

User documentation

DNAI

11/23/2018



Synopsis

This document is a user guide allowing you to learn how to use the various services of the DNAI software suite aimed at simplifying access to the field of artificial intelligence.

Learn how to use artificial intelligences that you have created but also to find the hidden gems among the creations of the community on our online Hub.

Learn the integration of artificial intelligences in video games with a complete tutorial on our extension for the Unity3D editor.

You can find our complete solution and documentation at <http://dnai.io>.



Metadata

Title	2019_UD1_DNAI
Date	09/07/2018
Authors	Adrien Wery, Axel Vencatareddy, Louis Vezia, Quentin Gasparotto, Victor Gouet, Mathieu Tavernier, Fernand Veyrier, Nicolas Constanty, Fiona Roch
Manager	Victor Gouet
Email	dnai_2019@labeip.epitech.eu
Subject	User documentation for the DNAI suite
Keywords	Documentation, user, DNAI, artificial intelligence, editor, graphical
Version	1.0



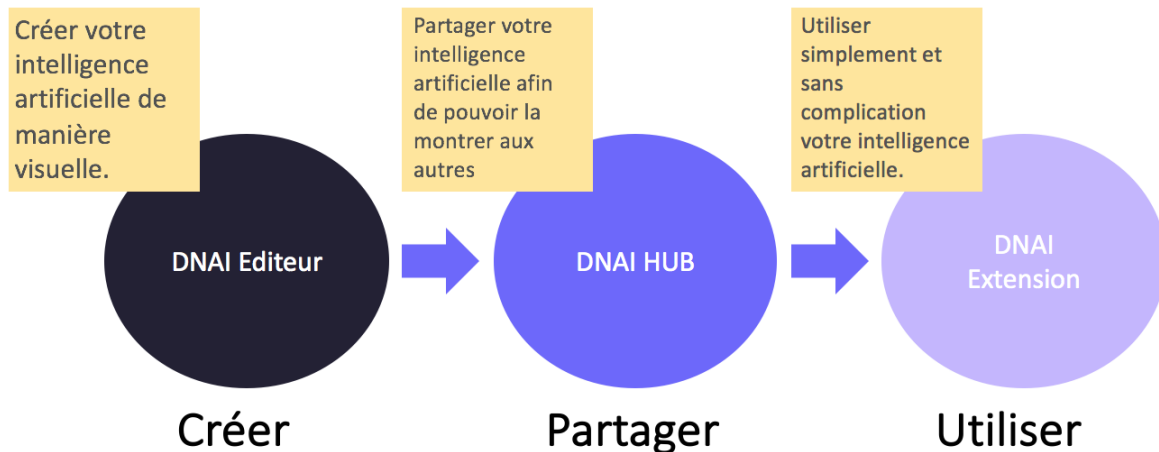
Table of Contents

Synopsis	1
Metadata	2
DNAI briefly	4
FAQ	4
How can I report a bug?	4
Unity3D extension	5
Introduction	5
What is the role of the Unity3D DNAI plugin?	5
What actions will I be able to perform via the plugin?	5
Starting the DNAI plugin	5
Plugin usage	6
Login to a DNAI account from Unity3D	6
Add and remove a DNAI AI from a project	7
Use an imported AI from the plugin	8
DNAI and scripting	8
FAQ	10
Do I need to have a DNAI account and log in to use the plugin?	10
Can I download an AI from the Hub without being connected?	10
Conclusion	11

DNAI briefly

DNAI (*Design Node for Artificial Intelligence*) is a solution where the aim is to quickly integrate artificial intelligences in projects (on Unity3D in this case).

The principle is as follows:



Our solution provides software that can create artificial intelligence, a HUB to share creations from the latter and an extension to Unity3D (software development video games) to integrate all the creations of the HUB.

Our solution allows to create, share and use artificial intelligences. However, a single user may, for example, focus solely on the creation of artificial intelligence. We want to create a self-help solution with a community aspect.

So, if you want to use one of our artificial intelligences, you just have to go on our HUB, download the one you want and integrate it with our plugin. You will not need to go through our Graphic Editor.

FAQ

How can I report a bug?

You can access our forum to report any issue you encountered while using our plugin.



Unity3D extension

Introduction

What is the role of the Unity3D DNAI plugin?

