TR11 Engine Advanced Training Documents

By Ji Jiabin

Edit History

|  |  |  |
| --- | --- | --- |
| *Date* | *Content* | *Editor* |
| 2018.1.17 | Created | Ji Jiabin |
| 2018.1.18 | Add Various Tips Topic and put some already have tutorial into it. | Lin Shaowen |
| 2018.1.19 | Add animhost topic | Ji Jiabin |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Contents

[1 Introduction 4](#_Toc504147221)

[2 Advanced knowledge 4](#_Toc504147222)

[2.1 Level Structure 4](#_Toc504147223)

[2.2 Elements Introduction 4](#_Toc504147224)

[Unit 4](#_Toc504147225)

[Object 5](#_Toc504147226)

[Action Graph 5](#_Toc504147227)

[2.3 Important Knowledge 6](#_Toc504147228)

[Slices 6](#_Toc504147229)

[Components 6](#_Toc504147230)

[3 Objects Topics 7](#_Toc504147231)

[3.1 How can I find an object I want? 7](#_Toc504147232)

[3.2 How do I know how this object works? 8](#_Toc504147233)

[3.3 How can I use an exist object? 9](#_Toc504147234)

[3.4 What is parent object or child object? 10](#_Toc504147235)

[3.5 How to store an object’s state? 11](#_Toc504147236)

[3.6 How to create my own object? 12](#_Toc504147237)

[4 Action Graph Topics 13](#_Toc504147238)

[4.1 How can I modify cameras through action graphs? 13](#_Toc504147239)

[4.2 What are the basic physics node I can use? 14](#_Toc504147240)

[4.3 How can I modify persistent data through action graph? 15](#_Toc504147241)

[4.4 How can I add VO scripts for level? 17](#_Toc504147242)

[Lara arrive some place 17](#_Toc504147243)

[Lara trigger SI model and seek for clues 17](#_Toc504147244)

[Some specific event happens 17](#_Toc504147245)

[4.5 How can I add an Animhost? 17](#_Toc504147246)

[5 Debug Topics 19](#_Toc504147247)

[6 Various Tips 20](#_Toc504147248)

# 1 Introduction

This document is about some further designer skill for TR11 engine. It aims at how to use this engine faster and better. We will sort this document by different topics so that you can search for what you want easily. After each topic, there might be some further tips.

Before reading this document, please first learn the basic training document provide by Edios on our P4. Otherwise, you will be no idea about what this document is talking about.

Basic training document P4 address:

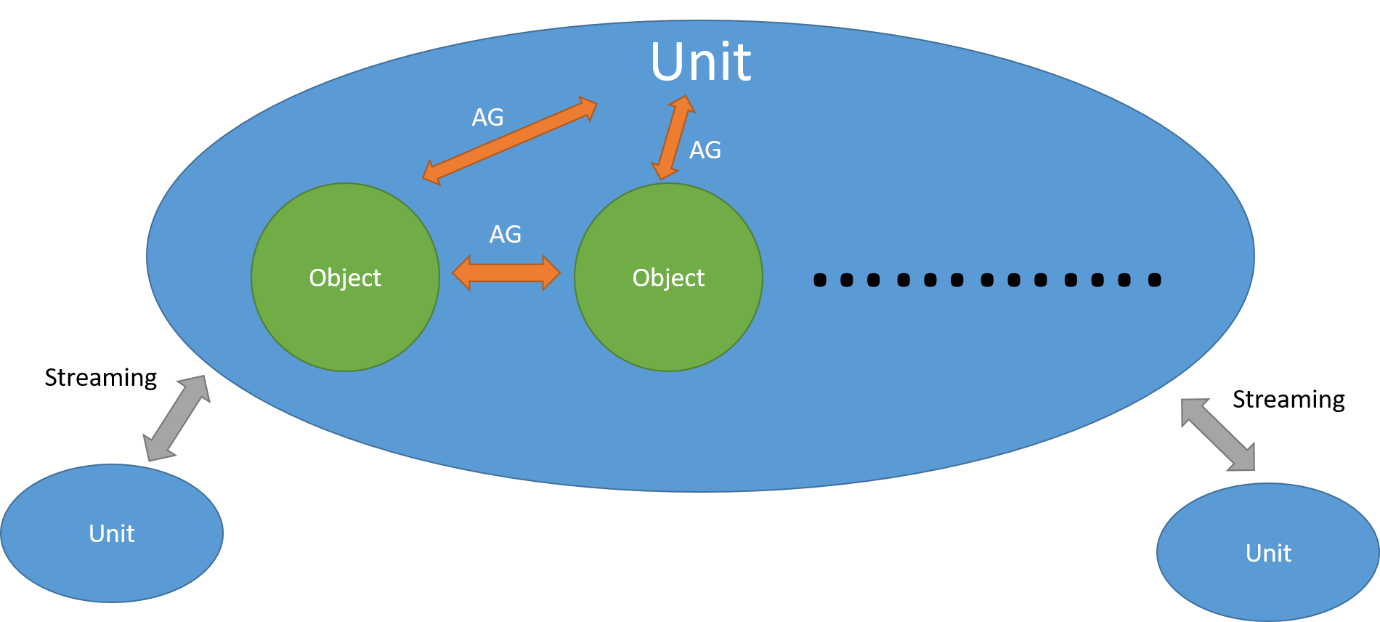
***//tr11/outsource/Training Docs/***

We also welcome everyone to share skills by editing this document.

# 2 Advanced knowledge

## 2.1 Level Structure

First let’s look at what a TR11 level’s structure is.

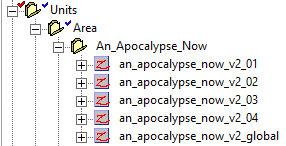


TR11 levels have several elements, ***Unit***, ***Object***, and ***Action Graphs,*** etc. (like meshes and wwises) A basic level is one unit. Units contains objects and other things like meshes and action graphs. Units are connected by streaming settings. Units and objects are talking through action graphs.

