**Unity Multiplayer Part 2 – Local Multiplayer**

**MonoBehavior shorthands**

* + Transform is shorthand for gameObject.GetComponent<Transform>();
  + Camera.main always returns the main camera (“whichever is tagged MainCamera”)

**InputManager** ( [Xbox 360 controller mappings](http://wiki.unity3d.com/index.php?title=Xbox360Controller) )

* + By default, Unity has setup Horizontal and Vertical setup for any controller you plug in
    - 2 sets of controls for Horizontal and Vertical
      * The top one is for keyboard and mouse WASD controls
      * The bottom (“Horizontal”, “Vertical”) is for controllers
  + Need to setup input for each individual playerController
    - Need to setup JoyNum configuration for all buttons foreach controller
      * Aka, when you want 4 player support, you need 4 sets of inputs each with same setting but the JoyNum is changed for each set.
      * If you enable get motion from all joysticks, then all players will be fighting with each other

**Initialization Logic**

* + Inside your code, you’ll need a script for initialization logic to set number of players
  + Prompt the user for how many players you want in the game
    - Check Input (make sure inputs are setup correctly for that number of ppl)
    - Make sure camera system is setup properly for that number of ppl.
    - Instantiate that number of player objects
  + Input.GetJoystickNames() returns some number of strings for number of joysticks are plugged in, and their names.
    - When you first start up and in the main menu, and there’s four controllers plugged in, you want to set up all four of those inputs for all four joysticks
    - In your game logic, you’ll need some code that checks player1 hits a button, and player2 hits a button (“any player press a key to join”)
    - Script that’s looking for input from all your controllers, then as an input is pressed from that controller, the script will kick off logic for that player
      * Get all controller names is only for knowing how many controllers you have connected to your machine at any given time (you only need to know the number of strings from Input.GetJoyStickNames() ).

**Dynamic Number of Player** (things to keep in mind):

* + All our players want to be prefabs
  + All our logic for our players want to have separate input logic from game logic.

**Separate Input Logic from Game Logic** (*how things are controlled vs how things happen in game*)

* For Example:
  + Say we have 3 different types of players in a game
    - Players
    - AI
    - Network Players
  + Lets say we have if (Input.getButton) {rb.Addforce;} in a single script
  + Input and mechanics in same code
  + It will work for players, but it won’t work for AI or Networked players
* A better way to do this: (Share the same Player Logic)
  + For Local Players:
    - PlayerLogic
    - LocalPlayerInput
  + For AI Players
    - PlayerLogic
    - AIInput
  + For NetworkedPlayers
    - PlayerLogic
    - NetworkPlayerInput
  + The different inputs are generally referred to as Controllers

**Audio Listener**

Only one camera in your scene can have an audio listener attached to it per game.

Audio is all relative

There a couple of ways around this:

* Put the audio listener wherever you want in the scene
  + When you play audio, you can move the audio listener to that position
  + Move the source of the audio to where the audio listener is located on the main camera
* You can drag in any audio file, and then access it via the inspector
  + There’s an option to Force to Mono (normalizes it)
    - When you put it on an audio source, you can modify the spatial blend
    - setting of 0 is 2d (flat volume)
    - The setting of 1 is full 3d
    - Put it somewhere in the middle (will be loud near first player camera)