Fire Commander 2020: Heterogeneous Firefighter Agent Reconnaissance Environment

Special Problem Student: Xiyang Wu
Supervision: Esmaeil Seraj (PhD Student) and Matthew Gombolay (PhD)

Institute for Robotics & Intelligent Machines
June 16th, 2020

CORE
ROBOTICS LAB

Georgia Institute of Technology

Background

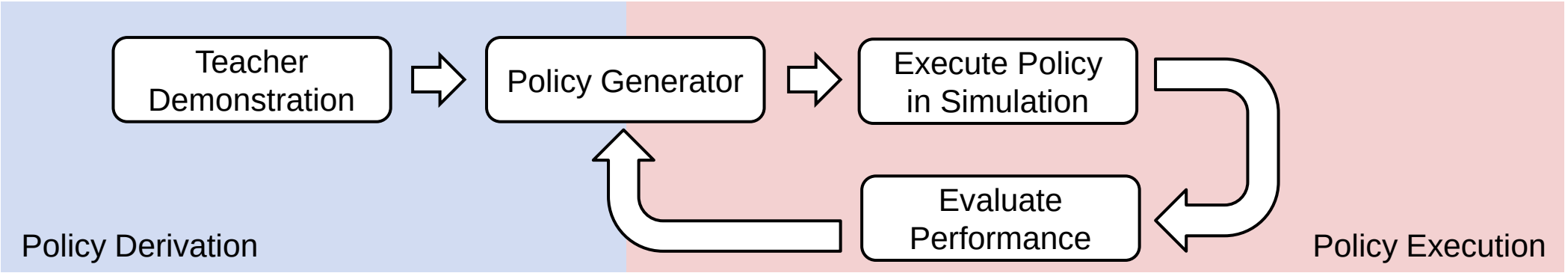
- **Wildfire**
  - Massively Destructive, Life-threatening
- **Fight Against Wildfire Manually**
  - Dangerous, Restricted by the Sensing Ability
- **Introduce Firefighter Agents to the Firefighting Task**
  - Provide the high-quality and real-time information
  - Protect human firefighters from the most dangerous tasks
- **Control Policy**
  - **Maximizing the Pruning Coverage:**
    - Lack of prediction ability for the fire state
  - **Adaptive Extended Kalman Filter (AEKF):**
    - Provide some sort of prediction ability, need to be optimized
  - **Learning from Demonstration (LfD):**
    - High Efficiency, optimal and feasible to human-robot interactive scenarios

Introduction

- **General Requirement**
  - **Investigation**
    - Investigate the application of the LfD method on the agent group control
  - **Environment Design**
    - Simulate the wildfire pruning environment, including the wildfire propagation model, target design, agent's motion policy, goal selection and trajectory mode
  - **Data Storage**
    - Design the proper data storage structure for scenario recording and animation reconstruction
  - **Reward Policy**
    - Determine the comprehensive reward policy that considers the agent's performance, the target's importance and damage, and the coverage of the wildfire
  - **User-friendly Interface**
    - Generalize the code, design the GUI, enable the versatile scene design

Control Policy

- **Agent Group Control Policy: Learning from Demonstration (LfD)** [1]
  - Develop optimal agent control policy by learning from teacher demonstrations, including the expert and non-expert teacher, or other basic control policies like the AEKF method
  - Map the learnt policy to the new scenario, optimize the policy generated with iterative method
    - Execute the control policy in the given scenario
    - Evaluate the policy's performance in the previous iteration, further optimize the policy
- **Merits**
  - Incorporate the strength of the traditional group control method and human experts, derive out a more optimal control policy
  - Adaptive and robust to uncertainty, feasible in the environment in lack of information



Developing Package

- **Simulation Environment: PyGame**
  - Popular video game design framework, mature community
  - Simple but well-designed module, satisfy the demand for interactive visual environment
  - Portable and small amount code, fast response

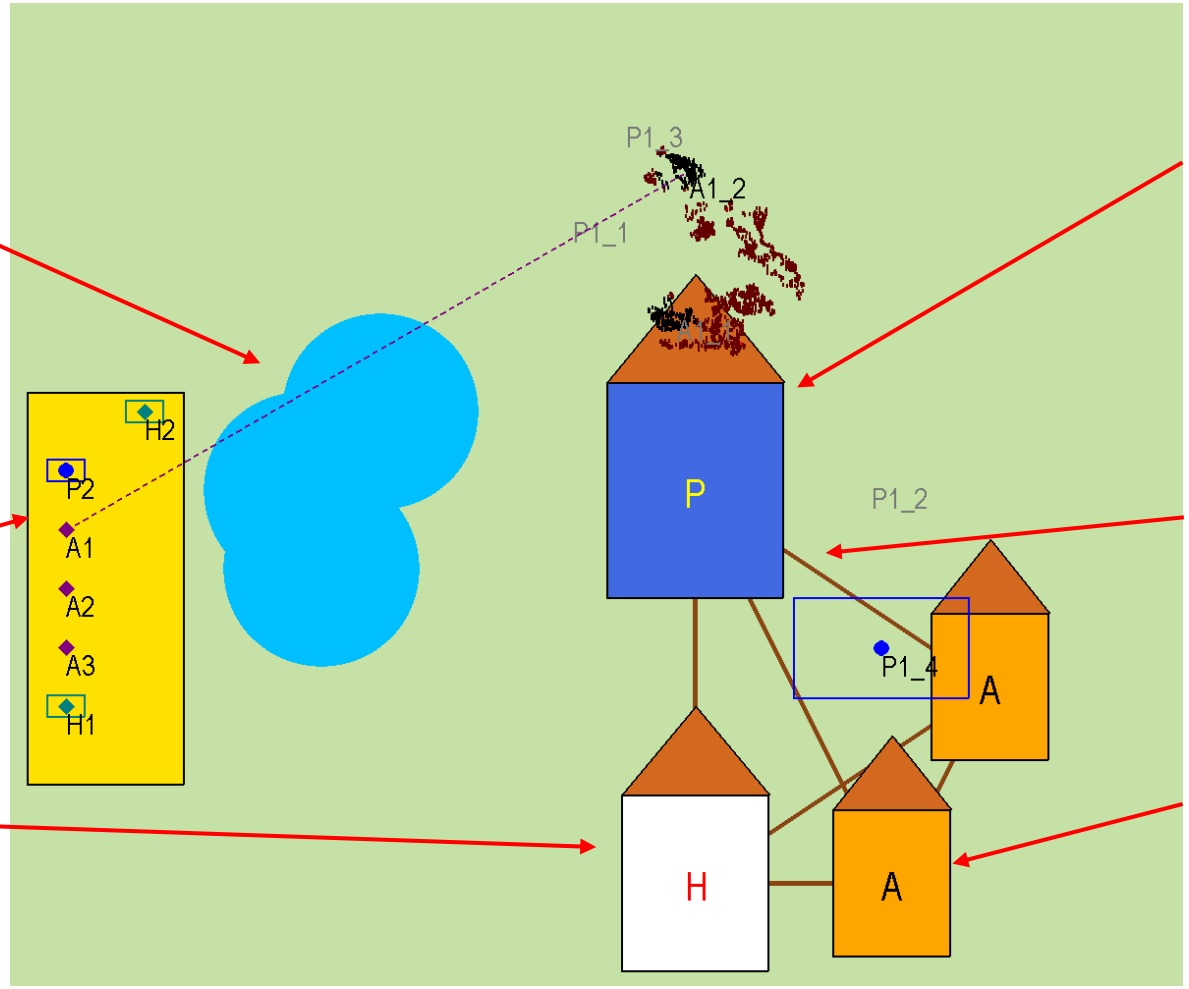
- **User Interface: PyGt 5**
  - One of the most common GUI design tool
  - Well-designed functions and controllers, modular design
  - Easy to integrate with other packages, including PyGame

Targets

Lake
- At most 5 in Each Scenario
- Composed by 3 100-pixel Circles
- Relative Position for Each Circle Center: (0, 0), (80, -80), (20, 80)
- Importance: **Low, Resistive to fire**

Agent Base (Vertical)
- Unique in Each Scenario, Close to the Edge of the Scenario
- Geometry: 160 × 400
- Capacity: 9 Agents
- Importance: **Very High**

Hospital
- At Most 5 in Each Scenario
- Geometry: 150 × 180
- Importance: **High**



Power Station
- At Most 5 in Each Scenario
- Geometry: 180 × 220
- Importance: **Very High**

Road
- Connect the Center of Each Target
- Line Width: 5
- Importance: **Low, Resistive to fire**

House
- At Most 5 in Each Scenario
- Geometry: 120 × 150
- Importance: **Medium**

Agents

- **Perception Agent**
  - Detect the fire spots that have been generated before and locate within the sensing scope
  - The sensing scope of the perception agent is adaptive based on the requirement
  - Wait at the terminal point of the trajectory if no further order is placed
- **Action Agent**
  - Put out the fire spots that have been sensed, while the pruning scope is fixed
  - Available only when the remaining water tank capacity is greater than 0
  - Return to the base if no further order is placed
  - **Note:** Action agents could put out all fire spots within a perception scope with at least one sensed fire spot
- **Hybrid Agent**
  - Agents that could both detect and put out the fire spots
  - Sensing scope is adaptive based on the requirement, while the pruning height equals to a certain value which is smaller than the action agent
  - Wait at the terminal point of the trajectory if no further order is placed

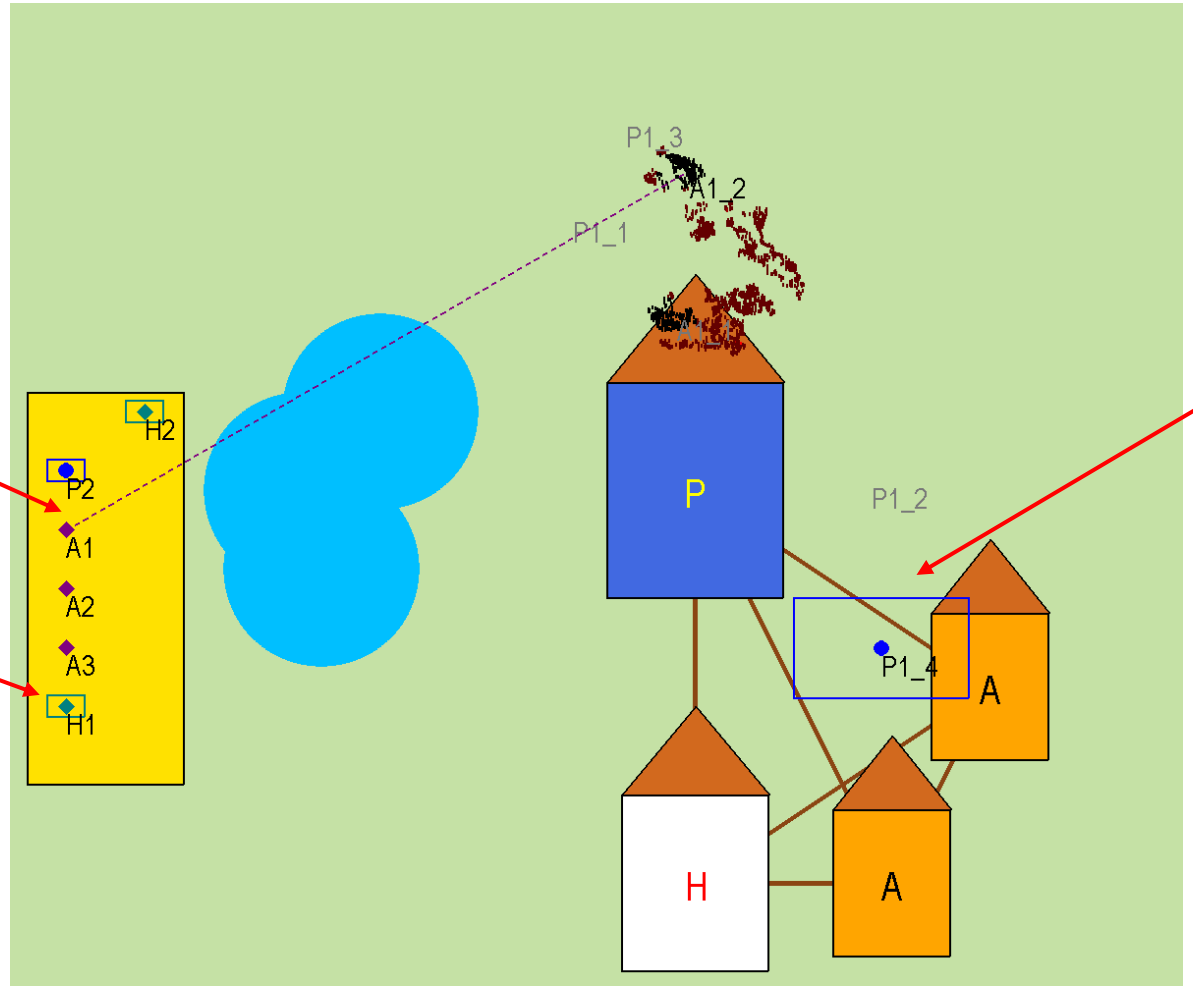
Agents

Action Agent (A)

- Purple diamond, No Visible Scope
- Default Pruning Height: 30 m (Unchangeable)
- Field of View: [π/6, π/4]

Hybrid Agent (H)

- Cyan diamond, Rectangle Scope
- Default Flight Height: [10, 100]
- Default Pruning Height: 20 m (Unchangeable)
- Field of View: [π/6, π/4]



Perception Agent (P)

- Blue diamond, Rectangle Scope
- Default Flight Height: [10, 100]
- Field of View: [π/6, π/4]

