DIY Virtual Reality

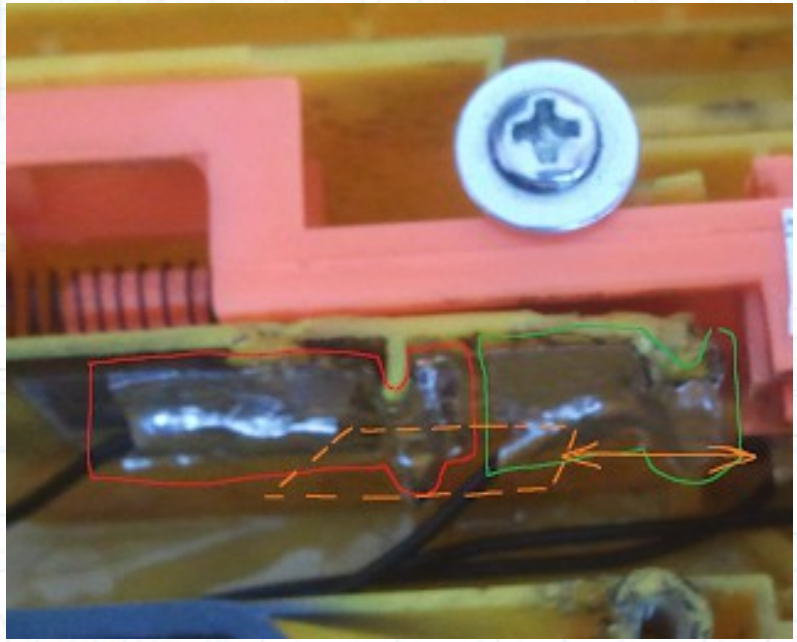
- Objective: Create a full VR experience
  - Virtual Vision with a 7" Tablet using Unity3D + Oculus Rift SDK + Android.
  - Movement System using Android and a phone.
  - Interactiong using a toy gun with Arduino + Android SDK.

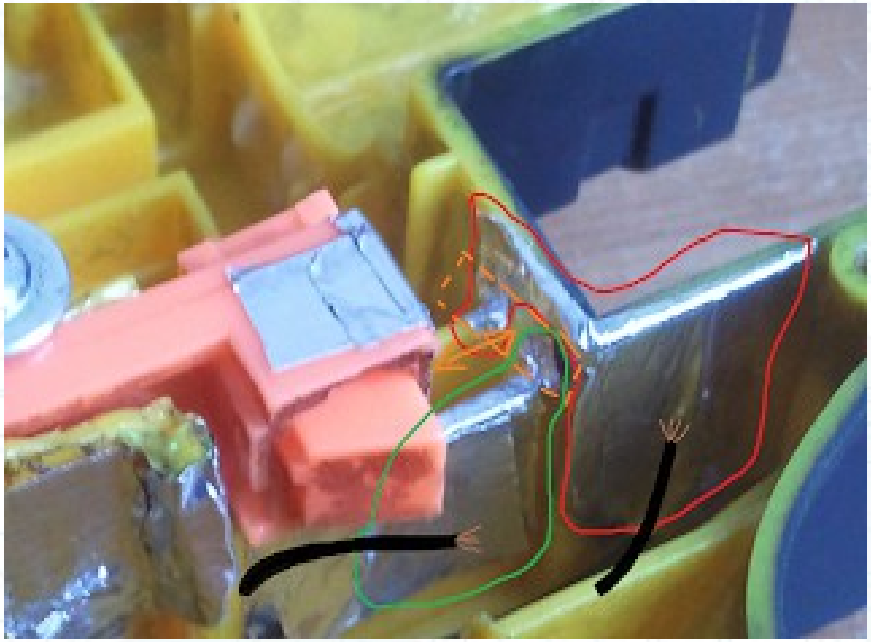
The Gun (Overview)

- Plastic gun modified to include an Arduino board (Makey Makey).

The Gun (Hardware)

- Create triggers using aluminum connected to moving parts and the board.





The Gun (Controller)

- Reprograming the Makey Makey board to send events when the circuits are opened/closed.
  - It will send a different key event for each situation:

```
if (inputs[i].pressed)
{
    if (inputs[i].bufferSum < releaseThreshold)
    {
        inputs[i].pressed = false;
        if(inputs[i].keyCode == CODE_TRIGGER)
        {
            Keyboard.write(CODE_TRIGGER_OFF);
        }
}
```

The Gun (Pose)

- Attach an Android device to the Gun to estimate the pose using the Gyroscope.
- How we avoid drifting?
  - Correct Gyro data with the compass and accelerometer data.

    http://www.thousand-thoughts.com/2012/03/android-sensor-fusion-tutorial/
  - Sometimes you have to give the option to the user to re-estimate the position.

