

Project Plan

Target System

This application is meant to be run cross-platform on any device that supports Java. The release of Java does not matter as long as it is a version of JRE 8.

Technologies to be Used

This project will utilize a variety of technologies in its development. One developer will be using Atom and IntelliJ to develop the Java code, while the other will be using Netbeans. To foster collaboration, a source code repository in the form of Github will be used. To encourage communication, the chat platform Discord will be used. Both project management documents were made with Markdown using plain text editors. The algorithm will be created using Google Docs. Flaticon will be used for any icons necessary for our project. Other images and assets were gathered from various online open source projects. All copyright usage is under the terms of fair use, as this project is strictly for education purposes.

Our group also plans to integrate a feature of Java Swing known as the LookAndFeel into our project. This will be done using an external jar file which will allow us to have an immaculate interface while avoiding having to style the components too much.

Scope of the Project

Final List of Deliverables

- title page with instructions and a variety of game modes to choose from
- creation of a point object to represent connect 4 pieces
- creation of a game state object to represent the current situation on the board
- creation of effective minimax AI for the player to play against
- creation of a mechanism to track the number of moves and possibly end the game once a move total has been reached
- creation of a checking algorithm to observe if the game is over
- creation of a file management system to load game states into the connect 4 application
- addition of menu bars to exit the application or save the game
- an instructions document to tell the player how to play connect 4
- creation of an options pane that allows the user to change various things about the game
 - piece color
 - difficulty of the AI
- creations of a statistics panel that allows the user to find various statistics
 - how many player 1 wins vs player 2 wins
 - how many AI wins vs human wins

Form In Which the Deliverables will be Delivered

- an executable .jar file which can be used to evaluate the application as a whole

- a .zip file with all of the source code and the assets used for the application included
- a link to the Github repository where all of the version control and tasks completed by each member are included
 - all documentation will be available at that repository as well

Role Distribution

The role distribution will be loosely based on the schedule posed through the Gantt chart attached. However, our group has taken into account that any further level of detail in planning role distribution will be for naught, as changes in scope will drastically change this role distribution. Hence, our group will be creating a general outline for role distribution here, but this is subject to change drastically during the process of the project.

Tasks To Be Undertaken and Completed by Andrew Chen

- creation of options panel
- creation of statistics panel
- creation of the main "game mode class"
 - this will be taken upon and expanded for Player vs AI and possibly also for Player vs Player through inheritance
- creation of the main structure of the GUI
 - this includes a basic structure for a grid and buttons
- creation of the main menu for the GUI
- creation of a file IO system to allow users to save statistics across different runs of a program
- system allowing players to save and load a game state from and to the connect 4 board
- writing the project charter

Tasks To Be Undertaken and Completed by Victor Gao

- creation of the minimax AI
- beautification of the GUI
 - this will include most of the various pictures and icons needed for the GUI
- console applications verifying that both the minimax AI and the two player systems work before implementation into the main GUI
- implementation of backend logic regarding win detection
- keeping track of the different meeting times and compiling that information into a document
- debugging of any GUI components and of backend logic excluding reading and writing to files

Tasks To Be Undertaken and Completed Through Collaboration

- writing this project plan
- writing the algorithm
- integration of the minimax AI into the GUI of the program
- version control

- this will be through the management of a Git repository hosted on Github

Risk Management

We recognize the risk that may be posed by having uncoordinated testing environments and different versions of our code and seek to fix this with the usage of version control and proper communication.

Version control will be done using Github. Before either developer makes any edits to the program, they will check that their testing environment uses only the most updated files, identical to those on the shared repository. After editing or adding files, developers will commit all of their changes immediately to the repository. To cover any issues that may come up, Discord will be used as the primary method of communication for this project. This software was chosen due to its ubiquity and ease of use for a team of two users. Meetings will be held twice a week to discuss the state of the project and any changes that need to happen.

As this is the first time that the members of this project are working collaboratively on a project, there are many steps that need to be taken to ensure that there are no issues with collaboration. Proper communication is key to this, as many situations such as both of us writing the same class at the same time can simply be avoided with communication.

Change management will be introduced with both members of the development team needing to communicate before implementing any change of scope through Discord.

Change Management

Understanding that changes in scope and the specification for our project are inevitable, the following system will be implemented to deal with such changes.

Step 1 - Receiving the Request

The request could be received either directly from Mr. A or from the specification. If it was from the specification, this could just be a change in scope due to a misunderstanding of the specification.

Step 2 - Impact Analysis

The developers will now have a meeting to discuss the implications of this change in scope on the project. Here, the time taken to make the change, the different methods that need to be changed, and impact on the overall schedule of the project will be considered.

Step 3 - Finalization

Before beginning to implement the change, the developers will have one final meeting to discuss the project. If this will prevent the team from finishing a much larger portion of the project, then the change in scope will be put on hold until the end of the project to implement. However, if the change is urgent, it will be implemented immediately.

Step 4 - Implementation

The change should be implemented after much communication between the developers. This step varies greatly with each different scope change.

Step 5 - Review

The developers will have one final meeting to discuss the success/failure of the change management procedure. If there are any large issues with the change, we may have to redo this procedure or request an extension. Otherwise, this change management procedure is finished.