Switch and Control Policy

- **Switch**
  - Use digit key 1 – 9 to switch among each agent, in this order:
    - Perception 1, Perception 2, …, Action 1, Action 2, … Hybrid 1, Hybrid 2,…
  - To fit the keyboard layout, the maximum agent number is 9
- **Planar Motion**
  - Fly along the planar trajectory composed by several goals. Goals are set by mouse click.
  - Goals must be set when the given agent's battery and water tank are all not empty
  - When the current agent is changed, the previous one will still move along the trajectory
- **Vertical Motion**
  - Use up and down arrow to adjust the flight height of the agent
  - Flight height is persevered after switching

Movement Policy

- **Planar Motion**
  - Follow the trajectory composed by the goal series
  - Move with the step size that equals to the agent's velocity during the middle of the trajectory
  - When the distance between the agent's current position and the goal is less than one step size, its next position will directly overlap with the goal
- **Vertical Motion**
  - When pressing up or down key, the flight height changes by 5 meters
  - When the adjusted flight height exceeds the upper or lower bound, the flight height will not change
- **Battery Constraint**
  - Agents could not move when their battery capacity is 0. If so, the agent will directly stop the current task and return the agent base
  - The battery consumption is 0.1 during planar flight, and 0.05 during waiting

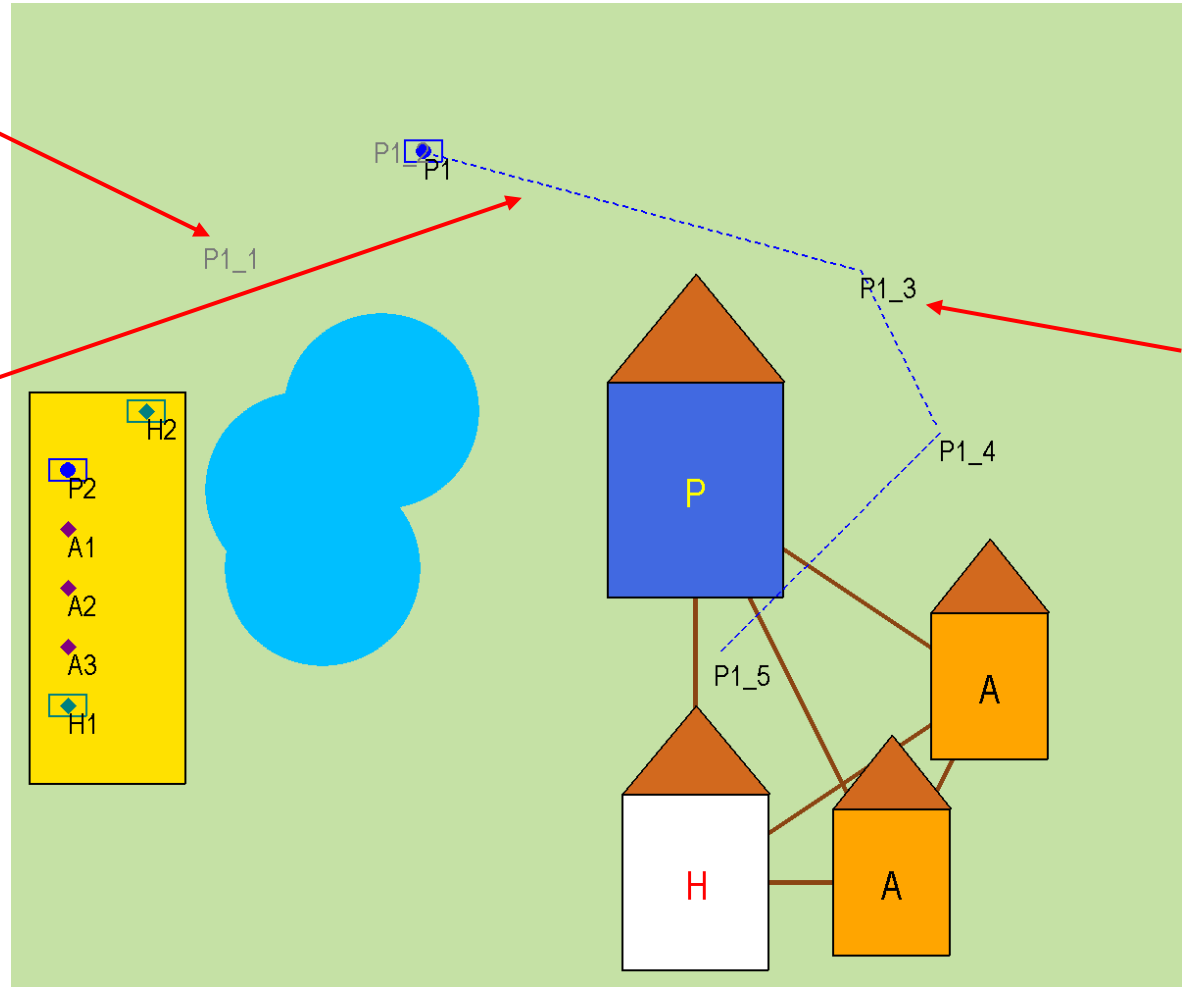
Goal and Trajectory

Passed Goal

- Marked with gray
- Only mark the goal position, ignore the passed trajectory

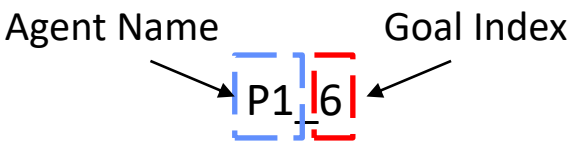
Trajectory

- Marked with dash line, using the same label color as the agent
- Present the trajectory that has not been passed yet



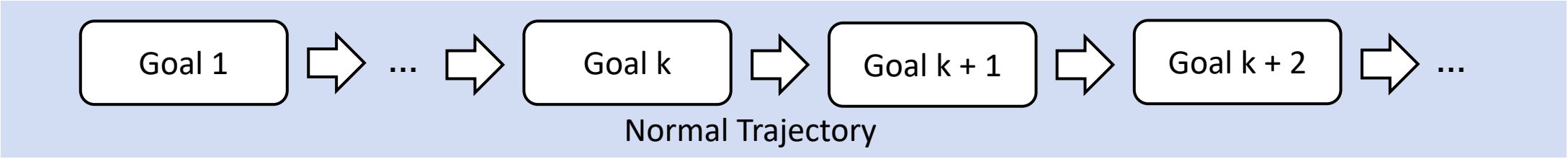
Pretending Goal

- Marked with black
- Positions are determined by mouse click
- If not being passed yet, connected with dash-line trajectory
- Goal Naming Policy:

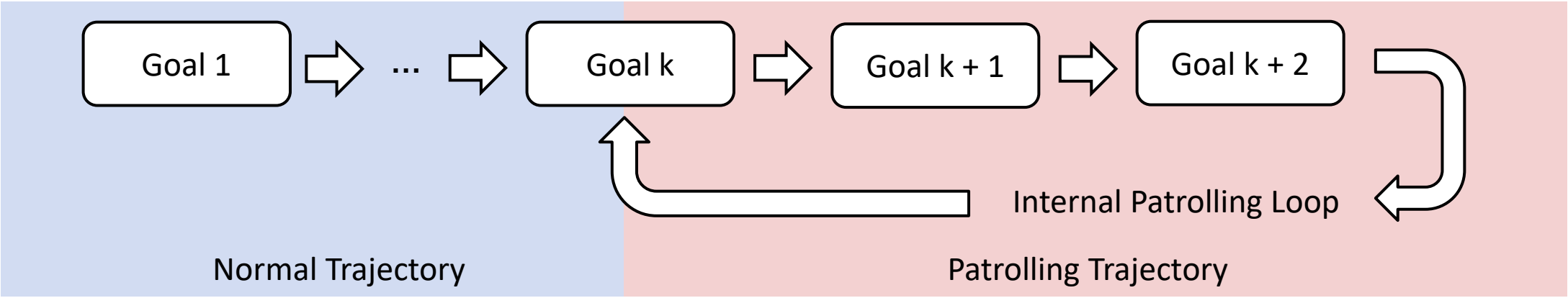


Normal and Patrolling Trajectories

- **Normal Trajectory**
  - Agent's trajectory follows the sequential order in the goal list

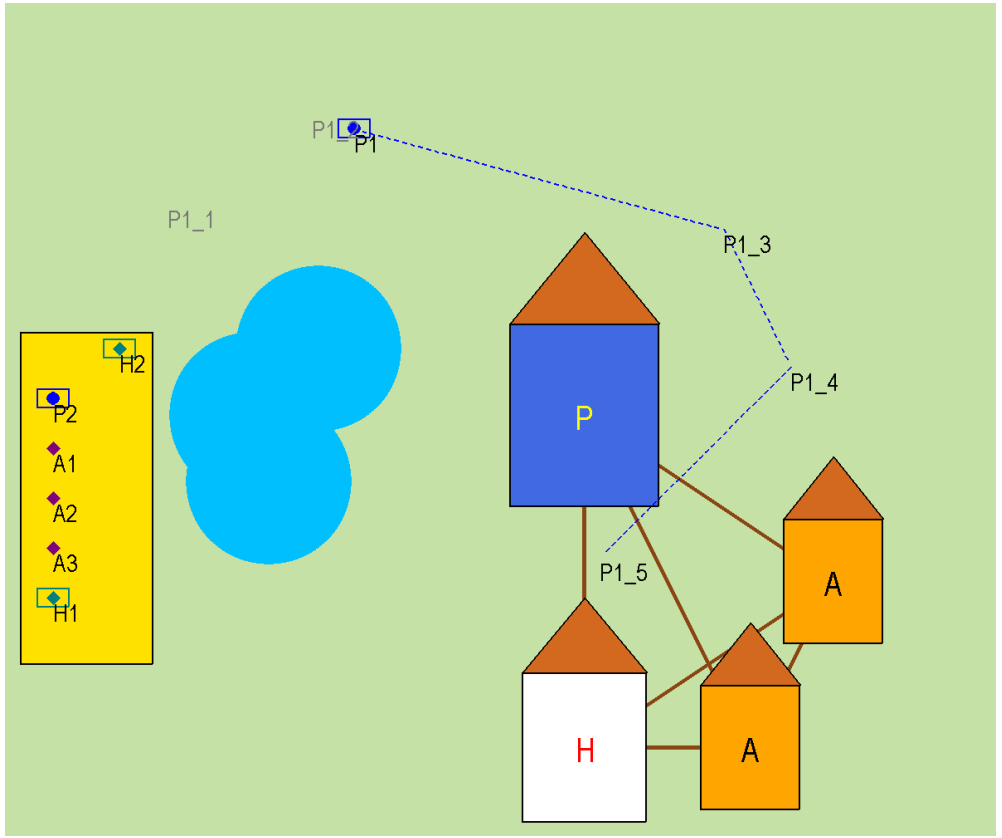


- **Patrolling Trajectory**
  - When the new goal locates close enough to one of existing goals in the goal list, goals in the list will form an internal loop trajectory called patrolling trajectory
  - When a new goal is added, the patrolling trajectory stops, and the previous patrolling goal list will be cleared. The agent follow the normal trajectory until a new patrolling loop is formed.

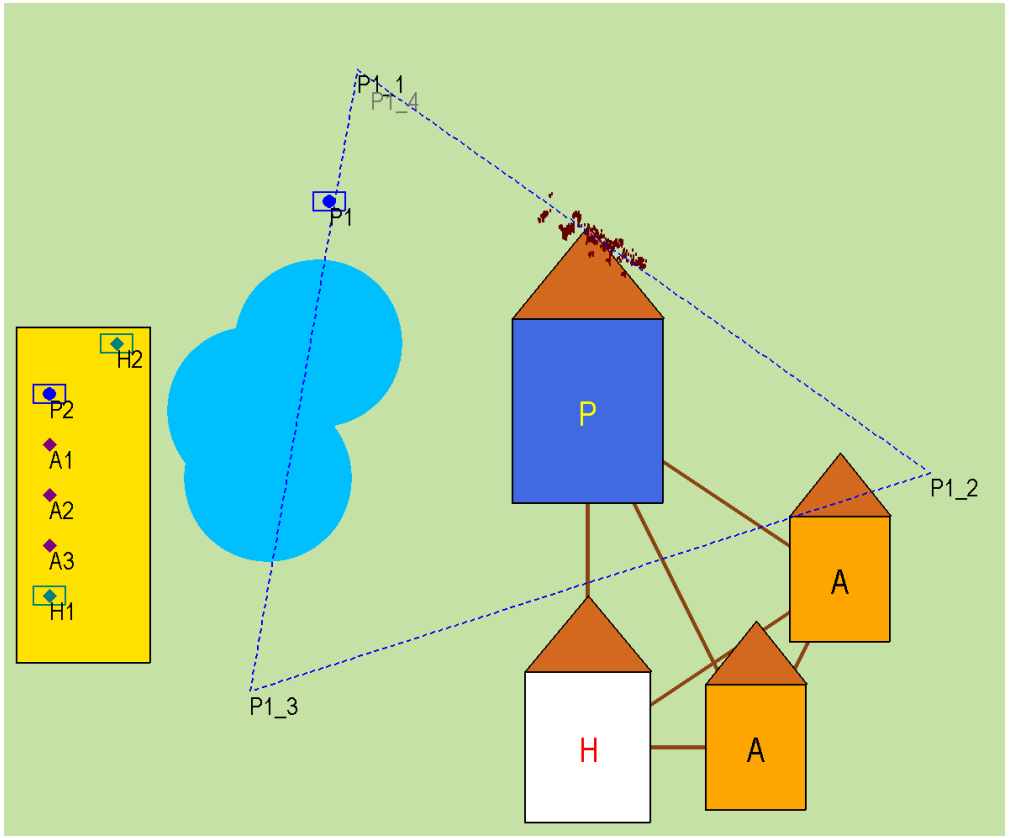


Normal and Patrolling Trajectory

- **Normal Trajectory**



- **Patrolling Trajectory**



FARSITE Wildfire Model

- **Wildfire Propagation Dynamics: FARSITE** [2]
  - Comprehensively consider the geographical, topographical and physical information about the environment, including the terrain, fuel and weather
  - Wildfire propagation dynamics using the FARSITE follows this form [3]:

$$q_t^i = q_{t-1}^i + \dot{q}_{t-1}^i \delta t = q_{t-1}^i + \frac{\partial}{\partial t} q_{t-1}^i \delta t$$

