Università degli studi di Milano
Edoardo Carrer (967549)

Goal Oriented Behaviour for AI in Unity

Table of Contents

1. Introduction **2**

2. Agents Description **3**
FSM 3
GOB 4
PARENTS 5
CHILD 6

3. Script interface **7**
Child 7
Parent 7
Goal 7
Action 8
Effect 10
Tool 10

4.Interact with the AI **11**

5. Last consideration and future works **11**

1. Introduction

This project focuses on the usage of GOB for simulation within Unity of a family with probable actions & reactions based on other NPC's action and random events with a limited amount of complexity and a structure that allows a designer to easily manage the behaviour. The GOB is a simple and flexible decision-making system that allows emergent behaviour and could be declined in a huge amount of different types of game that needs a discrete number of NPCs with believable behaviour. Basically the NPC evaluates a "discontentment" that is a sum of insistences that define how much is important to fulfil each specific goal in that moment.

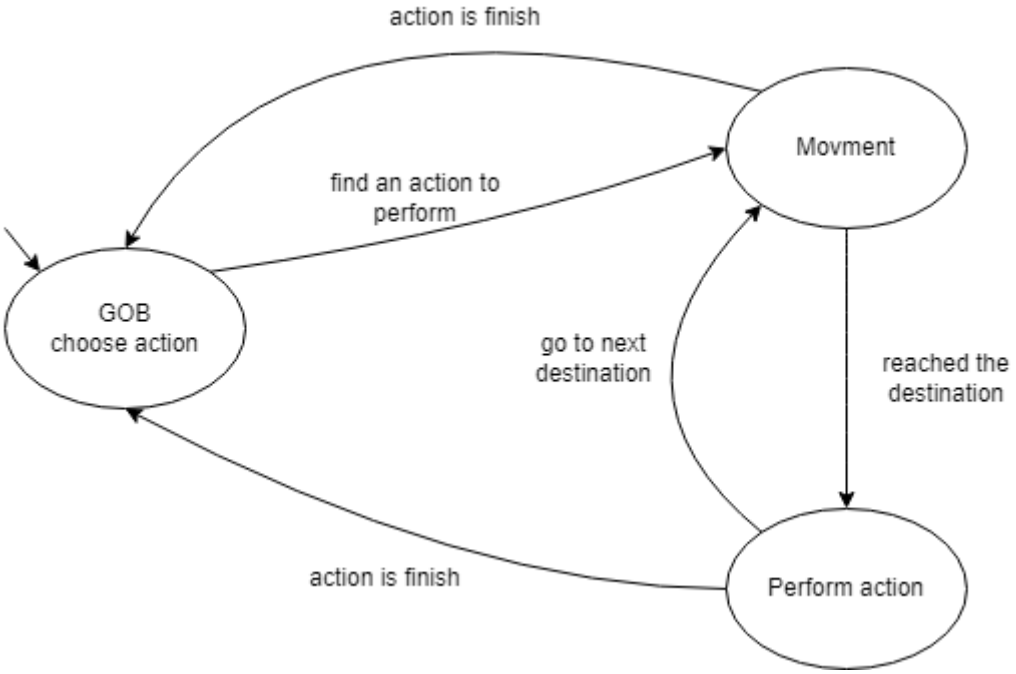
In this project there's a family composed of 3 NPCs, two parents and one child, in a "the sims" like environment with some rooms in the house, few external destinations, some resources and some tools. One hour of game time is 8 seconds, the day is divided in 4 part:

- **Night**: from 0 to 6
- **Morning**: from 6 to 12
- **Afternoon**: from 12 to 18
- **Evening**: from 18 to 24

2. Agents Description

FSM

The AI of the NPCs are implemented with a simple Finite state machine, with 3 states for separate the decision-making part with the movement and the performer state, so every of them has a list of goals depends from the role in the project, in this case we have 2 parents and one child, and they try to lower their discontentment. To fulfil this goal they have a set of actions that is composed of effects that affect one or more goals.



In the Choose action state the AI evaluate each available action impact on the discontentment value to found the best action to perform, then if at least an action is available to be performed the FSM pass to the Movement state

The Movement state simply sets a destination in the NavMeshAgent and the agent starts heading to the destination; if the agent reaches the destination it switches to the Execute Action state; the current action can be set as finished if something happens before reaching the destination, in which case the FSM switches to the Choose Action state.

