# Bolt Enhanced

Boost up your game development with Bolt

*If you have any issues or questions, please drop us a message*

- **Twitter**: https://twitter.com/yolostudiogame
- **Email**: contact@yolostudio.io
- **Website**: http://yolostudio.io
- **Discord**: https://discord.com/invite/gas3BMCRhE
- **Preview Feature playlist on Youtube**: https://www.youtube.com/playlist?list=PL--CMr7xgSMBgQppt6KoBNTp3yhu19un9

## Download

- Bolt Enhanced FREE
- Bolt Enhanced PRO

## Introduction

After working with Bolt for a long time, we're aware that there are some processes we can improve to save time and less pain when developing a Unity game using Bolt. This is the reason we develop the Bolt Enhanced asset.

Bolt Enhanced is a collection of utilities that help the creators boosting up the process when working with the Unity projects which use Bolt. There are several features included in this asset:

- **[PRO]** Bolt Generator
- Bolt Debug
- Bolt Statistic

- Bolt Pretty
- Bolt Variables
- Bolt Finder

Before going into further detail of all the features, let's dive into the Installation section to make Bolt Enhanced ready for use.

*Scroll down to the description section for more detail of each utility.*

## Change Logs

Version 1.3

- Added more features to Bolt Finder
    - Support finding usage of Literal units and their value
    - Support finding usage of Get/Set Variable (Object, Graph, Scene, Application, Saved)
    - All the other units are searched by its type.
    - Fix bugs and improve search performance
    - The user simply select the unit he wants to find usage and clicks the **Find Usage** button inside the **Find Usage** window.

Version 1.0

- **PRO** Features
    - Bolt Generator
        - Support generic math operators
            - Add (+)
            - Subtract (-)
            - Multiply (*)
            - Divide (/)
            - Modulo (%)
            - **Example**
                - 1 + 2 - 3 * 4 / 5 % 6
        - Support bitwise operators
            - And (&)

        - Or (|)
        - Xor (^)
        - Shift Left (<<)
        - Shift Right. (>>)
        - **Example**
            - 1 & 2 | 3 ^ 4 << 5 >> 6
    - Support round brackets for the operand priority
        - **Example**
            - 1 + 2 * (3 - (4 * (5 - 2)))
    - Support boolean operators
        - And (&&)
        - Or (||)
        - Equal (==)
        - Less (<)
        - Less Equal (<=)
        - Greater (>)
        - Greater Equal (>=)
        - **Example**
            - (1 + 1) == 2 && 1 * 2 > 2 || 1 / 2 >= 2 && 1 * 2 < 2 || 1 / 2 <= 2
    - Support other math functions
        - Acos
        - Asin
        - Atan
        - Ceiling
        - Cos
        - Exp
        - Floor
        - IEEERemainder
        - Log
        - Log10
        - Max
        - Min
        - Pow
        - Round
        - Sign
        - Sin
        - Sqrt

            - Tan
            - Truncate
            - **Example**
                - Sin(10) + Cos(20) + Min(10, 20) + Max(20, 30)
        - Support **if (condition)** command
            - **Example**
                - if (1 + 2 == 2)
                - if (1 + 2 == 2, 3, 4)
        - Support ternary format
            - **Example**
                - 1 + 2 == 2 ? 3 : 4
        - Support **while(), while(condition)** commands
        - Support **for(), for (startIndex, endIndex), for(startIndex, endIndex, step)** commands.
        - Support all Bolt Variables
            - Graph Variables (graph)
            - Object Variables (object)
            - Scene Variables (scene)
            - App Variables (app)
            - Saved Variables (saved)
            - Syntax
                - [**type**.variableName]
            - **Example**
                - [**graph**.testGraphVariable]
                - [**object**.testGraphVariable]
        - Support generate all the above syntaxes into
            - Bolt Flow file
            - Bolt Flow file with a custom name
            - Copy generated flow graph into Clipboard
        - Support generate all the above syntaxes into Bolt Graph
- Features, the following features are already included in this Bolt Enhanced FREE version.
    - Bolt Debug
        - A separate console window for Bolt
        - Allow jumping to the specific node that causes the log and highlight it
        - Custom Debug Unit for Bolt
    - Bolt Statistic
        - Count the state machine in the current scene
        - Count the flow machine in the current scene

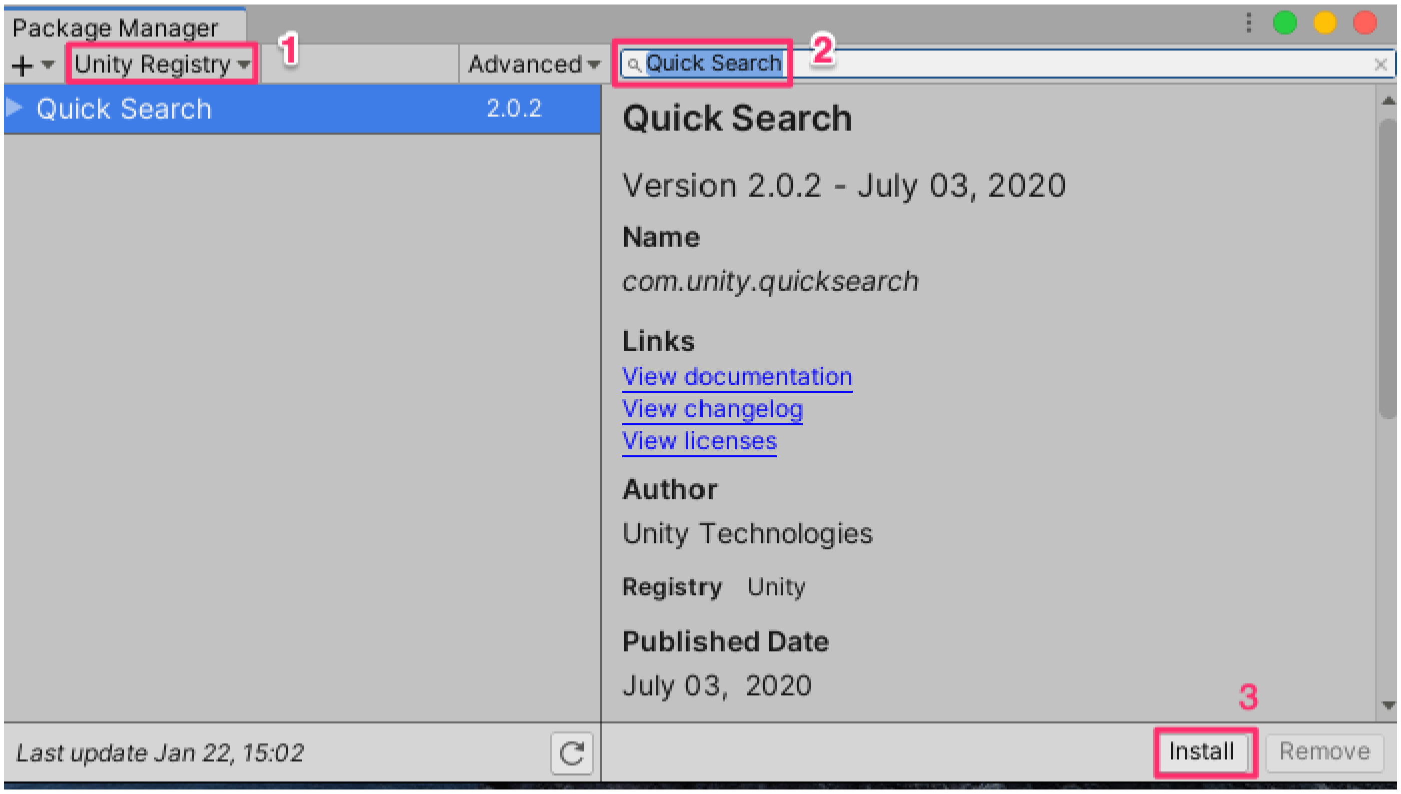
    - Bolt Pretty
        - Config snap (to the origin, to even number, to a number which is multiple of 10)
        - Config margin
    - Bolt Variables
        - Quickly delete all **Saved** Variable
        - Quickly delete all **App** Variable
        - Quickly delete all **Scene** Variable
    - Bolt Finder
        - Quick search bolt file in your project Find usage of (flow, state) macro in another macro (Multiple occur points)
        - Find usage of (flow, state) macro in the current scene (Embed, Macro, Multiple machines)
- Bugs Fixing
    - Fix the issue when building for the AOT platform.

## Installation

**Step 1.** Import and install Unity Bolt

**Step 2.** Install the **Unity Quick Search** asset:

From the Unity menu, select **Window -> Package Manager**



Quick Search

Version 2.0.2 - July 03, 2020

Name

*com.unity.quicksearch*

Links

View documentation

View changelog

View licenses

Author

Unity Technologies

Registry Unity

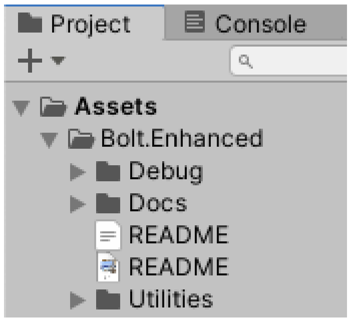
Published Date

July 03, 2020

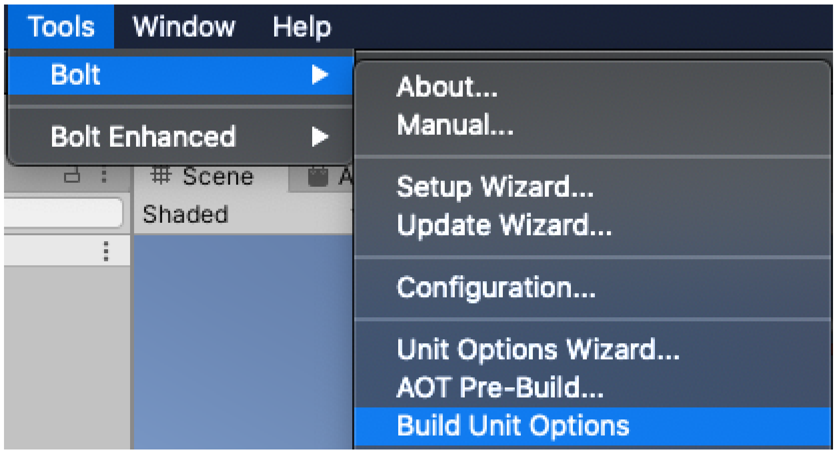
*Last update Jan 22, 15:02*

Select **Unity Registry** (1), search for **Quick Search** (2), click **Install** (3)

**Step 3.** Import the **Bolt Enhanced** package.



**Step 4.** Rebuild Bolt Unit Options. **Tools ➞ Bolt ➞ Build Unit Options**



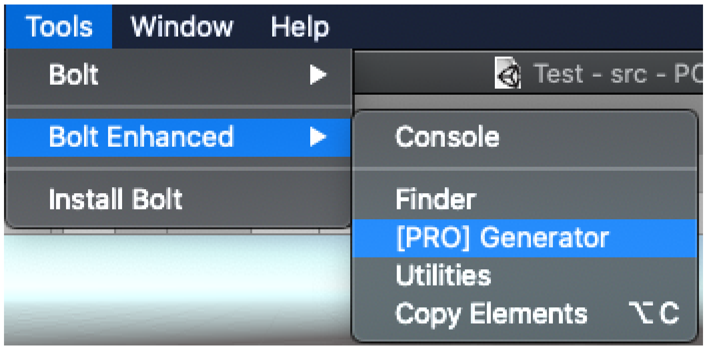
*We're ready to go!*

## [PRO] Bolt Generator

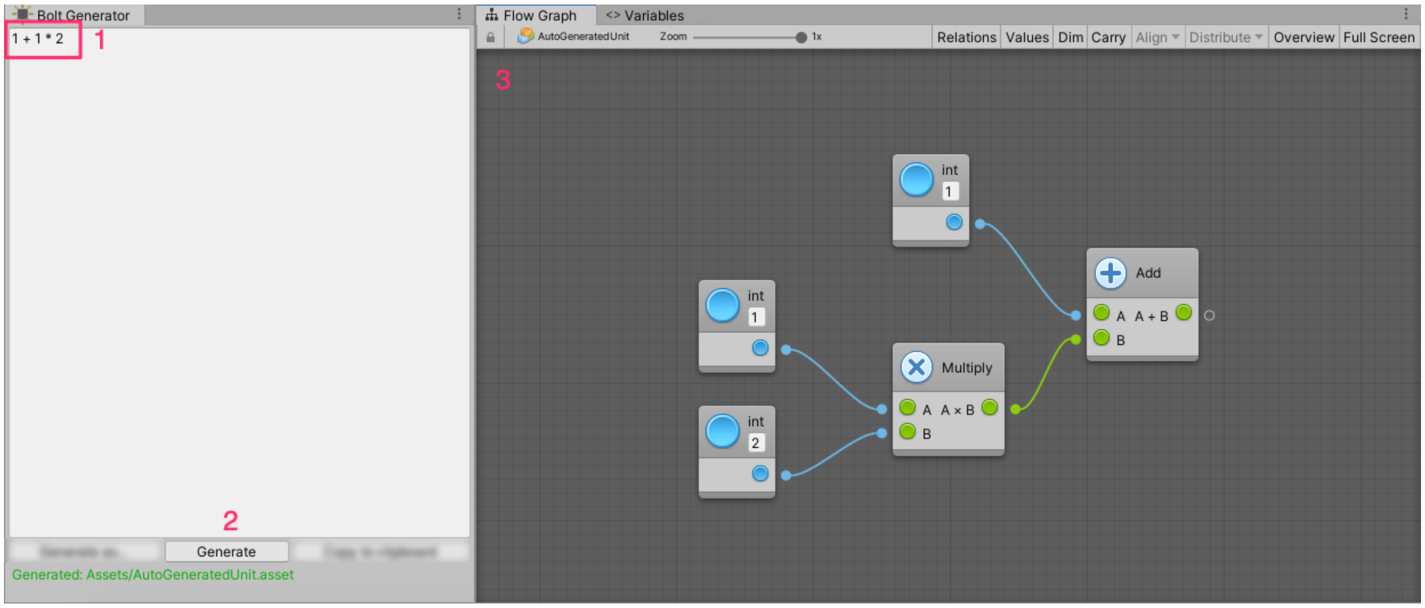
### Getting Started

**Step 1.** Open the Bolt Generator Window

From the **menu bar** of Unity Editor, **Tools ➞ Bolt Enhanced ➞ [PRO] Generator**

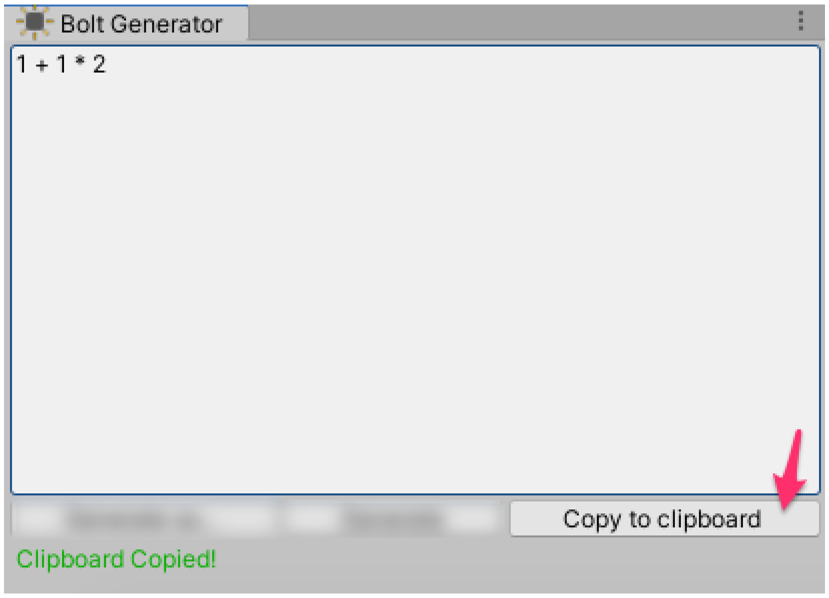


**Step 2.** **(1)** Type **1 + 1 * 2** inside Bolt Generator window, **(2)** Click Generate button, **(3)** Look at the result inside the **Graph window**.

