

Ploughing Mod

version 1.0 beta



by Burner

Ploughing mod beta

Author: Burner

Version: 1.0 beta

Date: 21.05.2012

Table of Contents:

1. General informations
2. Modifications in *moddesc.xml*/file
3. Modifications in *i3d* file
4. Modifications in vehicle *xm*/file

1. General informations

Mod features:

- simulation of furrow through the collapse of the wheels in places where furrow should be, without interference in the terrain,
- simulate jumpy terrain while driving on a ploughed area,
- increasing engine speed during ploughing,
- additional exhaust particle system which appears during ploughing.

Information regarding *xm*/code included in manual

Fragments of the gray code may already be in your file. In this case, please avoid them and only use the black code. Green text indicates comments.

2. Modifications in *moddesc.xml*/file

At the beginning of the installation, add a new specialization:

```
<specializations>
  <specialization name="PloughingSpec" className="PloughingSpec" filename="PloughingSpec.lua"/>
</specializations>
```

The next step is adding a new type of vehicle. The following example shows a new type of vehicle on the example of the tractor:

```
<vehicleTypes>
  <type name="X720PloughingSpec" className="Vehicle" filename="$dataS/scripts/vehicles/Vehicle.lua">
    <specialization name="PloughingSpec" />
    <specialization name="motorized" />
    <specialization name="steerable" />
    <specialization name="hirable" />
    <specialization name="aiTractor" />
    <specialization name="honk" />
  </type>
</vehicleTypes>
```

The name of the newly created type of vehicle should be given also in the vehicle *xm*/file. Rename schema in *xm*/file was presented in **Modifications in vehicle *xm*/file** section.

If your mod has already declared a new vehicle type *<vehicleTypes>*, you just add the newly created specialization *PloughingSpec*.

IMPORTANT! New specialization should be on top of the list. Otherwise, if the vehicle has elements based on hydraulics script, such as a cylinder or steering rack, there may be problems with the proper work of these elements.

3. Modifications in *i3d* file

General structure of elements in *i3d* file:

Main component

- main vehicle elements

Front right wheel component

- front right wheel joint index
- front right wheel
- additional wheel collision (*optional*)

Front left wheel component

- front left wheel joint index
- front left wheel
- additional wheel collision (*optional*)