The Unity3D DNAI is a bridge between artificial intelligences created by the DNAI editor and your Unity3D game. Whether you have created an AI from the editor yourself or recovered an AI from the DNAI HUB, you will need it to use these AIs under Unity3D.

Its role is to recover one or more .dnaï files and turn them into libraries that will be usable in your Unity3D game.

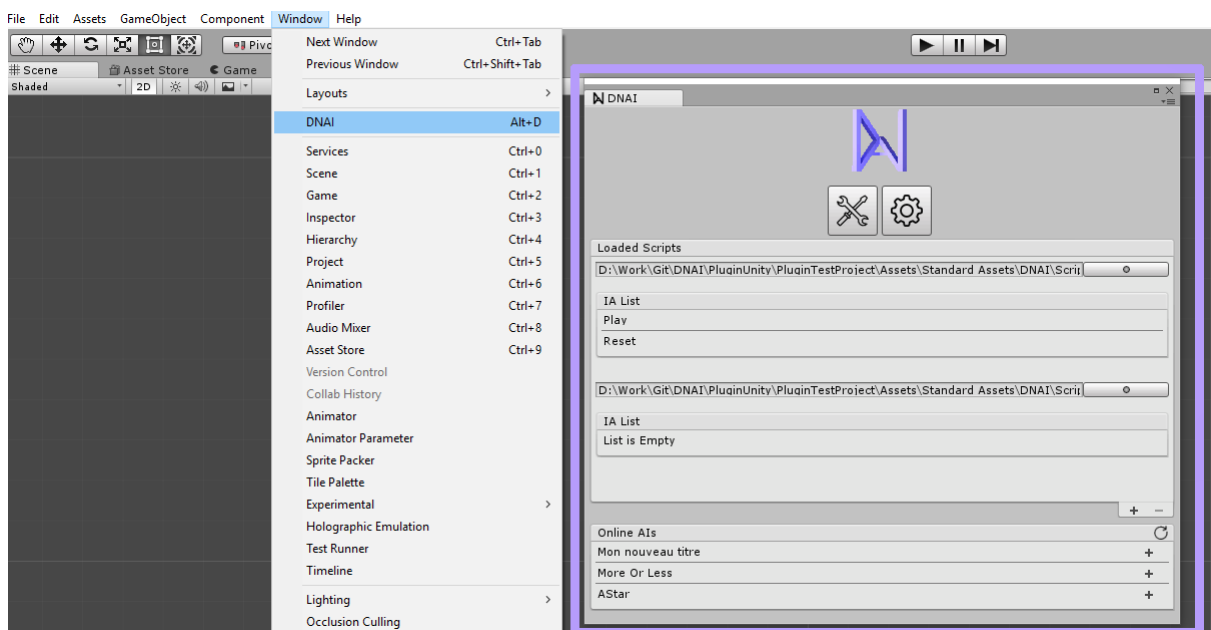
What actions will I be able to perform via the plugin?

It will be possible to perform several actions using the plugin:

- Fetch a .dnaï file and import it into your game
- Transform a .dnaï file into a library and use it in your game
- Log in to your DNAI account
- Recover your AI stored on the DNAI Hub

Starting the DNAI plugin

After downloading your plugin from the store, you will find it under the **Windows** tab or by pressing **Alt+D**.



