Virtual Online Shopping

CS39440 Major Project Report

Author: Laura Wilkinson (law37@aber.ac.uk)

Supervisor: Prof. Bernie Tiddeman (bpt@aber.ac.uk)

19th February 2019

Version 1.5(Draft)

This report is submitted as partial fulfilment of a BSc degree in Computer Science (G401)

Department of Computer Science
Aberystwyth University
Aberystwyth
Ceredigion
SY23 3DB
Wales, UK

Declaration of originality

I confirm that:

- This submission is my own work, except where clearly indicated.
- I understand that there are severe penalties for Unacceptable Academic Practice, which can lead to loss of marks or even the withholding of a degree.
- I have read the regulations on Unacceptable Academic Practice from the University's Academic Registry (AR) and the relevant sections of the current Student Handbook of the Department of Computer Science.
- In submitting this work, I understand and agree to abide by the University's regulations governing these issues.

Name: Laura Wilkinson

Date: 25/04/21

Consent to share this work

By including my name below, I hereby agree to this project's report and technical work being made available to other students and academic staff of the Aberystwyth Computer Science Department.

Name: Laura Wilkinson

Date 25/04/21

Acknowledgements

I am grateful to a lot of people, starting with the staff at Aberystwyth – who have motivated me and let me achieve my potential as a programmer. One of those people was Andra (adb). She shared insightful guidance about project management and prioritisation. She also showed a lot of interest and hype which kept me motivated

I also want to say thanks to my friends – Luca, Jack and Tobias, who have been there through this project – helping me visualise and think through ideas and debug when issues were out of my scope.

The project won an award during the Lovelace Colloquium – “Highly Commended”

I’d like to thank…

Table of Contents

1. Background, Analysis and Process

1.1. Abstract

During COVID19, online retail is as strong as ever. I will be exploring the use of Virtual Reality to support both retail stores and customers. The use of real-time simulation of clothing can contribute to this – customers are more likely to buy items if they can examine the goods. This will include identifying the fit and colour of selected clothing items. My aim is to simulate high street shopping but from the comfort of the customer's home.

The data set will be from ASOS, I will be displaying a small selection of clothing items. A feature I will be implementing is a save feature. This will allow the user to take away the outcome of the application to then purchase the items if they so wish on the website.

The key features of this application are to have the ability to view clothing items in Virtual Reality. The application is going to be developed with the HTC Vive – though the application will be available on any headset compatible with SteamVR.

The application will have a specific area to filter down the clothing choices and an interactive area to view both the clothes and the model. The user can customise the model to match their measurements to see how the clothing looks using sliders.

With this application, I hope to validate Virtual Reality in this use case and produce a functional prototype to promote the use of VR in the retail sector.

1.2. Background

Many images posted by online retail companies do not accurately represent an items fit. This has been an increasingly more problematic as Covid restrictions limit in-person interactions – decreasing shopping reliability. During this project, I wanted to explore other options and technologies to create a more reliable shopping experience.

According to background research, Mixed Reality has been explored in a similar context before by companies such as Zara and Topshop.

Zara used augmented reality app to bring virtual models to life (WBR Insights, 2021). This was done in 120 stores for a two-week window during April. It worked with outfit and window displays, as well as AR enabled packaging and promotional materials. The main purpose of it was to display real models showing of clothes in short video bursts, between 7-10 seconds. These features are also available online by hovering their mobile over a package delivery. Overall, it increased sales by 2% that year.

Topshop used a virtual fitting room in Moscow, with a Microsoft Kinect and AR (Indvik, 2011). Users could switch clothes using gestures and display them virtually on themselves. This was on May 10, 2011.

They have also used virtual reality to host a 360-degree runway (Inition Digital Limited, 2014). The hardware used was the Oculus Rift. It was hosted in London during Fashion Week 2014 and the experience feedback was overwhelmingly positive as it won several awards.

The AR approach could fit several use cases according to one journal (Baytar, 2020). It was the most successful when used to convey visual attributes such as coordination and style. The findings of the study indicated that AR is proficient at providing a base level of detail and made purchase intentions more favourable.

My motivation for suggesting this specific project and topic is rooted towards my passion for Virtual Reality. I worked with VR hardware during my industrial year and have decided to continue with it as a career. I also understand the frustrations of having to send back items of clothing when they are ill fitting.

1.3. Analysis

The problem I discovered was that previous AR solutions were only viable on the base level and didn't have enough detail to be reliable. **I took aspects from many applications and games**, to design an application that is both intuitive and informative.

I primarily looked at the ASOS site and the example data from the List V2 and details V3 API (apidojo, 2021), finding the most popular filtering options and I gathered my data from there. See more details for the research.

The research I did opened the application up into several parts:

- Displaying clothes on a mannequin.
- Filtering clothing.
- User Interface to change the model dimensions.
- Getting the data set.

I chose the VR route when creating the virtual dressing room because it's innovative technology, and I already own a VR headset – the HTC Vive. It also fits in with the current political climate during this pandemic as people cannot physically visit stores. It will help online shopping as it can simulate the accuracy of trying on clothing items. The aim is to prove that this concept could work and be used by the public at sometime in the future.

Developing an application AR was a valid route, as other approaches had done the same. The examples I found were from 2014, the technology has progressed from then. However, the core issues are still apparent as it does rely on external hardware for reliability.

With those in mind, I created the following user stories to make up the requirements:

- As a user, I want to be able to change the model's dimensions to be my own, so that I can accurately view the clothing fit.
- As a user, I want to be able to filter clothes, so I get specific items to try on virtually.
- As a user, I want to view my filtered choices, so I can make sure I've chosen correctly.
- As a user, I want to be able to browse the ASOS Catalogue, so I can view it in VR.

There are minor security challenges expected as the application will be offline based, any saved preferences of clothing will be on the local system and the main data pull from the ASOS API must only be done once.

This will assess your understanding of the problems for your project, how well you assessed alternative approaches and your justification for the approach you selected.

There should be a clear statement of the objectives of the work or the research question, which you will evaluate at the end of the project.

1.4. Process

1.4.1. Project Tracking

For the project, I used an automated Kanban board hosted on GitHub. I chose this platform because I had experience with it from other projects and found it was a good way to organise the project. The entire project was put into 5 columns – Bugs, To-Do, In Progress, Done and Merged. See Figure 1 for an example of the columns. Once a feature was completed, I opened a pull request. I used pull request as both a good practice for when I am working in a team and it also warns me of any merge conflicts. Any completed pull requests automatically closed the linked issues and moved them to the done column.

Tickets were created when unexpected features raised – following an agile methodology. Tickets were also given labels so I could set WIP limits regarding each aspect of the project. The labels were core, ui, unity, data, bug, research, and enhancement. I set myself at least one Unity, UI and Graphics label a week so the project would be more well rounded. With this system, I could filter the labels on the board if I had to focus on the 3D modelling for example.

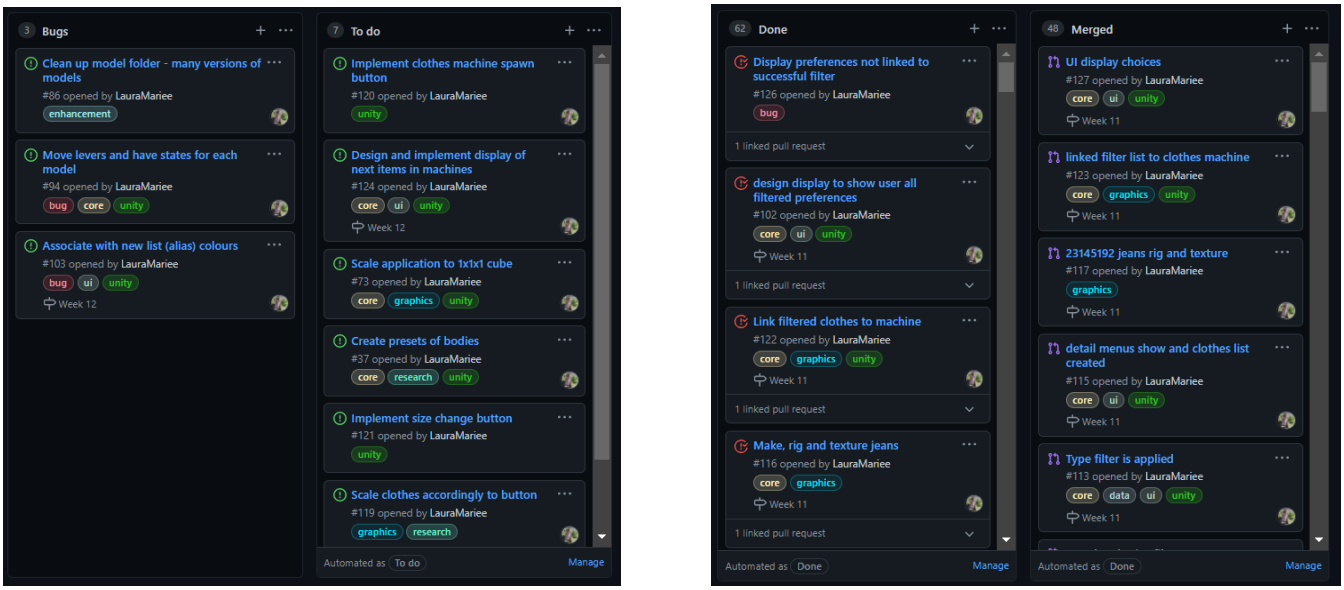


Figure 1: GitHub project board - columns

This project also took full advantage of milestones. I did this by cross referencing notes from the weekly group meetings, ensuring I hit my minimum weekly goals. Weekly goals were agreed upon with my supervisor and all changes and progress were documented in my blog (Wilkinson, 2021). Figure 2 shows a small selection of milestones used – along with the remaining open and closed tickets.

