Colegio San Ignacio de Loyola

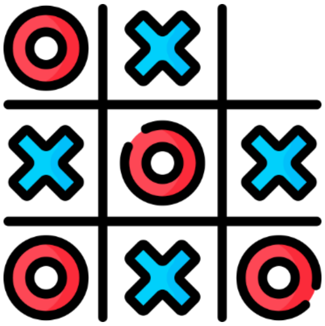
Introduction to Java programming

(a.k.a Advanced Programming)

Prof. Reynaldo Belfort, S.J.

Module 6 - Workshop 4

Project #1: Tic-Tac-Toe Game Development



Final deadline: March 20th, 2024

(before Holy Week. Short Presentations after Holy Week.)

Objective: To develop a console-based Tic-Tac-Toe game in Java that allows two players to play against each other.

Introduction

- This project will encourage teamwork, problem-solving, and Java programming skills, especially involving the management of arrays.

- This project is also meant to be a fun experience, while also providing you a good taste of how the software development industry works.

- Your team will give a **short presentation** (5min max.) on the unique game characteristics and/or features the team has developed, plus share highlights and challenges in their developing experience.

Teams:

Students will work in teams of 2 or 3, based on the teams that have been formed in the few past weeks. Each team will be responsible for designing, coding, and testing their version of the game.

>> Starter's Kit <<

All teams will be provided with various files to get them started. You and your team will not create the Tic-Tac-Toe game from scratch. Instead, you (and the team) will be provided with a code template, along with other files. These are:

1. **This document.**
2. **Professor's Tic-Tac-Toe Java code template** (.java file)
3. **Professor's Sample code for adding console-based** colors, animations, sample ASCII art, etc. (`JavaUtils.java` file)
4. **Individual assessment report** (.docx file) – to be filled out and turned in by each team member at the specified deadline.

Make sure to play/tinker with the .java files to familiarize yourself with the code structure, game mechanics, and any other code characteristics.

Work Distribution among Team Members:

Team of Two:

1. **Person 1 (team leader): Game Logic Development**
   a. Design and implement the core game logic (checking win conditions, handling player turns, and managing the game board).
   b. Implement input validation any time there is input requested by the user. Including the assurance that players can only make valid moves.
   c. Ensure their code is well-documented with comments explaining the logic and decisions.
   d. Lead the code integration efforts. That is, collect the code of the other team member and integrate it into one file.
   e. Write individual assessment report.

2. **Person 2: User Interface and Interaction**

    a. Design and implement the console-based user interface (UI) where players input their moves.
    b. Handle the display of the game board after each player's turn and at the start of the game.
    c. Work on the game's starting and ending sequences (introduction message, deciding who goes first, displaying the winner, or indicating a draw).
    d. Write individual assessment report.

Both team members should collaborate on:

- Initial planning and design, including drafting the Game Design Document.
- Ensure that the game logic and UI components work seamlessly together.
- Testing and debugging the entire game.
- Writing the team reflection report and preparing a short class presentation.

Team of Three:

1. **Person 1: Core Game Logic, Win Conditions**

    a. Focus on the internal logic of the game, including determining win conditions and handling the turn mechanism.

    b. Help person #2 if he is already done with task(a).

    c. Write individual assessment report.

2. **Person 2: User Interface (UI) and Player Interaction**

    a. Handle the input/output aspects of the game, ensuring a user-friendly way for players to make moves, see the game's state, and any other visual elements such as animations, colors, etc.

    b. Write individual assessment report.

3. **Person 3 (team leader): Enhancements, and Documentation**

    a. Work on additional features decided by the team to be added to the game (e.g., replay option, score tracking, AI opponent for extra credit).

    b. Lead the code integration efforts. That is, collect all individual codes and integrate them into one file.

    c. Ensure the entire game code is well-documented with comments explaining the logic and decisions.

d. Write individual assessment report.

e. Lead the writing of the game design document and the **team's reflection report**.

