|  |  |  |
| --- | --- | --- |
|  |  | Objectives - Building conventional 3D Models for our games using API Irrlicht.  - Define output file type for each kind of model.  - Define loading methods when importing in Irrlicht.  - Define texturing methods. |
|  |  |  |
|  |  | Primary stepsBuilding conventional 3D Models for API Irrlicht Step 1 Indeed, we need to fix rules for creating 3D Models because a lot of methods exist. There isn’t a lot of rules, for 3D modelers the work is almost the same and ask precisions:   * Low Poly Models but High details (mix) * No UV-Mapping for Environment (except exceptions). * Free imagination for the world, it’s your creation ;-)  Define output file type for each kind of model Step 2 Each model has his own file type, we must avoid formats owners:   * Environment: .OBJ Format * Characters: .X Format (**uncompressed**) * Trees: .X Format (**uncompressed**), because you’ll animate them (easy animation to make them moving because of the wind), programmers will program sheet falls, rain, etc. * Objects (animated): .X Format (**uncompressed**) * Objects (unanimated): .OBJ Format (**uncompressed**)  Define loading methods when importing in Irrlicht Step 3 You have choice.   * For .OBJ Files: You can use the .MTL file, which is associated to the .OBJ file, containing textures correspondences with the model. Or export .OBJ file and let the World Editor User create the association between model and textures using different materials of the model. * For .X Files: Let the World Editor user create the association between model and textures using different materials of the model. |
|  |  |  |
|  |  | Define texturing methods Step 4 You must organize models with a logic think. It means you must establish different parts inside the model using **materials**. I give you an example: (Imagine an house)   * The four faces of a house: you have choice of making a different texture for each face so one face is a material, or repeat the same texture on each face so the four faces are one material. The methods you will use in 3DS Max will be written in the exported 3D Model. * The door: you have different choices. First is to create UV-Map, which has the handle and the door, both in one material. Or create two different materials, one for the door, and one for the handle. * The chimney: you can imagine only one material with an easy and low texture because the player has weak chances to look VERY FINE the chimney… * Etc. |
|  |  |  |
|  |  | Using in World Editor When editing a node (take the house as example), the user can select a material, so material number 2 (when imagining you chose to create one material per house face) and apply texture to this second face, etc.  When editing a node, you can change colors of the selected material (ambient color, diffuse color, specular color and emissive color). If you can to create your own material type like “Normal Mapping”, you can create it in 3DS Max and apply the two textures (diffuse and normal mapped) on the model in the World Editor and select “Normal Mapping” in the “Material Type” combo box. Normal mapping is still under construction.  Notes: World Editor knows alone how to create collisions, mapping shadows, create actions in animations. |