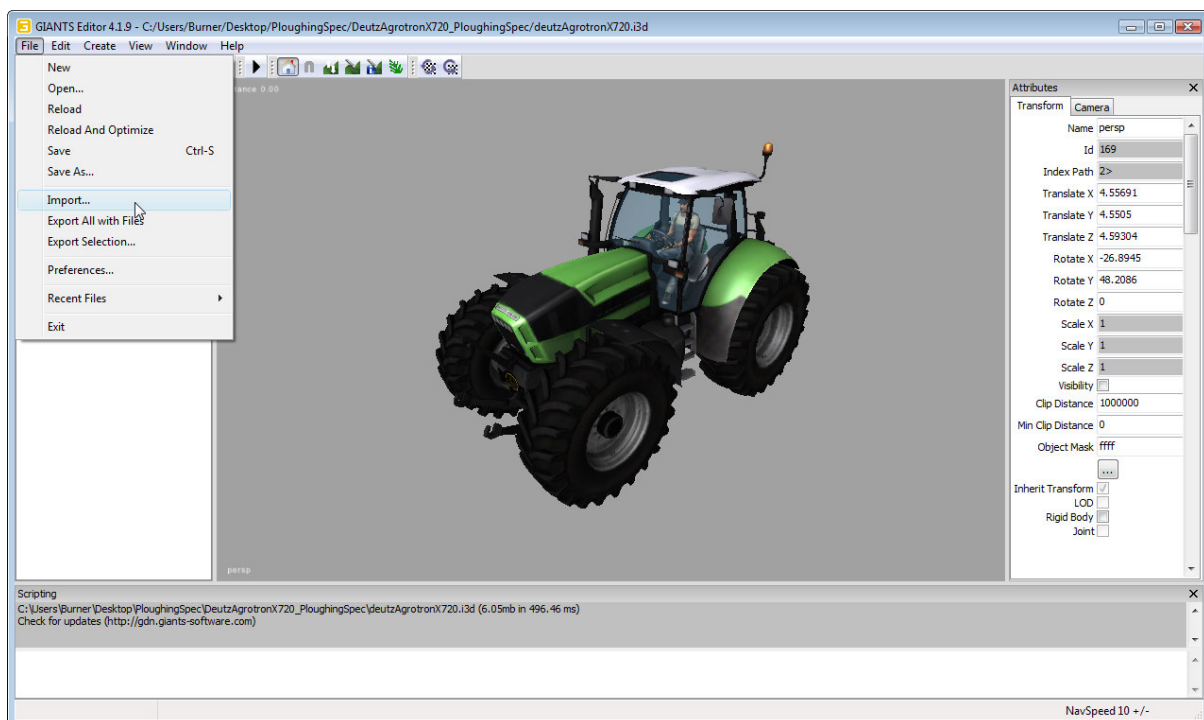
Back right wheel component

- back right wheel joint index
- back right wheel
- additional wheel collision (*optional*)

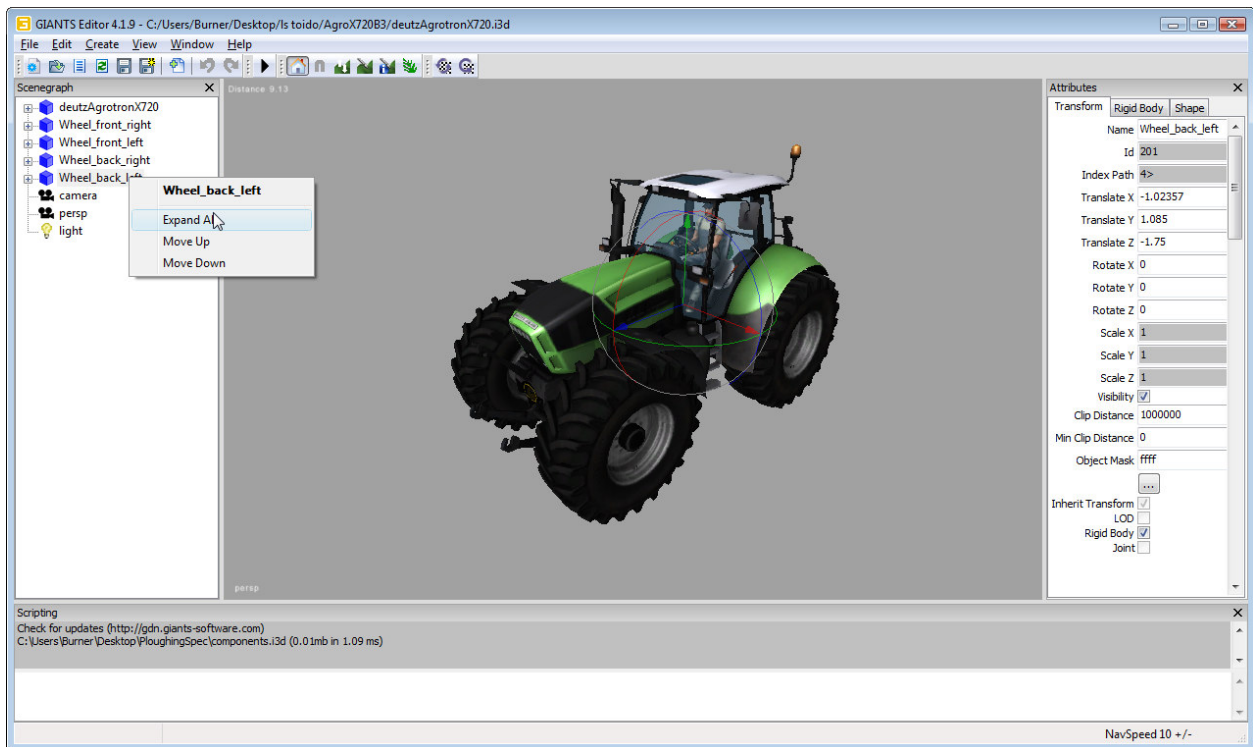
Back left wheel component

- back left wheel joint index
- back left wheel
- additional wheel collision (*optional*)