In the Perform action state, the agent firstly check if it is possible to perform the action, it's needed to re-check because in meanwhile the agent arrive to the destination another agent could have perform an action and so change resources or broke tools, then update the resource and the tool state, and after the duration of the action, the effects is applied on the goals insistences and on the number of resources, some actions as more destination to reach to perform, in this case the FSM pass to the movement state until it reaches the last destination.

GOB

The focus is on the choose action state that implement the Goal oriented behaviour, in this state the agent start with cycling over the action list, and for each there verify that the action can be performed checking if there is enough resources, available tools and it it is the right time of the day to perform it, then if there is no impediment to perform the action, the value of every goal insistencies is changed based on the estimation time to arrive at the destination and the time to perform the action, the agent doesn't know how much a goal changes every second but it take an estimate based on how it changed over time with this simple formula:

```
var rateSinceLastTime :float = changeSinceLastTime / timeSinceLast;
basicRate = 0.95f * basicRate + 0.05f * rateSinceLastTime;
```

- **changeSinceLastTime**: how much the insistence changes from last evaluation
- **timeSinceLast**: how much time is pass from last evaluation\
- **basicRate**: the actual rate of growing that the agent perceive
- **rateSinceLastTime**: the rate of growing perceive from the last evaluation

In the evaluation a goal near 0 insistence could have a negative basic rate that invalidates all the possible action and the agent can't perform any action, so the basic rate can't be lower than 0.

The effects of the action is evaluated to the specific goal insistence, and on the number of resources, then is evaluated the total discontentment, at the end every action discontentment is compared and the lower one is elected to be performed, then FSM passes to the Movement state. All these changes are not applied to the real world model, it's only an internal evaluation of values.

NPCs have some common goals and actions, but to differentiate the behaviour each of them have some different goals and actions.

Most goals insistence increases every second with a constant value, but there are some goals that are not time-dependent.

For each goal that reaches 100 of insistence the duration of the actions is increased by 20% of the initial value to simulate a stress state of the agent.

The common goals for every NPCs are:

- **Energy**, how much the NPC is tired
- **Bladder**, how much is insistent the need to go to the bathroom, if it reaches 100 of insistence it becomes 0 but the insistence of the Hygiene goal is put to 100.
- **Hygiene**, how much is insistent the need of a shower
- **Hungry**, how much the NPC is hungry
- **Fun**, how much the NPC want to take time for an hobby or passion

The way the effect's action was designed is that it has to take the agent to choose the next action in a believable way, so if an agent finishes a training, I expect to see it take a shower,

and the estimate of the rate changing do it making more important a goal that has a rapid increase.

The common type of actions are:

- **Sleeping/resting**: decrease the energy insistence
- **Eating**: decrease the hungry insistence and increase energy and bladder insistence
- **Use toilet**: decrease bladder insistence, need a WC
- **Wash up**: decrease hygiene insistence, need a shower

PARENTS

Parent1

Parent1 has two more goals to fulfil:

- **Work**, its changing rate depends on the amount of money remaining, it takes the NPC to reach the office and perform an action to obtain money.
- **House Safety**, it is a bit different goal, his insistence doesn't change with a constant value every second but it depends from the broken tools, every tool as an insistence that is sum when it broke depend from the importance(ex. WC is way more important than the console for the immediate use), the agent goes every time to the tool with the max insistence to repair it.

And also has these type of actions:

- **Working**: decrease work insistence and increase fun insistence
- **Repair**: decrease house safety insistence
- **Playing video game**: decrease fun insistence, need a console
- **Watch TV**: decrease fun, need a TV

Parent2

Parent2 has four more goals to fulfil:

- **Food needs**, it depends on the amount of raw food available, and it takes the agent to the market to buy it.
- **Family hungry**, it depends on the all agents hungry goals, it takes the max value of hungry insistence in the family to decide to cook, after there is the meal the insistence depends from the max value less the eating meal effect for each meal is available, for example if the insistences are 50 60 70, and there is 2 meal, the insistence will be 50, cause is taking in account that 2 of three people could already eat.
- **Clean house**, the amount of dirty accumulated during time in the house
- **Physical health**: the need to train.

