Assignment 2

SENG 401

Gerardo Garcia de Leon
30172099

Feb 25th, 2025

Category	Level	Rating	Location (File/line #)	Comments
Data Level				
Magic Numbers (instead of constants)	Data	A	Question.java/20	5 is used with no context
Data members are public	Data	C	GameRecord.java/10, 11, 12, 13, 14 QA.java/8, 9, 10, 11 Question.java/10, 11, 12, 13	
Out of bounds array indexes	Data			
Other, Which one?	Data	B	GameFrame.java/25, 91, 92, 110, 405 420, 429	The array name should be "rewards" not "rewords" causing confusion
Other, Which one?	Data	A	QA.java/11	Can change to a list rather than an array
Statement Level				
Commented-out code fragments	Statement	A	Question.java/57	Line is commented out and does not execute
Improper or inconsistent naming convention	Statement			
Incorrect conditions in loops or conditionals	Statement			
Other. Which one?				
Routine (Method/Function) Level	Routine			
Code is duplicated	Routine	C	GameFrame.java/195, 203, 211, 219, 227, 235	actionPerformed is repeated in

				every answer button and can be combined into one action listener
A method is too long	Routine		GameFrame.java/169-374	Method is too long and can be split into multiple methods
A loop is too deeply nested	Routine			
Parameter list has too many parameters	Routine			
A method uses more features of another class than its own	Routine			
Confusing or incorrect error messages	Routine			
Unhandled edge cases	Routine			
Redundant or unnecessary comments	Routine			
Lack of method headers	Routine			
Other. Which one?	Routine	A	QuestionLoader.java/26	Since sets only allow unique values, questionsList.size() is less than number, this causes an infinite loop
Class Level				
Inappropriate accessibility (public, private, etc.)	Class			

Inheritance hierarchies must be modified in parallel	Class			
Changes require parallel modifications to multiple classes	Class			
A class has poor cohesion	Class	B	MillionaireConnection.java/83, 114, 137, 158, 179, 200	Class handles multiple responsibilities such as login, register and retrieving money and records
A class doesn't do very much	Class			
One class is overly intimate with another	Class			
Other. Which one?	Class	A	GameFrame.java/52	answerButtons can be initialized in the beginning rather than inside the constructor
System Level				
Architecture issue	System			
Security issue	System			
Other. Which one?	System	A	MillionaireConnection.java/45	The catch statement should close the connection as it failed

PART 2:

Data level:
Replaced the magic number in Question.java with a variable that has a proper name. This improves readability and ensures that there is an explanation to the number being used in the code. Without this, it may cause confusion to what the number is meant to represent.
BEFORE:

```
16
17 ⌄    public Question(QA qa) {
18          text = qa.question;
19          int pos = 1;
20          while (pos < 5 && qa.wrongs[pos] != null) {
21              pos++;
22          }
23          size = pos + 1;
24
25          answers = new ArrayList<>();
```

AFTER:

```
16
17    public Question(QA qa) {
18        text = qa.question;
19        int pos = 1;
20        int MAX_NUM_OF_QUESTIONS = 5;
21        while (pos < MAX_NUM_OF_QUESTIONS && qa.wrongs[pos] != null) {
22            pos++;
23        }
24        size = pos + 1;
25
26        answers = new ArrayList<>();
```

Deleted the commented line in Question.java to eliminate useless code which was previously crowding the method. By deleting this, we ensure that every line inside the method brings value and is performing an action required to make the program run. Commented-out lines of code do nothing but extend the length of the code and add redundancy.

BEFORE:

```
47        @Override
48 ⌄    public String toString() {
49          StringBuilder builder = new StringBuilder();
50
51          builder.append(text).append("\n");
52          int pos = 0;
53          for (String ans : answers) {
54              builder.append(++pos).append(". ").append(ans).append("\n");
55          }
56
57          // builder.append("Correct: ").append(rightAnswerPos).append("\n");
58          return builder.toString();
59      }
60  }
```

AFTER:

```
48        @Override
49    public String toString() {
50          StringBuilder builder = new StringBuilder();
51
52          builder.append(text).append(str:"\n");
53          int pos = 0;
54          for (String ans : answers) {
55              builder.append(++pos).append(str:". ").append(ans).append(str:"\n");
56          }
57          return builder.toString();
58      }
59  }
60
```