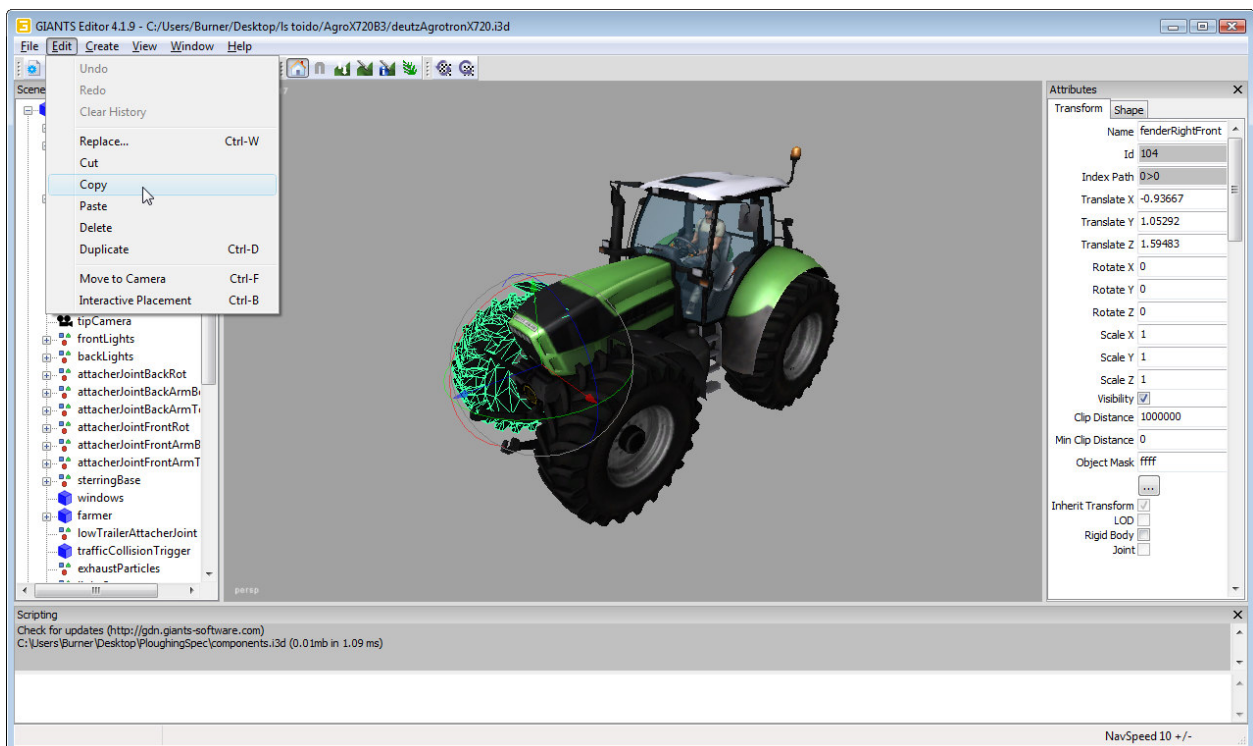
Open *i3d* file of vehicle using **Giants Editor** and import *components.i3d* file using *File -> Import ...* option or drag the file icon in the editor window.



