This document lists ideas, things to consider, look into and possible future endeavours for the project, anyone may add to the document and edit it, but please create a new heading below this list with your GitHub account followed by your input as this is needed by Sir306 to continue work and development on this project after the summer break.

Datasmith

1. Incorporate master materials to us material functions to control certain behaviours like on the FBX
2. Create instances of the master’s with different shading models
3. Implement functions to change datasmith
   1. Materials
   2. Add remove mesh on file changes
   3. Give users ability to turn certain datasmith assets? (i.e. hide roof mesh)
4. Create a way to map the different material shading models per mesh
5. Handle loading of mesh to be async and look at way to show load progress
6. Incorporate a way to load other cad models or give workflow to do so

Heatmaps

1. Heatmaps need to calculate for datasmith assets
2. Need a way to dynamically calculate floor levels or provide a way for user to specify number and heights
3. Refine generation to create meshes to handle 3d for all floors and switch to 3d when requested -> this is simple to do but hard to get runtime instant speeds
   1. A possible thought was to always create the number of vertices need for 3d and connect the corresponding indices for a face as I know the size resolutions of 2d and 3d so need algorithms to handle the 2d and 3d
4. Implement CVD default options
5. Need different options like a path line trace heatmap to show the most walked areas and congestion of spaces
6. Refine the cumulative map to be more beneficial and useful
7. Ensure the calculation for colour bandings are correct
8. Explore different ways to visualize 3D heatmaps:
   1. Maybe a volume one that reacts to agents
   2. Look at square 3d vertical to represent like a 3d histogram
   3. Can Voronoi heatmaps have a form of 3d representation
9. Could a grid heatmap that has values on be useful to show a density of a particular zone?

Pedestrians

1. Random colours need the hues toned down so not so visually bright, this makes it difficult to see heatmaps buildings and other people
2. Generate open-source mesh using blender or find model
3. Create the procedural animation to create VATs and use for when skeletal meshes are used
4. Incorporate a colour picker for pedestrian material colours?

Lighting

1. Imported datasmith assets seem to have the typical lumen lighting problem, need to look into where it is occurring. A thought is that the lights that get imported with the model are static and need to be dynamic
2. Sky light is horrible it needs a complete overhaul as the default is no good
   1. A diffuse skybox is really needed – look into the how
3. From demoing and experimenting being able to turn the global light off for datasmith assets is very powerful and shows how a building is lit internally this will be useful for users to see and use
4. Maybe provide UI controls for sky options like light position, colours and etc

MassAI

1. Lots of areas could do with some code cleanups and optimization, whether it be refactoring by creating methods for duplicate code or writing better for loops or custom logic to give better performance
2. Need to move the rendering of agents into Niagara this provides better GPU performance on the rendering
3. Agents need more animations
4. Need to find out the optimal number of skeletal meshes that can be visible in a scene and use IK or the procedural animation
5. There is no LODs being used on the current meshes, need to look at if VATs work with different LODs and if not create the required variations for different LODs and need to look at adapting camera distance to apply the correct LODs and animations as we likely want to see some sort of representation of a model a person but likely don’t need to animate, this is a problem that will become apparent when we look at things like stadiums
6. The entity fragments are quite large and it may be better to decouple the fragments into smaller ones, and possibly create more processors to handle different processing tasks and calculations, e.g. we should have a processor that handles that just reads a fragment and applies the animation and new location, another that calculates the new animation fragment information and another that updates the heatmaps rather than them all on one processor.
7. It may be desirable to open discussions about use of vehicles being a part of the viewing of simulations

HDF5

1. Need to implement HDF5 into unreal.
2. Work out how to translate HDF5 data into unreal data i.e. turn data into array of movement locations for mass ai
3. Work out the loading of DLLs
4. Creating code only modules
5. Learn how to use CMake to create the packages/libraries for builds

OpenCV

1. Work out the loading of DLLs
2. Creating code only modules
3. Learn how to use CMake to create the packages/libraries for builds
4. Use an external newer OpenCV library
5. Implement logic to create Voronoi heatmaps using OpenCV, the currently disabled Voronoi heatmap was discontinued when I couldn’t implement a way to up the performance of the generator which the plugin is abandoned and uses tons and tons of loops

