**ITEM 1**

**Rationale**

Angry bird

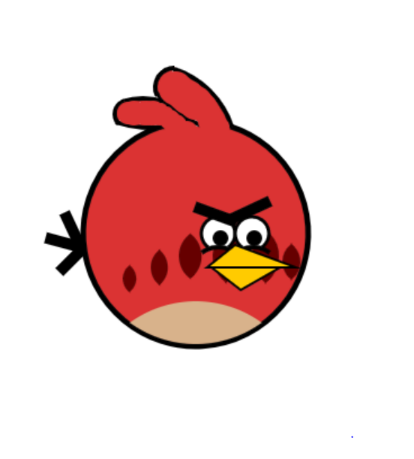
This is my static piece.

The reason I chose to make an angry bird out of shapes with code was that I thought I would challenge myself. An angry bird is full of different sized rectangles, lines, arcs and triangles, which I thought would be a lot more challenging that a house or a stickman and it would show off my skills in code.

Smiley face

This is my animated piece.

I looked a lot of different animated code online and decided to go with a smiley face as I have used 4 different animated pieces in one. Even though I chose the easier option of doing arcs, I feel that the way I have portrayed them and the way that I have animated them has made up for that.

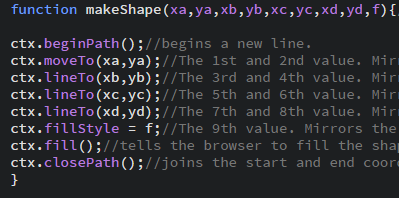




I thought that both of these pieces was a good idea because in a way they were both quite challenging and easy as I like this sort of code and I liked what I chose to make which made it easier and a bit more fun.

**Pseudo code**

Pseudocode is not a programming language; it is a simple way of describing a set of instructions that does not have to use specific syntax.



Function makeShape: The letter values mirror the values inside the brackets.

beginPath: Begins the path.

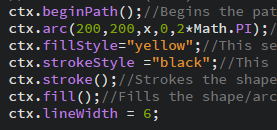
MoveTo: Moves the coordinates to these points.

LineTo: Joins the coordinates up from point back to the start point.

fillStyle: Tells what colour to fill the shape in.

fill: tells it to fill that shape.

closePath: Closes the path.



beginPath: Begins the path.

Arc: This is points of the arc for the shape.

fillStyle: Tells what colour to fill the shape in.

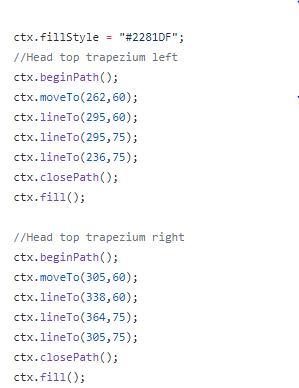
strokeStyle: Tells you what colour to stroke the outline in.

stroke:Tells it to stroke the shape.

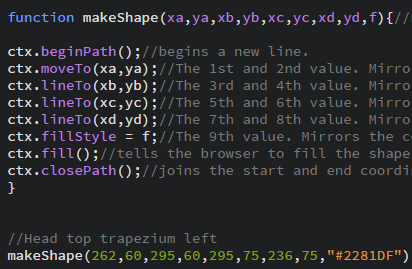
fill: Tells it to fill that shape.

lineWidth: Tells what width you want the stroke to be.

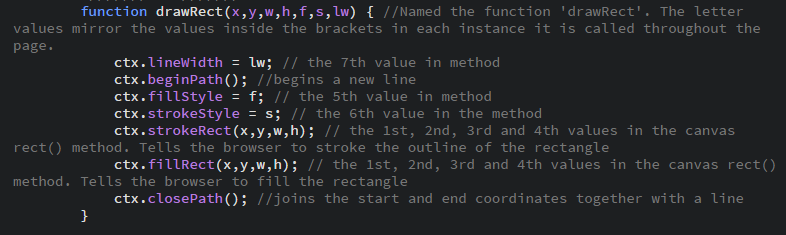
closePath: Closes the path.

