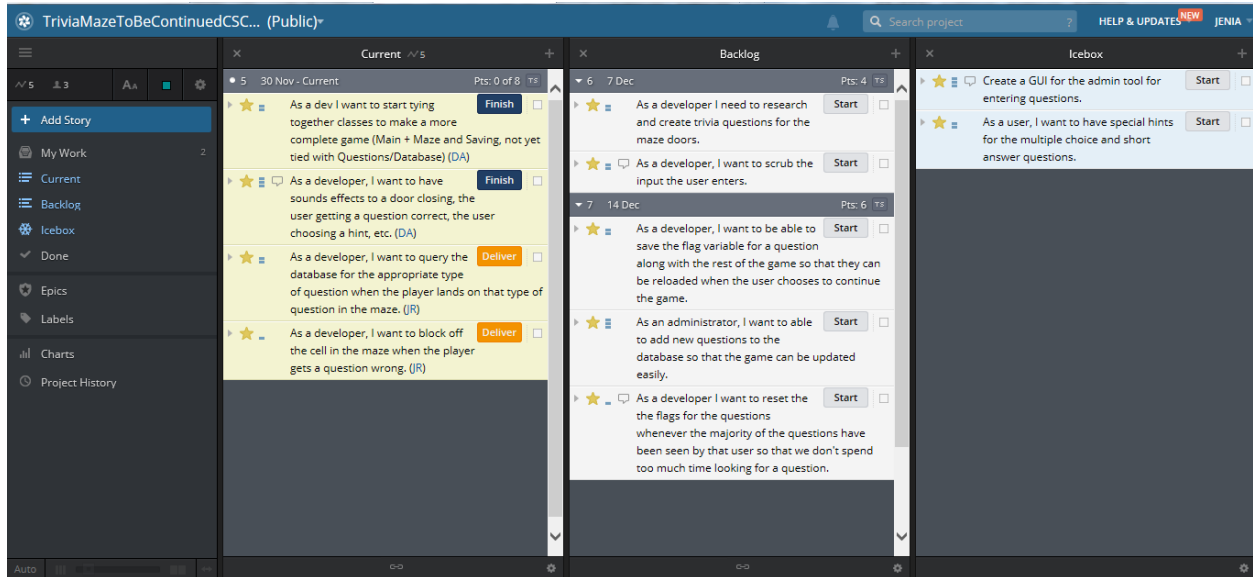


CSCD 350 Software Engineering

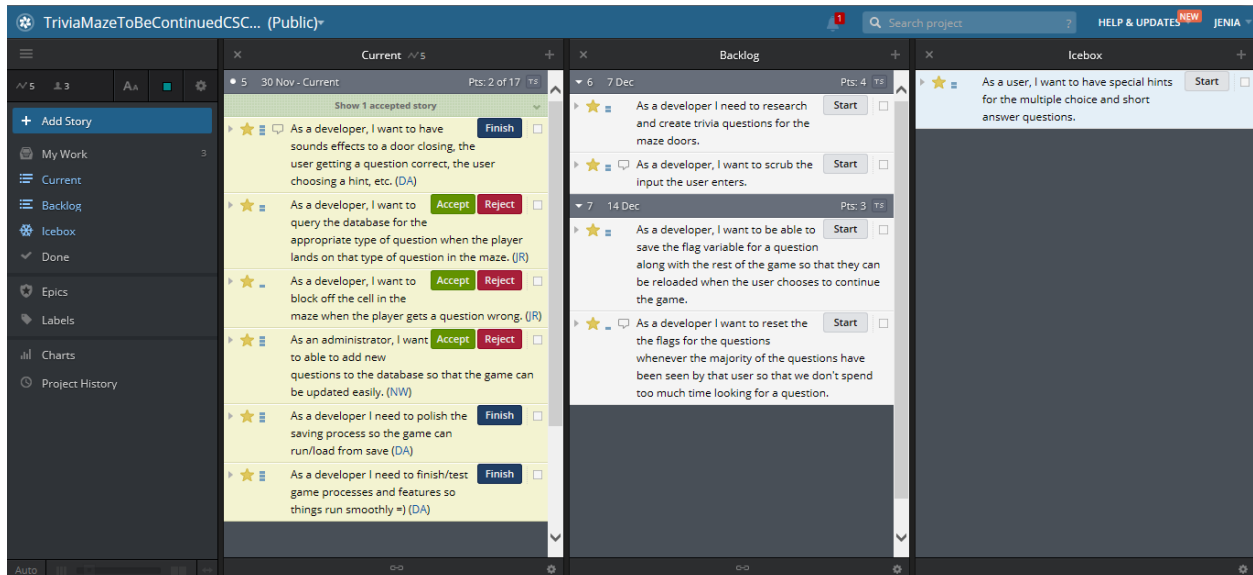
End of 4th Iteration

Jenia's work for Iteration 4:

Delivering user stories:



Stories accepted:



Committing the changes to GitHub:

The screenshot shows the GitHub web interface for the 'jenia_maze' repository. The commit history table lists the following commits:

Commit Message	Author	Time	Commits
Added QuestionHandler class for getting questions from the database	Jenia	just now	7
Update to the Maze	Jenia	9 days ago	4
Updates to the questions hierarchy	Jenia	9 days ago	10
Implementation of the questions hierarchy	Jenia	20 days ago	5
Files for maze implementation	Jenia	21 days ago	4

The detailed view of the latest commit shows the following files:

- DatabaseTests.java
- DatabaseUtility.java
- ErrorQuestion.java
- Maze.java
- Question.java
- QuestionFactory.java
- QuestionHandler.java

Code Snippets:

```
CellType.java Ellers.java MazeTester.java QuestionFac... ErrorQuesti... DatabaseUtil... DatabaseTes... Maze.java QuestionHan...
11 public class QuestionHandler
12 {
13     /*
14     * Retrieves a question of a the given type from the database.
15     * Parameters:
16     * String type - a description of the type of question needed
17     * DatabaseUtility db - an object that connects and retrieves questions
18     * from the database
19     * Returns:
20     * Question - either a true/false, multiple choice, or short answer question
21     */
22
23     public Question getQuestionFromDB(String type, DatabaseUtility db)
24     {
25         boolean validQuestion = false;
26         Question question;
27
28         do
29         {