Routine Level:
Shortened the method in GameFrame.java to ensure it is not too long. This was done by extracting the other methods and creating separate ones to increase readability and facilitate future maintenance of the code. The code was previously too long to read and was initializing everything in an unorganized manner. With this change, it is now clear and more organized what was changed to the code.
BEFORE (Shortened for sake of space, would require too many screenshots):

```java
169 ∨        private void initComponents() {
170
171              button_answer1 = new javax.swing.JButton();
172              button_answer2 = new javax.swing.JButton();
173              button_answer3 = new javax.swing.JButton();
174              button_answer4 = new javax.swing.JButton();
175              button_answer5 = new javax.swing.JButton();
176              button_answer6 = new javax.swing.JButton();
177              jScrollPane1 = new javax.swing.JScrollPane();
178              area_question = new javax.swing.JTextArea();
179              button_next = new javax.swing.JButton();
180              button_quit = new javax.swing.JButton();
181              label_next = new javax.swing.JLabel();
182              label_got = new javax.swing.JLabel();
183              label_stage = new javax.swing.JLabel();
184              timerBar = new javax.swing.JProgressBar();
185              button_addTime = new javax.swing.JButton();
186              button_remove = new javax.swing.JButton();
187              button_change = new javax.swing.JButton();
188
189              setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
190              setResizable(false);
191
192              button_answer1.setFont(new java.awt.Font("Consolas", 0, 12)); // NOI18N
193              button_answer1.setText("a1");
194              button_answer1.addActionListener(new java.awt.event.ActionListener() {
195                  public void actionPerformed(java.awt.event.ActionEvent evt) {
196                      answer1Action(evt);
197                  }
198              });
199
200              button_answer2.setFont(new java.awt.Font("Consolas", 0, 12)); // NOI18N
201              button_answer2.setText("a2");
202              button_answer2.addActionListener(new java.awt.event.ActionListener() {
203                  public void actionPerformed(java.awt.event.ActionEvent evt) {
204                      answer2Action(evt);
205                  }
206              });
207
208              button_answer3.setFont(new java.awt.Font("Consolas", 0, 12)); // NOI18N
209              button_answer3.setText("a3");
210              button_answer3.addActionListener(new java.awt.event.ActionListener() {
211                  public void actionPerformed(java.awt.event.ActionEvent evt) {
212                      answer3Action(evt);
213                  }
214              });
215
216              button_answer4.setFont(new java.awt.Font("Consolas", 0, 12)); // NOI18N
217              button_answer4.setText("a4");
218              button_answer4.addActionListener(new java.awt.event.ActionListener() {
219                  public void actionPerformed(java.awt.event.ActionEvent evt) {
220                      answer4Action(evt);
221                  }
```

```java
222             });
223
224             button_answer5.setFont(new java.awt.Font("Consolas", 0, 12)); // NOI18N
225             button_answer5.setText("a5");
226             button_answer5.addActionListener(new java.awt.event.ActionListener() {
227                 public void actionPerformed(java.awt.event.ActionEvent evt) {
228                     answer5Action(evt);
229                 }
230             });
231
232             button_answer6.setFont(new java.awt.Font("Consolas", 0, 12)); // NOI18N
233             button_answer6.setText("a6");
234             button_answer6.addActionListener(new java.awt.event.ActionListener() {
235                 public void actionPerformed(java.awt.event.ActionEvent evt) {
236                     answer6Action(evt);
237                 }
238             });
239
240             area_question.setEditable(false);
241             area_question.setColumns(20);
242             area_question.setFont(new java.awt.Font("Consolas", 0, 14)); // NOI18N
243             area_question.setLineWrap(true);
244             area_question.setRows(5);
245             area_question.setWrapStyleWord(true);
246             area_question.setCursor(new java.awt.Cursor(java.awt.Cursor.TEXT_CURSOR));
247             area_question.setFocusable(false);
248             area_question.setMargin(new java.awt.Insets(10, 5, 10, 5));
249             jScrollPane1.setViewportView(area_question);
250
251             button_next.setFont(new java.awt.Font("Consolas", 0, 12)); // NOI18N
252             button_next.setText("Next");
253             button_next.addActionListener(new java.awt.event.ActionListener() {
254                 public void actionPerformed(java.awt.event.ActionEvent evt) {
255                     nextAction(evt);
256                 }
257             });
258
259             button_quit.setFont(new java.awt.Font("Consolas", 0, 12)); // NOI18N
260             button_quit.setText("Quit");
261             button_quit.addActionListener(new java.awt.event.ActionListener() {
262                 public void actionPerformed(java.awt.event.ActionEvent evt) {
263                     quitAction(evt);
264                 }
265             });
266
267             label_next.setFont(new java.awt.Font("Consolas", 0, 12)); // NOI18N
268             label_next.setText("Next");
269
270             label_got.setFont(new java.awt.Font("Consolas", 0, 12)); // NOI18N
271             label_got.setText("Got");
272
273             label_stage.setFont(new java.awt.Font("Consolas", 0, 12)); // NOI18N
274             label_stage.setText("Stage");
```

