

Smooth meth converter



BzSoft

Table of contents

[Table of contents](#)

[About](#)

[How to use it](#)

[SMConverter](#)

[Handlers](#)

About

The plugin allows you to smoothly convert between two completely different humanoid models.

How to use it

To convert from one model to another you have to write this code:

```
// create the converter
var converter = SMConverter.Blend(from, to);

// setup conversion time
converter.BlendTime = 2f;

// add some handlers to get good blending effect
converter.AddHandler(new SmcZeroPressHandler());
converter.AddHandler(new SmcBumpDiffMaterialHandler(texture));
```

from, to – is the humanoid model GameObject-s that it have to convert from to. This code adds converter component to one of this object and setup timer for blending (conversion).

Then you can apply some handlers to get more appealing effects. It could be your own handlers or the ones that goes with the plugin.

SMConverter

Main object for conversion. Have some common properties for all handlers and do handle handlers.

Methods:

- Blend - static method. Setup conversion and start the conversion loop. Accepts two parameters that is the game objects of the model that it converts from and to.

Properties:

- BlendTime - the whole blend time needed for conversion
- BlendRate - you can access this field in handlers to get current blend rate state

Handlers

Handler is used to extend converter logic. To create and add handler you have to create a class with ISmcHandler interface implementation and add it to converter using method AddHandler.

Interface:

```
public interface ISmcHandler
{
    /// <summary>
    /// Setup method. Executed ones at the start of conversion
    /// </summary>
    /// <param name="converter">Converter component</param>
    /// <param name="thisObject">Object that we are converting
    from</param>
    /// <param name="other">Object that we are converting to</param>
    void Setup(SMConverter converter, GameObject thisObject,
    GameObject other);
    /// <summary>
    /// Called each frame from regular Update method
    /// </summary>
    void Update();
    /// <summary>
    /// Called each frame from regular LateUpdate method
    /// </summary>
    void LateUpdate();
    /// <summary>
    /// Finish of conversion
    /// </summary>
    void Finished();
}
```

Also package comes with this handlers inside:

- SmcZeroPressHandler - bone position merging and model width controlling
- SmcCallbackHandler - calls method when the conversion is done
- SmcBumpDiffMaterialHandler - substitute materials for the models during conversion