

Démarrer en OpenGL

- activer la gestion des textures 2D

- `glEnable(GL_TEXTURE_2D);`

- créer un nouvel objet texture(un numéro) = spécifier à OpenGL tous les paramètres de la texture, la texture elle-même, et aussi quelle texture on veut appliquer.

- `glGenTextures();`

-> autant de numéros que de textures demandées.

- quelle est la texture courante ?

- `glBindTexture();`

- caractéristiques de la texture : largeur, hauteur, format, etc... et bien sûr l'image elle-même

- `glTexImage2D();`

- `glTexParameter(.....);`

ex : texture 2 pixels par 2 pixels, en forme d'échiquier

```
GLubyte Texture[16] = {0,0,0,0, 0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF, 0,0,0,0};
```

```
GLuint Nom;
```

```
void InitGL()
```

```
{
```

```
...
```

```
glEnable(GL_TEXTURE_2D);           //Active le mode texture
```

```
glGenTextures(1,&Nom);             //Génère un n° de texture
```

```
glBindTexture(GL_TEXTURE_2D,Nom);  //Sélectionne ce n°
```

```
glTexImage2D ( GL_TEXTURE_2D,      //Type : texture 2D
```

```
0,                                  //Mipmap : aucun
```

```
4,                                  //Couleurs : 4
```

```
2,                                  //Largeur : 2
```

```
2,                                  //Hauteur : 2
```

```
0,                                  //Largeur du bord : 0
```

```
GL_RGBA,                            //Format : RGBA
```

```
GL_UNSIGNED_BYTE,                   //Type des couleurs
```

```
Texture );                          //Adresse de l'image
```

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
```

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
```

```
}
```

```
void Draw() // exemple : on plaque la texture sur un cube
```

```
{
```

```
...
```

```
glBegin(GL_QUADS);
```

```
// face 1
```

```
glTexCoord2i(0,0); glVertex3i(-1,-1,-1);
```

```
glTexCoord2i(1,0); glVertex3i(+1,-1,-1);
```

```
glTexCoord2i(1,1); glVertex3i(+1,+1,-1);
```

```
glTexCoord2i(0,1); glVertex3i(-1,+1,-1);
```

```
// faces 2, 3 ,4 ,5, 6
```

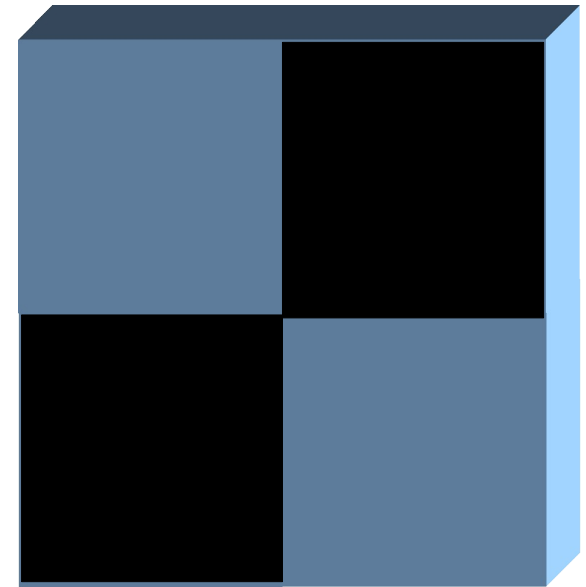
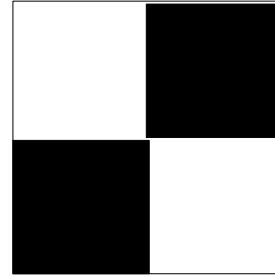
```
...
```

```
glEnd();
```

```
glutSwapBuffers(); pour GLUT
```

```
glutPostRedisplay();
```

```
}
```

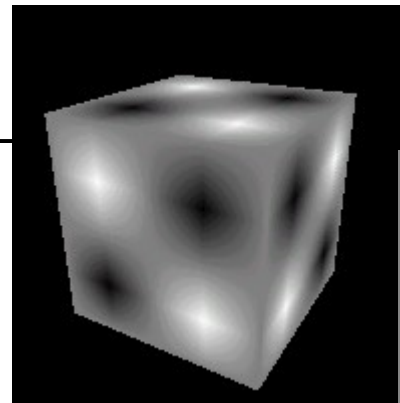


glTexCoord{1234}{sifd}[v]().

Filtrage

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
```

filtrage bilinéaire : `GL_LINEAR`
prendre le texel le plus proche :



si « magnification » `GL_NEAREST` :

un texel s'étend sur plusieurs pixels

si « minification » : `GL_NEAREST` :

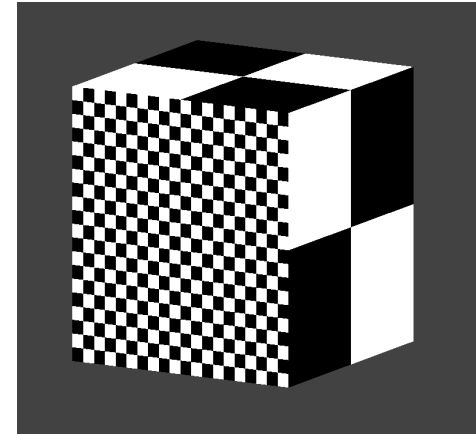
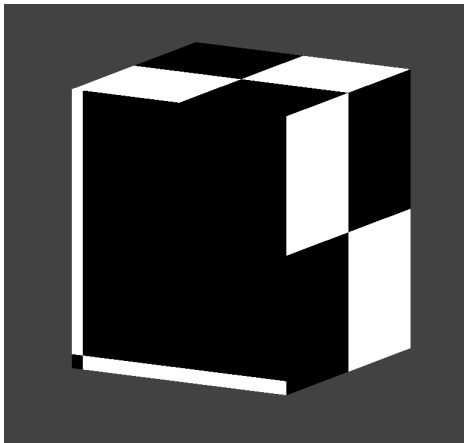
un pixel contient plusieurs texels.



Répétition

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
```

selon les coordonnées S ou T, GL_CLAMP (bridé) ou GL_REPEAT (répété)



MipMap

```
gluBuild2DMipmaps(GL_TEXTURE_2D,  
4, N, N, GL_RGBA,  
GL_UNSIGNED_BYTE, image);
```

```
glTexImage2D (GL_TEXTURE_2D, 0,  
4, N,N, 0, GL_RGBA,  
GL_UNSIGNED_BYTE, image);
```

```
glTexImage2D (GL_TEXTURE_2D,  
0, 4, N/2,N/2, 0, GL_RGBA,  
GL_UNSIGNED_BYTE, image);
```

```
glTexImage2D (GL_TEXTURE_2D,  
0, 4, N/4,N/4, 0, GL_RGBA,  
GL_UNSIGNED_BYTE, image);
```

...

```
glTexImage2D (GL_TEXTURE_2D,  
0, 4, 1,1, 0, GL_RGBA,  
GL_UNSIGNED_BYTE, image);
```