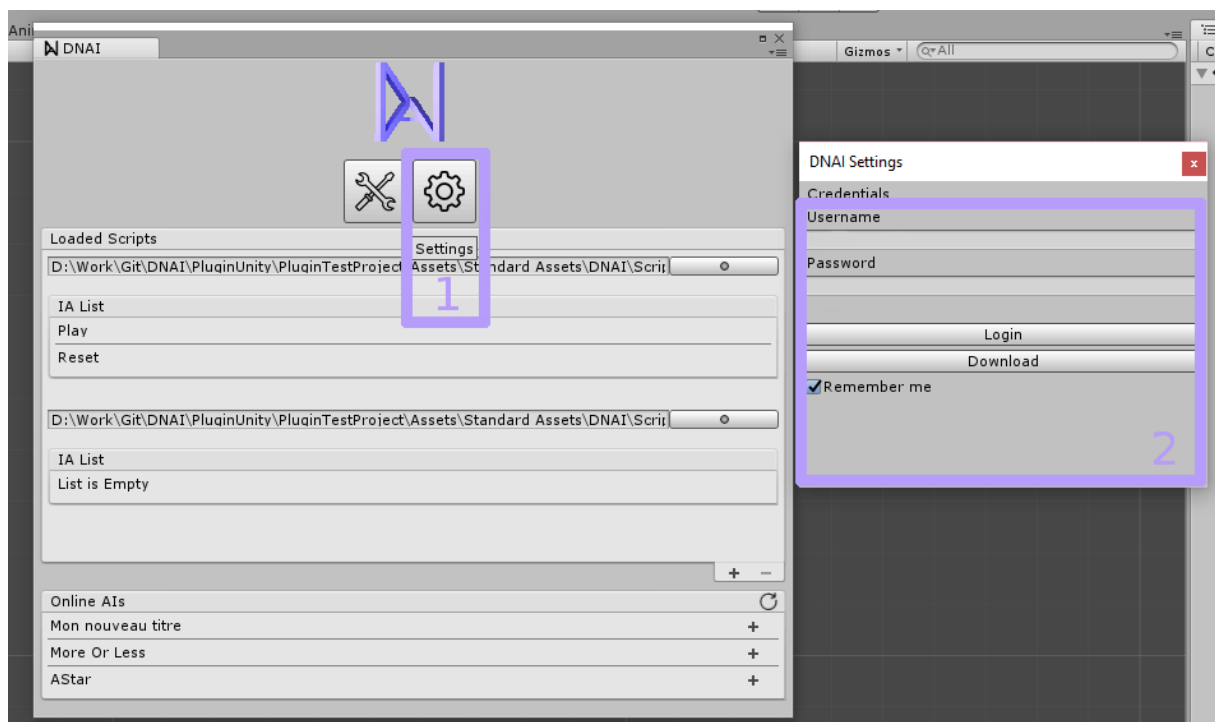


Plugin usage

Login to a DNAI account from Unity3D

To login to your DNAI online account from Unity3D, you must first open the DNAI plugin control panel from your Unity3D software. To do so, open the Windows / DNAI menu. The control panel window then will appear.

1. From this panel, select the Settings logo (the second logo below the DNAI symbol). The DNAI setting window opens.
2. Enter your username and password and select "login" to log in. "Connected" is displayed above the login button: you are now logged into your DNAI account.

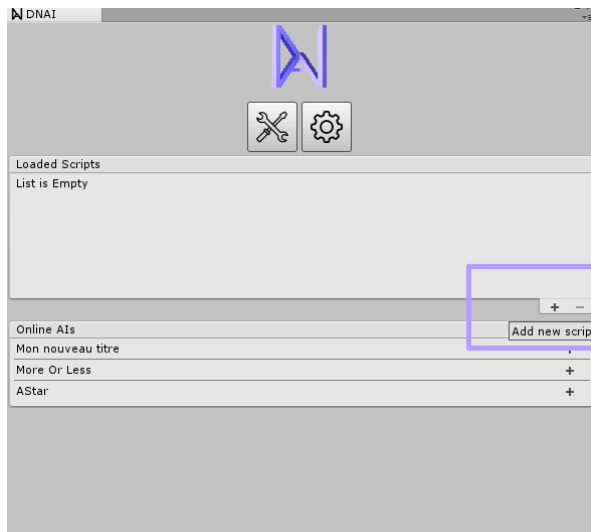


You can now close the settings window and re-center on the DNAI control panel. It allows you to add recovered AIs from the HUB or just through your computer.

