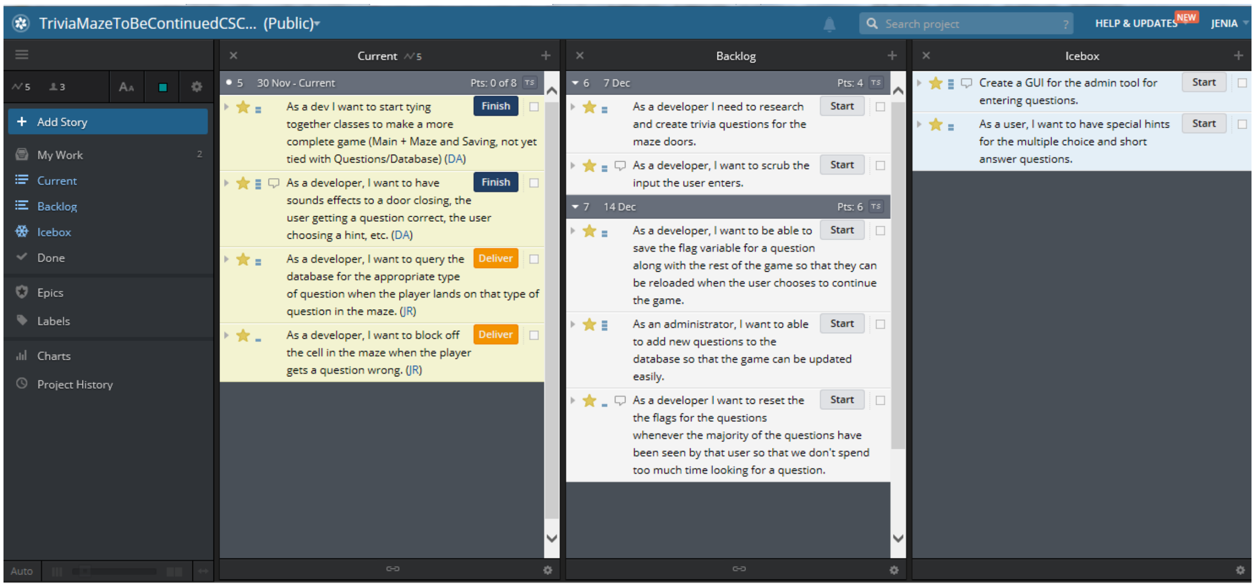
CSCD 350 Software Engineering

End of 4th Iteration

Jenia's work for Iteration 4:

Delivering user stories:



Current 5

5 30 Nov - Current Pts: 0 of 8

- As a dev I want to start tying together classes to make a more complete game (Main + Maze and Saving, not yet tied with Questions/Database) (DA) Finish
- 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) Finish
- 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. (JR) Deliver
- As a developer, I want to block off the cell in the maze when the player gets a question wrong. (JR) Deliver

Backlog

6 7 Dec Pts: 4

- As a developer I need to research and create trivia questions for the maze doors. Start
- As a developer, I want to scrub the input the user enters. Start

