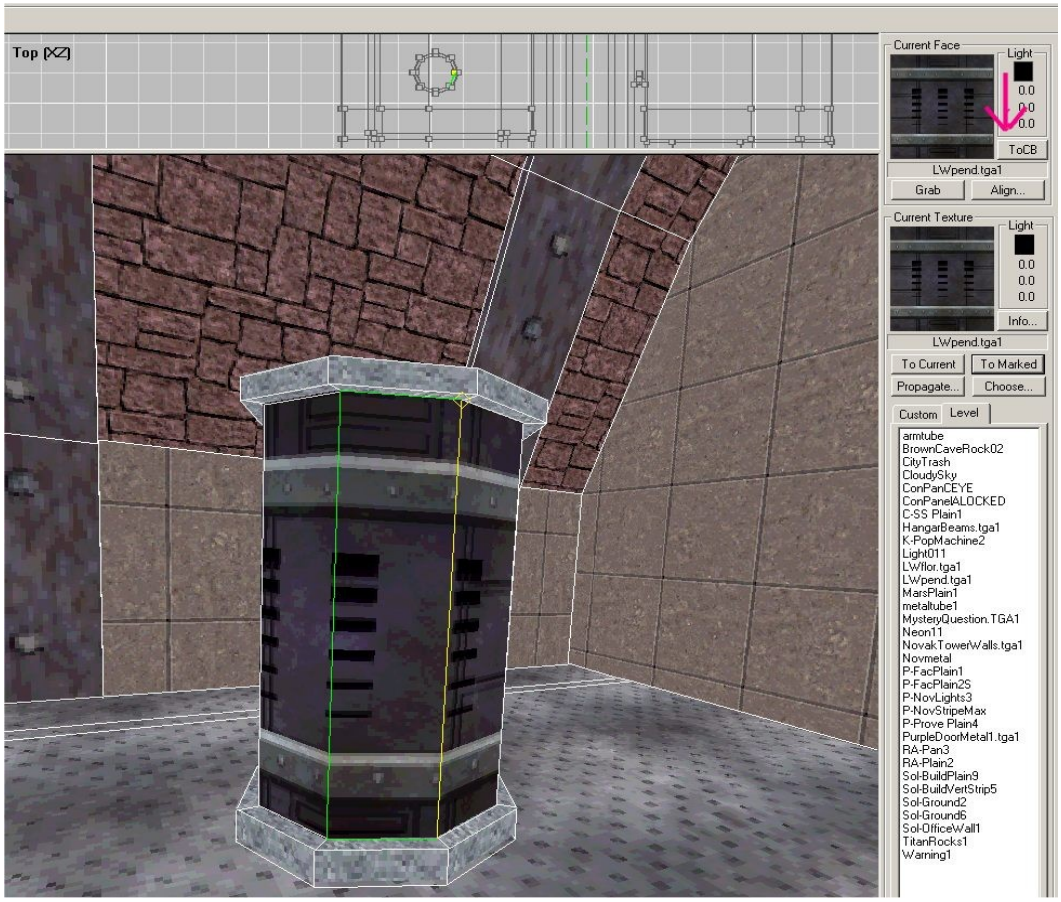
Micro Conseil 03 pour le D3Edit (V1.0) Atan

Comment utiliser le bouton ToCB de la barre des textures

ToCB est destiné à nous aider à scripter dans le DALLAS.
Il est utile quand il s'agit de remplacer des textures dans une pièce via un script DALLAS. ToCB peut alors servir à changer les textures d'un grand nombre de faces en insérant une ligne dans le script DALLAS.
ToCB va produire tout ce qu'il vous faut en un clic de souris et va le placer dans le presse papiers.
Il vous suffira de coller le contenu du presse papiers dans le fichier votreniveau.cpp.
Quand ensuite vous ouvrirez le DALLAS, vous pourrez utiliser la nouvelle fonction ainsi créée.

Voici un petit exemple d'utilisation de ToCB :
Le but est de remplacer les textures sur un cylindre pendant le jeu.
Ouvrir la pièce dans laquelle il y a le cylindre et sélectionner une des faces dont on veut faire changer la texture. Il ne faut pas utiliser cette texture à d'autres endroits de cette pièce!



Appuyer sur le bouton ToCB (flèche rouge). Si vous n'avez pas encore attribué un nom à la pièce, l'éditeur affichera un message pour vous demander de le faire. Entrez alors un nom pour la pièce dans le dialogue Room Properties. Il faudra ensuite appuyer à nouveau sur le bouton ToCB. On a l'impression qu'il ne se passe rien, mais toutes les choses nécessaires ont été copiées dans le presse papiers. Si vous n'avez pas encore de fichier votreniveau.cpp, il faut en générer un maintenant. Pendant que votre niveau est ouvert dans le D3Edit, démarrez le DALLAS, répondez OK à tous les messages, exécutez le script et sauvegardez le.