AFTER (Also shortened for sake of space):

```java
171     private void initComponents() {
172         initializeButtons();
173         initilizeAreaQuestion();
174         initializeLabels();
175
176         jScrollPane1 = new javax.swing.JScrollPane();
177         jScrollPane1.setViewportView(area_question);
178
179         timerBar = new javax.swing.JProgressBar();
180
181         setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
182         setResizable(resizable:false);
183
184         initializeLayout();
185
186         pack();
187     }// </editor-fold>//GEN-END:initComponents
```

```java
360     private void initializeButtons(){
361         initializeAnsButton1();
362         initializeAnsButton2();
363         initializeAnsButton3();
364         initializeAnsButton4();
365         initializeAnsButton5();
366         initializeAnsButton6();
367         initializeNextButton();
368         initializeQuitButton();
369         initializeAddTimeButton();
370         initializeRemoveButton();
371         initializeChangeButton();
372     }
373
374     private void initializeAnsButton1(){
375         button_answer1 = new javax.swing.JButton();
376         button_answer1.setFont(new java.awt.Font(name:"Consolas", style:0, size:12)); // NOI18N
377         button_answer1.setText(text:"a1");
378         button_answer1.addActionListener(new java.awt.event.ActionListener() {
379             public void actionPerformed(java.awt.event.ActionEvent evt) {
380                 answer1Action(evt);
381             }
382         });
383     }
```

```
384
385        private void initializeAnsButton2(){
386            button_answer2 = new javax.swing.JButton();
387            button_answer2.setFont(new java.awt.Font(name:"Consolas", style:0, size:12)); // NOI18N
388            button_answer2.setText(text:"a2");
389            button_answer2.addActionListener(new java.awt.event.ActionListener() {
390                public void actionPerformed(java.awt.event.ActionEvent evt) {
391                    answer2Action(evt);
392                }
393            });
394        }
395
396        private void initializeAnsButton3(){
397            button_answer3 = new javax.swing.JButton();
398            button_answer3.setFont(new java.awt.Font(name:"Consolas", style:0, size:12)); // NOI18N
399            button_answer3.setText(text:"a3");
400            button_answer3.addActionListener(new java.awt.event.ActionListener() {
401                public void actionPerformed(java.awt.event.ActionEvent evt) {
402                    answer3Action(evt);
403                }
404            });
405        }
```

```
    private void initilizeAreaQuestion(){
        area_question = new javax.swing.JTextArea();
        area_question.setEditable(b:false);
        area_question.setColumns(columns:20);
        area_question.setFont(new java.awt.Font(name:"Consolas", style:0, size:14)); // NOI18N
        area_question.setLineWrap(wrap:true);
        area_question.setRows(rows:5);
        area_question.setWrapStyleWord(word:true);
        area_question.setCursor(new java.awt.Cursor(java.awt.Cursor.TEXT_CURSOR));
        area_question.setFocusable(focusable:false);
        area_question.setMargin(new java.awt.Insets(top:10, left:5, bottom:10, right:5));
    }

    private void initializeLabels(){
        label_next = new javax.swing.JLabel();
        label_next.setFont(new java.awt.Font(name:"Consolas", style:0, size:12)); // NOI18N
        label_next.setText(text:"Next");

        label_got = new javax.swing.JLabel();
        label_got.setFont(new java.awt.Font(name:"Consolas", style:0, size:12)); // NOI18N
        label_got.setText(text:"Got");

        label_stage = new javax.swing.JLabel();
        label_stage.setFont(new java.awt.Font(name:"Consolas", style:0, size:12)); // NOI18N
        label_stage.setText(text:"Stage");
    }
```

