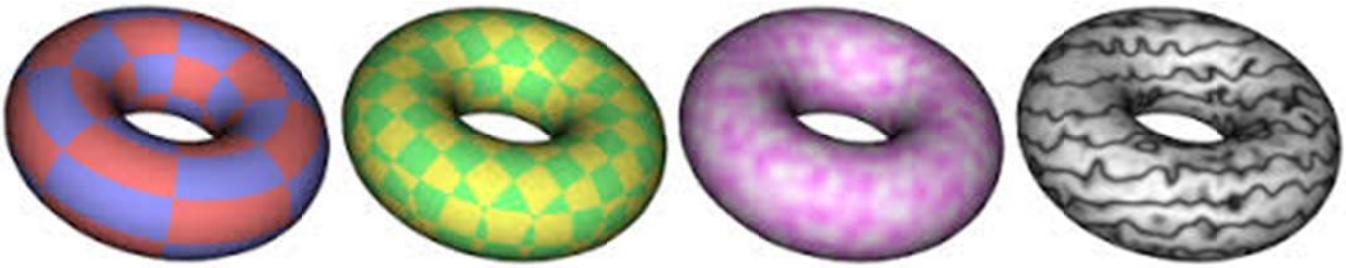
Programing Assignment #5

In this assignment you will texture your torus! See examples below:



Instructions:

Requirement 1 (60%) – Compute correct texture coordinates for your torus

You will reuse your code for creating a smooth torus and you will then add declarations to enable texture mapping. Just follow the steps below:

```
SnModel* sn = new SnModel;          // create your SnModel
GsModel& m = *sn->model();          // access the GsModel
rootg()->add ( sn );                // add to the scene

// 1. Generate the geometry:
...
Add your code here to generate the m.V, m.F, and m.N arrays for the torus
...

// 2. Create a "material group":
GsModel::Group& g = *m.G.push();
g.fi = 0;                           // The group starts at first face,
g.fn = m.F.size();                  // convers all faces,
g.dmap = new GsModel::Texture;      // and will be textured,
g.dmap->fname.set("C:/img.png");  // with this image (put here correct path and name!).

// 3. Make sure the number of materials matches the number of groups:
m.M.push().init();   // Only the diffuse component will come from the texture,
m.M.top() = ...;     // so add here any material properties you'd like.

// 4. Now you can add texture coordinates to be used per vertex:
int nv = m.V.size();
m.T.size ( nv );    // set same size as m.V array
for ( int i=0; i<nv; i++ )
{      m.T[i].set ( ... , ... ); // Compute your coordinates and put them here
}

// 5. Set parameters to enable texturing:
m.set_mode ( GsModel::Smooth, GsModel::PerGroupMtl );
m.textured = true;
```

Your texture should completely wrap around the torus in a nice way (see examples above). If you get a strange result most likely your texture coordinates are wrong.

Requirement 2 (30%) – Controls

In order to demonstrate that your object is correct and that your texture coordinates can be generated for any torus resolution, include the usual controls for changing the resolution of your torus (15%):

- '**q**' : increment the number of faces
- '**a**' : decrement the number of faces
- '**w**' : increment the *r* radius (by a small value)
- '**s**' : decrement the *r* radius (by a small value)
- '**e**' : increment the *R* radius (by a small value)
- **'d'** : decrement the *R* radius (by a small value)

Every time one of the keys above is pressed your application should instantly re-generate a new torus with updated texture coordinates. The result should always display the texture completely around the object for any chosen resolution.

Then, add the following control:

- '**space bar**' – pressing the space bar should have your application cycle among at least 3 different texture images! (15%)

To change textures you may simply rebuild your model with a different image file, or build 3 models in advance and use the visible(bool) method of every shape node to select which model to display. You may also try other solutions. Images are loaded and stored in a global resources manager, so a same image should not be loaded twice when you declare a same texture multiple times. If you are interested in exploring some more the support code you may also pre-load all textures using `GlResources::declare_texture()` and then just use texture ids in the `GsModel:: Group::dmap` of your model material group.

Requirement 3 (10%): Overall quality.

Everything counts here and, once again, there is no need to do anything complex, just make sure your project looks good and you will get full points.

Submission:

Please present and submit your PA as usual according to parules.txt. (Do not forget to upload your project before the deadline!)

Enjoy your texturing!