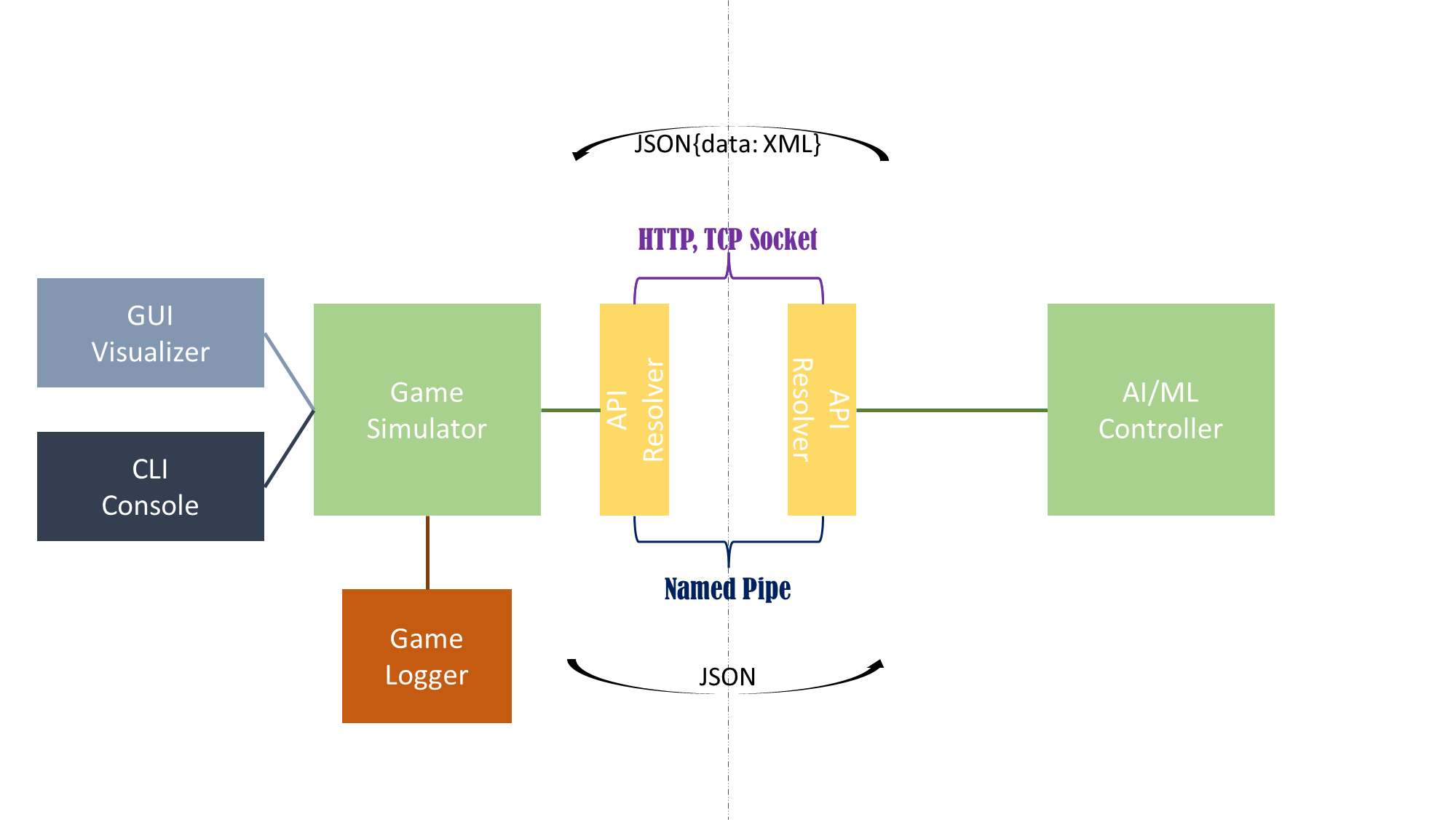
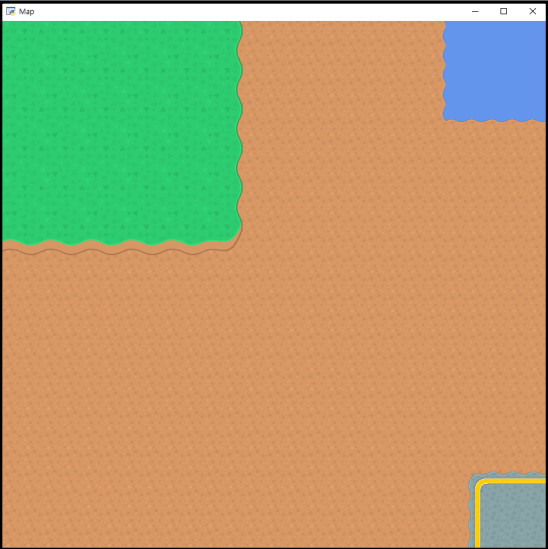
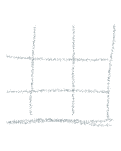
1. Overall Explanation
   1. FOS
      1. The **Fire Operational order Simulator game interface (FOS)** is a program that allows the development of an AI system that can provide operational orders for suppressing fire in a simulated environment.
   2. Internal Architecture
      1. AI/ML Controller: AI sample solution that can solve the game of the simulator.
      2. API Resolver: Layer that abstracts the user-selected API method such as HTTP, TCP Socket and Named Pipe(optional)
      3. Game Simulator: Fire Disaster Simulator which get input from API Resolver, calculate current status, and creates data that can be used to visualize game. For game visualization, the game simulation code should be completely abstracted and operate without problems in both the GUI and CLI environments.
      4. GUI Visualizer/CLI Console: It can be both a GUI window and a CLI console according to the interface mode settings that the user selects.
      5. Game Logger: Game Tracker/Logger that records the game progress in detail. (Required to discuss the need for real-time data transmission later)
   3. Interface
      1. The **Game Screen** has a resolution of 54 by 54. On the upper-left corner is the fire area, where the fire randomly occurs. The starting point is stationary at the lower-right corner of the screen, while a lake is located at the upper-right corner.