    - q_t^i is the position for fire spot i at time t. In 2-D environment, $q_t^i = [x_t^i,\ y_t^i]$
    - $\dot{q}_{t-1}^i$ is the fire propagation velocity

- **Fire Propagation Velocity**
  - For the fire propagation velocity, its simplified representation form is:

$$\begin{cases} \dot{x}_t^i = C sin\theta \\ \dot{y}_t^i = C cos\theta \end{cases}$$

    - *Θ* is the angle between the wind direction and the y-axis
    - *C* is the distance between initial fire position and the center of the ellipse. Its value relates to the fire growth coefficient, wind speed

FARSITE Wildfire Model

- **Fire Propagation Velocity**
  - The calculation process for *C* is [4]

$$C = \frac{R - \frac{R}{HB}}{2}$$

$$HB = \frac{LB + (LB^2 - 1)^{0.5}}{LB - (LB^2 - 1)^{0.5}}$$

$$LB = 0.936e^{0.2566U} + 0.461e^{-0.1548U} - 0.397$$

    - *R* is the fuel coefficient, which controls the fire spreading speed
    - *U* is the amplitude of wind speed

- **Fire Intensity**
  - Suppose the fire intensity at (x, y) is $I(x, y)$. Its value follows the multi-variable normal distribution

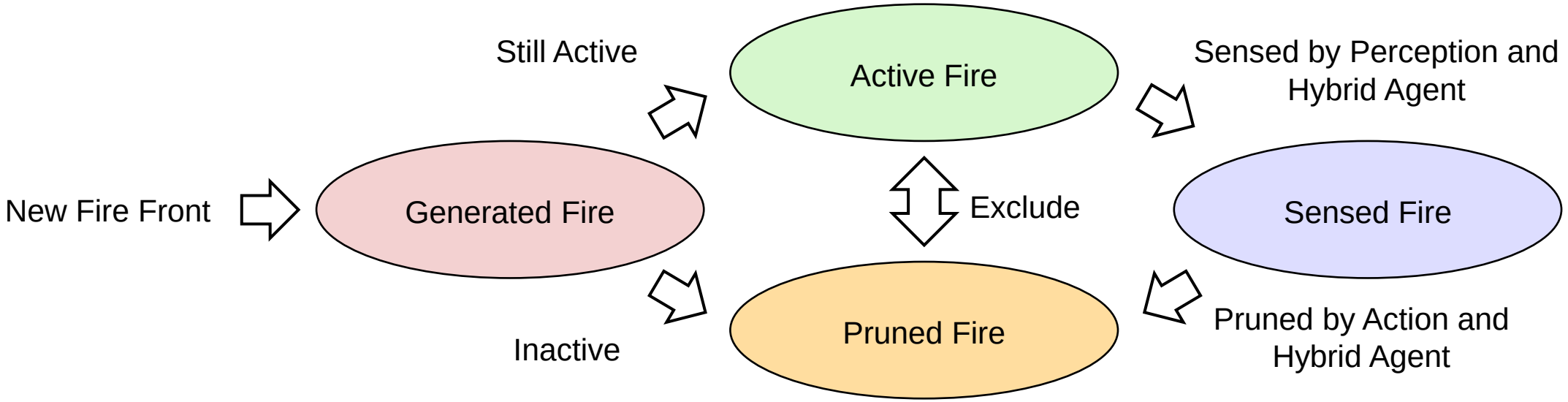
$$I(x,y) = \sum_{i=1}^{k} \frac{1}{2\pi\sigma_{x_i}\sigma_{y_i}} e^{-\frac{1}{2}\left[\frac{(x-x_f)^2}{\sigma_{x_i}^2} + \frac{(y-y_f)^2}{\sigma_{y_i}^2}\right]}$$

    - (x_f, y_f) is the center of the initial fire spot, $(\sigma_{x_i}, \sigma_{y_i})$ is the derivation in *x* and *y* direction

Reconnaissance Wildfire

- **Reconnaissance Wildfire**
  - In the reconnaissance wildfire mode, wildfire spots must be sensed before they could be pruned. In this case, wildfire spots that have been generated could be divided into the following genres:
    - **Sensed Fire:** Wildfire spots that have been sensed by the perception and hybrid agents
    - **Pruned Fire:** Wildfire spots that have been pruned by the action and hybrid agents

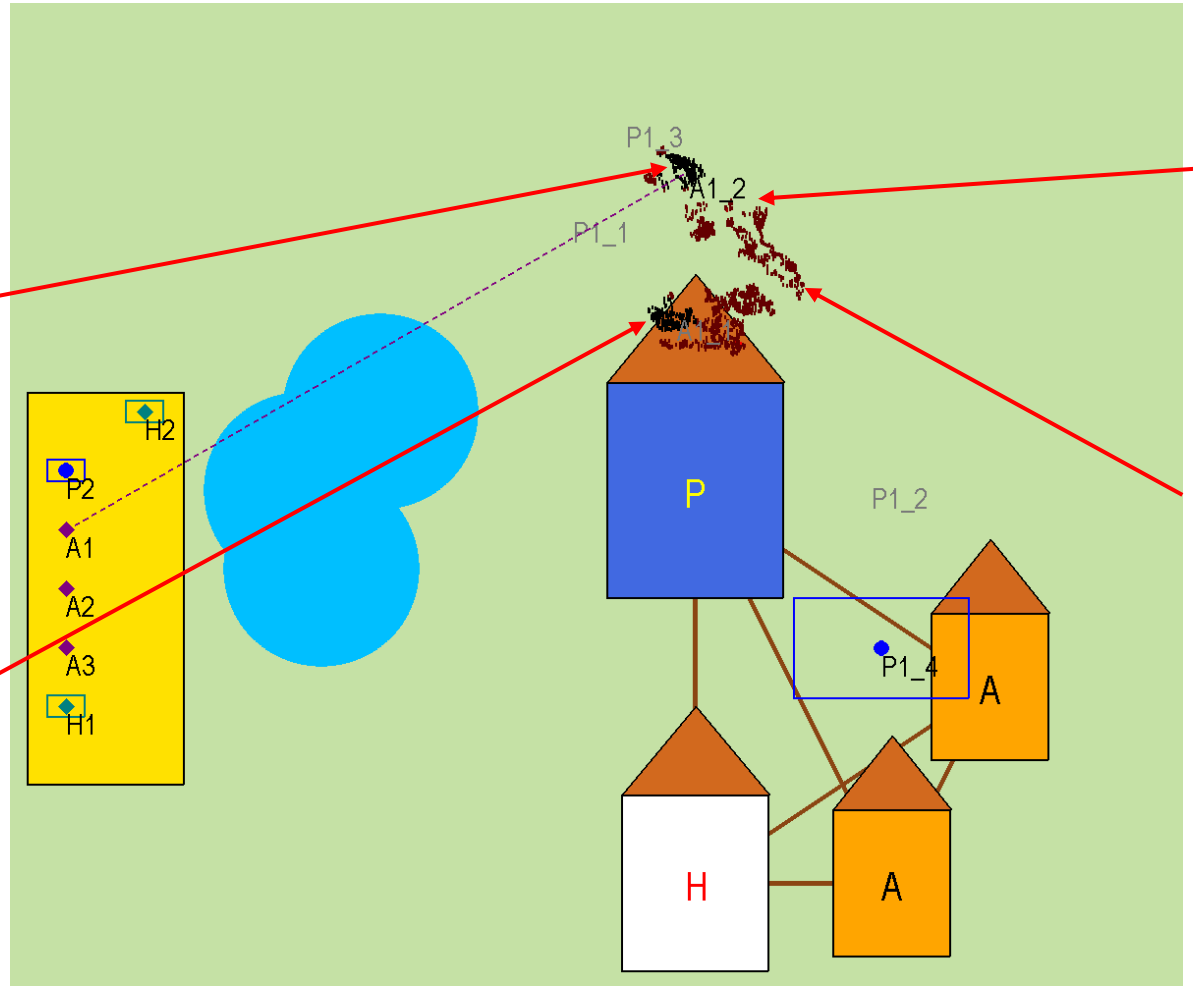
      Note: All the pruned fire spots are considered as sensed fire spots
    - **Active Fire:** Wildfire spots have not been pruned, including fire spots have or have not been sensed



Reconnaissance Wildfire Display

Pruned Fire
- 1-radius black points
- Composed by all fire spots that once locate with the sensing agent's scope
- New fire fronts could not be propagated into the pruned region
- **Note:** Fire spots must be sensed before they could be put out

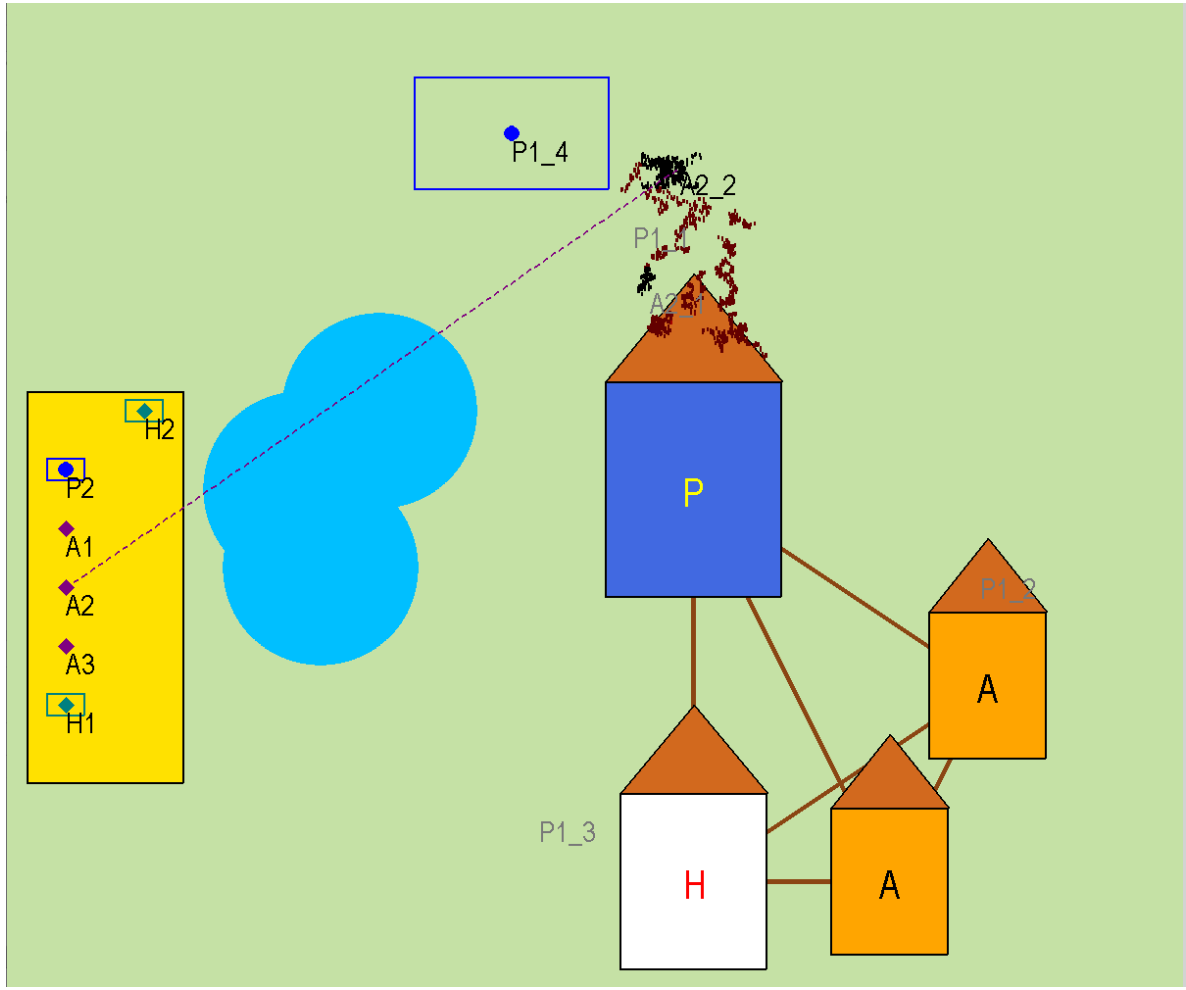
New Fire Fronts
- 3-radius black points
- New location determined by the FARSITE algorithm
- Updated in each iteration, could overlap with the previous fire spots



Active Fire Fronts (Not shown)
- 1-radius red points
- Composed by all new fire fronts generated before that have not been pruned

Sensed Fire
- 1-radius red or brown points
- RGB value is set based on the intensity, the range 100 - 255
- Composed by all the new fire spots that have been detected before

Information Display



Battery Capacity Info:

Agent	Remaining Energy	Estimated Energy Till Next Goal	Current Height
Perception 1:	249	249	100
Perception 2:	500	N/A	20
Action 1:	500	N/A	N/A
Action 2:	500	425	N/A
Action 3:	500	N/A	N/A
Hybrid 1:	500	N/A	20
Hybrid 2:	500	N/A	20

