Developer
/ Documentation
/ Unreal Engine
/ Unreal Engine 5.4 Documentation
/ Programming and Scripting
/ Blueprints Visual Scripting
/ Specialized Blueprint Node Groups
/ Timelines

Timelines

An Overview of Timelines in Unreal

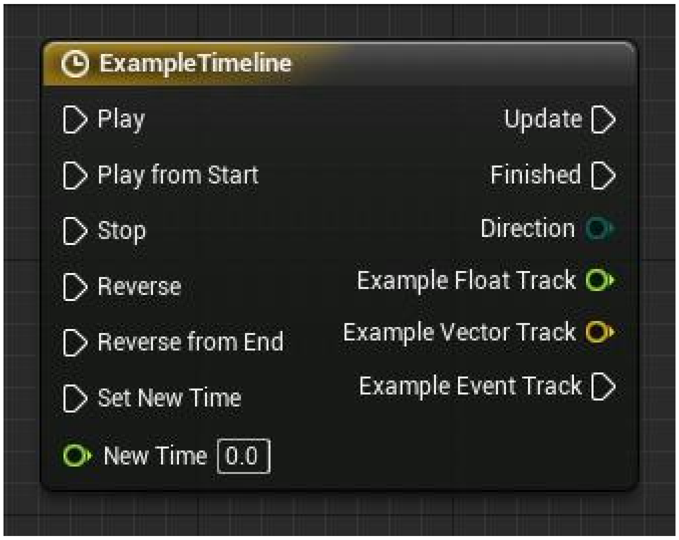
Choose your implementation method

Blueprints   C++

Timeline nodes are special nodes within Blueprints that provide time-based animation to be quickly designed and played back based on **events**, **floats**,**vectors**, or **colors** that can be triggered at keyframes along the timeline.

Timelines can be edited directly inside the Blueprint editor by **Double-clicking** on the Timeline in the Graph tab or in the My Blueprint tab. They are specifically built for handling simple, non-cinematic tasks such as opening doors, altering lights, or performing other time-centric manipulations to Actors within a scene.They are similar to level sequences as they both provide values such as floats, vectors, and colors to be interpolated between keyframes along the timeline.

Inputs and Outputs



Timelines come with the following input and output pins:

Input Pins

Item	Description
Play	Causes the Timeline to play forward from its current time.
Play from Start	Causes the Timeline to play forward from the beginning.
Stop	Freezes the playback of the Timeline at the current time.
Reverse	Plays the Timeline backwards from its current time.
Reverse from End	Plays the Timeline backwards starting from the end.
Set New Time	Sets the current time to the value set (or input) in the New Time input.
New Time	This data pin takes a float value representing time in seconds, to which the Timeline can jump when the Set New Time input is called.

Output Pins

Item	Description
Update	Outputs an execution signal as soon as the Timeline is called.
Finished	Outputs an execution signal when playback ends. This is not triggered by the Stop function.
Direction	Outputs enum data indicating the direction the Timeline is playing.

Timelines may have any number of additional output data pins reflecting the types of tracks created within them. These include Float, Vector, and Event tracks.

Overview (C++)

UTimelineComponent holds a series of **events**, **floats**, **vectors** or **colors** with their associated **keyframes**. They are inherited from UActorComponents

For further documentation, please see the overview about Actor Components

Timelines allow for time-based animation which are played from Events that can be triggered at keyframes along the timeline. They can be used to handle simple, non-cinematic tasks such as opening doors, altering lights, or performing other time-centric manipulations to Actors within a scene. They are similar to level sequences as they both provide values such as floats, vectors, and colors to be interpolated between keyframes along the timeline.

Inputs and Outputs

UTimelineComponents have robust methods that can be extended in native code, refer to the UTimelineComponent API reference for further documentation. If you would like to see examples on how to utilize Timeline Components in the engine, see one of the Timeline Example links in the section below.

ExampleTimeline.h

```
/** Start playback of timeline */
```

```
UFUNCTION(BlueprintCallable, Category="Components|Timeline")
ENGINE_API void Play();

/** Start playback of timeline from the start */
UFUNCTION(BlueprintCallable, Category="Components|Timeline")
ENGINE_API void PlayFromStart();

/** Start playback of timeline in reverse */
UFUNCTION(BlueprintCallable, Category="Components|Timeline")
ENGINE_API void Reverse();

/** Start playback of timeline in reverse from the end */
UFUNCTION(BlueprintCallable, Category="Components|Timeline")
ENGINE_API void ReverseFromEnd();

/** Stop playback of timeline */
UFUNCTION(BlueprintCallable, Category="Components|Timeline")
ENGINE_API void Stop();

/** Set the new playback position time to use */
UFUNCTION(BlueprintCallable, Category="Components|Timeline")
ENGINE_API void SetNewTime(float NewTime);

/** Signature of function to handle a timeline 'event' */
FOnTimelineEvent OnUpdate();
FOnTimeLineEvent OnFinished();
FOnTimeLineEvent ExampleTimelineEventTrack();

/** Struct that contains one entry for each float interpolation */
FTimelineFloatTrack ExampleFloatTrack;

/** Struct that contains one entry for each vector interpolation */
FTimelineVectorTrack ExampleVectorTrack;

/** Enum data indicating the direction the Timeline is playing */
ETimelineDirection Direction;
```

Copy full snippet

Timeline Examples

Fading Lights

An example use of a timeline in which we set up a fading light that also changes color.

Opening Doors

An example use of a timeline in which we set up a classic proximity-based opening door using Blueprints and C++.