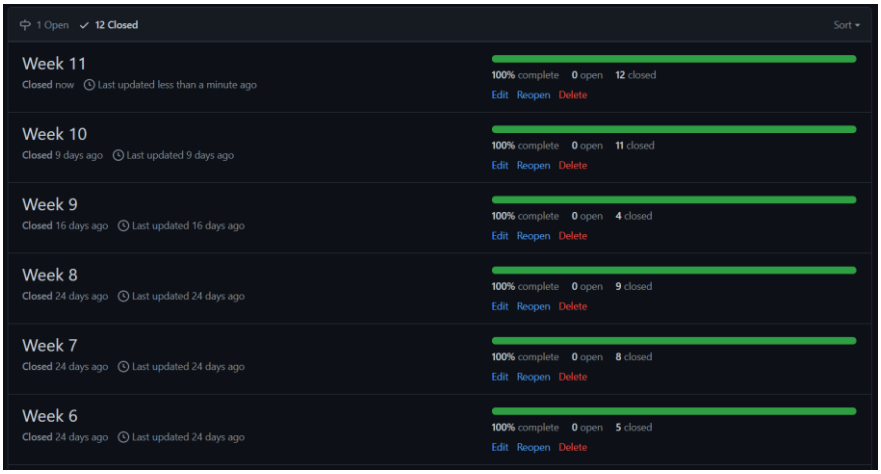


Figure 2: GitHub Milestones - week list

During this project, I also implemented a Github action – which marked stale issues and pull requests. This ensured I didn't forget about older tasks that needed to be done. It also allowed me to filter the "no-issue-activity" tag to ensure issues are tackled. See Figure 3 for the label change.

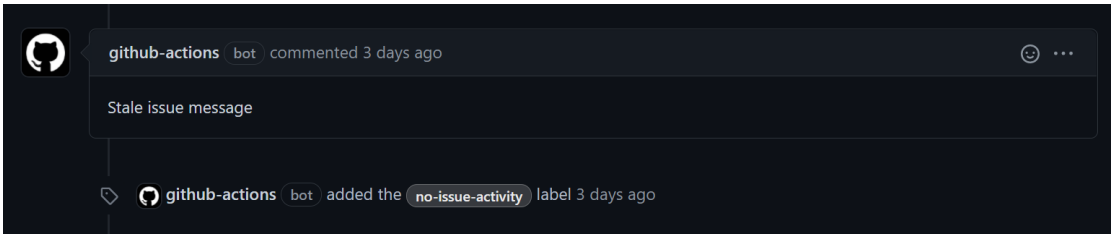


Figure 3: GitHub action - Stale issue list

1.4.2. Source Control

Throughout the project I used a total of 4 branches. I managed those using GitKraken – Appendix C iii, so I could have a visual representation of the branches**.** These branches were environment, models, testing and main.

The environment branch was used when I had to make any changes to the main scene in Unity – reducing errors and merge conflicts. The merge conflicts were generated by changes made by Unity to the reference files and other files that allow the engine to build.

The model branch was used for creating and testing the models within Unity – checking the lighting and object scaling for example. After models were completed – they got merged into the main branch. This was for all 3D models, including clothes.

The main branch was used to store the stable version of the application. The working components from the other branch were merged into this individually to maintain the applications stability.

2. Design

2.1. Overall Architecture

The software for the project is **Unity** and the IDE used was **IntelliJ Rider.** The software uses **Valve SteamVR** as the library to handle the VR output. More information is in Appendix A[2]. It uses many script components that work together and pass relevant data to each other.

2.1.1. Model Adjustment Levers

The Model Adjustment Levers control the dimensions of the body parts including waist, hip, neck, shoulders and bust. The script is simple as it uses an algorithm to calculate the levers range and assigns appropriate values from only the minimum and maximum. This object is referenced in the model to have a similar algorithm which assigns and scales the model.

I made the script attached to the model sliders as re-usable as possible, by reducing the amount public dependencies and iterating through a hardcoded list of strings instead to find the game objects. That both reduces de-referencing issues when it comes to game objects during development and makes the code clearer to any future developers.

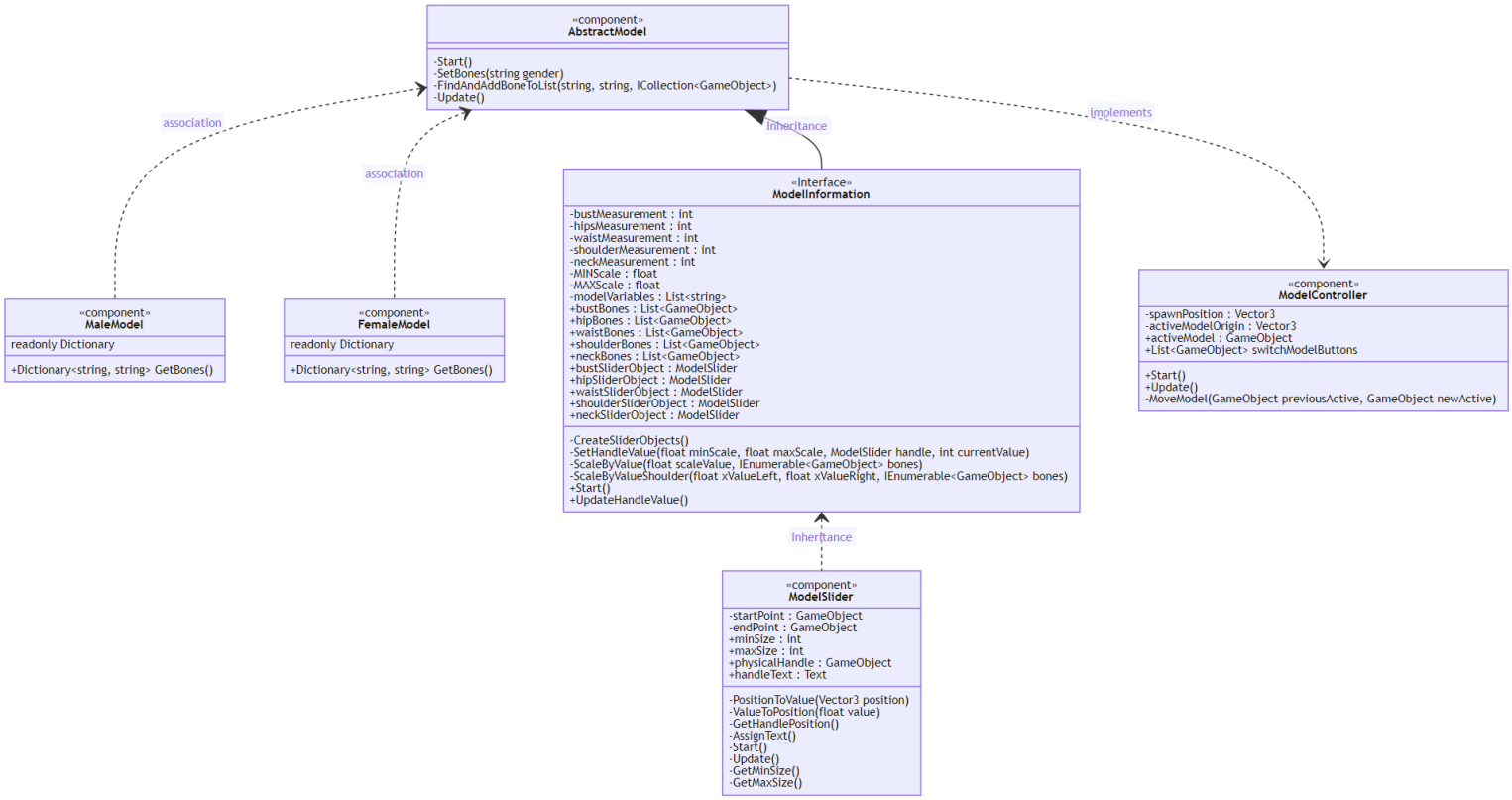


Figure 4: Model Lever Class Diagram

2.1.2. Model

The base model mesh's have been taken from a site online and I edited them for my use – the link for them is included in references. This was for a few reasons – the main being prioritisation of the other key components of the application. The models are royalty free and open source – to avoid any copyright issues.

The model works in tandem with the Model Adjustment Levers, the individual body parts are scaled according to the relative position of the lever. However, through research and finding implementation issues – I adapted the transform

of the shoulders by changing the x value instead. The outcome of scaling them up is deforming the model in an inhumane way.

The scene detects the active model on the podium and enables the BaseModel script.

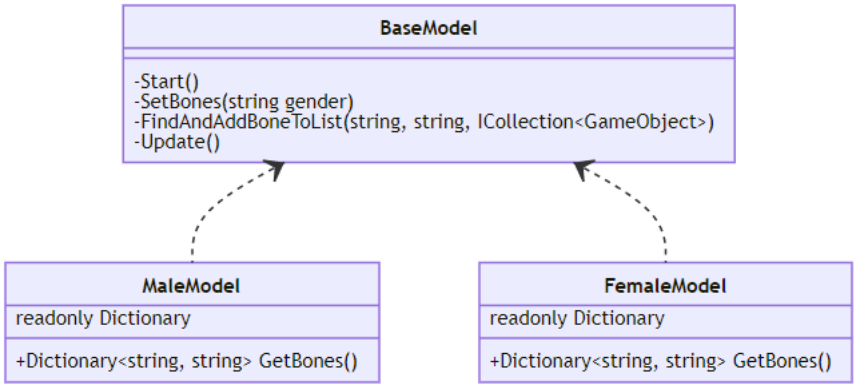


Figure 5: Model Component Class Diagram

Female and male model are different classes because the meshes have a different armature structure. The dictionary stores the name of the bones and that is given to the BaseModel class. The bones are found and set to be used in the ModelController script.

