UNREAL ENGINE

# Blueprint Notes 1

Steven Harris

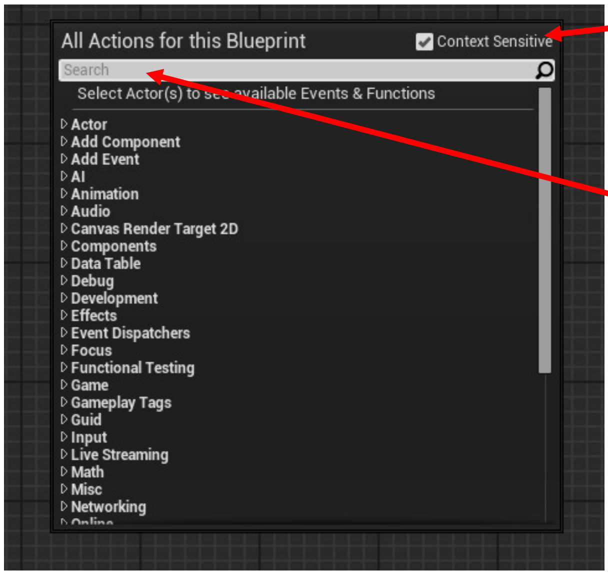
# TABLE OF CONTENTS

# BLUEPRINT

## 1. FINDING BLUEPRINT COMMANDS

### 1.1 RIGHT CLICK ON BLUEPRINT WORKSPACE

Right click on the Blueprint workspace to display list of Blueprint commands, as below:



**Contest Sensitive**

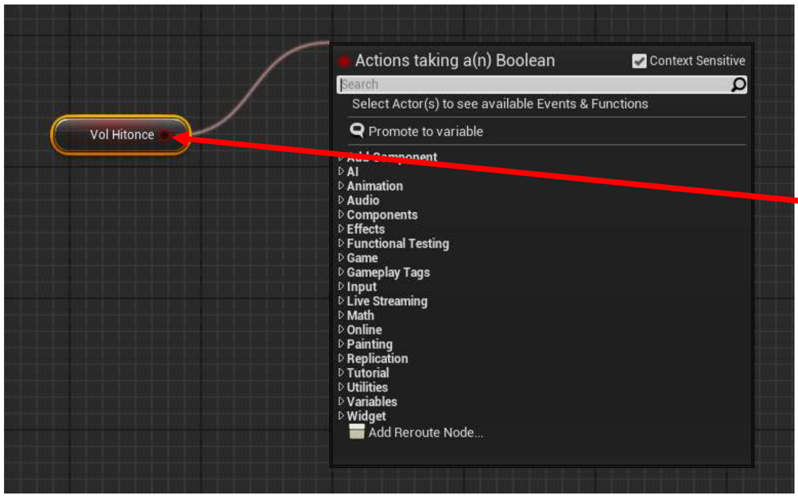
Tick this box to display only commands which are useful in the current type of blueprint.

**Search**

Type in here to filter commands based on the search text

### 1.2 DRAG OUT FROM OUTPUT NODE

From any node on an existing Blueprint sequence, click and drag on the node to display the search box, as below:



Drag with left mouse button from here

## 2. SELECTING AND MOVING NODES

Left mouse click = select one node.

Right click mouse = open node menu.

Hold CTRL + left click = manually select multiple nodes while CTRL is held down.

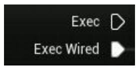
SHIFT + click & drag = creates a marquee selection area which selects nodes.

CTRL + click & drag = creates a marquee selection area which de-selects nodes which are currently selected.

Left click & drag = move currently selected nodes.