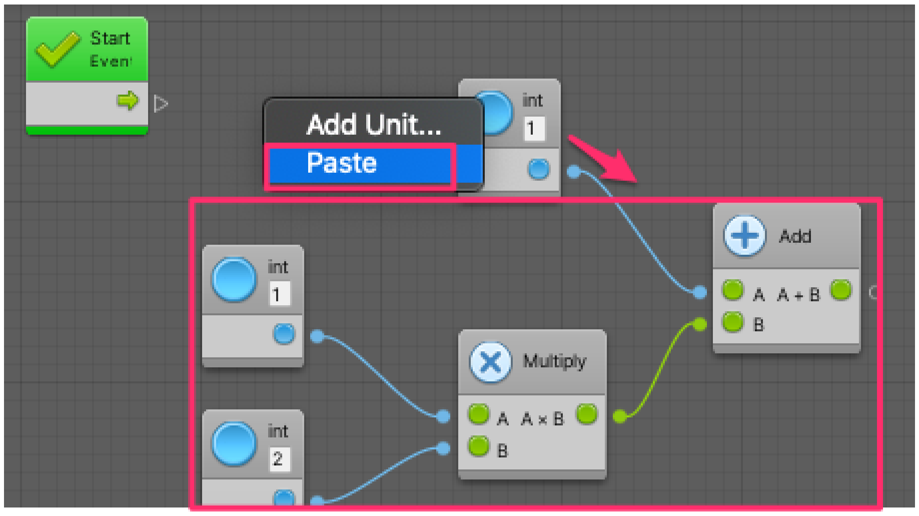


## Generate graph to the clipboard

**Step 1.** Click Copy to clipboard



**Step 2.** Select an existed graph and paste the generated graph here.



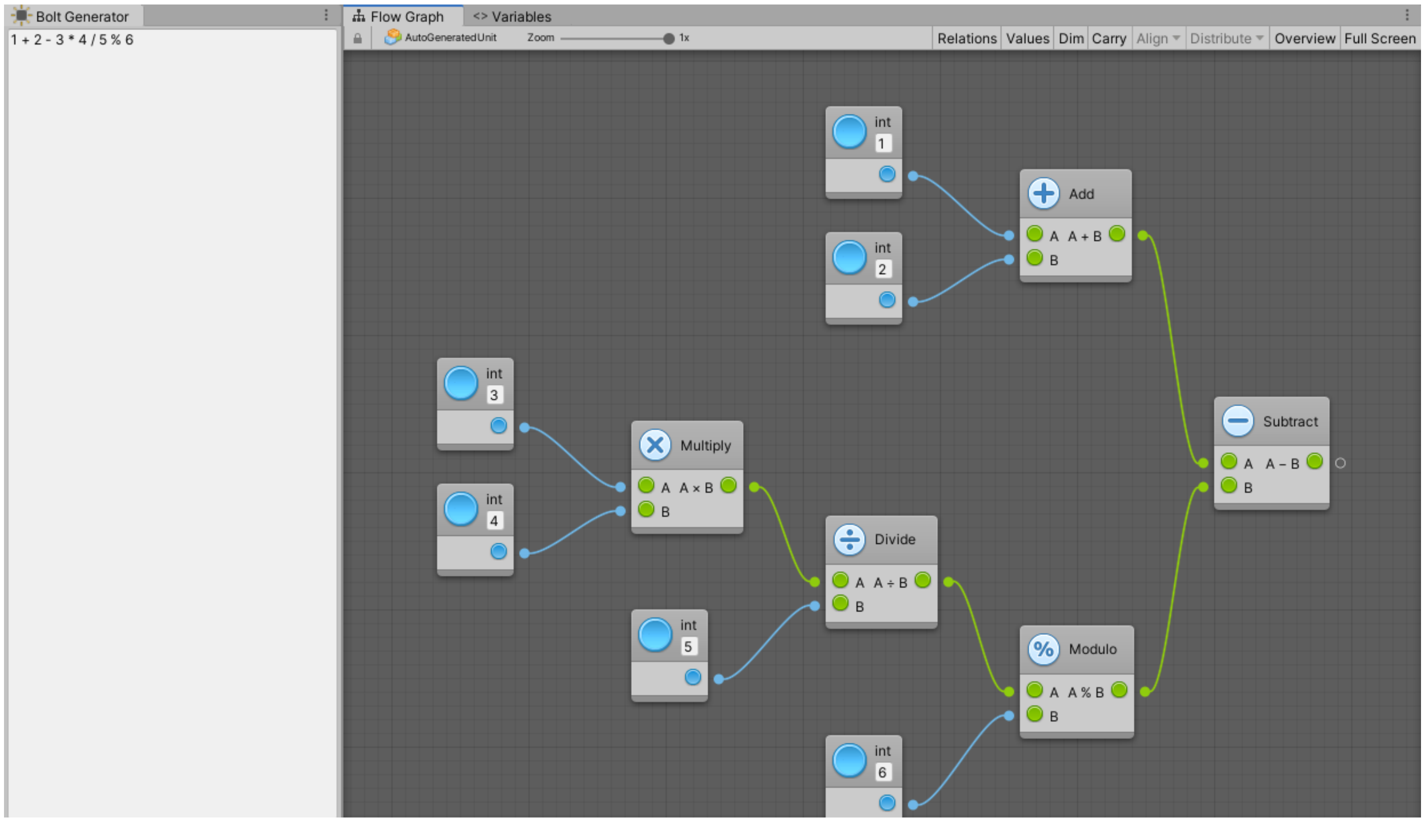
## Supported operators

### General math operators

- Add (+)
- Subtract (-)
- Multiply (*)
- Divide (/)
- Modulo (%)

Example 1

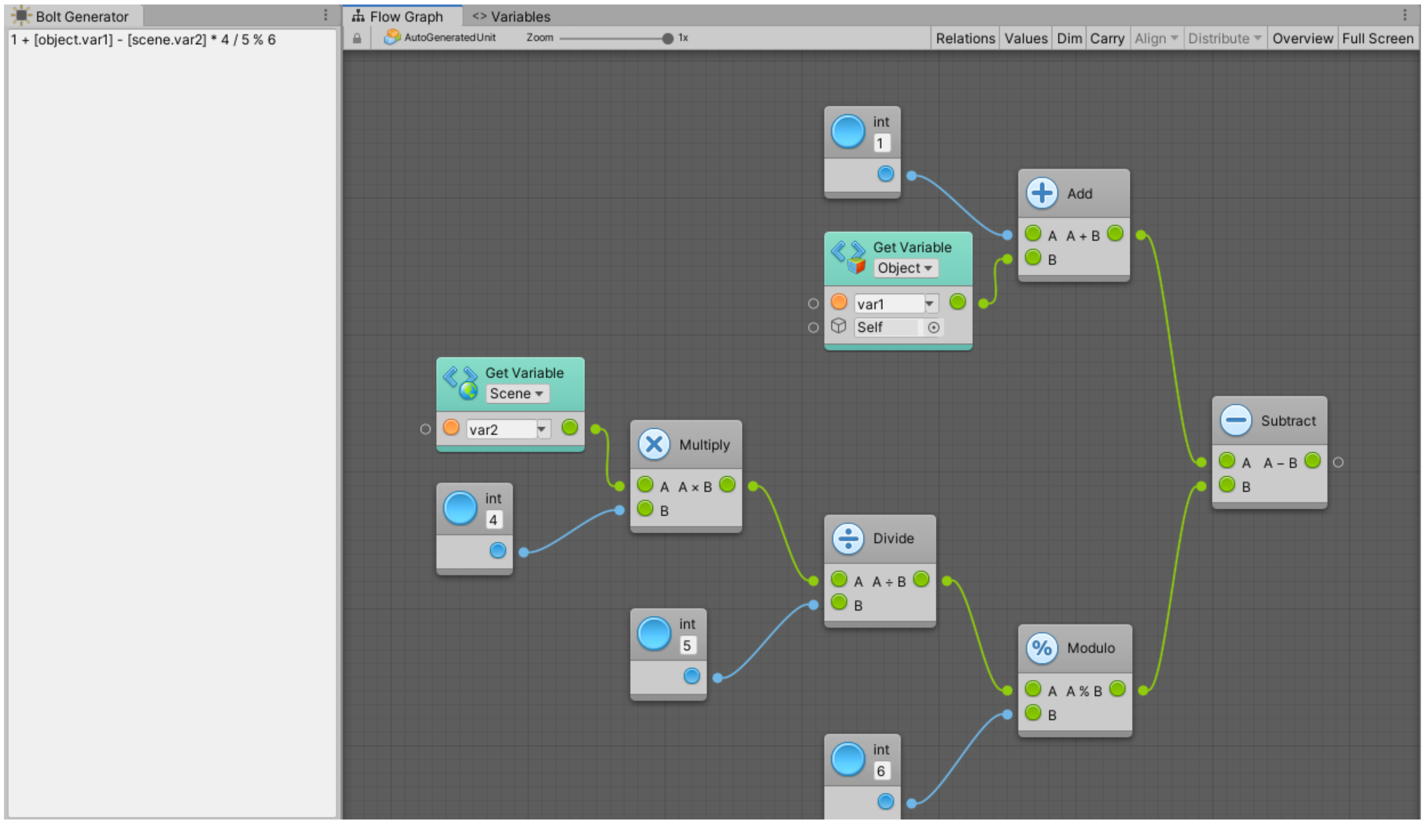
- Expression: **1 + 2 - 3 * 4 / 5 % 6**
- Generated graph



1 + 2 - 3 * 4 / 5 % 6

## Example 2

- Expression: **1 + [object.var1] - [scene.var2] * 4 / 5 % 6**
- Generated graph



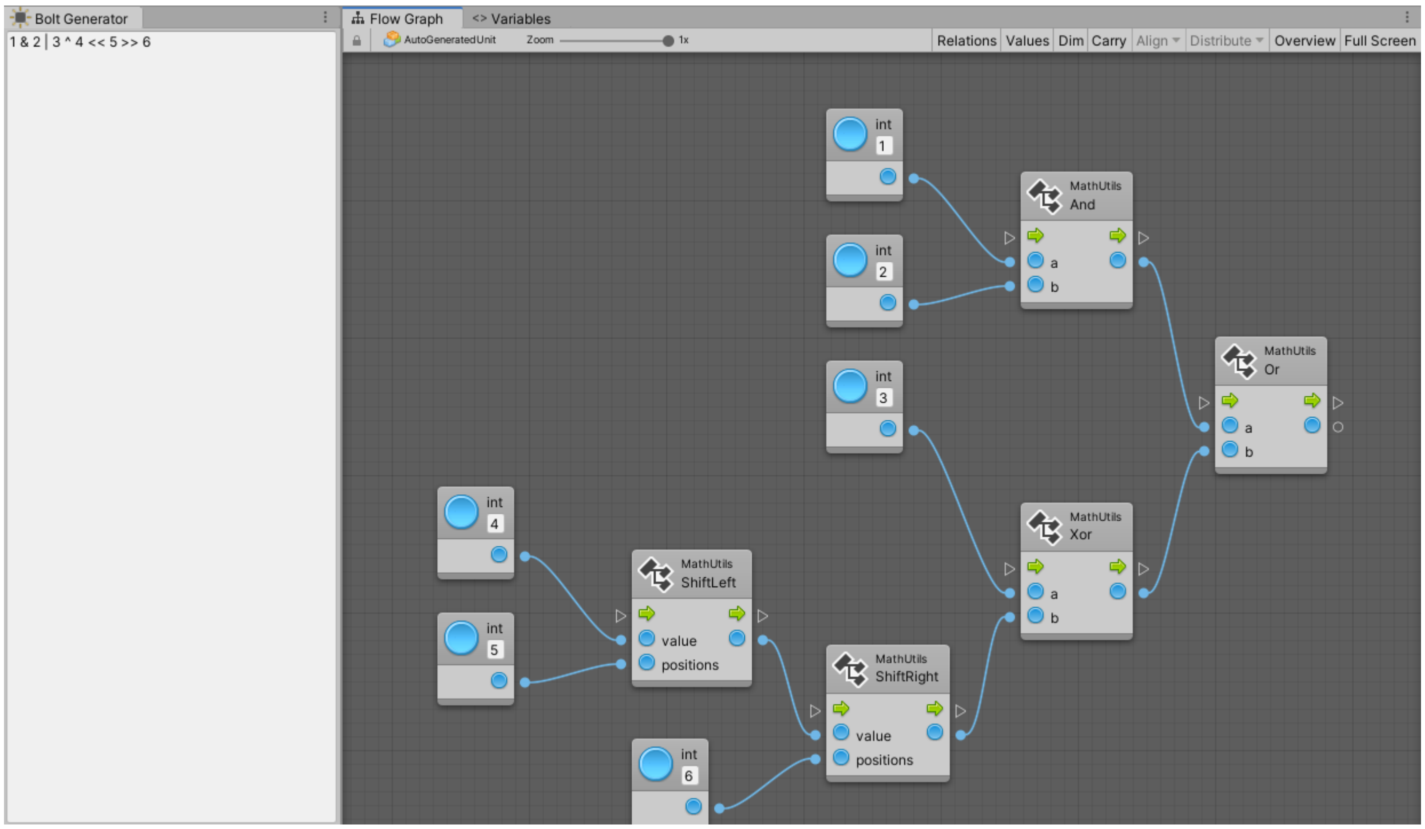
1 + [object.var1] - [scene.var2] * 4 / 5 % 6

**Support bitwise operators**

- And (&)
- Or (|)
- Xor (^)
- Shift Left (<<)
- Shift Right. (>>)

