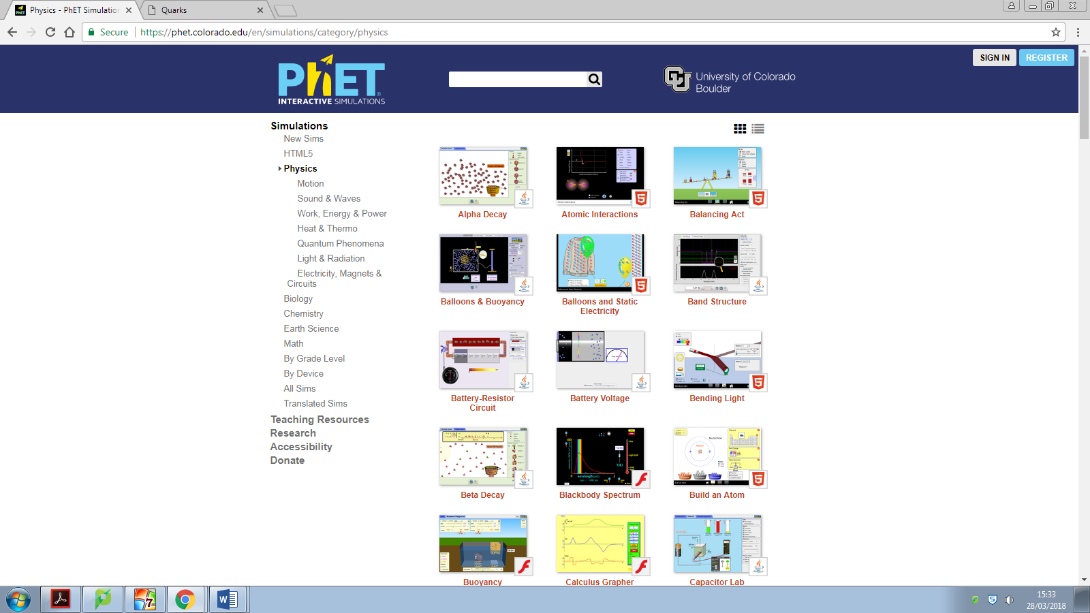
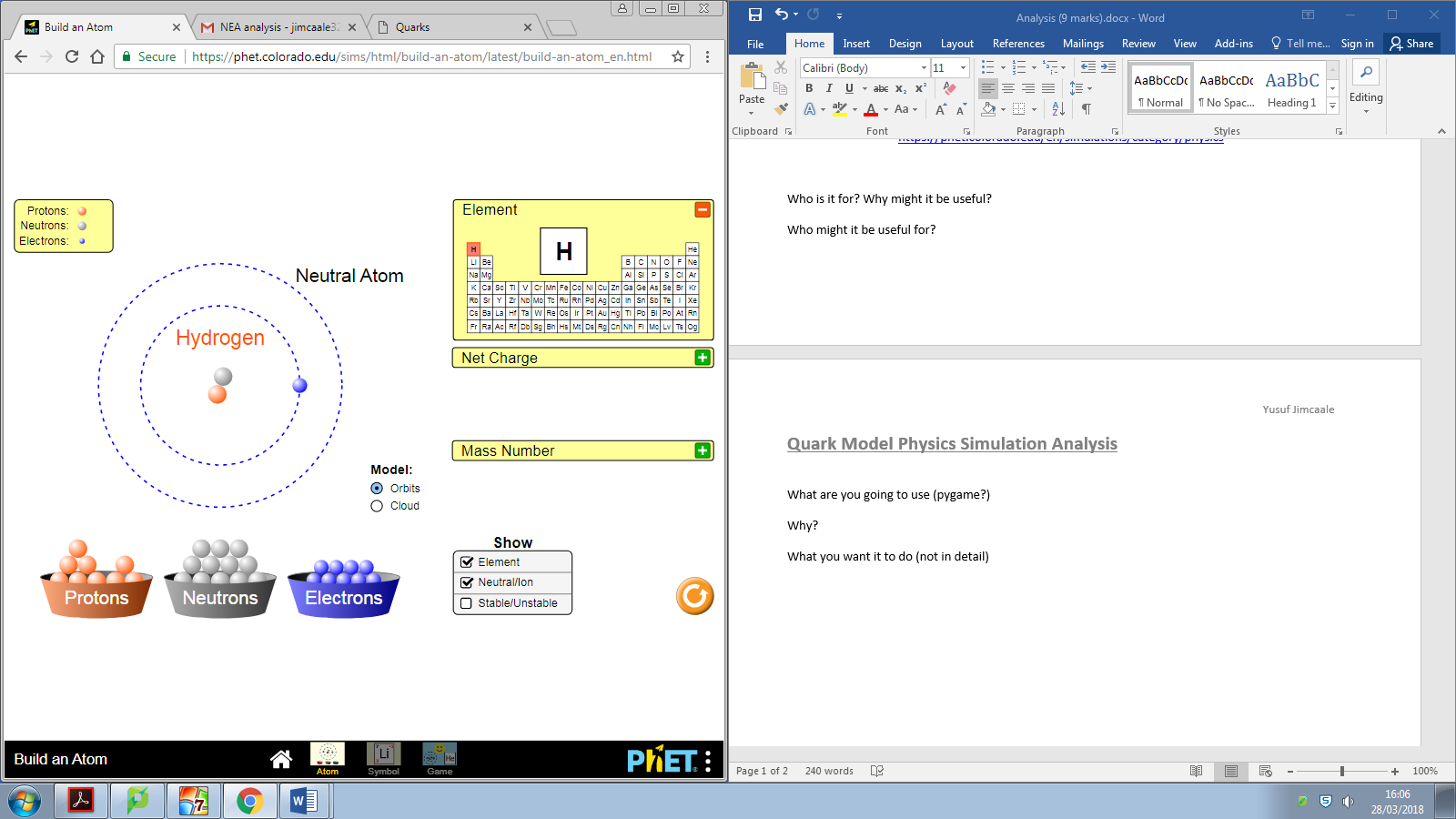
Throughout the course of my Physics lessons, my teachers often used computer-generated simulations to help explain topics with ease. The visual nature of the simulations (in my opinion) allowed for students (such as myself) to gain a more solidified understanding of the area in question. After several discussions with one of my physics teachers, I ended up finding the website from which he got many of these simulations – PhET Lab Simulations. After trying a couple of these simulations, I came across one, which really intrigued me – “Build an Atom”. The goal of this simulation was to show students how changing the number of protons and neutrons could also change an element and thus its properties. The program’s use of mouse interactions to press buttons and move around particles was something I really wanted to attempt in replicating (programmed in HTML 5). The website had a wide variety of simulations – which meant I was not short on ideas on what to do for my project. However, despite its vast wealth of resources, there were still some topics, which did not have simulations specifically for them.