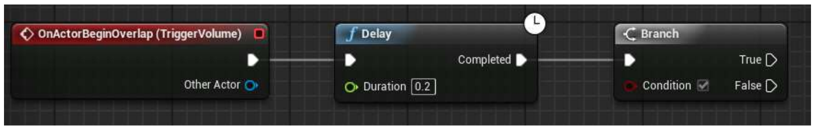
## 3. EXECUTION PINS

Execution pins control the order in which commands are executed. Connect execution pins together to create an ordered sequence of commands. White bordered execution pins are currently inactive. When an execution pin is connected to another valid execution pin, it turns white, see below.

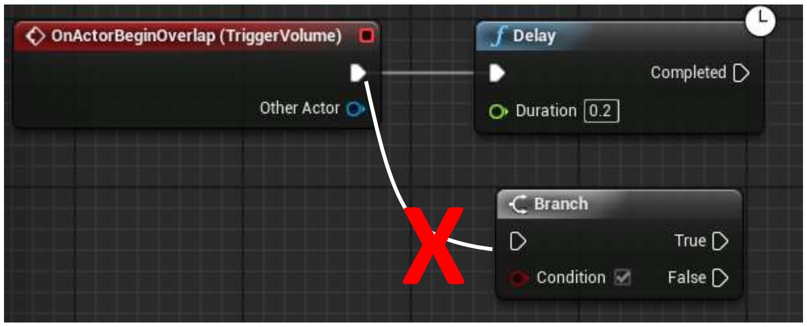


## 4. SEQUENTIAL SCRIPTING OF ACTIONS

Blueprint only allows *sequential* connection of execution nodes. *Parallel* connections were possible in UE3/UDK but UE4 no longer allows this.



ALLOWED

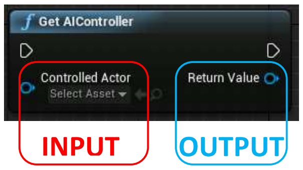


NOT ALLOWED

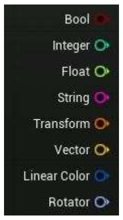
Parallel sequences create issues in the timing of events. In a parallel configuration you can never be sure which order events will be executed in the two parallel sequences, as only one event can ever be executed at a time, even if it looks like they are in parallel sequences.

## 5. BLUEPRINT PINS

Pins allow data to be connected to, and output from, individual nodes. Pins on the LEFT of the node are for INPUT. Nodes on the RIGHT are for OUTPUT.

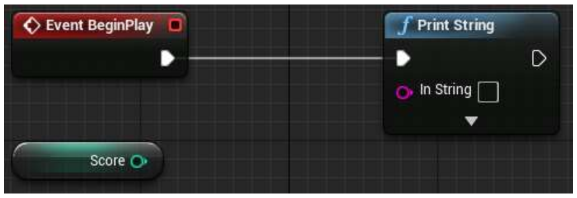


Pins are colour coded, depending on their data type, see below. There are other colours which represent data classes, but the list below shows the most common types. Pins become filled with their colour when they are connected to a variable.

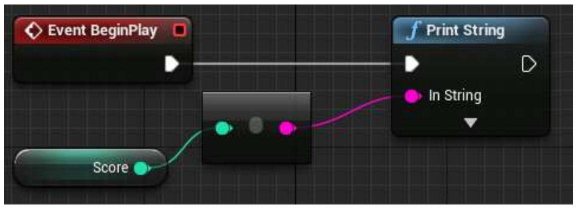


## 6. AUTO CASTING

It is possible to connect pins of different types/colours. This is only possible between certain data types, and Unreal Engine 4 will automatically convert the data if possible. This conversion process is called ‘casting a variable’. In the example below, the score variable is an integer, but we want to connect it to the PRINT STRING node.