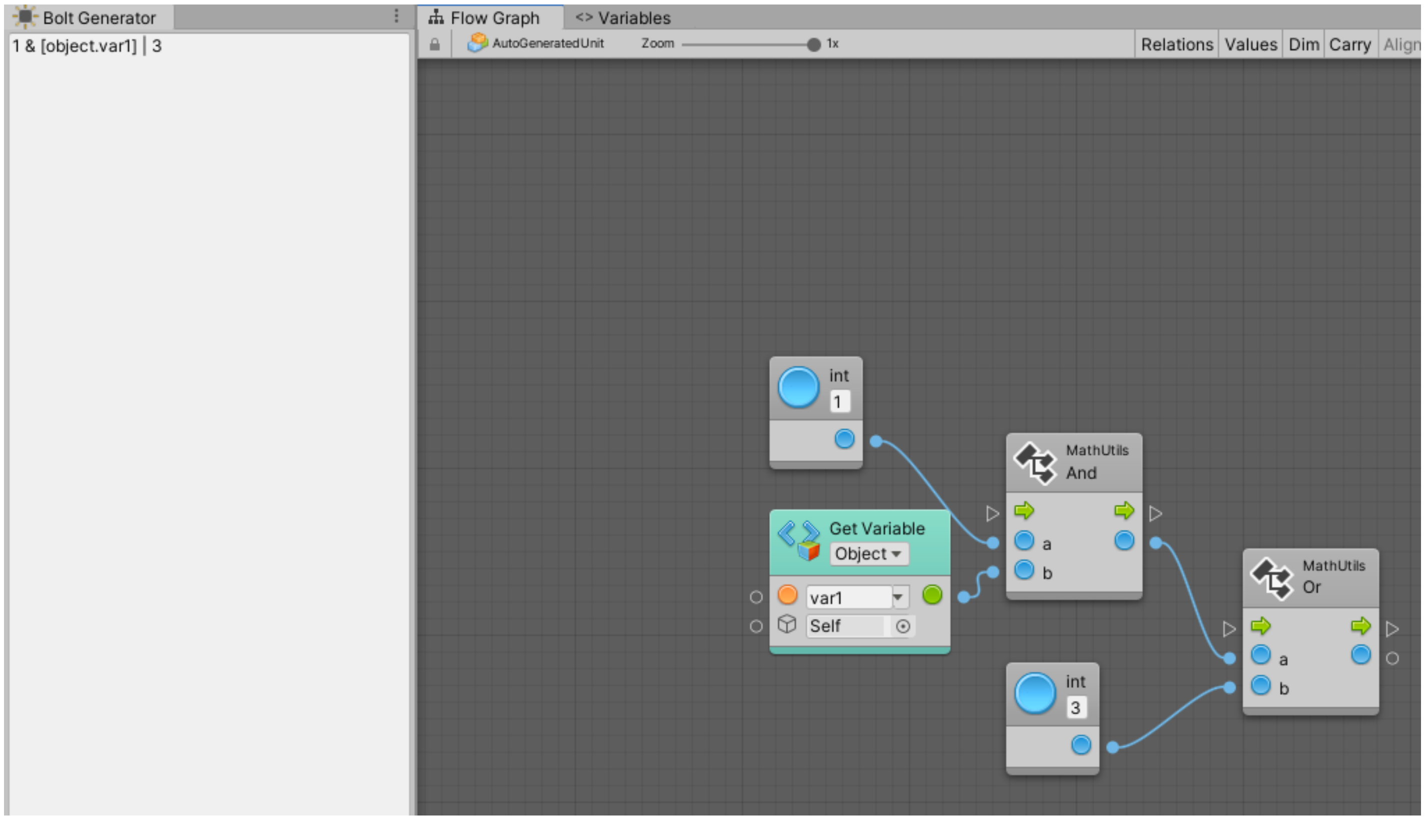
Example 1

- Expression: **1 & 2 | 3 ^ 4 << 5 >> 6**
- Generated graph



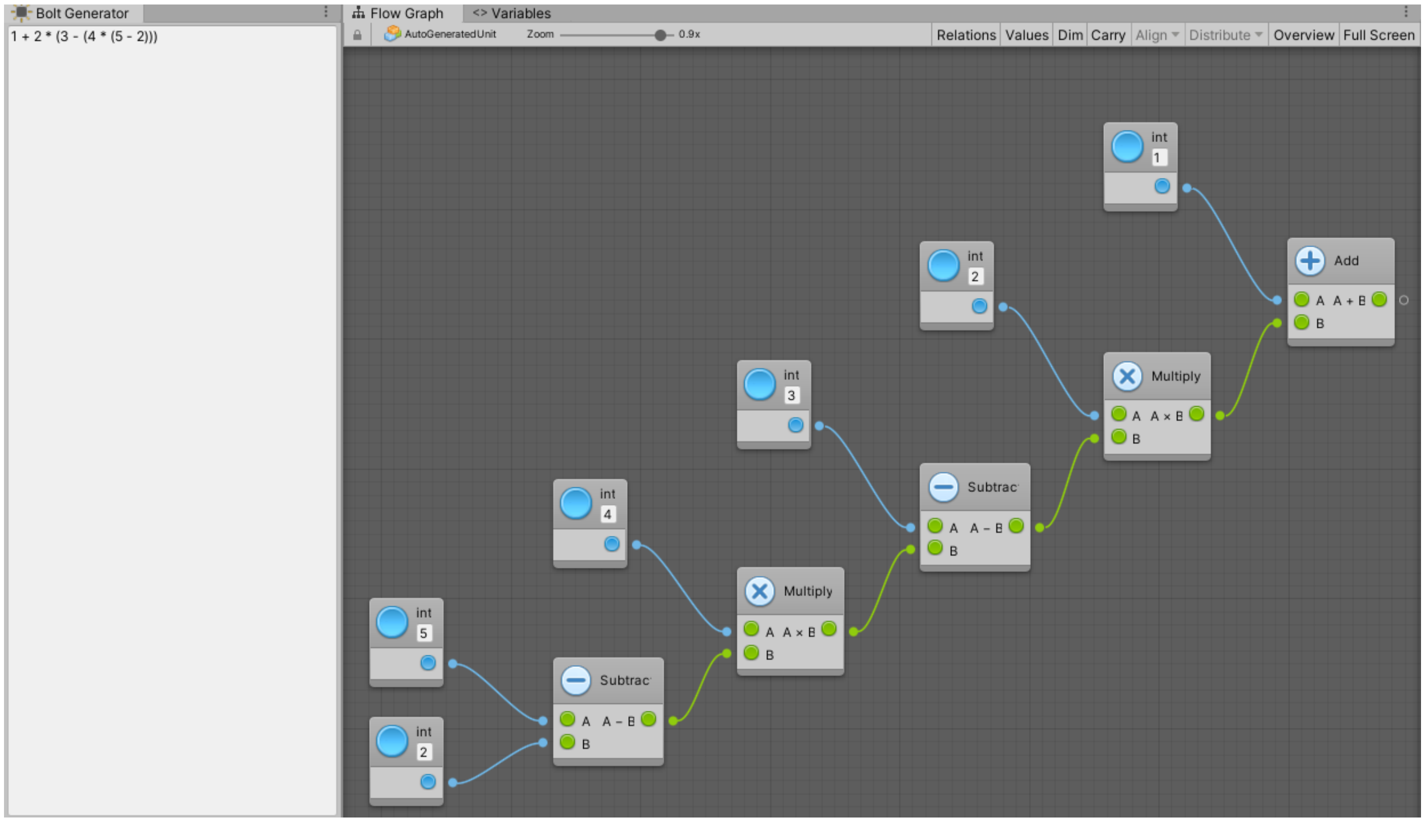
Example 2

- Expression: **1 & [object.var1]** | **3**
- Generated graph


1 & [object.var1] | 3

**Support round brackets for the operand priority**

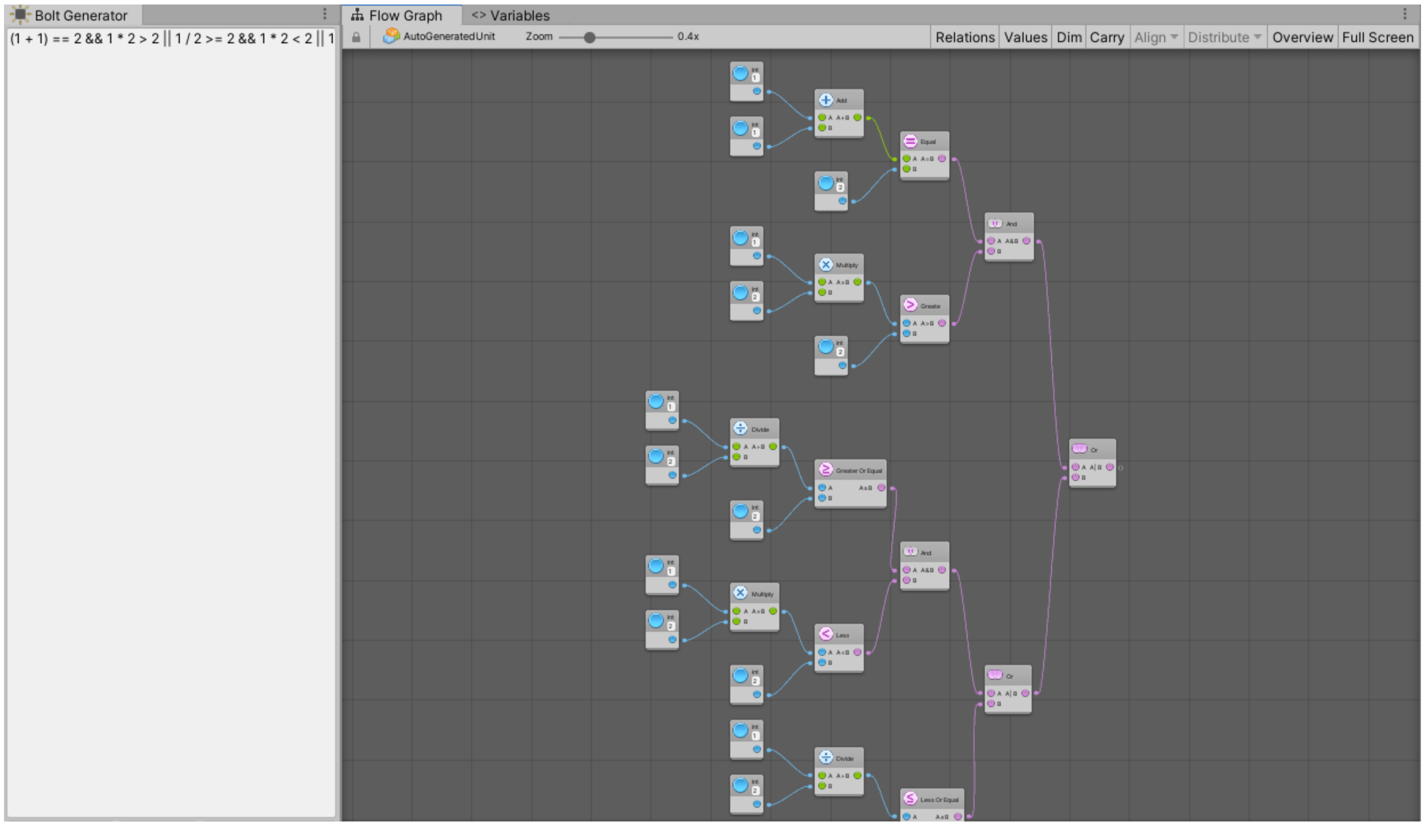
- Example
    - Expression: **1 + 2 * (3 - (4 * (5 - 2)))**
    - Generated graph



1 + 2 * (3 - (4 * (5 - 2)))

## Support boolean operators

- And (&&)
- Or (||)
- Equal (==)
- Less (<)
- Less Equal (<=)
- Greater (>)
- Greater Equal (>=)
- Example
  - Expresssion: **(1 + 1) == 2 && 1 * 2 > 2 || 1 / 2 >= 2 && 1 * 2 < 2 || 1 / 2 <= 2**
  - Generated graph

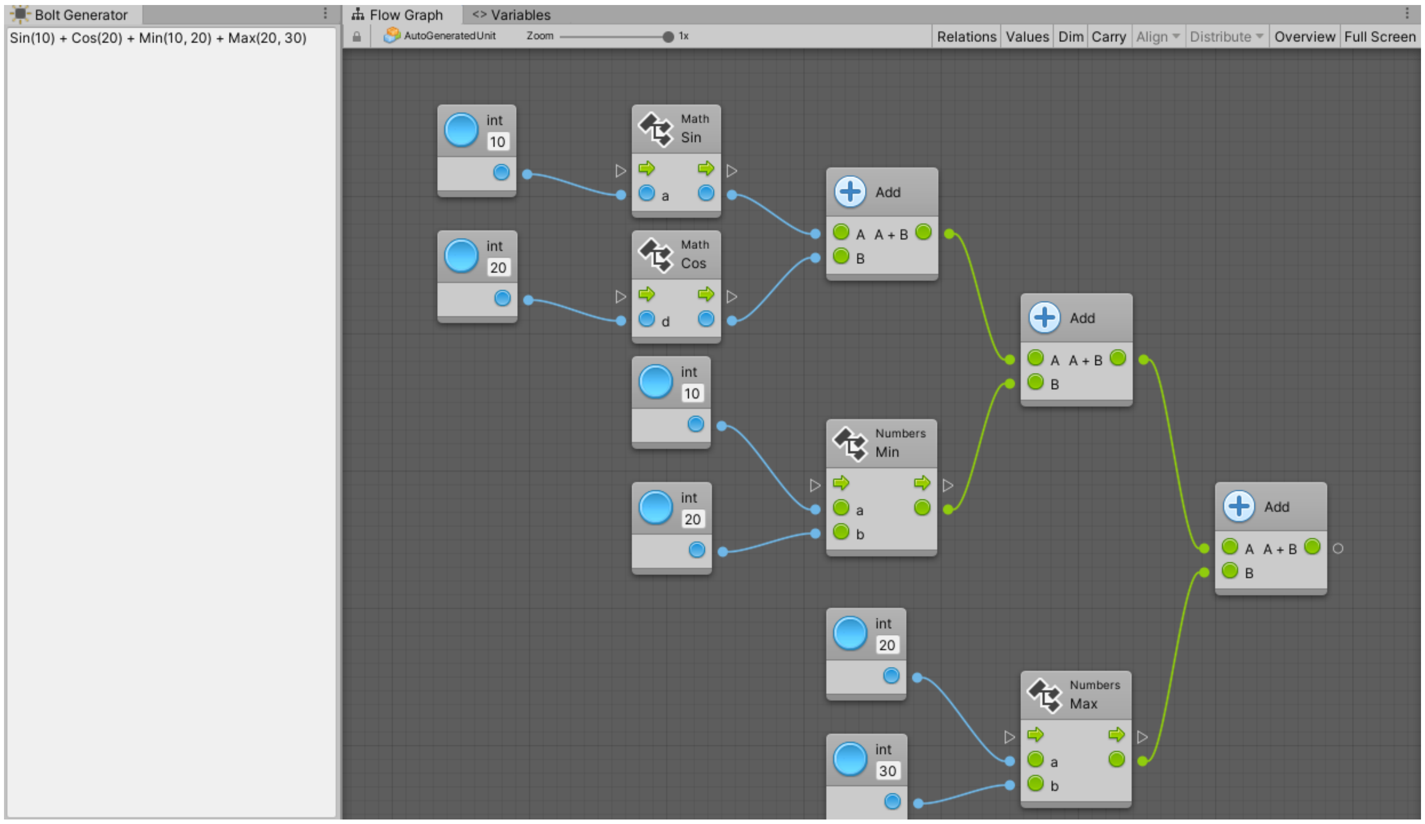


```
(1 + 1) == 2 && 1 * 2 > 2 || 1 / 2 >= 2 && 1 * 2 < 2 || 1
```

### Other math functions (None case sensitive)

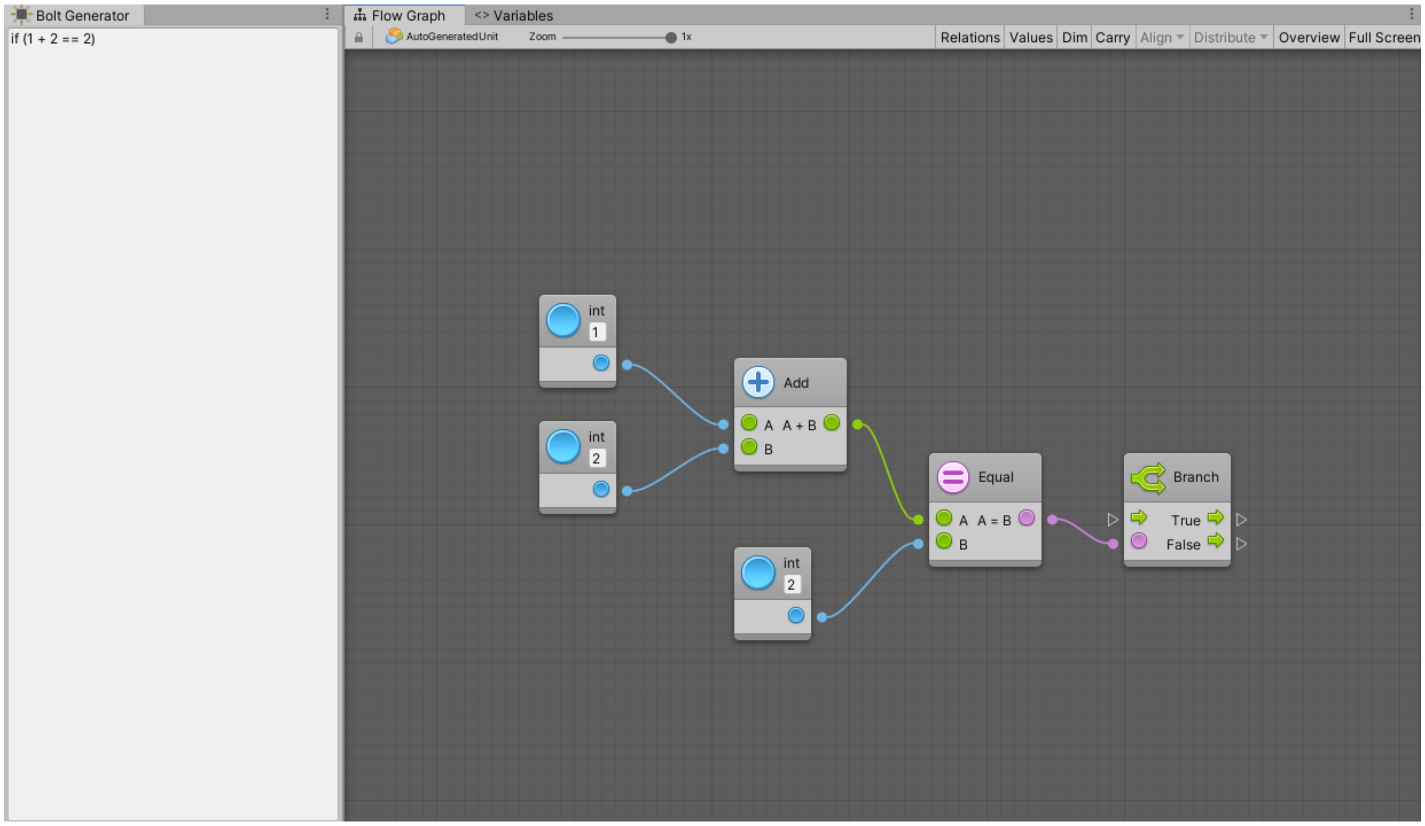
- Acos
- Asin
- Atan
- Ceiling
- Cos
- Exp
- Floor
- IEEERemainder
- Log
- Log10
- Max
- Min
- Pow
- Round
- Sign
- Sin
- Sqrt
- Tan