And also has these type of actions:

- **Going to market**: decrease food need insistence, increase the amount of raw food, decrease the amount of money
- **Cooking**: decrease family hungry insistence, decrease the amount of raw food and increase the amount of meal
- **Cleaning**: decrease clean house insistence
- **Training**: decrease Physic health insistence, increase energy and hungry, decrease amount of money
- **Watch TV**: decrease fun, need a TV

CHILD

Child has two more goals

- **Homework**, the needs to complete the homework
- **Social**, the needs to chat or to play with friends

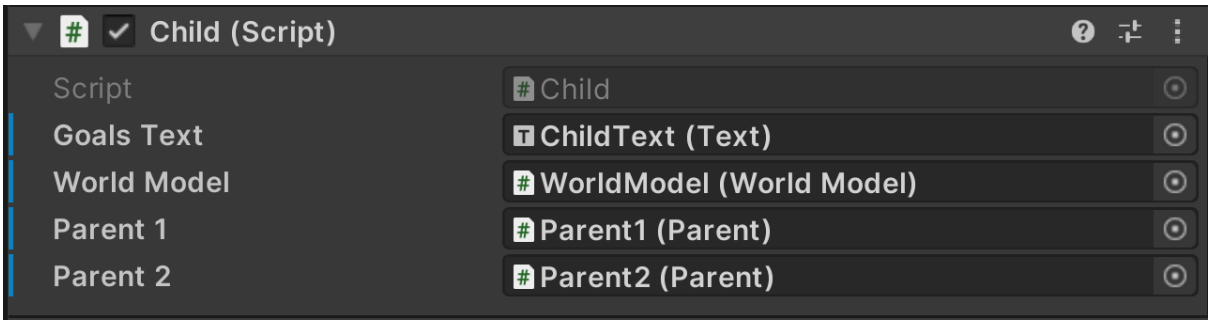
And also has these type of actions:

- **Play at park**: decrease fun and social insistence, increase hygiene and energy insistence
- **Social network**: decrease social insistence
- **Study**: decrease the homework insistence, increase energy and fun insistence
- **Play online video game**: decrease fun and social insistence, need a console

To add believability when the child reaches 100 of insistence on one goal set as critical it starts crying, the goal that becomes critical is added on the nearest parent goals list and a set of specific actions is added to that parent to perform actions that fulfil the child's goals, so the current action of the parent is stopped and a new evaluation starts.

