Programación de Videojuegos Parte II

Currículo de Aprendizaje

Desarrollado por IEEE y la Universidad del Norte

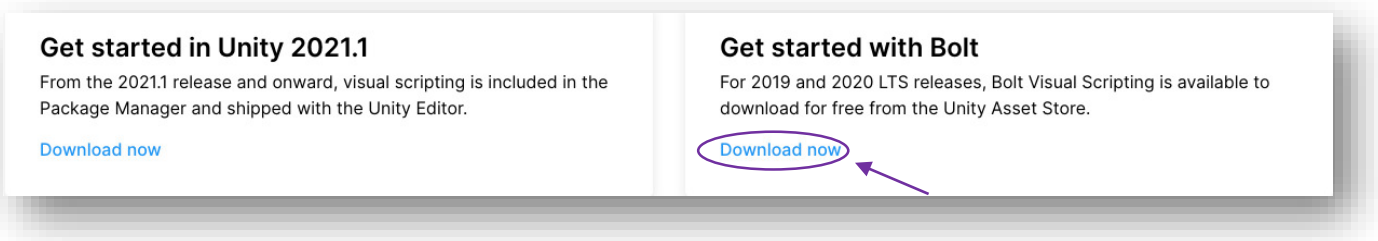
Tabla de Contenidos

Unity

Instalación de Bolt

1 Añadir el Asset a Unity: Para instalar Bolt, primero accedemos a https://unity.com/products/unity-visual-scripting

Buscamos más abajo lo siguiente:

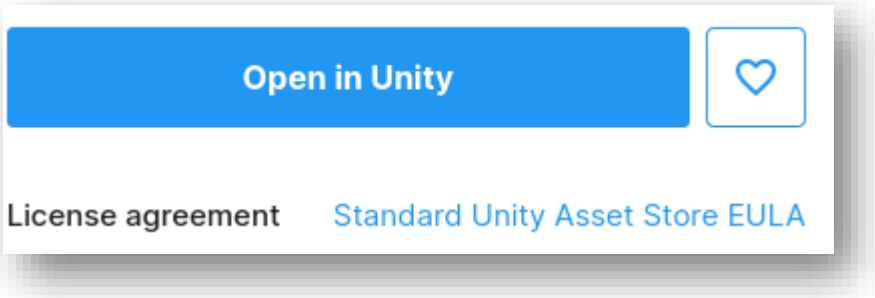


y realizamos clic en la parte señalada.

Esto requerirá iniciar sesión y aceptar distintos términos y condiciones si no se ha hecho antes.

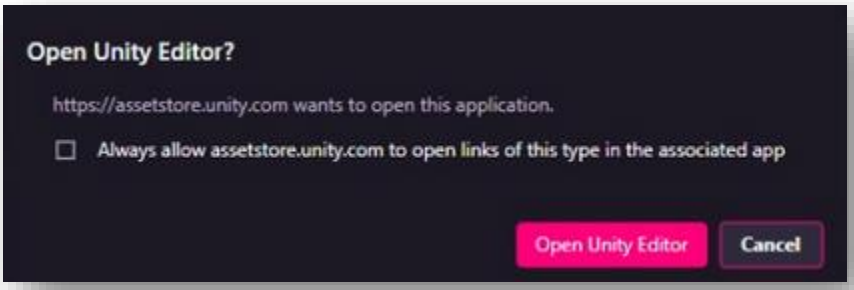
Después de eso, se llegará a la página del asset de Unity Bolt.

Hacemos click en el botón azúl de Add to My Assets y posteriormente cambiará a esto:



Al tener Unity instalado y un proyecto en mente, podremos instalar Bolt en él. No olvides iniciar sesión en Unity Hub con la misma cuenta.

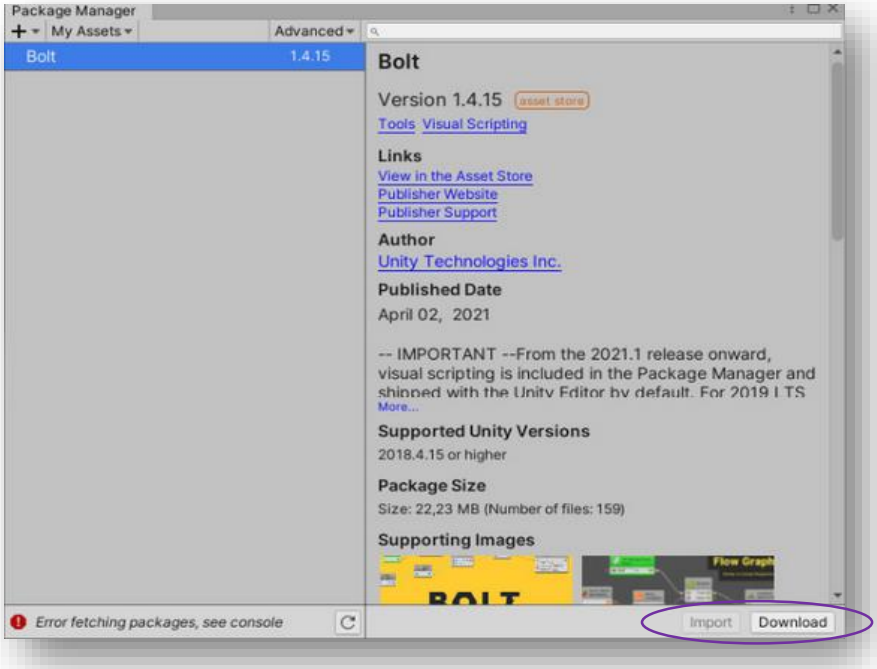
Presionamos el bo tón azúl y aparecerá una ventana en la que seleccionamos Abrir con el editor de Unity.