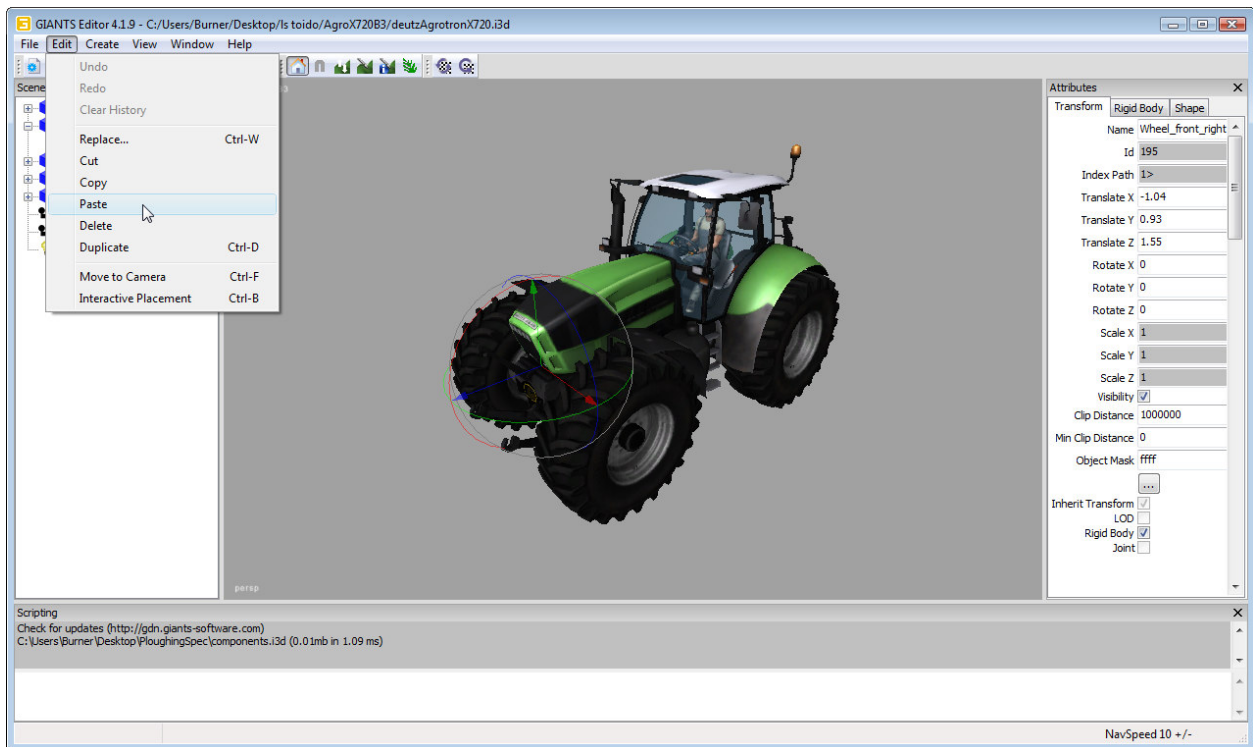
Order the components so that the newly imported items were located just below the main component and in the order in which the wheels are now arranged in the machine. This will facilitate typing indexes in *xm/file*. To move component, click the right mouse button and select *Move Up* or *Move Down*.