Water Tank Info:

Agent	Remaining Pruning Times
Action 1:	10
Action 2:	10
Action 3:	10
Hybrid 1:	10
Hybrid 2:	10

Score:	-23577.75
Overall Firefighting Score:	13.24
Perception Score:	43.29
Action Score:	30.59
Safe Facilities:	4

Battery Capacity Info

- Remaining Energy: Current remaining energy in the agent's battery
- Estimated Energy Till Next Goal: The estimated energy left when the agent arrives the next goal (Could be Negative)
- Current Flight Height (Sensing and Hybrid Agent)

Water Tank Info

- Computed by the remaining pruning times (Firefighter and Hybrid Agent)

Online Score Display

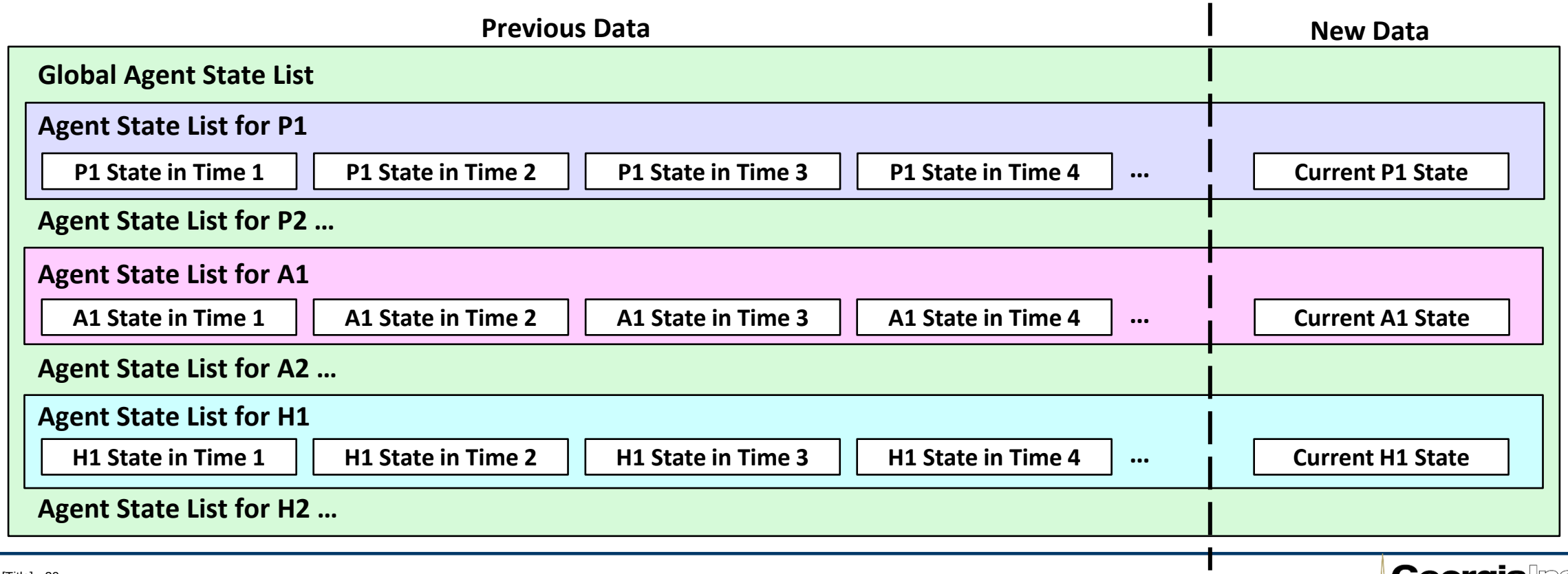
- **Negative:** Total Negative Score
- **Positive:** Overall Firefighting Score, Perception and Action Score, The Number of Safe Facilities

Data Storage

- **Motivation**
  - Transfer information between different part of the program
  - Track the scenario state at each moment
  - Recreate the scenario for animation reconstruction and LfD policy generation
- **Stored Data Type**
  - **Agent Info**
    - Current Agent State (Perception, Action and Hybrid Agent)
  - **User Data**
    - Keyboard Action and Goal Info
  - **Target Info**
    - Target Loci (House, Hospital, Power Station)
    - Agent Base Loci, Lake Loci
  - **Fire Info**
    - Fire Coordinates Info (Active, Sensed, Pruned)

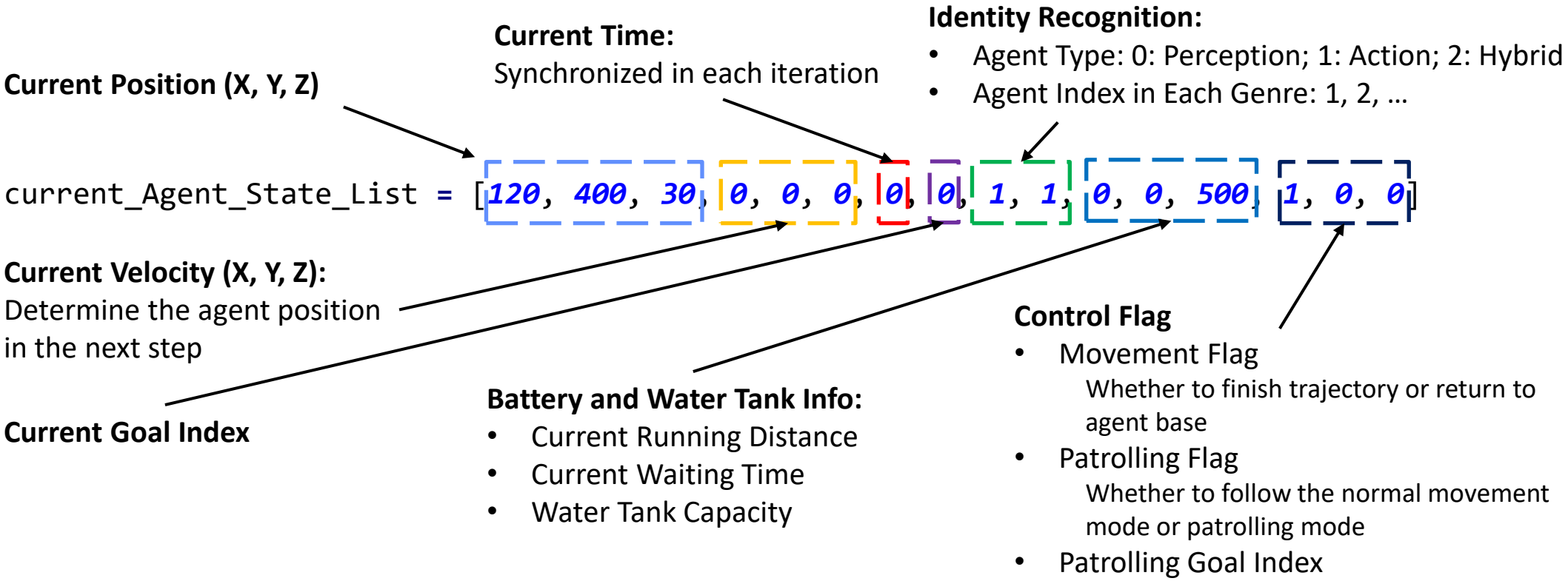
Agent State List

- **Global Agent State List Structure**
  - Record each agent's state at each moment during the simulation
  - **Hierarchy Structure**: Global Agent State (A List for All Records) -> Agent State List for P1 (All Records for Agent P1) -> P1 State in Time 1 (Record for Agent P1 at Time 1)



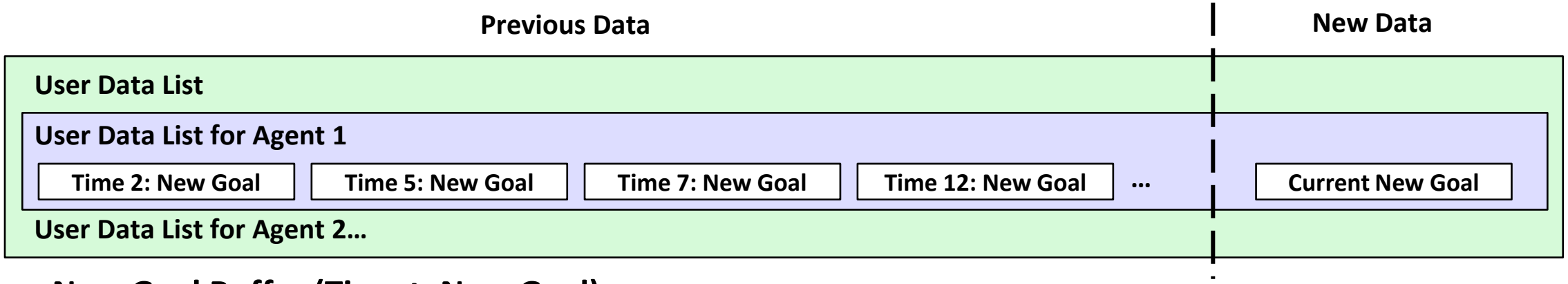
Agent State List

- **Current Agent State List (Record for Agent X at Time t)**
  - 16-element list. Describe the state for agent X at the given moment t
- **Sample Structure (Record for P1 at Time 0)**

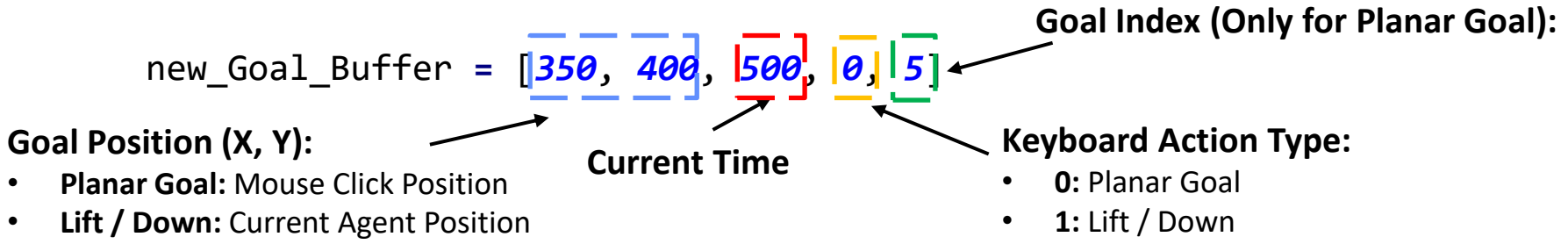


User Data List

- **User Data List**
  - Monitor and record the keyboard action
  - Store the goal list for trajectory generation, including the normal and patrolling trajectory

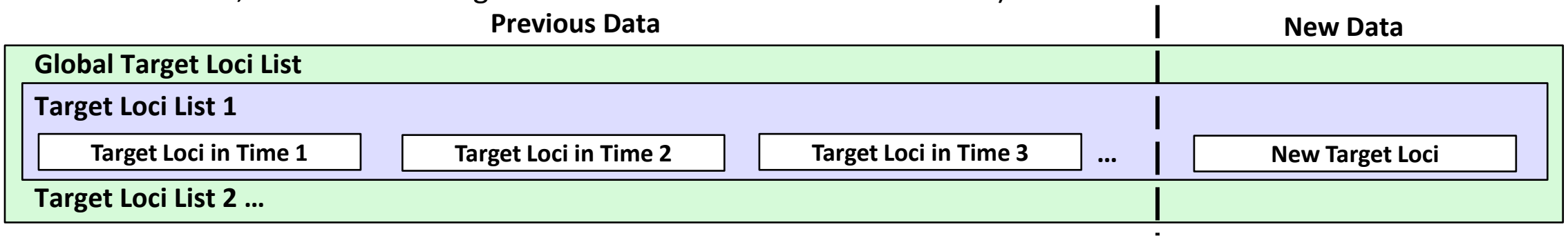


- **New Goal Buffer (Time t: New Goal)**
  - New goal information for agent X acquired at time t

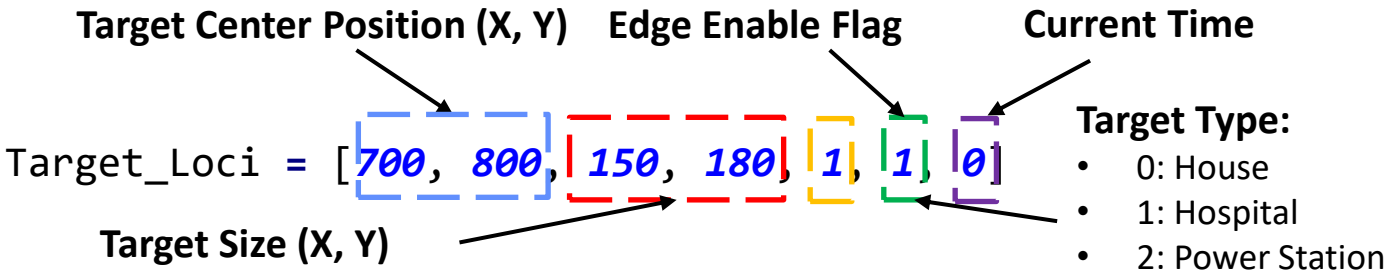


Target Loci List

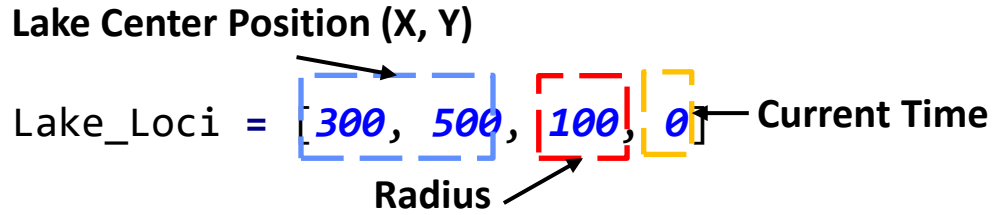
- **Target Loci and Lake List (Using the Same Template)**
  - Record the status of all targets or lakes at each moment
- **Sample Target Loci List**
  - For lake list, substitute all target loci list units with lake loci list units)