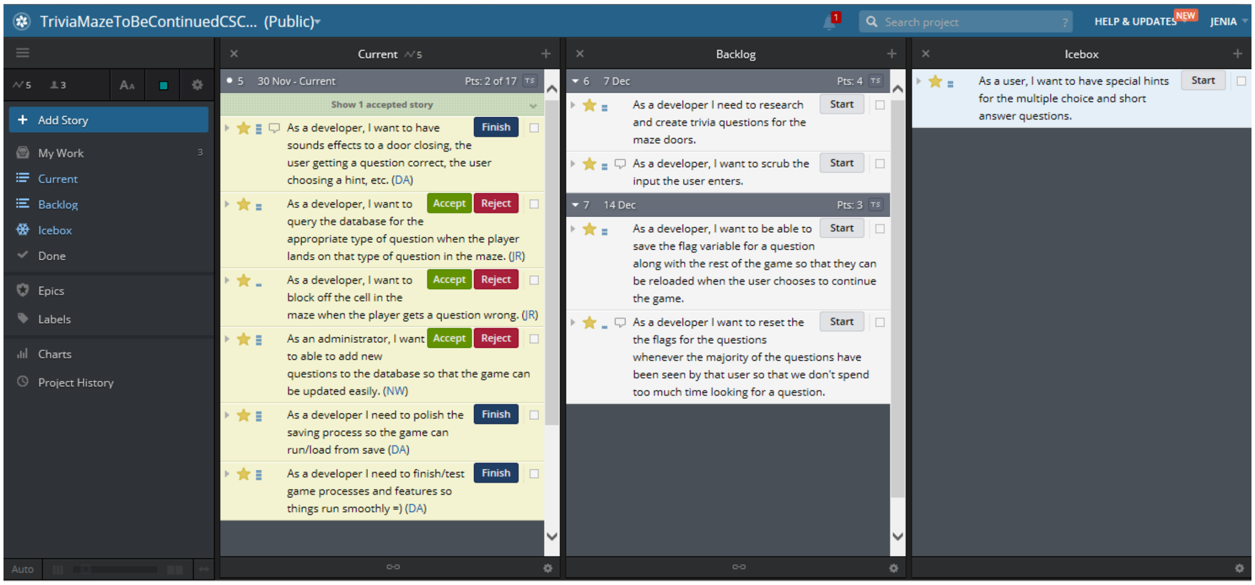
7 14 Dec Pts: 6

- 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. Start
- As an administrator, I want to able to add new questions to the database so that the game can be updated easily. Start
- As a developer I want to reset the 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. Start

Icebox

- Create a GUI for the admin tool for entering questions. Start
- As a user, I want to have special hints for the multiple choice and short answer questions. Start

Stories accepted:



Current 5

5 30 Nov - Current Pts: 2 of 17

Show 1 accepted story

- 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) Finish
- 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. (JR) Accept Reject
- As a developer, I want to block off the cell in the maze when the player gets a question wrong. (JR) Accept Reject
- As an administrator, I want to able to add new questions to the database so that the game can be updated easily. (NW) Accept Reject
- As a developer I need to polish the saving process so the game can run/load from save (DA) Finish
- As a developer I need to finish/test game processes and features so things run smoothly =) (DA) Finish

Backlog

6 7 Dec Pts: 4

- As a developer I need to research and create trivia questions for the maze doors. Start
- As a developer, I want to scrub the input the user enters. Start