30             question = db.retrieveQuestion(type);
31             if (question.getError() == false)
32                 validQuestion = true;
33         } while (!validQuestion);
34
35         return question;
36     }
37     //end getQuestionFromDB
38
39     /*
40     * This method displays the question to the player, obtains an answer,
41     * checks the answer, and then updates the maze.
42     * Parameters:
43     * Question question - either a true/false, multiple choice, or short answer question
44     * Maze maze - a maze
45     */
46
47     public void handleQuestion(Question question, Maze maze)
48     {
49     }
```

```

MazeTester.java QuestionFac... ErrorQuesti... DatabaseUtil... DatabaseTes... Maze.java »11
386     } //end else
387 } //end verifyDirection
388
389
390 /*
391  * Either open up or block off the path position in the maze depending on
392  * if the player answered the question correctly or not.
393  * Parameter:
394  * boolean success - If the user answered the question correctly
395  */
396
397 public void postUpdate(boolean success)
398 {
399     if (success)
400     {
401         maze[this.curRow][this.curCol] = CellType.VISITED;
402     } //end if
403     else
404     {
405         maze[this.curRow][this.curCol] = CellType.WALL;
406         setPosition(this.prevRow, this.prevCol); // Go back to the previous position
407     } //end else
408 } //end postUpdate
409
410
411 /*
412  * Checks if a path exists to the goal position from the current position.
413  * If not, we want to end the current game.
414  */
415
416 public boolean pathExists()
417 {
418
419     CellType[][] mazeCopy = createCopy(this.maze);