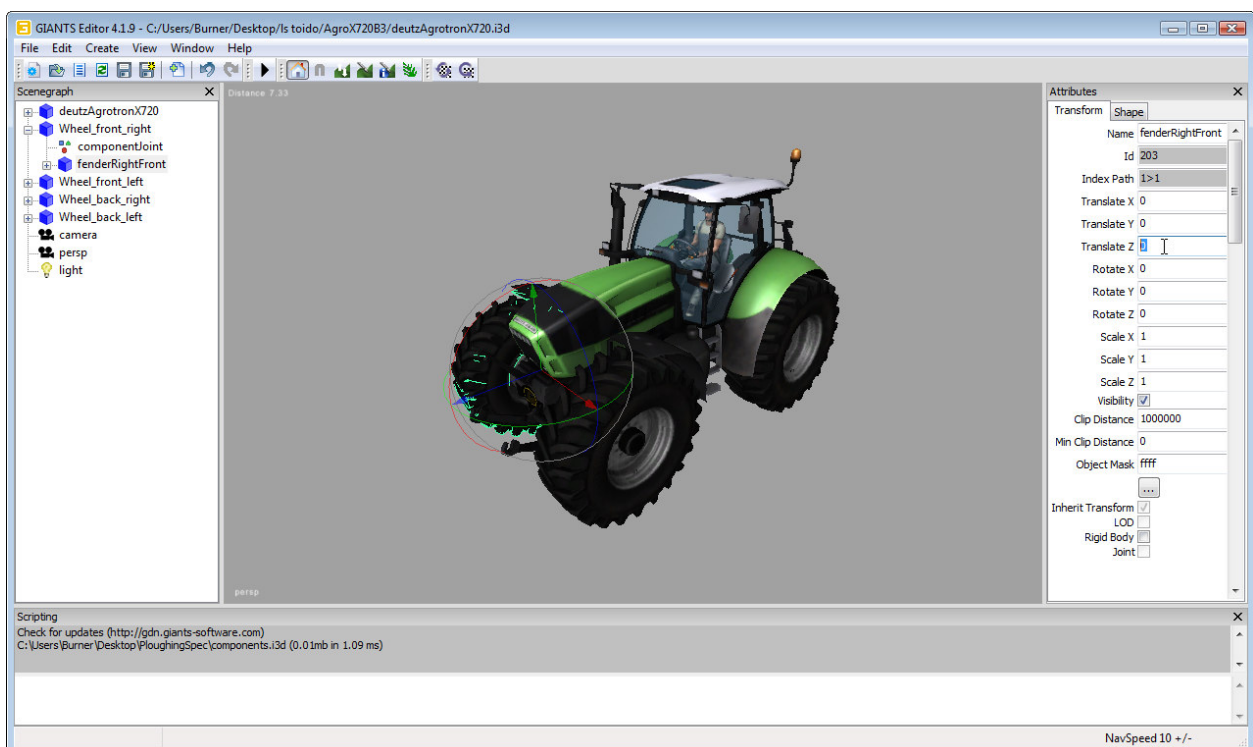
Expand the branch with main elements of component, select first wheel and copy it to clipboard (*Edit -> Copy*).



Select first of the newly added wheel component and paste copied wheel to it (*Edit -> Paste*).