**Critique of techniques**

This is the code I started with on brackets as I was getting to grips with how to make different shapes that were not rectangles. I then started to create functions for near enough every piece of code as it looked a lot neater and was easier than expected.



The code above is a function. It made everything a lot simpler and a lot more enjoyable once I got the hang of it.

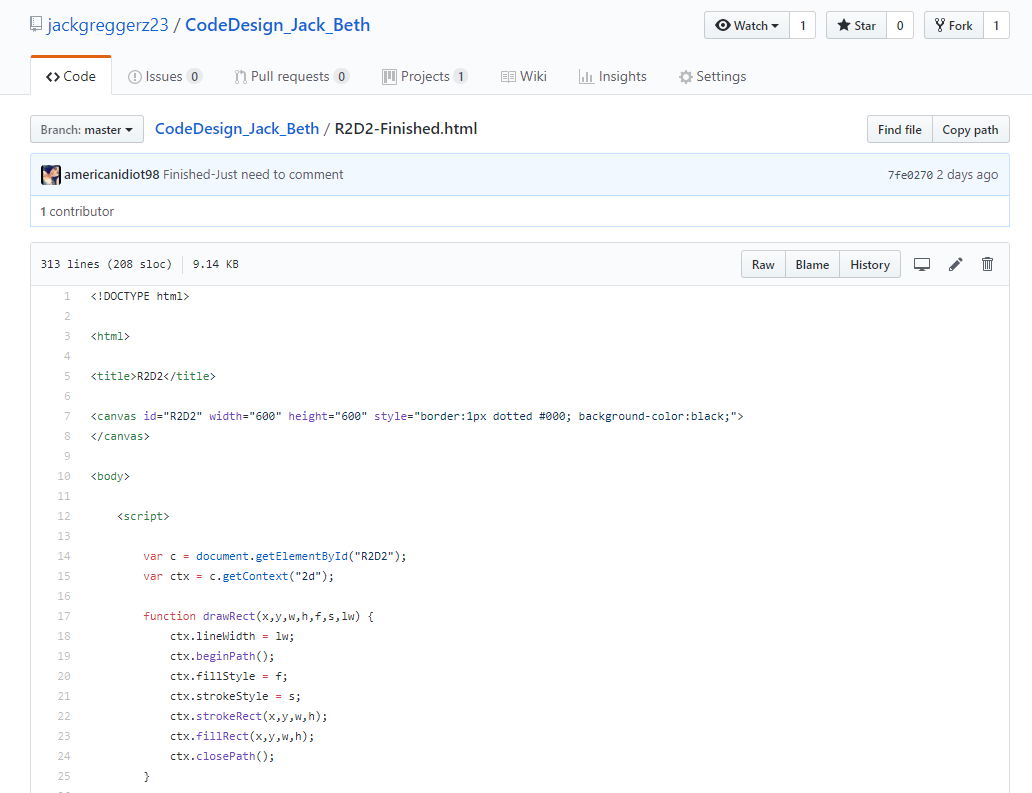
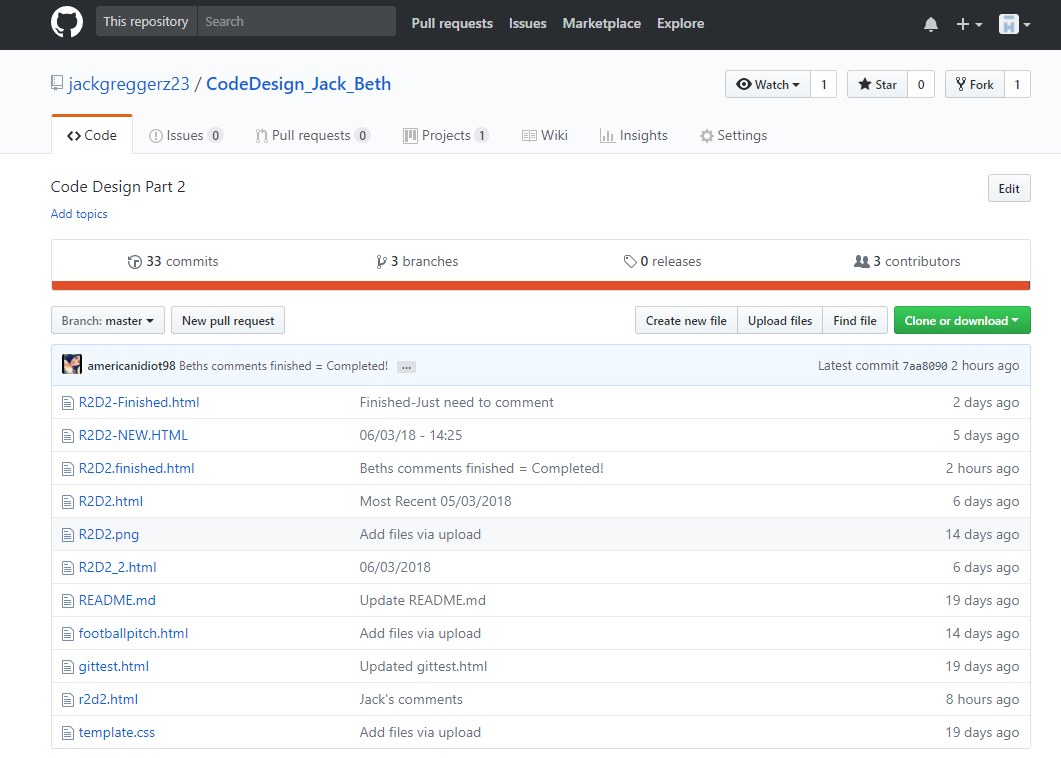