- Truncate
- Example
    - Expression: **Sin(10) + Cos(20) + Min(10, 20) + Max(20, 30)**
    - Generated graph

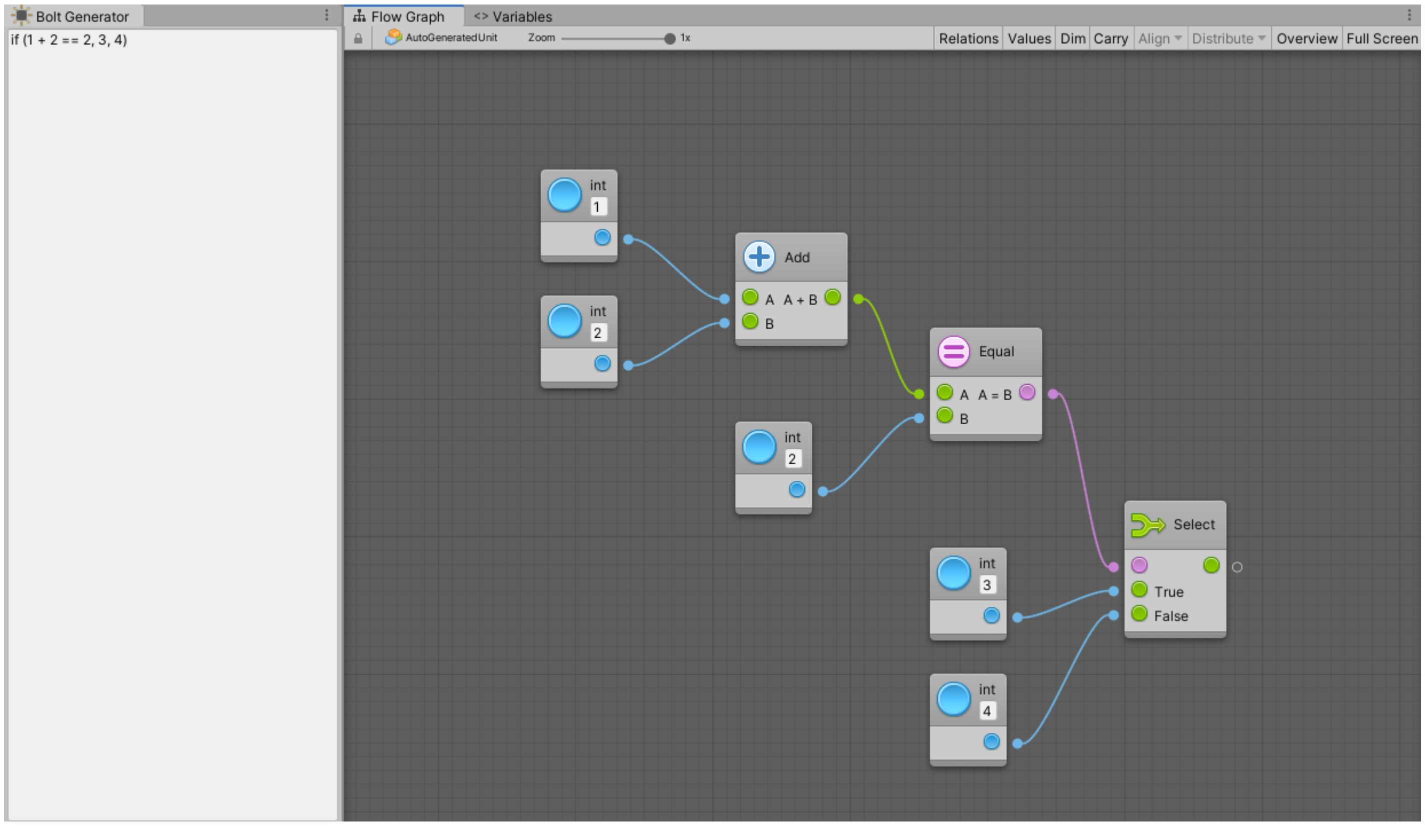


### Support if (condition) command

- **Example 1**
    - Expression: **if (1 + 2 == 2)**
    - Generated graph
- **Example 2**
    - Expression: **if (1 + 2 == 2, 3, 4)**
    - Generated graph



if (1 + 2 == 2)

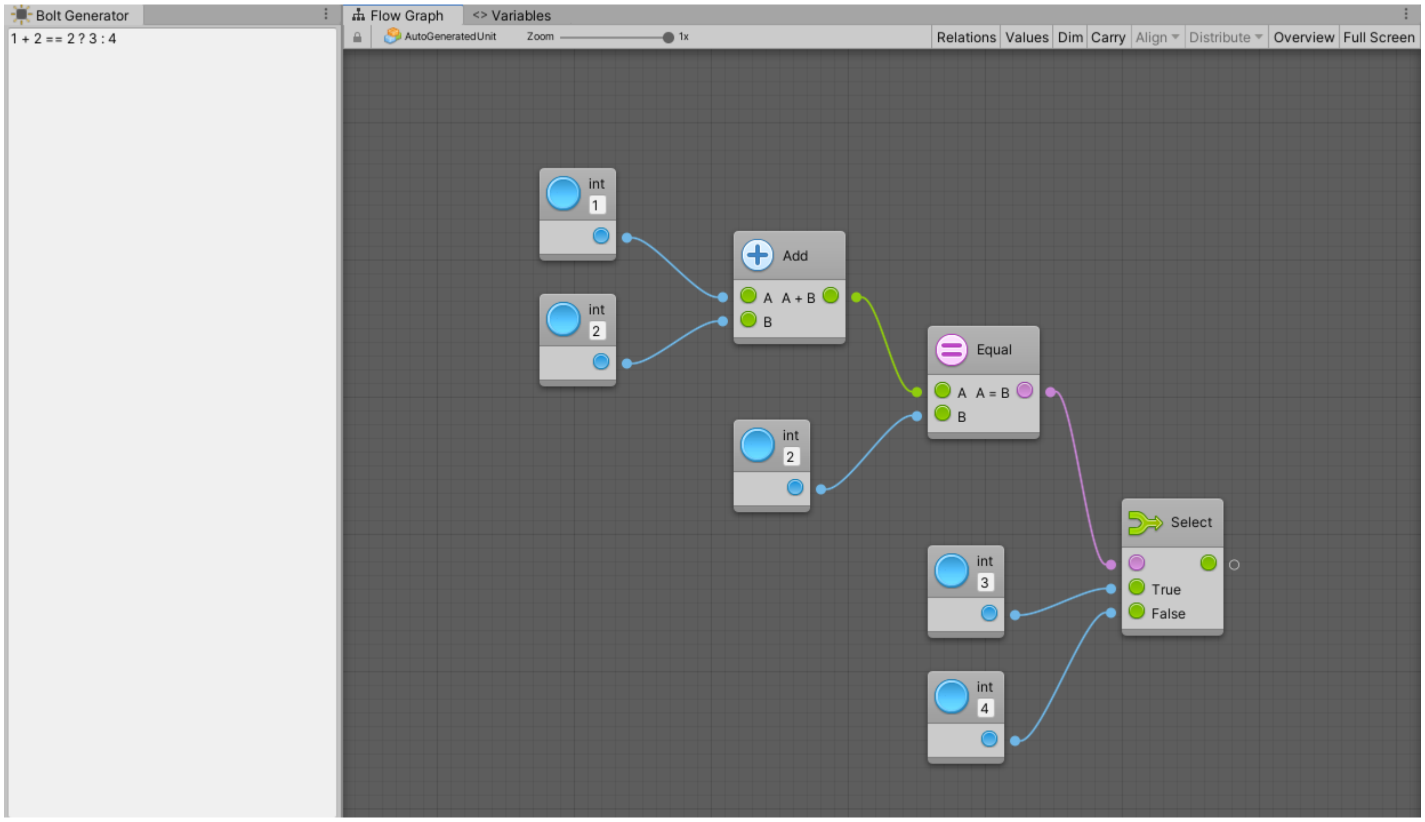


if (1 + 2 == 2, 3, 4)

## Support ternary format

- **Example**
  - Expression: **1 + 2 == 2 ? 3 : 4**

- Generated graph

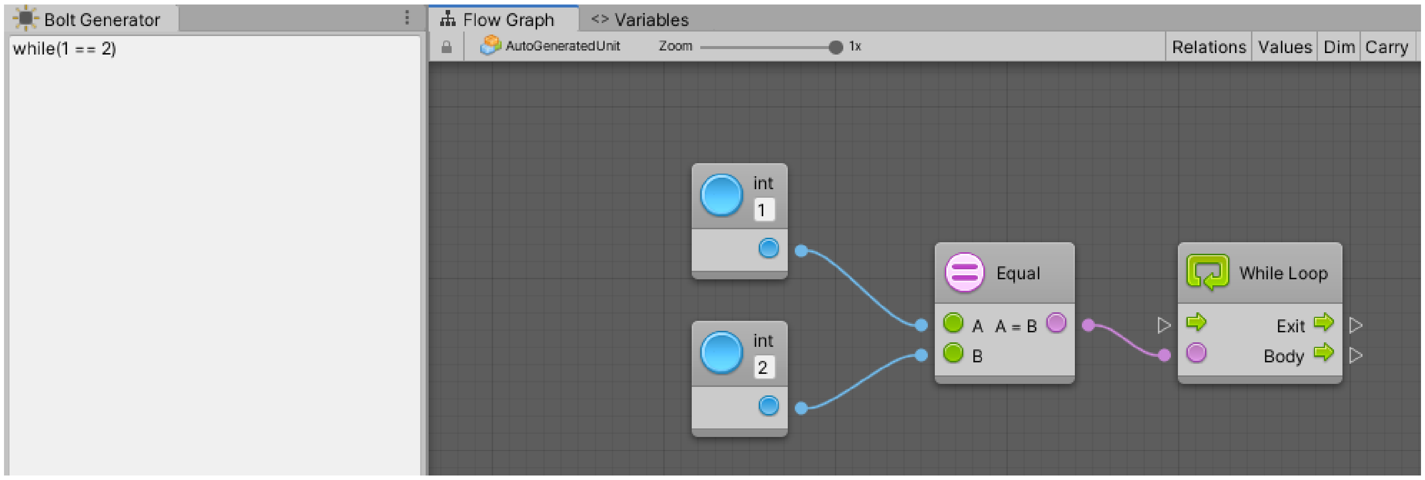


```
1 + 2 == 2 ? 3 : 4
```

Support **while(), while(condition)** commands

Example 1

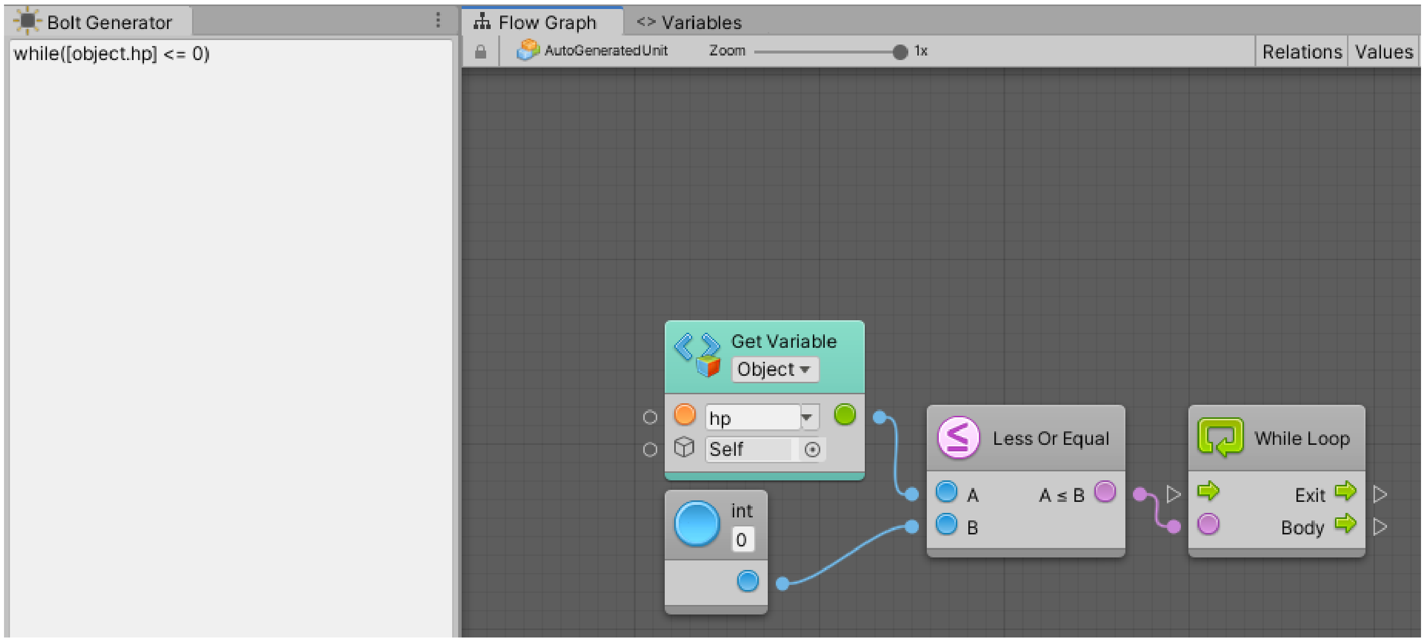
- Expression: **while(1 == 2)**
- Generated graph



```
while(1 == 2)
```

Example 2

- Expression: **while([object.hp] <= 0)**
- Generated graph

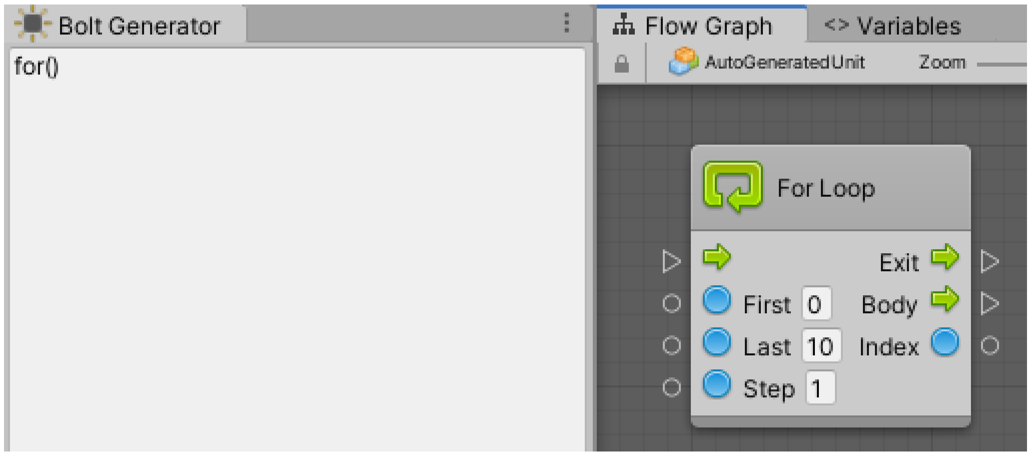


```
while([object.hp] <= 0)
```

Support **for(), for (startIndex, endIndex), for(startIndex, endIndex, step)** commands.