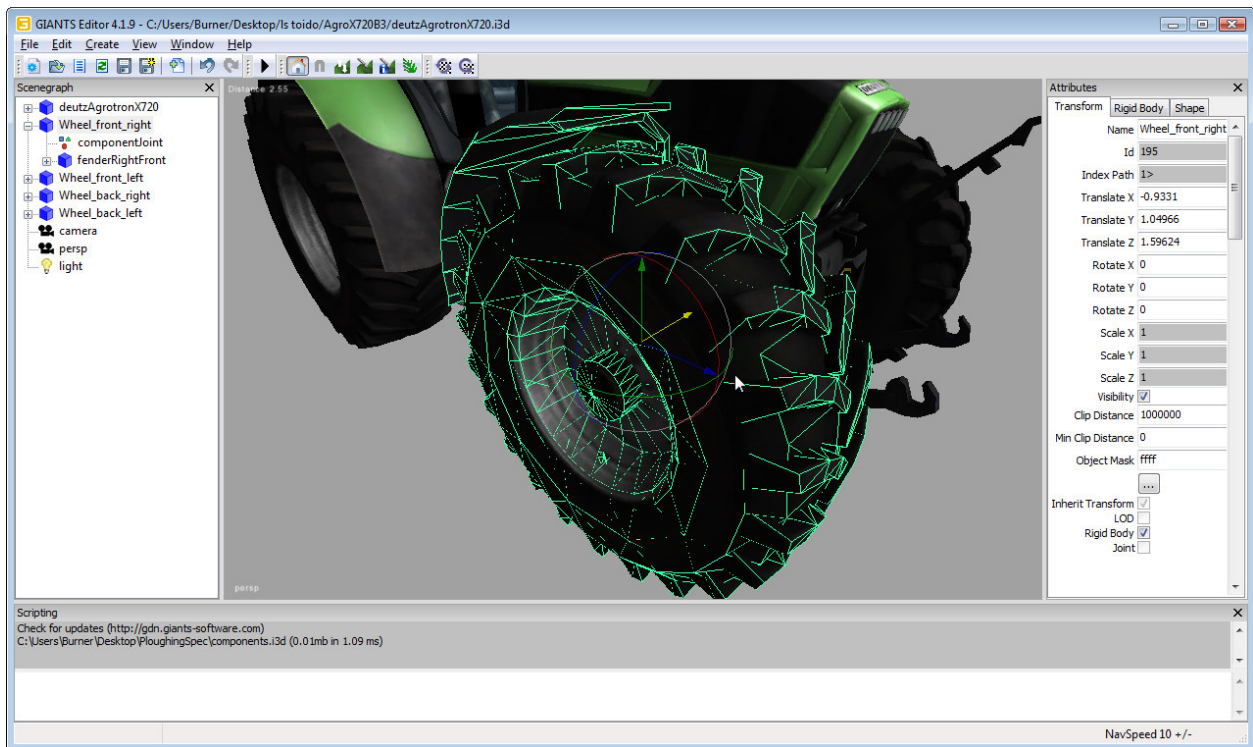


Set the parameters *Transform X*, *Transform Y*, *Transform Z* in copied wheel to „0“.

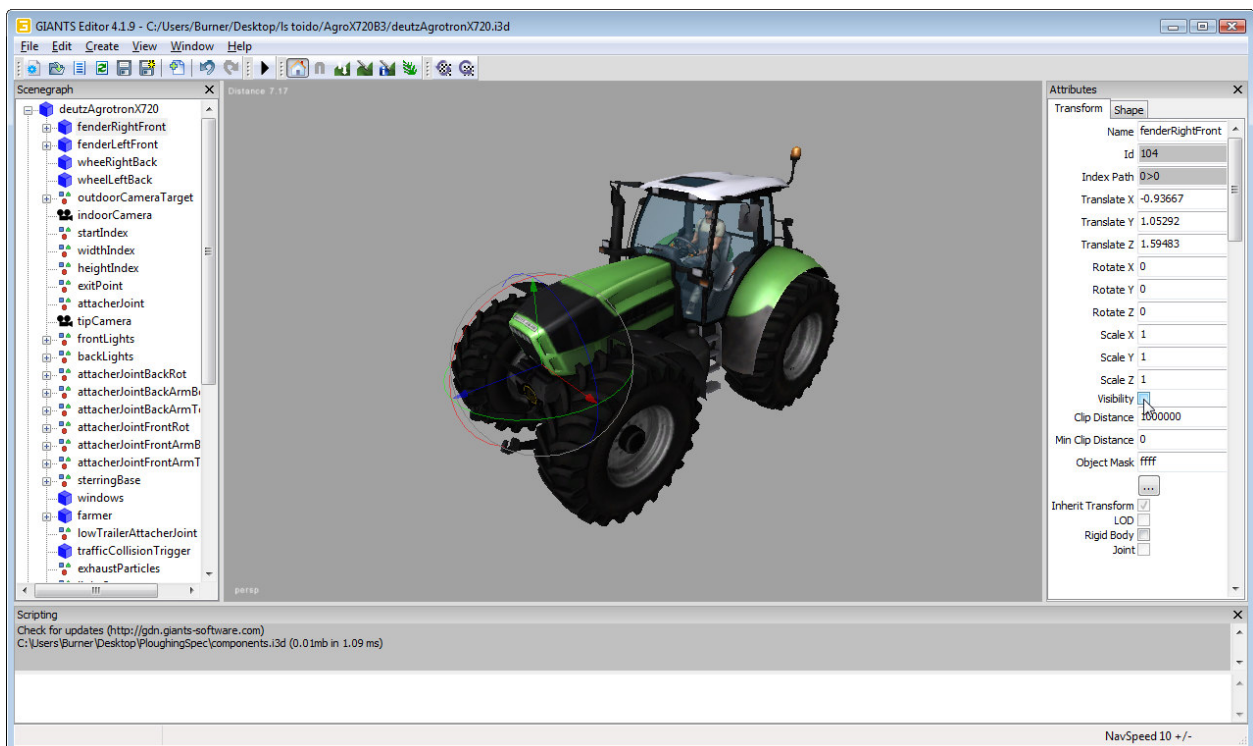


If wheel has a additional collision, copy it to the wheel component, whereas in old additional collisions, in the *Attributes* panel, uncheck *Rigid body* field or remove collision completely. Complete removal of the old collision is not recommended because the order of items may change and you will need to recalculate all indexes again.