The Gun (OTG Connection)

- Using an OTG cable to connect the board to the phone and listening to key events...

```
@Override
	public boolean onKeyDown(int keyCode, KeyEvent event)
	{
		if(keyCode == GUNEVENTS.TRIGGER_ON.getValue())
		{
			onFiring(true);
		}
		if(keyCode == GUNEVENTS.TRIGGER_OFF.getValue())
		{
			onFiring(false);
		}


	    return super.onKeyDown(keyCode, event);
	} }
```

Movement

- Using a phone in the pocket we can...
  - Detect the orientation of the user's hip.
  - Detect if the user is crouching...
  - Detect steps implementing a pedometer!

Note: We could even implement the pedometer directly in the tablet... but then we will need to use not only the “peaks” in the accelerometer magnitude but the rhythm to avoid confusion with the head movements.

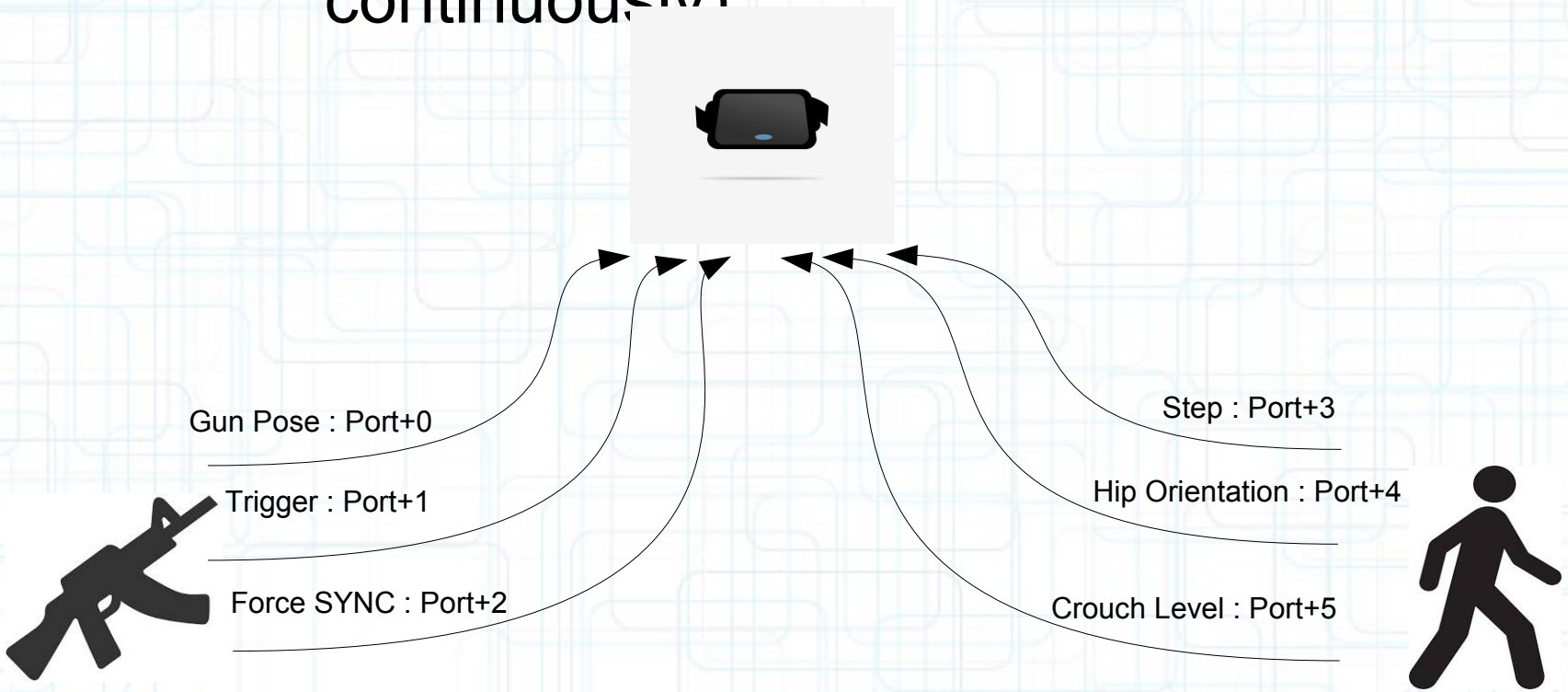
The vision System

- Oculus rift SDK takes care of allmost everything for us but...
  - We need to put the tablet in our head.
  - OR is for PC... and it has a few modules that will make our app silently crash (DevideImposter and GameController modules)

Using an existing Unity Game (Angry Bots demo) we can simply replace the main character and the default camera with the OR cameras, tweak a bit the values and we are done!