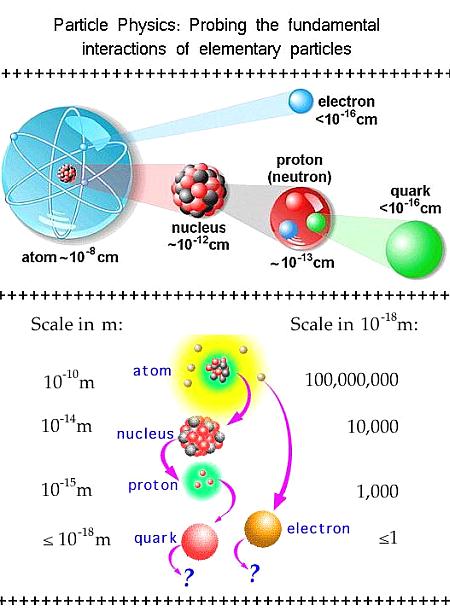
Whilst being taught about quarks and the Standard Model, I realised that there were no simulations at all for it – and given the nature of the topic, many of my classmates found themselves unsure and confused. After seeing this, I came up with the idea to create a simulation program, which would allow users to mix and match quarks together and allow them to pick correct configurations, to successfully create subatomic/composite particles. The program’s user interface will take inspiration from the “Build an Atom” simulation – with the solution including a particle space, quark buttons and a place where information about the particle is outputted. In many ways, my project aims to adapt this “PhET Simulation” and apply it to a very different concept.

