# Included Classes – Java

There are three overall packages in the code distributed for this project: One package for Connect Four (c4), one for 3-player Tic-Tac-Toe (ttt3player), and one for traditional 2-player Tic-Tac-Toe (tictactoe). The 2-player Tic-Tac-Toe code is intended to give you examples to review and to help you diagnose errors in your Minimax and/or Alpha-Beta Pruning implementations. I will not grade, or even look at, anything related to 2-player Tic-Tac-Toe.

## Connect Four Package

Classes in the c4.mvc package define the game rules and the interface for playing the game. Two interfaces are available: A console text interface and a “silent” interface. The ConnectFourModel class contains the game logic and has the functions that players can use to get information about the game (e.g., the current state of the grid, what moves are available). You should not modify any code in this package.

*Note:* I have provided a third interface that uses a GUI. This GUI was originally designed for Human vs. Human play. I’m not 100% confident in its use for Human vs. AI games or AI vs. AI games.

Classes in the c4.players package define players in the game, including the agent you will develop for this project. You have been provided with a HumanPlayer class that allows you to play, and a RandomPlayer class that makes random moves. Your agent should be part of this package.

The c4.ConnectFour class is your main class to start the game. Functions have been provided to play a single game or to play many consecutive games. You may wish to pit your agent against the RandomPlayer, your classmates’ agents, other people, or yourself.

## TTT3Player Package

This package is for the 3-player version of Tic-Tac-Toe. The structure of this package mimics the structure of the c4 package. The only difference is that there is no “silent” interface.

You should only need to work in ttt3player.players.TTT3PlayerAIPlayer (your agent class) and TTT3Player.java (the main class to start the game).

## TicTacToe Package

This package is for the traditional 2-player version of Tic-Tac-Toe. The structure of this package mimics the structure of the other two packages.

This package is provided purely for testing and demonstration purposes. Several examples of functions that you need to write are included in this package. I also recommend testing your code for Question 5 (the alpha-beta pruning algorithm) in this game, since the depth of the game tree is much smaller.

======================================================

# Included Classes – Python

The Python code is organized into several files. The game that each file applies to should be evident by its name: Connect Four (c4), 3-player Tic-Tac-Toe (ttt3player), and traditional 2-player Tic-Tac-Toe (tictactoe). The 2-player Tic-Tac-Toe code is intended to give you examples to review and to help you diagnose errors in your Minimax and/or Alpha-Beta Pruning implementations. I will not grade, or even look at, anything related to 2-player Tic-Tac-Toe.

## Connect Four

Classes in c4model.py define the game rules and logic, and it has the functions that players can use to get information about the game (e.g., the current state of the grid, what moves are available). You should not modify any code in this file.

Classes in c4view.py provide an interface for playing the game. Two interfaces are available: A console text interface and a “silent” interface. Classes in c4controller.py communicate between the model and selected interface. You should not modify any code in these files either.

Classes in c4players.py define players in the game, including the agent you will develop for this project. You have been provided with a HumanPlayer class that allows you to play, and a RandomPlayer class that makes random moves.

The connect4.py script is your main class to start the game. Functions have been provided to play a single game or to play many consecutive games. You may wish to pit your agent against the RandomPlayer, your classmates’ agents, other people, or yourself.

## TTT3Player

These scripts are for the 3-player version of Tic-Tac-Toe. Classes in ttt3player\_mvc.py are similar to those in c4model, c4view, and c4controller. There is no silent interface for this game, but otherwise, the classes are similar. You should not modify the code in this file.

You should only need to work in ttt3player\_players.py to add to your agent class and ttt3player.py (the main class to start the game).

## TicTacToe

These scripts are for the traditional 2-player version of Tic-Tac-Toe. The structure of these files are similar to those for TTT3Player.

This package is provided purely for testing and demonstration purposes. Several examples of functions that you need to write are included in this package. I also recommend testing your code for Question 5 (the alpha-beta pruning algorithm) in this game, since the depth of the game tree is much smaller.