Vous trouverez alors le fichier votreniveau.cpp dans le dossier D3-SDK/Osiris. Ouvrir ce fichier avec un éditeur de texte (comme Notepad), et chercher la zone suivante dans le texte du fichier votreniveau.cpp :

```
// ============================================================
// Start of Custom Script Block - DO NOT EDIT ANYTHING BEFORE THIS
// ============================================================
/**{CUSTOM_SCRIPT_BLOCK_START}** DO NOT EDIT! **/

// Enter your custom script code here
```

Placez le curseur à cet endroit et coller le contenu du presse papiers (Ctrl+V)
Vous verrez alors un texte dans ce genre de celui-ci :

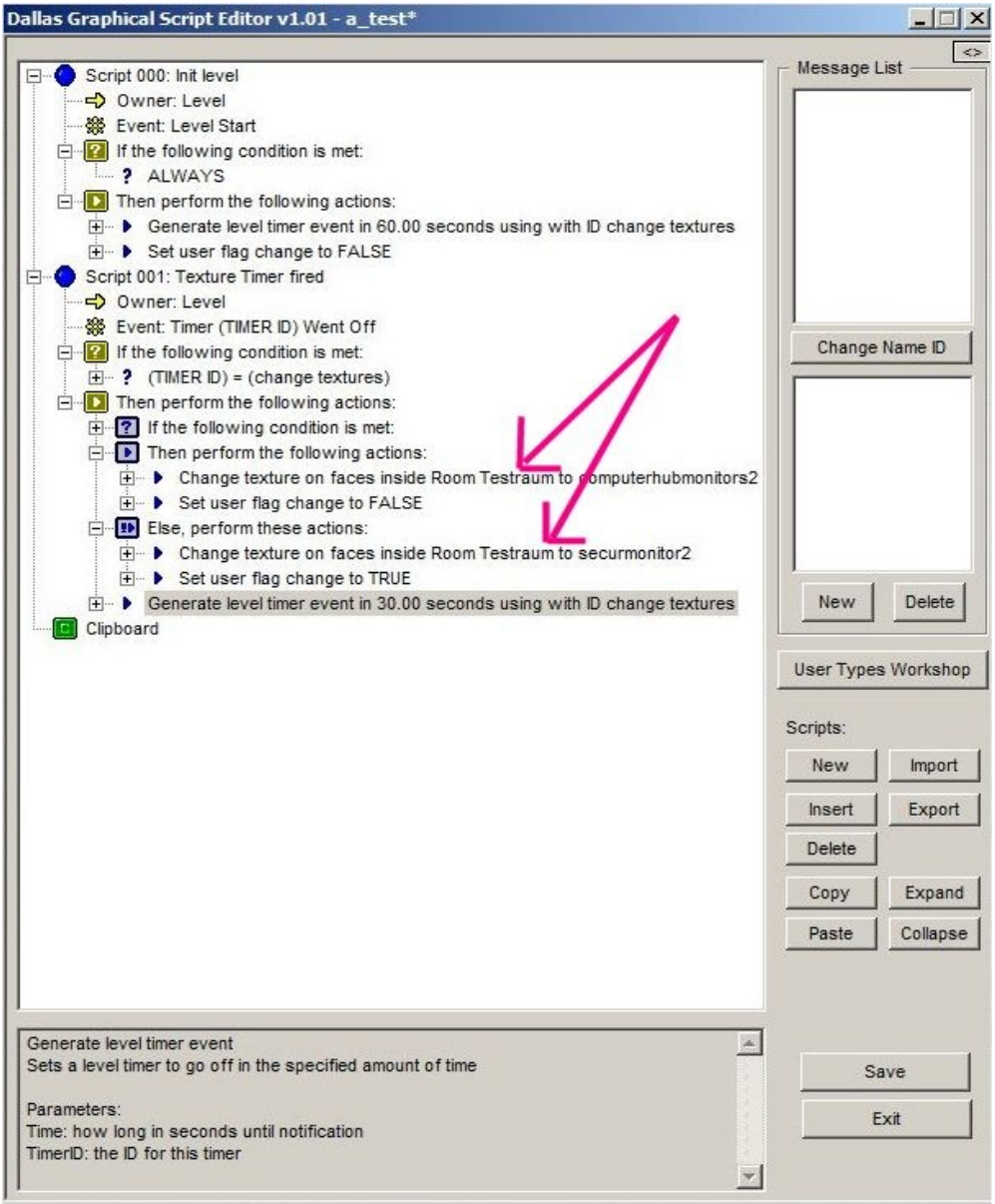
```
//Place the generated code inside your level.cpp Custom Script Block
#define NUM_FACES 11

typedef struct {
char *room_name;
int         room_id;
int         face_num;
} tTextureInfo;

tTextureInfo texture_info[NUM_FACES] = {
{"Testraum",     0,    92 },
{"Testraum",     0,    97 },
{"Testraum",     0,    102 },
{"Testraum",     0,    107 },
{"Testraum",     0,    112 },
{"Testraum",     0,    117 },
{"Testraum",     0,    122 },
{"Testraum",     0,    127 },
{"Testraum",     0,    503 },
{"Testraum",     0,    507 },
{"Testraum",     0,    514 },
};
/*
$$ACTION
Custom
Change texture on faces inside Room Testraum to [u:Texture]
aSetTextureForRoomFaces
Sets the faces to a given texture

Parameters:
  TextureName: the texture to assign
  Room: Not needed
$$END
*/

bool TextureInfoInitialized = false;

void aSetTextureForRoomFaces(int texture_id)
{
int j;

            if(!TextureInfoInitialized) {
            for(j=0;j < NUM_FACES;j++)
            texture_info[j].room_id = Scrpt_FindRoomName(texture_info[j].room_name);
            TextureInfoInitialized=true;
            }

            for(j=0;j<NUM_FACES;j++)
            aRoomSetFaceTexture(texture_info[j].room_id,texture_info[j].face_num,texture_id);
}
```