How everything communicates

- Same LAN for the 3 elements and UDP messages. (trying to get RT)
  - Using 1 different agreed port offset for each message (most are send continuously)



Communication (Clients)

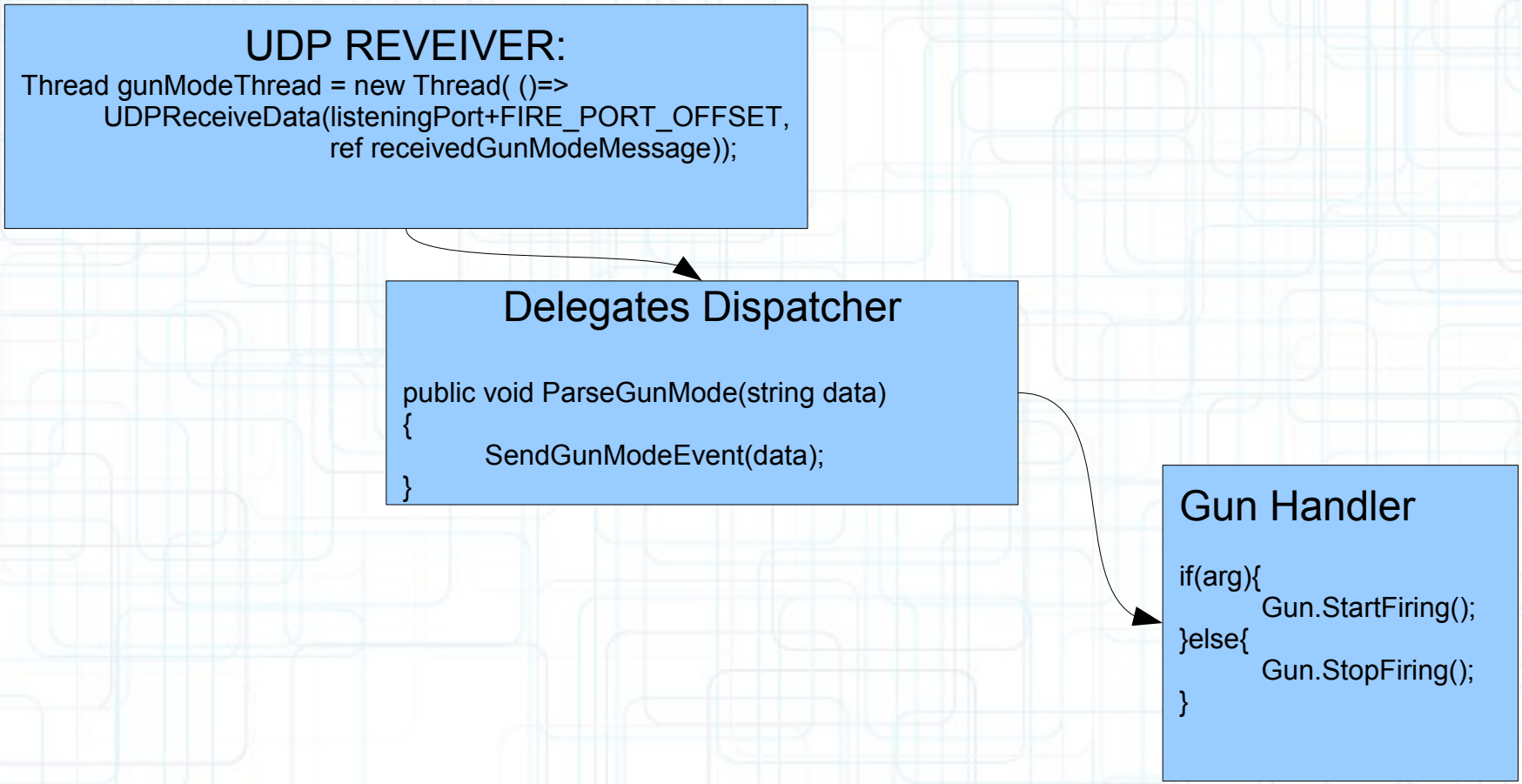
- The clients use listeners to the device events and send a DatagramPacket with the information as a string

```
@Override
	public void onOrientation(double[] orientation)
	{
		if(!viewModeButton.isChecked())
		{
			String msg = ""+orientation[2]+";"+orientation[0];
			orientationText.setText(msg.replace(';', '\n').replace(':', '\n'));
			new SendMessage().execute(PORTFORWARDING.MOV_ORIENTATION.getValue(),"UDP",msg);
		}
	}
```

```
private void sendOverUDP(InetAddress IP, int PORT, String message) throws IOException
    {
      DatagramSocket socket = new DatagramSocket(PORT);
            socket.setBroadcast(true);
            DatagramPacket packet = new DatagramPacket(message.getBytes(), message.length(),IP , PORT);
            socket.send(packet);
            //Log.i("SEND",message);
            socket.close();
    }
```

Communication (Server)

- The Server (Game) continuously poll the ports and when information arrives redirect it to the GameObjects that need it.



What's next?

- 3D print a proper helmet.
- Use a realistic gun, or several!
- Walk backwards? Jump?
- Joystick in Gun for easy playing on PC (Half Life 2) with Real Oculus Rift

Questions?