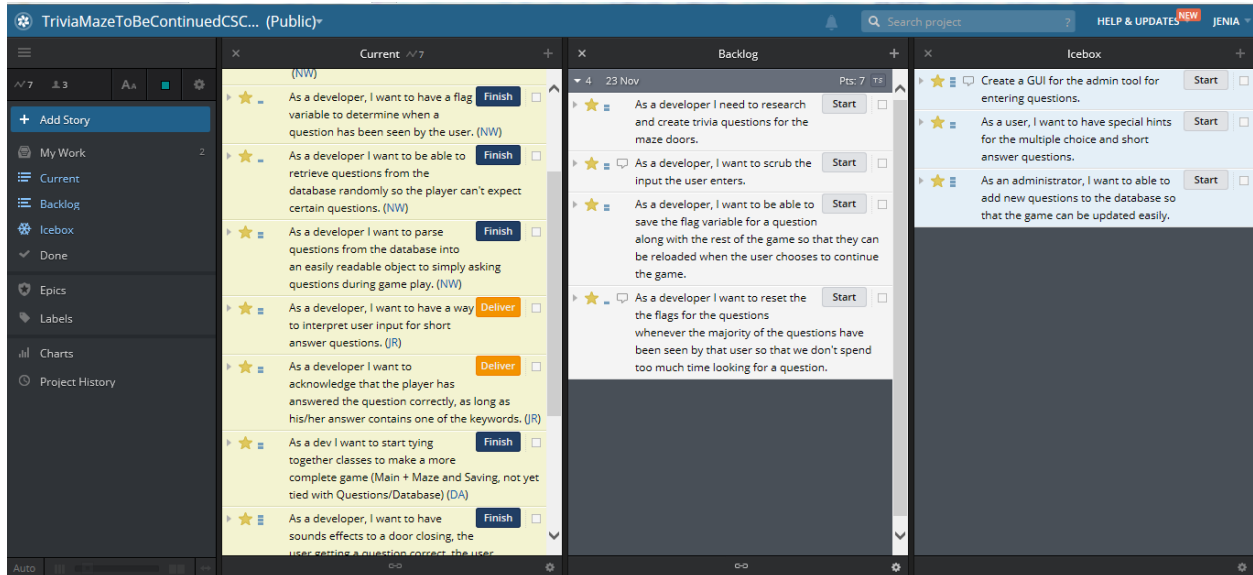


CSCD 350 Software Engineering

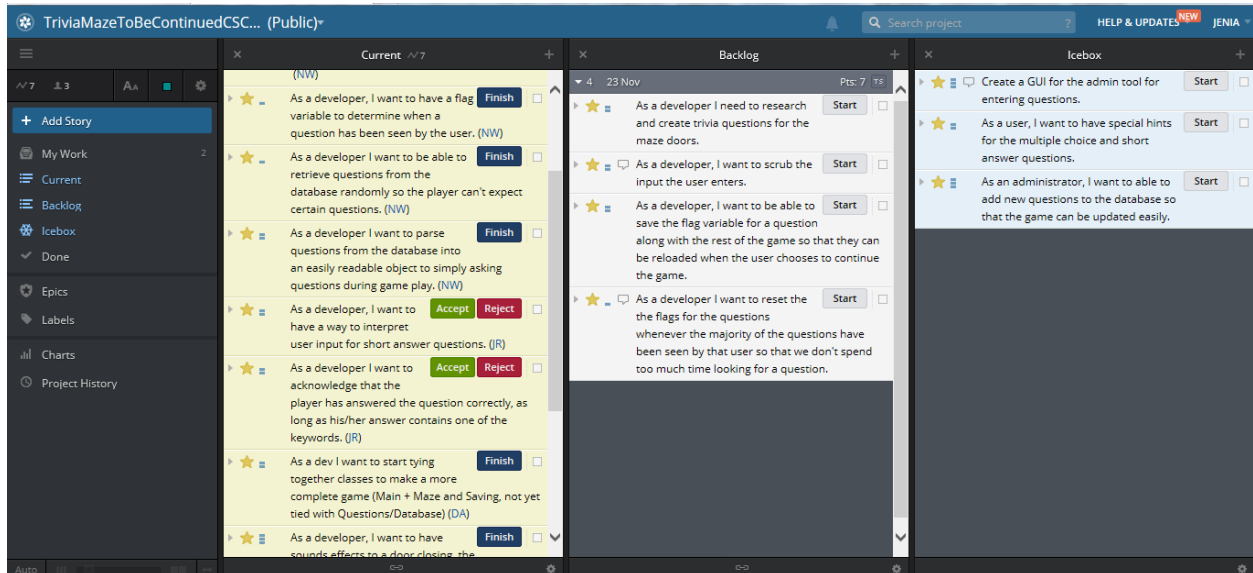
End of 3rd Iteration

Jenia's work for Iteration 3:

Delivering user stories:



Stories accepted:



Committing the changes to GitHub:

The image displays two screenshots of the GitHub web interface, showing the commit history and file changes for the 'jenia_maze' repository.

Top Screenshot:

- Repository:** jenia_maze
- Branches:** master, jenia_maze
- Commit History:**
 - Updates to the questions hierarchy** (10+ changes, just now by Jenia)
 - Implementation of the questions hierarchy** (5+ changes, 10 days ago by Jenia)
 - Files for maze implementation** (4+ changes, 11 days ago by Jenia)
- Commit Details:** Jenia 6836308
 - Commit Message:** Updates to the questions hierarchy. Several updates to the questions hierarchy. Unit tests and a UML diagram are also provided.
 - Files Changed:**
 - AllTests.java
 - MultipleChoiceQuestion.java
 - MultipleChoiceQuestionTest.java
 - Question.java
 - QuestionFactory.java
 - QuestionsHierarchyUML.pdf
 - ShortAnswerQuestion.java
 - ShortAnswerQuestionTest.java
 - TrueFalseQuestion.java
 - TrueFalseQuestionTest.java

Bottom Screenshot:

- Repository:** jenia_maze
- Branches:** master, jenia_maze
- Commit History:**
 - Update to the Maze** (4+ changes, just now by Jenia)
 - Updates to the questions hierarchy** (10+ changes, 3 minutes ago by Jenia)
