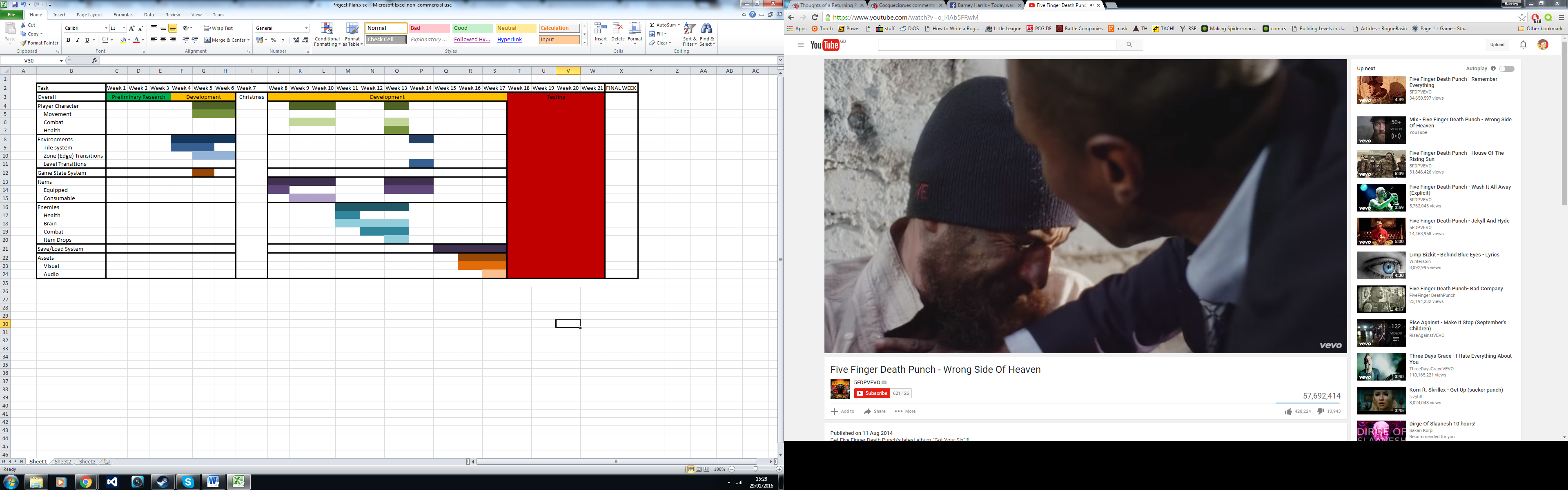
FUCK IT, let’s DO THIS.

LoZ Clone – Redux

# Project Layout

The first thing to do, before writing a single line of code, is working out what the project actually *needs*.

The original plan I made when I was using OpenGL for the project fell through for two reasons. The first was a mixture of external complications and my own flawed time management. However, the main reason I struggled to keep up with my plan is that it solely took into account



Once the overworld map was completed, I started working on integrating the underground, starting with caves. The main concern with implementing cave entrances and exits was how to handle checking where to transition to. The two options that made the most sense were:

1. Setting the collision value of transition tiles to 0 so they can be traversed, and having a manual check that transitions the player to the correct dungeon, cave, or overworld room.

2. Adding a new type of tile value to the collision map that represents a “transition point”, and creating an array of transition destinations that the game can check when one of the points is triggered.

---

Transition Start = Position of tile, compared against Link Pos + Vieport.  
To put player at end of transition, Link Pos = Transition position %16 and %11 respectively, and then viewport = transition – link. Puts link in the correct position in room then uses that to figure out which room.

How to read in transitions:   
tStart.x, tStart.y | tSMap | tDest.x, tDest.y | tDMap

INSIDE CAVE:

Two modes: Pre-interacted, post-interacted

On enter, pull text string for cave, draw letters one at a time. Once drawn, let link move.  
Once link interacts with item, do interactions and then all but flames disappear.

THIRTY SIX CAVE ENTRANCES.

Cave Text: Centralised. Position = middle – length of string \* ??  
If too long, multiple lines. Line = 22.