

PlayStandAnim.lua

What does this script do?

This script will play animations that are defined to use animatedVehicle specialization.
It works in MP as well as rotate help message when animation is at 1 or -1.

Specializations needed

- animatedVehicle
- PlayStandAnim

Code that needs to be placed in .xml file (not moddesc.xml)

Copy **green** part and add it between the **red** marked code if you want to add more parts.

```
<standardAnimations>
  <part name="animationName" inputName="BUTTONNAME" helpTextNamePos="110n_help_text_open"
  helpTextNameNeg="110n_help_text_close" />
</standardAnimations>
```

helpTextNamePos and helpTextNameNeg

Help messages are defined in the moddesc.xml file and should look something like this

```
<text name="110n_help_text_open">
  <en>English text</en>
  <de>German text</de>
</text>
```

No help message.

```
<part name="animationName" inputName="BUTTONNAME" />
```

Use only one help message.

```
<part name="animationName" inputName="BUTTONNAME" helpTextNamePos="110n_help_text_open" />
```

or

```
<part name="animationName" inputName="BUTTONNAME" helpTextNameNeg="110n_help_text_close" />
```

Use two help messages that rotates based on animations direction.

```
<part name="animationName" inputName="BUTTONNAME" helpTextNamePos="110n_help_text_open"
helpTextNameNeg="110n_help_text_close" />
```

inputName

inputName reads buttons from moddesc.xml file and must have the same name in both .xml files.

```
<input name="BUTTONNAME" key1="KEY_b" button="BUTTON_2" />
```

```
key1="KEY_b"
```

to change button replace the letter b in "KEY_b"

```
Button="BUTTON_2"
```

to change button replace the number 2 in "BUTTON_2"

Note: this is for game pads and in other words not needed but there are users that use game pads and don't want to use both keyboard and game pad so I recommend you to add support for it.

common used are

"BUTTON_2" - (b)

"BUTTON_6" - (x)

"BUTTON_5 BUTTON_6" - (o)

Code that needs to be placed in moddesc.xml

Add green part between red part and if you dont have it then add it.

```
<specializations>
  <specialization name="PlayAnimations" className="PlayStandAnim" filename="Scripts/PlayStandAnim.lua" />
</specializations>
```

Same as above but don't replace red part.

```
<vehicleTypes>
  <type name="trailer" className="Vehicle" filename="$dataS/scripts/vehicles/Vehicle.lua">
    <specialization name="animatedVehicle" />
    <specialization name="PlayAnimations" />
  </type>
</vehicleTypes>
```

Note: PlayStandAnim.lua needs to be placed in a folder with the name **Scripts**

How to make an animation with animatedVehicle

To make an animation you need to have these lines in you .xml file.

green is the first animation and if you want more just animation just copy that and place it between red parts.

```
<animations>
  <animation name="openDoor">
    <part node="0" startTime="0" duration="4" startRot="0 0 0" endRot="0 0 -120"/>
  </animation>
</animations>
```

part node="0" is the node you want to animated

startTime="0" if you want to delay the start of the animation

duration="4" tells how long the animation is going to be

the next part is not needed in the line if you dont use it but you must use code 1 or 2 (both works to).

code 1 - Rotation

startRot="0 0 0" - where the animation starts and this value needs to be changed to the current X,Y,Z value that the current node have

endRot="0 0 -120" - where the animation ends also same as above

code 2 - Translation

startTrans="0 0 0" - same as startRot

endTrans="0 0 20" - same as endRot

If you want more animation parts that you want triggered at the same time then copy

```
<part node="0" startTime="0" duration="4" startRot="0 0 0" endRot="0 0 -120"/>
```

and add it under the same line

Example: You got 2 signs that you want to rotate at the same time.

Example 2: How a line looks like when using code 1 and code 2 at the same time

```
<part node="0" startTime="0" duration="4" startRot="0 0 0" endRot="0 0 -120" startTrans="0 0 0" endTrans="0 0 20"/>
```

(Addition) How to move collision with the animation

Specializations needed

- cylindered

Add **green** as below.

```
<vehicleTypes>
  <type name="trailer" className="Vehicle" filename="$dataS/scripts/vehicles/Vehicle.lua">
    <specialization name="cylindered" />
    <specialization name="animatedVehicle" />
    <specialization name="PlayAnimations" />
  </type>
</vehicleTypes>
```

Note: All custom .lua scripts MUST be placed after the last standard specialization, in this case its the **blue** part.

You need to make the collision you want moved a new compound in both GE and script

Script

look for a similar line in the .xml file

```
<components count="1">
  <component1 centerOfMass="0 0 0" solverIterationCount="20" />
</components>
and change to something like this (this adds 1 new compound)
<components count="2">
  <component1 centerOfMass="0 0 0" solverIterationCount="20" />
  <component2 centerOfMass="0 0 0" solverIterationCount="20" />
  <joint component1="0" component2="1" index="0>0|0" rotLimit="0 0 0" transLimit="0 0 0" />
</components>
```

also notice the

```
<joint component1="0" component2="1" index="0>0|0" rotLimit="0 0 0" transLimit="0 0 0" />
```

this tells what component to join (`component1="0" component2="1"`)

as well as where the transform group is (`index="0>0|0"`) (this is where the component will rotate/translate around.

`rotLimit="0 0 0"` and `transLimit="0 0 0"` should be set as its written.

This connect the animated node with the component you want moved with animation.

```
<movingTools>
  <movingTool index="0>0" componentJointIndex="0" anchorActor="0"></movingTool>
</movingTools>
```

`index="0>0"` - this is the node you are animating

`componentJointIndex="0"` - look above for `<joint` and start counting, starting with 0 and then change the value "0" to what you counted to.

Want more "connecters" copy the `<movingTool....` line and add under the line you just copy ed.

Side Note:

If the components collision is over lapping each other then you will get some weird movements on the mod and the way to fix it is to use this

```
<collisionPair component1="0" component2="1" enabled="false"/>
```

Gaints Editor

Check the example mod to see how that's made.

(Addition) Want to check a value from one of the parts

You need to know some basic .lua scripting before attempting this section.

To check for example if the first animation part is finished then you can use something like this

`self.PAs[1].aFinished`

1 - represent the first part, start count from 1.

`aFinished` - can be used to tell if the animation is done or not.

want to set a specif part then you can write something like this

`self:setPlayAnim(3, false);`

3 - represent the third part, start count from 1.

`false` - tells what direction to play, false and true is the only values that can be used.

By using the next script below it will change from true and false it self without you needing to make 2 seperated functions for it.

`self:setPlayAnim(3, not self.PAs[3].aFinished);`

Last thing

That's all I have for you and don't upload the example mod without this file as it don't have all the features as the original its only in this zip file to show how to set the script up that's the purpose with it.

If you got any problems with it then use the support topic on <http://www.ls-uk.info>

Manual and script writer:

Xentro

Credits for the example mod is following:

Modell: PowerGlenn

Textur: solanz

Ingame: solanz

Scripts: Xentro