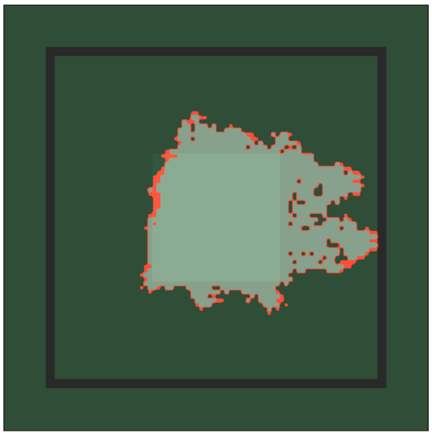
Test-Planning Document

We consider two overall requirements in the Fire Spread Simulation Software. The first is a safety property and the second is a requirement on a accuracy attribute of the system.



- R1: The state of the cell that the Firefighter deployed can not be burning.
- R2: The cell with burning state won't be set as burning again during each iterate and the burnt cell won't be able to change to any other state during each iterate.
- R3: The cell with zero flammability will stay as flammable during the entire simulation.

Priority and Pre-requisites

- R1 [High Priority]: This is a safety requirement The correctness of the software's predictions can significantly affect public safety in the case of a fire emergency. Therefore:
  - We should devote a reasonably high level of resource to ensuring we meet the requirement.
  - According to the Chapter 3, Since the Accuracy is high priority then it requires at least two different T&A approach:
  - In accordance with the principle of partitioning, the requirement 1 has been broken down into two distinct components: the control component, responsible for determining the appropriate time to deploy firefighters, and the fire detection component, which ensures that requirement R1 is upheld at all times within each cell. The deployment command is subject to the limitations imposed by the fire detection component, which must return a value of "False" before firefighters may be deployed. Additionally, the fire detection component will be executed in conjunction with any attempt made by the control component to deploy firefighters.
  - The simulation software should utilize a straightforward or conventional internal data format for optimal visibility. Additionally, to gauge progress or evaluate performance against established objectives, the software should return the number of burning cells, flammable cells, and burnt cells after each iteration. Furthermore, the software should also return the Fire Occurrence parameters for potential future fire points to ensure that the results align with cell state at next iteration.
  - The Inputs and the outputs of the cell state are
    - Inputs:
      - Flammability: A continuous parameter with lower bound 0 and upper bound 1.

            - Burning time: An integer variable, "t", represents the number of iterations after which a cell will cease burning from the time it starts burning.
            - Initial state: Initial state represent the state of the cell before simulation. (E.g. Initial Fire location)
            - Raining: Is true if is raining false otherwise
            - Wind: Is true If the wind is large enough that need to be take account into the simulation, false otherwise.
        - Outputs:
            - Burning: The value in that cell equals to 1 if the cell is burning
            - Burnt: The value in that cell equals to 2 if the cell is Burnt
            - Flammable: The value in that cell equals to 0 if the cell is Flammable
            - Fire Detection: Ture is the select cell is burning, false otherwise
        - Specification:
            - Firefighter_ Deploy is only true when Fire Detection is False and the FireCounter = 0.
    - Focussing on R1 there are two task need to schedule in the plan:
        - A review or inspection of the Fire Detection component should be conducted to verify that the requirement has been properly implemented by the specific code and that there is no irrelevant code present that does not contribute to fulfilling the requirement.
        - Additionally, after each iteration, a test should be made between the previous iteration's parameters and the current cell state to ensure that the correlation between the output and the input is logical and consistent.
- R2 & R3: The requirements R2 and R3, which pertain to the accuracy of the software, are of lower priority compared to safety requirements and thus, less effort will be placed on these requirements during development.
    - This type of testing can only be effectively conducted at a later stage of the system development and would likely be considered a system-level test. Furthermore, it is crucial to consider validation issues when implementing this test and that the validation process may require the logging of the system's performance.
    - This suggests the **following tasks** need to be scheduled into the testing:
        - Generating synthetic data to test the cell state component and flammability component.
        - Scaffolding: Building a simulation environment to mimic the fire scenario, allowing for early testing using synthetic data.
        - Designing the logging system to capture real performance
        - Designing and implementing analytics for the fire simulation results data.
        - Incorporating collected data into early testing

Process

As per the requirements, early stages of development should focus on generating synthetic data and determining the format for representing the fire scenario, cell state, and firefighter information. This can be done concurrently with other activities. Further development and testing will be conducted closer to deployment and during operation.

Collecting real-world data is also crucial as it often requires significant time and communication with relevant departments to obtain accurate geographical data, and ensuring that the data format used in the software is widely recognized.

Risk

- R2&R3: The synthetic data is may unrepresentative, and this will cause the inaccurate when test the accuracy of the software.
- R2&R3: The scarce resources available for testing since the data format that used in the simulation may differ from the format that widely used.
- R2&R3:Generating a representative dataset for testing the software may require significant resources.