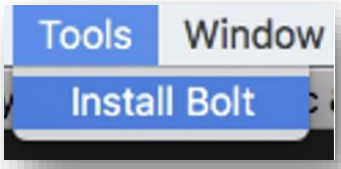


2 Usar Package Manager para importar Bolt: Para importar Bolt desde Package Manager, es necesario presionar Import y posteriormente Download, ubicados en la sección señalada.

Ahora aparecerá una pestaña llamada Tools entre Component y Window.

Desde ahí podremos importar Bolt.





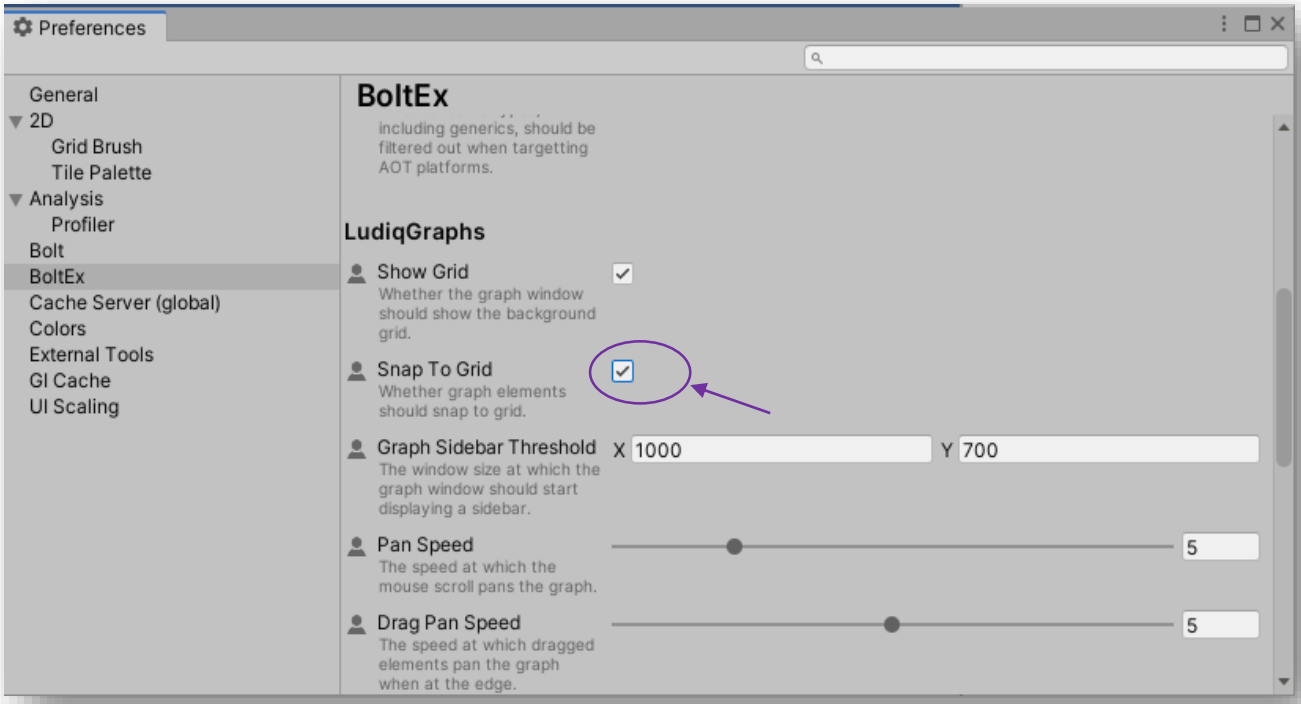
3 Instalación de Bolt: Al presionar Install Bolt, aparecerá la siguiente ventana:

Presionamos Next y tendremos la sección encargada de decidir el formato de los nombres de Bolt.

Aquí es recomendable elegir Human Naming debido a que es más amigable al usuario. En las Assembly options no hay que realizar cambios y proseguir hasta la última página en donde se debe escoger Generate.

Con esto, hemos concluido la instalación de Bolt.

Antes de seguir, es recomendable ir a Edit y posteriormente Preferences… en la Barra de Herramientas y habilitar Snap To Grid debido a que facilita la conexión de elementos del grafo.


Preferences
General
2D
Grid Brush
Tile Palette
Analysis
Profiler
Bolt
BoltEx
Cache Server (global)
Colors
External Tools
GI Cache
UI Scaling
BoltEx
including generics, should be filtered out when targetting AOT platforms.
LudiqGraphs
Show Grid
Whether the graph window should show the background grid.
Snap To Grid
Whether graph elements should snap to grid.
Graph Sidebar Threshold
The window size at which the graph window should start displaying a sidebar.
X 1000
Y 700
Pan Speed
The speed at which the mouse scroll pans the graph.
5
Drag Pan Speed
The speed at which dragged elements pan the graph when at the edge.
5

