

Overview

Flow Graph is a visual scripting system that is embedded in the CryENGINE Sandbox Editor. The main advantage of the Flow Graph Editor is that users do not need to have any scripting or programming knowledge.

Simple and complex logic can be built with only a few clicks and without requiring any scripting or coding. A huge library of nodes provides the user with everything to fully control the entities and AI in a level.

In addition to being the main tool used for creating mission logic in single-player levels, the Flow Graph can also be used to prototype gameplay, effects, and sound design. Levels can have multiple graphs performing different tasks at the same time.

- [The Flow Graph Window](#)
- [Adding and Editing Nodes](#)
- [Creating and Editing Links](#)
- [Creating and Managing Flow Graphs](#)
- [Importing and Exporting Flow Graphs](#)
- [Flow Graph Module System](#)
- [Flow Graph Debugger](#)
- [Flow Graph Node Composition](#)
- [Flow Graph Tokens](#)
- [Flow Graph Node Reference](#)

Use Overview

In Flow Graph, each such node is a logical element (Flow Component) where each component has an arbitrary number of typed inputs and outputs, all inputs (except Events and any) contain default port values that can be modified with the Editor and as such input ports also serve as component properties.

When components receive a value or an event on a given input this input is said to be activated, and the Activate method of a component is called only once per step with potentially several activated input ports. This method then can check which ports were actually activated, retrieve the value of the input port and take any specific action.

A component can also arbitrarily Activate any of its Outputs, if this output is connected to the input port of another component, output event, or value it will immediately propagate the link and will activate the input port of another component.

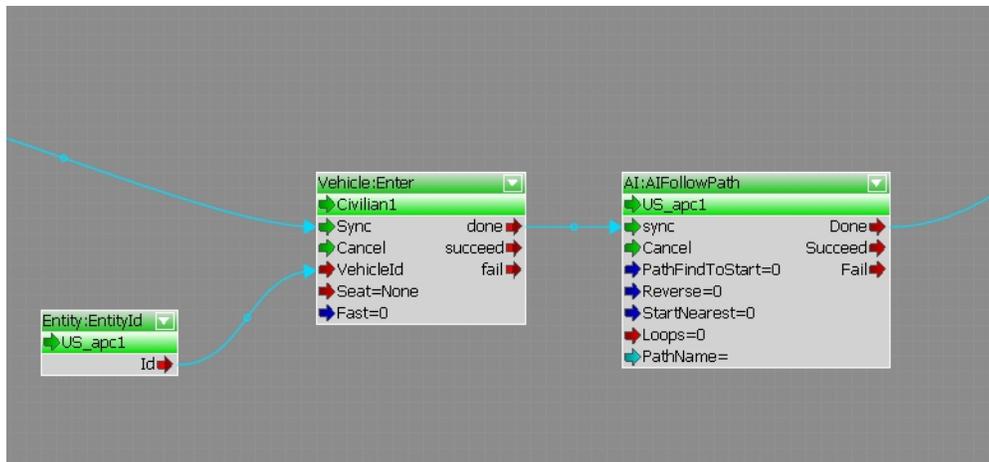
The output of any flow component can be connected to the input of another flow component with a directed edge. Outputs and Inputs are type'd, they can be: bools, floats, ints, strings, any type, or event type.

Type	Description
Any type	Is usually used to pass any value through flow component from input to output. (example: delay trigger will propagate whatever input it receive string or number to its output with delay without anyhow affecting this value).
Event type	Is a special type that does not contain any value and is only used to activate specific input of flow component, so that component can process input event.

When Output of one type connected to input of different type the reasonable type conversion on output value is applied when it propagates.

There's a set of Entity components one per each entity class type, that exposes entity functionality (its events) Such components can be added in editor from selected entities and they internally hold an EntityId of the entity they linked with.

The Flow Graph uses nodes to represent entities or behaviors that can be controlled by linking them to other nodes. Nodes can be interconnected by links:



The Flow Graph logic is stored in the XML format and can be easily exported for use in the other levels. As a graph is always created and stored on a specific entity, the graph is always exported along with the object. Layers are fully supported in the Flow Graph system.

Terminology

Term	Description
Graph	A single Flow Graph is referred to as a <i>graph</i> .
Nodes	Nodes are the representation of entities (Entity node) or components (Component node) that perform certain operations.
Component Node	A Component node is a node that does not represent an actual entity from the level, but performs a special action. Component nodes can have a target entity set, on which to operate.
Entity Node	Entity nodes represent the entities in the level. The input and output ports depend on the ports defined in the entity.
Links	Links connect nodes. They are visualized as lines drawn between the ports of connected nodes.
Ports	Nodes have input and output ports. These ports are used as a connection for links from other nodes. Ports are visualized as colored arrows on both sides of the node.
Graph Entity	An entity that contains a graph is referred to as a <i>graph entity</i> .