2.1.3. Switch Model

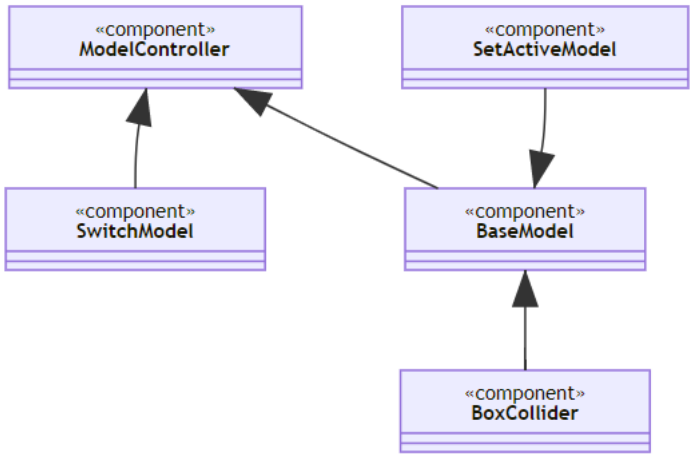
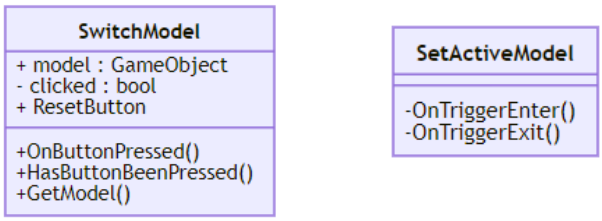


Figure 6: Switch Model Flow Diagram

This uses a mixture of unity components and scripts to find and set an active model.



2.1.4. Filtering

The application uses the filterController script to do the main operations and filterUI to handle the button inputs and displaying.

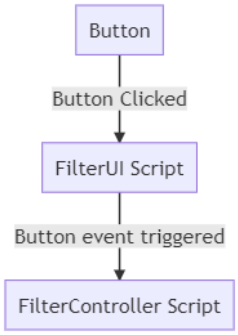
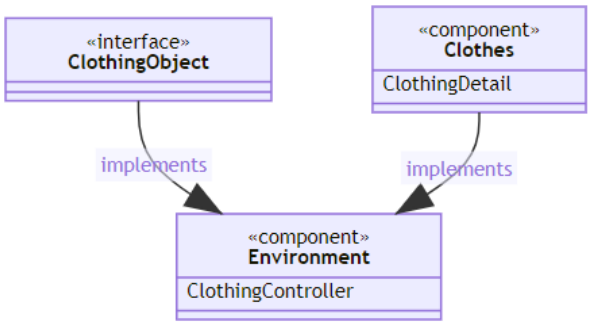


Figure 7: Information Filtering Diagram

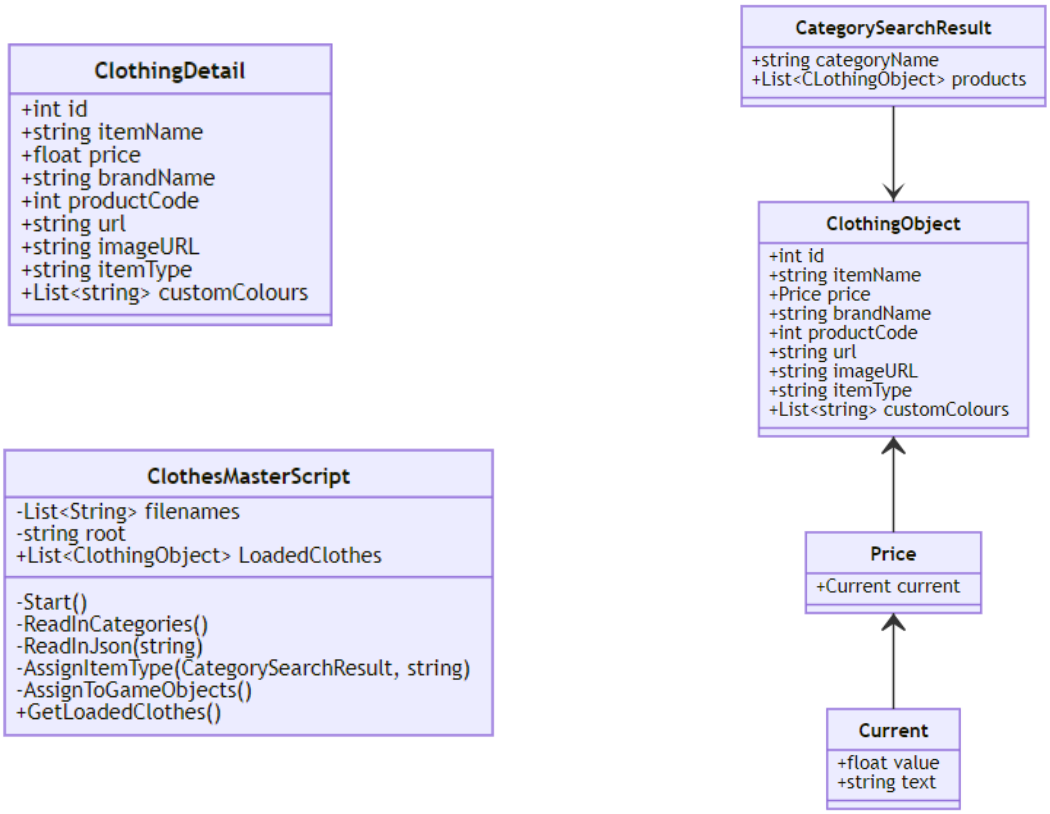
Add separate class diagram for filter UI and filter controller.

2.1.5. Clothes

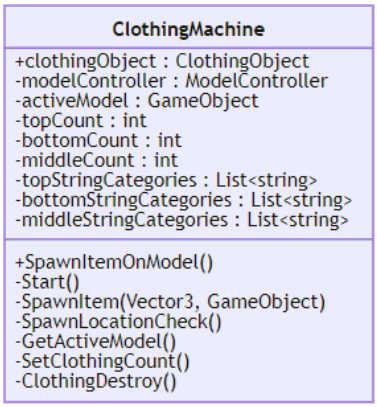


The clothingObject interface is the data structure that stores the data from the API pull. This gets transferred to the ClothingDetail which assigns the clothes via the Clothing controller script. All objects are assigned the details on the application start up.

A breakdown of the classes is below:



Clothing Spawn:



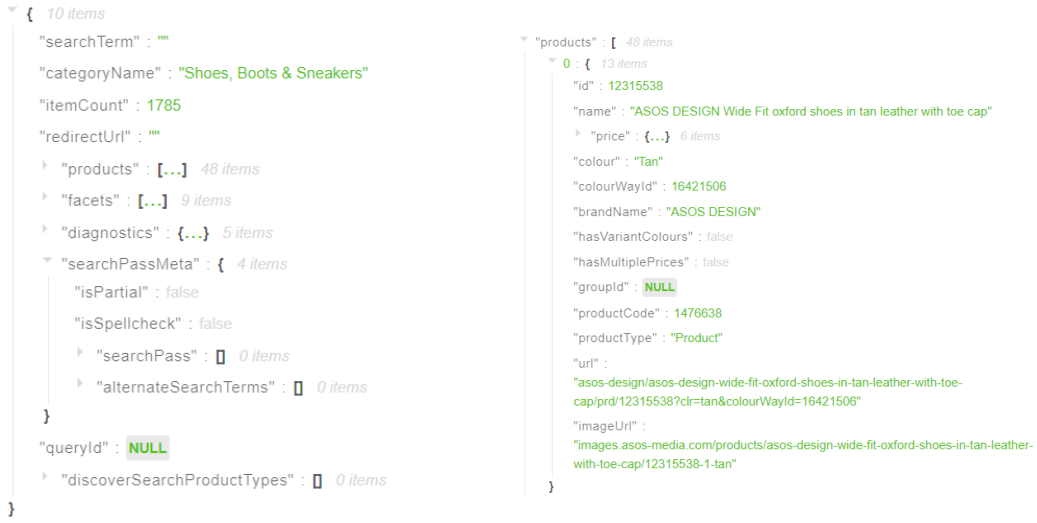
Class diagram of link to filtering UI

2.1.6. ASOS API Data

For the ASOS data pull, I used a third-party link called RapidAPI. I used this because it was free for my level of usage and simple to understand. It also allowed data end points to be reached from the browser.

It has an in-built tester for the end points so I can manually make queries.

I decided to write the pull in python in PyCharm 2020.2.3. I made it as an internal file for ease of use of other developers. This also future proofs the data pull so you get the data that fits and limits the errors within the main Virtual Online Shopping. This uses two end points to fill the data requirements, so the script must be custom.



The figure above shows the example data provided by RapidApi. This is the specifically from the ListV3 endpoint, which returns a number of products from a category.

2.2. Detailed Design

2.2.1. ASOS API Python Script

In this script, I got all the categories from the categories JSON file. This is shared with another script which assigns game objects. It is also used to create a query loop when pulling data.

Due to an unforeseen problem, the List API data pull doesn't include colours. This could be due to some clothing items being mix and match or not being assigned colours in general.