Conectar bloques

Set Up

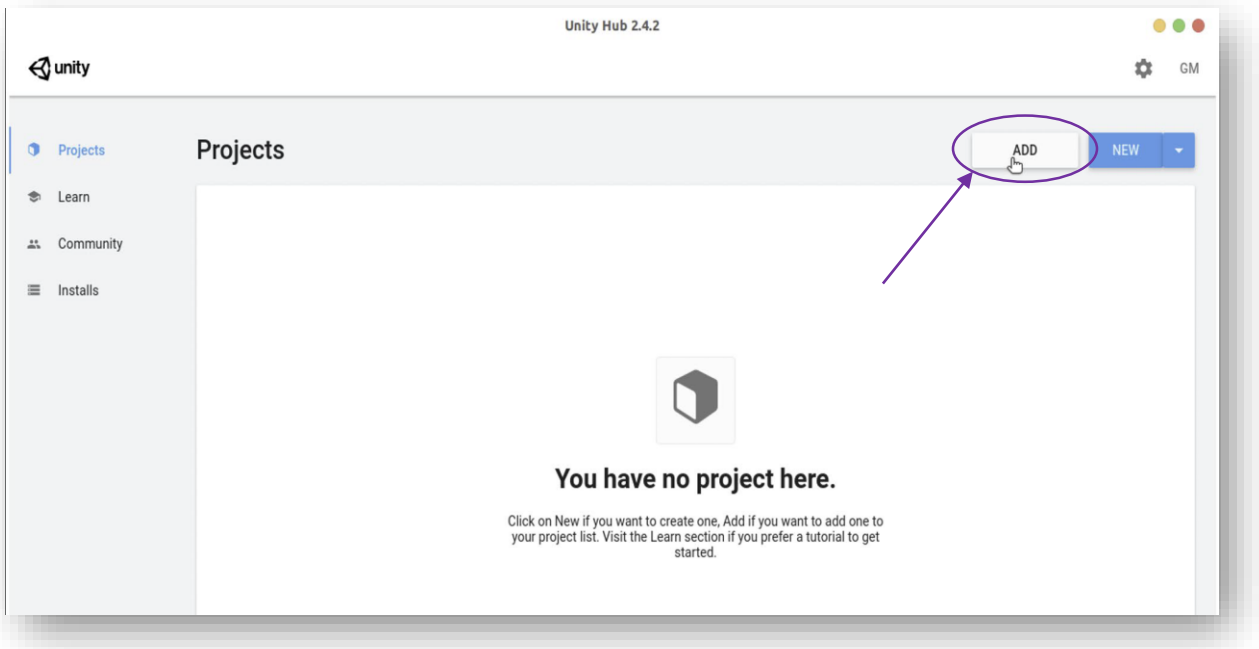
Requerimientos:

- Unity con Bolt instalado
- Archivo del proyecto de prueba descargado

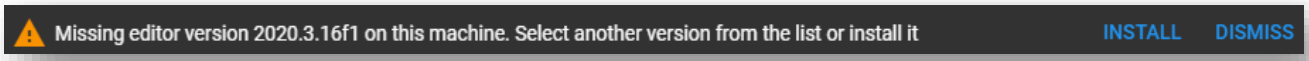
Desarrollo

A continuación, se explicará el funcionamiento del juego base y el uso de varios componentes de Bolt en Unity.

1. Abrir Unity Hub en el computador.
2. Abrir el proyecto nombre-del-proyecto, haciendo click en el botón ADD y posteriormente navegando a la carpeta de este. Recordar que no puede estar comprimido (.rar,.zip).