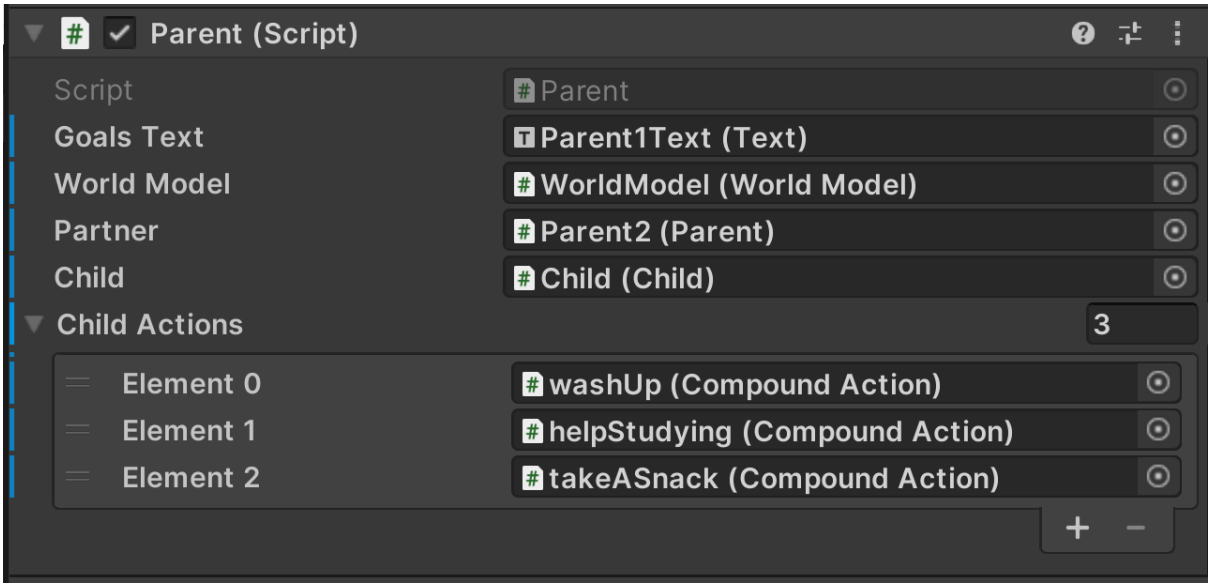
3. Script interface

Child



- **Goals Text**: text where is written all the goals insistence
- **World model**: representation of the world model, to have access to the amount of resources
- **Parent1**: reference to Parent1
- **Parent2**: reference to Parent2

Parent



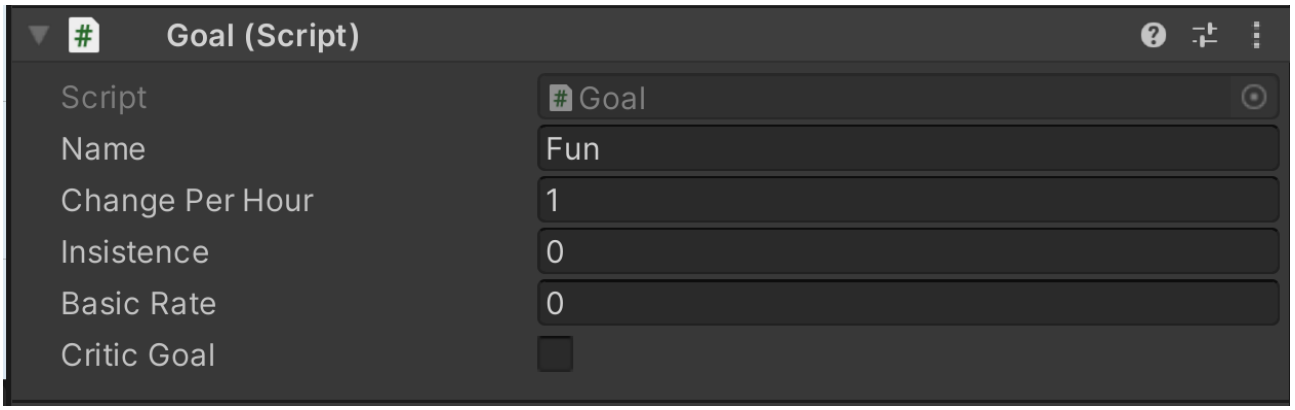
- **Goals Text**: text where is written all the goals insistence
- **World model**: where is save the amount of resources
- **Partner**: reference to other parent
- **Child**: reference to Child
- **Child Actions**: actions that is evaluated only when child is crying

Goal

Goal is composed by:

- **Name:** string to identify a goal

- **Change Per Hour:** how much the goal change each game hour
- **Insistence:** more high is this value, more is important to fulfil the goal
- **Basic Rate:** how is changing the insistence during time, add believability giving more importance to goals that are increasing fast
- **isCritic**: if the agent needs help when the goal reaches 100 of insistence

