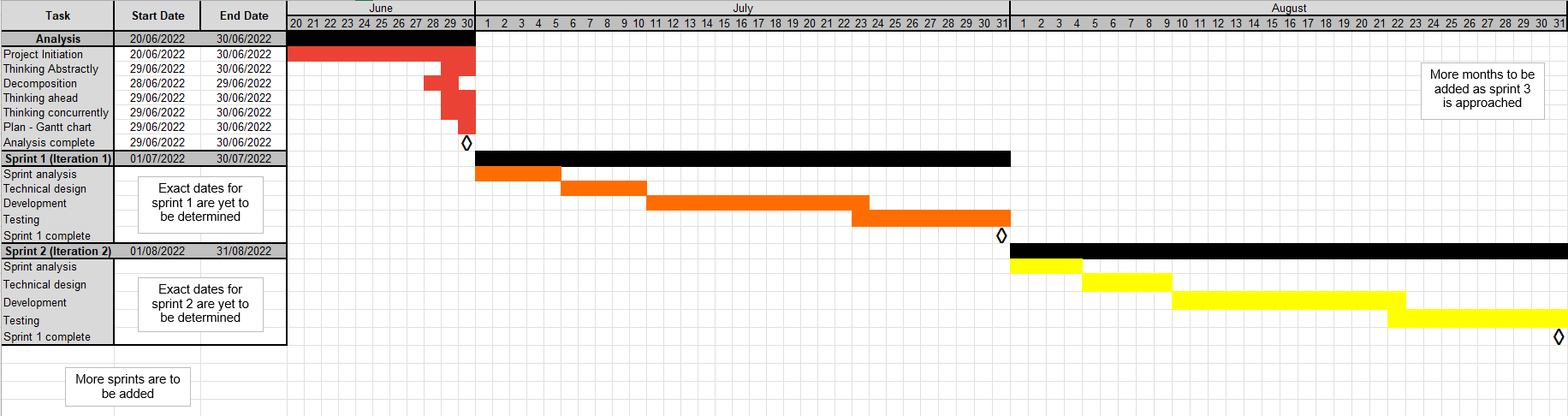
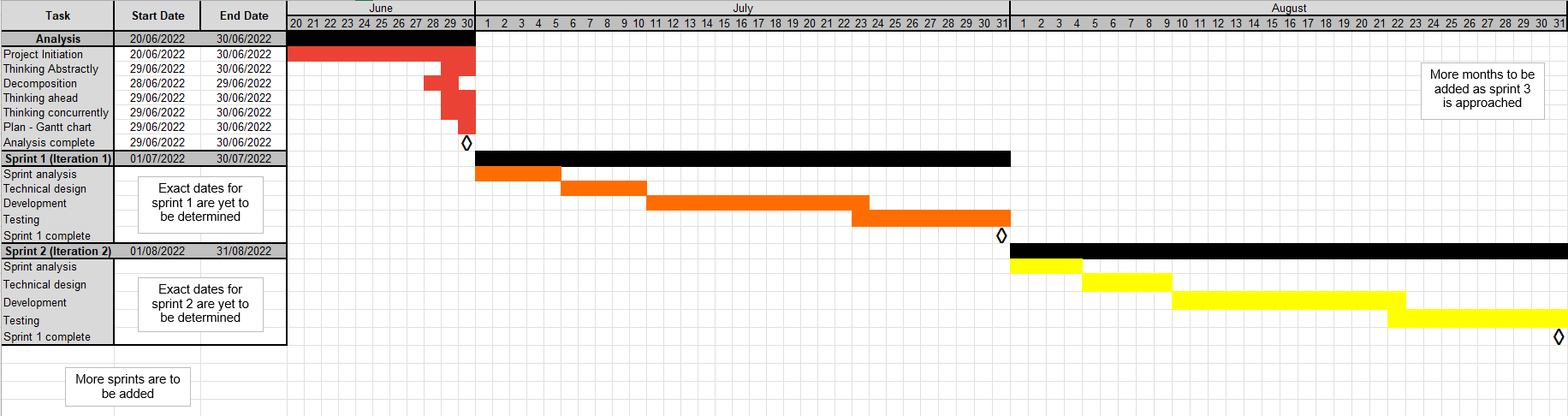
High Level Plan

Fractal Explorer, Louis Durston-Wyatt

# Development Lifecycle / Gantt Chart

The project will follow the agile development life cycle, with 3 to 5 development ‘sprints’ before the product is completed. Each cycle will take approximately 1 month to be completed, with the final product being mostly developed by December 2022; this allows constant working prototypes to be developed and improved upon using the guidance given by stakeholders.

The below Gantt chart shows the progression of the project development from sprint to sprint, for the first three sprints (including sprint zero, the analysis stage):



Chart

Description automatically generated

# Decomposition

Decomposition is the breaking down of large, more complex problems into smaller, more manageable ones. This is a crucial tool in programming as it allows different tasks to be split into independent subroutines, which are linked together by the logical structure of the program. In this project, I am using decomposition through my decomposition diagram, in which I have decomposed the features of the program into boxes, with a flow chart representing the program logic that links them. This will help me to focus on more specific tasks, and provides natural checkpoints for development of each section.

This decomposition diagram can be seen below:

Timeline

Description automatically generated

# Thinking Abstractly

Abstraction is the process of removing the excessive, unnecessary details in a problem, leaving behind the key features. When thinking abstractly, one must analyse what details are relevant to the problem. In this project, I will use abstraction to exclude details such as information about the user, which is unnecessary for the project.

# Thinking Ahead

Thinking ahead about how the different components of a problem are structured (such as inputs, outputs, and subroutines) helps developers build programs that are intuitive to use and helps to save development time/resources. This includes creating reusable subroutines (saving development/testing time) and setting preconditions (requirements which must be met before a subroutine can be executed). An example of how I will think ahead in this project is by keeping in mind what data/inputs will be passed from the Java Explore program to the HSHL Cinematic program.

# Thinking Concurrently

Concurrent thinking is the process of completing multiple tasks simultaneously, by giving slices of time to these different tasks; this can include the use of multi-threading. Thinking concurrently will be crucial in the HSHL compute shader part of my program’s development, as millions of inputs to an equation will be iterated simultaneously on the GPU.