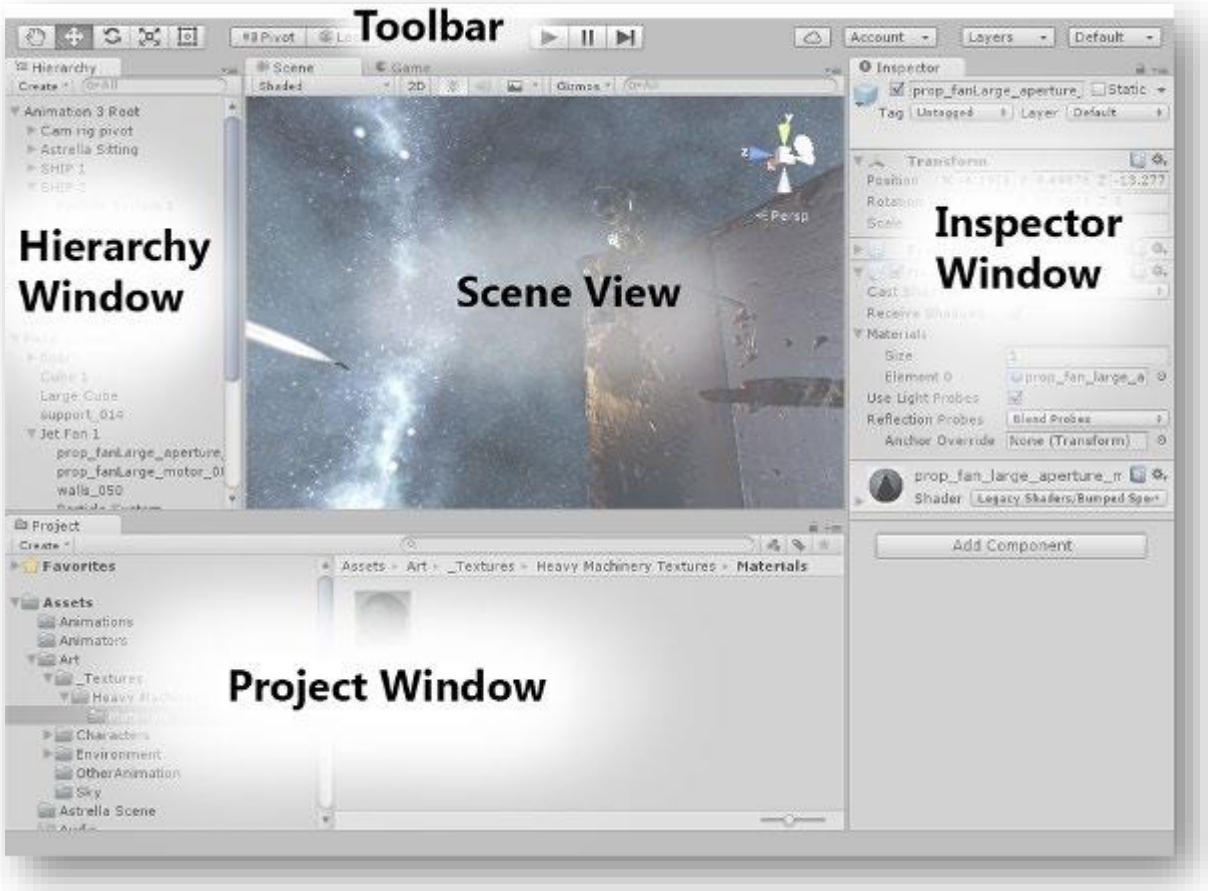


Después, hacer click en el proyecto de Unity. Si aparece un mensaje de este tipo, puedes presionar Install en esta y la siguiente opción para proceder, dejando todas las configuraciones por defecto en la ventana que aparece posteriormente.



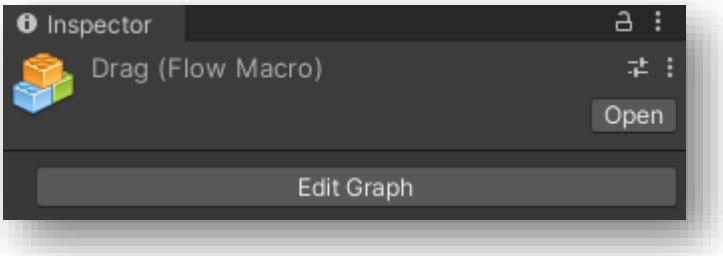
3 Una vez instalada la versión, volvemos a Projects en el menú izquierdo y hacemos click en el proyecto nombre-del-proyecto.

Demo: Conectar Bloques:

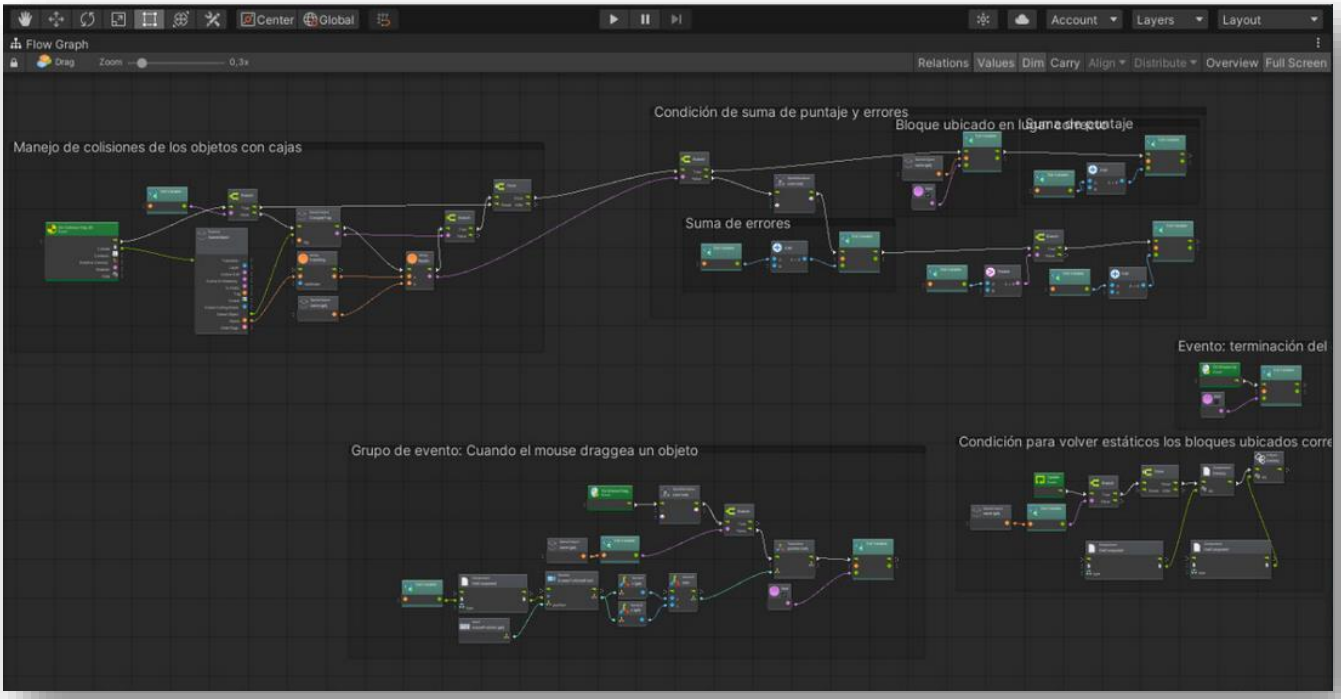


1 Abrir el grafo: Unity posee múltiples secciones en su interfaz, en Project Window abrir el archivo Drag ubicado en la sección de Assets.

