

UNIT CREATION TUTORIAL

Written by Niek ten Brinke (aka niektb)

Introduction

This tutorial covers the workflow of adding and editing units. You will learn how to create templates and link them at the correct way. In the end you will have a fully featured axeman.



Table of contents

First start.....	1
Setting up a template.....	3
Setting up an actor.....	4
Make the unit trainable.....	6
Result.....	7

First start

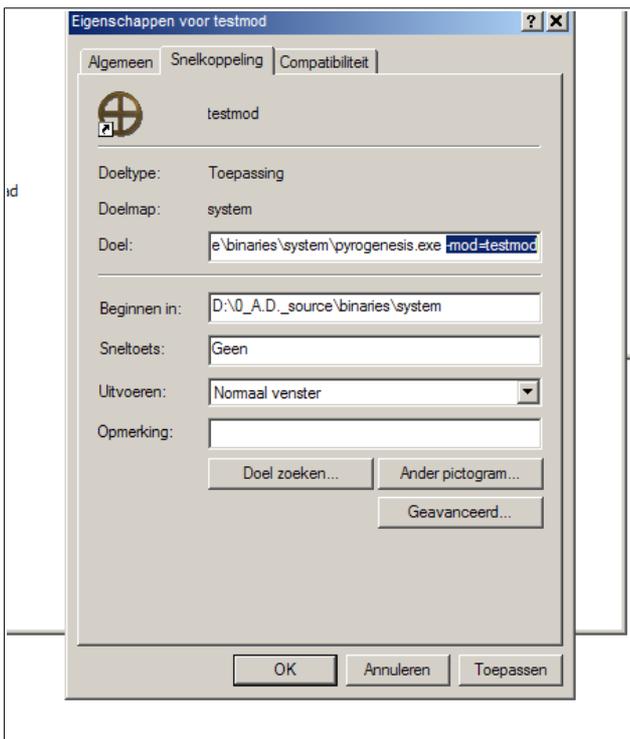
If you haven't done already, download the game and install it in any directory you like or obtain the source code using TortoiseSVN: <http://trac.wildfiregames.com/wiki/TortoiseSVN_Guide>. I'm using the SVN version in this tutorial.

Navigate to the directory of the game, in my case: <D:\0_A.D._source>. Then go to <\binaries\data\mods>. Here you will find a few folders, one of them is <public>. In this folder you will find all the data and gameplay scripts related to the main game of 0 A.D.

The game works with the so called mod folders. Everything you put in there will override the main

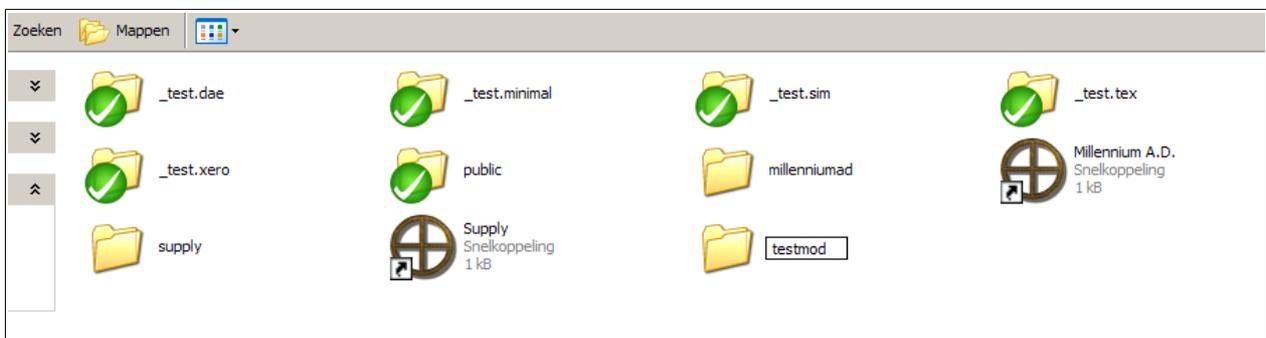
game's data, the only requirement is that you load them.

Do so by creating a shortcut targeting pyrogenesis.exe located in <\binaries\system>.



Add “-mod=testmod” to the target and press OK.

Go back to the mods directory and create a folder called <testmod>.



You will notice a few additional maps in my folder, ignore them.