## 2.2 Elements Introduction

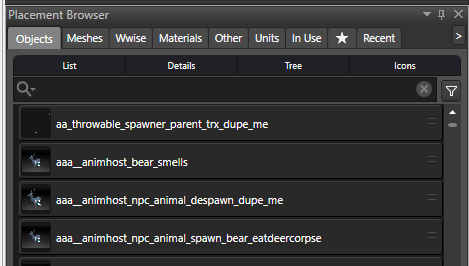
### Unit

Unit is just what we called a “map”. It contains basic terrain, skybox and so on. We usually start a level through create a new unit. It’s the basic foundation of our level.



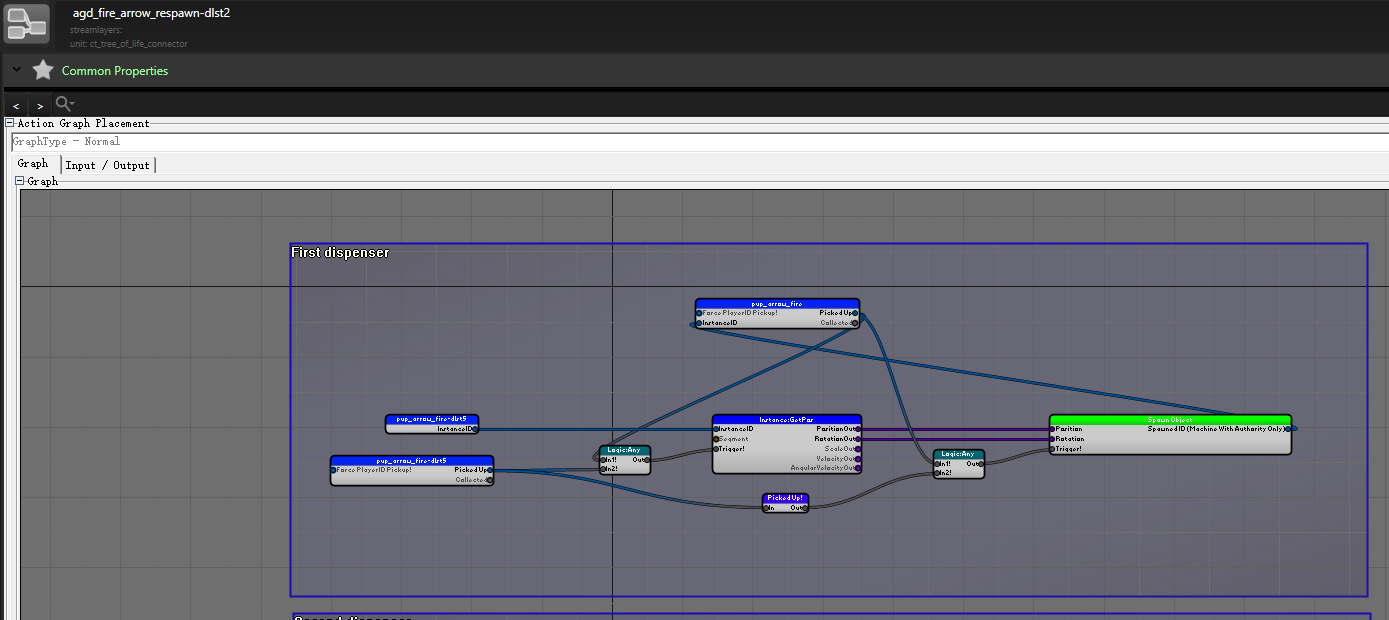
### Object

Object is the basic element of our level. It is more like a group of different things, a template. We can also add components to object to implement some pre-made functions. ***How to create or use an object will be the key point to master this engine.***



### Action Graph

Action Graph is a visual script tool in TR11 engine. It achieves functions for units and objects and defines relations between them.



## 2.3 Important Knowledge

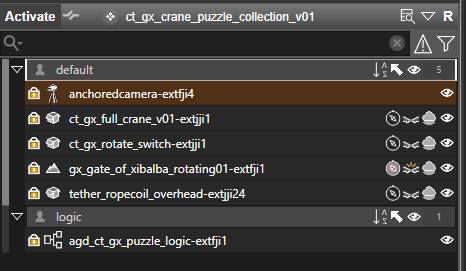
### Slices

Slice is like a folder in objects. We can create different slices in order to manage the objects well. Also, different users can checkout different slices. That enables us to edit same object in same time.

For designer, we should better put functional things in designer slices. So artists will not affect our works.

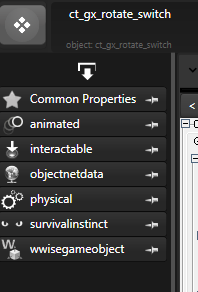
Tips

* Always name your slice clear.
* For designer, we should better put functional things in designer slices. So artists will not affect our works.



### Components

Components are pre-set modules for objects. Most of them are used to achieve specific function for objects. It is worth mentioned that Netdata is also one of the component, usually we treat it same as Input / Output because they’re more similar.