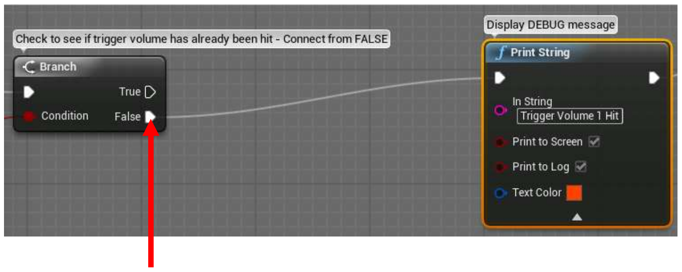


If you drag the pin from the integer SCORE to the string pin IN STRING, an extra node appears in order to CAST the integer to a string. If casting is possible Unreal Engine 4 will apply it automatically.

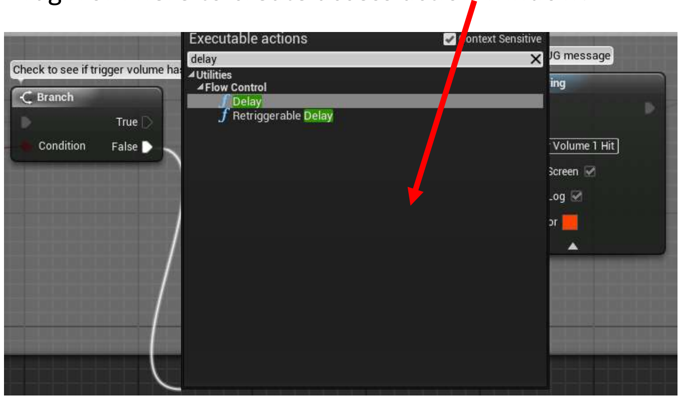


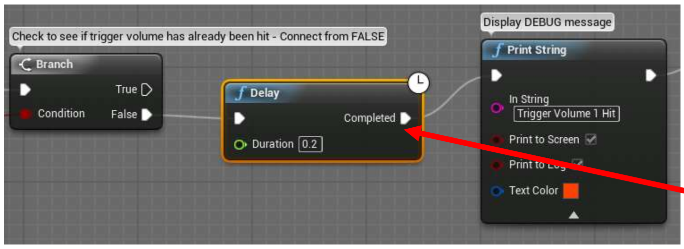
## 7. INSERT A COMMAND INTO AN EXISTING FLOW

To insert a new command into an existing flow, there is no need to disconnect the two sequences, simply click and drag from the first node, select the desired sequence and it will be inserted into the flow. This is another way in which UE4 aids sequential flow.



Drag from here to create access action window.

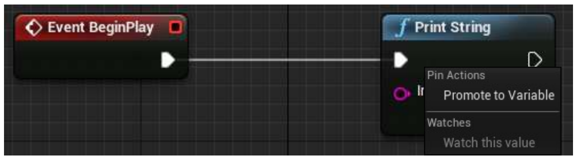




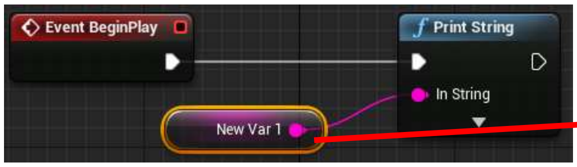
New action inserted along existing line

## 8. PROMOTE TO VARIABLE

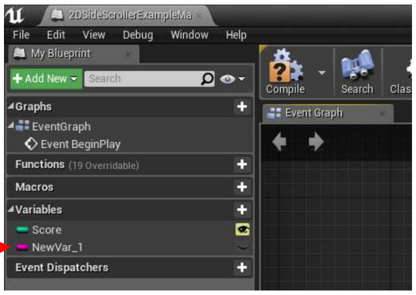
You can create a variable from a data pin automatically. Right click the pin and select PROMOTE TO VARIABLE, see below.



This creates a new variable, which will be automatically attached to the pin. The variable will appear in the VARIABLES section of the MY BLUEPRINT tab, see below.