To fix this, I decided to factor in another API pull, detail v3. This needed the object ID for the endpoint to return successfully. To achieve this, I used this end point during the main loop, extracting the colour variations.

When the file has been written/pulled – in between the loop- the file is then "cut open" and the colour category is pasted in. The file is then closed, and the next file is ready to be processed.

An alternative approach was to create a second python script, which is run after the List data pull. This would assign the data to a new data structure within Unity. I decided against doing this as it was inefficient and overly complicated. It would also make the architecture confusing.

2.2.2. Model Levers and Scaling

Virtual Scaling
Scaling was done in several ways. I used the reference tables from the ASOS website for sizes and had that as the value inputs. As for the visuals, I based them off several sources. One of the ways I did it was by using a website, in which you can enter measurements and it would visualise the human body. This was used this as a base for both the Male[6] and Female body[5]. I used this to make the dimensions of the body look realistic.

User Scaling
The data gathered for Tops was for both male and female but due to time constraints, I couldn't implement the data. The tables of data are here:

Female - Petite:

Bust (cm)	Waist (cm)	Hips (cm)
78	60	83.5
80.5	62.5	86
83	65	88.5
88	70	93.5
93	75	98.5
98	80	103.5
103	85	108.5
110.3	92.5	116

Female – Curve:

Bust (cm)	Waist (cm)	Hips (cm)
109	91	116.5
116	98	123.5
123	105	130.5
130	112	137.5
137	119	144.5
144	126	151.5
151	133	158.5

Male:

Chest Size (cm)	Neck Size(cm)
76-81	36
81-86	37.5
86-91	38.5
91-96	39.5
96-101	41.5
101-106	43.5
106-111	45.5
111-116	47.5

The male data table is smaller than the female because less information is available.

2.2.3. Filtering

I made several design decisions when it came to this aspect regarding the filter criteria.

I chose to only save one price selection – the application only saves the minimum and maximum value from the chosen option. This is the only exception as it makes logical sense. The value is then filtered from those values.

I came across an issue when filtering the colours. Most brands aren't consistent when it comes to colour naming – especially French brands like Lacoste. To remedy this, I planned to create a list of strings that could alias them. This both helps the users as it simplifies the colours to choose and how the code processes it.

2.3. User Interface Design

2.3.1. Environment

Above is one of the initial sketches of the scene and how I wanted the application look. I also did another sketch whilst figuring out the features: see figure. To not make the scene feel cramped or claustrophobic, I added a window in the back – which opens the scene whilst not overwhelming the user with empty space. It also is a good tool to show off the skybox I selected from this website.

The scene also does not have a roof, this is done on purpose. I got the inspiration from the SteamVR Home- **Appendix C i**. My favourite part about sitting in SteamVR Home is admiring the quality of the environment and how it can take you out of reality with its realism and beauty. With that in mind, I wanted to bring that feeling to my application.

I chose a darker skybox as I wanted to highlight the different lighting types in the scene itself – creating a more of a relaxed atmosphere.

As for the room decoration, I decided on using some of the info-graphics from the ASOS website regarding size measurements. I thought it would be informative to the users to show them how to accurately measure themselves so they could make the best of the application.

2.3.2. Interaction

My aim is to utilise as many VR design aspects as possible, making sure all controls and UI are intuitive and clear to use. I also kept the scene as small as possible – so the user can walk around the scene without use of teleportation.

This includes certain interactions:

- Slider Handles
- Physical push buttons
- Scroll Bars
- Click buttons
- Teleportation

Teleportation

I put teleportation in the most active spots, the visuals showing when the teleport button is pressed. There is also a plane which is a teleport area – to give users more freedom when travelling around.

I chose this as the primary form of movement for several reasons. The first one being that that the application can be played in a small area or even standing position. The environment is compact to reflect this.

I also positioned the small, active spots for ease of use for the user – they're placed in front of all the key components of the environment. It both gives the user a hint of what to do and removes some hassle of re-positioning themselves to be able to interact with the features.

Buttons

I used a variety of buttons for the user interaction for a few reasons. The first reason is that the action needed for the button depends on the location. For example, a 3D button on a podium is instinctively going to be physically pushed. It also fits on the podium correctly and looks normal.

A 2D button is more suited to being on the y axis, on a wall for example. With this, a user would instinctively reach and click it. 2D elements suit better in this context as they provide a level of flexibility in the UI design of the application.

Slider Handles

Sliders move within the range of the start and end point, which is a pre-built prefab in SteamVR. This is a good tool as you can select a value from a large range without taking much space.

Scrollbars

I decided to use a vertical scrollbar in the filtering UI display as it's intuitive and removes the need for oversized placeholders for the buttons. They can be interacted by dragging the scrollbar components or the panel itself.

Scrollbars are also a built-in component of Unity's UI system – making it a simple but effective tool.

2.3.3. Model

Originally, I designed an interface in which you could cycle through a selection of poses. This would be attached to the base of the model.

This was done to user could have a more detailed look at how the clothes fit and see the clothes in a range.

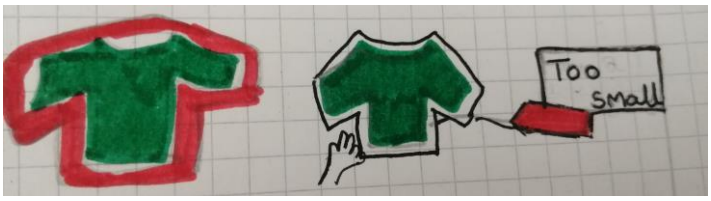
2.3.4. Clothes

Details

The UI that shows the clothing details triggers when the users hand hovers next to the clothing object. This shows the name, item type and price. It triggers a UI panel to show which is styled like a clothing tag. In the product, the UI was unfortunately not stylised, but I plan to improve it in the future.

Fit Error

I designed a UI which shows errors when the items of clothing don't fit.



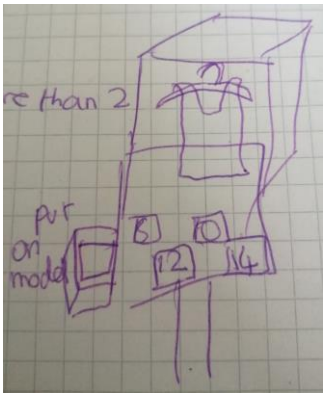
In the figure, this shows the two variations of the design. The first on shows a red highlight surrounding the object.

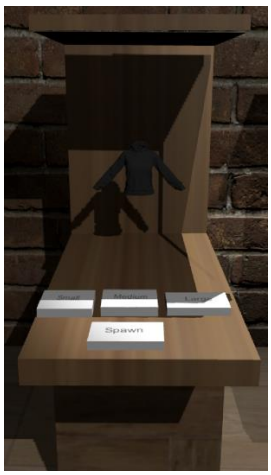
I decided against this for several reasons – the highlight could distort the colour of the object and it could also be a hindrance to the users who don't care about the size issue.

I designed another UI which only triggers on hand hover. It gives the user a choice of whether they want to see the UI. I incorporated with the clothing detail UI – see above.

Unfortunately, that was not implemented.

2.3.5. Clothes Spawning





The figures above show my original design variations for the interface of the clothes spawning. I wanted to incorporate this Interface into a 3D object because that is one of the fundamentals of VR. It also makes the scene more complex and interesting to interact with.

Button Variations
For the sizing, there was a variety of routes to explore to display the options the best.

I thought about using the slider concept to select the size of the clothing – that would cross reference the data I had previously collected. It would also re-use a component within the project, cutting down the development time. Though, the issue with this is that that there wouldn't be enough space to accurately select the value the user wanted.

The second idea was to use push buttons instead to represent each of the available sizes.

The third idea was to keep the user interface simple and use 3 push buttons:

- Small
- Medium
- Large

Design Variation - Wardrobe

Originally, I came up with a wardrobe concept which shows the filtered clothes and allows the user to grab and place the items of clothing on the model. However, I opted for a simpler system for several reasons. The first reason is that this concept would make the number of 3D models double, as it would need to show a "hung" form and the model wearing it.

It would also add more complexity and error checking on the development side as I would have a lot more test cases for if the user "lost" the clothing item for example. This has been a previous issue in a Unity project – in which it was an issue that I couldn't fix.

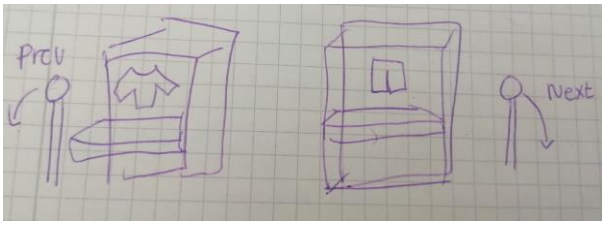
It would also be more inefficient on the architecture side as scaling the clothes could be tricky. I did not come up with a way to integrate the spawning system that I designed – the scripts would be harder to link and therefore harder for other developers to understand.

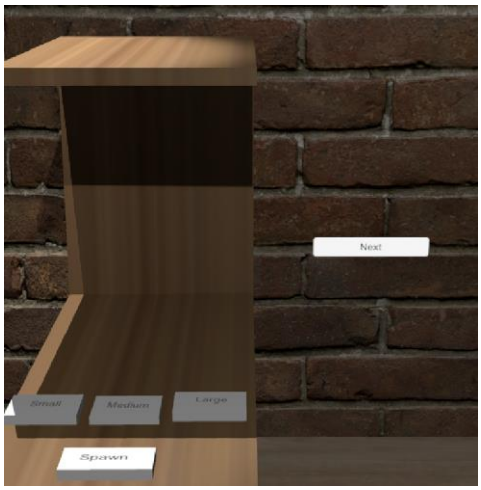
Clothing Reference Images:

In the original plan, I was going to use a script to download the images of the clothing items – supplied by the image URL. These were going to be placed just above the buttons as seen in figure.