Select wheel component and move it to a new wheel coincided with the old.



After adjusting wheel position, select the old wheel, located in the main collision model and in the *Attributes* panel, uncheck *Visibility* field or remove wheel completely. Complete removal of the old wheel is not recommended because the order of items may change and you will need to recalculate all indexes again.



Repeat procedures for the other three wheels, copying them in subsequent collisions. Wheel order in your mod may be different than presented in the documentation.

4. Modifications in vehicle *xm*/file

General structure of components code:

```
<components count="5">
  <!-- main component -->
  <component1 centerOfMass="0 0 0" solverIterationCount="190" />
  <!-- front right wheel component -->
  <component2 centerOfMass="0 -0.8 0" solverIterationCount="190" />
  <!-- front left wheel component -->
  <component3 centerOfMass="0 -0.8 0" solverIterationCount="190" />
  <!-- back right wheel component -->
  <component4 centerOfMass="0 -0.8 0" solverIterationCount="190" />
  <!-- back left wheel component -->
  <component5 centerOfMass="0 -0.8 0" solverIterationCount="190" />

  <!-- front right wheel joint (componentJointIndex = 1) -->
  <joint component1="0" component2="1" index="1">0" rotLimit="0 0 0" transLimit="0 0 0" breakable="false"/>
  <!-- front left wheel joint (componentJointIndex = 2) -->
  <joint component1="0" component2="2" index="2">0" rotLimit="0 0 0" transLimit="0 0 0" breakable="false"/>
  <!-- back right wheel joint (componentJointIndex = 3) -->
  <joint component1="0" component2="3" index="3">0" rotLimit="0 0 0" transLimit="0 0 0" breakable="false"/>
  <!-- back left wheel joint (componentJointIndex = 4) -->
  <joint component1="0" component2="4" index="4">0" rotLimit="0 0 0" transLimit="0 0 0" breakable="false"/>

  <!-- disable collisions between existing components -->
  <collisionPair component1="0" component2="1" enabled="false"/>
  <collisionPair component1="0" component2="2" enabled="false"/>
  <collisionPair component1="0" component2="3" enabled="false"/>
  <collisionPair component1="0" component2="4" enabled="false"/>
</components>
```

Number of components in your mod may be different from those mentioned above.

General structure of main specialization code:

```
<frontRightWheel wheelIndex="1" componentJointIndex="1" furrowDepth="0.3" fallSpeed="0.25" tireWidth="0.4" />
<frontLeftWheel wheelIndex="2" componentJointIndex="2" furrowDepth="0.3" fallSpeed="0.25" tireWidth="0.4" />
<backRightWheel wheelIndex="3" componentJointIndex="3" furrowDepth="0.3" fallSpeed="0.25" tireWidth="0.45" />
<backLeftWheel wheelIndex="4" componentJointIndex="4" furrowDepth="0.3" fallSpeed="0.25" tireWidth="0.45" />
```

Main parameters description:

- **<frontRightWheel />**
front right wheel data,
- **<frontLeftWheel />**
front left wheel data,
- **<backRightWheel />**
back right wheel data,
- **<backLeftWheel />**
back left wheel data,
- **wheelIndex="[...]"**
wheel index, number of wheel in *<wheels>* list:

```
<wheels autoRotateBackSpeed="2.5">
  <!-- front right wheel (wheelIndex = 1) -->
  <wheel rotSpeed="70" rotMax="34" rotMin="-34" driveMode="2" repr="0" driveNode="0|0" radius="1.038"
  deltaY="0.20" suspTravel="0.25" spring="100" damper="20" mass="0.42" lateralStiffness="25" />
  <!-- front left wheel (wheelIndex = 2) -->
  <wheel rotSpeed="70" rotMax="34" rotMin="-34" driveMode="2" repr="1" driveNode="1|0" radius="1.038"
  deltaY="0.20" suspTravel="0.25" spring="100" damper="20" mass="0.42" lateralStiffness="25" />
```

```

<!-- back right wheel (wheelIndex = 3) -->
<wheel rotSpeed="0" driveMode="1" repr="2" radius="1.128" deltaY="0.15" suspTravel="0.2" spring="150"
damper="20" mass="0.5" lateralStiffness="25" />
<!-- back left wheel (wheelIndex = 4) -->
<wheel rotSpeed="0" driveMode="1" repr="3" radius="1.128" deltaY="0.15" suspTravel="0.2" spring="150"
damper="20" mass="0.5" lateralStiffness="25" />
</wheels>,

