

EB GUIDE tutorial

Using script curves for animations

Version 6.11.0.210526170413

Copyright © 2021 Elektrobit Automotive GmbH

Legal notice

Confidential and proprietary information

ALL RIGHTS RESERVED. No part of this publication may be copied in any form, by photocopy, microfilm, retrieval system, or by any other means now known or hereafter invented without the prior written permission of Elektrobit Automotive GmbH.

All brand names, trademarks and registered trademarks are property of their rightful owners and are used only for description.

This document contains links to websites owned and operated by third parties. Elektrobit has no control of the content of any third party website and therefore takes no warranty nor liability for damage or loss caused in connection with the use or reliance on any information, material, products or services contained or accessed through any such linked website.

1. Tutorial: Using script curves for animations

TIP**Default window layout**

All instructions and screenshots use the default window layout. If you want to follow the instructions, we recommend to set the EB GUIDE Studio or EB GUIDE Monitor window to default layout by selecting **Layout > Reset to default layout**.

Use a script curve when you want to define your own curve for an animation. Defining your own curve can be necessary when the other animation curves are not suitable or when you just want to define a custom curve. In this tutorial you are going to create a simple model with two script curves for two animations. You are going to create the following elements:

- ▶ A View state
- ▶ Two Rectangle widgets
- ▶ Two Animation widgets with script curves that animate the positions of the rectangle widgets

This results in a model with two rectangles. One rectangle moves down. The other moves to the side.

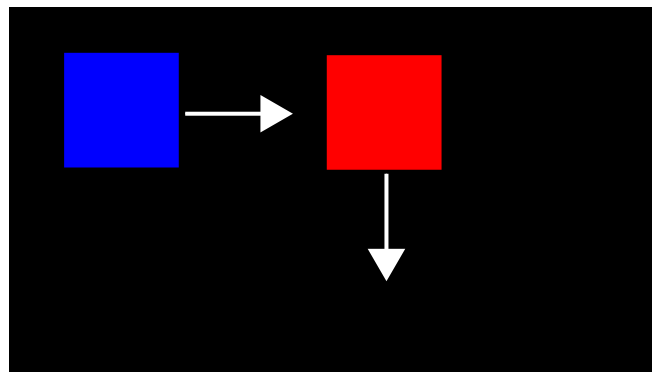


Figure 1. The rectangles with their movement direction

Approximate duration: 15 minutes.



Creating the first script curve

Prerequisite:

- The **Main** state machine contains an Initial state and a View state called `FirstState` and a View called `FirstView`.
- The Initial state has a transition to `FirstState`.
- The content area displays the `FirstView` View.

Step 1

From the **Toolbox** component, drag a Rectangle into the View and rename it to `BlueRectangle`

Step 2

In the **Properties** component, set the `fillColor` to blue.

Step 3

From the **Toolbox** component, drag an Animation into the view and rename it to `MoveAnimation`.

Step 4

In the **Datapool** component, add a datapool item of type `Float` and rename it to `xFloat`.

Step 5

In the **Navigation** component, select `BlueRectangle`.

Step 6

In the **Properties** component, go to the **User-defined properties** category, and click +.

A menu opens.

Step 7

In the menu, select **Conditional script**.

Conditional script 1 is added to the **User-defined properties**.

Step 8

Rename `Conditional script 1` to `StartBlueAnimation`.

Step 9

Next to `StartBlueAnimation`, click .

The EB GUIDE Script editor opens.

Step 10

Enter the following script:

```
function(v:arg0::bool)
{
    f:animation_play(v:this->^->"MoveAnimation")
}
```

Step 11

In the **Navigation** component, select `FirstView`.

Step 12

In the **Animation editor**, next to the **Animated properties** click + and select `FirstView`.

The **Animation properties** dialog opens.

Step 13

Under `BlueRectangle`, select the `x` property and then the `Script curve`

Step 14

Click **Accept**.

`Script curve 1` is added to the **Animation editor**.

Step 15

Rename Script curve 1 to BlueCurve.

Step 16

In the **Properties** component, next to the `curve` property click .

The EB GUIDE Script editor opens.

Step 17

Enter the following script:

```
function(v:diff::int, v:t_anim::int)
{
  dp:xFloat+=0.2
  f:floor(dp:xFloat*dp:xFloat)
}
```



Creating the second script curve

Prerequisite:

- You have finished the previous instruction.

Step 1

From the **Toolbox** component, drag a Rectangle into **FirstView** and rename it to RedRectangle.

Step 2

In the **Properties** component, set the `fillColor` to red.

Step 3

In the **Datapool** component, add a datapool item of type `Integer` and rename it to `1_diff`.

Step 4

Add another datapool item of type `Integer` and rename it to `2t_anim`.

Step 5

Select RedRectangle.

Step 6

In the **Properties** component, go to the **User-defined properties** category, click + and add a property of type conditional script.

Conditional script 2 is added.

Step 7

Rename Conditional script 2 to StartRedAnimation.

Step 8

Next to StartRedAnimation click .

The EB GUIDE Script editor opens.

Step 9

Enter the following script:

```
function(v:arg0::bool)
{
    f:animation_play(v:this->^->"MoveAnimation")
}
```

Step 10

In the **Navigation** component, select `FirstView`.

Step 11

In the **Animation editor**, next to the **Animated properties** click + and select `FirstView`.

The **Animation properties** dialog opens.

Step 12

Under `RedRectangle`, select the `y` property and then the `Script curve`.

Step 13

Click **Accept**.

`Script curve 2` is added to the **Animation editor**.

Step 14

Rename `Script curve 2` to `RedCurve`.

Step 15

In the **Properties** component, next to the `curve` property, click .

The EB GUIDE Script editor opens.

Step 16

Enter the following script:

```
function(v:diff::int, v:t_anim::int)
{
    dp:"1_diff"=v:diff
    dp:"2t_anim"=v:t_anim
    v:t_anim/2::int
}
```



Saving and testing the EB GUIDE model

Prerequisite:

- You completed the previous instruction.

Step 1

To save the project, click  in the command area.

Step 2

To start the simulation, click  in the command area.

The animation is played at the start of the simulation.