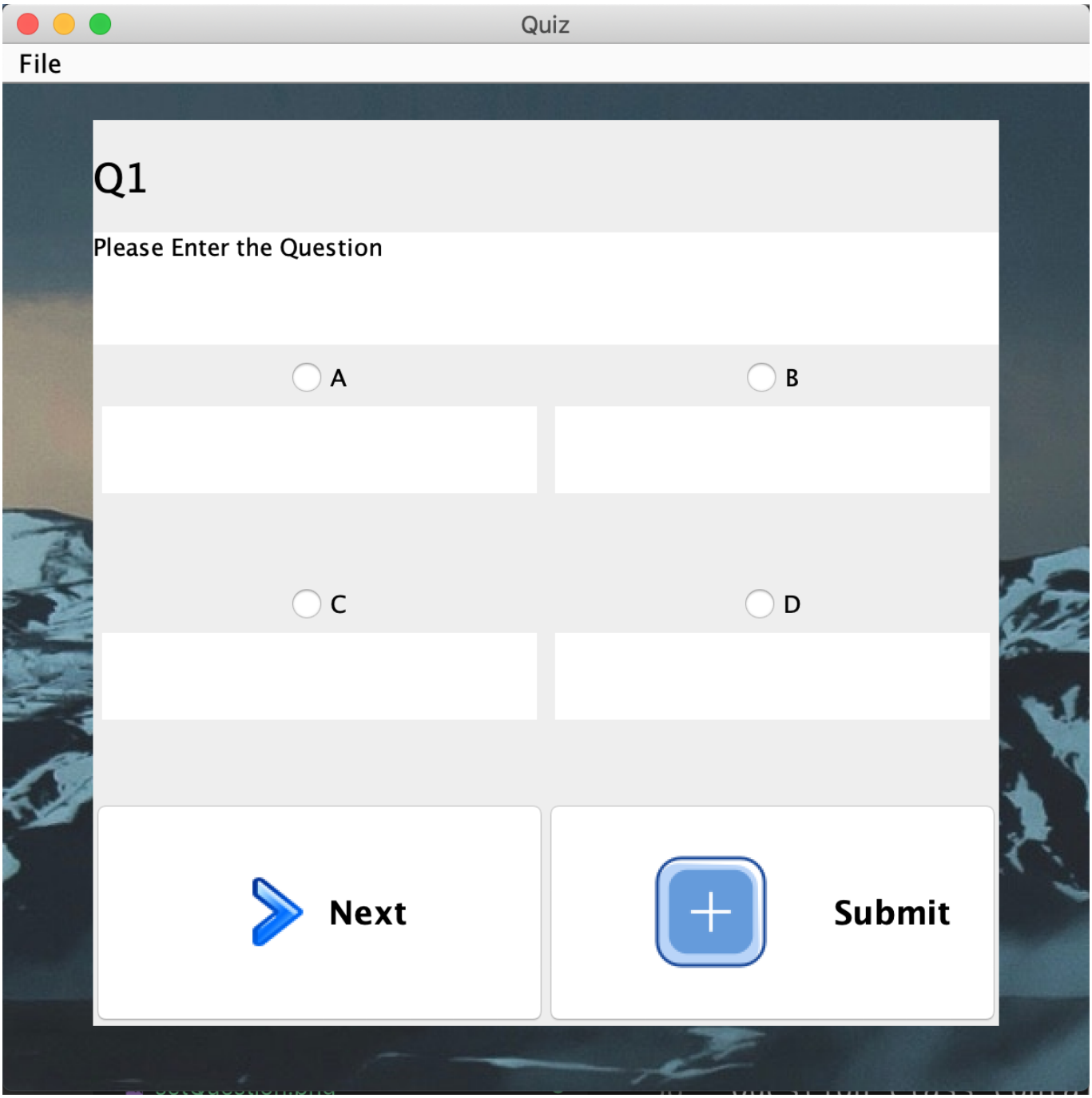
This game's UI interface is designed by the Java Swing and AWT. Considering architecture of MVC structure, `Question` and `QuestionList` classes are the model components. Then, view components contain `GameStarts`, `QuestionCreate`, `SelectMenu` and `TopMenu` classes. Lastly, `Integrate` class controls the model and interfaces.

The purpose of this game is based on the online platform called Kahoot. People can design their own questions and share with their friends. Without using database and web technology, this game is only designed for single player.