Most common and useful components are

***Physics*** Control object’s physics attribute.

***Objectnetdata*** Store net data for object.

***Intractable*** Set interact profile, like interact point position, animation and so on.

***Dynamicmarkup*** Set ropes for gameplay, you can control rope length, material etc.

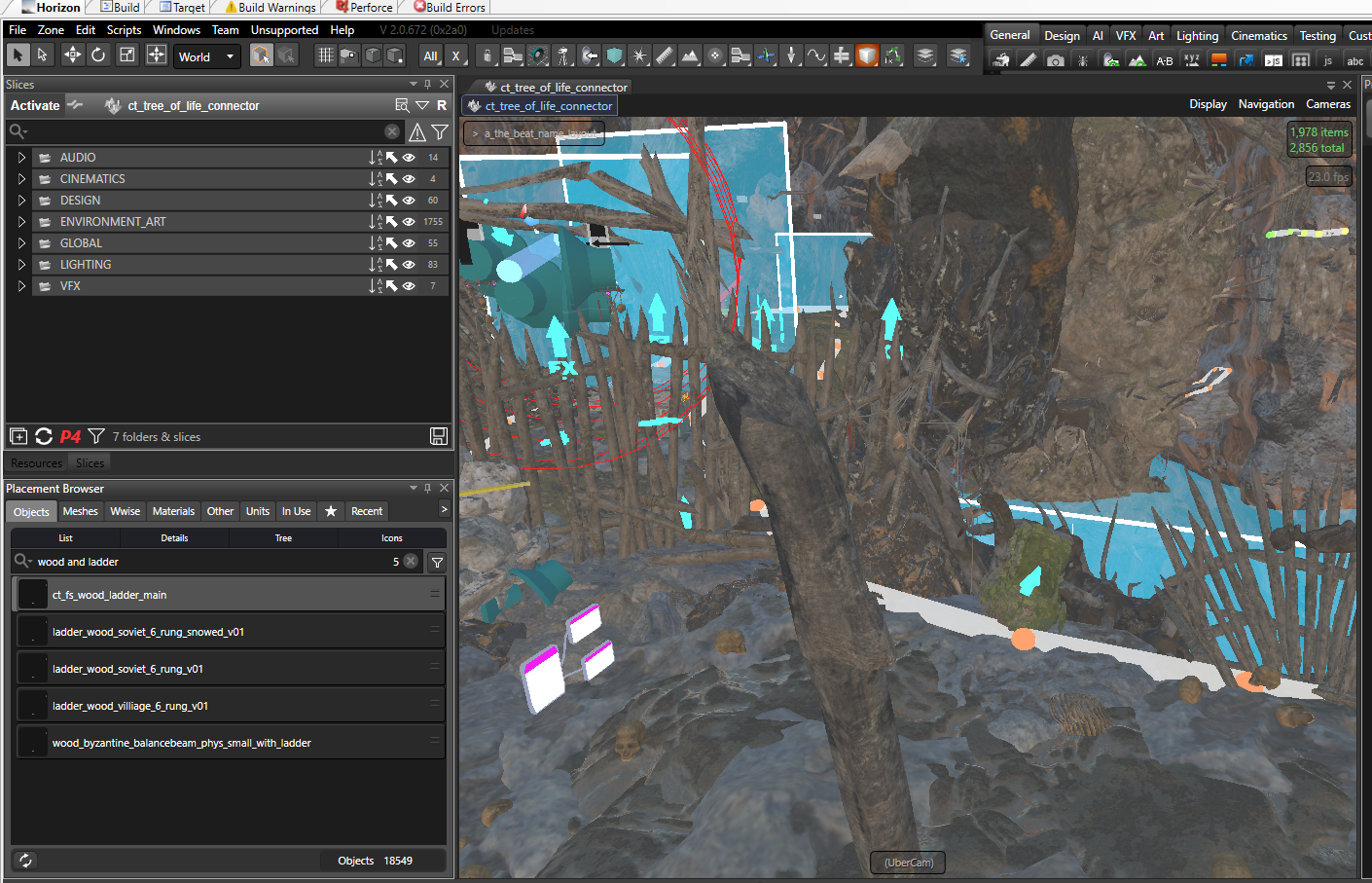
***Survivalinstinct*** Set SI model properties.

# 3 Objects Topics

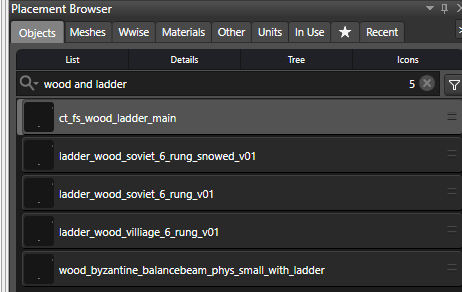
## 3.1 How can I find an object I want?

By two ways, the first one is play through current level and see if there is any object you want.

The second way is search by name in placement browser.



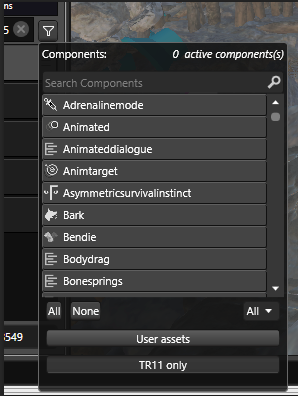
Check placement browser and insert the word you want to search. For example, I want a wooden ladder. So I just search “wood and ladder”.



You can see the result and check if these objects are what you want.

Tips

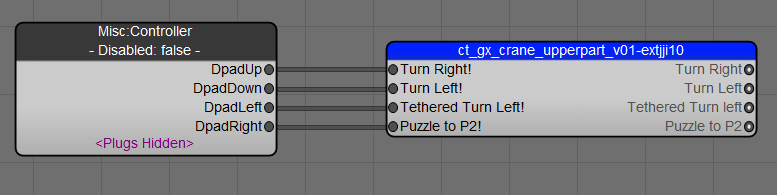
* You can use filter to do some advance searching.
* If you can’t find some object, try switch ***User assets*** and ***TR11 only***



Filter

## 3.2 How do I know how this object works?

You can place this object into a test level and use ***misc\_controller*** node to trigger each object input.



Then you can trigger this object by using controller and observe in the level, see what will happened.

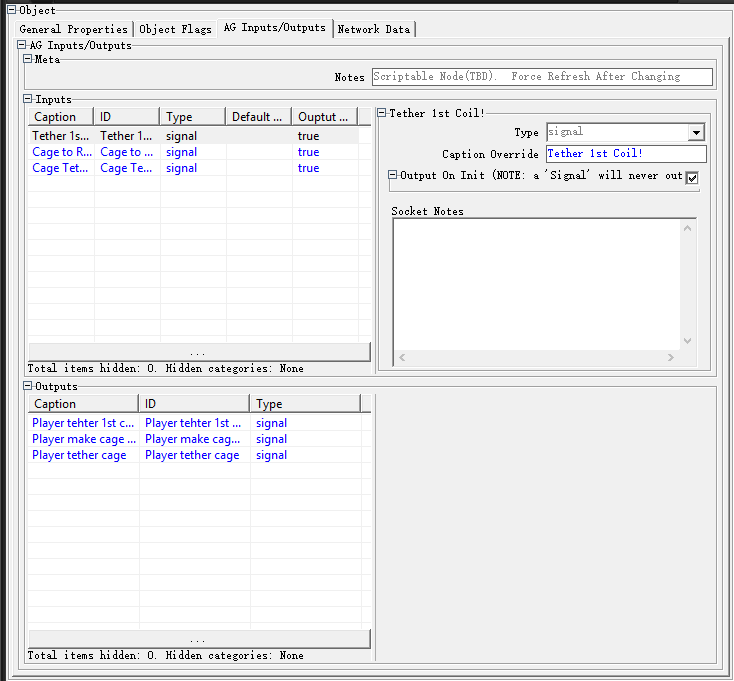
You can also play the level which contains this object and see how it works.