Whilst the simulation is running, the user will be able to press labelled buttons which then generate colour coded quarks onto the particle space; the quark tally will also be incremented by one (depending on what quark button was pressed). They will be able to do this until the particle space becomes full, by which the user will be given a message letting them know of this. If the quarks in the particle space match a configuration of that of a type of particle (i.e. “up”,”up”, “down” for a proton), then the particle’s name and information will be outputted (stored in a class/read from a text file). Composite particles can either contain two or three quarks depending on their configuration. Twelve quark buttons will be at the disposal of the user – which includes six quarks and six anti-quarks (the antimatter version of quarks), with a combination of 12 possible particles to create – with each one having its information such as name, charge and type being stored beforehand. In addition to these 12 buttons, there will be a “REFRESH” button which wipes out the contents in the particle space, particle information and the quark tally – enabling the user to start with a clean slate and help facilitate experimentation with different combination types.

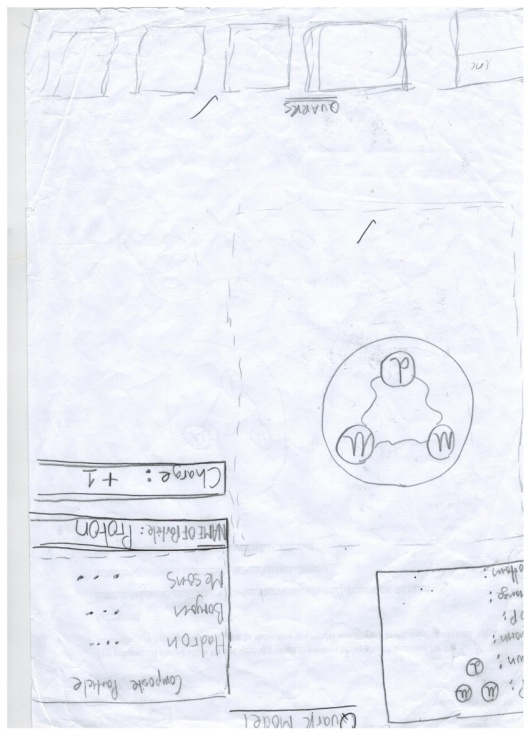


**Screenshot of “Build an Atom” Simulation** - <https://phet.colorado.edu/sims/html/build-an-atom/latest/build-an-atom_en.html>

**Screenshot of PhET Physics Simulations** - <https://phet.colorado.edu/en/simulations/category/physics>



Particle Diagram showing how quarks make up composite matter <https://sites.google.com/site/csetstudyguidechemistry/home/1-2c---atomic-structure>

****

My preliminary sketch of my proposed solution

## Independent Feedback

The solution to this project is to enable for the topic of Fundamental particles to be represented in the form of a simulation - so that students can more easily understand the topic. As a student myself studying physics currently, this project will allow me to combine both my physics knowledge and my coding/computer science skills into creating a simulation. After proposing this idea to my physics teachers – they gave me advice on how to make the simulation work effectively as a teaching tool – including extra information about the particles; adding extra “facts” which work to add a bit of entertaining value for the students.

* include data from questionnaire (analyse it)
* student feedback about programming
* Teacher feedback??

## Systems used

In terms of programming language, I am planning on using python to code my solution; having used it for a couple of years now, I will not have to spend any extra time learning a new coding language. In addition, the scale of my project will mean that my final solution will be perfectly suited to it. Upon looking for python libraries, I came across pygame. This library seems to be the most appropriate for my solution as it is able to deal with user inputs as well as creating a game window from which the simulation could be played.