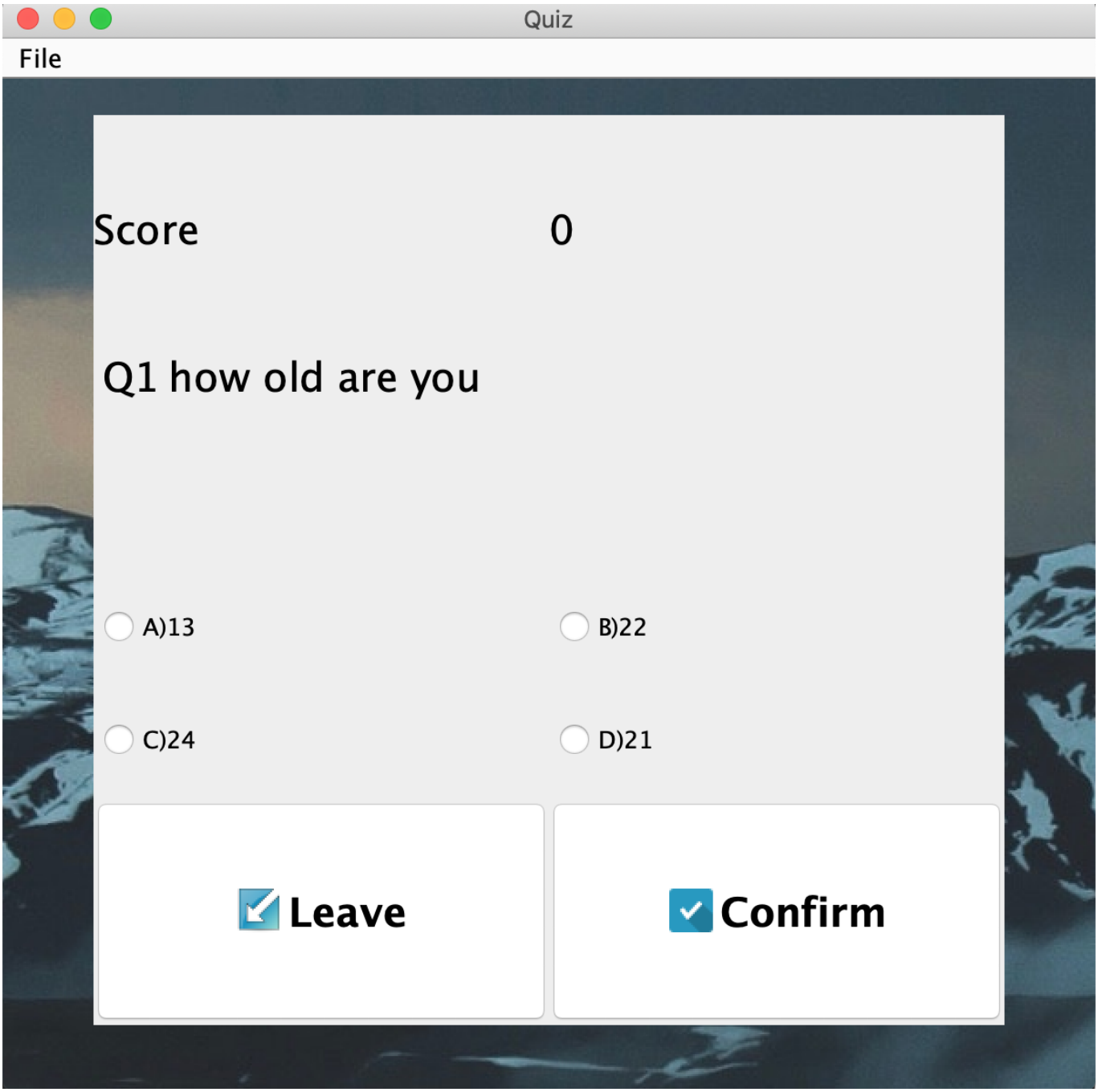
The Starting menu is designed as the below figure:



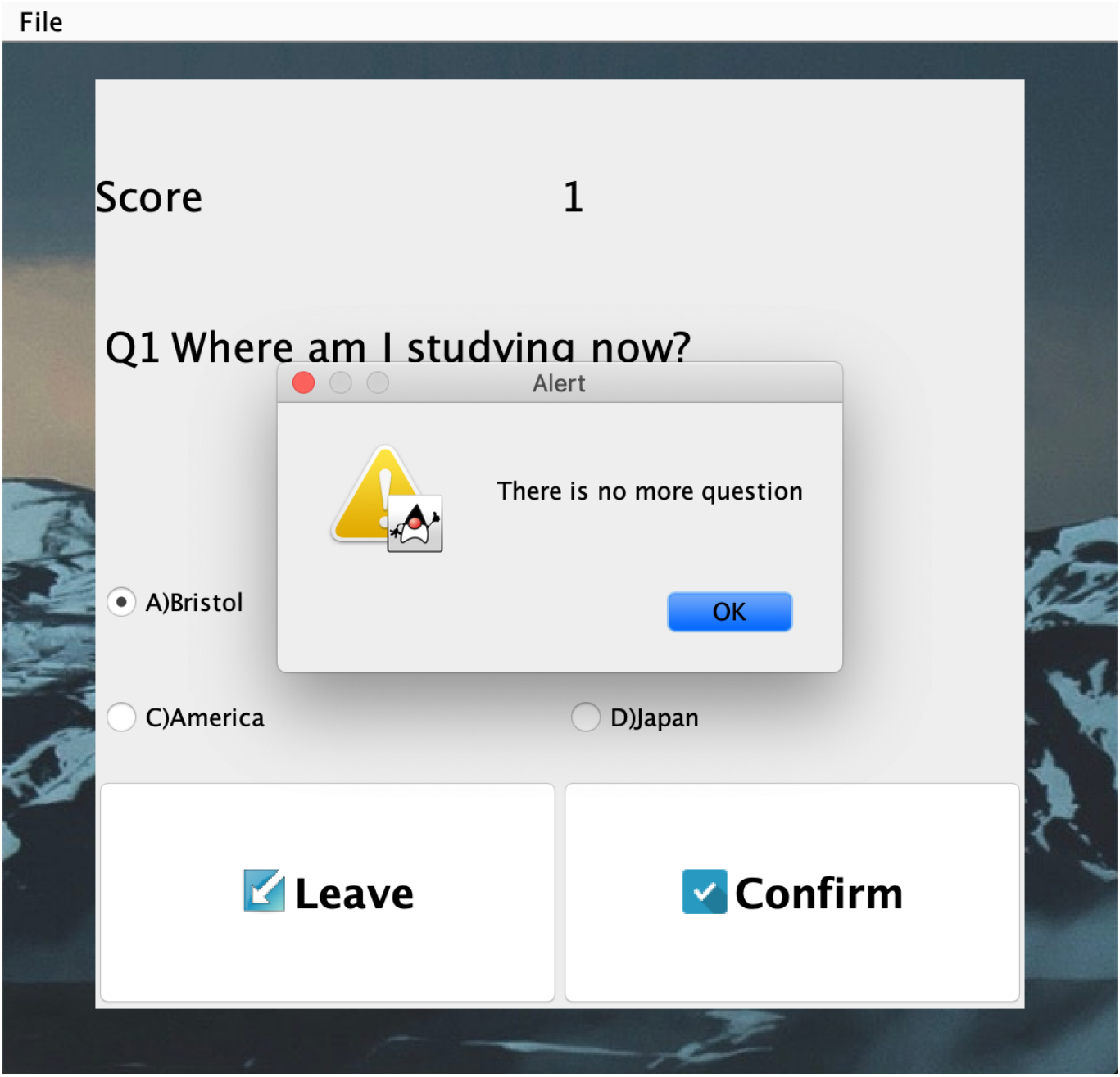
Game creation is designed as the below figure:



Game is designed as the below figure:



End of game is designed as the below figure:



2. Model Design

- Question Class

Question class contains question, selections and answer fields. Particularly, selections field is used the HashMap whose key is Answer type and value is the string.

- `setSelection` method use the key (A, B, C, D) and put the string into the hash table.

```
6   HashMap<Answer,String> selections = new HashMap<Answer,String>();
7   Answer answer;
8
9   void setSelection(Answer ans, String selection){
10     selections.put(ans,selection);
11  }
```

- `setQuestion` method can set the question and all selections.

```
12   void setQuestion(String q, String A, String B, String C, String D, Answer ans)
13   {
14      this.question = q;
15      this.answer = ans;
16      setSelection(Answer.A, A);
17      setSelection(Answer.B, B);
18      setSelection(Answer.C, C);
19      setSelection(Answer.D, D);
20   }
```

- QuestionList Class

QuestionList class contains the Array List of Question type.

- `addQuestion` method can insert the new question
- `numberOfQuestion` method can return the number of questions
- `removeAllQuestion` method can reset the question list.

