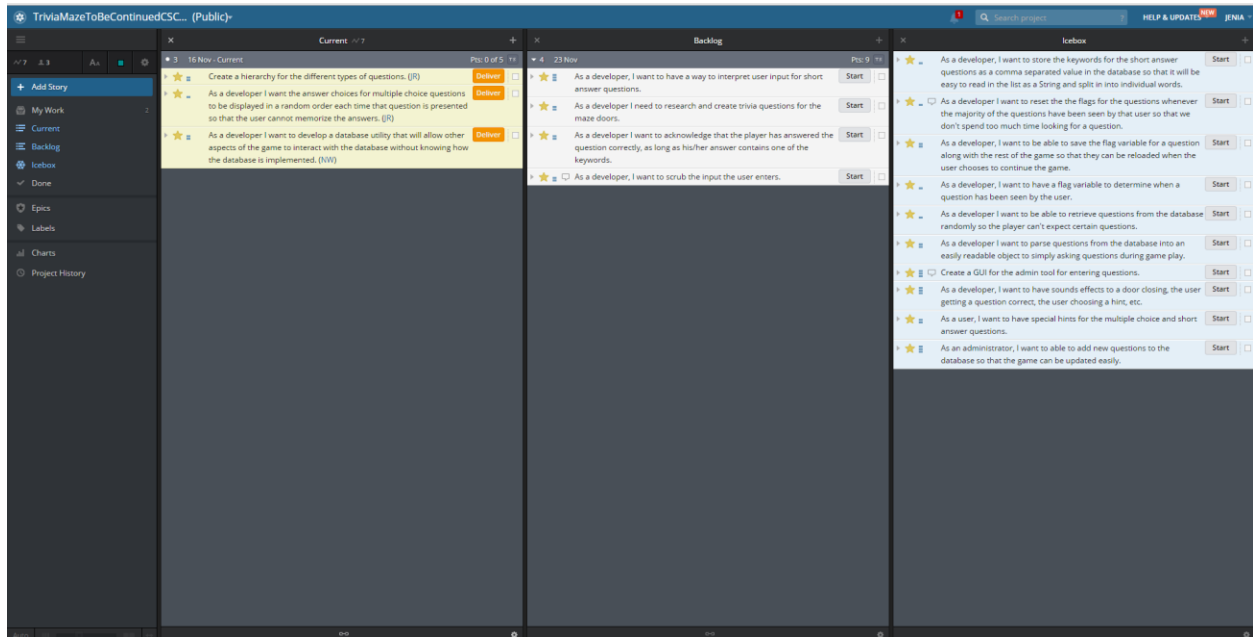


CSCD 350 Software Engineering

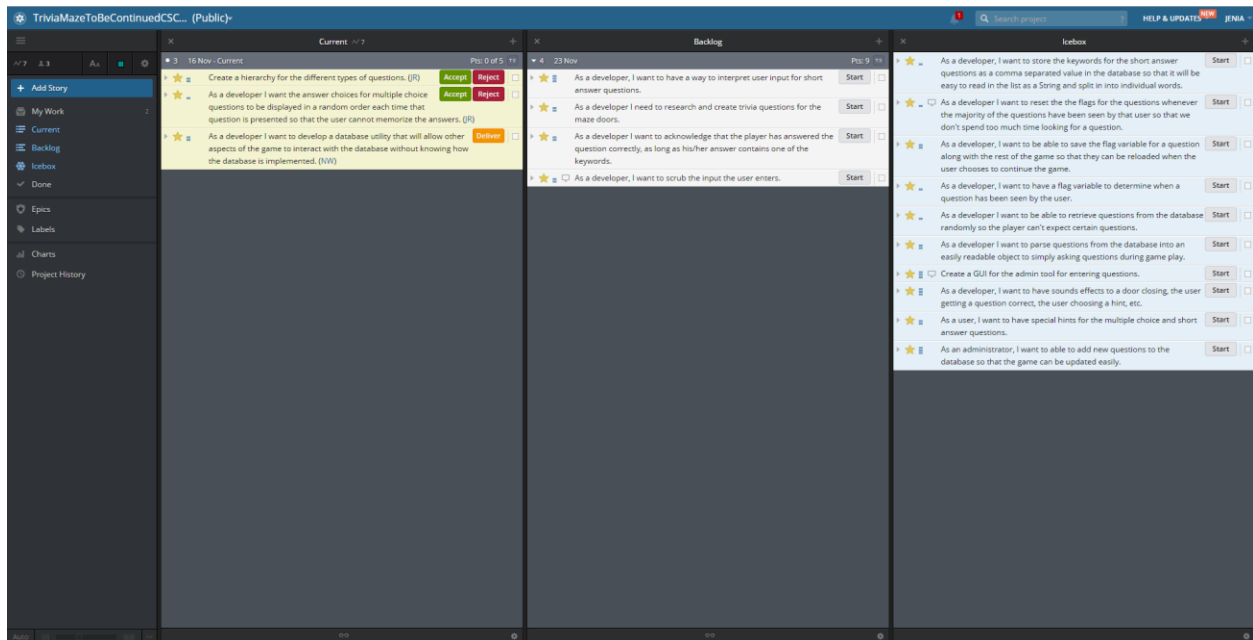