 - Implementation of the questions hierarchy** (5+ changes, 10 days ago by Jenia)
 - Files for maze implementation** (4+ changes, 11 days ago by Jenia)
- Commit Details:** Jenia 91eae63
 - Commit Message:** Update to the Maze. Added the functionality to randomly fill the question slots in the maze with an indicator for one of the three question types.
 - Files Changed:**
 - CellType.java
 - Ellers.java
 - Maze.java
 - MazeTester.java

Code Snippets:

The image displays two screenshots related to a Java project named TriviaMaze.

The top screenshot shows the GitHub repository interface for `jenia_maze`. The left sidebar lists repositories: `GitHub`, `CSCD350Team`, `CSCD350TriviaMaze`, `TheEternalConundrum`, and `Tutorial`. The main area shows the commit history for the `jenia_maze` branch. The latest commit, titled "Update to the Maze" by Jenia, is highlighted. The commit message and code changes are visible, showing updates to the maze implementation and question hierarchy.

The bottom screenshot shows the Eclipse IDE with the `triviamaize/src/triviamaize/AllTests.java` file open. The code is as follows:

```
1 package triviamaize;
2
3 import org.junit.runner.RunWith;
4
5 @RunWith(Suite.class)
6 @SuiteClasses({ MultipleChoiceQuestionTest.class,
7                 ShortAnswerQuestionTest.class, TrueFalseQuestionTest.class })
8 public class AllTests {
9
10
11 }
12
13
```

