Author	Phone	Date	Version
Oskar Holmstrand	0739800377	2005-12-14	X

WORKFLOW FOR STATIC MESHES FOR FOREIGN GROUND

ABSTRACT
This document proposes a workflow for static meshes, as relevant for a graphics artist.

DOCUMENT HISTORY

Version	Date	Author	Comments
PA1	2005-05-12	Oskar Holmstrand	First draft
PA2	2005-05-18	Oskar Holmstrand	Changed to CVS lingo, and added steps for open which is required
PA3	2005-05-18	Oskar Holmstrand	Added ”update” to workflow
X	2005-12-14	Oskar Holmstrand	Changed to Bachelor Thesis version

Objective

The reason for this document is to ensure that each graphics artist has the opportunity to work as effective as possible in a shared environment under the rules placed explicitly or implicitly by UT2004, leaving more time for creative work than issues concerning shared data. Although it may seem cumbersome, each step is actually pretty easy.

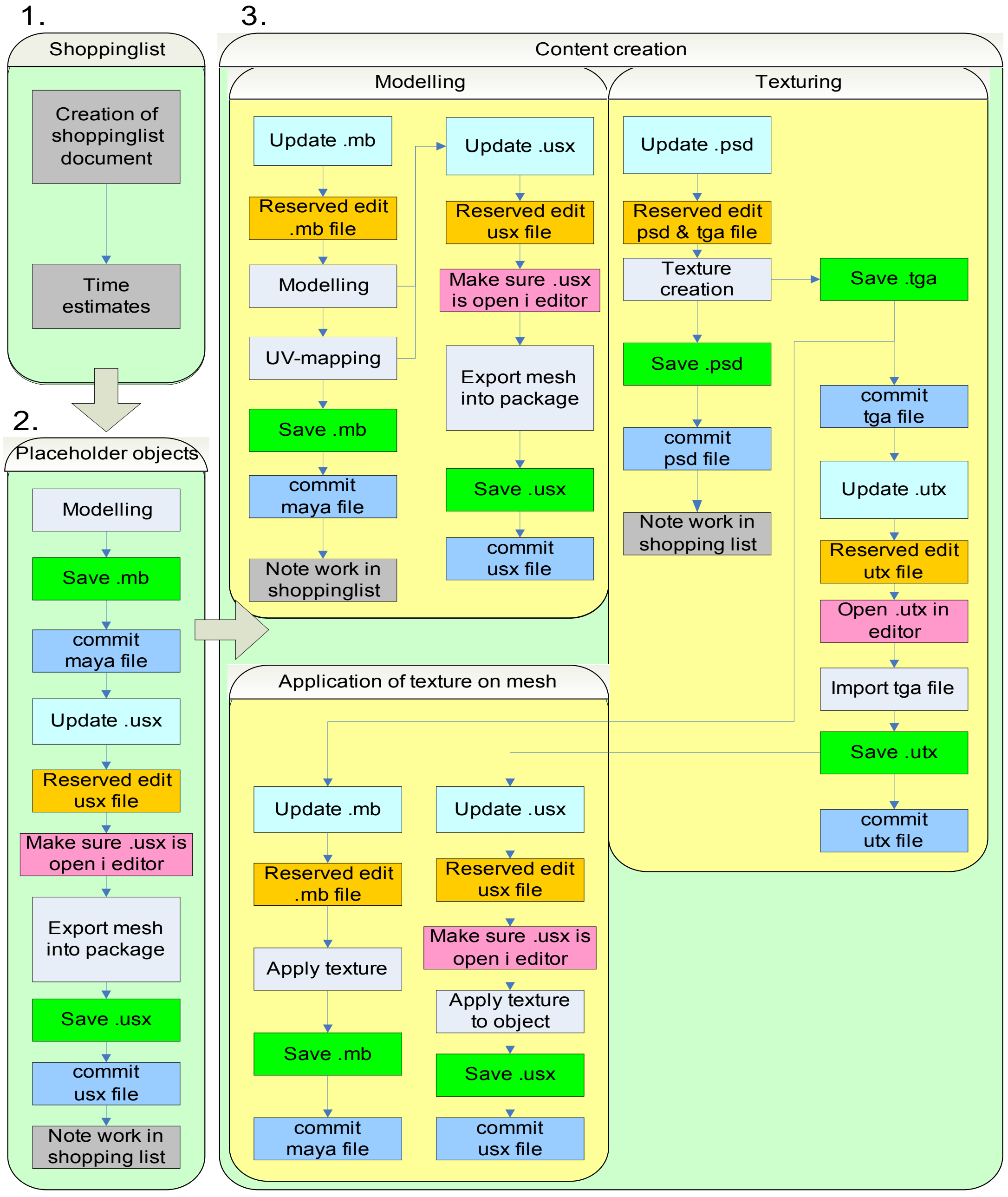
The workflow chart is the main thing you will be looking at. The text is simply there to justify, clarify and further expand the chart.