```

no default value, if you specify an invalid index or an empty value, animation of the wheel will be completely ignored,

componentJointIndex="[...]"

wheel component joint index, number of joint in *<components>* list (example how to count joints was presented above, in general structure of components code), no default value, if you specify an invalid index or an empty value, animation of the wheel will be completely ignored,

- **furrowDepth="0.3"**
furrow depth, depth at which wheel collapses, default value is *0.3*,
- **fallSpeed="0.25"**
collapse speed, time in which wheel will collapse completely, expressed in seconds, default value is *0.25*,
- **tireWidth="0.4"**
wheel/tire width, default value is *0.4*.

Optional specialization code:

```

<workExhaustParticleSystems count="1">
  <workExhaustParticleSystem1 node="25" file="$data/vehicles/particleSystems/exhaustParticleSystemBig.i3d" />
</workExhaustParticleSystems>

<increaseRpmDuringPloughing value= "true" />

```

Optional parameters description:

- **<workExhaustParticleSystems>**
additional exhaust particle system, which appears during plowing,
- **node="[...]"**
index of object whose coordinates determine position of the additional exhaust particle system, no default value, if you specify an invalid index or an empty value, animation of the wheel will be completely ignored,
- **file="[...]"**
additional exhaust particle system file path, no default value, if you specify an invalid path or an empty value, additional exhaust particle system will be completely ignored,
- **<increaseRpmDuringPloughing value="[true|false]" />**
increasing engine speed during, default value is set to *true*.

If adding specialization required to create new type of vehicle *<vehicleTypes>*, name of the newly created type should be given in *<vehicle>* block, at the beginning of the *xml*/file:

```

<vehicle type="X720PloughingSpec">

```

IMPORTANT! After moving wheels to new components in *i3d* model, *repr* and/or *driveNode* indexes have changed and must be recalculated.

Therefore, ploughs in the game are designed for riding on flat terrain, after driving into a furrow, the end part may protrude above the ground. This phenomenon occurs most frequently when using the larger rotary plows. The problem can be solved by increasing the maximum limit rotation of attacher in *X* axis (*maxRotLimit*). In this way, after lowering plough adjusts its position to the shape of the terrain.

Below is an example code with changed *maxRotLimit* attacher value:

```
<attacherJoints>
  <attacherJoint index="14|0|0" rotationNode="14" maxRot="-14 0 0" rotationNode2="14|0" maxRot2="9 0 0"
    maxTransLimit="0 0.5 0" maxRotLimit="8 0 20" moveTime="1.5">
    <bottomArm rotationNode="15" translationNode="15|0" referenceNode="15|0|0" zScale="-1" />
    <topArm rotationNode="16" translationNode="16|0" referenceNode="16|0|0" zScale="-1" />
  </attacherJoint>
  <attacherJoint index="17|0|0" rotationNode="17" maxRot="10 0 0" rotationNode2="17|0" maxRot2="-7 0 0"
    maxTransLimit="0 0.4 0" maxRotLimit="8 0 20" moveTime="1.5">
    <bottomArm rotationNode="18" translationNode="18|0" referenceNode="18|0|0" zScale="1" />
    <topArm rotationNode="19" translationNode="19|0" referenceNode="19|0|0" zScale="1" />
  </attacherJoint>
</attacherJoints>
```

IMPORTANT! After moving wheels to new components in *i3d* model, *repr* and/or *driveNode* indexes have changed and must be recalculated.