```

```

CellType.java Ellers.java MazeTester.java QuestionFac... ErrorQuesti... DatabaseUtil... DatabaseTes... Maze.java » QuestionHan... »8
91
92 /*
93  * Find the type of question to obtain depending on the value
94  * at the given position.
95  * This method is to be called if getLandOnQuestion returns true;
96  * Returns:
97  * String - A String indicator for the type of question
98  */
99
100 public String getQuestionType()
101 {
102     CellType ques = this.maze[this.curRow][this.curCol];
103     if (ques == CellType.TFQUESTION)
104         return "truefalse";
105     else if (ques == CellType.MCQUESTION)
106         return "multiplechoice";
107     else if (ques == CellType.SAQUESTION)
108         return "shortanswer";
109     else
110         return "error";
111 } //end getQuestionType
112
113
114 /*
115  * Returns a boolean value indicating whether the current position is a
116  * question space or not. If this method calls true, we want to call
117  * getQuestionType.
118  * Returns:
119  * boolean - true if the current space is one of the three types of questions
120  */
121
122 public boolean getLandOnQuestion()
123 {
124     return (this.maze[this.curRow][this.curCol] == CellType.TFQUESTION ||
125         this.maze[this.curRow][this.curCol] == CellType.MCQUESTION ||
126         this.maze[this.curRow][this.curCol] == CellType.SAQUESTION);
127 } //end getLandOnQuestion
128

```

17 commits
 5 branches
 0 releases
 3 contributors

Your recently pushed branches:

davidBranch (1 minute ago)
 [Compare & pull request](#)

Branch: davidBranch ▾
 CSCD350TriviaMaze / +

This branch is 9 commits ahead, 11 commits behind master.
 [Pull request](#)
[Compare](#)

David2Walker Saving maze + functional fixes
 Latest commit d3f9a17 a minute ago

.idea	Saving maze + functional fixes	a minute ago
out/production/CSCD350TriviaMaze	Saving maze + functional fixes	a minute ago
CSCD350TriviaMaze.iml	merged to work with intellij (hopefully)	15 days ago
CellType.java	Added Maze Files	8 days ago
Ellers.java	Added Maze Files	8 days ago
Item.java	Added Trivial Util + save feature	15 days ago
Maze.java	Saving maze + functional fixes	a minute ago
MazeTester.java	Added Maze Files	8 days ago
Player.java	Saving maze + functional fixes	a minute ago
README.md	Revert "Revert "Initial commit""	24 days ago
TriviaMain.java	Saving maze + functional fixes	a minute ago
saved.ser	Saving maze + functional fixes	a minute ago
triviaUtil.java	Saving maze + functional fixes	a minute ago
triviaquestions.txt	engine framework + questions txt	17 hours ago

Example run of starting up the game and saving the maze + player info

"C:\Program Files (x86)\Java\jdk1.8.0_40\bin\java"

What would you like to do?

1) Load saved game

2) Start a new game

2

Before Find Path

W W W W W W W W W

W W W W W W W W W

W S O W O W O G W

W W O W O W O W W

W W O O O O O W W

W W W W O W O W W

W W O O O W O W W
W W W W W W W W W
W W W W W W W W W

BEFORE setting the final maze!

W W W W W W W W W
W W O W W W O W W
W S . W # W . G W
W O . O # O . W W
W O W W
W W O O # W O W W
W W # # # W O W W
W W W W W W W W W
W W W W W W W W W

After setting the final maze!

W W W W W W W W W
W W O W W W 2 W W
W S 3 W 3 W O G W
W O 3 1 O 1 O W W
W O 2 3 3 1 O W W
W W O 3 3 W 1 W W
W W O 2 2 W 1 W W
W W W W W W W W W
W W W W W W W W W

curRow = 2 curCol = 1

Total number of possible paths to the end: 364

W W W W W W W W W
W W O W W W 2 W W
W S 3 W 3 W O G W
W O 3 1 O 1 O W W
W O 2 3 3 1 O W W
W W O 3 3 W 1 W W
W W O 2 2 W 1 W W
W W W W W W W W W
W W W W W W W W W

What's your name?