All three team members should collaborate on:

- Collaborate on planning and design,
- Ensure that the game logic and UI components work seamlessly together.
- Testing and debugging the entire game.
- Writing the team reflection report and preparing short class presentation.

Project Requirements:

1. **Game Design Document (GDD) (1-3 pages)**: Before coding, each team must submit a simple game design document. This document should outline:

   a. **Basic information:** Team name and team members. Date.

   b. **Game information:**

      i. Game title and game rules.
      ii. How the game determines wins or draws.

   c. **Game features:** Decide as a team which game features will be added to this game. See #2.j below for the list of all available game features that could be added to the game.

   d. **User interface design:** Basic sketch of how the **Tic-Tac-Toe game board** should look like (use computer's Wordpad or similar note-taking software. Or sketch on paper and then take photo with your smartphone). Remember to consider the use of colors.

      i. **Note:** There are many ways of drawing a Tic-Tac-Toe board game. Use the internet and/or your own creativity to come up with your team's drawing board.

   e. **Role assignments:**

      i. **For a team of two:** who is going to be the team leader (and thus do the corresponding tasks for the leader, as detailed above).

      ii. **For a team of two:** who is going to be the team leader, person #2 and person #3 (and thus do the corresponding tasks as the person, detailed above in this document).

2. **Code Implementation**:

   a. The game must be implemented in Java.

   b. The game should be console-based, with players inputting their moves via the command line.

   c. The game must display a welcome message, which includes:

      i. Game title (can be ASCII Art – See `JavaUtils.java` from Starter Kit)

      ii. Team name `"Developed by the GameChangers"`.

      iii. (optional) Introduction and/or instructions.

   d. The game must be developed based on the code structure provided by the professor as a template.

      i. **Important note:** If the professor determines that a very different code structure has been used (i.e. taken from the internet or created by generative AI), team members run the risk of receiving **30% off** their total grade.

   e. The code should check for win conditions (three in a row, column, or diagonal) and draw conditions (no spaces left).

   f. The game must alternate turns between two players, with the game displaying which player's turn it is (Player X or Player O, or by the player's name).

   g. After each move, the current state of the board should be displayed in the console.

   h. **ASCII Colors:** As a minimum, both the X and O must appear in different colors, to better identify each of these characters on the board. Colors may also appear in other parts of the game, as desired.

   i. **Input validation:** The game should check for invalid inputs (e.g., out-of-range moves, or positions that are already taken) and prompt the user to try again.

   j. **Game features:** The game must incorporate **two** of the following features (bonus points for adding an extra game feature):

      i. **Player names** – inserted by each player at the beginning of game.
      ii. **Scoring board & tracking** – counts how many times a player has won for each replay, or assigns points for each placement of a letter in the board. Or any other creative use of a scoring system.
      iii. **Simple AI opponent –** no need to use actual AI **algorithms.** A simple algorithm that mimics some sort of intelligence works just fine. I.e. you

could use the generation of random numbers, where numbers correspond to a location on the board.

iv. **Replay** – after a game session has ended (by win or draw).

v. **Game console animations** – hint: the use of delays allow for animating text…!

3. **Code documentation**: Each team must document its code with comments to explain the logic behind key sections. Thus, not every single line of code must be commented on. Instead, comment on the key sections of the code.

4. **Reflection Report**:

   a. **Team Reflection Report (TRR) (1-2 pages):** After completing the project, each team will submit a reflection report discussing: (1) what they learned, (2) the challenges they faced, and (3) how they overcame them. This report should also include how tasks were divided among team members.

   b. **Individual Assessment Report (1-page max.):**

      i. Mid-way throught the development process and after completing the project, each team member will submit their own evaluation of how the team members are doing in terms of their contribution to the project.

      ii. **This is a confidential report that only the professor will see.** This will help the professor assess how the group is doing in terms of contribution efforts from each member.

