

States

States are used by the planner to determine of a set of actions can be used to achieve a goal. The term states covers goal, effects and preconditions.

State

Key and value. Where the key can be a precondition or an effect.

For example, there can be a precondition of Key: `HasAxe` Value: `1` to use the action `CutTree`

Goal

The goal is set in the inspector on the Agent class. You can choose a goal to be once or not. The priority is as it says. You may have multiple goals and they can all be achieved. The planner will choose the one with higher priority and that will determine which set of actions the agent will use.

A goal is essentially the desired effect or state to be in. Every action is built to achieve a goal. Not all actions directly take you to the goal. For example, the `GoTo` action effect will often time allow the agent to get in range to perform another action, that action effect may or may not achieve the goal.

For example `GoTo` → `Attack`, where the `GoTo` effect is to be `InRange` and the `Attack` effect is to `HurtTarget` so the goal here is `HurtTarget`



Goals should not take too long to achieve and should be achievable

World State

The world state is a blackboard for all Agents to share. A state in the world is persistent and can satisfy an Agent's precondition.

This is useful if let's say you are making an office simulation. A world state can be the number of toilets available. The Action `UseRestroom` will have a precondition of

`Toilets > 1`

```
// Adds a state called "StateName"
World.AddState("StateName", 1);
// Adds one to StateName
World.ModifyState("StateName", 1);
// Sets StateName to 100
World.SetState("StateName", 100);
World.RemoveState("StateName");
```



A world state is one way for agents to communicate to each other, like the internet!

Agent State

An Agent's state is similar to the world state but it is only for the Agent. On either an Agent or an Action, accessing States is the same as the World State.

```
States.AddState("Tiredness", 1);
States.ModifyState("Tiredness", 1);
States.SetState("Tiredness", 1);
States.RemoveState("Tiredness", 1);
```