Overview

AI Sequencer is a system to simplify and reduce the level logic needed to manually control AI characters by level designers and it works on top of the current flowgraph system interface.

Goals

- Reduce the amount of flownodes necessary to script the idle actions by automating much of the micromanagement logic needed to manually control the character and handle all the different scenarios that can occur during gameplay (like for example, canceling the actions when the agent gets alerted or resuming the actions when he goes back to idle).
- Provide a context to the individual actions that an agent performs when they are manually controlled by level design. This simplifies the AI code and makes it more stable.
- Add a validation step to the level setup, minimizing the occurrence of bugs and providing better debug information when they happen.

Concept

An AI Sequence is a chain of actions to be performed by the AI entities. The *AISequence:Start* and *AISequence:End* define the beginning and the end of the sequence, while the AI sequence action nodes, like *AISequence:Move* or *AISequence:Animation* define the order of the actions to be executed. Only one sequence can be active for a individual agent and it can only perform one action at a time.

A simple example which will make the agent walk to the tag point 10, then go to the tag point 4 and play the animation AI_idleBreakBored and finally run to the tag point 3:
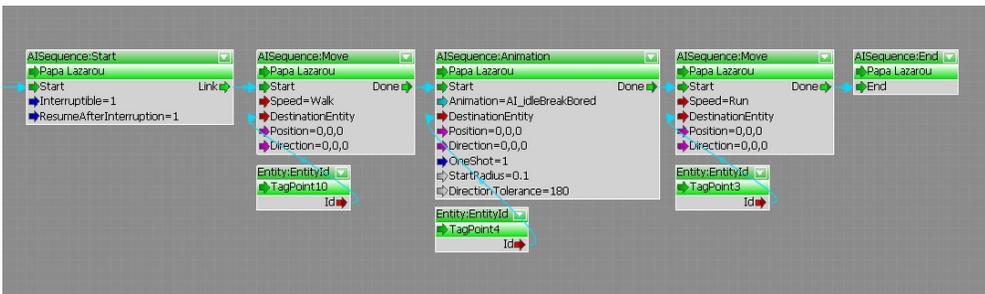
They can be looped as well, like in this example which will make the agent continuously to move between the tag point 10 and 3:
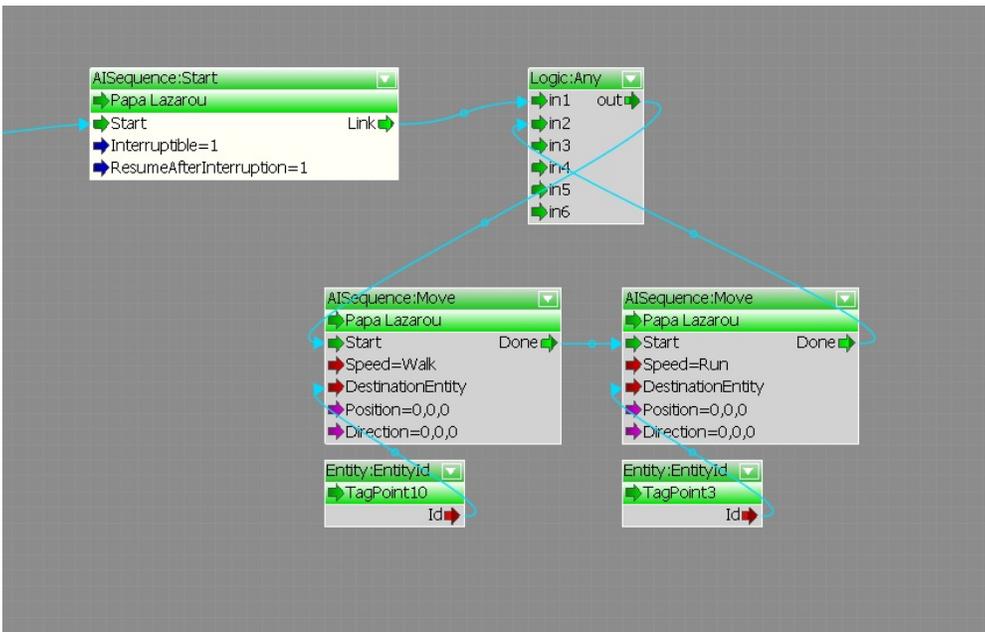
Sequences can also have conjoined sync points. In this example, both agent will start by doing a move action and will only continue to the next action when both have arrived at their first destinations:
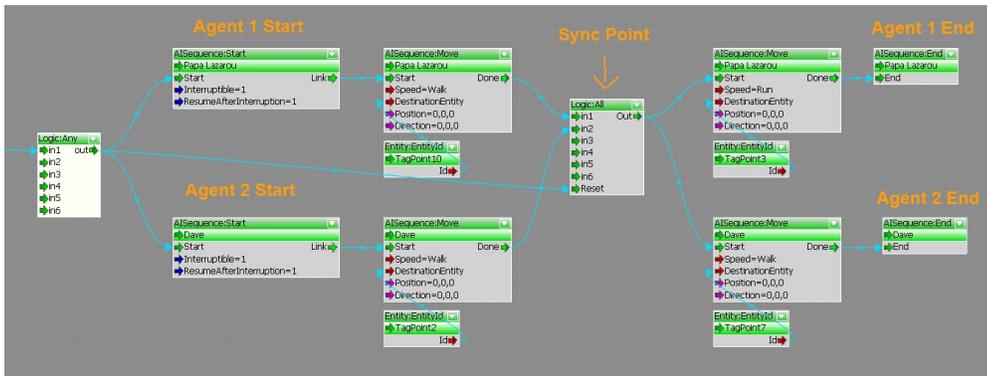
Interrupting / Resuming

With the interruptible flag set to true, the AI sequence will automatically stop executing as soon as AI entity leaves the idle state, taking care of stopping any action that could be active or triggered after that.

If it also set to resume after interruption, the sequence will also be resumed when the AI goes back to the idle state. It will resume from the start or from the latest bookmark. Bookmarks are defined with the AISequence:Bookmark flownode can be placed in the middle of a sequence and they will basically work as checkpoints from where the sequence should resume from.

In this example, if the agent is interrupted while moving to the second tag point and then goes back to idle, instead of starting from the beginning, it will resume the sequence by going directly again to the second tag point.

Valid Flownodes

To minimize possible bugs and unexpected behavior, only flownodes supported by the AI Sequence system can be used to define an AI sequence chain and if and there will be an error message when the AI sequence is triggered if an unsupported node is used.

The currently supported ones are:

- All AISequence flownodes.
- All Logic flownodes.
- All GameToken flownodes.
- Sound:PlayDialog.
- Sound:Dialog flownodes.
- Sound:PlaySoundEvent.
- Actor:AliveCheck.
- Debug:DisplayMessage.
- Debug:DisplayTag.
- Debug:DisplayTagAdv.
- Entity:BeamEntity.
- Entity:CheckDistance.
- Entity:EntitiesInRange.
- Entity:GetPos.
- Entity:Pos.
- Movement:MoveEntityTo.
- Movement:RotateEntity.
- Movement:RotateEntityEx.
- Physics:ActionImpulse.

If necessary, other flownodes can be added to list as long as they are compatible or changed to be compatible with the AI Sequence system.