Example 1

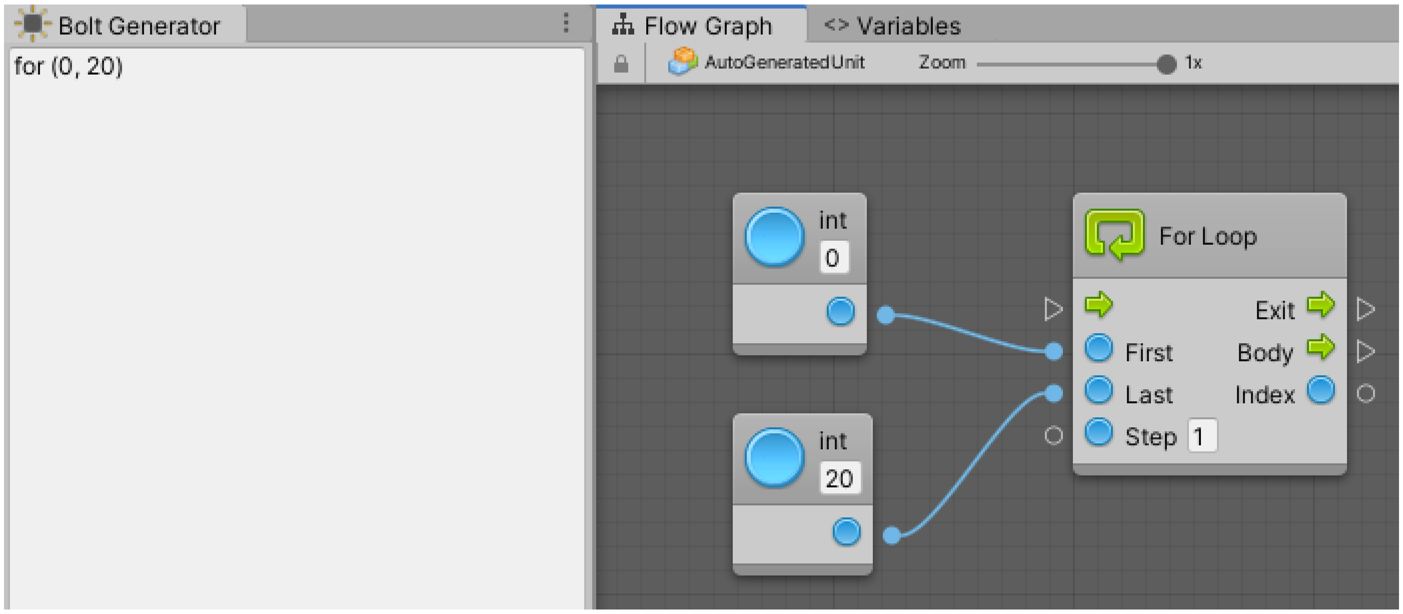
- Expression: **for()**
- Generated graph



```
for()
```

Example 2

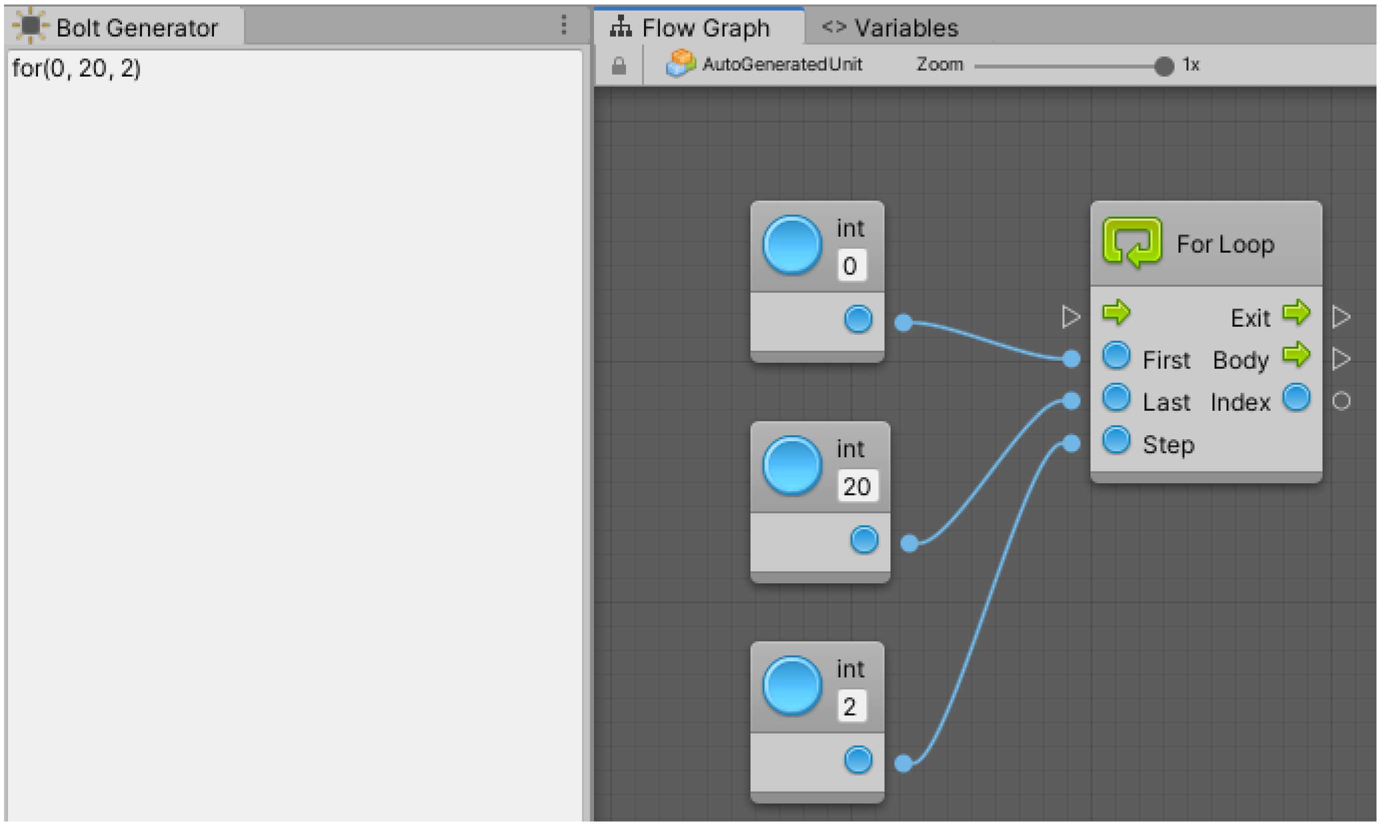
- Expression: **for (0, 20)**
- Generated graph



```
for (0, 20)
```

Example 3

- Expression: **for(0, 20, 2)**
- Generated graph



```
for(0, 20, 2)
```

**Support embed all Bolt Variables**

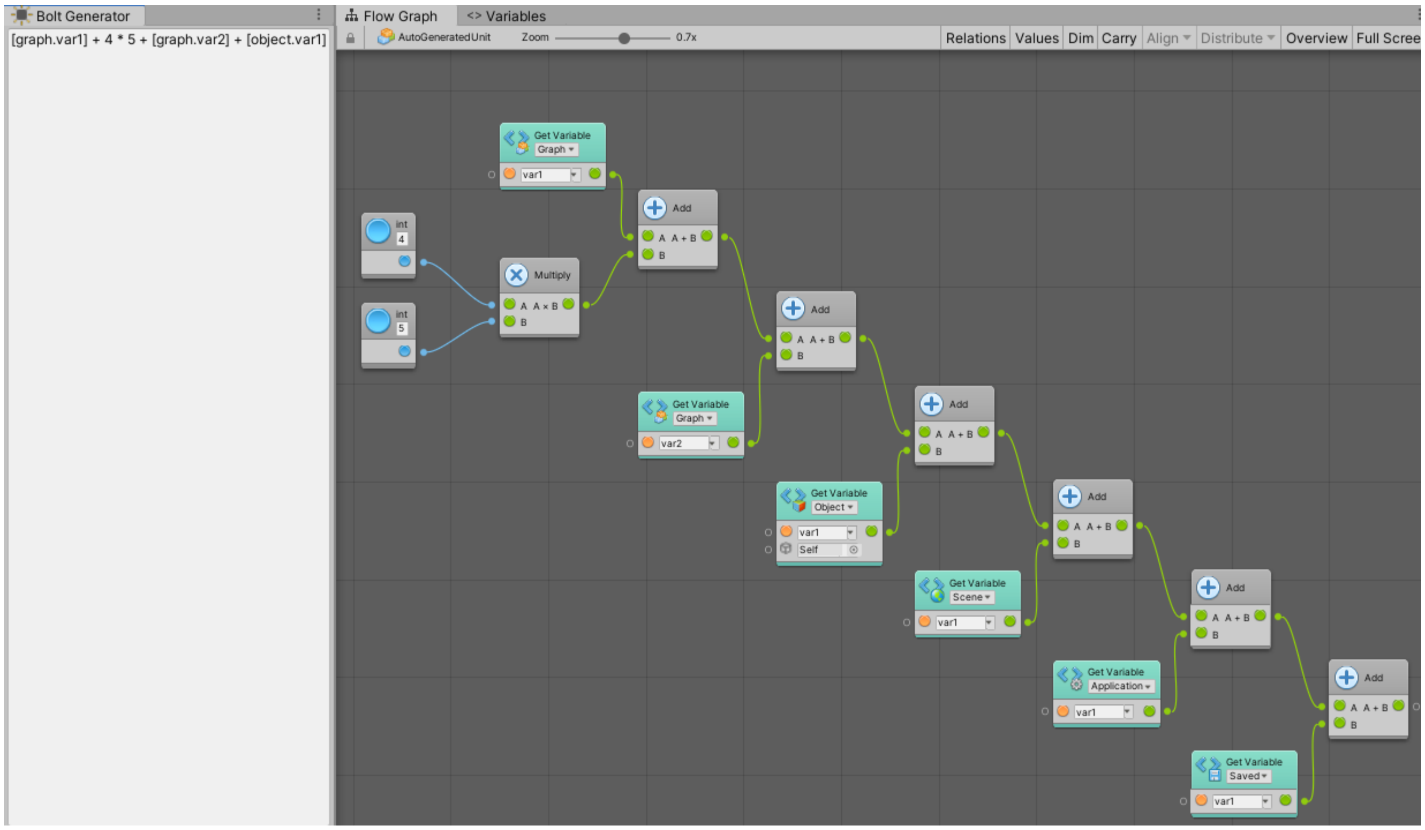
- Graph Variables (graph)
- Object Variables (object)
- Scene Variables (scene)
- App Variables (app)
- Saved Variables (saved)

Syntax

- [**type**.variableName]

Example

- Expression: **[graph.var1] + 4 * 5 + [graph.var2] + [object.var1] + [scene.var1] + [app.var1] + [saved.var1]**
- Generated graph


[graph.var1] + 4 * 5 + [graph.var2] + [object.var1]

## Bolt Debug

The standard Unity Debug Log unit and Console Window is a powerful Logging Utility for the game development with the Scripting. But with the project using **Bolt**, it's pretty hard for finding the specific unit which shows the log message, especially when you're working on a huge, spaghetti flow graph.

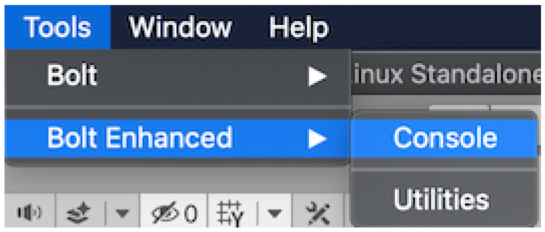
**Bolt Debug** is developed with the purpose of reducing your pain when you want to debug your Bolt project. Here are the current features of Bolt Debug:

- A separate Console Window for Bolt
- Allow jumping to the specific node that causes the log and highlight it
- Custom Debug Unit for Bolt

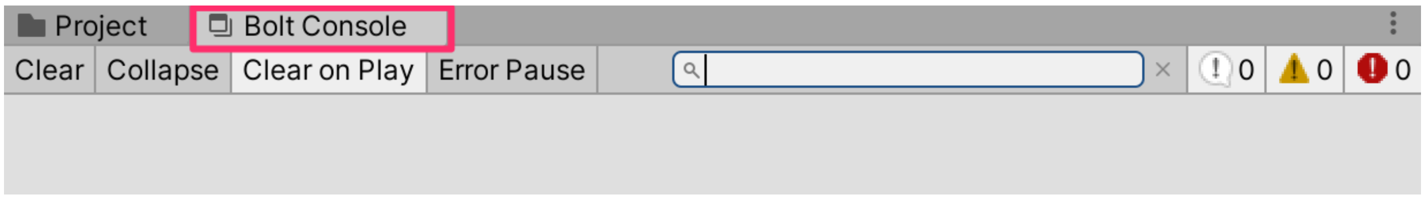
## Getting Started

**Step 1.** Open the **Bolt Console** window.

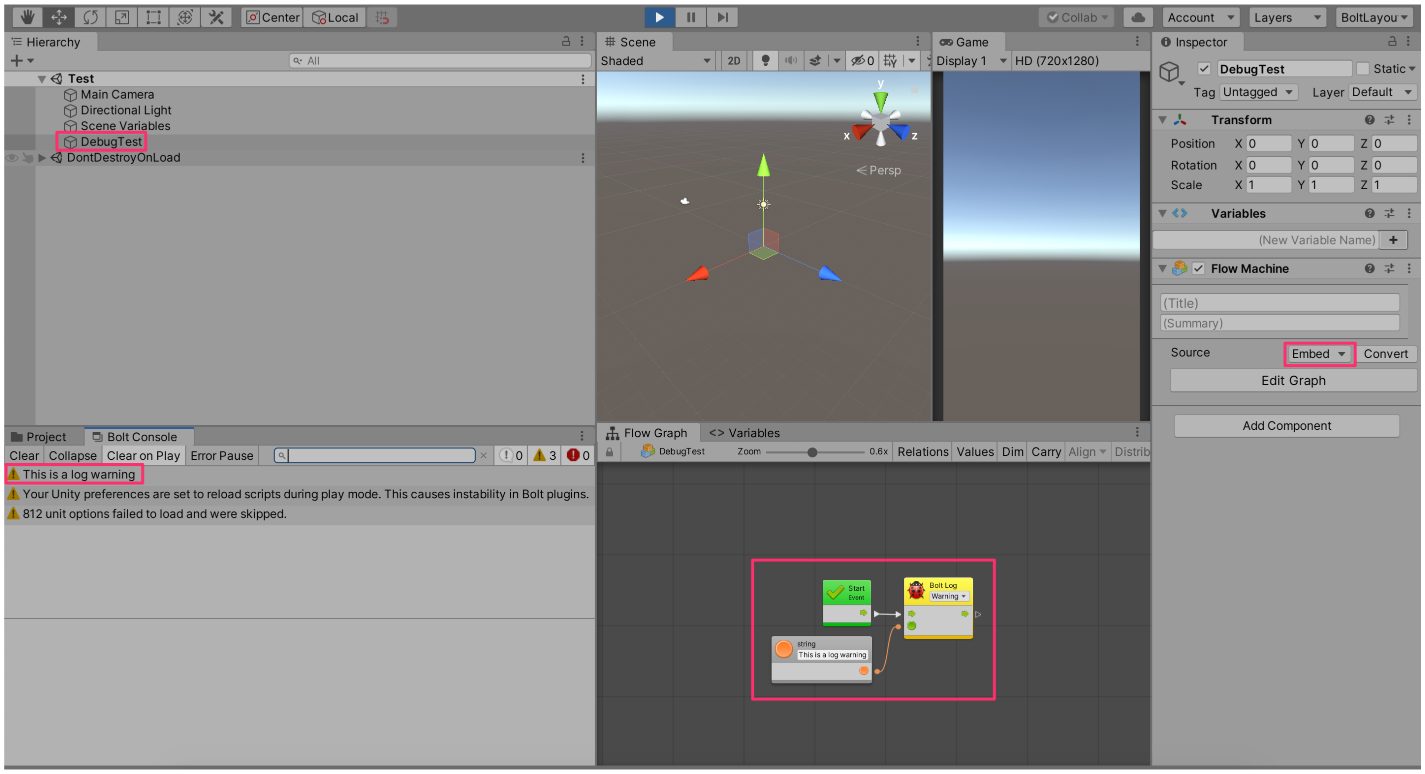
From the **menu bar** of Unity Editor, **Tools ➞ Bolt Enhanced ➞ Console**