Ahora será necesario presionar Edit Graph en la sección Inspector Window.



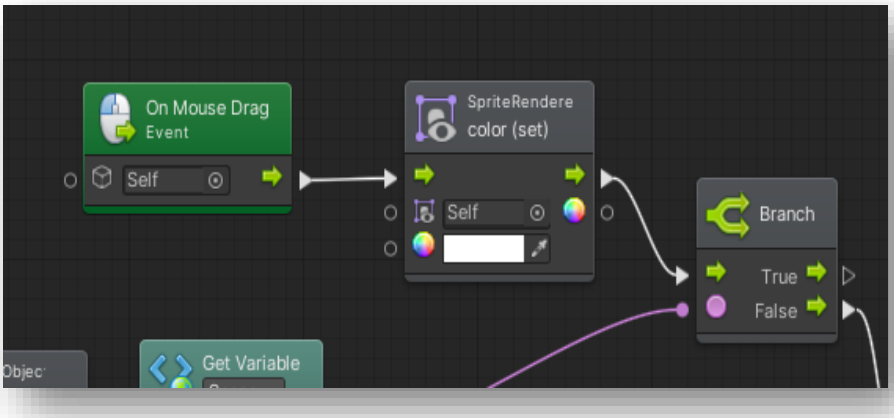
Ahora podremos visualizar el grafo del proyecto

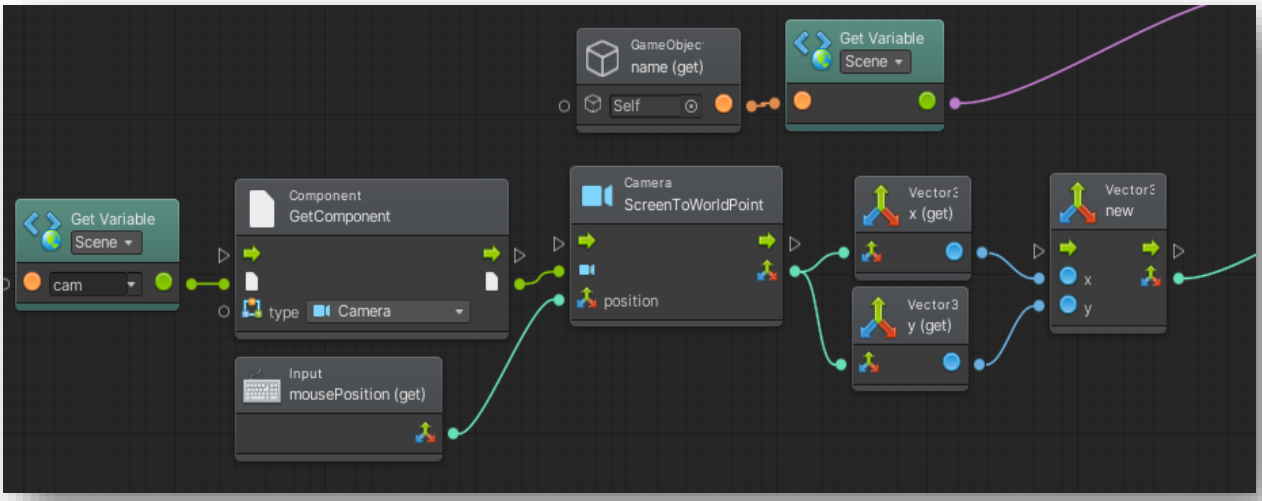


2 Grupos de eventos: Con el fin de observar los distintos componentes de este código, hablaremos de los distintos elementos del Flow Graph.

2.1 Cuando el mouse draggea un objeto:

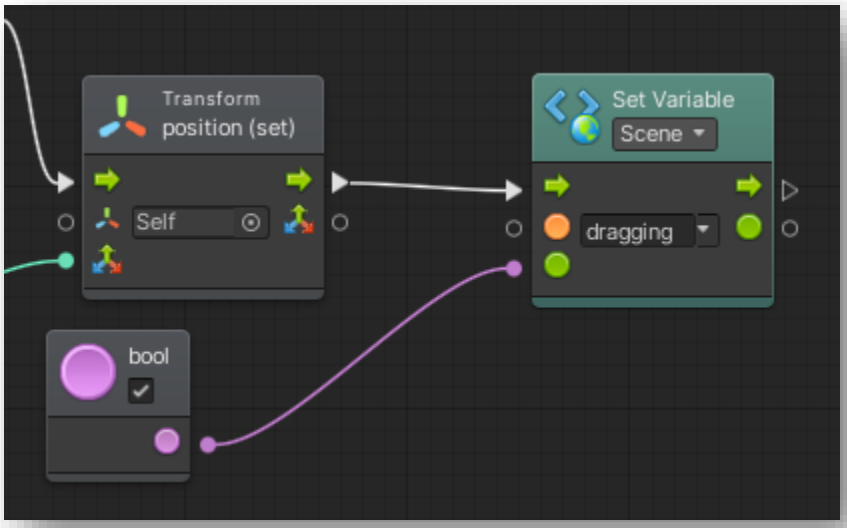
Esta sección del código comienza por un On Mouse Drag, por lo que se ejecuta cuando un usuario arrastra un bloque en el juego. El Branch, encontrado siguiendo la línea blanca intenta evitar que se genere el resto del código.





