Sensors

What is a Sensor?

A sensor is a **Unity Component** but it is more of a concept that is necessary to the GOAP system. You don't necessarily need to inherit `Sensor` but inheriting it will get access to the Agent.

Why Sensors?

Most the time actions may rely on other components, such as an **Attack** action needs a target and same goes to the **GoTo.** Most, if not all the time, the target will be the same. In this situation, an **EyeSensor** to set the target help decouple and keep the project **maintainable**.

Here's is an example of setting the target on the Agent and setting the state.

```
public class DirectTargetSensor : Sensor
    {
        public Transform Target;

        public override void OnAwake()
        {
            AgentData.Target = Target.transform;
            Agent.States.AddState("hasTarget", 1);
        }
}
```

In the example above, the Agent already has a target in the Data, but let's say you are not using the that BasicAgentData pack. You can directly use the value of the sensor.

Say you have an action that requires to be **InShootingRange.**

```
public class RangeSensor : Sensor
{
```

```
    public float Range = 5;

    private void Update()
    {
      var distance = AgentData.DistanceToTarget;
      if(distance <= Range)
        Agent.States.Set("InShootingRange, 1);
      else
        Agent.States.Remove("InShootingRange");
    }
}
```

Here's the cool part, your actions can use the Range value from the sensor and you don't have random numbers assigned everywhere

```
public class GoTo : BasicAction
{
    public RangeSensor RangeSensor;

    public override EActionTask Perform()
    {
      if(_agentData.DistanceToTarget <= RangeSensor.Range)
        return EActionStatus.Success;

      return EActionStatus.Running;
    }
}
```