Unfortunately, I couldn't implement the UI in time – though the image links have been saved and a script to download the images has been created.

Next Lever UI:



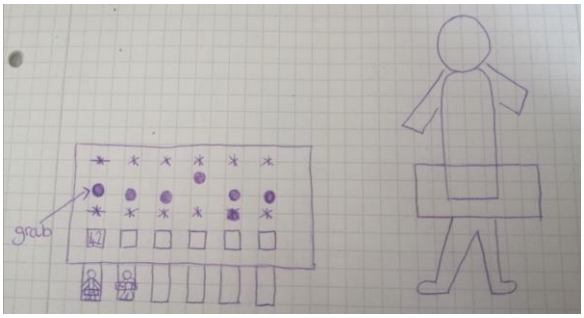


The figures above show the starting design sketch and the final project. I originally wanted to implement two levers which pivoted around a point – they would move the filtered clothes selected forwards and backwards.

Due to time constraints, I was able to implement the logic in the code but not the levers in the scene. In its place I used a click button labelled "Next".

It is positioned close to the machine and further away from the filtering panels to not confuse the user.

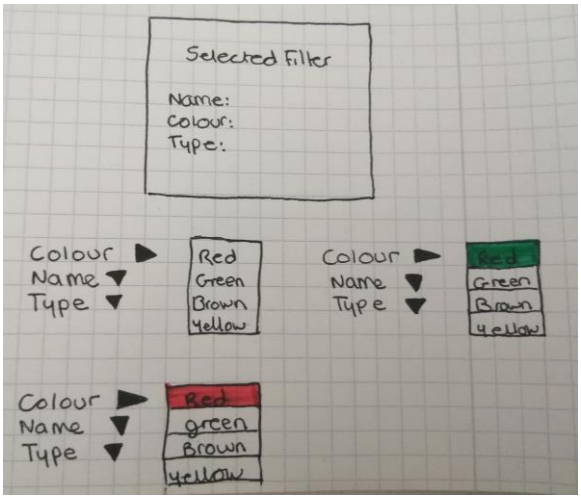
2.3.6. Model Adjustment Levers

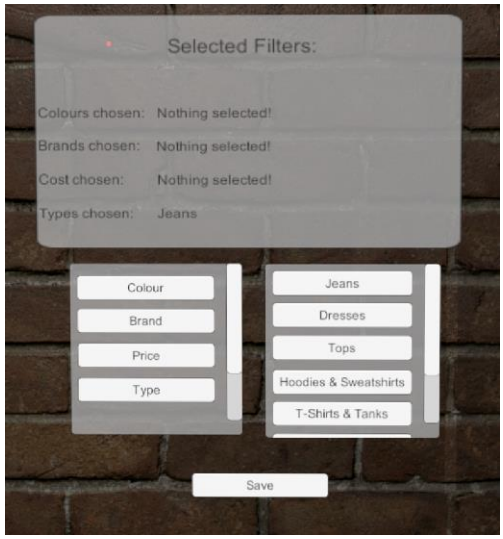


The UI on the model slider panel has multiple images attached via planes – which show the body part that would be affected. The sketches show the original design. I decided on putting them at the top of the levers so they were always in sight, leaving less confusion to the user to which bod part they were modifying.

I decided against adding additional highlights on the physical 3D model for multiple reasons. One reason is that the player should have a minimum number of concentration points, because that increases the complexity of the task. It makes it more complicated for first time users.

2.3.7. Filtering





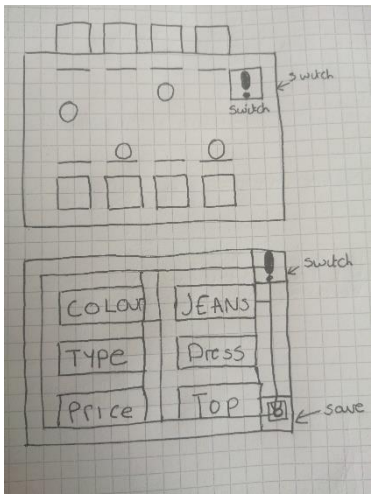
To filter information, I have created a scrolling panel on the wall next to the clothing items. This can open another scrolling panel which shows a more detailed list of options.

The chosen option goes green when it has been selected and added to the filter list. The option also turns red when the option has been removed. I chose these colours because green is associated with validation and red is associated with errors.

I designed this as so because I wanted to keep it simple – the user must process their filtering choices at this stage.

Design Variation – Filtering table

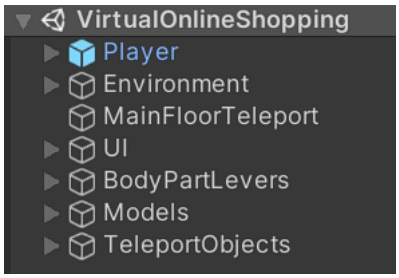
This design merged the Model Levers view with the UI for the filtering. I wanted to do this to limit the amount of moving the user has to do within the scene. The filtering view would be triggered by a button which is constant on both displays. The display would use a similar system to the implemented method – showing a new level of options that can be selected and filtered.



I decided against this design because it adds more layers of complexity. It also increases the users focus points as the user must interact with the button panel whilst also checking that they entered the correct filter inputs. I also didn't want to overwhelm the user with information panels in one specific area – The user should experience and appreciate the entire scene.

3. Implementation

3.1.1. Environment



The general environment structure uses an empty game object as placeholder folder, as does the UI folder – this helped me find components efficiently within scripts and it kept all the components organised better. See Figure blank for the final hiarchy within Unity. The origin point was also manually set at (0, 0, 0).

Lighting
The overall scene has an area light, which mimics the sun. It is positioned in the centre so it would produce even shadows. The shadows have been set to soft so they would fit in the scene and not be too harsh.

The following components have a spotlight attached:

- Clothes Machine
- Model Lever Table
- Idle Models
- Model Selection Table

The reasons for this are that I planned to use the lights to guide the user and to highlight the important components. In the future, I want to implement a tutorial-style system for using the application for the first time. This would be done by simply enabling and disabling lights in a certain order to show the user the application flow.

Window
This was achieved by copying a wall panel and giving it a transparent grey material to mimic a window. Smaller cuboids were also created to break up the window and giving it more realism. These cuboids were assigned a brown material and placed at intervals.

Poster
I created the posters by creating a cube, re-sizing it and adding the infographic as a material. I also added a slightly bigger version of the cue to use as frame for the poster, making it stand out more on the wall. I used the snipping tool to get it off the site.

Walls
The walls were made up of cuboids placed next to each other – so the texture wouldn't distort. The wall textures were taken off WEBSITE HERE - both the normal map and texture. Dragged the material to the component, went into detail view and assigned the normal map.

Floor

The floor texture was taken off Texture Haven – see appendix B i. The floor was made up of cuboids placed next to each other – so the texture wouldn't distort.

Runway and Model Stand

The model base was made by adding a cylinder into the scene and decreasing the y value until it was almost flat. It has a simple grey material attached to it.

3.1.2. Interaction

Floor Teleports

This was implemented by placing a plane on the ground and assigning it the teleport material. The teleport material is a prefab created by **SteamVR**. I also had to put a teleport listener into the room for the application to process them.

Spot Teleports

They were placed at the most important objects within the scene:

- Clothes Machine
- Model Lever Table
- Model Selection Table

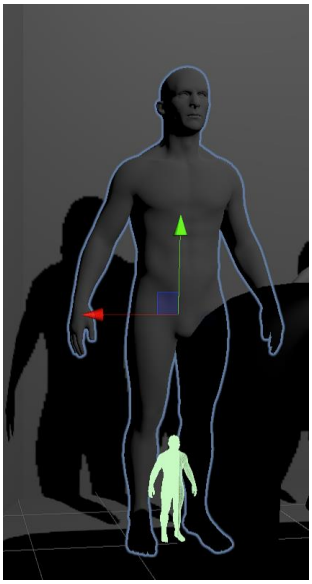
This is for ease of access for the user and gives them a visible hint about the important components of the scene.

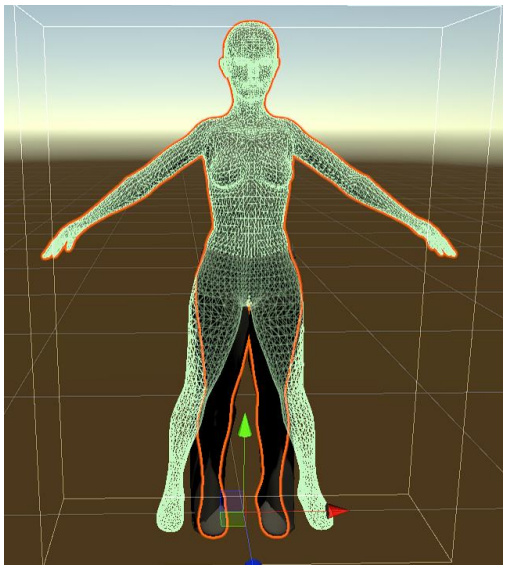
3.1.3. 3D Models

The base models of the male and female were sourced from a website called TurboSquid – see Appendix B[ii]. I did this because I needed detailed models, and this was the quickest way to prototype. The models have either got the FemaleModel or MaleModel script attached to its parent object – depending on the base model. The also have the BaseModel script attached to the parent object.