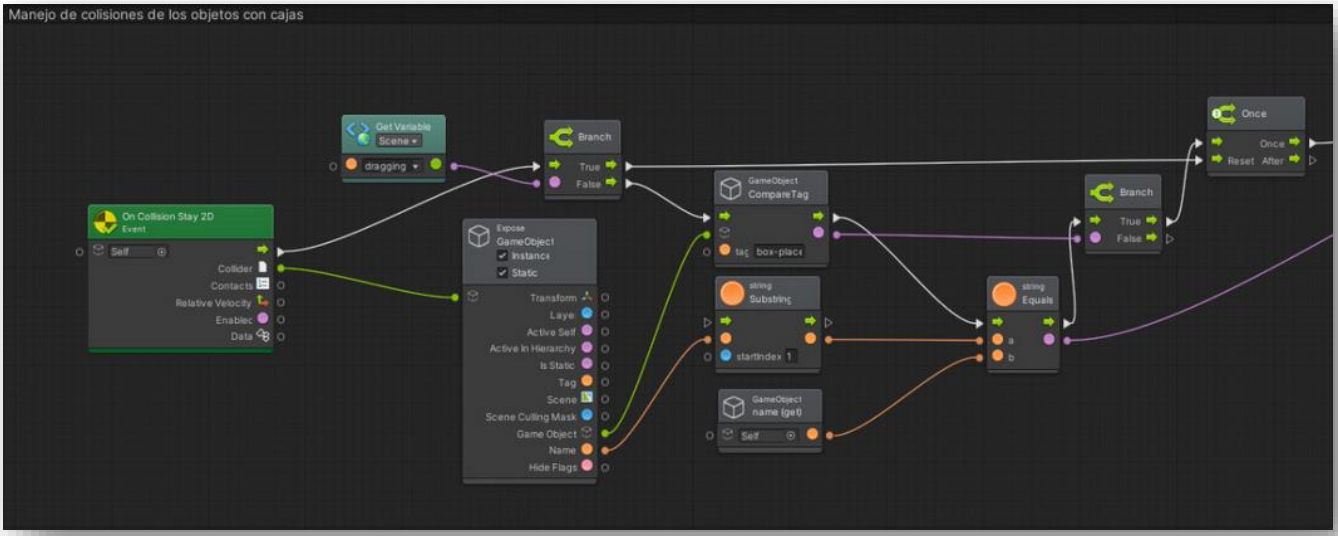
Observando el resto del código nos damos cuenta de que la condición del Branch depende de si es capaz de obtener un GameObject de la variable de escena. Esto con el fin de evitar continuar la ejecución si el usuario está manteniendo click presionado sin estar tomando un bloque en particular. Cada vez que se visualice una línea de color morado se tratará de un booleano, en otras palabras, solo comunicará falso o verdadero.

Siguiendo desde cam, se adquiere la cámara la posición actual del mouse y se guarda en un objeto de Vector 3 llamado new mediante x(get) y y(get) que se encargan de obtener las componentes X y Y del futuro vector.



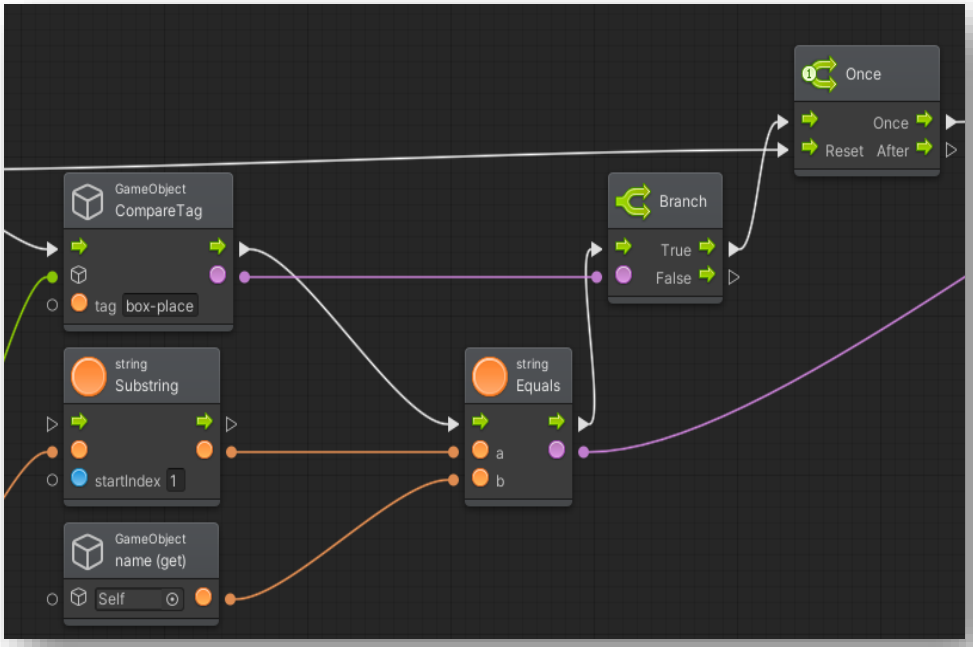
Con toda esta información, se define si el objeto está siendo arrastrado y su posición, además de ser guardado en la variable de escena dragging.

2.2 Manejo de colisiones de los objetos con cajas:



Se encarga de verificar el estado de los objetos en esta sección, activándose cuando se suelte un bloque encima de una caja.