Tips

* Most of the time, you can learn this object through its input names. Like the one I put above, you can image, this object will be able to turn left and right.
* Some object may not have any feedback. Then you need to open the action graph to see how it works exactly.

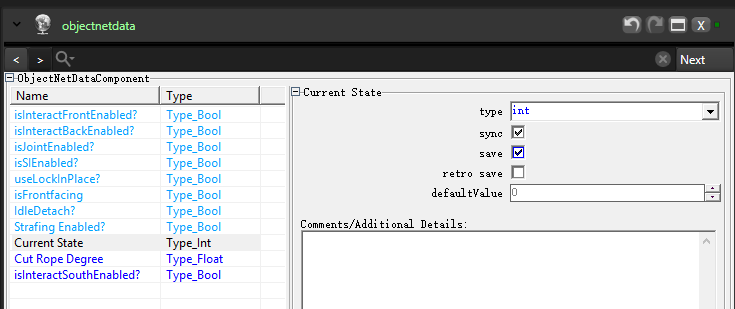
## 3.3 How can I use an exist object?

Three point you should notice. Input, Output and Netdata.



Inputs are usually triggers of an object. They can be thought same as the function in programming. We can send data to the input and make the object do the certain things through its action graphs.

Outputs. They can be thought as the return value in programming. Some inputs or events (e.g. interactions) will give back a return value, it will send to you through the output.



Netdata is used to store parameters. Linked with action graph, Netdata also can control different state of objects through action graph. It can be used in initialize or restore the value or status.

Recommend steps

* Check this object in a test level as I mentioned before
* Put it in your level or object
* Link its inputs / outputs in your action graph
* Check or override Netdata to set its state
* Debug if necessary (Later we will talk about this part)

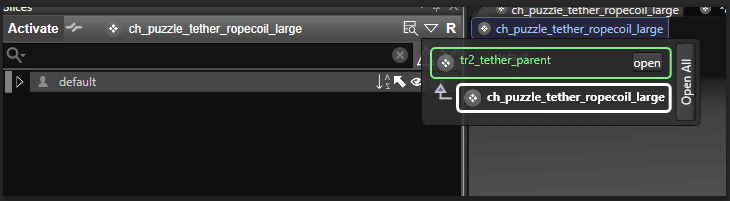
Tips

* Always remember, due to action graphs, you can override Netdata of an object which may help you set different states and parameters of an object.



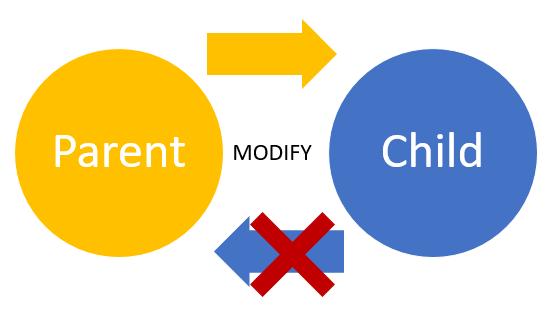
## 3.4 What is parent object or child object?

Object can set other objects as its parent.



Once the setting is done. Current object will become child object and the object you select will be the parent.

Child object will inheritance all things from its parent.



You can modify a child object, which will only affect this object. But if you edit the parent object, all the child objects will be changed.

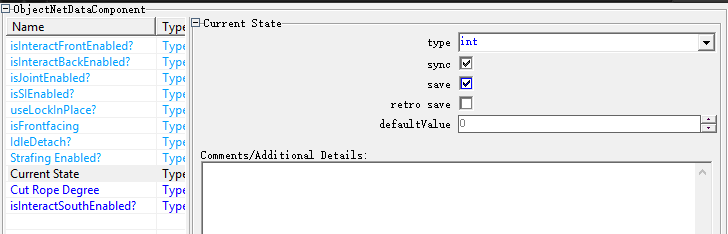
If you think well before start working, you will realize that this parent child system can save you a lot of time.

Tips

* Always keep an eye on edit a parent object. That may cause serious problems in other objects or even affect other levels.

## 3.5 How to store an object’s state?

Usually we can use object’s Netdata to do so.



Click “save” will make this Netdata saveable. It will save when player set a checkpoint.

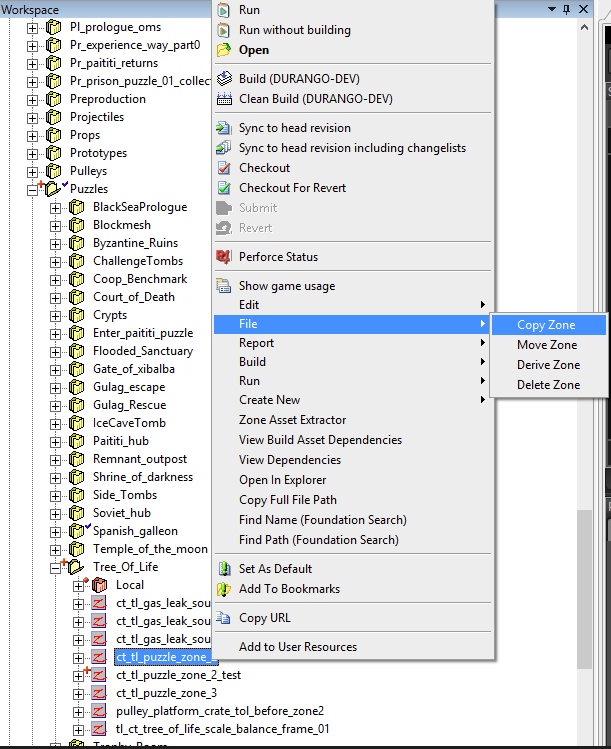
Click “retro save” is almost the same. But this Netdata will be save immediately once it is changed.