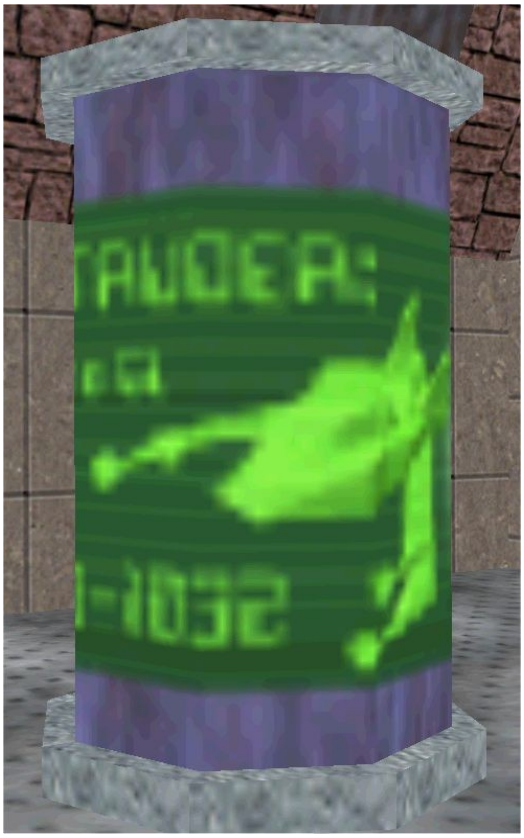
Sauvegarder le fichier votreniveau.cpp et ouvrir l'éditeur de script DALLAS.

L'exemple de script qui suit montre comment on peut utiliser la nouvelle fonction. Un minuteur va faire alterner deux textures.
Les flèches rouges pointent vers la fonction customisée qu'on a collée dans votreniveau.cpp.
Elle se trouve dans Action->Custom sous le nom suivant :

Change texture on faces inside Room Testraum to [u:Texture].

Dans le DALLAS, [u:Texture] ouvre une boîte de sélection pour choisir les textures. On en choisit deux. Après avoir exécuté le compilateur, on place les fichiers générés votreniveau.dll et votreniveau.msg dans la mission (mn3/hog) et on sauvegarde la mission. On fait ensuite un tour dans la mission et on voit si le résultat nous plaît. Vous pouvez, si vous le voulez, prendre pour le minuteur d'autres valeurs que celles qu'on voit dans l'exemple.





Note:
Si vous voulez utiliser cette fonction plus d'une fois dans votre script, il faut effectuer manuellement quelques modifications dans le fichier .cpp généré.
Voici ce qu'il faut changer :

```
#define NUM_FACES xx  => #define NUM_FACES_2 xx

} tTextureInfo;  => } tTextureInfo_2;

tTextureInfo texture_info[NUM_FACES] = {   =>  tTextureInfo_2 texture_info_2[NUM_FACES_2]

aSetTextureForRoomFaces  => aSetTextureForRoomFaces_2

bool TextureInfoInitialized = false;  =>  bool TextureInfoInitialized_2 = false;

void aSetTextureForRoomFaces(int texture_id) =>  void aSetTextureForRoomFaces_2(int texture_id)

if(!TextureInfoInitialized) { =>  if(!TextureInfoInitialized_2) {

for(j=0;j < NUM_FACES;j++) =>  for(j=0;j < NUM_FACES_2;j++)

texture_info[j].room_id = Scrpt_FindRoomName(texture_info[j].room_name);
 =>  texture_info_2[j].room_id = Scrpt_FindRoomName(texture_info_2[j].room_name);

for(j=0;j<NUM_FACES;j++)  =>  for(j=0;j<NUM_FACES_2;j++)

aRoomSetFaceTexture(texture_info[j].room_id,texture_info[j].face_num,texture_id);
 =>  aRoomSetFaceTexture(texture_info_2[j].room_id,texture_info_2[j].face_num,texture_id);
```

Augmenter _2 à _3 si vous voulez utiliser trois fois la fonction, etc...
Voilà l'astuce.