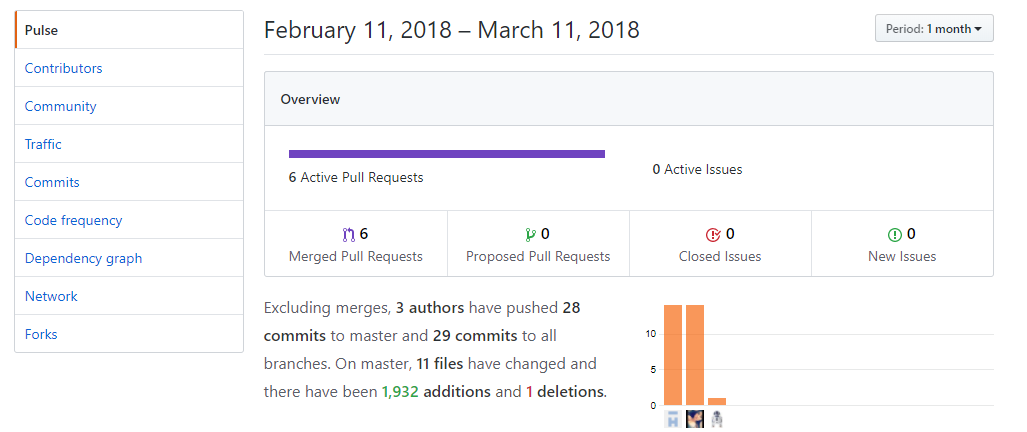
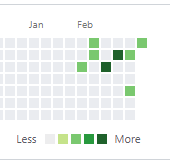
The code above is showing that the comments I have written are well thought out. I took time to think about how to word my comments so I and whomever else that read them could understand.