Created one common action listener in GameFrame.java and connected them to all buttons which determine the input using a switch statement. This was done to remove the duplicated code in each button for "actionPerformed" and simplified it. The new version of the code is now more readable and eliminated the repetition of code.

BEFORE (Shortened for space):

```
194            button_answer1.addActionListener(new java.awt.event.ActionListener() {
195                public void actionPerformed(java.awt.event.ActionEvent evt) {
196                    answer1Action(evt);
197                }
198            });
202            button_answer2.addActionListener(new java.awt.event.ActionListener() {
203                public void actionPerformed(java.awt.event.ActionEvent evt) {
204                    answer2Action(evt);
205                }
206            });
208            button_answer3.setFont(new java.awt.Font("Consolas", 0, 12)); // NOI18N
209            button_answer3.setText("a3");
210            button_answer3.addActionListener(new java.awt.event.ActionListener() {
211                public void actionPerformed(java.awt.event.ActionEvent evt) {
212                    answer3Action(evt);
213                }
214            });
```

AFTER (Shortened):

```java
362    ActionListener commonListener = new ActionListener() {
363        @Override
364        public void actionPerformed(ActionEvent evt) {
365            switch (evt.getActionCommand()) {
366                case "answer1":
367                    answer1Action(evt);
368                    break;
369                case "answer2":
370                    answer2Action(evt);
371                    break;
372                case "answer3":
373                    answer3Action(evt);
374                    break;
375                case "answer4":
376                    answer4Action(evt);
377                    break;
378                case "answer5":
379                    answer5Action(evt);
380                    break;
381                case "answer6":
382                    answer6Action(evt);
383                    break;
384                case "next":
385                    nextAction(evt);
386                    break;
387                case "quit":
388                    quitAction(evt);
389                    break;
390                case "addTime":
391                    resetTimeAction(evt);
392                    break;
393                case "remove":
394                    removeWrongAction(evt);
395                    break;
396                case "change":
                       changeAction(evt);
```

```java
419    private void initializeAnsButton1(){
420        button_answer1 = new javax.swing.JButton();
421        button_answer1.setFont(new java.awt.Font(name:"Consolas", style:0, size:12)); // NOI18N
422        button_answer1.setText(text:"a1");
423        button_answer1.setActionCommand(actionCommand:"answer1");
424        button_answer1.addActionListener(commonListener);
425    }
```

```java
427    private void initializeAnsButton2(){
428        button_answer2 = new javax.swing.JButton();
429        button_answer2.setFont(new java.awt.Font(name:"Consolas", style:0, size:12)); // NOI18N
430        button_answer2.setText(text:"a2");
431        button_answer2.setActionCommand(actionCommand:"answer2");
432        button_answer2.addActionListener(commonListener);
433    }
```

```java
435    private void initializeAnsButton3(){
436        button_answer3 = new javax.swing.JButton();
437        button_answer3.setFont(new java.awt.Font(name:"Consolas", style:0, size:12)); // NOI18N
438        button_answer3.setText(text:"a3");
439        button_answer3.setActionCommand(actionCommand:"answer3");
440        button_answer3.addActionListener(commonListener);
441    }
```

Class Level:

The MilionaireConnection.java class should be refactored and separated to handle less responsibilities as the current code has very poor cohesion. It currently handles creating the database connection, retrieving the questions, logging in, registering, retrieving a user's balance, adding records to the database and retrieving previous records. These can all be separated into separate classes that can handle similar tasks to improve cohesion and allow the MillionaireConnection class to handle the database connection and other similar processes. This is necessary as a class with poor cohesion makes it increasingly difficult to maintain a program as it is unorganized, and each class does not perform what it is meant to perform. By improving this, the issue is resolved, and the program can be more easily tested and maintained for future updates.