Intro:

In this tutorial we are going to learn how to start your game, and how to handle our first custom values.

Clip 1:

First let’s learn how to register our game on the server, to do this we go to the TickTacToeTuturialMain class and write doRegisterOnServer(), calling this function tells the server that our game is fully loaded and is ready to play.

Clip 2:

After registering the server we will get a list of custom values, to understand their importance let’s play our game in the emulator.

Clip 3:

To do this, we are going to learn how to operate the emulator.

Go into the Emulator directory, and start index.html

Clip 4:

Now in the browser window that opened you will see the emulator definition page, we will just go over a few of the customizable values, for more information you can visit our wiki project.

“The file name”, the relative path to were your game is at (pause)

“CONTAINER\_gameWidth” the game’s width

“CONTAINER\_gameHeight” the game’s height

“Number of Players” the amount of players you want the emulator to simulate, in our case 2 players (pause)

“Number of Viewers” the amount of viewers you want the emulator to simulate, in our case none (pause)

After setting all these values we will click “lunch game” to start the simulation

Clip 5:

Notice that when the simulation starts there are two clients and a server, now let’s press start on these windows.

Notices the server log begins to fill up with all sorts of information, let’s go over it and see what everything means (pause)

Num, represent’s the order in which the actions took place

User represents the name of the user the line is about

FunctionName, the name of the function that is called, can either be a function name, representing a function was sent to a user, or doFinishedCallback, representing that a user finished processing a certain function.

Clip 6:

Now let’s play the game we made in the first tutorial in the emulator.

Clip 7:

See how the mouse clicks have a strange effect on the game? This is because when the game is compiled and is run normally its axis origin is x as zero and y as zero, but when your game is run inside a container the x and y positioning may change, depending on where your game is played.

To solve this let’s go back to the TickTacToeTuturialMain class

Clip 8:

After we register the game, the gotCustomInfo callback is called in our game, to compensate for changing our game’s x and y positions the container will send the game his x and y positions, we put the game’s x position in our game’s logic by using T custom, CUSTOM\_INFO\_KEY\_gameStageX with a default value of 0, meaning that if no value is received 0 will be used, same goes for the y value (pause)

Now let’s go to the game’s logic and see how we use these values, notice that the clickSquare function has changed and now it has xMod and yMod, adjusting the mouse’s absolute position