We can also use action graph to reset Netdata due to persistent data.

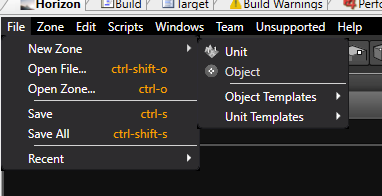
## 3.6 How to create my own object?

EDIOS has made a lot of objects, and most of them can work very well, just take and use it. But sometimes you still need to create some objects. You can do it by different ways.

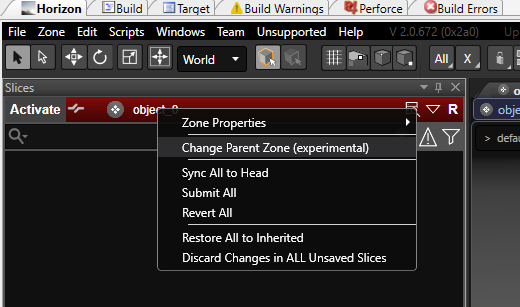
If you just want to edit a current object and use it in other places. You can copy and rename it, then edit it and submit.

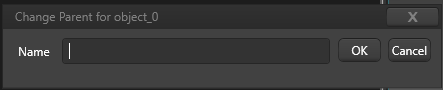


If you need to create a brand new object, you can use the template in engine to help you.

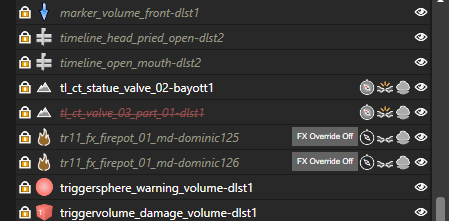


And if you still need to use a exist object as a reference to save some work, you can just select it as parent.





Select change parent zone and then type the name of parent object you want to select.



3

2

1

You can see 3 different states in this picture. This will only happen in child objects. That means,

1. Inherited from parent object
2. Inherited from parent but current deleted
3. Add only for child object

Also, you can add different components to achieve other functions for your object.

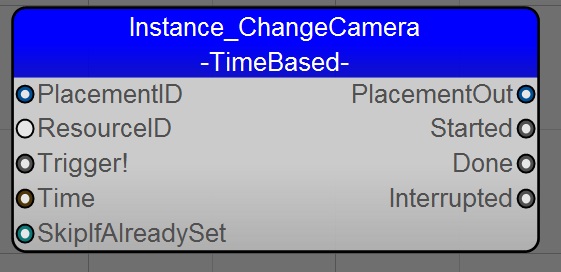
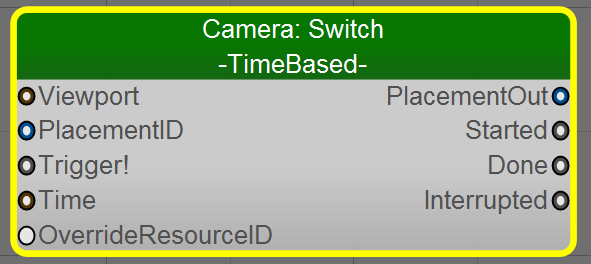
# 4 Action Graph Topics

## 4.1 How can I modify cameras through action graphs?

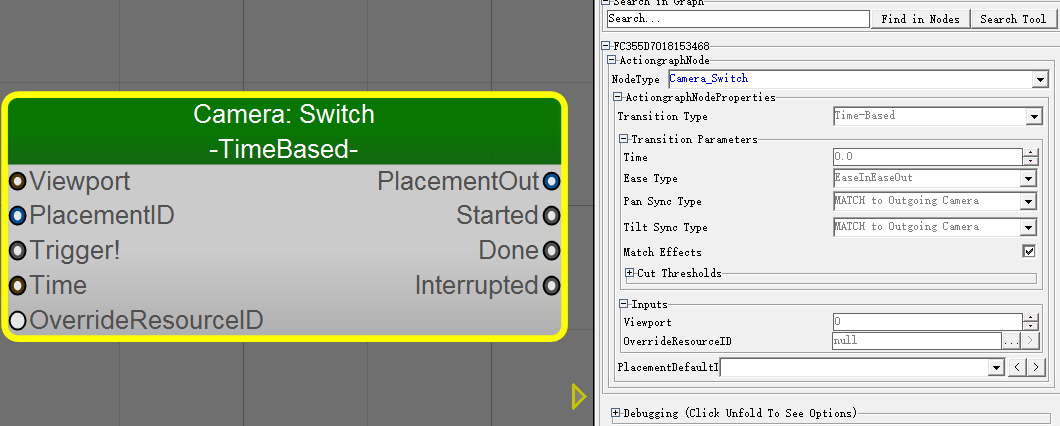
You can find detail camera tutorial at [\\cnshasdfsv1\Common\Projects\Tomb Raider\Design\01\_LEVEL\_DESIGN\_TRAINING\_Stuff\camera\_for\_traversal.pptx](file:///\\cnshasdfsv1\Common\Projects\Tomb%20Raider\Design\01_LEVEL_DESIGN_TRAINING_Stuff\camera_for_traversal.pptx) (written by Ji Feng)

Here are some advanced knowledge.

There are two type of camera nodes in this engine.