The IDE also shows the JUnit runner output in the console, indicating that the tests passed successfully.

```
ShortAnswer... MultipleChoi... TrueFalseQu... ShortAnswer... MultipleChoi... AllTests.java »
98 @Override
99 public boolean checkCorrectAnswer(String input)
100 {
101     char ch = input.toUpperCase().charAt(0);
102     String answer = getAnswerAtIndex(ch);
103     return answer.equalsIgnoreCase(this.getCorrectAnswer());
104 } //end checkCorrectAnswer
105
106
107 /*
108  * This overrides the hook for the Question superclass.
109  * Multiple choice questions have the special functionality to have
110  * the answer choices randomly shuffled.
111  */
112
113 @Override
114 public void specialFunction()
115 {
116     String [] choices = this.getChoices();
117     for (int i = 0; i < choices.length; i++)
118     {
119         int j = (int)(Math.random() * choices.length);
120         String temp = choices[i];
121         choices[i] = choices[j];
122         choices[j] = temp;
123     } //end for
124 } //end specialFunction
125
126
127 /*
128  * Maybe eliminate one or two of the wrong answer choices.
129  * Notify the user of the wrong choice(s).
130  */
```

```
ShortAnswer... MultipleChoi... TrueFalseQu... ShortAnswer... MultipleChoi... AllTests.java »
54 {
55     assertEquals("Olympia", test.getCorrectAnswer());
56 } //end testGetCorrectAnswer
57
58
59 @Test
60 public void testGetChoices()
61 {
62     assertEquals("Seattle", test.getChoices()[0]);
63     assertEquals("Olympia", test.getChoices()[1]);
64     assertEquals("Spokane", test.getChoices()[2]);
65     assertEquals("Washington DC", test.getChoices()[3]);
66 } //end testGetChoices
67
68
69 @Test
70 public void testCheckCorrectAnswer()
71 {
72     /* These are acceptable answers */
73     String a1 = "B";
74     String a2 = "b";
75
76     assertEquals(true, test.checkCorrectAnswer(a1));
77     assertEquals(true, test.checkCorrectAnswer(a2));
78
79     /* These are unacceptable answers */
80     String a3 = "a";
81     String a4 = "C";
82
83     assertEquals(false, test.checkCorrectAnswer(a3));
84     assertEquals(false, test.checkCorrectAnswer(a4));
85 } //end testCheckCorrectAnswer
86
```

```
ShortAnswer... MultipleChoi... TrueFalseQu... ShortAnswer... MultipleChoi... AllTests.java »6
60 public void testGetKeywords()
61 {
62     assertEquals("red", test.getChoices()[0]);
63     assertEquals("white", test.getChoices()[1]);
64     assertEquals("blue", test.getChoices()[2]);
65 } //end testGetKeywords
66
67
68 @Test
69 public void testCheckCorrectAnswer()
70 {
71     /* These are acceptable answers */
72     String a1 = "red";
73     String a2 = "WHITE";
74     String a3 = "A color is blue.";
75     String a4 = "Red is on the French flag";
76
77     assertEquals(true, test.checkCorrectAnswer(a1));
78     assertEquals(true, test.checkCorrectAnswer(a2));
79     assertEquals(true, test.checkCorrectAnswer(a3));
80     assertEquals(true, test.checkCorrectAnswer(a4));
81
82     /* These are unacceptable answers */
83     String a5 = "RRED is there!";
84     String a6 = "b lue";
85     String a7 = "White";
86     String a8 = " w@hite";
87
88     assertEquals(false, test.checkCorrectAnswer(a5));
89     assertEquals(false, test.checkCorrectAnswer(a6));
90     assertEquals(false, test.checkCorrectAnswer(a7));
91     assertEquals(false, test.checkCorrectAnswer(a8));
92 } //end testCheckCorrectAnswer
93
```

```
MultipleChoi... TrueFalseQu... ShortAnswer... MultipleChoi... AllTests.java TrueFalseQu... »6
35     test.setQuestion("The length of a year on earth is 365 days.");
36     test.setCorrectAnswer("F");
37 } //end setUp
38
39
40 @After
41 public void tearDown() throws Exception {}
42
43
44 @Test
45 public void testGetQuestion()
46 {
47     assertEquals("The length of a year on earth is 365 days.", test.getQuestion());
48 } //end testGetQuestion
49
50
51 @Test
52 public void testGetCorrectAnswer()
53 {
54     assertEquals("F", test.getCorrectAnswer());
55 } //end testGetCorrectAnswer
56
57
58 @Test
59 public void testCheckCorrectAnswer()
60 {
61     assertEquals(true, test.checkCorrectAnswer("f"));
62     assertEquals(true, test.checkCorrectAnswer("F"));
63     assertEquals(false, test.checkCorrectAnswer("false"));
64     assertEquals(false, test.checkCorrectAnswer("T"));
65 } //end testCheckCorrectAnswer
66
67 } //end TrueFalseQuestionTest
68
```