Interested Party Topics

1. Different geometry formats such as WKT -> this format needs a polygon to triangle conversion method to work with the mesh generator
2. Create smoke/toxin clouds -> look at ways other parties and software declare this data
3. Door counter/path flow UI to display statistics to show useful information like how many agents go through a door
4. Health bars for simulations that are looking at smoke/toxic layer influences
5. Would be good to be able to generate an exportable png of things like heatmaps at any given time
6. What would users like to be able to add dynamically to the simulation: lights, geometry, events (i.e. Natural Examples: Tsunami/wildfire/earthquake, Non-Natural Examples: Shooter/ Chem lab accident/ arson)
7. Custom avatars/materials?

Documentation/Readings

1. Check all assets used and what is the licence used or licenced should be displayed with those assets if able to use (Currently the avatars need to be replaced to be able to share the source code)
2. Monitor EULA changes to software
3. Provide User Document – including control, manual, terms of use on certain parts
4. Add the correct licences to the repo and specify where if not open source, such as some may require stating creative commons for certain external assets or plugins
5. Add appreciation/sponsors/references to the page and splash screen??
6. Need to read and learn more about GPU programming to leverage more performance and control around certain aspects of the project
7. Ideally find a way to instance skeletal meshes on the GPU
8. Learn about buffer loading data so simulation data can be queried for quick look ups and only store a buffer amount and current buffer for simulations -> currently the whole simulation is loaded and stored across a shared memory management but this could become large and unusable or unstable with larger data sets, while having it all in memory makes it able to instantly change the current timestep frame it could be made to look at the data more efficiently if we store important information (i.e. we know simulation total time and the frequency of data so when a time is picked we can go to an index of the closest sample to achieve the same effect and load that and corresponding buffers)
9. Need to bring back the destroying of agents and respawning of agents to capitalize on hardware usage)
10. I am setting this up to conduct my research project to operate in VR why not bring it in and learn what optimizations would be needed for it to work on larger simulations and how we could provide a way to navigate the environment effectively and prevent cyber sickness on long term use (this isn’t a problem that has been solved by the industry but steps on how to mitigate it)
11. If we are adding ways to simulate events, will the events have triggers for users like epilepsy, what design techniques and preset knowledge is out there to mitigate this?

Nicks Ideas/Thoughts

1. We have a way to load the geometry and predefined agent trajectories, why not include a way to conduct live experiments for users wearing VR and/or Mocap to add more dynamic data – would this mean the agents inside should react to the user as well?
   1. If implemented we must export this data to give researchers the ability to look, review and replay it.
   2. Theres a technique to shift the environment by a few degrees or reorient geometry to flip under certain conditions that allows specific space to be used to give the participant the feel and trickery of endless environment – would adding this logic be beneficial for given the option on using live participants in a simulation for large environments or will this skew certain information like walk cycles? If only using head position, then no but would need to use a filter to convert it to what the actual participant thought they did
2. If we show dark environments, should we give an option to slow down speeds given an areas level of lighting or assume the user will give simulation data to do this?
3. Currently we are using materials and material instances, but could they be improved by converting them to use virtual texturing, I have used them in the past and with some work can provide really good optimizations. Do virtual textures work with vertex animation textures and techniques?
4. I should decouple the logic that drives calculations into there own classes so they can be referenced by the things that use them and simplify current classes while also providing users who may want to verify or tweak the code to their needs
5. Ideally the file and folder need to be addressed and cleaned up, but due to rapid prototyping this got slightly neglected
6. Look into the performance of mega lights and the cost they have – Epic markets them as able to add as many lights as you want but there has to be a cost and draw backs that come with using them like if we use it will that stop dynamic lighting and the ability to add lights dynamically at runtime.
7. Need to look at what is the bare minimum information to be supplied on pedestrian data, and how to create or circumvent missing information like could we just supply number of agents and locations per time step without rotation? The math to work out is doable and can be simple if agents are constantly moving forwards but side stepping would need more work