Create the following map structure like you see in the public mod:

<\simulation\templates\units>

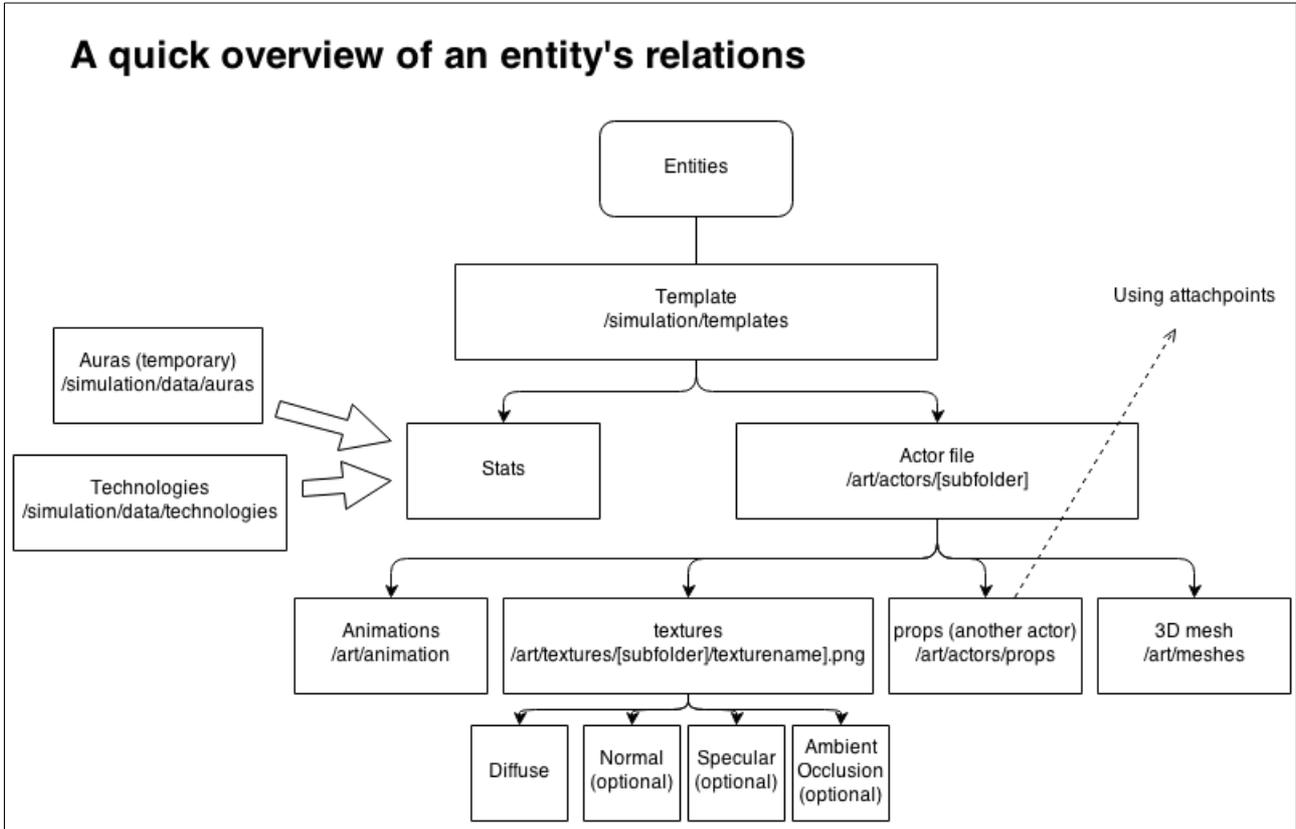
<\simulation\templates\structures>

<\art\actors\units\gaul>

To save time and effort we will use already existing meshes, textures and animations (we therefore do not create those maps), and do not write our template from scratch.

Well done! You now have the ability to launch your mod and know the map structure needed for editing stats – including auras – and visuals.

You might want to use this image to keep overview:



Setting up a template

Good. Now let's head over to the template. In <simulation\templates\units>, create a file called "gaul_champion_axeman.xml" and open it in a text editor of your choice. Finally paste the following code in it:

```

<?xml version="1.0" encoding="utf-8"?>
<Entity parent="template_unit_champion_infantry_swordsman">
  <Cost>
    <Resources>
      <metal>75</metal>
    </Resources>
  </Cost>
  <Attack>
    <Melee>
      <Hack>20.0</Hack>
      <Pierce>0.0</Pierce>
      <Crush>0.0</Crush>
    </Melee>
  </Attack>
</Entity>
  
```

```

    </Melee>
    <Charge>
        <Hack>30.0</Hack>
        <Pierce>0.0</Pierce>
        <Crush>0.0</Crush>
    </Charge>
</Attack>
<Identity>
    <Civ>gaul</Civ>
    <GenericName>Heavy Infantry</GenericName>
    <SpecificName>Gaul Axeman</SpecificName>
    <History>?</History>
    <Icon>units/celt_champion_infantry_gaul.png</Icon>
    <RequiredTechnology>unlock_champion_units</RequiredTechnology>
    <SelectionGroupName>units/gaul_champion_axeman</SelectionGroupName>
</Identity>
<UnitMotion>
    <WalkSpeed>7.0</WalkSpeed>
    <Run>
        <Speed>21.0</Speed>
    </Run>
</UnitMotion>
<VisualActor>
    <Actor>units/gaul/champion_axeman.xml</Actor>
</VisualActor>
</Entity>

```

Let me explain the most important tags to you.

At the second line you see something saying about a parent xml file. Everything about templates in 0 A.D. is working with parents and children. That is why we do not need to have every single tag in our file: it only overrides what is in the parent file.