New variable created from pin

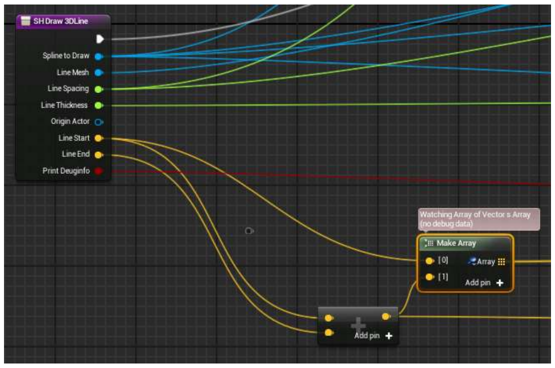


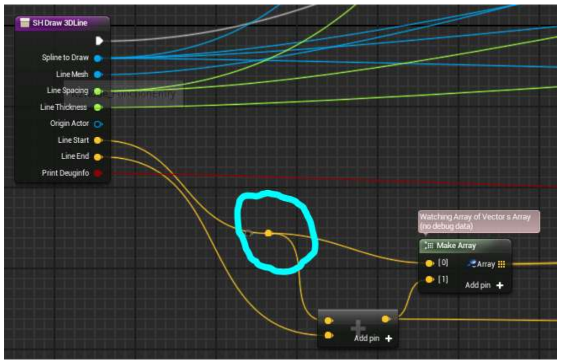
Variable created in VARIABLES section

## 9. REROUTE NODE

This ability is good for minimising multiple data connection coming off from one node.

To create a reroute node, click-drag off a pin which is already connected to another node, then release the left mouse button. A list of nodes will appear. At the bottom of this list is the ADD REROUTE option. Select this option to create a new pin which can be used to split the connection from the source pin, see below.



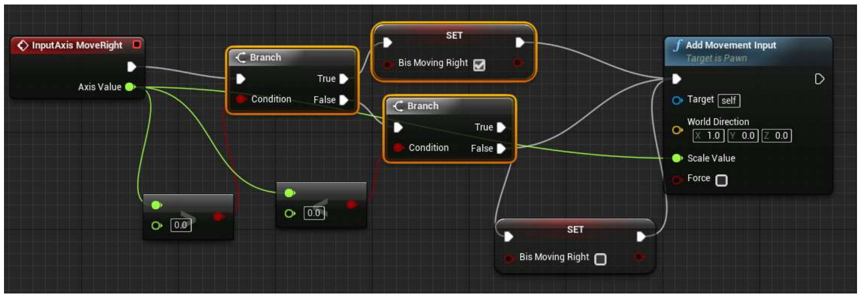


Move the Reroute node = CTRL + click drag

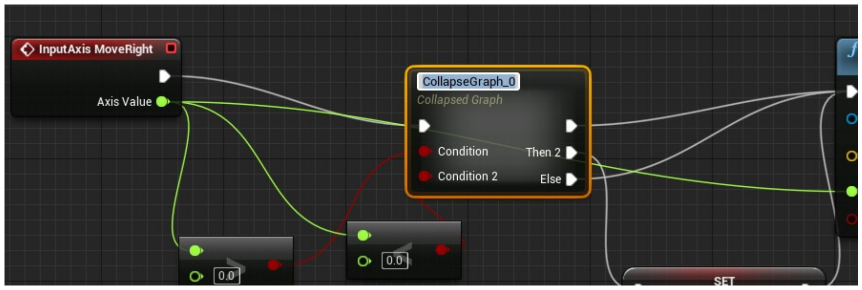
Remove the Reroute node = ALT + click

## 10. COLLAPSING NODES

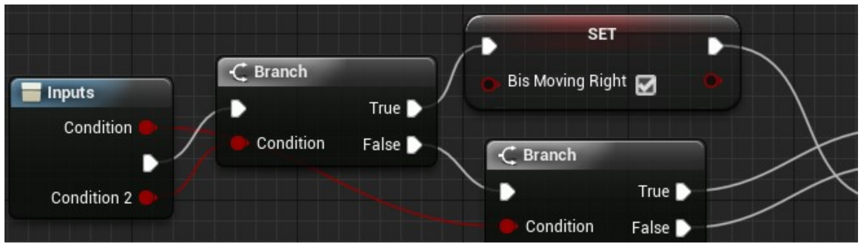
Complex node sequences can be cleaned up by collapsing them into one condensed node, to make the code easier to read. In the example below, three nodes have been selected (yellow border). Right click on the selected nodes and select COLLAPSE NODES.



