## **Criterion A: Planning**

**Description of the Scenario**

My friend, Yubo Su, often plays Tic-Tac-Toe with me on free times. As we play the game, Yubo found that the game is not fair: due to the lack of variation in the game, the first play will not lose if he plays in a certain way, no matter what the second player does. Because we used to play this game on pens and paper, there isn’t a record of the results, so we already forget who win more times. To have better playing experience, Yubo wants to have a more complicated Tic-Tac-Toe with a recording for the result of each game (see appendices 3). In the discussion, Yubo suggested various ways to modify Tic-Tac-Toe, such as extending the board, making it be a three-player game, or adding more layers to make it three-dimensional. After evaluating the feasibility of each scheme, we decided to modify the game to be a 3D version of Tic-Tac-Toe on computers. Because Yubo has no programming ability, he asked me to work on the design and programming part, and Yubo will focus on testing. My adviser, Mrs Anderson, teaches me Computer Science and has enough knowledge and experience to guide me.

(196)

**Rationale for the Proposed Product**

Commonly the Tic-Tac-Toe is played on paper if it is in 2D, but converting to 3D, drawing on paper is not efficient because one piece of paper cannot show all the surfaces of the cube. The 3D model would be able to be turned and clicked. And the database can automatically record the winner and loser after each round and create a rank. So the modifying should be a computer application.

The multiple layers will be represented as a cube divided into 27(3x3x3) cells. Players could select the cells alternatively. Once one player occupies all cells in one line, that player wins this game.

To achieve Yubo’s requirements, I used the following technologies to develop the product:

* Graphical User Interface (GUI) features in Java FX allow interactions with the end-users directly. Features like labels, TextFields, and buttons, the client can see and operate the product directly without seeing the code or entering a command. Mouse/touch board listener and keyboard listener can be added into the GUI to fulfil different end-users’ preferences. Thus, using GUI would help the client to identify the visual elements in the design phase and testing phase and ensure that the game is user-friendly.
* Java has object-oriented programming (OOP) features which allow for organizing the logic relationship of functions and classes in the product.
* The algorithm created by Java classes can be connected with the GUI, which will judge the process of the game. Several objects (cells and board) would be organized as individual classes to set and store some characteristics.
* Java DB provides a database which can store and sort user information (may include username, password, game points, and rank), which allow different end-users to play this game with their performance recorded separately.
* The developer (me) can get all features (OOP, Java DB and Java FX) he needs from NetBeans.

(306)

**Criteria for Success**

A successful product should have these criteria:

* Multiple scenes to show GUI elements

Visual design and user’s interaction with the GUI: Users can log in, sign-up, play, and see the output (prompts, movements, and game results) on the GUI

Users are allowed to switch between the scenes

(between the login scene and sign-up scene, from the login scene to the game scene, from the game scene to the resulting scene, and quit result scene) and quit the application using buttons:

Enter / Sign-up / Back / Next / Quit

* Multiple accounts can be stored

Registration: users can create accounts if they don’t have one using usernames and passwords

Authorization: Only users with an account are allowed to play the game (the account is for ranking), to make sure that the ranking is correct and to prevent cheating

Updating: Updating: Each account will have a playing history in the database, where wins and losses can be updated after each round of game played

Sorting: A rank of the top five players could be shown in the result page

* A 3D shape of the cubes

Mouse and keyboard action listeners: keyboard typing and mouse motion, dragging, scrolling, and clicking; all the cells can be selected

All the cells can be selected

* User-friendly

Multiple ways to turn and zoom the cubes

The property of each cell is clear (who took this cell)

The indication of selected cells of cubelet is clear (who took this cell) A See-through diagram to help users to understand the rule is provided, which can be a 2D representation of the three layers, and the actions on the cubes are also shown on the diagram

* Checking algorithm

A winner should be identified when the same person occupies three cells in a line. The three cells in a line could be horizontal, vertical, and diagonal in the same layer or across the layers