- **Target Loci List Unit**
  - Loci information for target X at time t

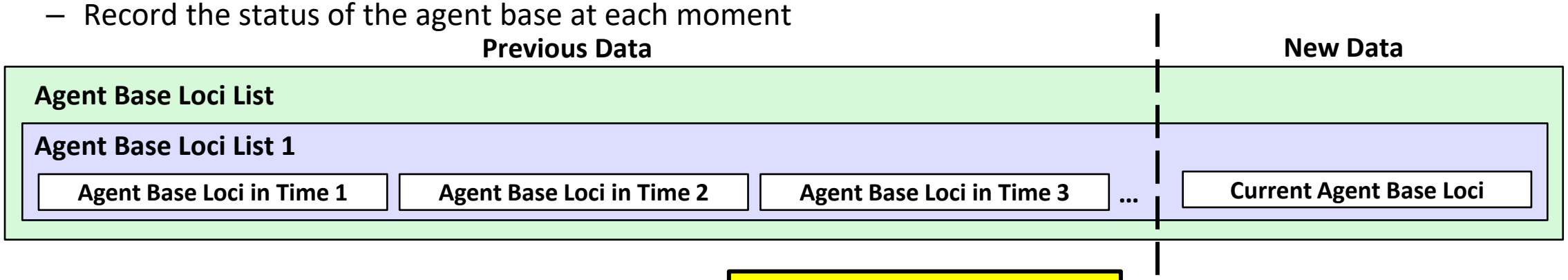


- **Lake Loci List Unit**
  - Loci information for lake X at time t

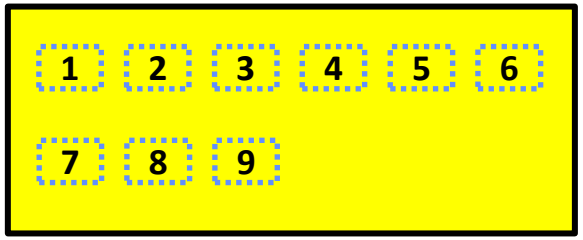


Target Loci List

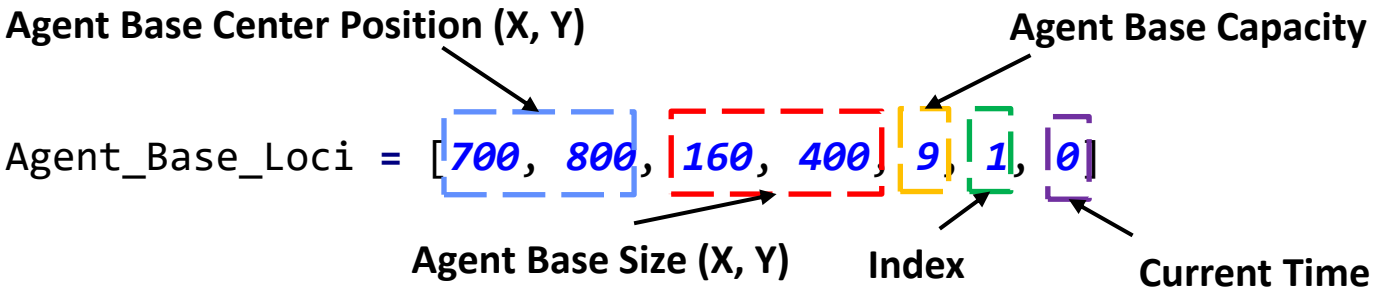
- **Agent Base Loci List**
  - Record the status of the agent base at each moment



- **Agent Base Arrangement**
  - Constant list, record the agent's relative position in the agent base

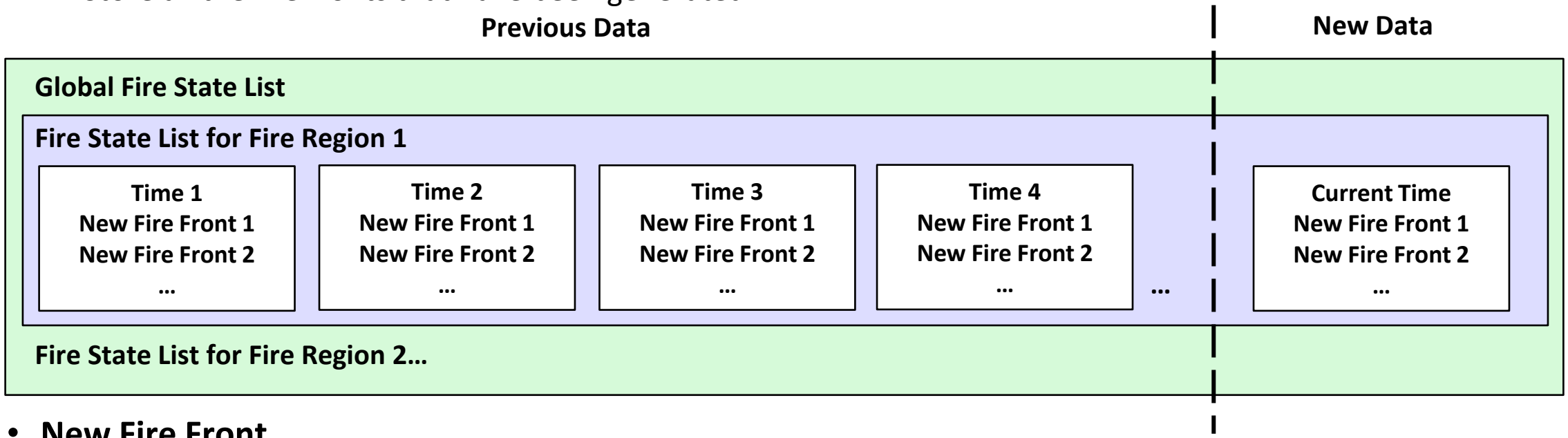


- **Agent Base Loci List Unit**
  - Record the loci and capacity information for the agent base at time t

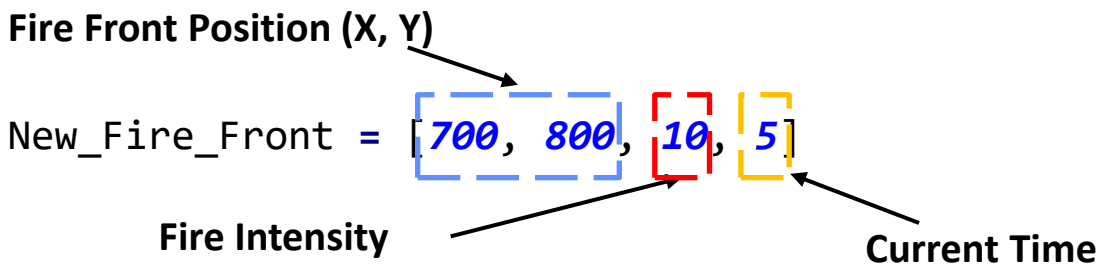


Wildfire List

- **Fire State List**
  - Store all the fire fronts that have been generated

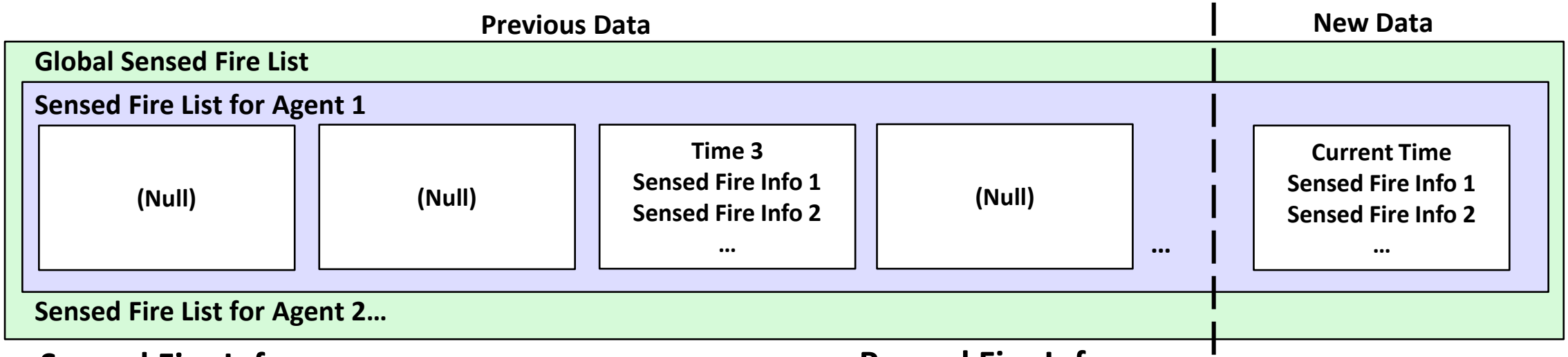


- **New Fire Front**
  - Record the position, intensity and generated time for new fire fronts generated at time t



Wildfire List

- **Sensed and Pruned Fire List (Using the Same Template)**
  - Store all the fire fronts that have been sensed or pruned in the given interval. If no fire fronts are sensed or pruned, return null list



- **Sensed Fire Info**
  - Record the position, intensity and velocity for sensed fire spots

```
Sensed_Fire_Front = [700, 800, 10, 5]
```

Fire Front Position (X, Y)

Fire Intensity

Fire Velocity, To compute the CoM of the sensed region

- **Pruned Fire Info**
  - Record the position, intensity for pruned fire spots

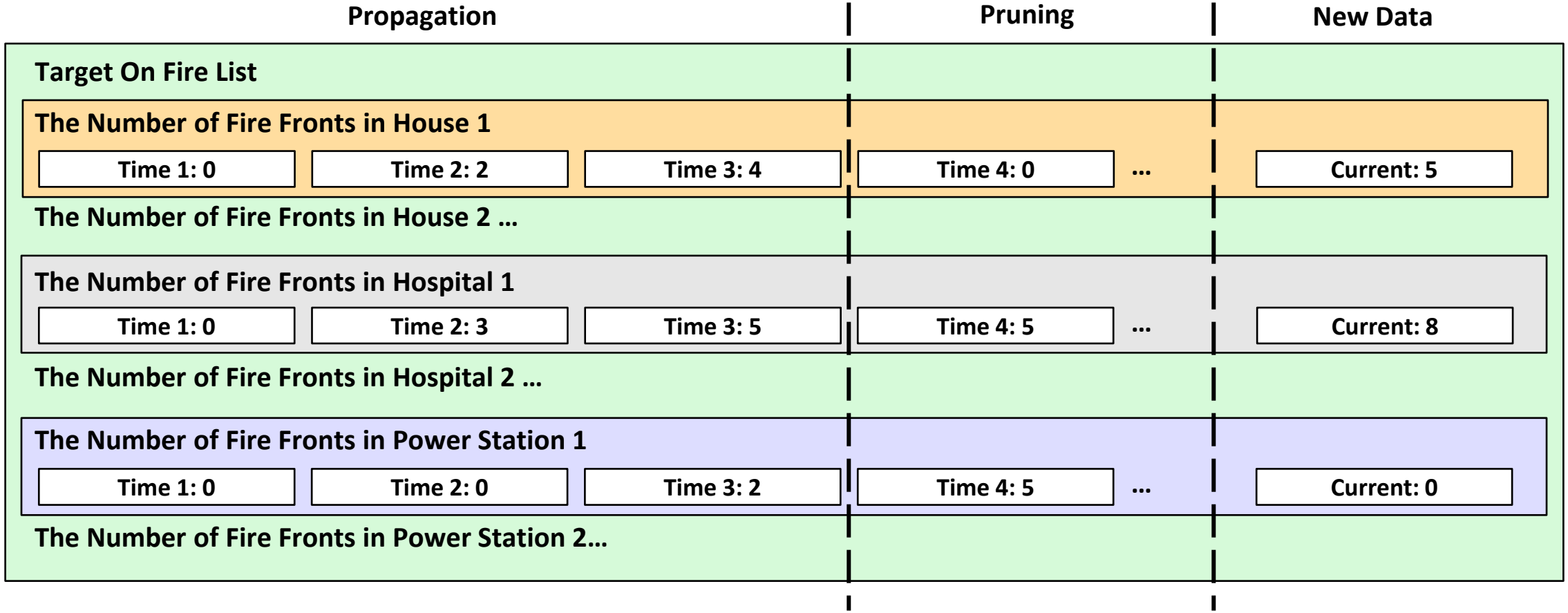
Fire Front Position (X, Y)

Fire Intensity

```
Pruned_Fire_Front = [700, 800, 10]
```

Target On Fire List

- **Target On Fire List**
  - Store the number of fire fronts that locate inside each target at each moment



Score Policy

- **Requirement**
  - Encourage users to put out as many fire spots as possible
  - Urge users to protect targets, especially the most important ones
  - Comprehensively consider the perception, pruning and target protection performance
  - Normalized, clear and straightforward
- **Feature**
  - **Weighted Score**
    - Fire spots are propagated into the targets will cause serious penalty, while the penalty caused by fire spots on the grassland is relatively slight
    - All kinds of scores are normalized in the same scope before summing up
  - **Coordinate-wise Policy**
    - Two fire spots propagated into the same location in different time interval will be taken as one spot before they have been put out

