CS112 - Final Project

Fall 2015

Brian Nguyen

December 1, 2015

Contents

[Introduction 3](#_Toc436779297)

[Program Design 3](#_Toc436779298)

[GUI 3](#_Toc436779299)

[Program Features 4](#_Toc436779300)

[Computer vs. Computer 4](#_Toc436779301)

[Program Considerations 4](#_Toc436779302)

[Game Engine 4](#_Toc436779303)

[Simple\Smart 4](#_Toc436779304)

[Testing 4](#_Toc436779305)

[Debugging 4](#_Toc436779306)

[Exception Handling 4](#_Toc436779307)

[Main Flow 4](#_Toc436779308)

[Pseudo Code 5](#_Toc436779309)

[Program Details 5](#_Toc436779310)

[Classes 5](#_Toc436779311)

[OO methodologies 5](#_Toc436779312)

[Encapsulation 5](#_Toc436779313)

[Polymorphism 5](#_Toc436779314)

[Inheritance 6](#_Toc436779315)

[Constructor 6](#_Toc436779316)

# Introduction

The objectives of this project are to implements a tic-tac-toe game that will demonstrate the Object Oriented Technologies using Java.

# Program Design

## GUI

The GUI I will be using involves using Jpanels and ActionListeners in order to allow for smooth player interactions with the game board that will be made 300x300 pixels. The grid layout will involve 3 columns and 3 rows in which the first move will always be an x and the second move will always be an o.

-Console

My console will be managed in eclipse mars. Ideally the consoles should maintain the user input data and translate it to the proper array. The array will be managed in its own class and return the data through the main class of the program.

-Graphics

The textbox will display the frame of a 3x3 matrix taking in char ‘x’ or char ‘o’ using the the javaxswing to create borders and a jpanel to represent the matrixes using the grid layout function. The mouse event will draw x and o.

# Program Features

## Program Considerations

## Game Engine

The engine I plan to use for the game is Java Eclipse’s built-in engine. Once the code is complete and working without bugs in Eclipse one could transfer the same program to libgdx or any other engine as long as it’s implemented into the Game Loop.

## Click and Play

The game will involve using action listeners from the java.awt. package.

## Testing

Testing will be done by first using two played by human objects in order to play the game and see how the row and columns will be read by the program. The user will click on the 3x3 array and see how exactly the array is effected by the mouse click event and if there will be an oval drawn or if there will be an x drawn.

## Debugging

Testing how each player wins will be important to see if the project functions properly.

Also testing through the program and seeing how a draw will work Is also important. Finally, making sure the program closes properly must be tested by making sure that the loop will end.

### Exception Handling

When the user wins the game, has a draw or loses, the mouse click event will stop drawing objects causing the game to shut down and only giving the user the possibility of exiting.

Also the user will not be able to click the same box twice since isOccupied will check to see if a 1 or 2 is present as well as if it is ‘X’ or ‘o’.

*Original: In order to play x or o, the user must determine this by typing in the row followed by the column, however indexarrayoutofbounds will handle if the user types anything outside of the array space.*

# Main Flow

The main flow of the program will be pseudo code block and ran atomically in an ACID manner. The main portion of code will run flowing through to the other classes as needed and then continuing on to the next portion of the main code.

## Pseudo Code

TicTacToe is a game that involves a 3x3 matrix that will be made up of arrays.

The game will involve a player taking turns X being the first and O being the second.

The program will display a welcome message indicating the first move and asking the user to choose the slot he would like to make a move in.

Using action listeners the user will click somewhere to fill out the box that will be drawn in by the draw function.

TicTacToe is a 3 state game with a winner, loser, or draw. My job is to make sure that if the columns are filled diagonally, vertically, or horizontally, to present the user with a win or lose else all other statements end in a draw.

# Program Details

## Classes

There will be multiple other classes in order to help me create the framework of the game. Many objects will involve the drawing in of the x and o as well as creating the grid layout and arrays that show if there is a valid winner for the game. My tictactoegame will extend the jframe to pull from its functions that involves jlabel and jpanel. Finally I will use a constructor to pull from the frame class in order to put it in my main tictactoe game where I will run the game.

## OO methodologies

My OO methodology involves creating a board and a main method in order to execute my classes and the others will manage the display through reading the boolean expressed through the game board.

The following OOM will be applied:

### Encapsulation

I will be placing private variables in the TicTacToeBoard Class that establish the frame using arrays that will hold the columns, called gameboard.

### Polymorphism

My TicTacToeFrame class will extend the jFrame class which comes from the javax.swing\* package.

### Inheritance

My tictactoegame will pull from the constructor and inherit the whole game that has been made in the frame.

There will also be inheritance from the jframe and jawt packages being used for action listeners. Here I will be establishing the game changes in arrays and using characters to fill out the drawn in X and O.

### Constructor

My cell grid in my TicTacToeFrame class contains my constructor which contains the arrays that are used to create the Jpanel as well as hold the cells. Also my constructor will be used as the framework behind all of my code as the arrays used in the constructor will be the one that’s are modified when actions are performed on my game.