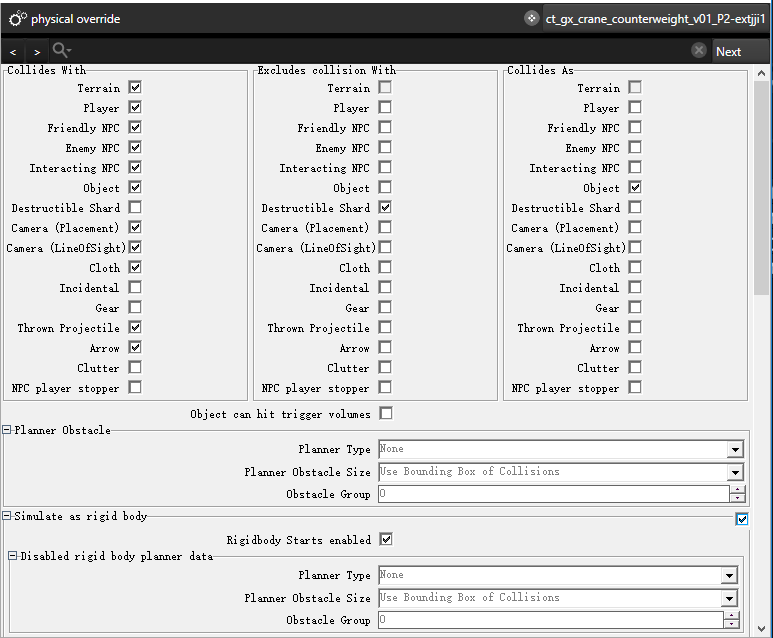
The basic functions are almost the same. You can change Pan, Tilt of player’s camera. But you should always remember, after you change your camera, you should always reset it to avoid bugs. And once you use one type, you should reset camera through the same type of node.



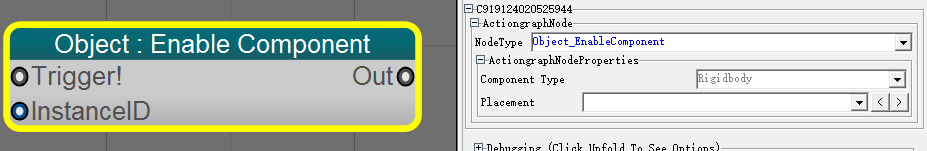
Here you can also select override camera resource ID for this node. There are a lot of pre-set camera profiles. You can even create your own one.

## 4.2 What are the basic physics node I can use?

Physics system in TR11 is very useful but also very complex. For new designer, I suggest that you should first learn about rigid body and simple joint movement.



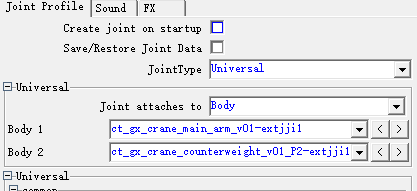
You should enable physics rigid body first in object component.



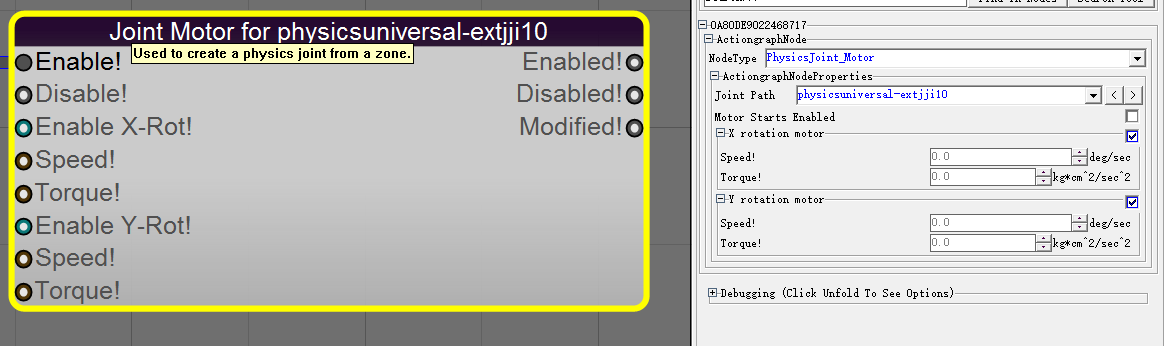
Then, you use ***Object\_EnableComponent*** to enable/disable this object’s rigid body.



For joint, you can find them here. There are different joints, you can check them later by yourself.



After create a joint, remember to link it with objects.



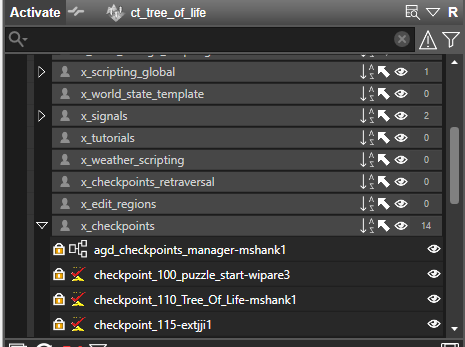
Then you can use ***Physicsjoint\_Motor*** to control it. You can set joint’s speed and torque.

Tips

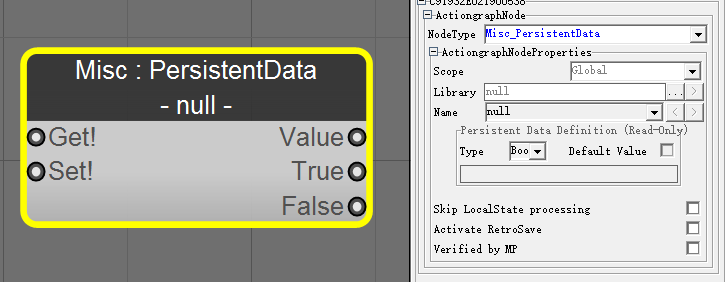
* A joint can only be put inside an object.
* Always use only one ***Physicsjoint\_Motor*** for each joint, otherwise it’s hard to debug and you may cause bugs.
* Several joints may cause bugs, always plan your joints wisely.
* Joint’s limits are not absolute. If the object receive too big physical force, the joint’s limits will break. That’s also why some ridiculous bugs occurs.
* Object’s center of gravity, mass will also affect joint, use your knowledge of physics wisely.
* If you want to give birth to a physics object, always remember, first enable its rigid body, and then create the joint. Otherwise, it will directly fall to ground.
* If you want a joint to keep still, you should set its speed to 0.