Score Policy

- **Negative Reward**
  - **Total Negative Reward**

$$0.1 \times \text{Number of active firespots} + \text{Penalty Coef} \times \text{Firespots Number in Targets}$$

    - **Penalty Coefficients: 0.1** Per Fire Spot, **1** Per House, **2** Per Hospital, **5** Per Power Station, **5** Per Agent Base
    - **Active Fire Spots:** All fire spots in each single coordinates that have not been pruned (Not Sensed + Sensed, Excludes the pruned fire spots)
    - **Fire Spots in Targets:** The active fire spot number in each targets in the given interval
  - **Expected Negative Reward**

$$0.1 \times (\text{Number of active firespots} + \text{Number of pruned firespots}) + \text{Penalty Coef} \times \text{Firespots Number in Targets}$$

    - **Penalty Coefficients:** The same
    - **Fire Spots:** All fire spots in each single coordinates that have ever been generated
  - **Negative Reward Ratio**

$$\frac{\text{Total Negative Reward}}{\text{Expected Negative Reward}} \times 100\%$$

Score Policy

- **Positive Reward**
  - **Perception Score**
    - The ratio of sensed fire spots in all the fire spots generated

$$\frac{\text{Number of discoverd firespots}}{\text{Total number of firespots}} \times 100\%$$

  - **Action Score**
    - The ratio of pruned fire spots in all the sensed fire spots

$$\frac{\text{Firespots that have been put out}}{\text{Number of discovered firespots}} \times 100\%$$

  - **Safe Facility Score:**
    - The ratio of safe facilities (Not on fire during the whole simulation) with the number all the facilities

$$\frac{\text{Number of facilities that have been saved}}{\text{Total number of facilities}} \times 100\%$$

Score Policy

- **Final Score**

$$\text{Perception Score} + \text{Action Score} + \text{Safe Facility Score} - 3\times \text{ Negative Reward Ratio}$$

- **General Evaluation**

Grade	Final Score
Failed	< 50
Fair	50 - 60
Almost There!	60 - 80
Well Done	80 - 90
Excellent	>90

Animation Reconstruction

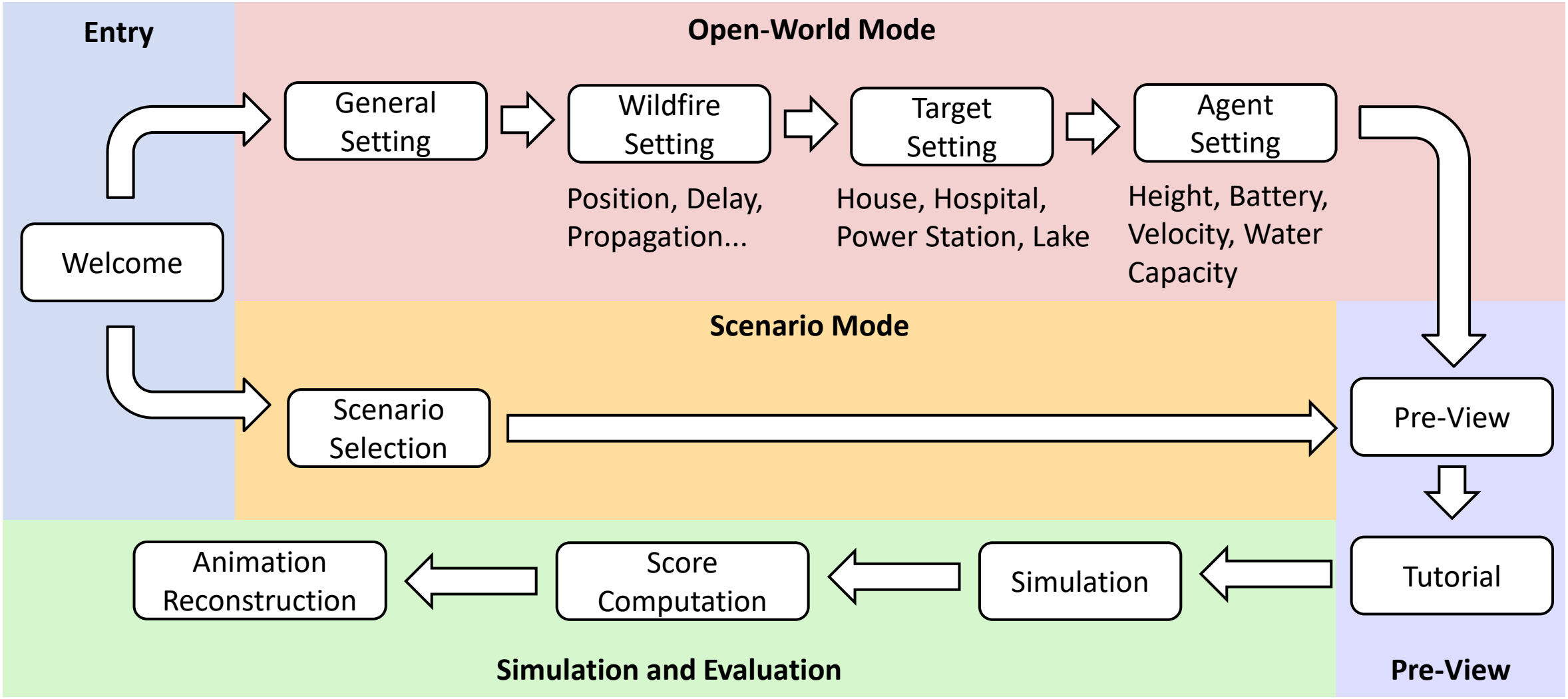
- **Motivation**
  - LfD uses the screenshot the generate the control policy, but online image I/O in Python is time-consuming.
    - **Maximum Simulation Refreshing Frequency:** Iteration with Online Image I/O: **10 Hz**; Pure Iteration: **100 Hz**
- **Solution**
  - Use the .pkl file saved to re-construct the scene during the simulation
- **Procedure**

Load Data	Reconstruct Scenario	Take Screenshots	Create Animation
Load the data from the .pkl file, which is the status of each object at each moment	Reconstruct simulation with for loop: • **Static:** Houses, Hospitals, Power Stations, Lakes… • **Dynamic:** Agents, Wildfire Propagation • **Ignoring:** Agent Trajectory and Goal Marks	Use the screenshot function to save the status of the simulation at each moment	Create the animation with the images saved when the simulation finished

Graphical User Interface

- **Motivation**
  - Complicated and confusing parameters in the environment setting
    - **Concise:** Help the user to design the environment step by step
    - **Interactive:** Enable the modification, Visualize the completed environment
  - Requirement for the various scenario with different settings
    - **Comprehensive**: Cover all the significant parameters in the environment
    - **Convenient:** Offer several pre-determined scenarios

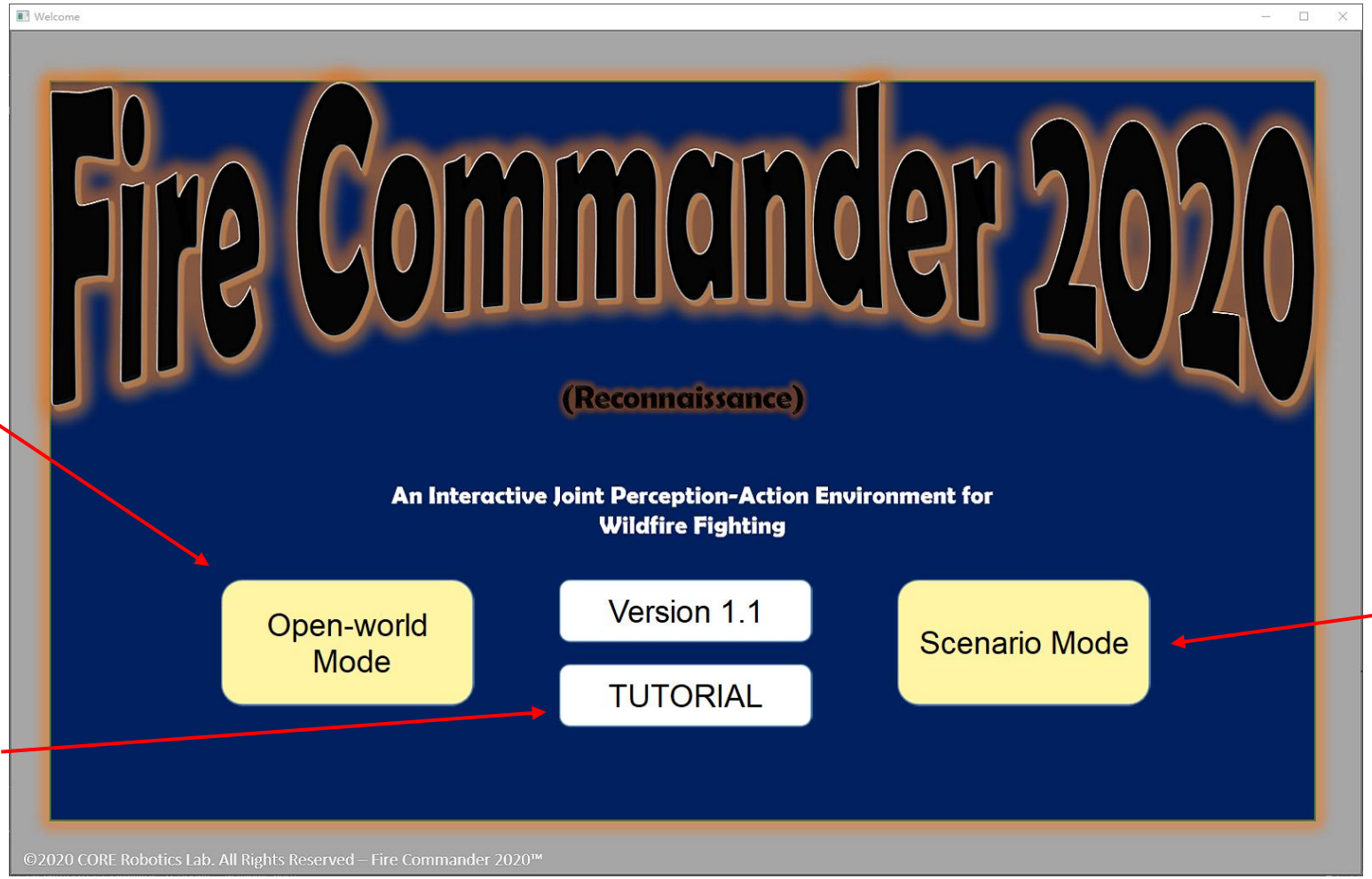
GUI Frame


Entry
Open-World Mode
Welcome
General Setting
Wildfire Setting
Position, Delay, Propagation...
Target Setting
House, Hospital, Power Station, Lake
Agent Setting
Height, Battery, Velocity, Water Capacity
Scenario Mode
Scenario Selection
Pre-View
Tutorial
Simulation
Score Computation
Animation Reconstruction
Simulation and Evaluation
Pre-View

Welcome Page

Open-World Mode
Design the environment step by step

Tutorial
Learn about the regulation of the environment
Note: Could not enter the simulation from the tutorial page that enters from the welcome page

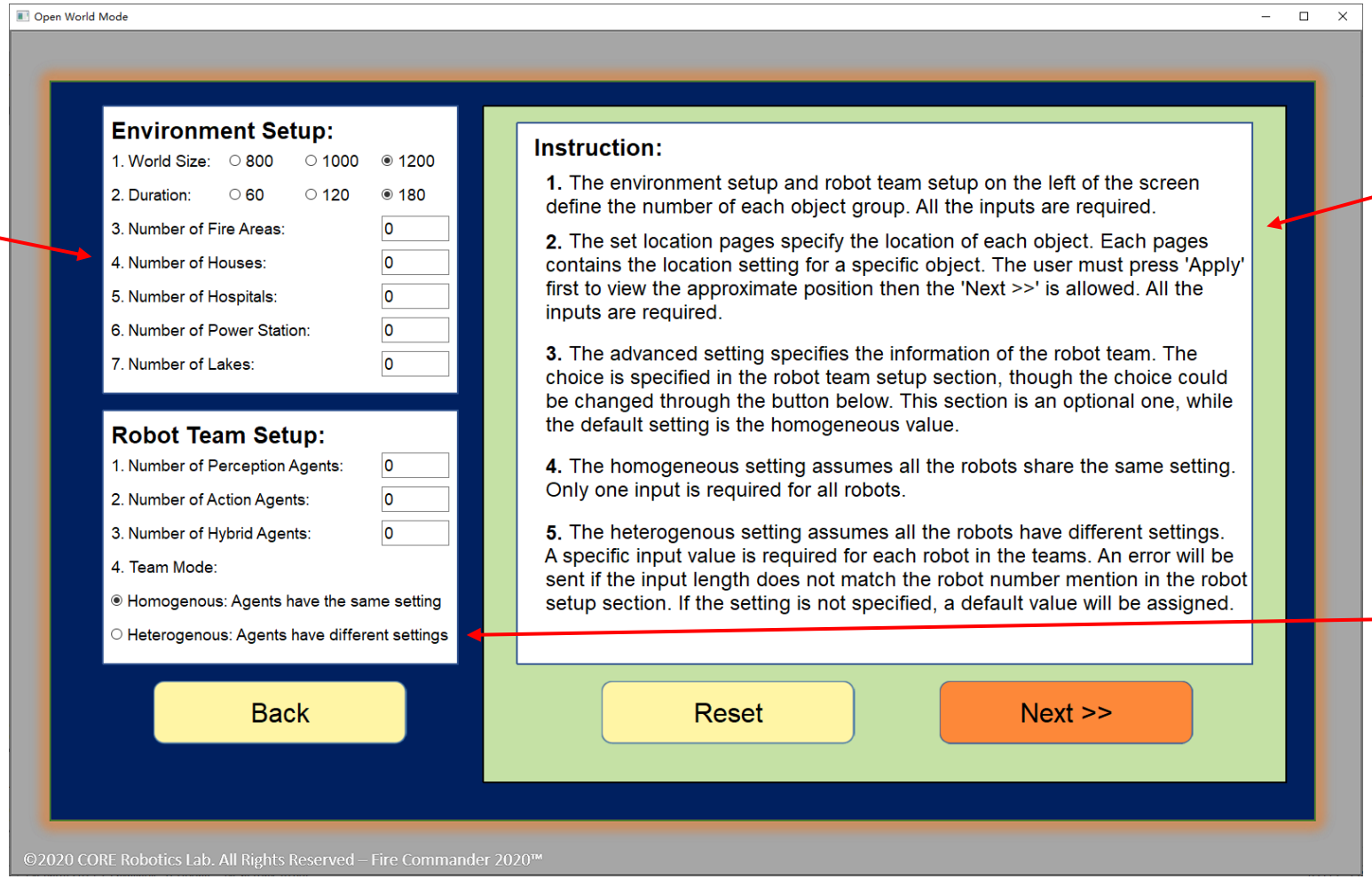


Scenario Mode
Use the pre-determined scenarios for simulation

General Setting Page

Environment Setting

- **World Size**
  800 / 1000 / 1200
- **Duration**
  60 / 120 / 180
  The user can either exit the program by pressing the exit button or wait until the due time
- **Target number**
  Categorical, Maximum target number is 5