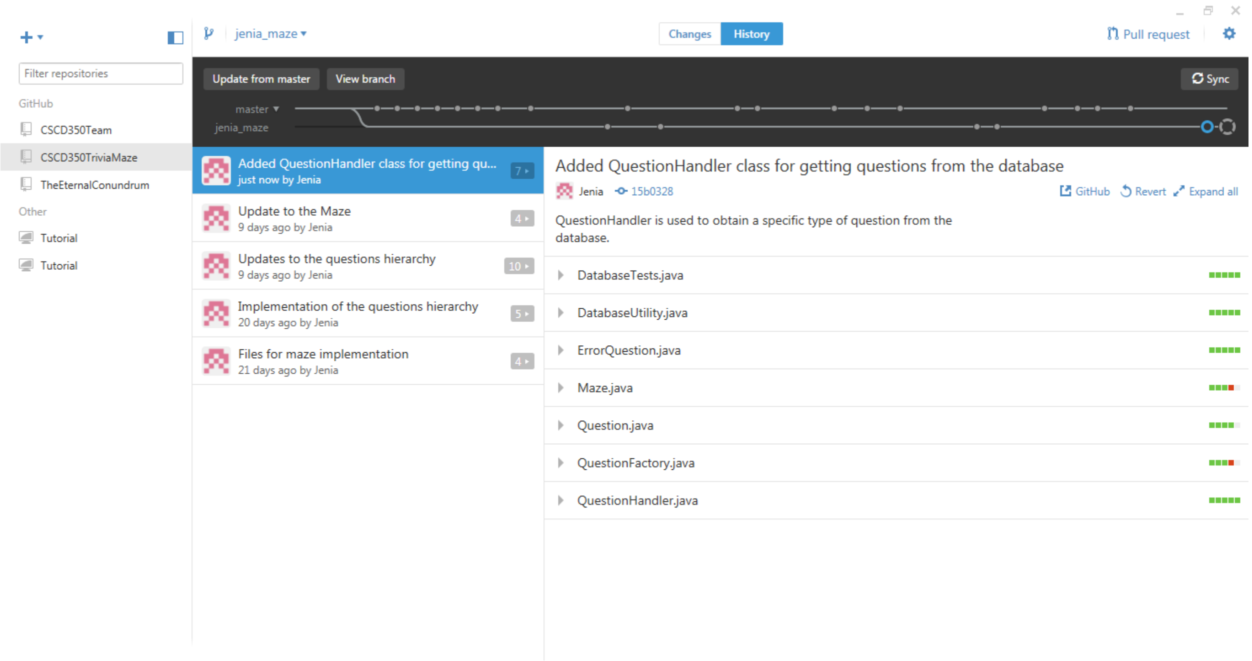
7 14 Dec Pts: 3

- 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. Start
- As a developer I want to reset the 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. Start

Icebox

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

Committing the changes to GitHub:



Added QuestionHandler class for getting qu... 7
just now by Jenia

Update to the Maze 4
9 days ago by Jenia

Updates to the questions hierarchy 10
9 days ago by Jenia

Implementation of the questions hierarchy 5
20 days ago by Jenia

Files for maze implementation 4
21 days ago by Jenia

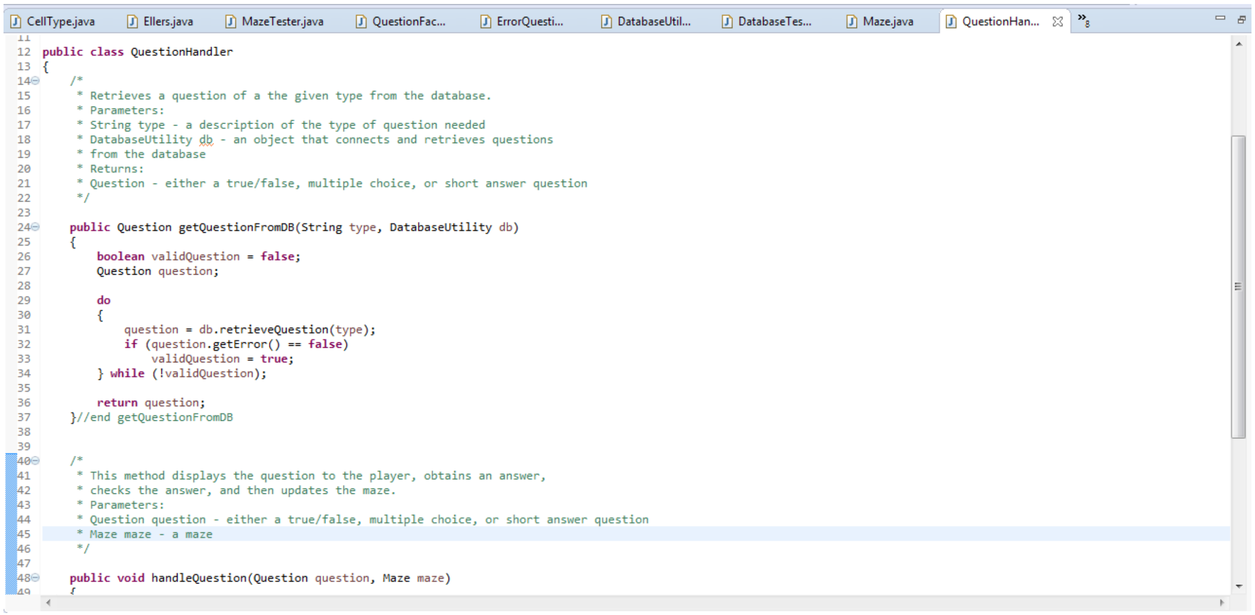
Added QuestionHandler class for getting questions from the database