Workflow for static meshes

The workflow should be divided into two types of workflow. First you have to project size workflow issues, conserning what to do first in order to have an effective project. Then you have the local workflow, conserning the flow of effectively working in a shared environment. This document will be addressing both, as they are relevant for the construction of **static meshes**.

In this list, the shoppinglist(1) and the placeholder objects(2) are the things that should be done in the project before starting the other tasks (3).

Static meshes, workflow



Shoppinglist

The first step in the content creation pipeline is creating the shoppinglist. The shoppinglist should include all meshes that should be in the game, no matter how small or big. This list may then be appended to later. It should also always be updated with info on what has been completed.

Placeholder objects

In order to have a functional workflow, we have to have placeholder objects in unreal as fast as possible (i.e. a week ago). These objects will then slowly be replaced by more sofisticated versions. The placeholder object doesn't have to be more than a box. What is important with the placeholder object is that it has the **correct placement** and the **correct size**. This makes sure that no extra work has to be done when updating the objects.

There should be placeholder objects for **ALL** static meshes that is going to be used, before starting any other refinement.

The placeholder objects **MUST** be saved with the correct name (see the directory structure document & the shopping list) from the beginning, and be residing in the correct package with the correct name. Later refinements should always overwrite the old ones, to ensure that everything gets updated correctly.

CVS

In a project where several people work on the same data structures, a program like CVS is a must. Heavy discipline must be maintained while working on shared files (in UT2004's case, the package files), while files generally worked on by just yourself doesn't need the same discipline. Although keeping the CVS versions of the source files consistent with your local versions is positive, since it allows others to troubleshoot data more easily. Always commit files which you have completed your work on.

Modelling

Before modelling, always remember to mark the correct .mb file as reserved edit. This ensures that nobody else works on the same model as you. In the beginning of the project, the file you check out will most often be a placeholder file, which you will refine into something more aesthetically pleasing. An important thing to remember when modelling is to not override the boundaries of the placeholder object without careful consideration first, since this object may be used in a level, and it's size is therefore an integral part of some other asset.

Only mark the package file as reserved edit just before exporting, since this is a file that many other people may be working on too, then try to commit it when you're finished with your export. The maya file can be left as reserved edit until you decide your work with it is done.

Texturing

The texture is the most advanced workflow wise for static meshes. You have the .psd which is the original – this file should describe how you actually want the texture to look. Then you have the .tga file which is an intermediate format for UnrealEd to import, and you have the .utx file which is the actual file being used by Unreal.

When working on textures, the first thing to make sure is, as always, to have the .psd marked as reserved edit. You might aswell mark the .tga file aswell. After creating your texture, save and commit the .psd and the .tga.

We want the texture to not only be on the actual staticmesh in unreal, but also on our maya file for preview and consistency reasons. This means that you will probably want to have the .mb file marked as reserved edit while working with the texture too.

When the texture is completed, mark the texture package in which it will reside as reserved edit. Import it, again with the **correct name and location** stated in the directory structure document, compress it (RMB -> compress -> DXT3) and save the package. Then commit the package. Before importing into your local version of the package, make sure it is up to date with the CVS version!

After this, you will have to mark the static mesh package as reserved edit, apply the texture to the static mesh, and then commit the static mesh package back in. Again, make sure that you have runned update before so that you're not saving an old version!

Updating

Updating of a file should be done as often as you need to be sure that you are working on the newest version. For maya binaries, this may be seldom, but for the shared package files, this is most likely every time before exporting.

Testing

Always test your additions before commiting them into CVS. This is especially important for the shared package files. Testing is most important in the beginning of the project, when knowledge about trouble areas is more scarce.