## 4.3 How can I modify persistent data through action graph?

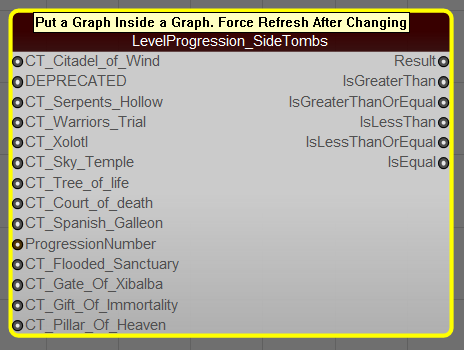
Persistent data is always linked with checkpoints in you levels. Sometimes it’s also used to restore some objects’ state.



Usually we can find this slice under ***Global*** slice in our level. There is a reference inside the action graph. Or you can open a level to check. (I suggest to open GOX)



We could use this node to get/set persistent data. You need to choose the library and name first.



Or you can use this sub graph to get and compare it also.

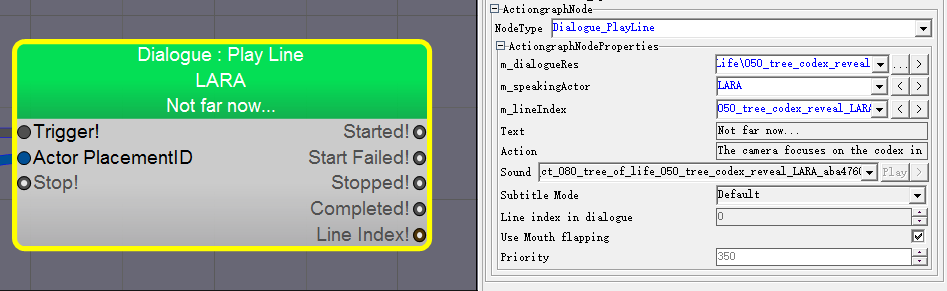
Remember, you should always set a checkpoint after persistent data is changed. Otherwise the change will not be saved.

Tips

* You should first arrange your persistent data well. Usually we set the start of connector to 010, the start of main room to 100. Add 10 each time the progression is changed.
* You should always delay 0.1 secs to set checkpoint after persistent data is changed. Otherwise the data may not be saved.

## 4.4 How can I add VO scripts for level?

Usually, VOs are done by Edios but we need to implement VO scripts first.



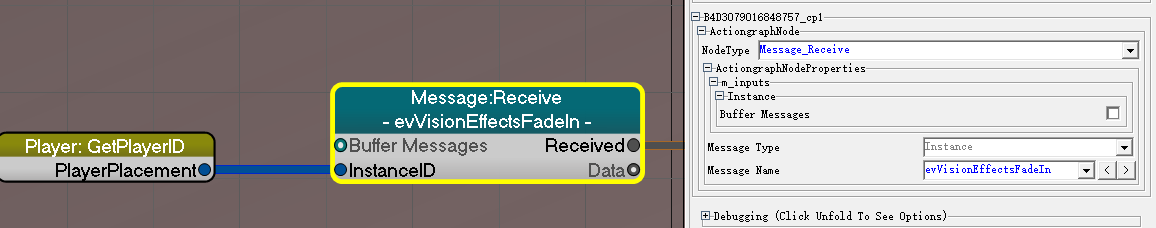
We use ***Dialogue\_Playline*** to play VOs.

VO trigger conditions are varied. Here are the most common ones.

### Lara arrive some place

Use trigger volume or planes to check if Lara get in position.

### Lara trigger SI model and seek for clues



Use ***Message\_Receive*** node to detect whether player has triggered SI model.

### Some specific event happens

Add an object/action graph output, then use this signal to trigger VO.

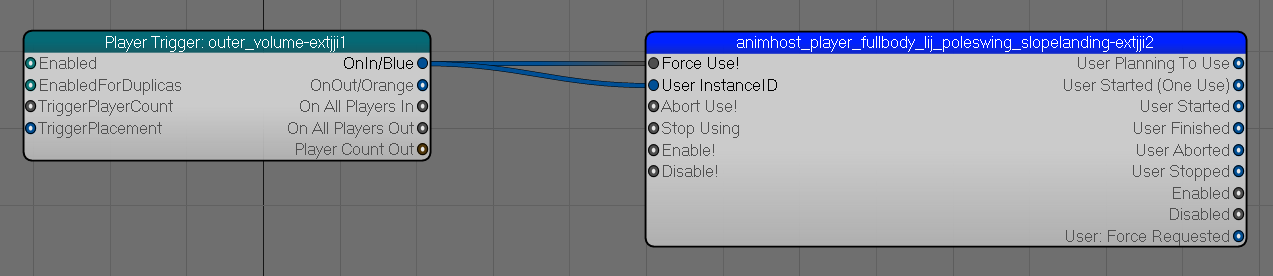
Tips

* Always remember to add ***Logic\_Once*** before VO triggers.
* Better put a ***Log\_Print*** at the same time VO triggers to debug.

## 4.5 How can I add an Animhost?

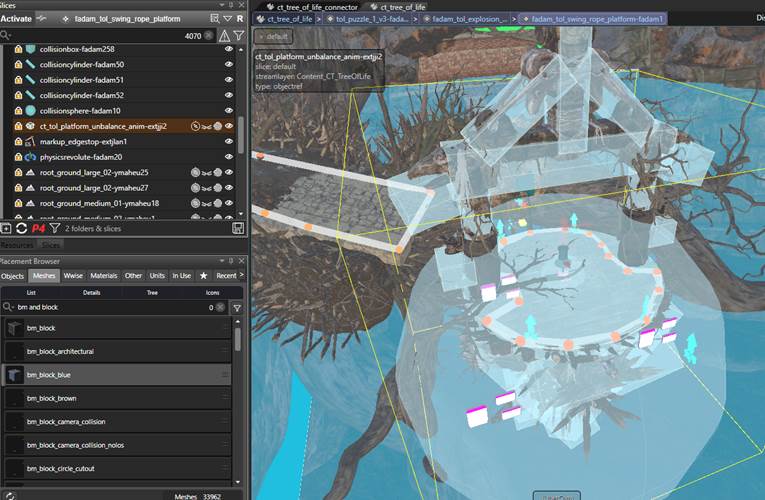
First you should find the animhost you want to add. You can check it in a test level first.