텍스트이(가) 표시된 사진

자동 생성된 설명





















* + 1. The **Controller** is where the operation orders are made. It is internally connected to the Game screen so the result can be visually showed there.
  1. Aircraft
     1. The program features five types of drones, helicopters, and airplanes, each with unique properties such as cost, velocity, and the ability to suppress fire.
     2. To avoid obvious aircraft selection options, the properties of each aircraft are adjusted to have pros and cons relatively.
     3. When each round starts, every aircraft gets a random value of water filled. After the aircraft returns from an operation the value is 0, and the value increases as times passes according to a formula. Once an order is made again, the water filling process stops and preparing time will begin.
  2. Fire Area
     1. Fires randomly occur on the 3 by 3 square located in the upper-left corner of the Game Screen. Each portion of the square has its own coordinates.
  3. Rounds
     1. The program has three rounds. The first round has one fire randomly generated on the 3 by 3 square. The second round has two, and the third round has three fires randomly generated.
  4. Review Criteria
     1. The best AI system will be determined based on its performance in terms of the least amount of time and cost required to suppress the fires in each round.
     2. Total Score: (ex) time^2 + log(cost) [less is better]

1. Data Tables
   1. Aircraft Spec Sheet

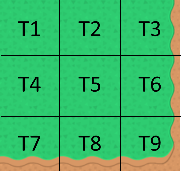
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID# | Aircraft | Velocity  (km/h) | Estimated Time to Ready  (min) | Cost Of Operation  (USD) | Cover Area | Time to Full-Fill Water Tank (on base)  @ X%(min) | Possibility to Suppress the Fire (when Water Tank is full)  (%) |
| D1 | Drone | 80 | 10 | 500 | 0=None | 7.2 - 0.072x | 60 |
| D2 | Drone | 100 | 12 | 600 | 0 | 6.8 - 0.068x | 70 |
| D3 | Drone | 50 | 8 | 400 | 0 | 7.5 - 0.075x | 50 |
| D4 | Drone | 90 | 10 | 550 | 0 | 6.7 - 0.067x | 65 |
| D5 | Drone | 70 | 9 | 450 | 0 | 7.4 - 0.074x | 55 |
| H1 | Helicopter | 200 | 15 | 1000 | 1=1Tile | 6.5 - 0.065x | 75 |
| H2 | Helicopter | 190 | 14 | 950 | 1 | 6.8 – 0.068x | 70 |
| H3 | Helicopter | 210 | 16 | 1100 | 1 | 6.1 - 0.061x | 80 |
| H4 | Helicopter | 195 | 14 | 900 | 1 | 6.7 - 0.067x | 70 |
| H5 | Helicopter | 205 | 15 | 1050 | 1 | 6.3 - 0.063x | 80 |
| A1 | Airplane | 350 | 45 | 4000 | 2=Diagonal | 4.0 - 0.04x | 88 |
| A2 | Airplane | 280 | 50 | 3500 | 2 | 3.6 - 0.036x | 90 |
| A3 | Airplane | 390 | 45 | 5200 | 2 | 3.4 - 0.034x | 97 |
| A4 | Airplane | 270 | 60 | 3200 | 2 | 3.6 - 0.036x | 85 |
| A5 | Airplane | 300 | 15 | 4500 | 2 | 3.2 - 0.032x | 95 |