**ITEM 2**

**Code management tool – Github**

<https://github.com/jackgreggerz23/CodeDesign_Jack_Beth>

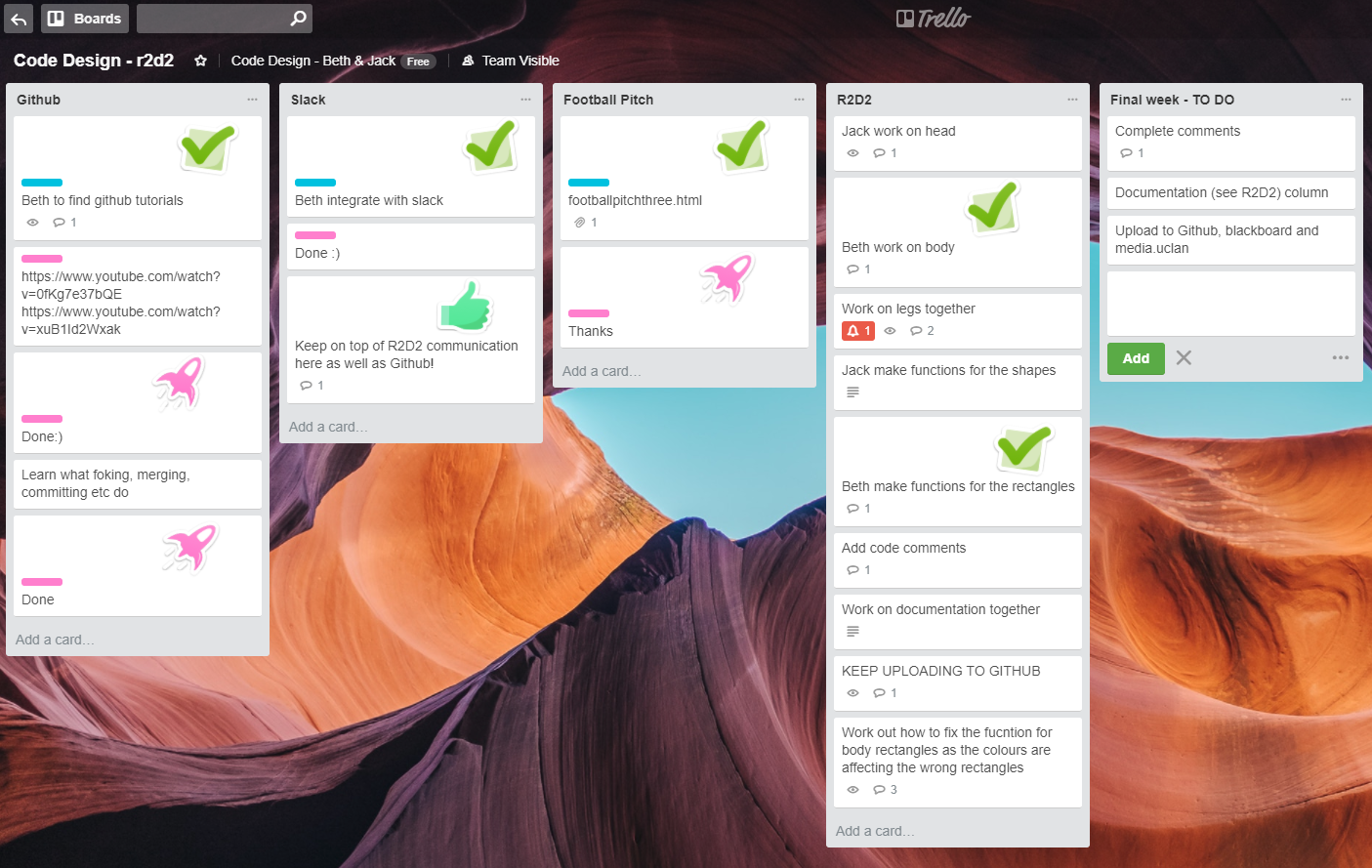


This is the evidence of my use of Github as the code management tool. Me and my teammate Jack uploaded our work here each time we had completed a section of the work. In order to version control, we committed, merged, pulled and forked eachother work. This elimnated the chances of work being lost and reduced the amount of time it takes to combine eachothers work together.