Short Presentation

- As mentioned at the beginning of this document, your team will give a **short presentation** (5-7 minutes) on the unique game characteristics and/or features the team has developed, plus share highlights and challenges in their developing experience.

- This presentation can simply be based on the Game Design Document (GDD) plus the Team Reflection Report (TRR).

- Participation of each team member at the presentation is expected.

Submission:

Each team member will submit the following:

1. **Source code of individual work (at Stage two)**. See deadline below.

2. **Final version of game:** Each team member should submit the same final .java file. This file should be named: `TicTacToe_[teamName].java`

3. **Two (2) individual assessment reports:** in a PDF document. **These are confidential**. It will help the professor determine the contribution efforts among team members.

Apart from all these files, the **team leader** must also submit:

4. the **Team Reflection Report** on behalf of the team. This should be a PDF document.

Deadlines:

- **Game Design Document:** Monday, March 12th, 2024 by 11:59pm.
- **Stage 2 progress report:** Friday, March 15th, 2024 – Asynchronous class-day
  - **Current coding progress:** Each team member's .java file on what they are currently working on. Turn in whatever you have. No points are assigned here. This is just to assess how each team member is doing, so that the professor can give support if needed.
  - **Individual assessment report #1:** use the .docx template provided in the Starter's Kit.
- **Final version of the game with documentation:** Thursday, March 20th, 2024

  Each team member will provide:

  - A single .java file, with the entire game. Thus, each team member must turn in the same and final .java file.
  - This file should be named: `ticTacToe_[teamName].java`
  - The **team leader** must also submit the Team Reflection Report on behalf of the team.

Evaluation Criteria (Rubric):

Each team member will receive points through the following criteria:

- **50% | Functionality**: Does the game run as expected without errors?
- **10% | Code Quality**: Is the code well-organized, commented on, and adhering to Java coding standards?
- **15% | Collaboration**: How effectively did the team work together? This will be assessed mainly through the reflection reports plus other efforts as needed (e.g. interviewing individual students).
- **15% | Creativity**: While the basic game is simple, points will be awarded for creative enhancements based on the **game features** added (e.g., adding a score tracker, implementing a simple AI opponent, etc.).
- **10% | Short presentation:** Have the team members provided a summary of development efforts based on the GDD and TRR documents?
- **5+ Bonus |** Team added an extra game feature to their game app (a part of the 2 mandatory game features).

Recommended development approach

The development process will take place in four stages:

- **Stage 1** | Initial meeting: Game Planning & Design

  Tasks:
  - Draft the Game Design Document. Thus, decide as a team the various features and characteristics of the game app. Don't worry if later in the development process the team decides to change the game characteristics.

- **Stage 2** | Individual Development & Testing
  - Person #1 & Person #2: develop Java programs independently following the task list provided in the "Work Distribution among Team Members" section of this document.

    - Person #3: might need to wait until persons 1 & 2 finish their part. Additional features could be added.

- **Stage 3** | Integration & Testing
    - Please check the team members "must collaborate on" section for a task list.
- **Stage 4** | Preparing Documentation & Presentation

Tips:

- **BACKUP, BACKUP, BACKUP!** Always have a second .java file saved somewhere else in your computer. Also, saving your .java files in OneDrive (provided by Colegio San Ignacio) is highly recommended.

- **Sending/Sharing code across team members:** assuming your team decided to use Visual Studio Code as the IDE to develop this project, refer to the following website for a tutorial on how to use LiveShare (similar to Google Docs)/
    - Share a project and join a collaboration session in Visual Studio Code | Microsoft

Resources:

- Refer to the JavaUtils.java file for coding resources that can help the team achieve the various game characteristics.
- Course presentations and workshops, as reference for Java coding.
- You are also allowed to browse the internet to get help for specific Java concepts.
- **Use of Generative AI is not allowed in this project!!**

Happy Coding!!!