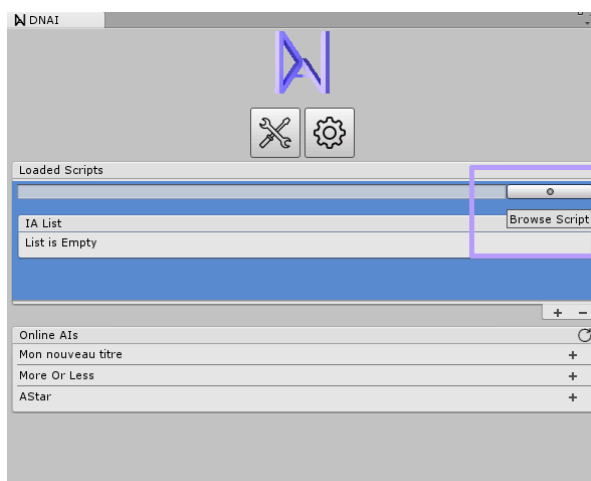


Add and remove a DNAI AI from a project

AI recovered from my computer

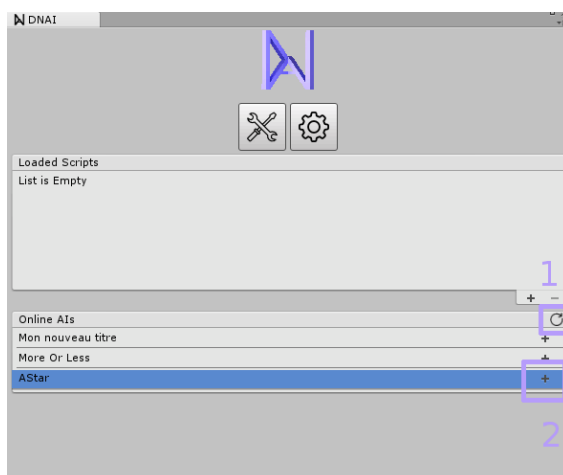


To add a recovered AI from your own computer, first open the DNAI control panel from Unity3D. Select the "+" cross of the "Loaded Scripts" section to add a new script slot to fill.



Click the button to the right of your script slot and select in your files the one you want to import.

IA recovered via my DNAI account

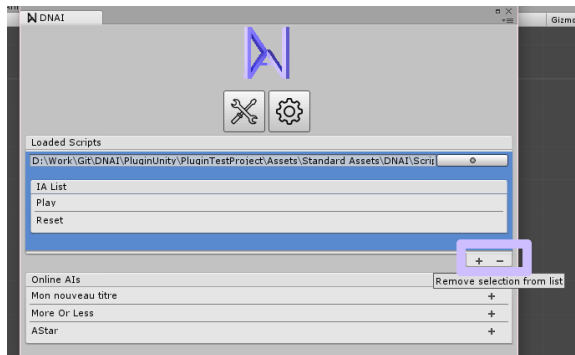


After logging in, you can retrieve an AI uploaded to the HUB from the Unity3D DNAI control panel:

- 1- Refresh your online AI list by clicking on the refresh button
- 2- Add the AI you want by clicking on the "+" button



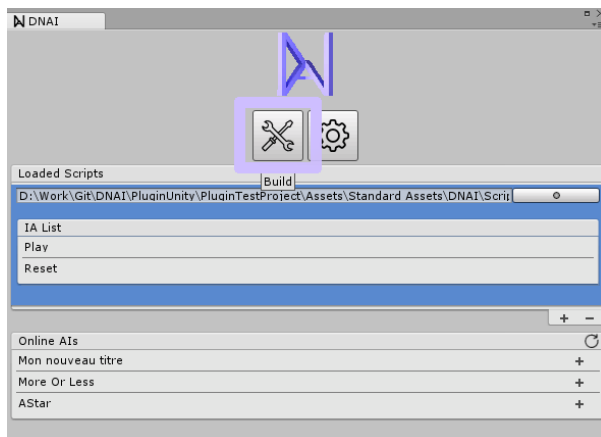
Remove an AI from my project



To remove an imported AI from your project, open the DNAI control panel from Unity3D.

Select the desired script then click on the "-" button at the bottom right to delete it.

Use an imported AI from the plugin



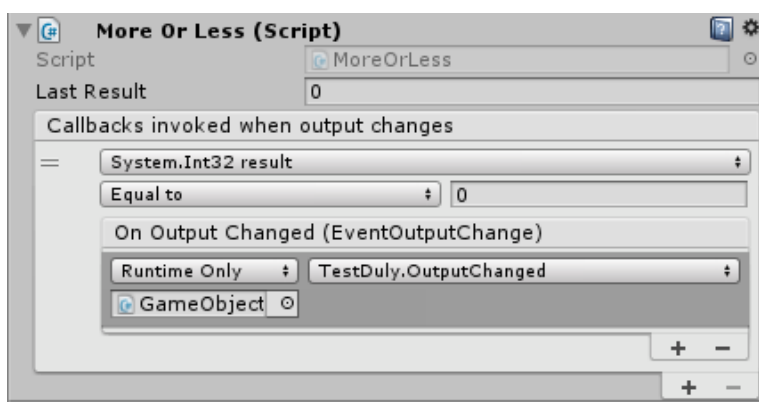
In order to use the AIs you have imported into Unity3D, you will need to generate code from them. To do this, open the DNAI control panel under Unity3D and select the "build" button below the DNAI logo.