David

What would you like to do?

- 1) Traverse Maze
- 2) Save Game
- 3) Quit without saving

2

Would you like to overwrite your save? (Y/N)

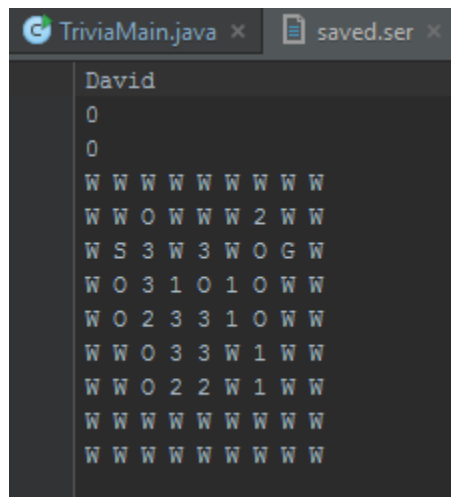
Y

What would you like to do?

- 1) Traverse Maze
 - 2) Save Game
 - 3) Quit without saving
- 3

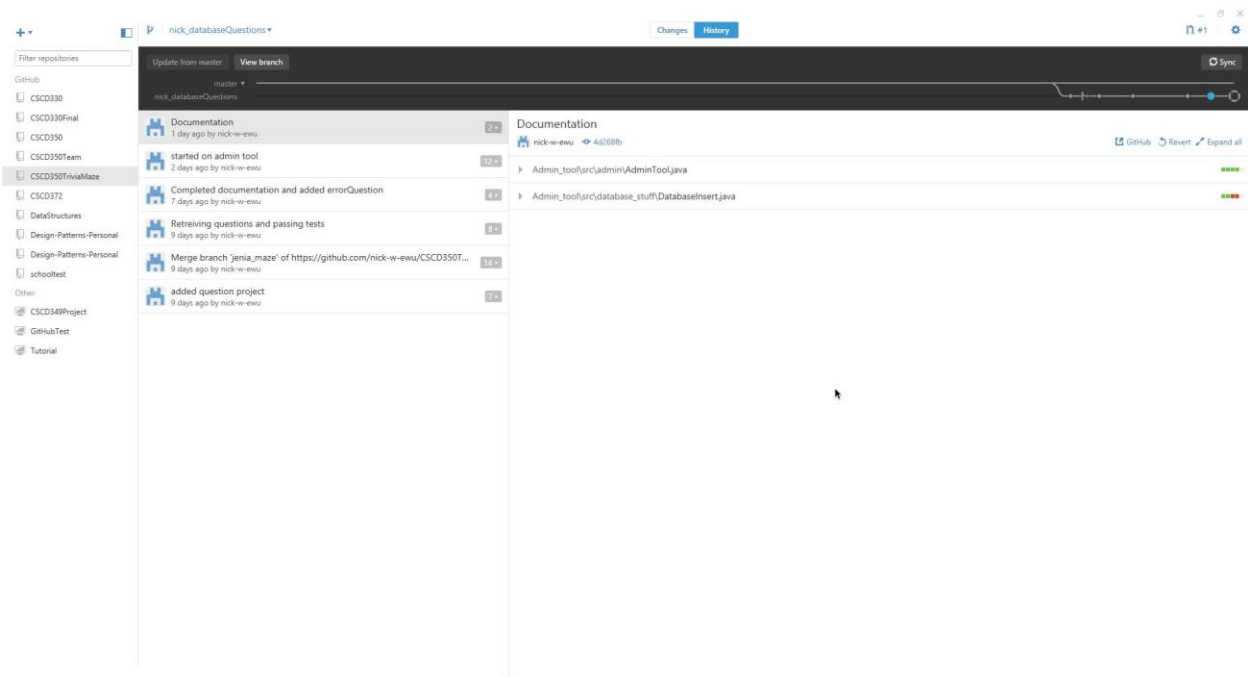
Process finished with exit code 0

Picture of the saved .ser file

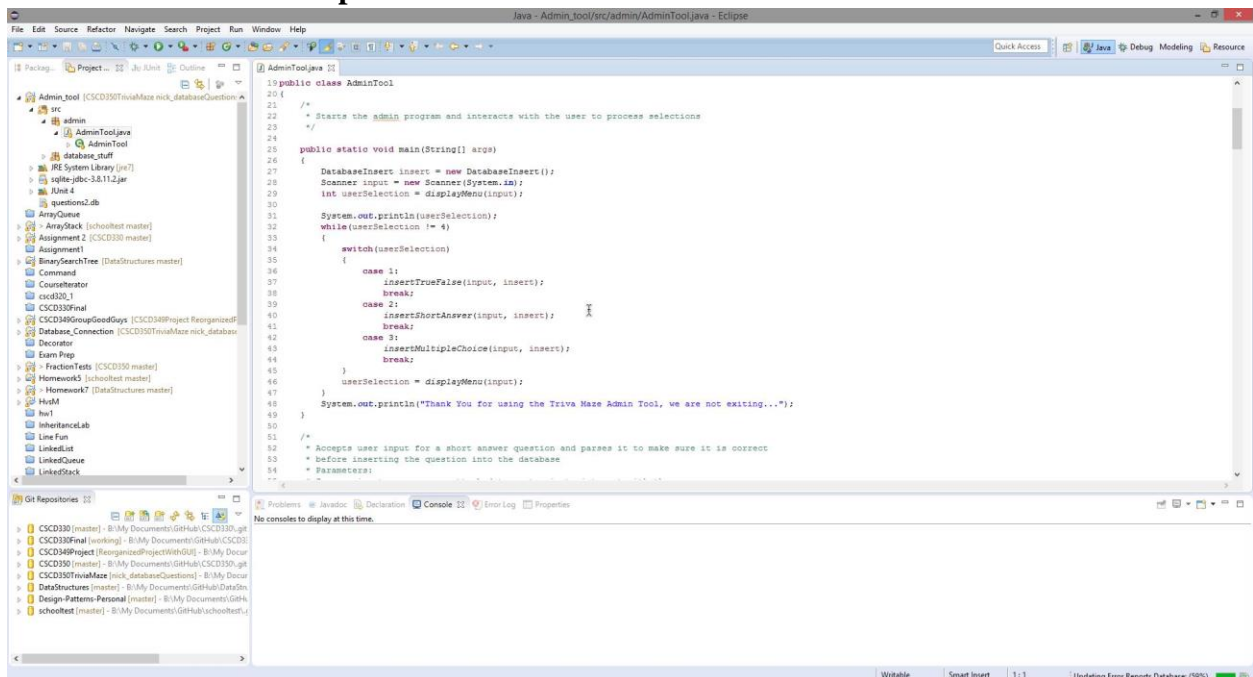


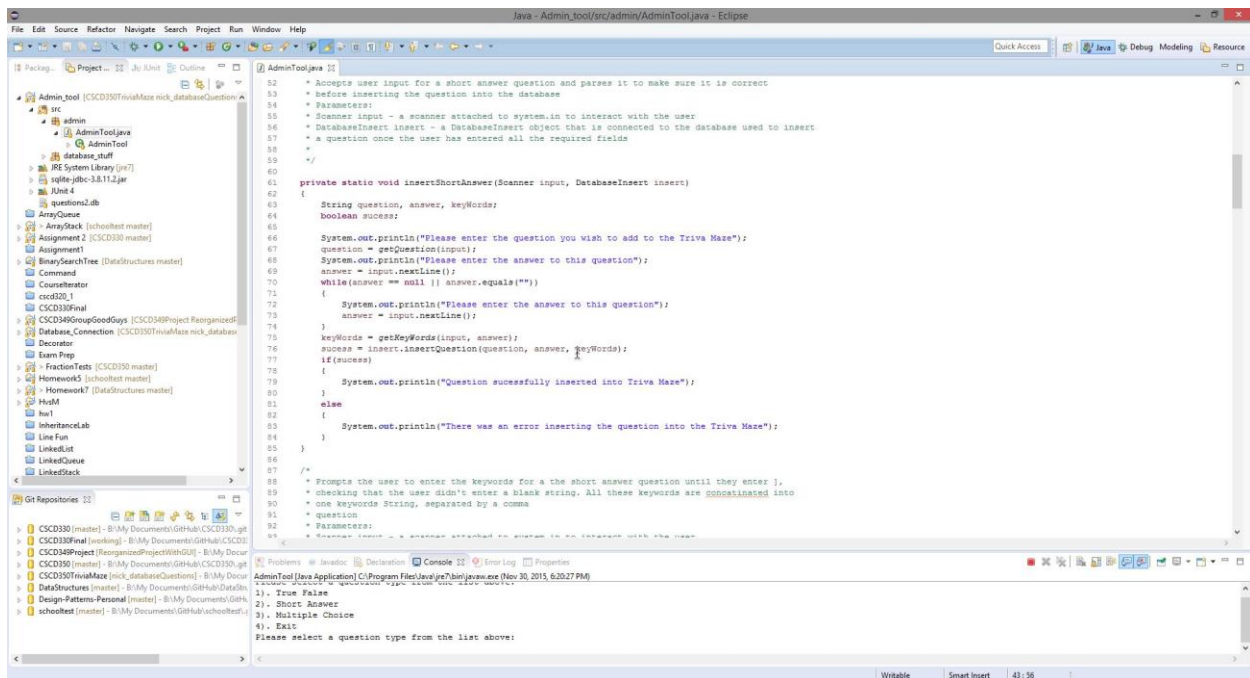
Nick's 4th Iteration

GitHub Branch



Admin Tool Code Examples





Admin Tool Running, Main Menu



Pivotal Tracker, Admin Tool Story Finished

The screenshot displays the Pivotal Tracker interface for a project named "TriviaMazeToBeContinuedCSC... (Public)". The interface is divided into three main columns: "Current", "Backlog", and "Inbox".

Current Column: This column contains five user stories, each with a star icon, a description, and a "Finish" button. The stories are:

- As a developer, I want to have sounds effects to a door closing, the user getting a question correct, the user choosing a hint, etc. (DA)
- As a developer, I want to query the database for the appropriate type of question when the player lands on that type of question in the maze. (R)
- As a developer, I want to block off the cell in the maze when the player gets a question wrong. (R)
- As an administrator, I want to be able to add new questions to the database so that the game can be updated easily. (NW)
- As a developer I need to polish the saving process so the game can run/load from save (DA)

Backlog Column: This column contains four user stories, each with a star icon, a description, and a "Start" button. The stories are:

- As a developer I need to research and create trivia questions for the maze doors.
- As a developer, I want to scrub the input the user enters.
- As a developer, I want to be able to save the flag variable for a question along with the rest of the game so that they can be reloaded when the user chooses to continue the game.
- As a developer I want to reset the flags for the questions whenever the majority of the questions have been seen by that user so that we don't spend too much time looking for a question.

Inbox Column: This column contains one user story with a star icon, a description, and a "Start" button:

- As a user, I want to have special hints for the multiple choice and short answer questions.

The interface also includes a sidebar on the left with navigation options: "Add Story", "My Work", "Current", "Backlog", "Inbox", "Done", "Epics", "Labels", "Charts", and "Project History". The top bar shows the project name, a search bar, and a "HELP & UPDATES" link.