Design Diagram

Main Controller

Graph (Main Application Thread)

Game GUI (Separate GUI Thread)

Graph initializes all threads. Agents also communicate back to Graph when new agents are created

Sensor (Multiple threads)

Uses Blocking Queue to communicate with neighboring sensors

Agent (Multiple Threads) Can communicate with sensor it is on

Log (Same for all threads)
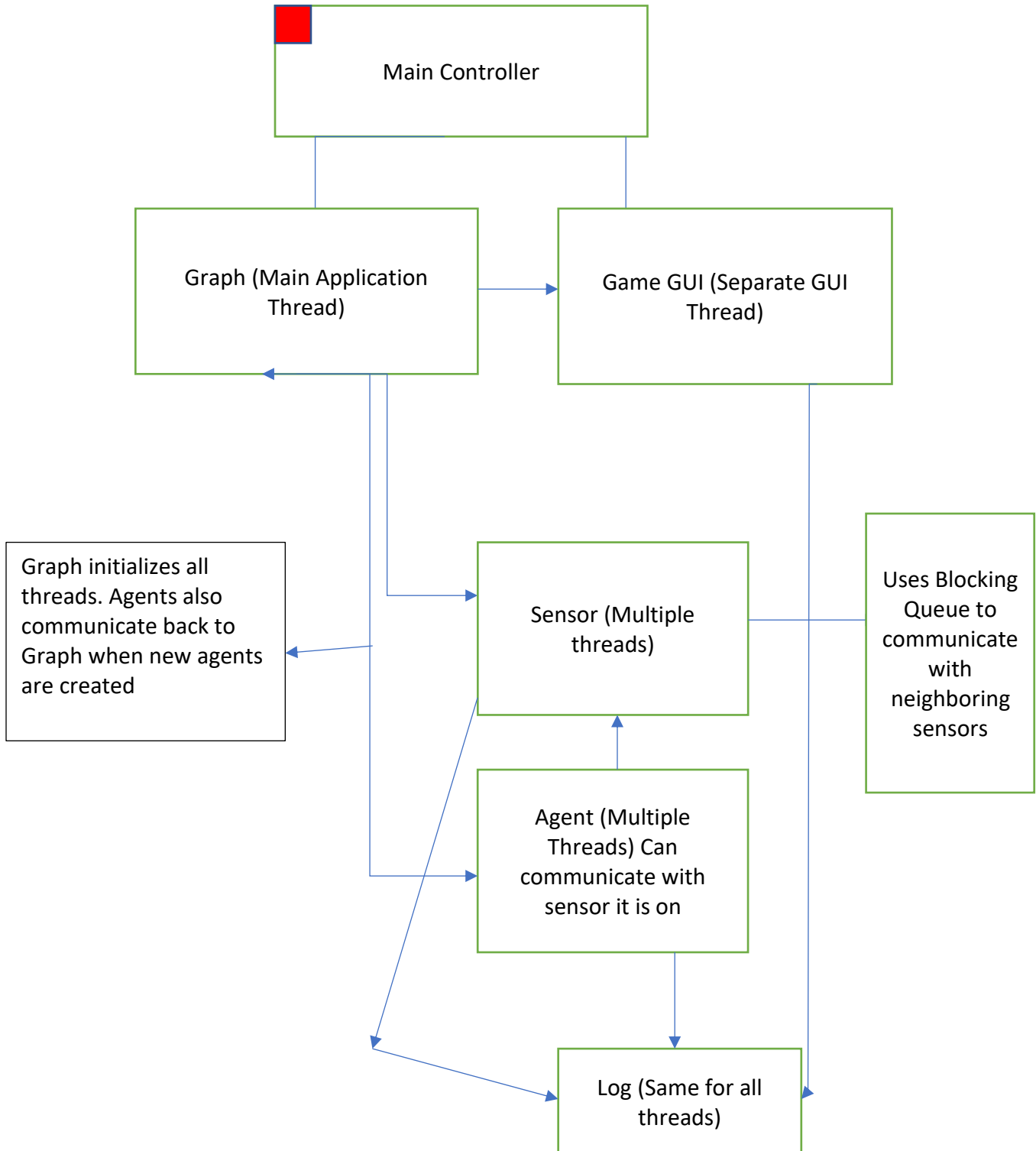
Design Description

<u>MAIN:</u> Maintains the order of the simulation. Brings together classes Sensor, Agent, Log, Graph and GUI.

<u>SENSOR:</u> Contains information of each sensor in the graph, such as the sensor's state(blue, yellow, on fire), whether sensor has agent and also, each sensor informs its neighbors whether it is on fire or not. Random fire(in case of agent not reaching yellow node) starts from here.

<u>AGENT:</u> Contains information of each agent created in the simulation, allows agent to reach the first yellow node and allows agents to clone themselves.

<u>GRAPH:</u> Reads Configuration file, determines scalability according to size of graph and initializes threads for sensors and agents.

<u>GUI:</u> Takes the information contained in classes Sensor and Agent to build a visual representation of the simulation.

<u>LOG:</u> Helps to display messages from the simulation in the GUI.