Intro:

In this tutorial we are going to learn how to manage collective data.

Clip 1:

Till now you’ve learned only how to manage subjective data, in this game’s case it was the game moves. all function’s associated with subjective data start with a do… prefix, in this chapter we are going to learn about collective data management such as who’s turn it is, or who is the wining player, this data has to be unanimous, we can’t have two players believing they won, or two players believing it’s their turn in a turn based game, functions that have an effect on collective data are marked with a doAll prefix, this functions must be called by all the players simultaneously, and therefore will only appear in the callbacks, gotMatchStarted, gotStateChanged and gotMatchEnded.

Clip 2:

The first commend we will look into concerning collective data is the doAllSetTurn, the function gets a user id to set the turn to and time for the user to make his move, if the player doesn’t make his move in specified time he automatically loses, in our class we call this function from the next Turn function.  
The next turn function sets the turn based on the all player ids array that was received in the gotStateChanged callback, thus making sure all players will call the same player.

Clip 3:

The next commands we will learn is doAllEndMatch, which gets an array of Player Match Over elements, every Player Match Over class must be created by its static create function.

Every Player Match Over will contain the user id of the player the entry is about, the percentage of the pot the player deserves and the score of the player in the game.

In this example you can see that the game ends as soon as one of the wining conditions is achieved, because the game over function goes over all the game players and ends the game for them.

Clip 4:

Now let’s visit the emulator and see the changes we’ve made (pause)

Notice that now each player can only play on his turn, and that the game actually ends when a victory or a tie is achieved.