Intro:

In this tutorial we are going to learn how to build our multiplayer game’s basic file structure.

Clip 1:

First you should notice the come2play\_as3 directory, get into it (pause) you will see a directory named api, you should not touch it as the files in it are responsible for the communication between your game and our server

Clip 2:

Now go back into the tutorial directory and go into the chapter 1 directory , notice the TicktactoeTuturial FLA file, this file is going to use us throughout the chapters of this tutorial as the graphic source ,this file will not contain any code

Clip 3:

Now go into the ticktacktoe Tutorial directory(pause) in it you will see 4 files , tickTackToeMove, a class representing a players move, TickTacToeGraphic, a class handling all of your game graphics, tickTacToeLogic, a class responsible for all the game’s logic and rules, and tickTacToeMain.

Clip 4:

Let’s open the tickTacToeMain class, notice that this class extends ClientGameAPI this is what gives our game its connectivity to the server, this class will be responsible for all the communication between the game client and the server.

Clip 5:

Now let’s go into the tickTacToeLogic class, notice this class contains an instance of the game’s graphics, you should make sure no graphical action is made in this class, and that no logical action is made outside of it, this well make debugging your game much easier later on.

If you’ll look carefully you will notice an event listener on our graphic, this event listener triggers the clickSquare function, notice that the function doesn’t have a direct effect on the game’s logic, and instead it passes the information through an event.

Clip 6:

Now let’s go into the main class once again, notice that there is an event listener on the game’s logic, this listener triggers the gotUserMove function that then sends back the information to the logic through the make Turn function, and this will make sense later on.

Clip 7:

The Make Turn function commits the move to the game board, and calls the graphic class to represent the move on the board.