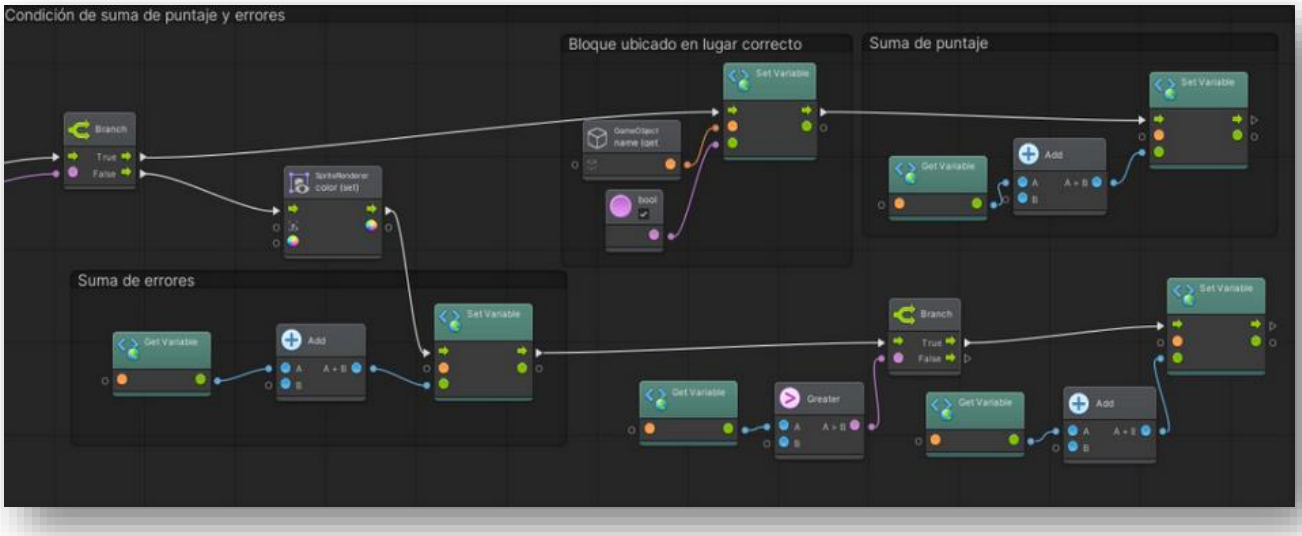
Siguiendo la flecha de color blanco se encuentra un Branch. Este es un bloque de control que se encarga de dividir el flujo en base al estado de la entrada. En este caso, True y False dependen del estado de dragging.

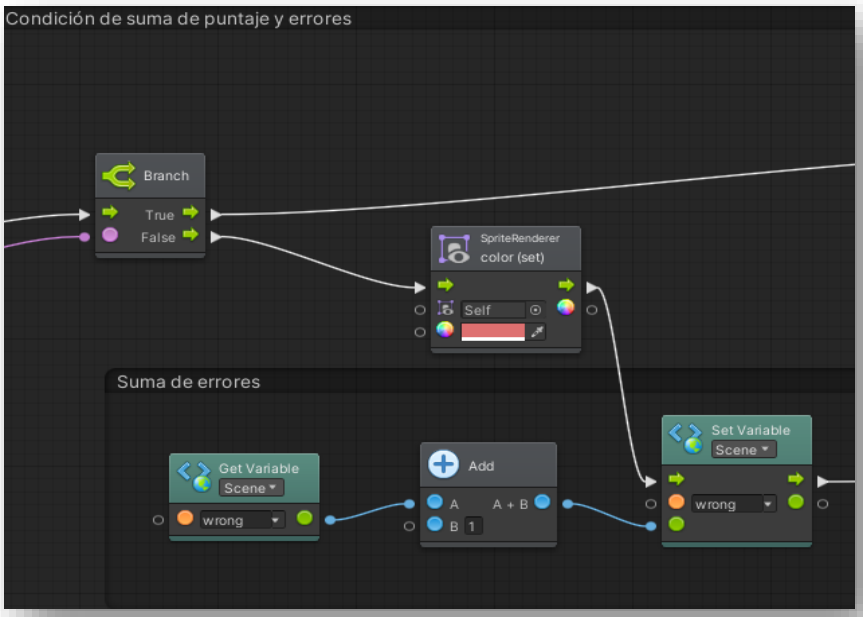


Esta sección se dedica a comparar el valor del objeto, en este caso el nombre de este.

La salida morada se encarga de enviar este resultado con el fin de generar la puntuación del usuario en la sección de Condición de suma de puntaje y errores.