End of 2nd Iteration

Jenia's work for Iteration 2:

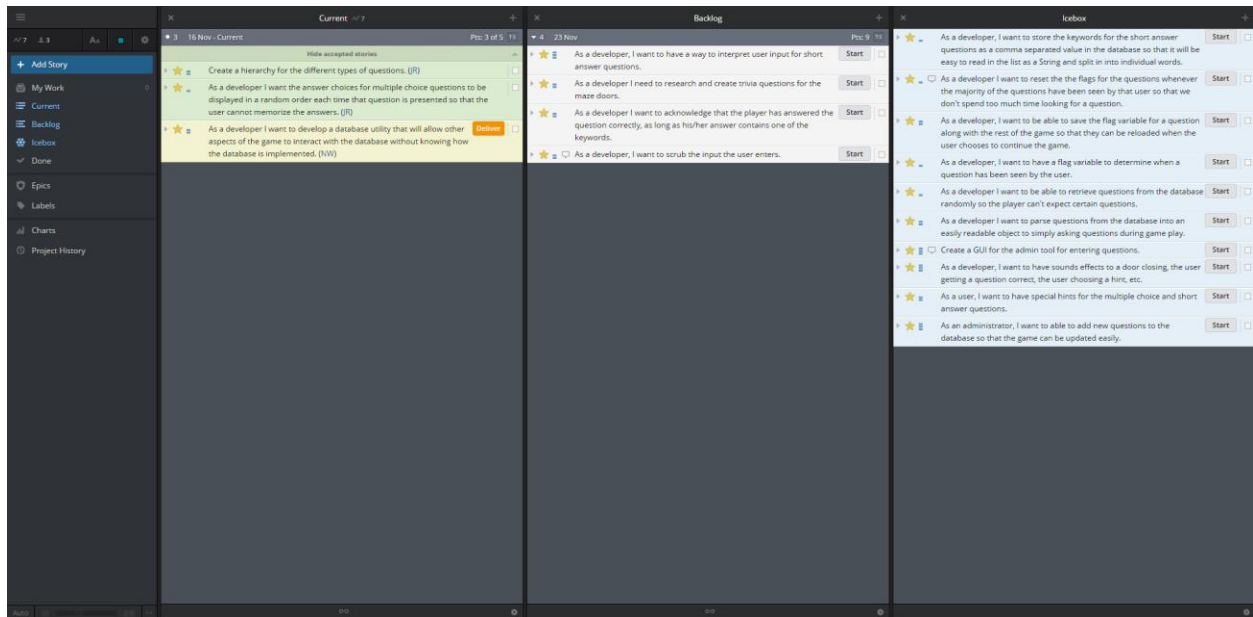
Finishing stories:



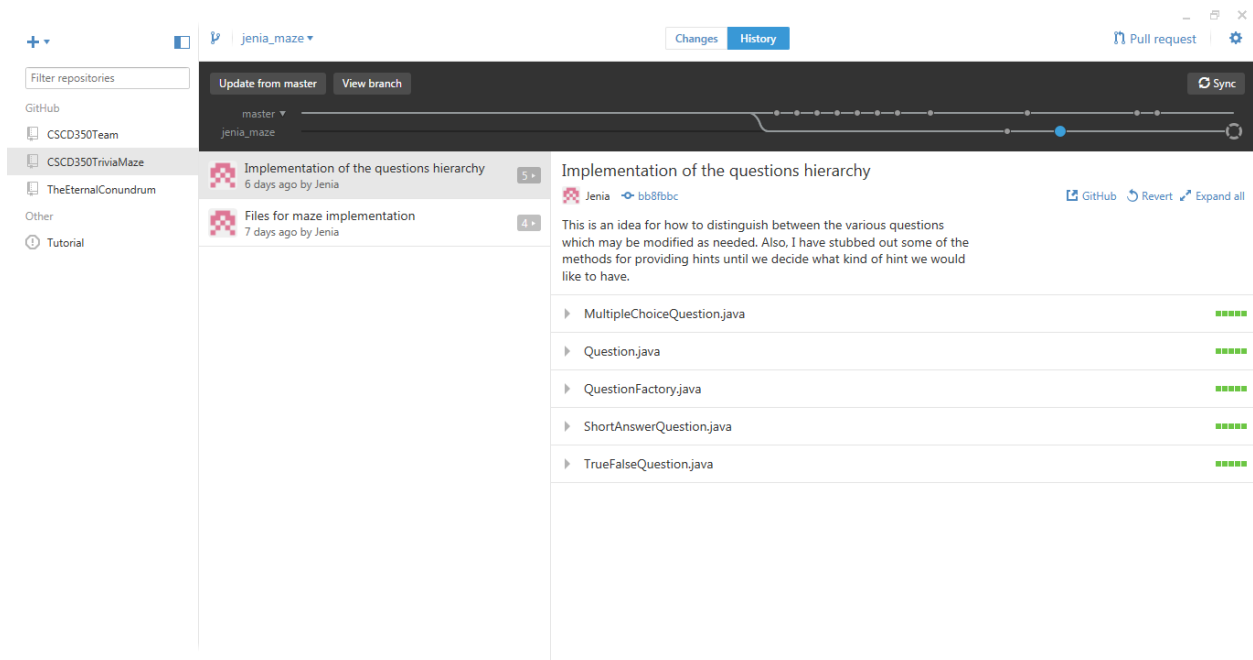
Delivering user stories:



Stories accepted:



Committing the changes to GitHub:



Code Snippets:

This screenshot shows a code editor interface for a project named 'jenia_maze'. The left sidebar displays a file tree with folders for 'GitHub', 'CSCD350Team', 'CSCD350TriviaMaze', 'TheEternalConundrum', 'Other', and 'Tutorial'. The main editor area is titled 'Implementation of the questions hierarchy' and shows the implementation of the 'MultipleChoiceQuestion' class. The code is as follows:

```
14 +
15 + public class MultipleChoiceQuestion extends Question
16 + {
17 +     private String[] choices; // a special object for an answer choice may be used here
18 +
19 +     public MultipleChoiceQuestion()
20 +     {
21 +         super(null, null);
22 +         super.setPattern("(?i)[abcd]");
23 +     }
24 +
25 +     public MultipleChoiceQuestion(String question, String correctAnswer, String[] choices)
26 +     {
27 +         super(question, correctAnswer);
28 +         this.choices = choices;
29 +         super.setPattern("(?i)[abcd]");
30 +     }
31 +
32 +     public String[] getChoices()
33 +     {
34 +         return choices;
35 +     }
36 +
37 +     public void setChoices(String[] choices)
38 +     {
39 +         this.choices = choices;
40 +     }
41 + }
```

This screenshot shows the same code editor interface, but now displaying the implementation of the 'Question' abstract class. The code is as follows:

```
5 + /**
6 +  * Question.java
7 +  * Author: Jenia Rousseva
8 +  * Revision: N/A
9 +  * Date: 11/09/2015
10 +  * This file provides the implementation of an abstract class for a generic
11 +  * question. All questions must have a question itself and a correct answer.
12 +  * The class also contains an abstract method called hint.
13 +  * Reference:
14 +  *
15 +  * https://docs.oracle.com/javase/tutorial/java/IandI/abstract.html
16 +  */
17 +
18 + public abstract class Question
19 + {
20 +     private String question;
21 +     private String correctAnswer;
22 +     private String pattern; // a regex pattern for input validation
23 +
24 +     public Question(String question, String correctAnswer)
25 +     {
26 +         this.question = question;
27 +         this.correctAnswer = correctAnswer;
28 +         this.pattern = "(.*)"; // matches anything
29 +     }
30 +
31 +     public String getQuestion()
32 +     {
```