Model Colliders

I added the mesh collider before scaling the model in the Unity scene, if done otherwise - figure blank would happen.



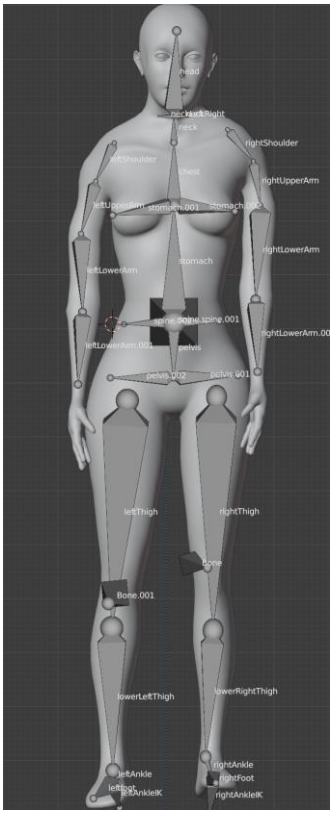


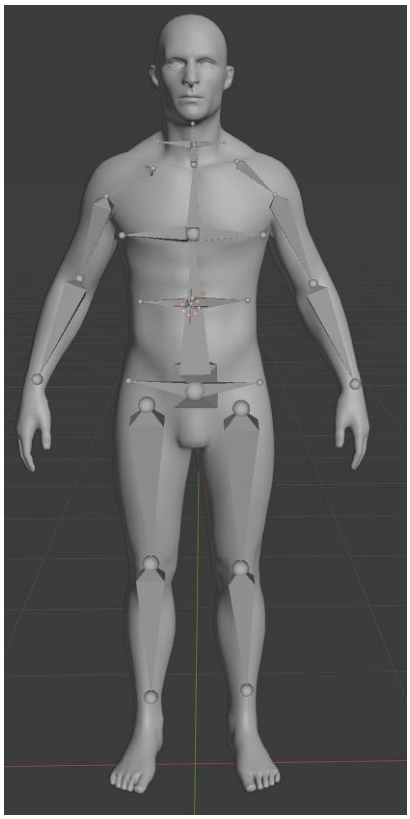
The issue originated from exporting the posed model from blender – I didn't reset the first keyframe to the posed model, meaning that the mesh body was not generated and exported correctly. I would fix this in the future – giving the application more function.

This means that the model cannot wear clothes without causing unexpected bugs.

Model Posing
Unfortunately, I had issues posing the model and having Unity generate the matching mesh collider. This meant that the clothes couldn't react to the clothes as the mesh was inaccurate. **Chosen poses in details**

Model Rigging





The model has two types of armature – one is to pose the model, the other is to change the model's dimensions.

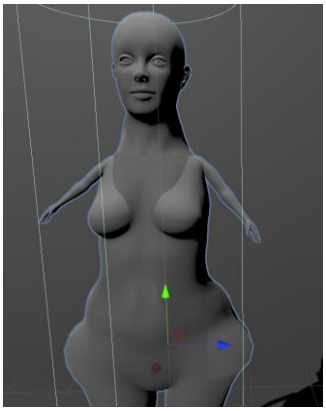
One set of bones has been placed in the key areas which are the neck, bust, waist and hips. I chose these places because they are the main areas of the body that sizing charts refer to.

The other set have systematically been put on the human body, with joints in logical places. This is so the model can be posed within Unity and Blender.

The male and female models unfortunately have different bone structures as I refined the way I structed the bones in the hierarchy. I would like to fix this in the future, removing the need for a separate class to create the BaseClass.

Model Transformations:

Transformations of armature were done on a case-by-case basis as there were many small issues that I had to fix. One of those was scaling all the different body parts. They worked in different ways as some body parts – the shoulders for example, had to be moved along the X axis rather than scaling the bone up. This is because it had child components – which made up the arm. Unity scales all components which are children – resulting in unwanted model distortion.

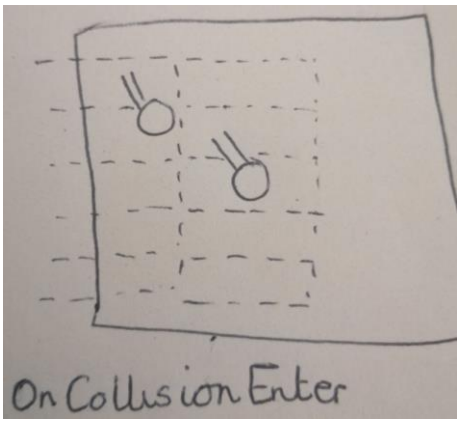


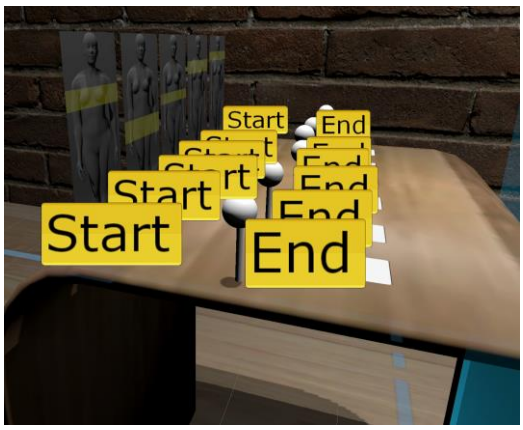
There was an issue when moving the shoulders within Unity. I had to learn and use the different transformation spaces when changing the transform of the model objects.

Local space transforms the model relative to its parent. In the first figure – this illustrates how a parent with a big difference of transformation can affect the model. Global space uses the model's transformation, transforming to the parent component to the scenes root. This is shown in the second figure. I fixed this by making sure the parent's root had no scale and rotation set.

I had to manually move and test the limits of the different body parts in order to get a clear limit of where each body part can scale. I found that the values were different for the male and female and I set them accordingly.

3.1.4. Model Scaling and Levers





I implemented this by using maths to calculate the range of the values and matching them to the range of the lever positions. This meant that the program calculated the range on start, and that it is always accurate. The advantage is that it made it a very flexible component as all it needed was the minimum and maximum integer value, and the physical handle component.

Alternative Approach – Collider boxes

This design used a system made up of collider boxes, on which I was going to manually set the levers. I soon realised how complicated this could be when I analysed the dataset. I got the data from the ASOS website via the sizing charts. See Details. I chose to use two of the tables for the project – Curve and Petite. This data proved to be too inconsistent with the ranges for me to manually set intervals.

3.1.5. Clothes Physics

I used a new package (EZ soft bones) to implement the cloth instead of using the inbuilt cloth component within Unity. I decided to use a new package for a few reasons.

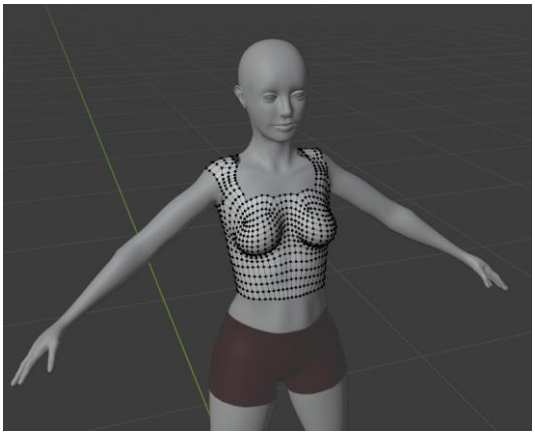
The internal package uses a system of weighting points, in which you can edit how the cloth reacts in certain physics situations. It also needs to be pinned to an object – to keep it anchored. The issue with this is that I needed the clothing objects to be dynamic in scaling and to recognise the model collider mesh component as a rigid body collider. This component is limited as it only accepts capsule and circular colliders. I am also not pinning meshes to the models as it needs to display many clothes and it is difficult/computationally expensive to do on runtime.

The new package accepts a mesh body and simulates physics by using armature. This fits in better with the project architecture as I can dynamically scale the bones in the armature without distorting the mesh too much. The only downside is that there was limited documentation, this meant it took longer than anticipated to implement.

I came across issues during development whilst trying to figure out how the components worked – it took a lot of fine tuning. For example, I gave the clothing the bone parameters of the model, instead of itself. I thought this would set the mesh collider instead of activating the cloth physics.

3D Clothes Modelling

The software I used to create the clothes was Blender 2.91.0 – the process was time consuming but returned good results. I started by selecting the faces of the mesh off the first model, expanding those into their own object and then using the sculpt tool for the fine details. Figures _insert_here_ show this.