* 1. Unit Table

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ordered | Available | Estimated Time of Return | Estimated Time of Departure | Estimated Time of Arrival | Base | Aircraft ID# | Current Water Level |
| False | True |  |  |  | A | D1-A |  |
| True | False | => value | 0630 | 0730 | B | D1-B |  |
| False |  |  |  |  | C | H1-A |  |
|  |  |  |  |  |  | H2-B |  |
|  |  |  |  |  |  | A5-B |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

* 1. Target List

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Targeted | Target Name | Priority | Latitude | Longitude | Type | Threat | Estimated Probability of Success |
| True | T1 |  | 280 | 20 | Fire |  | From 100% (decrease by time) |
| False | T2 |  | 280 | 60 | Fire |  |  |
|  | T3 |  | 280 | 100 | Possible Fire |  |  |
|  | T4 |  | 240 | 20 | None |  |  |
|  | T5 |  | 240 | 60 | … |  |  |
|  | T6 |  | 240 | 100 |  |  |  |
|  | T7 |  | 200 | 20 |  |  |  |
|  | T8 |  | 200 | 60 |  |  |  |
|  | T9 |  | 200 | 100 |  |  |  |
|  | Lake |  | 270 | 270 |  |  | N/A |



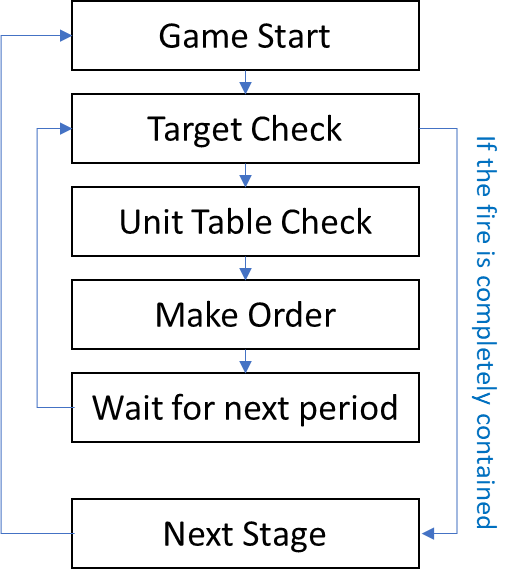
* + 1. As time goes on, estimated probability of suppression of each target is going to lower.
    2. Level of threat will be randomly selected at the beginning of each play and represent possibility of spreading of fire. And ‘Priority’ comes from the level of Threat.
  1. Operation Order

|  |  |  |  |
| --- | --- | --- | --- |
| Order No# | Aircraft ID# | Mission Type | Target |
|  |  | 1 | T1 |
|  |  | 2 | T1, T2 |
|  |  | 3 | T2 |
|  |  | 4 | T1, T2 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

* + 1. Mission Type: [1] Direct to LOCATION, [2] Indirect to LOCATION, [3] Water Fill from Lake and [1], [4] Water Fill from Lake and [2]

\*Drone only can be type 1

1. Flow Chart



1. Order Sequence
   1. Operational Order
      1. Each order consists of route, type of aircraft to use based on their properties and whether to fill the aircraft’s water tank from the lake or fly directly to the fire.
      2. An order is made every specific(20min) period. Even though the order made on the previous period is on process, a new order will be made.

Diagram, table

Description automatically generated

0600 0620 0640 0700 0720 0740 0800

* + 1. Once an order is made, the aircrafts cannot be controlled anymore before the operation ends. (Available after ETR of each aircraft)

1. Application Programming Interface between Game Simulator and AI Controller
   1. HTTP
   2. TCP Socket
   3. Named Pipe (20% faster than TCP Socket)

#. No GUI Mode

* We need to adapt at least two API method to help the participant under the various situations like no visual screen env or that wants to run simulator and controller on different devices.
  + - * [POST] register:
        + Body : {}