In the <Cost> tag the training costs are defined, we did set the metal costs to 75. The other costs are defined in the parent file.

In <Identity> all info about the unit is explained. You might notice that we called the unit “Gaul Axeman” and that we are requiring a technology to be researched before the Axeman gets trainable.

Lastly we see the <VisualActor>. This is a link to an actor file in which the visuals are defined. In the template we just created, all the rest is defined.

Setting up an actor

Now, let's go to <\art\actors\units\gaul> and create the actor file we defined above.

Paste this code into the file:

```

<?xml version="1.0" encoding="utf-8"?>
<actor version="1">
  <castshadow/>
  <group>
    <variant frequency="1" name="Base">
      <animations>
        <animation file="biped/inf_sword_ready_e.dae" name="Idle"
          speed="100"/>
        <animation file="biped/inf_sword_ready_e.dae" name="Idle"
          speed="90"/>
        <animation file="biped/inf_sword_ready_e.dae" name="Idle"
          speed="85"/>
        <animation file="biped/inf_sword_ready_e.dae" name="Idle"
          speed="75"/>
        <animation file="biped/inf_sword_ready_e.dae" name="Idle"
          speed="110"/>
        <animation file="biped/dudewalk_swordshield.psa" name="Walk"
          speed="110"/>
        <animation file="infantry/sword/move/run/isw_s_off_01.psa"
          name="Run" speed="8"/>
        <animation file="infantry/sword/move/run/isw_s_em_03.psa"
          name="Run" speed="8"/>
        <animation event="0.5"
          file="biped/inf_2_hand_sword_attack_b.dae"
          name="attack_melee" speed="150"/>
        <animation file="infantry/general/death/inf_01.psa" name="Death"
          speed="200"/>
        <animation file="infantry/general/death/inf_02.psa" name="Death"
          speed="400"/>
        <animation file="infantry/general/death/inf_03.psa" name="Death"
          speed="200"/>
      </animations>
      <mesh>skeletal/m_pants_tunic.dae</mesh>
      <props>
        <prop actor="props/units/heads/celt_helmet_b.xml"
          attachpoint="helmet"/>
        <prop actor="props/units/heads/head_celt.xml"
          attachpoint="head"/>
        <prop actor="props/units/shields/celt_round_b.xml"
          attachpoint="shield"/>
        <prop actor="props/units/weapons/axe_single.xml"
          attachpoint="r_hand"/>
      </props>
    </variant>
  </group>
  <group>
    <variant frequency="1" name="chainmail-1">

```

```

        <textures><texture file="skeletal/celt_su3_soldurii_1.dds"
            name="baseTex"/></textures>
    </variant>
    <variant frequency="1" name="chainmail-2">
        <textures><texture file="skeletal/celt_su3_soldurii_2.dds"
            name="baseTex"/></textures>
    </variant>
    <variant frequency="1" name="chainmail-3">
        <textures><texture file="skeletal/celt_su3_soldurii_3.dds"
            name="baseTex"/></textures>
    </variant>
</group>
<material>player_trans.xml</material>
</actor>

```

Above you see the animations, the props and the textures defined. With props you should think of small stuff like shields, helmets and weapons (in case of unit props).

Make the unit trainable

So far, so good. We now have the stats and the visuals. Now let's think of a location to train the unit. I'm choosing the barracks. Therefore I need to copy the "gaul_barracks.xml" from <simulation\templates\structures> and copy it in the right folder in our own testmod.

```

18 <Obstruction>
19   <Static width="20.0" depth="20.0"/>
20 </Obstruction>
21 <ProductionQueue>
22   <Entities datatype="tokens">
23     units/gaul_infantry_spearman_b
24     units/gaul_infantry_javelinist_b
25     units/gaul_infantry_slinger_b
26     units/gaul_champion_infantry_barracks
27     units/gaul_cavalry_swordsman_b
28     units/gaul_cavalry_javelinist_b
29     units/gaul_champion_cavalry_barracks
30     units/gaul_champion_axeman
31   </Entities>
32   <Technologies datatype="tokens">
33     upgrade_rank_advanced_infantry
34     upgrade_rank_advanced_cavalry
35     upgrade_rank_elite_cavalry
36   </Technologies>
37 </ProductionQueue>
38 <VisualActor>
39   <Actor>structures/celts/barracks.xml</Actor>
40   <FoundationActor>structures/fndn_5x5.xml</FoundationActor>
41 </VisualActor>
42 </Entity>
43

```

Edit the file by adding "units/gaul_champion_axeman" (this is your template file) to <ProductionQueue>.

Result

Well done, you made it to complete the tutorial! Run the game using your shortcut. The result should be like below. If it gives some errors than likely you made some spelling errors. Go back and fix them accordingly to the errors. You might not like everything, but don't hesitate to change things to your own flavor.



Well done, you made it to create your own soldier! Since you have come this far, you have learned the basics from template modding and reusing existing art. You should even be capable of creating your own civilization now (from the template's point of view)!