3. Graphic Learning

- Swing Components

- `JFrame`

`JFrame` can set the window's properties , such as the size of window, action's listener, etc.

```
57          main.setSize(600,600);
58          main.setLayout(null);
59          main.addWindowListener(this);
```

```
84          main.setVisible(true);
```

- `addWindowListener` which is implemented by the WindowListener class and override the windowClosing method to exit the system, when user close the window.

-`JPanel`

Using different `layout manager` and `JPanel` can design the layout. For example, `GridLayout` class can use the column and row to manage the layout.

```
JPanel creator = new JPanel(new GridLayout(4,1));
cmenu = new QuestionCreate(creator);
cmenu.next.addActionListener(this);
cmenu.submit.addActionListener(this);
```

`FlowLayout`, instead, sort the components to a line and can set the components different alignment.

```
// Question
JPanel firstRow = new JPanel(new FlowLayout(FlowLayout.LEFT));
number = new JLabel("Q"+num);
question = new JLabel(q);
number.setFont(labelFont);
question.setFont(labelFont);
```

- JButton

`JButton` class can set a bundle of properties, such as icon, action and font.

```
52    ImageIcon confirmIcon = createImageIcone("/images/confirm.png", "Confirm");
53    ImageIcon cancelIcon = createImageIcone("/images/leave.png", "Cancel");
54    confirm = new JButton("Confirm",confirmIcon);
55    confirm.addActionListener(this);
56    confirm.setFont(btnFont);
```

In `ImageIcon` class, the image is loaded by `createImageIcone` function and return the ImageIcon.

```
64    private static ImageIcon createImageIcone (String path, String description)
65    {
66       if(path != null)
67       {
68          return new ImageIcon(GameStarts.class.getResource(path) , description);
69       }else{
70          System.err.println("Could not find file: " + path);
71          return null;
72       }
73    }
```

Besides, font can be initialised by `Font` class

```
26        Font  btnFont  = new Font(Font.SANS_SERIF,  Font.BOLD, 22);
27        Font  labelFont = new Font(Font.DIALOG, Font.PLAIN, 22);
```

To deal with the action, actionPerformed is override from the `ActionListener` class.

```
90     public void actionPerformed(ActionEvent e){
91        if(e.getSource()==file.close){
92           System.exit(0);
93        }
```

- `JLabel`

`JLabel` example is below and using `setText` method to change the label text.

```
31    JLabel scorelabel = new JLabel("Score");
32    score = new JLabel(Integer.toString(numofcorrect) );
33    scorelabel.setFont(labelFont);
34    score.setFont(labelFont);
```

```
84    void setNewQuestion(String q,int num, String A, String B, String C, String D, Answer ans){
85        number.setText("Q"+num);
86        question.setText(q);
87        selectA.setText("A)" + A);
88        selectB.setText("B)" + B);
89        selectC.setText("C)" + C);
90        selectD.setText("D)" + D);
91        this.ans = ans;
92    }
```

- `JTextArea` and `JRadioButton`

`JTextArea` is a text input which can use `getText` to read the text. `JRadioButton` and `ButtonGroup` are used to limit the choice can be only choose one.

```
JPanel questionRow = new JPanel(new GridLay
question = new JTextArea("Please Enter the
numberQuestion = new JLabel("Q"+number);
numberQuestion.setFont(labelFont);
questionRow.add(numberQuestion);questionRow
// Selection A and B panel
JPanel firstRow = new JPanel(new GridLayout
JPanel panelA = new JPanel();
JPanel panelB = new JPanel();
selectionA = new JTextArea(3,20);
labelA = new JRadioButton("A");
selectionB = new JTextArea(3,20);
labelB = new JRadioButton("B");
```

```
JPanel questionRow = new JPanel(new GridLayout(2,1));
question = new JTextArea("Please Enter the Question");
numberQuestion = new JLabel("Q"+number);
numberQuestion.setFont(labelFont);
questionRow.add(numberQuestion);questionRow.add(question);
// Selection A and B panel
JPanel firstRow = new JPanel(new GridLayout(1,2));
JPanel panelA = new JPanel();
JPanel panelB = new JPanel();
selectionA = new JTextArea(3,20);
labelA = new JRadioButton("A");
selectionB = new JTextArea(3,20);
labelB = new JRadioButton("B");
panelA.add(labelA);panelA.add(selectionA);
panelB.add(labelB);panelB.add(selectionB);
```