**Step 2.** Put the **Bolt Console** window somewhere inside the Unity Editor area.

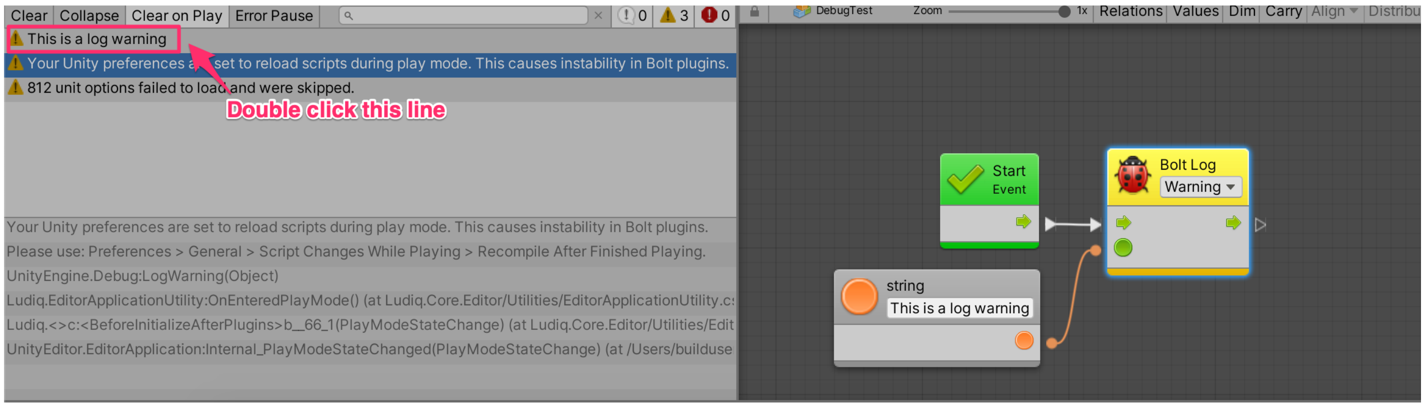


**Step 3.** Right-click on the Flow Graph, select **Bolt Enhanced/Bolt Debug/Bolt Log** to add the **BoltLog** unit, then create a **String Literal** unit with a text value, connect the **String Literal** unit to the **BoltLog** and observe the message shows in the **Bolt Console** window. Let's zoom out the **Graph** a little bit for testing the next step.



```
This is a log warning
Your Unity preferences are set to reload scripts during play mode. This causes instability in Bolt plugins.
812 unit options failed to load and were skipped.
```

**Step 4.** From the **Bolt Console** window, double click on the **"This is a log warning"** line. Observe the **Graph** window will automatically focus and highlight the **Bolt Log** unit.



```
This is a log warning
Your Unity preferences are set to reload scripts during play mode. This causes instability in Bolt plugins.
812 unit options failed to load and were skipped.

Your Unity preferences are set to reload scripts during play mode. This causes instability in Bolt plugins.
Please use: Preferences > General > Script Changes While Playing > Recompile After Finished Playing.
UnityEngine.Debug:LogWarning(Object)
Ludiq.EditorApplicationUtility:OnEnteredPlayMode() (at Ludiq.Core.Editor/Utilities/EditorApplicationUtility.c
Ludiq.<>c:<BeforeInitializeAfterPlugins>b__66_1(PlayModeStateChange) (at Ludiq.Core.Editor/Utilities/Edit
UnityEditor.EditorApplication:Internal_PlayModeStateChanged(PlayModeStateChange) (at /Users/builduse
```

**NOTE:** The feature *"automatically focus and highlight the node"* can only work in the **Unity Play** mode. So keep the **Unity Editor playing** when double click on the log message line.

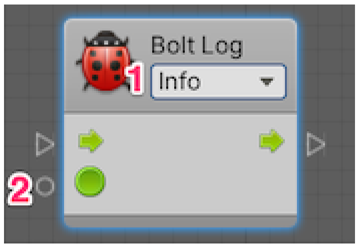
Done!

For more usage of **Bolt Debug**, let's navigate to the directory **Bolt.Enhanced/Debug/Examples** which is already included in the asset.

## Super Units

**Category**

BoltEnhanced ➞ Bolt Debug ➞ Bolt Log



This is used for showing the log message. This custom unit is required in order to work with Bolt Console Window. Basically, this unit sends some extra information that helps to locate the specific Bolt unit.

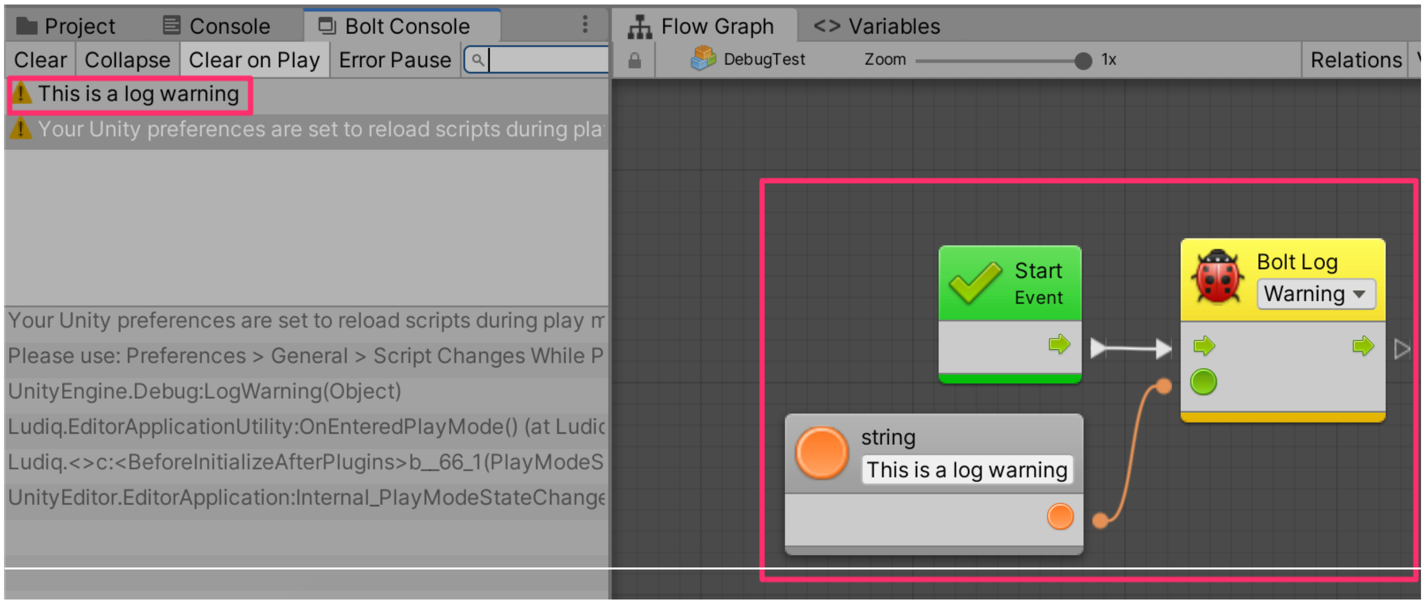
1 - **Log Type**: The type of Log

There are 3 options:

- **Info**: Equal to **Debug.Log** unit
- **Warning**: Equal to **Debug.Warning** unit
- **Error**: Equal to **Debug.Error** unit

2 - **Log Message**: The message you want to show on the Bolt Console Window

**Example**



```
This is a log warning
Your Unity preferences are set to reload scripts during pla
```

```
Your Unity preferences are set to reload scripts during play m
Please use: Preferences > General > Script Changes While P
UnityEngine.Debug:LogWarning(Object)
Ludiq.EditorApplicationUtility:OnEnteredPlayMode() (at Ludic
Ludiq.<>c:<BeforeInitializeAfterPlugins>b__66_1(PlayModeS
UnityEditor.EditorApplication:Internal_PlayModeStateChange
```

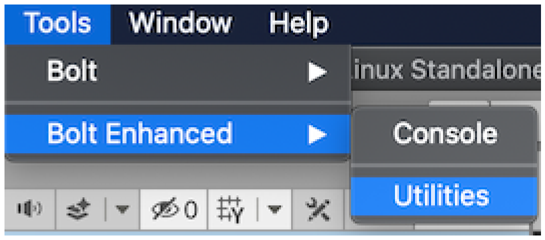
## Bolt Statistic

### Getting Started

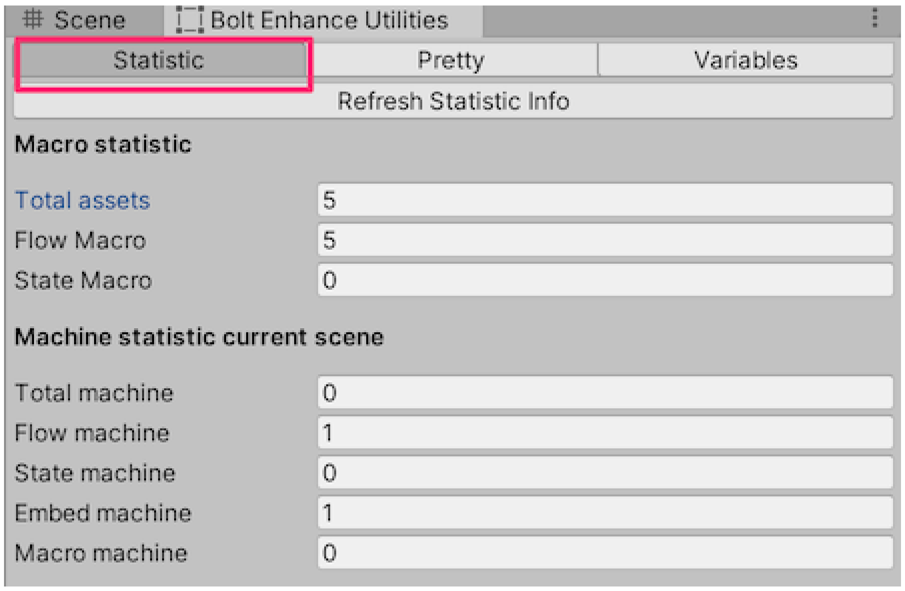
This feature basically collects the statistical information about your Bolt project.

To use this feature:

**Step 1:** From the Unity Menu Bar, **Tools ➞ Bolt Enhanced ➞ Utilities**.



**Step 2:** Select the "**Statistic**" Tab and observe the information.



Macro statistic

| | |
|---|---|
| Total assets | 5 |
| Flow Macro | 5 |
| State Macro | 0 |

Machine statistic current scene

| | |
|---|---|
| Total machine | 0 |
| Flow machine | 1 |
| State machine | 0 |
| Embed machine | 1 |
| Macro machine | 0 |

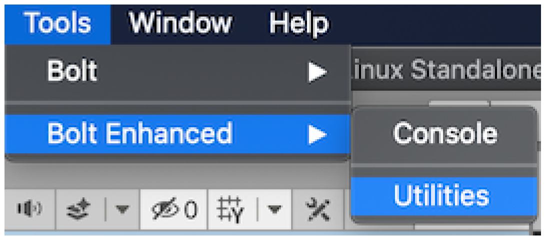
## Bolt Pretty

### Getting Started

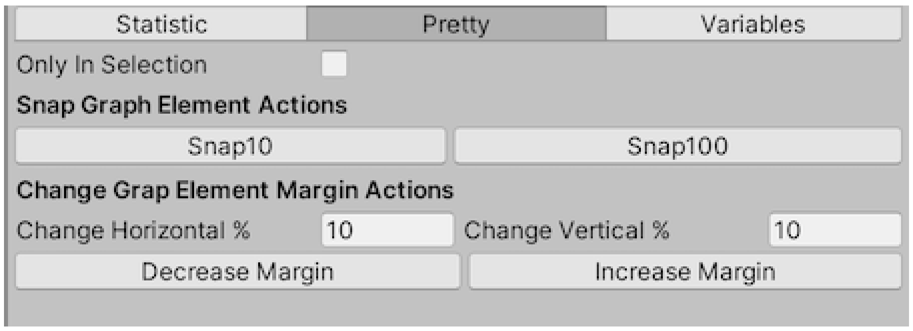
This feature allows you to re-organize your flow graph to make it easier for reading.

To use this feature:

**Step 1:** From the Unity Menu Bar, **Tools ➞ Bolt Enhanced ➞ Utilities**.



**Step 2:** Select the "**Pretty**" Tab and try its functions.



Statistic | Pretty | Variables

Only In Selection

**Snap Graph Element Actions**

Snap10 | Snap100

**Change Grap Element Margin Actions**

Change Horizontal % 10 | Change Vertical % 10

Decrease Margin | Increase Margin

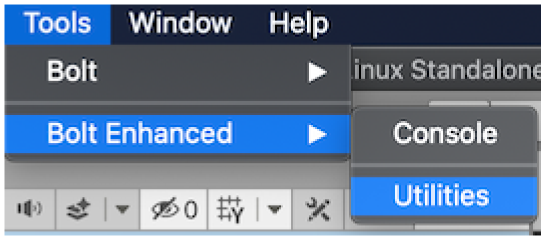
## Bolt Variables

### Getting Started

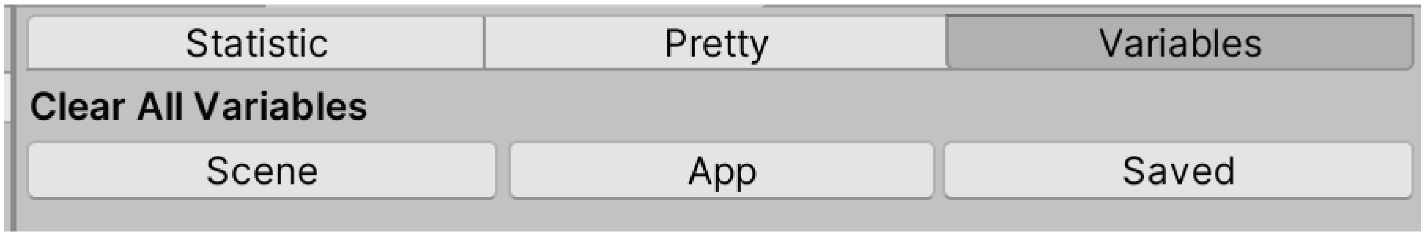
This feature allows you to quickly delete all the saved variables, the app variable, the scene variable.

To use this feature:

**Step 1:** From the Unity Menu Bar, **Tools ➞ Bolt Enhanced ➞ Utilities**.



**Step 2:** Select the "**Variables**" Tab and try to delete some variables.



## Bolt Finder

Do you have a struggle when trying to find where a flow graph is used? It's a little more painful when a flow graph is used inside an embedded flow graph. This is our motivation for developing the Bolt Finder feature.

Here are the implemented functions in this asset

- Quick search bolt file in your project
- Find usage of (flow, state) macro in another macro (Multiple occur points)
- Find usage of (flow, state) macro in the current scene (Embed, Macro, Multiple machines)

We hope you find this feature useful and save your time in developing the Unity game with Bolt.