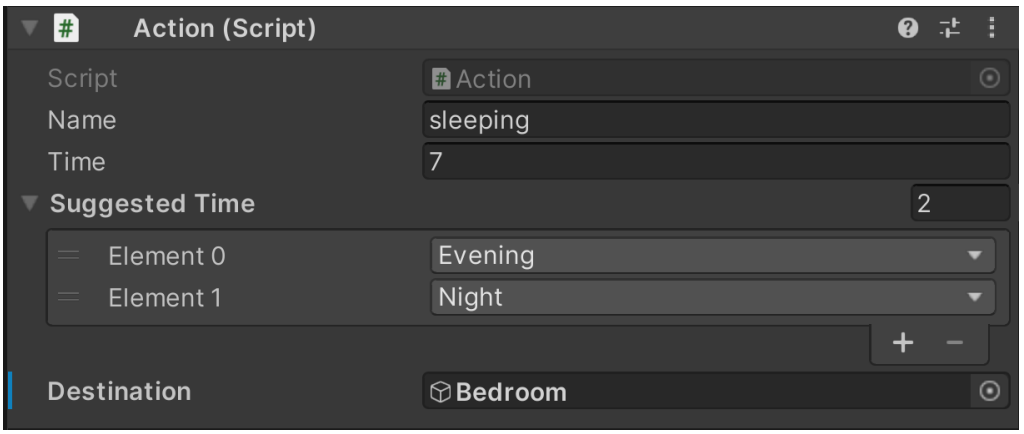


Action

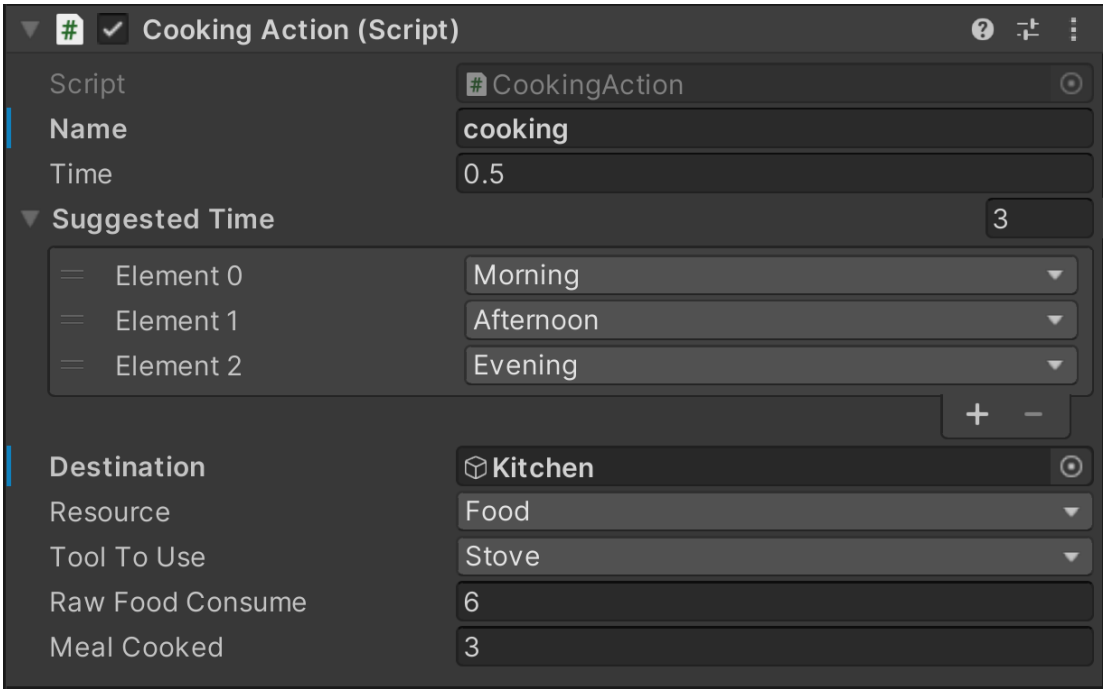
Action could have different effects that affect differents goals and also goals of other agents