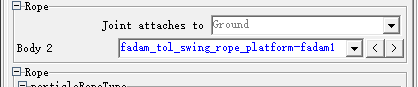
Then you can put this animhost in the level or object you want. Use action graph to trigger it.



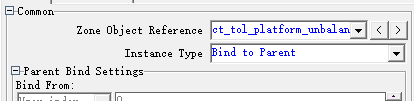
For example, here is a typical setting. Remember to send the user instance ID to Animhost Node. Otherwise it will not be triggered correctly.

Sometime we need to play an animhost on a moving platform. If we don’t set it correctly, you will see Lara is playing animation in original place and while platform is moving away. Here is the way about how to play an animhost on a moving platform.

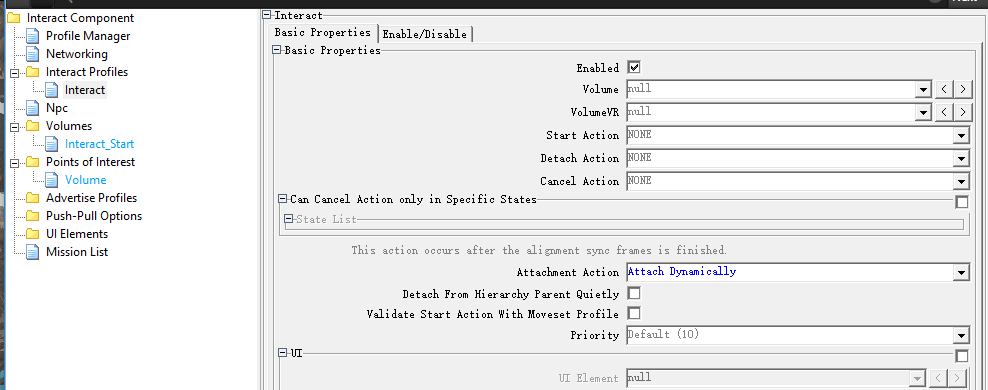




Put your animhost inside this platform object. This platform object should be the one which directly connected with joint.



Set it to ***bind to parent.***



Open the animhost object, check interactable component, and change this setting to ***Attach Dynamically.***

# 5 Debug Topics

Working in progress…..

But remember, CDC option is your most powerful weapon!

Tips

* No need to know the whole things in the level, just try to find the key point
* Use breakpoint and print to help you find what goes wrong
* Notice the sequence of triggering the nodes, especially when the node need additional parameters
* Know what the target object can do, it doesn’t mean you need to know how it achieves
* If you need to modify an object which has been in a solid logic, pls do not modify the input and output as far as possible
* If you try to replace an object in a solid logic, pls keep the input and output as same with previous as possible
* If you want to modify an object in another object, don’t be lazy, pls make a input to do that things
* Don’t just think, do more unit test, then more you will know

# 6 Various Tips

Here are a collection of tutorials we made along the development. They are pretty specify and detail. Have a look if you encounter a similar issue.

This list will continue to update.

1. [How to display 'Tomb Completed' after grabbing codex](Various_tips/How%20to%20display%20'Tomb%20Completed'%20after%20grabbing%20codex.docx)
2. [How to do ragdolls in water.](Various_tips/How%20to%20do%20ragdolls%20in%20water.docx)
3. [How to save physics joint state](Various_tips/How%20to%20save%20physics%20joint%20state.docx)
4. [How to split levels & set signal.](Various_tips/How%20to%20split%20levels%20&%20set%20signal.docx)
5. [How to do Cells&Portal.](Various_tips/Cells&Portal.pdf)
6. [How to trigger a VO when the Survival Instinct is triggered by the player.](Various_tips/How%20to%20trigger%20a%20VO%20when%20the%20Survival%20Instinct%20is%20triggered%20by%20the%20player.docx)
7. [How to work offline when p4 is down(P4离线工作流程)](Various_tips/How%20to%20work%20offline%20when%20p4%20is%20down(P4离线工作流程).docx)
8. [How to do traversal camera.](Various_tips/camera_for_traversal.pptx)
9. [How to Integrate Birds and Animals](Various_tips/LivingHistory_IntegratingBirdsandAnimals_V3.DOC)
10. [Lara求生本能下设置特定Mesh发光&[Tutor]Mesh light in lara's survival](Various_tips/Lara求生本能下设置特定Mesh发光&%5bTutor%5dMesh%20light%20in%20lara's%20survival%20.docx)
11. [Level building tips](Various_tips/Level_building_tips_class.pptx)
12. [Deep Water Piranhas Guide](Various_tips/Deep%20Water%20Piranhas%20Guide%20-%20Game%20Design%20Portal%20-%20Confluence%20-%20Tomb%20Raider%2011.pdf)
13. [Navigation Hints – White marks](Various_tips/Navigation_Hints-Whitemarks.docx)
14. [Shortcuts and cheats](Various_tips/PP-Shortcutsandcheats-020517-0936-2.pdf)
15. [Timeline Tips from David](Various_tips/Timeline%20Tips%20from%20David.docx)
16. [TR11\_Wind\_Logic\_Tutorial](Various_tips/TR11_Wind_Logic_Tutorial_JiJiabin.pptx)
17. [(ART) TRAVERSAL\_AXE\_CLIMB\_GUIDE](Various_tips/TRAVERSAL_AXE_CLIMB_GUIDE.pdf)
18. [Puzzle values (trx)](Various_tips/trx_puzzle_values.docx)
19. [Interesting test & lessons regarding to animals](Various_tips/TRXI%20Systems%20Test%20Animals%2020171013.docx)
20. [(ART) 如何把场景中多个模型优化成一个整体,并且制作LOD](Various_tips/如何把场景中多个模型优化成一个整体,并且制作LOD.DOCX)
21. [Signs and feedbacks](Various_tips/Signs%20and%20Feedbacks_CC.PPTX)