Jenia 15b0328

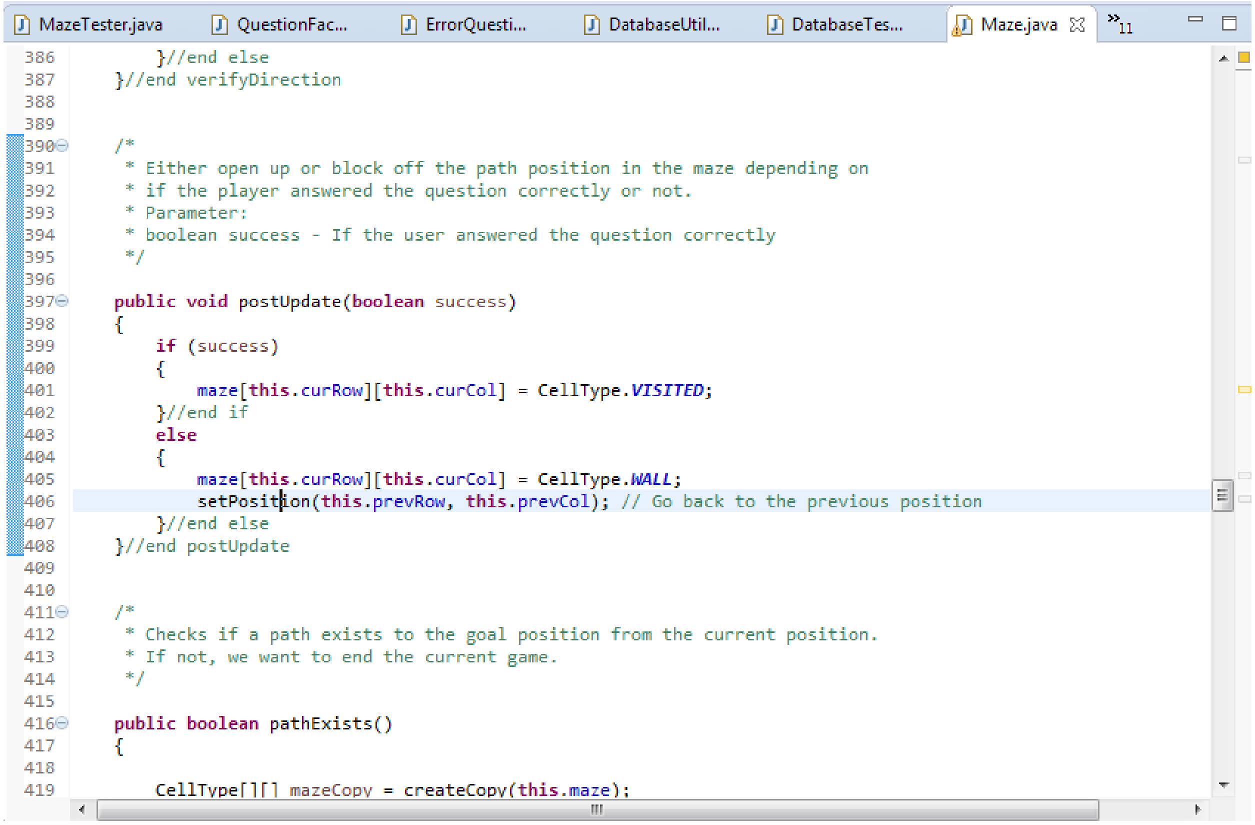
QuestionHandler is used to obtain a specific type of question from the database.