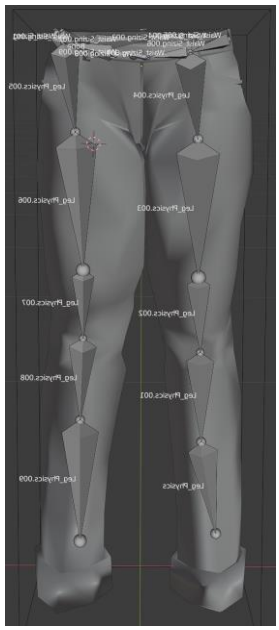


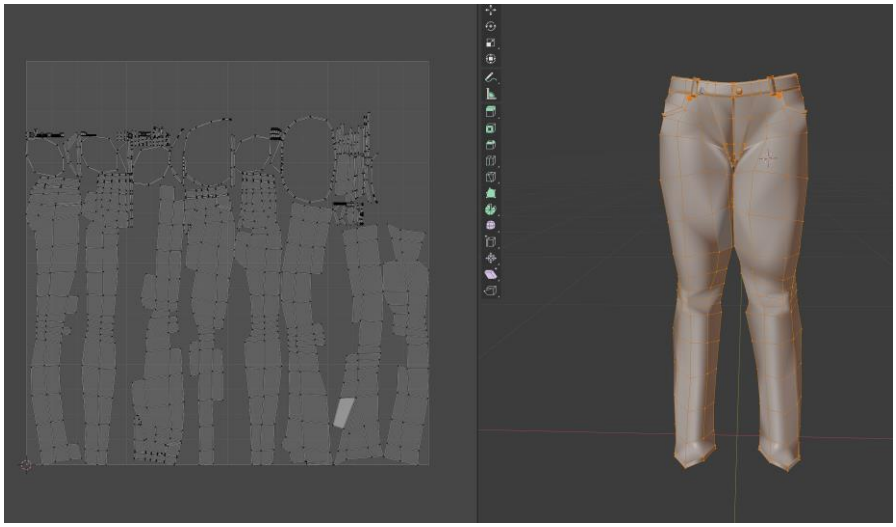
There were many issues with this as the sculpt tool could reduce the size of the clothing object – making it so it wouldn't fit the model anymore. In this process, I had to keep refering to the exported object in Unity to make sure it fit the base dimensions of the intended model.

I also had to make sure that each clothing model had the correct normals because the sculpt tool edited them during the process. I also had to add a set of faces in the inside of the model as Unity would only show the outside due to Culling. This had to be done before the sculpt process started because otherwise the model gets distorted and broken. See above figure.

I made sure to plan and add in the correct armature – following the correct naming conventions for each type. For example, the figure below shows the bones created for the dynamic sizing – I decided on using "Waist_sizing". The naming of the bone armature which controls the physics is "Leg_Physics".

These naming conventions are in place, so anyone developing the application is aware which set of armatures has been created for the physics engine and which is being used for the dynamic scaling.





My method to texture the clothes was to hand paint the UV maps. The figure above shows the two side by side before the UV map was exported to GIMP.

I used Texture Haven – see appendix B [i], to source the the base imges and normal maps for all of the clothes. For example, I sourced a jeans texture and imported it into GIMP I then changed the transparenency layer. I added another layer in which I implemented the colour.

In many of the models cases, I used the same model and re-textured them slightly as a lot of the items were very similar. For the items with logos, I either used the snipping tool on the preview image in the product page on ASOS, or searched the imternet. I then found the face in the UV map which correlates to it and pasted the logo onto it. This kept the clothing item as close to the real clothing item being advertised.

I then imported this texture into Unity and it created it into an assignable material. In Unity, I dragged the textures with the correct names (Object IDs)

3.1.6. Filtering

The clothes filtering interface was built up by several components including panels and buttons. I positioned and created all the buttons in the scene and decided to hide the parent component. This made it easier to show the options as it only takes one line to enable them.

Panels
In the script, the menus are found within the Unity hierarchy instead of being manually added within the editor. I chose to do this because it decreases the chance of Unity de-referencing the component – increasing the applications reliability.

The UI is made up of two different types of panels – a general panel and detail panel. The detail panel opens depending on which category button is pressed.

The parent of the entire UI system has the FilterUI script attached to it and that contains all the methods for the button clicks. This sets and returns variables that are processed in the controller script. It also contains the filtered lists which are created and modified.

I added the appropriate details for the clothing objects that have been loaded in. This included all brands, colours, clothing type and cost.

Buttons
The buttons work by returning the name of the button and adding it to the respective list. When the “Save” button has been activated, the parameter lists are received by the FilterController script which process it.

The parameters list from that component are used to compare the variables of the clothing objects that have been loaded in. If an object colour matches for example, then it is added to the new spawn list. The Filter Controller script uses the spawn list it created and assigns a clothing object to the Clothing Machine. Within this, it also spawns a scaled down version of the clothing object, for the preview.

3.1.7. Clothes Spawning

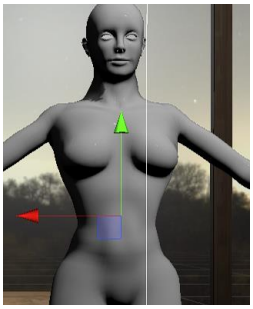
Above is the 3D Model used for the final product of the Clothes Machine. It has 2 types of buttons associated with it – Spawn and Sizing.

Spawn Button
The ClothesMachine script finds the assigned object and uses logic to spawn the clothing item onto the correct point – the figures show the position of the top, middle and bottom spawn point on the female model.

The machine gets the clothing object assigned to it and finds the correct spawn point of the active model. The spawn point is defined by a string and is assigned by the clothes holder – depending on the item type.

To get the clothing model, the file structure finds the spawning folder and assigns the spawn point to one of the children. This is found within the code and both models (Male and Female).

This is set up, so the current selection of clothes moves with the active model. It allows the users to keep their choices on the model.



The ClothesMachine script limits the model to one item of clothing on each spawn point for example. This is so the user can view outfits without spawning too many items in the same position – compromising performance. The configuration of the models clothing limits is to either have 1 top and one bottom, or 1 middle item.

This could be modified in the future easily by editing the method within ClothesMachine.

Sizing
I planned to do the clothes sizing, increasing of an interval of 1 for each setting. The bones attached to the clothes would have been scaled up and a size would have been calaculated according to the data tables. Unfortunatley, I was unable to implement this in time as I needed to research and figure out how to correctly apply those tranformations.

Next button
The next button is associated with the clothes machine as it resets clothes the previously assigned clothes, sets the new item and spawns it in the correct position.

4. Testing

4.1. Testing Approach

My approach to testing was to use a mixture of manual and integration tests – with a small number of unit tests.

I used debug statements during development to track and validate that the correct information was being passed though along with the Rider Debugger in Unity.
I tested in both 2D and VR because the hardware needed specific inputs in the scripts. Tracking hand poses for example is impossible as the hardware is not connected in 2D mode.

4.2. Automated Testing

4.2.1. Unit Tests

I planned and attempted to create the below unit tests, though I came across many issues whilst trying to do this. The main problem was the learning curve as I hadn't written unit tests before. After some research, I realised that these sorts of tests are not fully suitable as the project scripts are dependent on each other.

I struggled linking other dependent scripts to my tests, making the process more complex and confusing. I proceeded to test my application manually and I used debug statements in the code to track which methods are called.

Models:

Abstract Class:

Method	**Test**
SetBones()	Find correct gender.
SetBones()	Create correct bonelist.
Update()	Update handle value.

Model Slider:

Method	**Test**
getHandlePosition()	Get handle position.
getMinSize()	Get min value.
getMaxSize()	Get max value.
getPositionValue()	Get position value.

Method	**Test**
CreateSliderObject()	Find ModelSlider object.
UpdateHandleLever()	Update ModelSlider value.
ScaleByValue()	Get scale value.
ScaleByShoulder()	Get scale value.

Switch Model

Method	**Test**
onButtonPress()	Button clicked.
HasBeenPressed()	Return value.
GetValue()	Model value.

Set Active Model:

Method	**Test**
OnTriggerEnter()	Object enter trigger.
OnTriggerExit()	Object exit trigger.

4.3. User Testing

Test	Expected Result	Actual Result	Pass/Fail
Model Lever changes value when moved – 3D	Move within specified range	Move within specified range and hand posed around it.	Pass
Model Lever changes value when moved – 2D	Move within specified range		Pass
Display correct value when Model Lever moved	UI on the table updates with correct values as value does.	UI on the table updates with correct values as value does.	Pass
Correct body part moves when specific lever is moved.	Body part scales as expected	Body part scales as expected.	Pass

Test	Expected Result	Actual Result	Pass/Fail
Correct clothing data is loaded – console statements			Pass
Data is displayed when hand hovered		Panel is displayed but data is null.	Fail
Preview is shown in clothes machine	Item should scale down and spawn in machine.	Item is scaled down and spawned.	Pass
Tops/Hoodies spawn in correct place	Items should spawn on top spawn point on model.	Items sometimes spawn on top spawn point.	Fail
Jeans/Loungewear spawn in correct place	Items should spawn on bottom spawn point on model.	Items sometimes spawn on bottom spawn point.	Fail
Dresses spawn in correct place	Items should spawn on middle spawn point on model.	Items sometimes spawn on middle spawn point.	Fail

Test	Expected Result	Actual Result	Pass/Fail
Model moves to active zone on button press	The model associated with the button appears on active zone.	The associated model appears on active zone and is assigned active model.	Pass
Scale possible active model's body parts.	The active model body parts scale when the levers are moved.	All body parts scale when the lever is changed.	Pass
Move possible active model's body parts.			
Assign active model – console statements	The console and component display the name of the active model.	The console and component display the name of the active model.	Pass

Test	Expected Result	Actual Result	Pass/Fail
Hint button shows UI if pressed	Hint UI panel shows in scene.	Hint UI panel shows in scene.	Pass
Hint button hides UI if pressed twice	Hint UI panel hides if panel was open.	Hint UI panel hides if panel was open.	Pass
Spawn button shows error UI if no model has been selected	Error UI shows at the model spawn point.	Error UI shows at the model spawn point but only from one machine	Fail
Error UI hides after 3 seconds	Error UI de-spawns after 3 seconds.	Error UI de-spawns after 3 seconds.	Pass
Options go green when selected	Button goes green temporarily when pressed.	Button goes green temporarily when pressed.	Pass
Options go red when unselected	Button goes red temporarily when pressed.	Button goes red temporarily when pressed.	Pass
Multiple colour options can be selected			Pass