2.3 Condición de suma de puntaje y errores:

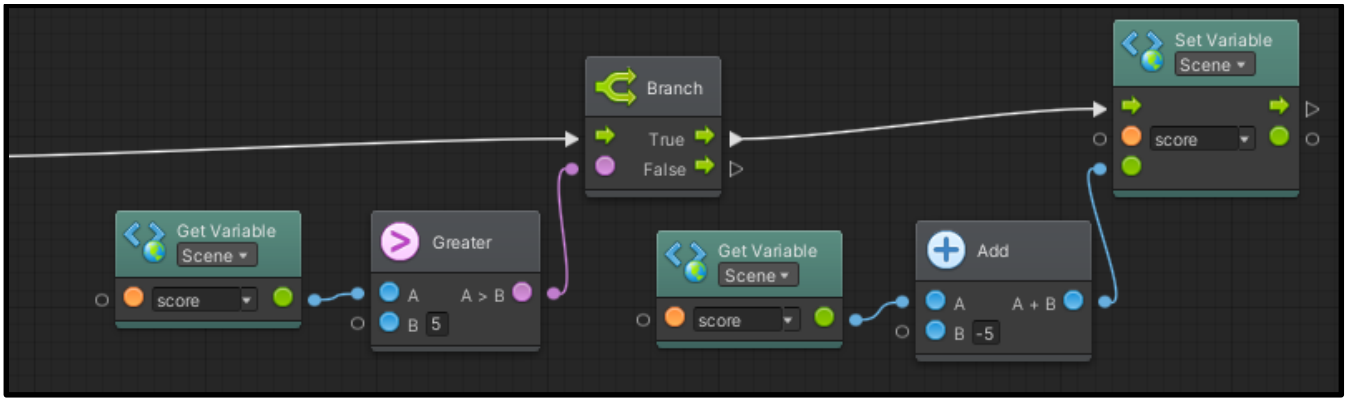




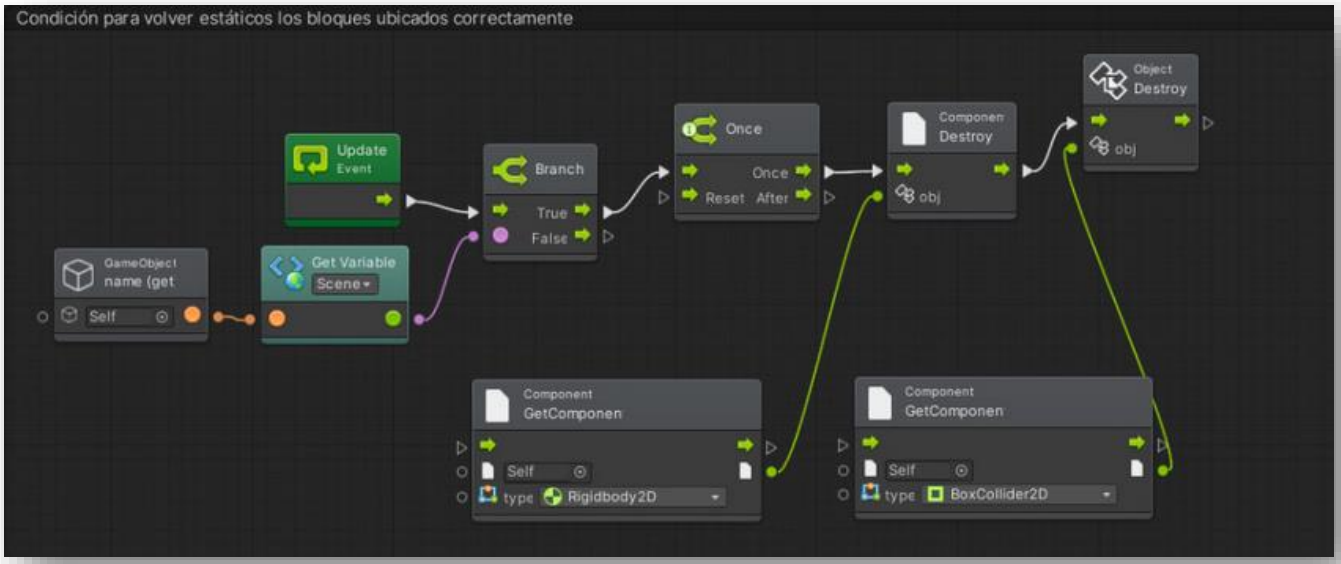
Desde Manejo de colisiones de los objetos con cajas, donde se verificó si el nombre de la caja era igual al del objeto, el recorrido de False sumará 1 al contador de errores wrong en la sección llamada Suma de errores.

Siguiendo por el camino de False tenemos que por cada error se le restará 5 al puntaje total a menos que el puntaje sea menor o igual a 5. Esto con el fin de evitar puntajes negativos.

Por otra parte, True seguirá a la siguiente sección encargada de asignar al contador de puntaje score el valor actual mas 10.

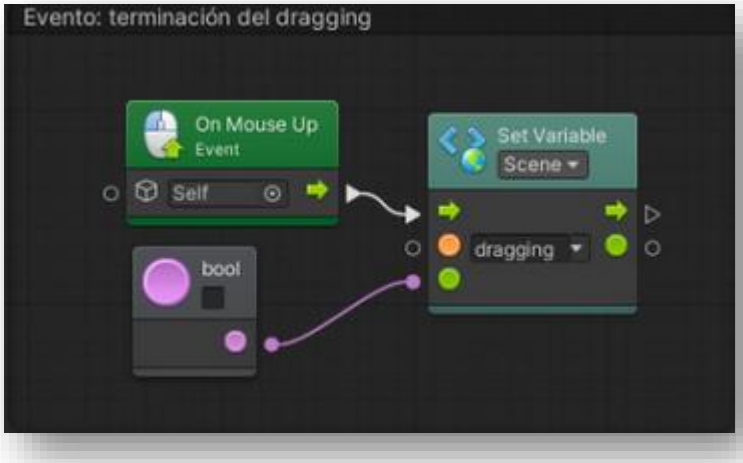


2.4 Condición para volver estáticos los bloques ubicados correctamente:



Component Destroy elimina los componentes RigidBody2D y BoxCollider2D con el fin de evitar que los bloques ubicados correctamente se muevan de esa posición.

2.5 Evento: Terminación del dragging:



Se encarga de asignar a dragging el valor de falso cuando se deje de presionar click. Importante para Manejo de colisiones de los objetos con cajas.