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

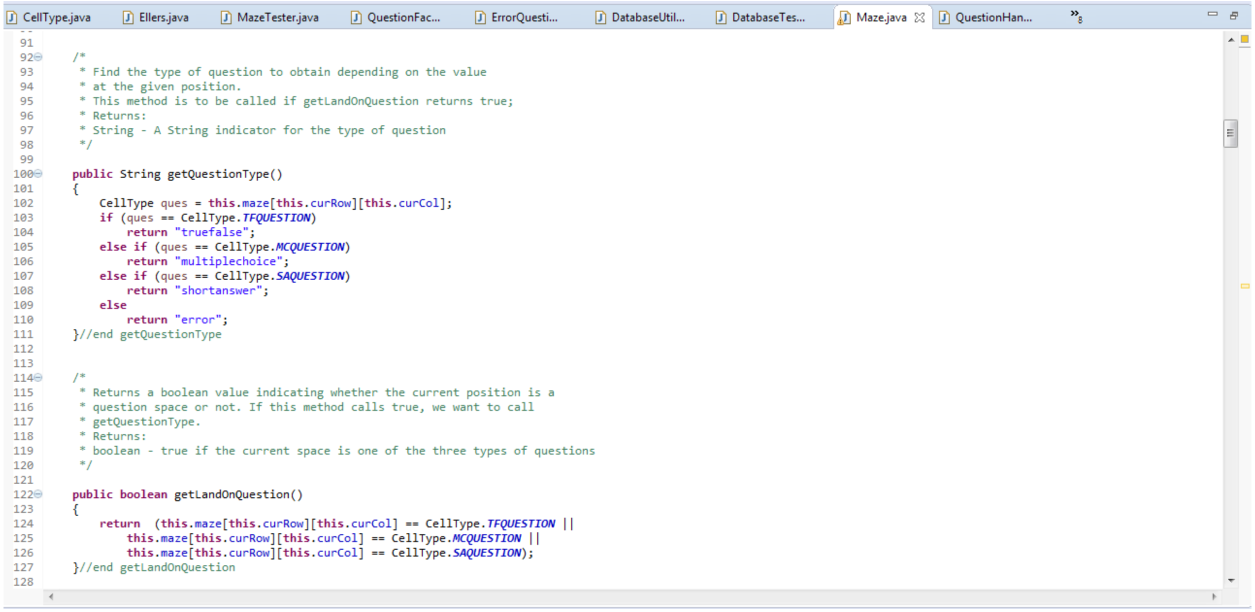
Code Snippets:



```
public class QuestionHandler
{
    /*
     * Retrieves a question of a the given type from the database.
     * Parameters:
     * String type - a description of the type of question needed
     * DatabaseUtility db - an object that connects and retrieves questions
     * from the database
     * Returns:
     * Question - either a true/false, multiple choice, or short answer question
     */

    public Question getQuestionFromDB(String type, DatabaseUtility db)
    {
        boolean validQuestion = false;
        Question question;

        do
        {
            question = db.retrieveQuestion(type);
            if (question.getError() == false)
                validQuestion = true;
        } while (!validQuestion);

        return question;
    }//end getQuestionFromDB


    /*
     * This method displays the question to the player, obtains an answer,
     * checks the answer, and then updates the maze.
     * Parameters:
     * Question question - either a true/false, multiple choice, or short answer question
     * Maze maze - a maze
     */

    public void handleQuestion(Question question, Maze maze)
    {
```



```java
        }//end else
    }//end verifyDirection


    /*
     * Either open up or block off the path position in the maze depending on
     * if the player answered the question correctly or not.
     * Parameter:
     * boolean success - If the user answered the question correctly
     */

    public void postUpdate(boolean success)
    {
        if (success)
        {
            maze[this.curRow][this.curCol] = CellType.VISITED;
        }//end if
        else
        {
            maze[this.curRow][this.curCol] = CellType.WALL;
            setPosition(this.prevRow, this.prevCol); // Go back to the previous position
        }//end else
    }//end postUpdate


    /*
     * Checks if a path exists to the goal position from the current position.
     * If not, we want to end the current game.
     */

    public boolean pathExists()
    {

        CellType[][] mazeCopy = createCopy(this.maze);
```



```java

    /*
     * Find the type of question to obtain depending on the value
     * at the given position.
     * This method is to be called if getLandOnQuestion returns true;
     * Returns:
     * String - A String indicator for the type of question
     */

    public String getQuestionType()
    {
        CellType ques = this.maze[this.curRow][this.curCol];
        if (ques == CellType.TFQUESTION)
            return "truefalse";
        else if (ques == CellType.MCQUESTION)
            return "multiplechoice";
        else if (ques == CellType.SAQUESTION)
            return "shortanswer";
        else
            return "error";
    }//end getQuestionType


    /*
     * Returns a boolean value indicating whether the current position is a
     * question space or not. If this method calls true, we want to call
     * getQuestionType.
     * Returns:
     * boolean - true if the current space is one of the three types of questions
     */

    public boolean getLandOnQuestion()
    {
        return  (this.maze[this.curRow][this.curCol] == CellType.TFQUESTION ||
            this.maze[this.curRow][this.curCol] == CellType.MCQUESTION ||
            this.maze[this.curRow][this.curCol] == CellType.SAQUESTION);
    }//end getLandOnQuestion

```