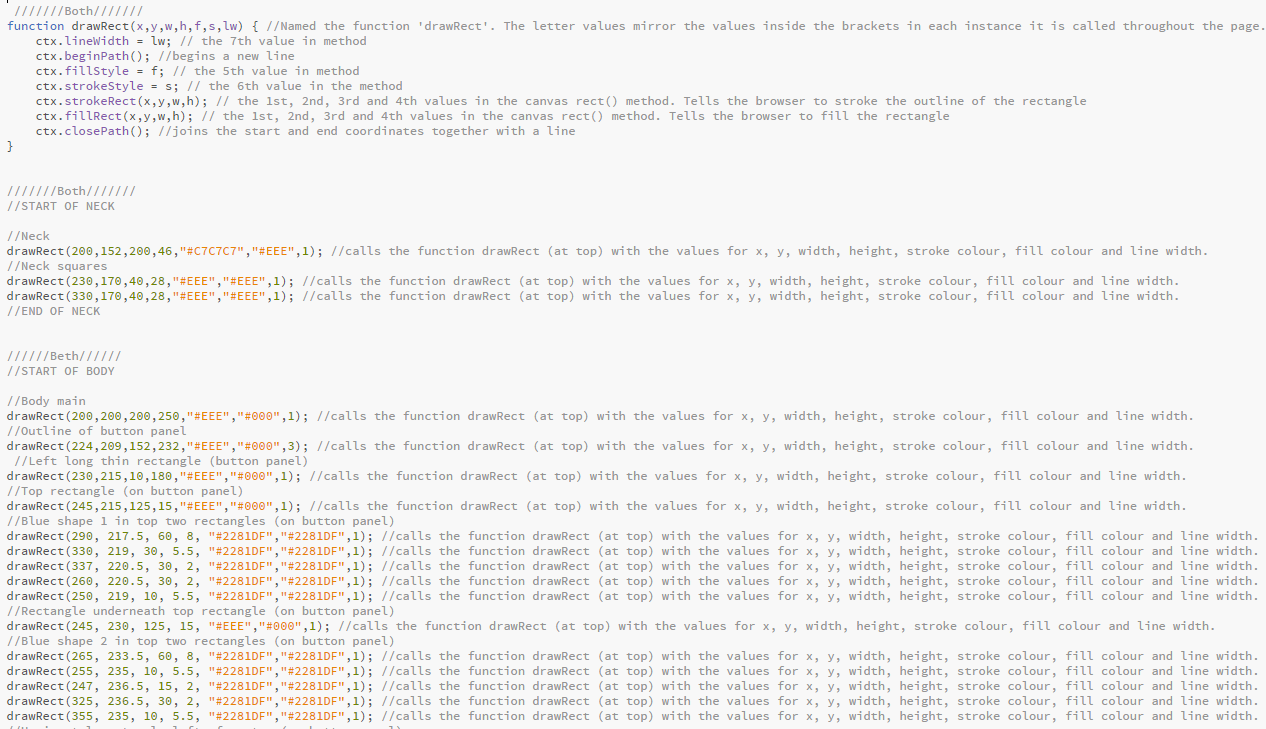
**Communication tool – Trello**

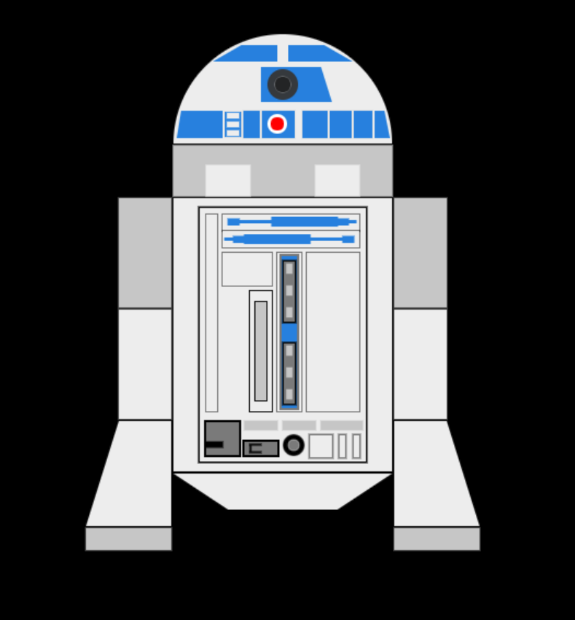
https://trello.com/b/Kxet66sN/code-design-r2d2



We used Trello as our communication tool throughout the process. We updated it when necessary with tasks to complete and responses when they were completed. It worked well as it gave us some structure to the process.

**Code Structure**



In order to keep a mutual understanding of the work we agreed on a structure to the code prior to starting. We agreed to give each section of code a short, but understandable name, for example, ‘Arse2D2’. We put these comments in block capitals, as we knew we would want to find these comments before we looked for the more specific comments in the section in question. We called each section we were the main contributors of by our name or both if we felt we were equal contributors, but we both contributed to all sections in some way.