Once the operation is complete, you can use your imported AIs.

DNAI and scripting

Once the AI is imported and compiled, a script is generated. We can integrate the AI into the game.

This allows to expose two elements: inputs (here **LastResult**) and a list of callbacks that can be invoked when the value of an output changes.



Depending on the type of the variable, different options are displayed.

It is possible to inherit this script in order to customize it.



```
5 using System.Collections.Generic;
6 using UnityEngine;
7
8 namespace DNAI.MoreOrLess
9 {
10     ...public class MoreOrLess : MonoBehaviour
18     {
19         ...public List<ConditionItem> _cdtList;
21         ...public int lastResult;
29
30         public MoreOrLess();
31
32         public int result { get; }
33
34         ...public void ExecutePlay();
44         public void ExecuteReset();
45
46         public enum COMPARISON
47         {
48             MORE = 91,
49             LESS = 672,
50             NONE = 941
51         }
52     }
53 }
```

That's what the class looks like. It follows a certain nomenclature: all the methods will be called "Execute ----" and will correspond to the names that will be given in the plugin window below the loaded script.

In order to be able to assign a dynamic method in Unity3D through callbacks, you must respect the following signature:
Public void FxName (EventOutputChange e)

```
5 using System;
6
7 namespace DNAI.MoreOrLess
8 {
9     public class EventOutputChange : EventArgs
10     {
11         public MoreOrLess Invoker;
12         public object Value;
13         public Type ValueType;
14
15         public EventOutputChange();
16     }
17 }
```

Invoker is the script that invoked the callback, Value is the value of the variable that has changed, ValueType its type (string, int, class...).

Classes can also contain nested classes as in the Pacman script. These nested classes can contain any type of method.

```
7
8 namespace DNAI.Astar2
9 {
10     ...public class Astar2 : MonoBehaviour
18     {
19         ...public List<ConditionItem> _cdtList;
21
22         public Astar2();
23
24         public class Position
25         {
26             public float X;
27             public float Y;
28             public float Z;
29
30             public Position();
31         }
32         public class PosGraph
33         {
34             public object links;
35             public object nodes;
36
37             public PosGraph();
38
39             public int appendNode(Position node, PosGraph @this);
40             public Position getNode(int index, PosGraph @this);
41             public void linkNodes(int from, int to, bool bidirectionnal, PosGraph @this);
42             public object pathFindAstar(int from, int to, PosGraph @this);
43             public object rebuildPath(PosGraph @this, int from, int to, object preceders);
44         }
45     }
46 }
```

This allows for example to manipulate nodes in a position graph to use an AStar algorithm.



FAQ

Do I need to have a DNAI account and log in to use the plugin?

No, it is possible to use the DNAI plugin by recovering AIs from your own computer created from the editor, so not necessarily by being connected.

Can I download an AI from the Hub without being connected?

No, because to download a HUB AI, you will need to log in to your account to recover the desired script.

The DNAI plugin for Unity3D will allow you to manage your AI from your DNAI account or import your own AI generated from the DNAI editor, to finally use them in your games or projects under Unity3D.



Conclusion

To conclude, DNAI is a software suite aimed at simplifying access to the field of artificial intelligence.

To do so, we set up the possibility to integrate files containing artificial intelligence in your video games with our Unity3D extension.

No longer worry about learning obscure concepts specific to the field of AI and retrieve the creations of our community on our online Hub.

For the most curious and creative, our graphic editor will allow you to go further and offer your own intelligences to the community.

If you encounter any problem, do not hesitate to check out our support forum.

Stay informed of upcoming news. Please visit <https://dnai.io> for further informations.