Brief Instruction on the GUI

Robot Setting

- **Agent number**
  Categorical, Maximum agent number is 9 in total
- **Team mode**
  Whether the agents have the same setting (Flight Height, Battery, Velocity, Water Tank Capacity)

Target Setting Page

Info Display

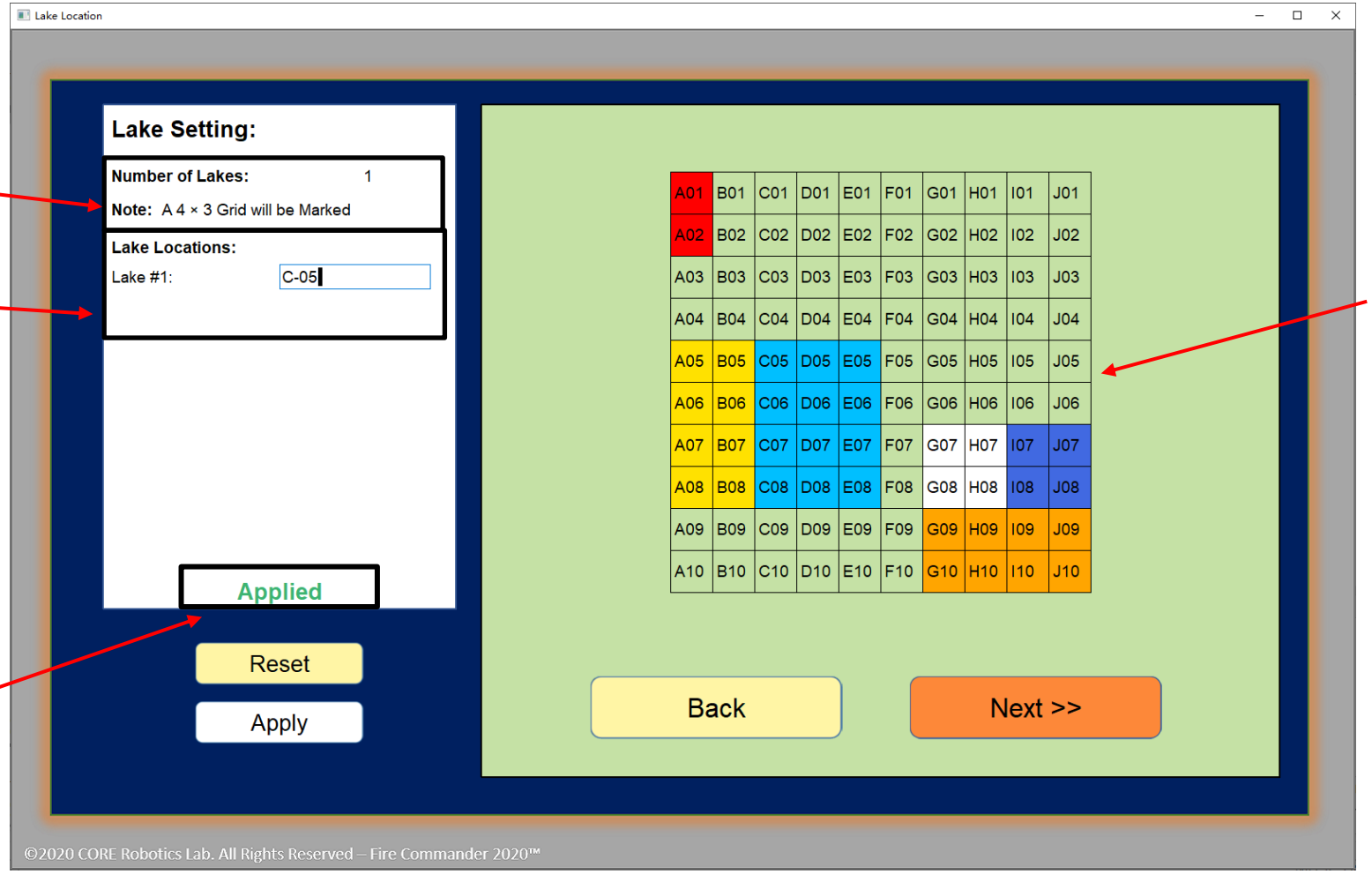
- Target Number
- Symbol Size on the Grid Map

Target Location

- In coordinates, Set One-by-one

Applied Flag

- Initial Status or After Any Target Location Changed:
  Not Applied
- Press Apply Button and All the Inputs are Valid:
  Applied



Grid Map

Use the grid and following symbols to represent the scenario generated

- Grassland
- Fire Region 1 × 1 Grid
- Agent Base 2 × 4 (Horizontal) / 4 × 2 (Vertical) Grid
- House 2 × 2 Grid
- Hospital 2 × 2 Grid
- Power Station 2 × 2 Grid
- Lake 4 × 3 Grid

Wildfire Setting Page

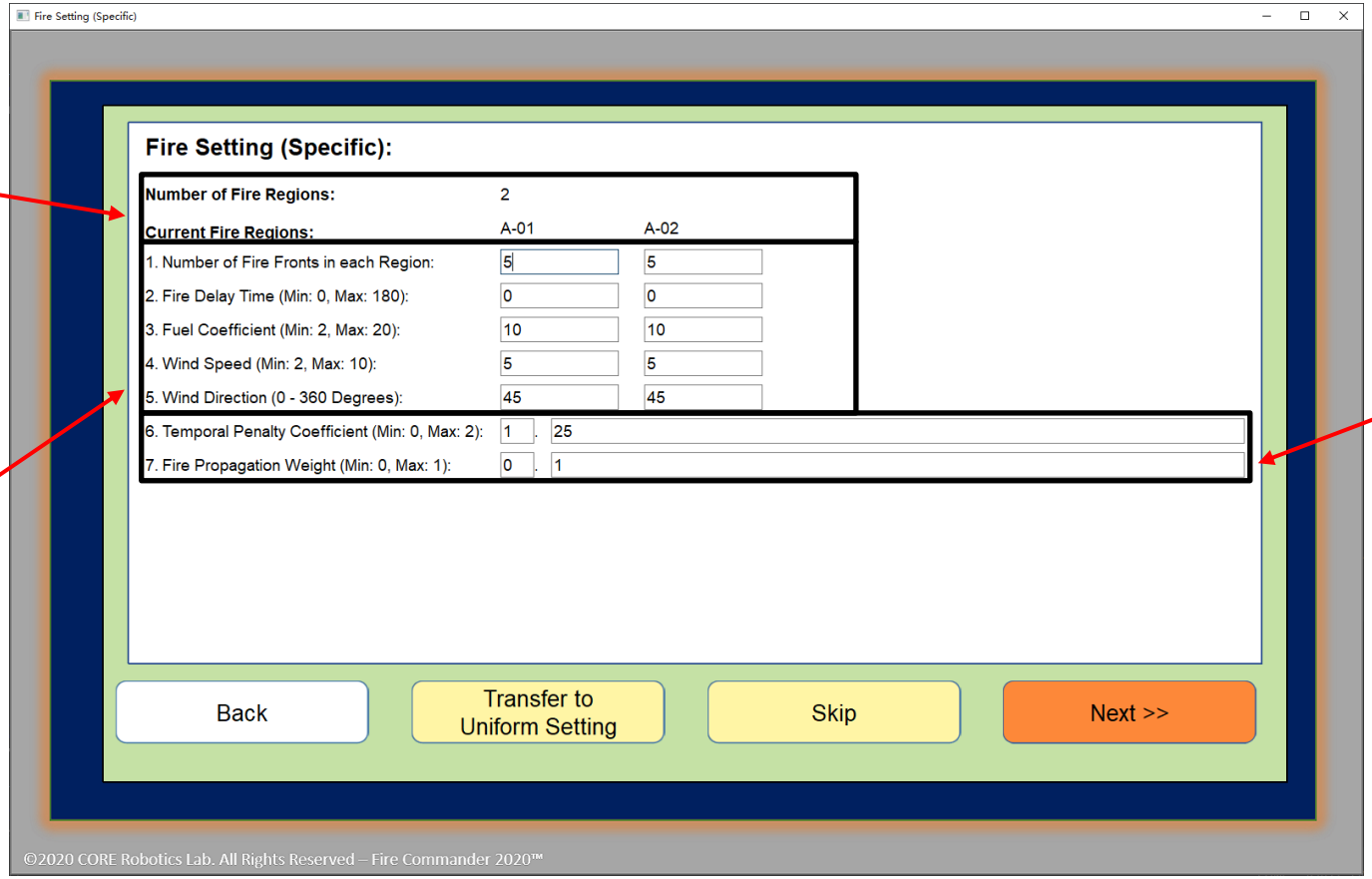
- **Location Setting:** The same as the target setting
- **Parameter Setting:** Uniform (Homogenous Setting) / Specific (Region-wise Setting)

Info Display

- Fire Region Number
- Coordinates on the Grid Map

Fire Setting

- Fire Front Number in Each Region
- Delay Time
- Propagation: Fuel Coefficient, Wind Speed and Direction



Score Parameter

- Parameters used to compute the negative reward introduced by the wildfire propagation

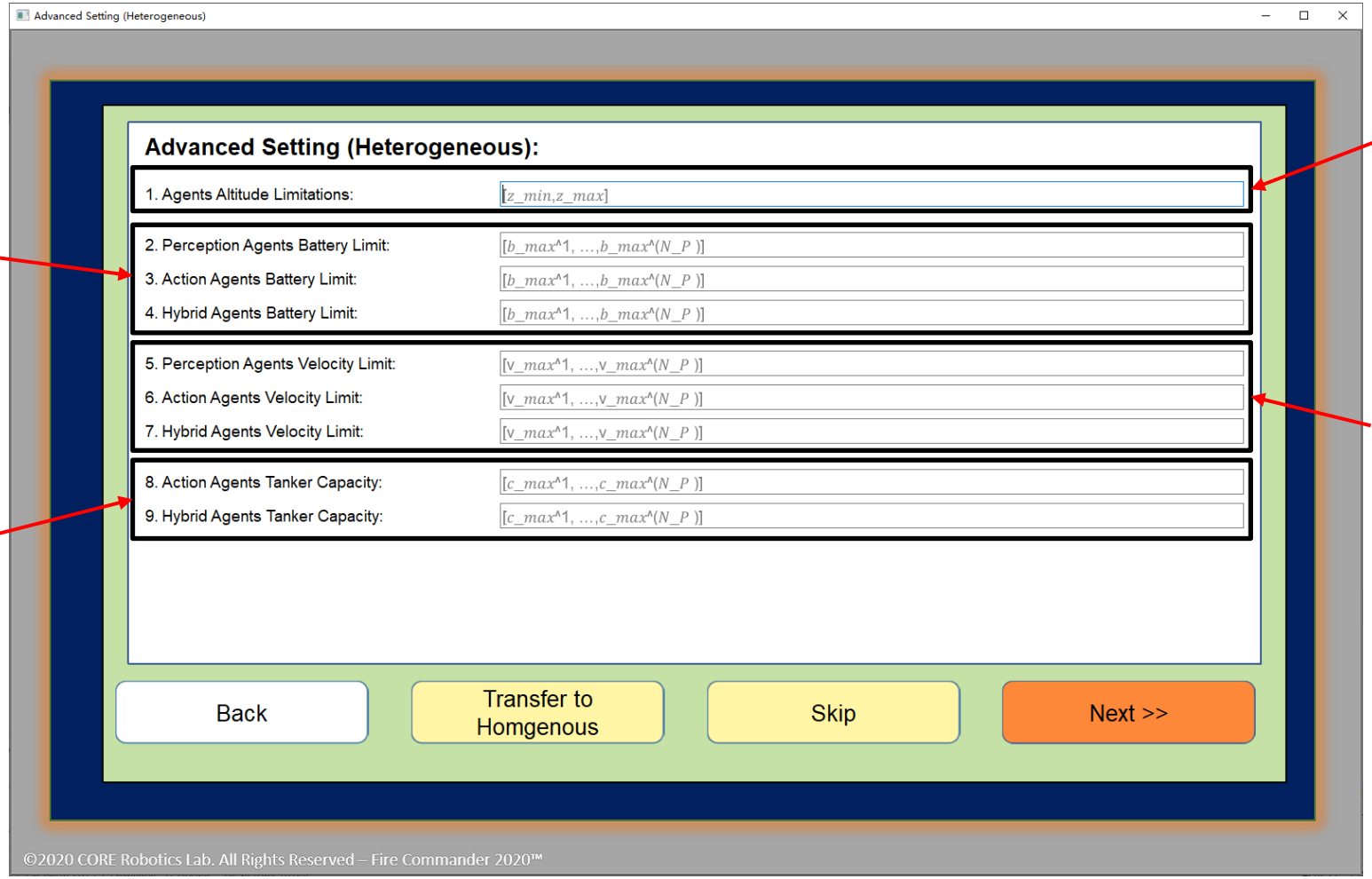
Agent Setting Page

Battery Limit

- Determine the battery capacity
- Consumption During Waiting: 0.05 / Iteration
- Consumption During Flight: 0.1 / Iteration