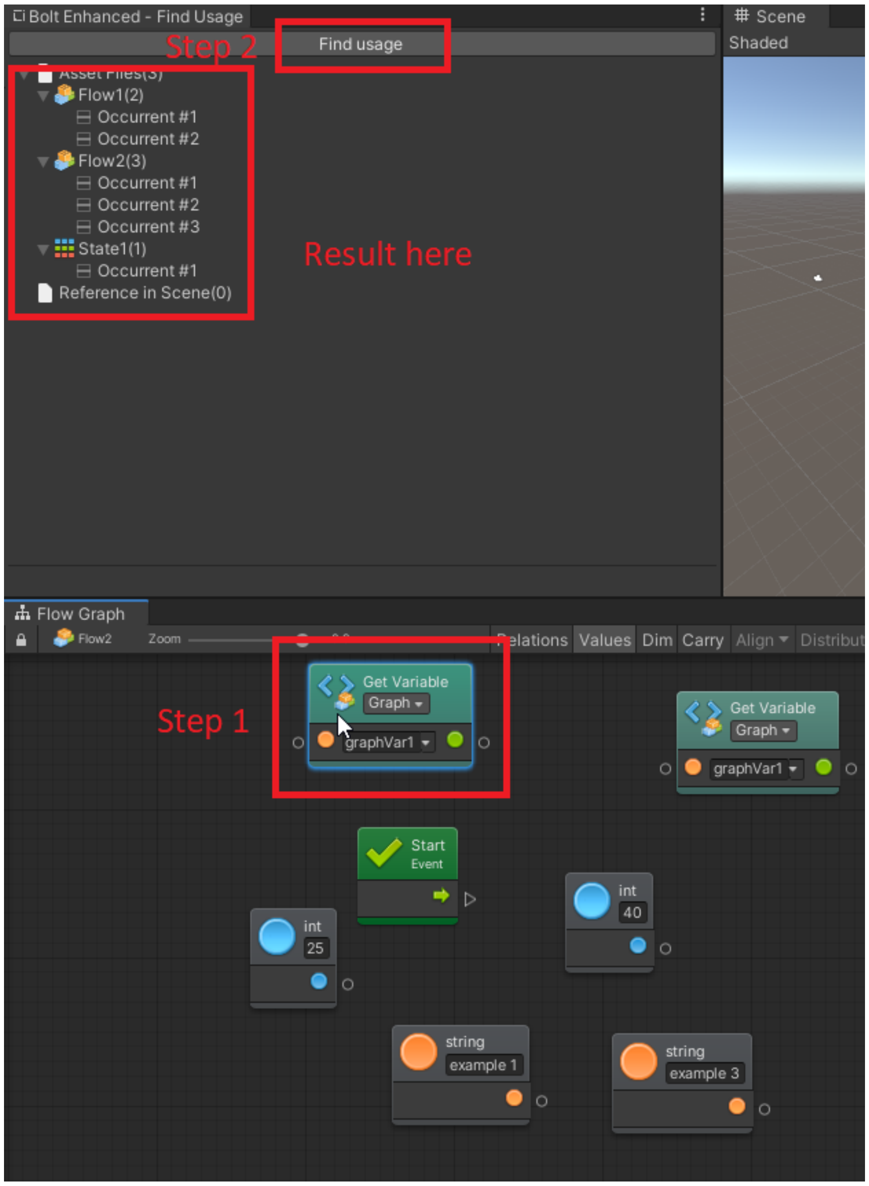
### Getting Started

There are 4 ways to find usage of a Flow Graph.

**Option 1**

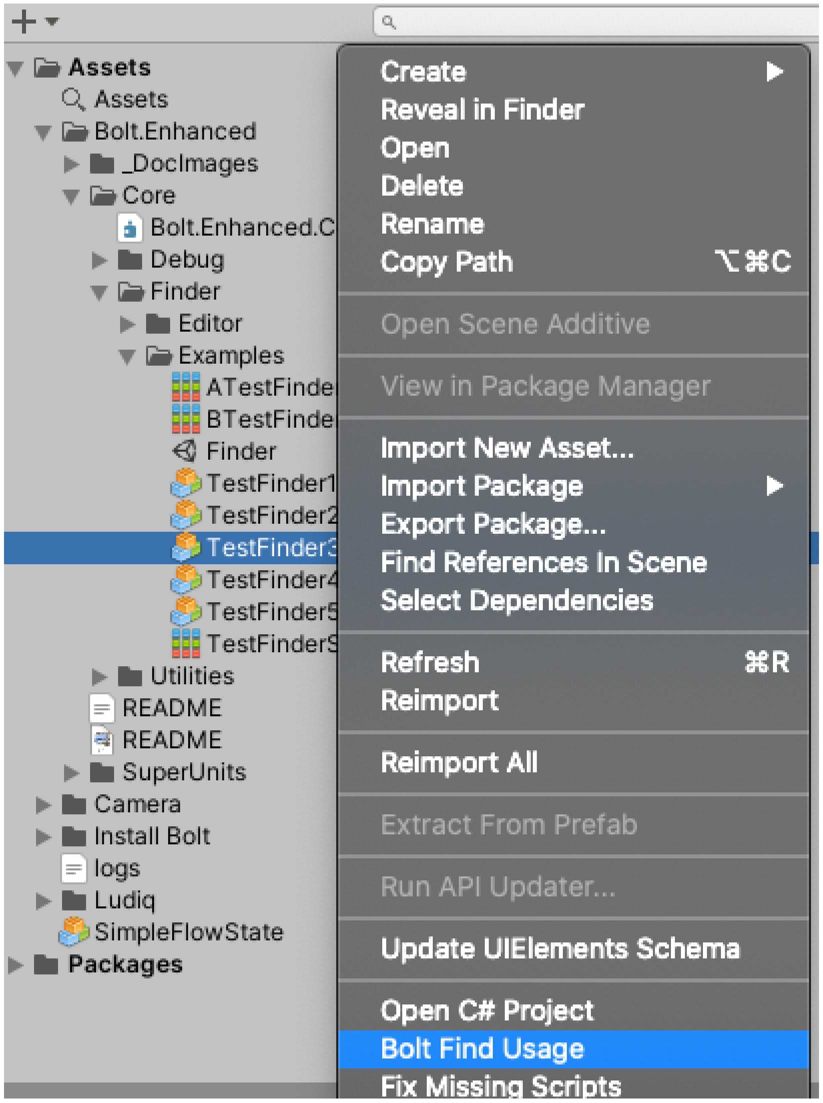
**Step 1.** From the Flow Graph window, select a unit you want to find usage.

**Step 2.** From the Find Usage window, click the Find usage button and observe the result.

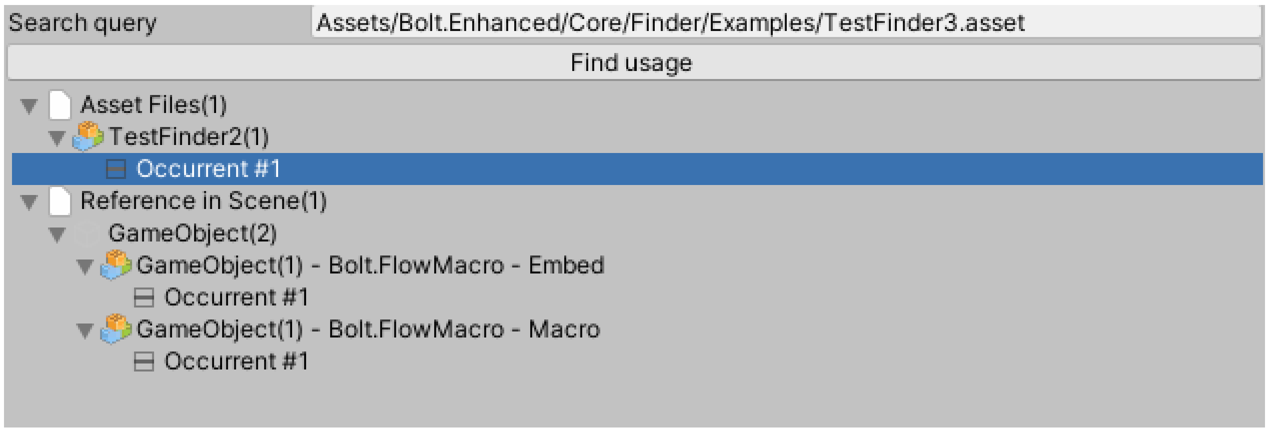


## Option 2

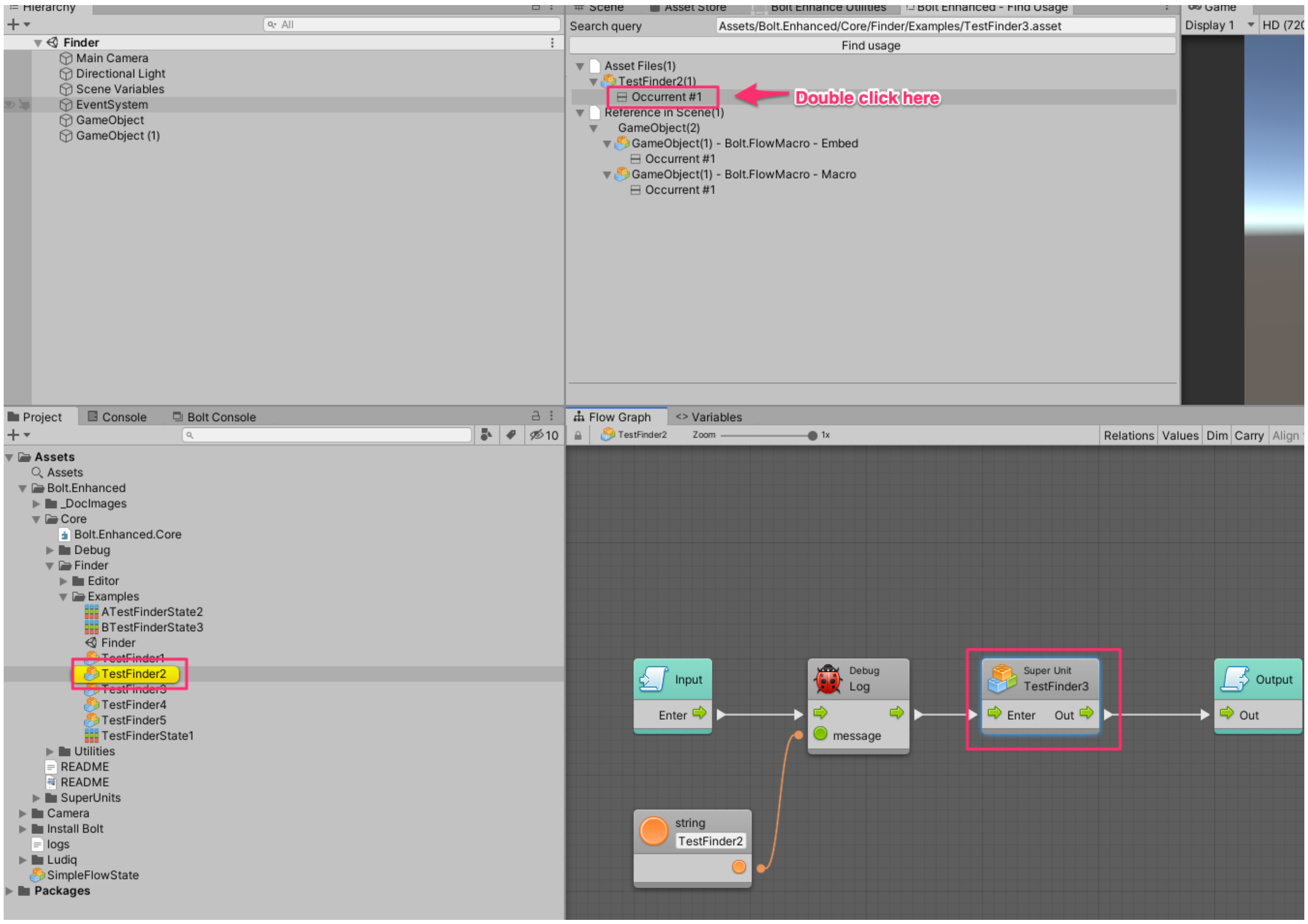
**Step 1.** From the Project Panel, **right-click** on the Flow Graph, select the **Find Usage** in the context menu panel,



**Step 2.** Observe the result inside the **Bolt Enhanced - Find Usage** window, the list of all flow graphs or objects which use the **TestFinder3** flow is listed here.


Search query
Assets/Bolt.Enhanced/Core/Finder/Examples/TestFinder3.asset
Find usage
Asset Files(1)
TestFinder2(1)
Occurrent #1
Reference in Scene(1)
GameObject(2)
GameObject(1) - Bolt.FlowMacro - Embed
Occurrent #1
GameObject(1) - Bolt.FlowMacro - Macro
Occurrent #1

**Step 3.** Let's try to click on one of the asset results to open it.

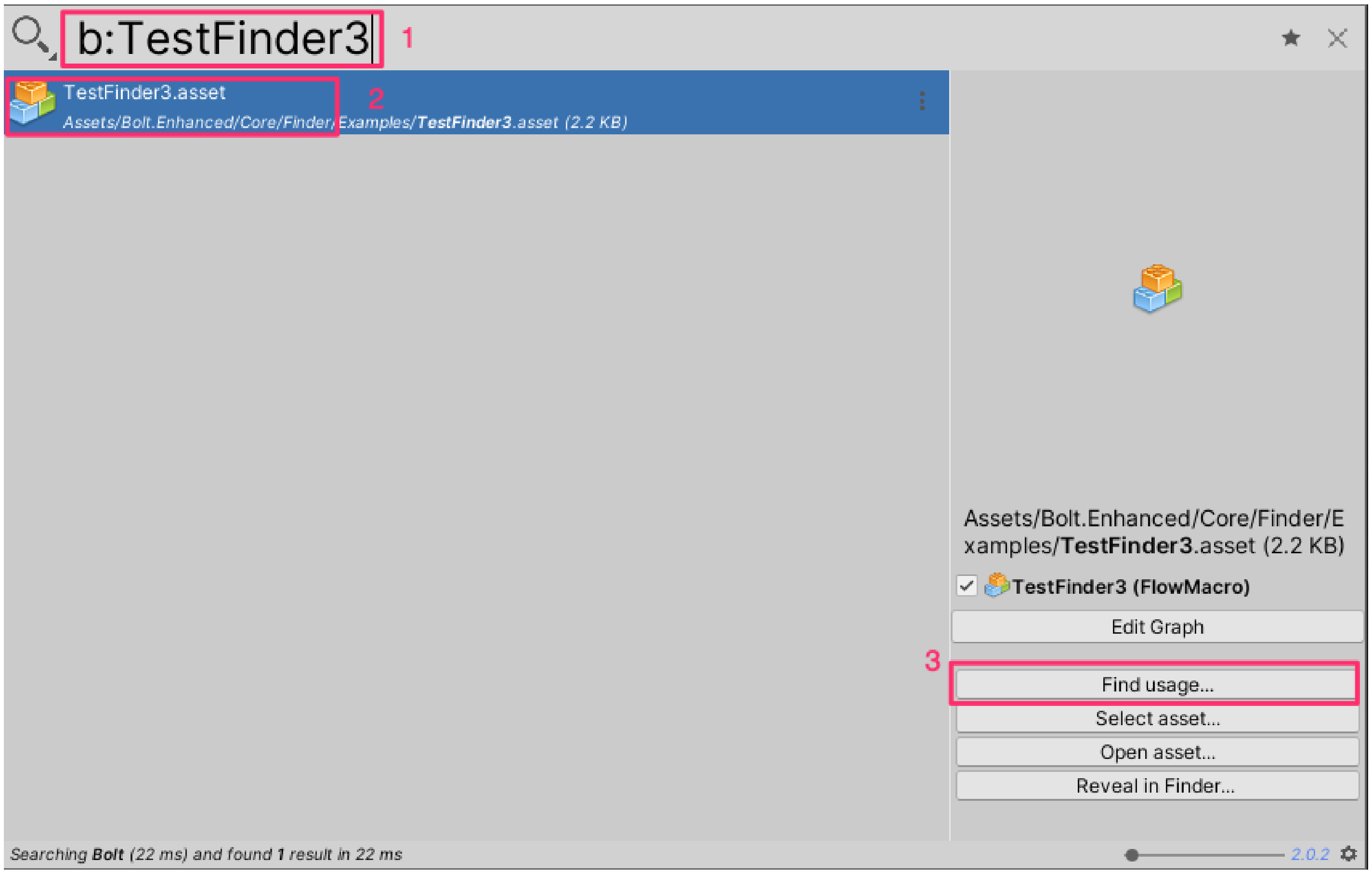


## Option 3. Finding usage using the Unity Quick Search

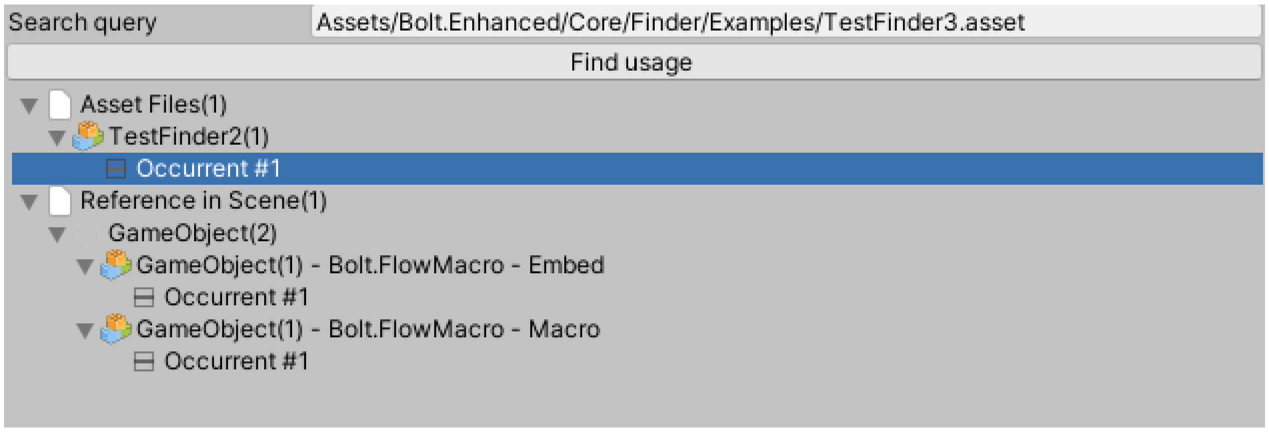
**Step 1.** From the Unity menu, select **Help -> Quick Search** to open Quick Search and type **b:TestFinder3** (1) for searching all bolt flow assets, select the asset results (2), click on **Find usage...** (3) from the right panel.