+

▼

jenia_maze ▼

Changes

History

Pull request

⚙️

Filter repositories

GitHub

CSCD350Team

CSCD350TriviaMaze

TheEternalConundrum

Other

Tutorial

Update from master

View branch

Sync

master ▼

jenia_maze

Implementation of the questions hierarchy

5 +

6 days ago by Jenia

Files for maze implementation

4 +

7 days ago by Jenia

Implementation of the questions hierarchy

Jenia

bb8fbbc

GitHub

Revert

Collapse all

```

11 +
12 + public class QuestionFactory
13 + {
14 +     public QuestionFactory() {}
15 +
16 +     public Question createQuestion(String type) //may change the type of the parameter to
17 +     int or char
18 +     {
19 +         Question question = null;
20 +         if (type.equals("shortAnswer"))
21 +         {
22 +             question = new ShortAnswerQuestion();
23 +         }
24 +         else if (type.equals("multipleChoice"))
25 +         {
26 +             question = new MultipleChoiceQuestion();
27 +         }
28 +         else if (type.equals("trueFalse"))
29 +         {
30 +             question = new TrueFalseQuestion();
31 +         }
32 +         return question;
33 +     } //end createQuestion
34 +
35 + } //end QuestionFactory

```

+

▼

jenia_maze ▼

Changes

History

Pull request

⚙️

Filter repositories

GitHub

CSCD350Team

CSCD350TriviaMaze

TheEternalConundrum

Other

Tutorial

Update from master

View branch

Sync

master ▼

jenia_maze

Implementation of the questions hierarchy

5 +

6 days ago by Jenia

Files for maze implementation

4 +

7 days ago by Jenia

Implementation of the questions hierarchy

Jenia

bb8fbbc

GitHub

Revert

Collapse all

```

4 + * ShortAnswerQuestion.java
5 + * Author: Jenia Rousseva
6 + * Revision: N/A
7 + * Date: 11/09/2015
8 + * This file provides the implementation of a short answer question object.
9 + * These types of questions differ from other types in that a set of keyword(s)
10 + * will be used to match the user's answer to the correct answer.
11 + * Reference:
12 + *
13 + * https://docs.moodle.org/22/en/Regular_Expression_Short-Answer_question_type
14 + * https://docs.moodle.org/29/en/Short_answer_analysis
15 + */
16 +
17 + import java.util.Scanner;
18 +
19 + public class ShortAnswerQuestion extends Question
20 + {
21 +     String [] keywords; // acceptable answer choices; this can be changed to only one
22 +     keyword
23 +     String hint; // may use a special object
24 +
25 +     public ShortAnswerQuestion()
26 +     {
27 +         super(null, null);
28 +     }
29 +
30 +     public ShortAnswerQuestion(String question, String correctAnswer, String[] keywords)

```

jenia_maze

ChangesHistory

Pull request

Filter repositories

GitHub

CSCD350Team

CSCD350TriviaMaze

TheEternalConundrum

Other

Tutorial

Update from masterView branch

Sync

masterjenia_maze

Implementation of the questions hierarchy
6 days ago by Jenia

Files for maze implementation
7 days ago by Jenia

Implementation of the questions hierarchy

Jenia

bb8fbbc

GitHubRevertCollapse all

```
15 +
16 + public class TrueFalseQuestion extends Question
17 + {
18 +     public TrueFalseQuestion()
19 +     {
20 +         super(null, null);
21 +         super.setPattern("(?i){tf}");
22 +     }
23 +
24 +     public TrueFalseQuestion(String question, String correctAnswer)
25 +     {
26 +         super(question, correctAnswer);
27 +         super.setPattern("(?i){tf}");
28 +     }
29 +
30 +     @Override
31 +     public void printQuestion()
32 +     {
33 +         super.printQuestion();
34 +         System.out.println("\n\nChoose T or F:\n");
35 +     }
36 +
37 +     @Override
38 +     public void hint() // cannot have hints here
39 +     {
40 +         // TODO Auto-generated method stub
41 +     }
42 +
```