Water Capacity

- Only for the action and hybrid agent
- Determine the time number that the agent could put out the fire in the given region



Agent Flight Height

- From Minimum Height to Maximum Height

Velocity

- Determine the maximum velocity for each agent
- Generally, the step size of the agent equals to the velocity. However, when the distance between the goal and the agent is small enough, the agent will directly move to the goal

Agent Setting Page

- **Standard Input List Format**
  - E.g. There are 1 perception and 2 hybrid agents in the environment
  - **Homogenous**
    - Height Limitations: [10, 100]
    - Perception Battery Limit: [500]
    - Hybrid Battery Limit: [500]
  - **Heterogenous**
    - Height Limitations: [(10, 100), (10, 100), (10, 100)]

      Follow this order: 1st Perception, 2nd Perception… 1st Action, 2nd Action ... 1st Hybrid, 2nd Hybrid …
    - Perception Battery Limit: [500]
    - Hybrid Battery Limit: [500, 500]
- **Note:** If the user leave several inputs empty and wants to generate the scenario, the program will automatically fill the list with the default value

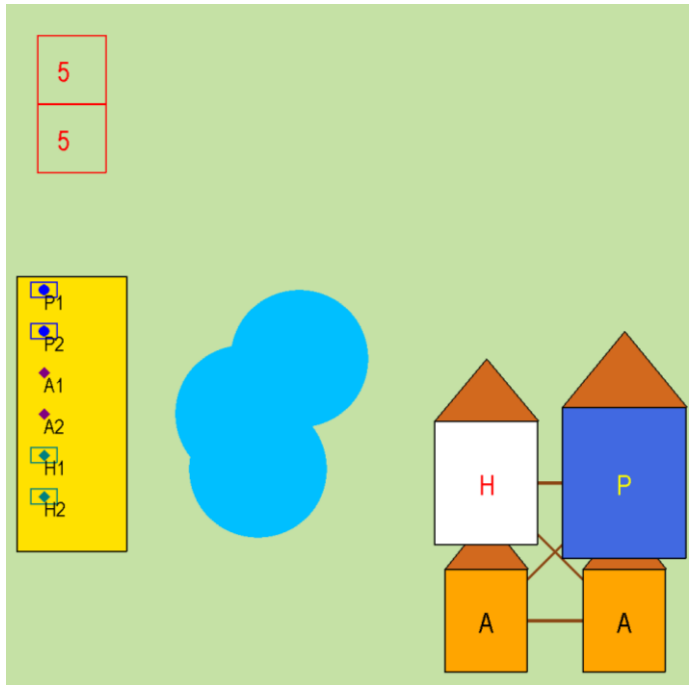
Pre-View Page

- **Pre-View**
  - Display the static objects in the scenario, including all kinds of targets, agent base and lakes
  - For agents, mark their initial positions on the agent base
  - For the fluctuate elements like the wildfire region, mark the scope of their initial position, with the number of new fire front generated at each moment

- **Grid Map**

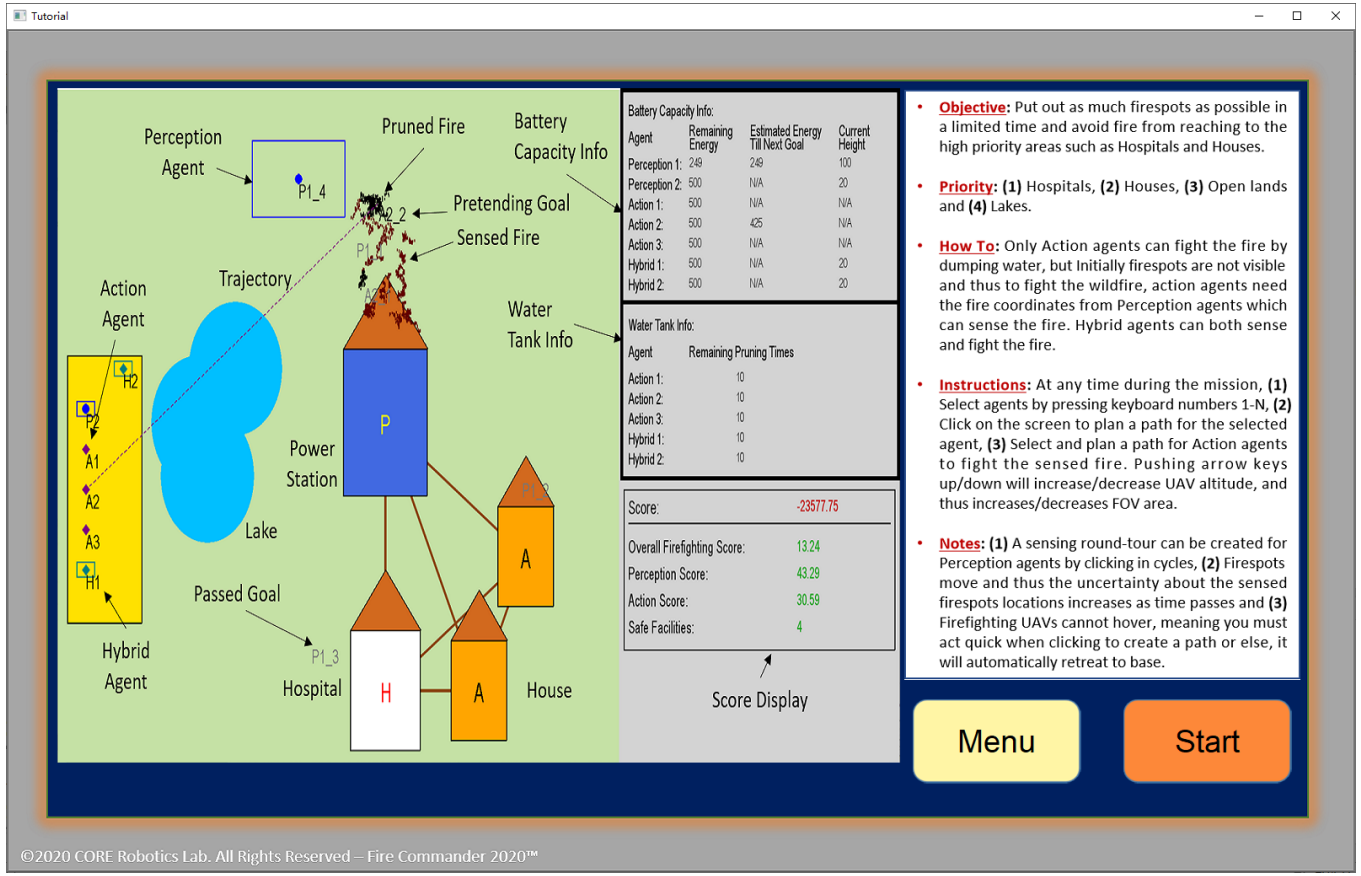
A01	B01	C01	D01	E01	F01	G01	H01	I01	J01
A02	B02	C02	D02	E02	F02	G02	H02	I02	J02
A03	B03	C03	D03	E03	F03	G03	H03	I03	J03
A04	B04	C04	D04	E04	F04	G04	H04	I04	J04
A05	B05	C05	D05	E05	F05	G05	H05	I05	J05
A06	B06	C06	D06	E06	F06	G06	H06	I06	J06
A07	B07	C07	D07	E07	F07	G07	H07	I07	J07
A08	B08	C08	D08	E08	F08	G08	H08	I08	J08
A09	B09	C09	D09	E09	F09	G09	H09	I09	J09
A10	B10	C10	D10	E10	F10	G10	H10	I10	J10

- **Pre-View**



Tutorial Page

- **Tutorial**
  - Instruct the control policy of the simulation environment
  - Offer the returning to the menu function to enable the re-design option before simulation begins



Score Display Page

General Evaluation

Positive Reward

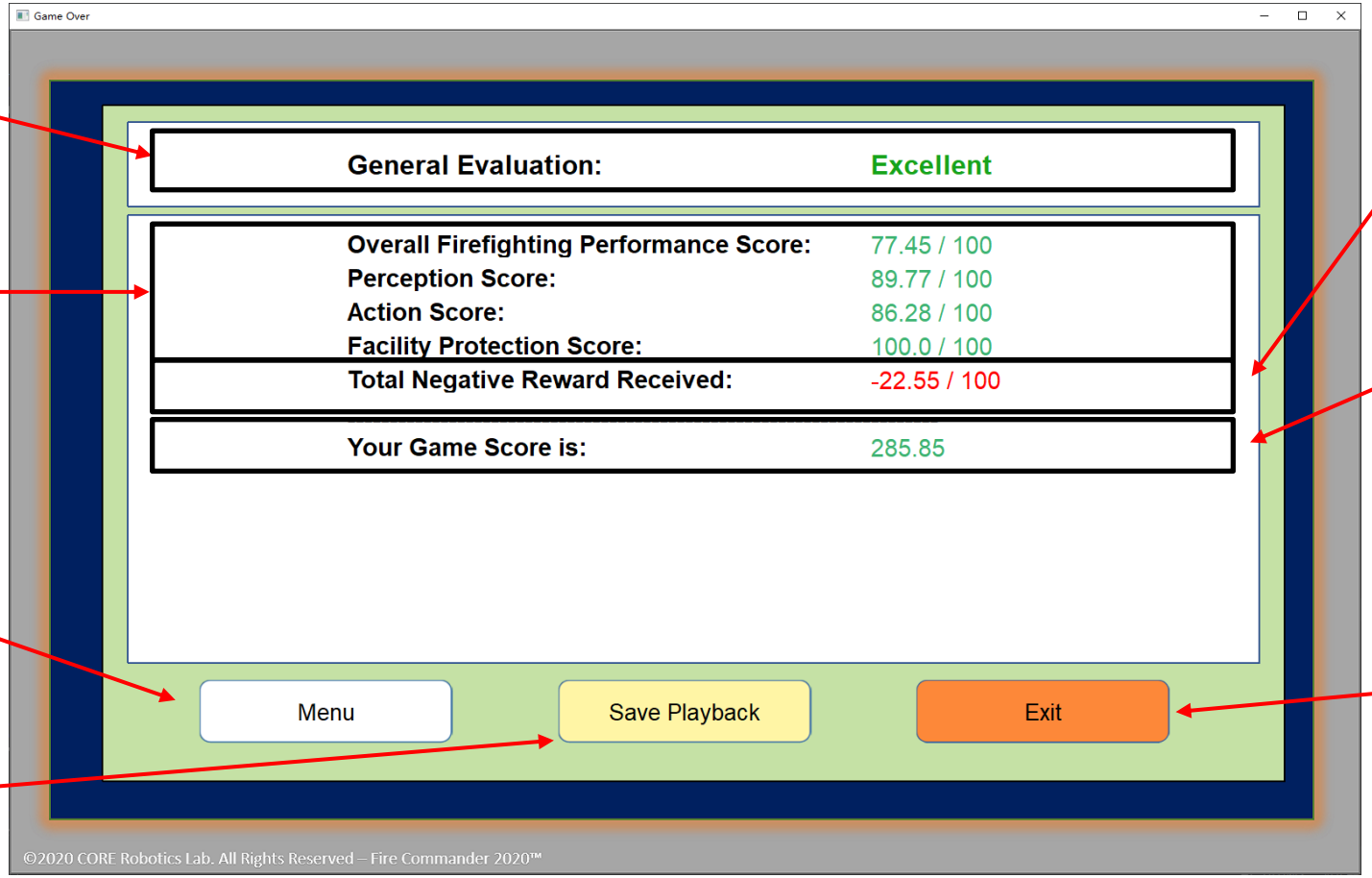
- Overall Firefighting Performance
- Perception Score
- Action Score
- Facility Protection Score

Menu

Back to welcome page, restart the simulation

Save Playback

Animation Reconstruction



Negative Reward

- Total Negative Reward Ratio: Final Online Total Negative Reward / Expected Negative Reward

Final Score

Exit

Directly exit the GUI program without reconstructing simulation

Future Work

- Further modify the score calculation policy
- Finish the scenario setting
- Complete the documentation
- Package the executable file, upload it onto the online platform

Reference

[1] Argall, Brenna & Chernova, Sonia & Veloso, Manuela & Browning, Brett. (2009). A survey of robot learning from demonstration. Robotics and Autonomous Systems. 57. 469-483.

[2] Finney, Mark A. *FARSITE, Fire Area Simulator--model development and evaluation*. No. 4. US Department of Agriculture, Forest Service, Rocky Mountain Research Station, 1998.

[3] Seraj, Esmaeil, Andrew Silva, and Matthew Gombolay. "Safe Coordination of Human-Robot Firefighting Teams." *arXiv preprint arXiv:1903.06847* (2019).

[4] Pham, Huy X., et al. "A distributed control framework for a team of unmanned aerial vehicles for dynamic wildfire tracking." *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017.