This is to be a working drive for the 2018 SURE project on TB ABMs using FLAMEGPU

I suggest we keep brief notes here of meeting outcomes and what’s being worked on

---

Day 1: suggested

* practicalities, working location & resources, meeting tempo
* FLAME GPU resources and relevant examples to work through
* discuss one aspect of population modelling and work on coding up
* other thoughts?

*Meeting Pete@ 10 for 30 mins*

*Meeting Pete & Mozhgan @ 14 for 30 mins*

*(both rm 2004)*

**10am, Monday 11th June**

Normally Flame GPU is used with a static XML file describing the population which is then used for iterated simulations, whereas we want to create the population using Flame GPU itself and then simulate based on that population, so our distribution covers all relevant variables.

First problem is, given data containing information about the number of people in various sex/age categories and the number of households of particular sizes, assign each person to a household to create a sample population.

Can do this by dividing the output table into blocks, then the first section will be an identity matrix as we can reorder the people arbitrarily, with the remaining sections being Latin hypercubes.

**2pm, Monday 11th June**

Flame GPU branch v1.5 contains features like host agent creation which will be necessary for the project, but for now we should work with the master branch and the given examples to become more familiar with how the system works.

First thing to work on is the tutorial sheet on the Flame GPU website using the predator prey model as an example, to get used to compiling and running the Flame GPU code and writing in C to add my own features to the model.

Can get help with Flame GPU from the PhD students in the graphics research lab, and example code can be compiled and run from within Visual Studio on the Windows machine there.

**10am, Wednesday 13th June**

Pete was off sick today, so he couldn’t make it to our meeting.

**9am, Friday 15th June**

The approach used to assigning people to households works fine, although it should be tested to make sure that the assignment output matches the original histogram that was used as input.

Need to try and get an understanding of how Flame GPU works, for example where the XML file is generated from and eventually how the iterations are run and a visualisation is produced.

Eventually, should look at the host agent creation in the new version of Flame GPU as this will be relevant for our project, when we generate our population before running the simulations.

**Weekly Reflective Log, Week 1**

The first week of my project has gotten off to a good start. I coded up a solution to a small problem relating to the project, that assigns people in various age and sex categories to households of various sizes, and started off getting to grips with Flame GPU which is the tool I will be using as the project moves forward. I need to make sure that I manage my time effectively and develop a good understanding of the various factors relevant to the project, so that my solutions stay applicable to the real life problems we are trying to solve. Next week I will try to add more factors to my code simulating people in households and also develop a better understanding of how Flame GPU functions.

**2pm, Monday 18th June**

Managed to get the test functioning so that the households algorithm definitely produces an output corresponding to the data that was input, and the right number of people is assigned to each household.

Next major problem will be understanding the various types of building that people visit on a daily basis and incorporating these into the simulation, with some being buildings that people tend to visit at the same time each week and some being buildings that people visit sporadically.

Also need to look at close contact between people in the community and the factors affecting it, for example how people within an age category tend to primarily interact with people in the same age group.

**2pm, Tuesday 19th June**

Got a much better understanding of how projects are created within Flame GPU, including the folder structure and how files should be arranged, and initialised my own project using dummy code from one of the examples.

The host agent creation example mostly makes sense except for some parts where memory is allocated for creation of agents, which will take some more time to get to grips with due to lack of experience with the way memory is allocated in C.

Should implement an example program where agents have an age and a probability of death proportional to their age, as this will have a known result so we can test if the host agent creation is working properly.

**2pm, Wednesday 20th June**

Program with aging and probability of death that scales with age has been implemented, although the existing birth rate makes it difficult to see the effect that changing the parameters has.

Should change the name of death probability to time step as this is a more accurate representation of what the variable means, and get rid of the birth rate so that the effects of changing parameters are clearer.

Also, should try to get a sense of how efficiently the program runs, by trying to calculate the amount of time taken per agent per time step, so that we can get a good benchmark for how fast the final project will be able to run.