You can use the HotKey for opening **Quick Search**

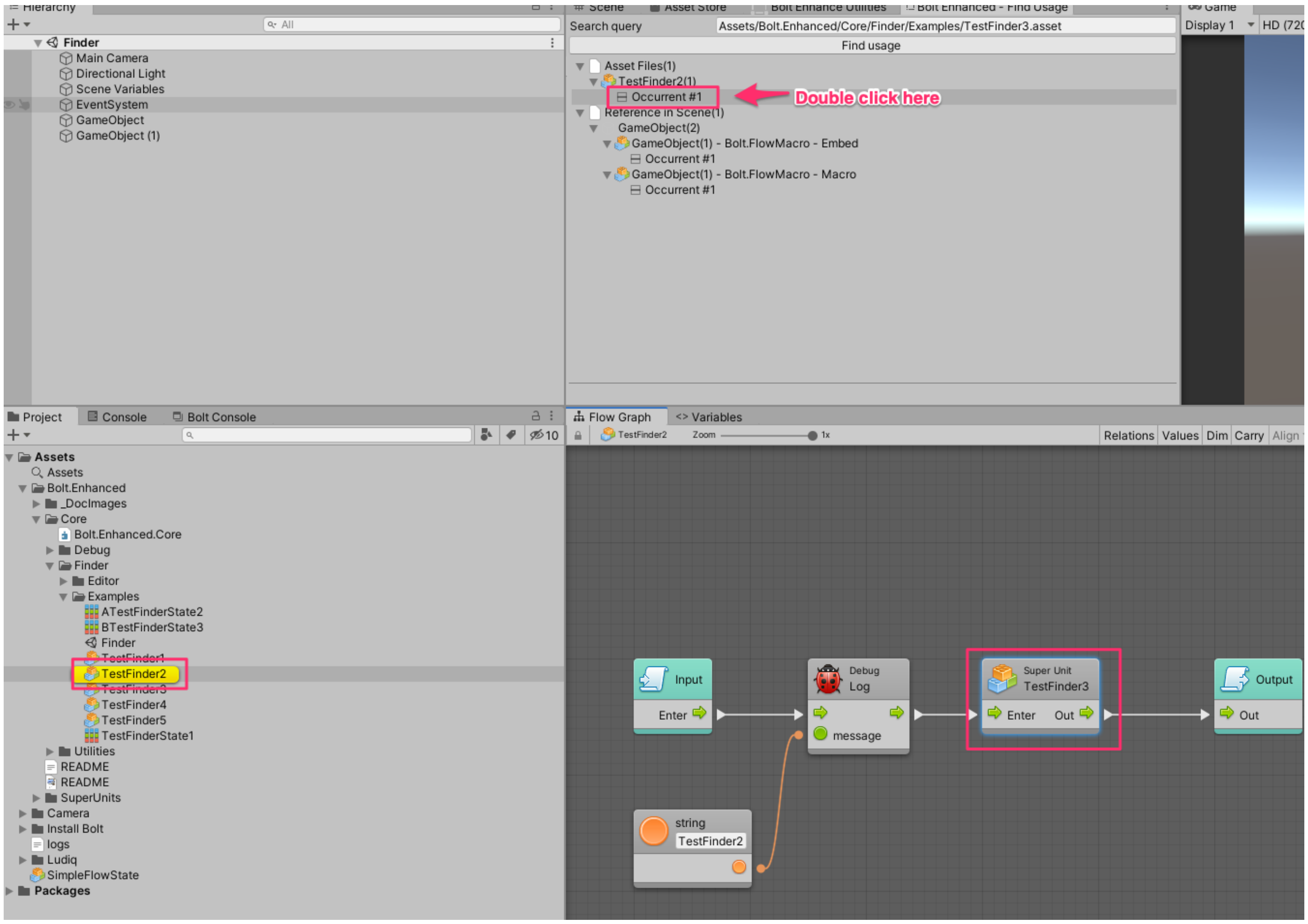
- Window/Linux: `Alt + '`
- Mac: `Option + '`


b:TestFinder3
1
TestFinder3.asset
Assets/Bolt.Enhanced/Core/Finder/Examples/TestFinder3.asset (2.2 KB)
2
Assets/Bolt.Enhanced/Core/Finder/Examples/TestFinder3.asset (2.2 KB)
TestFinder3 (FlowMacro)
Edit Graph
3
Find usage...
Select asset...
Open asset...
Reveal in Finder...
Searching Bolt (22 ms) and found 1 result in 22 ms
2.0.2

**Step 2.** Observe the result inside the **Bolt Enhanced - Find Usage** window, the list of all flow graphs or objects which use the **TestFinder3** flow is listed here.

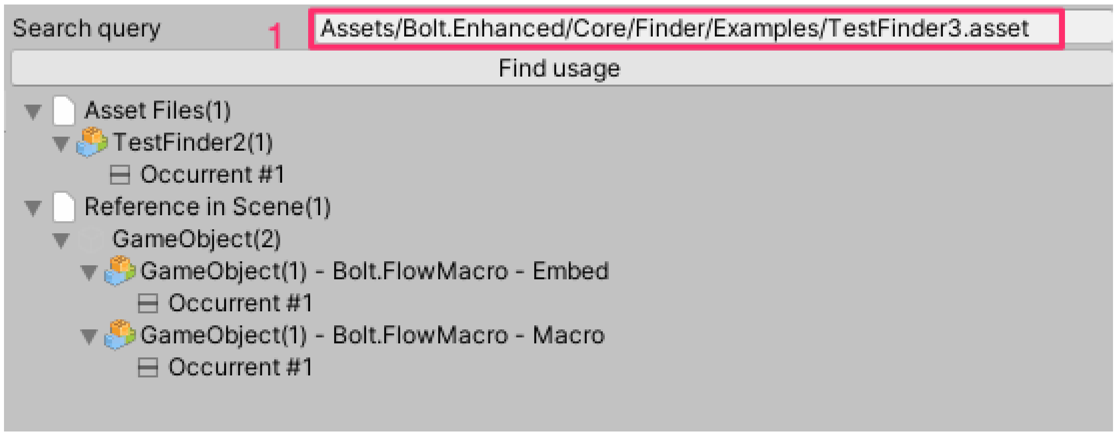

Search query
Assets/Bolt.Enhanced/Core/Finder/Examples/TestFinder3.asset
Find usage
Asset Files(1)
TestFinder2(1)
Occurrent #1
Reference in Scene(1)
GameObject(2)
GameObject(1) - Bolt.FlowMacro - Embed
Occurrent #1
GameObject(1) - Bolt.FlowMacro - Macro
Occurrent #1

**Step 3.** Let's try to click on one of the asset results to open it.



## Option 4. Finding the usage using the Search query inside the Bolt Enhanced - Find Usage window.

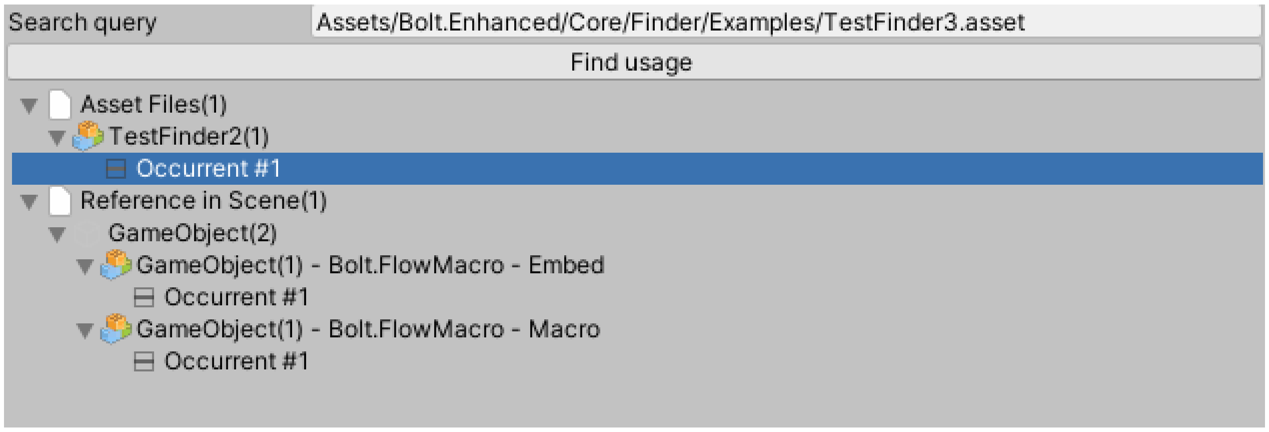
### Step 1.


Search query 1 Assets/Bolt.Enhanced/Core/Finder/Examples/TestFinder3.asset

Find usage

- Asset Files(1)
  - TestFinder2(1)
    - Occurrent #1
- Reference in Scene(1)
  - GameObject(2)
    - GameObject(1) - Bolt.FlowMacro - Embed
      - Occurrent #1
    - GameObject(1) - Bolt.FlowMacro - Macro
      - Occurrent #1

**Step 2.** Observe the result inside the **Bolt Enhanced - Find Usage** window, the list of all flow graphs or objects which use the **TestFinder3** flow is listed here.

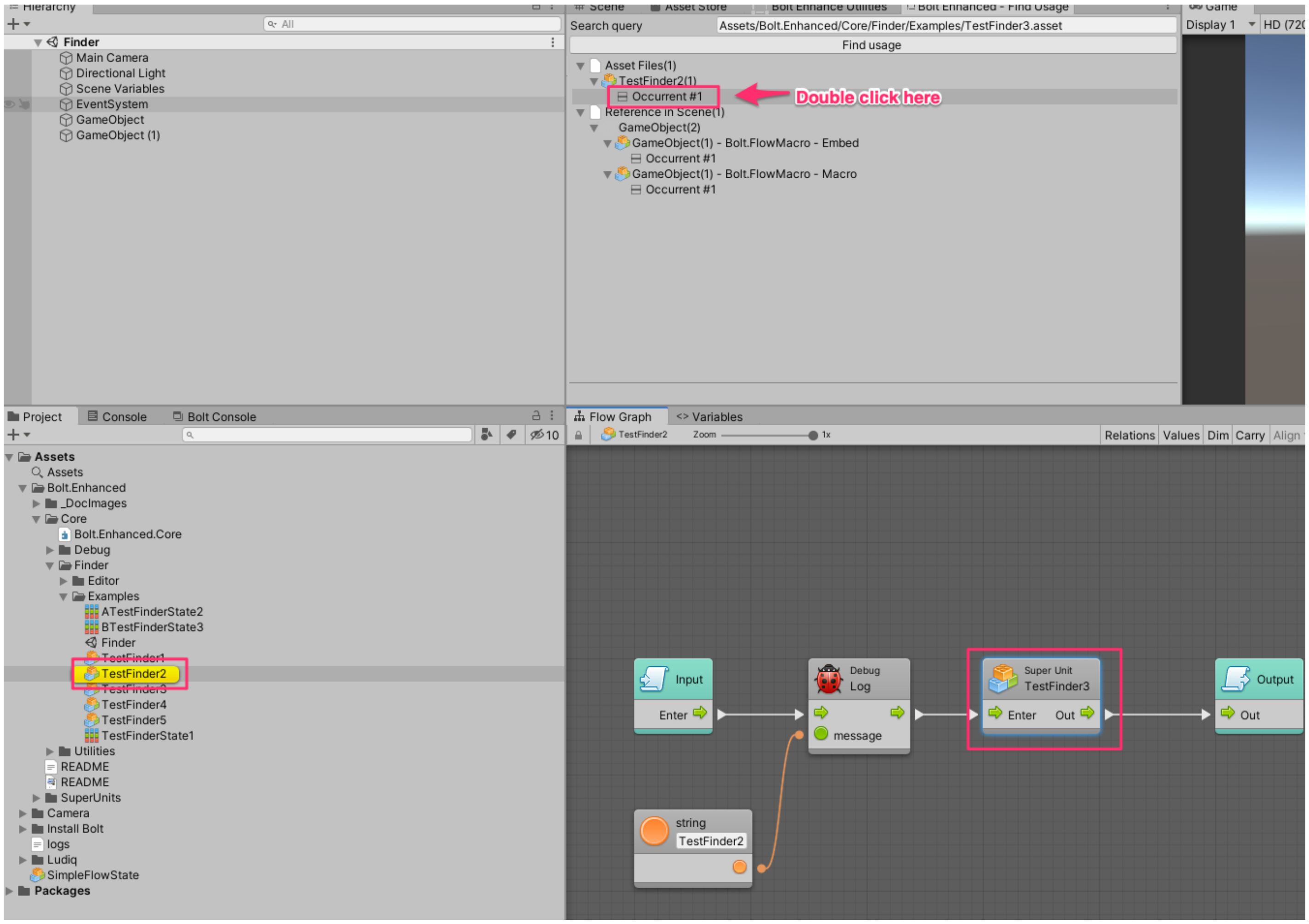


Search query Assets/Bolt.Enhanced/Core/Finder/Examples/TestFinder3.asset

Find usage

```
▼ Asset Files(1)
  ▼ TestFinder2(1)
      Occurrent #1
▼ Reference in Scene(1)
  ▼ GameObject(2)
    ▼ GameObject(1) - Bolt.FlowMacro - Embed
        Occurrent #1
    ▼ GameObject(1) - Bolt.FlowMacro - Macro
        Occurrent #1
```

**Step 3.** Let's try to click on one of the asset results to open it.



## Support

If you have any issues, questions, or want another example, please drop us a message:

- **Twitter**: https://twitter.com/yolostudiogame
- **Email**: contact@yolostudio.io
- **Website**: http://yolostudio.io
- **Discord**: https://discord.com/invite/gas3BMCRhE
- **Preview Feature playlist on Youtube**: https://www.youtube.com/playlist?list=PL--CMr7xgSMBgQppt6KoBNTp3yhu19un9