This creates on condensed node, which you can rename to help describe the contents. This can be expanded back to the original multiple nodes, by selecting the condensed node, right click and select EXAPND NODE.



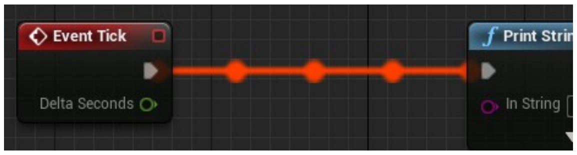
Double clicking on the collapsed node will open a new tab at the top of the editor and show you the contents of the collapsed node, see below.



## 11. LIVE DEBUGGING

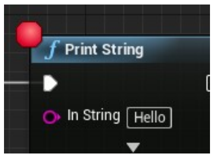
### 11.1 LIVE FLOW

The flow of logic through the execution wires can be visualised live, during runtime. Place the Blueprint window so that it is visible while you run the game. The flow down the execution wires will be shown by orange pulses travelling down the wires. This is very helpful for visualising the game logic during runtime.



## 12. BREAKPOINTS

Breakpoints can be used to stop the flow of through the Blueprint at runtime, and still keep the game running so that you can check the contents of variables etc. To add a checkpoint, right click on a node and select ADD BREAKPOINT. This adds a red circle to the node indicating a breakpoint is active.



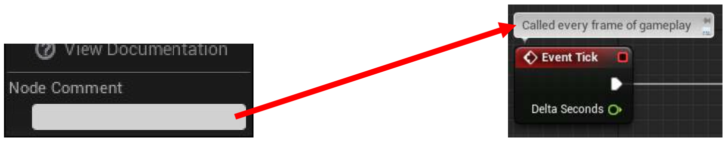
The DEBUG window should also be enabled so you can view the contents of variables at runtime. Select WINDOW | DEBUG to make this visible. With a breakpoint inserted, run the game and when the code reached the node with the breakpoint, the game will stop, and you can view the contents of the debug window.

Breakpoints can be enabled and disabled, using the right click menu on the node.

## 13. COMMENTS

### 13.1 NODE COMMENTS

Comments can be added to individual nodes. Right click on a node, at the bottom of the menu, add your comment in the NODE COMMENT box, see below.

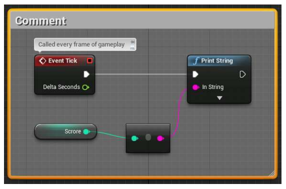


Right click on node, add comment here Comment displayed on node

To remove a node comment, right click on the node and delete the comment from the node comment box in the menu.

### 13.2 COMMENT BOX

A group of nodes can have a comment box placed around them. Select the nodes and press the C key to add the comment box.

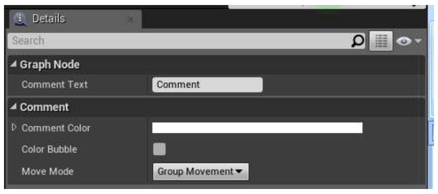


If you click and drag the header of the comment, this moves the box and its contents. To move only the comment box, and leave the contents behind, hold SHIFT and drag the comment header.

To delete a comment box, select the comment box and press delete.

#### 13.2.1 COMMENT BOX DETAILS

Each comment box can be customised by changing its properties. Select the comment box and change these in the comment tab.



Graph Node
Comment Text: Comment
Comment
Comment Color
Color Bubble
Move Mode: Group Movement