The action could be

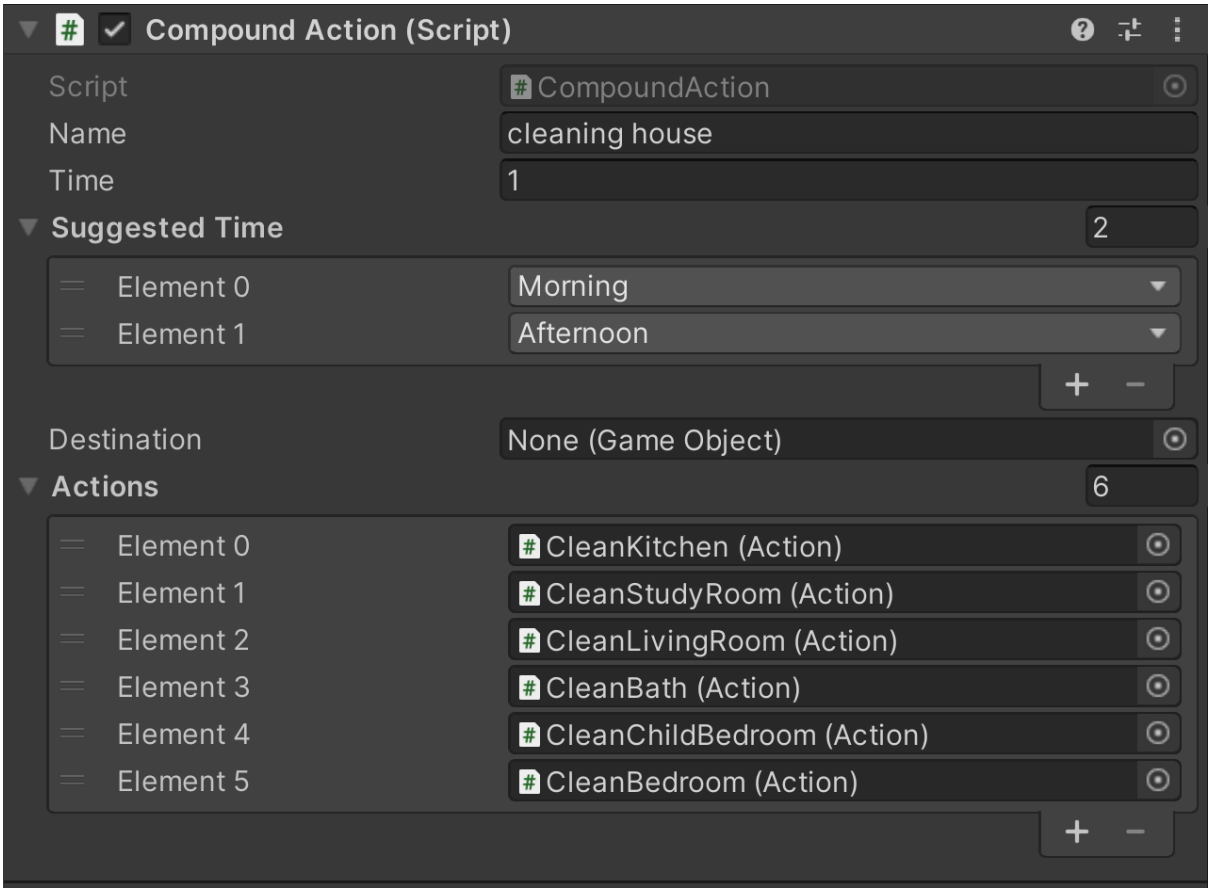
- Normal, no need of tools and resource, just affect insistence when finish, like sleeping, and could be personalised easily from the inspector. Every action is composed by:
  - **Name:** string to identify an action(useful for debug)
  - **Time:** the duration of the action in game hours
  - **Suggested Time:** the part of the day when is possible to perform the action
  - **Destination:** transform where is need to go to perform the action



- Specific, depends from the action, it changes the way it uses resources and tools or how affects other goals, some action inherited from this class and add some attributes.
  - **Resource:** the type of resource that is involved in this action
  - **Tool To use:** the type of tool that is involved in this action



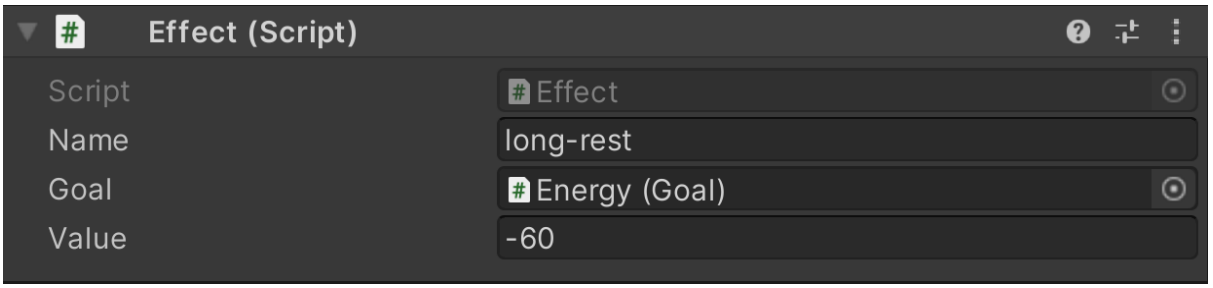
- Compound, that encapsulate other actions, so every single action is performed and it finishes when the last action of the sequence is finished, the destination is taken by the first action in the sequence, and then it changes every time an action finishes.
    - **Actions**: contains the whole sequence of actions to perform