Multiple type options can be selected			Pass
Multiple brand options can be selected			Pass
One price can be selected			Pass

Test	Expected Result	Actual Result	Pass/Fail
Spawn button triggers item spawn	Clothes spawn into the scene	Clothes spawn into the scene	Pass
Jeans spawn correctly on the body	Jeans spawns on lowest spawn point	Jeans spawns on lowest spawn point	Pass
Tops spawn correctly on the body	Tops spawns on top spawn point	Tops spawns on top spawn point	Fail
Hoodies spawn correctly on the body	Hoodies spawns on missile spawn point	Dresses spawns on missile spawn point	Fail
Loungewear spawn correctly on the body	Loungewear spawns on lowest spawn point	Loungewear spawns on lowest spawn point	Fail

Test	Expected Result	Actual Result	Pass/Fail
Spawn button triggers item spawn	Clothes spawn the correct item into the scene	Application finds and spawns the correct clothing object into the scene.	Pass
Jeans spawn correctly on the body	Jeans spawn on lowest spawn point	Jeans spawns on lowest spawn point.	Pass
Tops spawn correctly on the body	Tops spawn on top spawn point	Tops spawn on top spawn point.	Fail
Dresses spawn correctly on the body	Dresses spawn on missile spawn point	Dresses spawn on middle spawn point.	Fail
Hoodies spawn correctly on the body	Hoodies spawns on lowest spawn point	Hoodies spawn on top spawn point.	Pass

Test	Expected Result	Actual Result	Pass/Fail
Spawn button triggers item spawn	Clothes spawn into the scene	Clothes spawn into the scene	Pass
Jeans spawn correctly on the body	Jeans spawns on lowest spawn point	Jeans spawns on lowest spawn point	Pass
Tops spawn correctly on the body	Tops spawns on top spawn point	Tops spawns on top spawn point	Fail
Hoodies spawn correctly on the body	Hoodies spawns on missile spawn point	Dresses spawns on missile spawn point	Fail
Loungewear spawn correctly on the body	Loungewear spawns on lowest spawn point	Loungewear spawns on lowest spawn point	Fail

Test	Expected Result	Actual Result	Pass/Fail
Can display preferences on panel.	Panel displays chosen preferences	Panel displays chosen preferences	Pass
Strings are formatted correctly	Strings are displayed in a good way – no issues	Strings are displayed in a good way – no issues	Pass
Next button increases list			
Next button changes model in clothes machine			
Clothing List loops round in clothes machine			

5. Critical Evaluation

The requirements were correctly identified as I did not come across any surprises during development. I did not meet the scope however as I hit a learning curve with the 3D modelling. Many features didn't get completed because of this unfortunately.

Most of the design decisions were correct, though the UI designs of the filtering could have been more complex – the design was the default UI component provided by Unity. I didn't prioritise it until the end of the project. I considered the texturing as polish for after all the features had been developed – which left the application a little basic.

The other designs of the stage and skybox were to my vison and gave the application a great effect of not making the user feel so enclosed and creating a "fantasy" environment they wouldn't see in real life.

A more suitable toolset could have been chosen though the tools I used were proficient in creating the software, they served their purpose. They were mostly all open-source.
I could have used a student licenced version of Adobe Photoshop and have downloaded pre-set brush sets which could have helped and sped up the texturing process.

The software met some of the needs of the users, an environment where you could filter and show clothes in VR was produced. It could be more accurate with the projects scaling, and the cloth physics could have been implemented into the final product. The concept of the project has been created – with a little more time I could have a more features polished and developed.

My project aims were mostly achieved, though I didn't do: Saving data selected, dynamic scaling of the clothing.

If I was starting the project again, I would spend less time on making scripts and architecture more structured and cleaner and focus on prototyping more 3D models and working on the clothes. I would prefer to make them myself instead of downloading and texturing them myself.

One part of the project that went well is the project management – I achieved the goals I set out in the timeframe in an organised way. My use of source control stopped any major problems with the software. An example of this is using branching; I had one specific branch for the scene changes, preventing conflicts created by Unity compiling the scene.

An issue during development was that I over-estimated the timeline in relation to the scope. I overcome a steep learning curve when it came to modelling but I was too ambitious with the features.

Another issue I encountered was the scaling of the entire application. I wanted to make it as accurate as possible, using Blenders in built scale conversion tool on each of the models. Though, as I didn't create all of the models myself – the ranges of scales increased and I had to do it by sight when in VR.

- **The models do not have the ability to pose or look like they are wearing the clothing, due to keyframe issues in blender when I pose them. I planned out the UI and positioning of them with lots of research on clothing sites.**

6. Bibliography

apidojo. (2021, April). *Asos API Documentation*. Retrieved from Rapid API: https://english.api.rakuten.net/apidojo/api/asos2?endpoint=apiendpoint_9b326d70-567d-47a9-b5cc-f1b58990e004

Baytar, F. C. (2020). Evaluating garments in augmented reality when shopping online. *Journal of Fashion Marketing and Management*, Vol. 24, 667 - 683. Retrieved from Journal of Fashion Marketing and Management: https://doi.org/10.1108/JFMM-05-2018-0077

Indvik, L. (2011, May 10). *Topshop Lets Shoppers Try on Clothes Virtually in Moscow Store*. Retrieved from Mashable: https://mashable.com/2011/05/10/topshop-augmented-reality-fitting-room/?europe=true,

Inition Digital Limited. (2014, November 17). *Topshop: Virtual Reality Catwalk Show*. Retrieved from Inition: https://www.inition.co.uk/case_study/virtual-reality-catwalk-show-topshop/

MPI IS Perceiving Systems Department. (2011). *Female Body Visualizer*. Retrieved from body visualizer: https://bodyvisualizer.com/female.html

MPI IS Perciving Systems Department. (2011). *Male Body Visualizer*. Retrieved from Body Visualizer: https://bodyvisualizer.com/male.html

Unity. (2019). *Unity 2019.4.13*. Retrieved from unity 3D: https://unity3d.com/unity/whats-new/2019.4.13

WBR Insights. (2021, February). *Future Stores East*. Retrieved from Zara's Augmented Reality App Brings Virtual Models to Life in Stores: https://futurestoreseast.wbresearch.com/blog/zara-augmented-reality-app-virtual-model-strategy

Wilkinson, L. (2021). *Virtual Online Shopping Development Blog*. Retrieved from https://lwilkinson.dev: https://lwilkinson.dev/vosBlog.html

7. Appendices

A. Third-Party Code and Libraries

i. **EZSoftBone Package –** This package was used to control the physics of the clothing objects. Version 1.6.0 was used. The package is open source, and it is available from the Unity Asset Store for free. This component was used without modification. A link to more information about this asset can be found here: https://unitylist.com/p/td5/EZ-Soft-Bone

ii. **SteamVR –** This package was used in conjunction with Unity to handle all the VR aspects. That includes input handling and managing the view for example. Version 1.17.6 was used and was used without modification. It is free and more information about the asset can be found here: https://store.steampowered.com/app/250820/SteamVR/

B. Models and Textures

i. **Texture Haven** – The reference site for base textures – they were used as an enhancement for the models. The site was https://texturehaven.com/textures/. This is an open source website which provides textures.

ii. **Mannequin Models -** The Male and Female mannequin was used within the application to be the main models for the clothing items. The models were modified in the application by scale. The model is open source
Female link: https://www.turbosquid.com/3d-models/free-obj-mode-female-base-mesh/654424
Male Link: https://www.turbosquid.com/3d-models/base-mesh-male-character-3d-model-1559946

C. Software

i. Unity – Version 2019.4.16

ii. IntelliJ Rider –
iii. Git Kraken

D. Ethics Submission

In case of any further queries, please visit www.aber.ac.uk/ethics or contact ethics@aber.ac.uk quoting reference number **19070**.

Assesment Details

AU Status
Undergraduate or PG Taught

Your aber.ac.uk email address
law37@aber.ac.uk

Full Name
Laura Wilkinson

Please enter the name of the person responsible for reviewing your assessment.
Neil Taylor

Please enter the aber.ac.uk email address of the person responsible for reviewing your assessment

nst@aber.ac.uk

Supervisor or Institute Director of Research Department
cs

Module code (Only enter if you have been asked to do so)
CC39440

Proposed Study Title
Virtual Online Shopping

Proposed Start Date
25 January 2021

Proposed Completion Date
01 June 2021

Are you conducting a quantitative or qualitative research project?
Mixed Methods

Does your research require external ethical approval under the Health Research Authority?
No

Does your research involve animals?
No

Does your research involve human participants?
No

Are you completing this form for your own research?
Yes

Does your research involve human participants?
No

Institute
IMPACS

Please provide a brief summary of your project (150 word max)

Where appropriate, do you have consent for the publication, reproduction or use of any unpublished material?
Not applicable

Will appropriate measures be put in place for the secure and confidential storage of data?
Yes

Does the research pose more than minimal and predictable risk to the researcher?
Not applicable

Will you be travelling, as a foreign national, in to any areas that the UK Foreign and Commonwealth Office advise against travel to?
No

Please include any further relevant information for this section here:

Is your research study related to COVID-19?
Yes

***18**

If you are to be working alone with vulnerable people or children, you may need a DBS (CRB) check. Tick to confirm that you will ensure you comply with this requirement should you identify that you require one.
Yes

Declaration: Please tick to confirm that you have completed this form to the best of your knowledge and that you will inform your department should the proposal significantly change.
Yes

Please include any further relevant information for this section here:

Flag key
*18) *18) Flag to denote covid-19 research