Effect

Effects are in the same gameobject with the corresponding action. Is composed by:

- **Name**: string to identify an effect(useful for debug)
- **Goal**: the goal that the effect affects
- **Value**: the value that affect the goal insistence

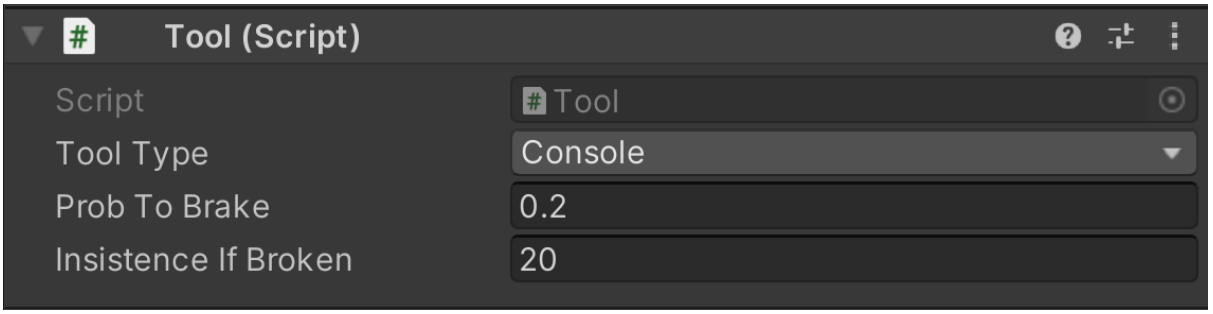


Tool

Tools are objects in a specific location that is needed to perform an action, it could break and can be used from one agent at time.

It's composed by:

- **ToolType:** the type of tool that represent
- **Prob to brake:** the starting probability that the tool brake when an agent use it, every time it is used this value increase
- **Insistence if broken:** the insistence added to the safety house goal after the tool broke



4.Interact with the AI

To test AI there is some action that is possible to do to affect the NPCs:

- Pushing an icon tool to break it.
- Pushing button event to change some goals insistence of NPCs:
    - Damaged food event: raw food is set to 0
    - Crunching event: work insistence increase by 50
    - Exam event: homework insistence increase by 50
    - Stomachache event: bladder insistence increase by 50
    - Friend's call event: social and fun insistence increase by 50

5. Last consideration and future works

The GOB is very flexible, so it is possible to modify some small parts of the code to achieve different results depending on the game, for example, it is possible to easily implement goal-oriented action planning that gives the agent the ability to estimate a sequence of actions, and it could be a good improvement for credibility, but this leads directly to a huge increase in complexity, $O(n*m^k)$ vs. the $O(n*m)$ of the normal GOB, where n is the number of goals, m the number of actions and k the length of the sequence of actions, which is not acceptable for a game with many NPCs and actions, A* could be a solution but it only needs one goal to be fulfilled and so it was not suitable for the project, it could be a better solution in a game where each agent has a specific task to fulfil and there are no daily life goals.

For the calculation of discontentment there is another option instead of the estimate change described above, it is possible to have the fixed value of the rate changing, but for this project, which also has some goals that change regardless of time, the estimate change solution gives more credibility when evaluating the best action.

In a complete game, with more NPCs, it's possible increase believability with:

- Add mood that represent psychological state of the agent
- Add dialogues of agent that depends from the overall discontentment and specific events
- Add a value to each relationship with other agents to affect the type of dialogue based on mood and the overall discontentment
- Add different types of resources with different effects and cost
- Improve the estimated time to arrive in a place with precomputed pathfinding
- Add animation for movement and performing state
- Making events occur randomly over time