

INF2705 Infographie

Spécification des requis du système

Travail pratique 6

Déplacement de surface

Table des matières

1	Introduction	2
1.1	But	2
1.2	Portée	2
1.3	Références	2
2	Description globale	3
2.1	But	3
2.2	Travail demandé	3
2.3	Fichiers fournis	6
3	Exigences	7
3.1	Exigences fonctionnelles	7
3.2	Exigences non fonctionnelles	7
A	Liste des commandes	8
B	Figures supplémentaires	9
C	Apprentissage supplémentaire	9

1 Introduction

Ce document décrit les exigences fonctionnelles et non fonctionnelles du TP6 « *Déplacement de surface* » du cours INF2705 Infographie.

1.1 But

Le but des travaux pratiques est de permettre à l'étudiant d'appliquer directement les notions vues en classe.

1.2 Portée

Chaque travail pratique permet à l'étudiant d'aborder un sujet spécifique.

1.3 Références

1. Site du cours INF2705
2. Site du cours INF2990
3. Guide de programmation C++ (INF2990).

2 Description globale

2.1 But

Le but de ce TP est de permettre à l'étudiant mettre en pratique l'affichage des surfaces avec OpenGL et le déplacement d'une surface dans les nuanceurs. Ce travail pratique lui permettra de se familiariser avec les nuanceurs de tessellation, le calcul des normales et les fonctions d'illumination. Il lui permettra aussi de mettre en pratique le déplacement de sommets à partir de textures de même que l'affichage en stéréoscopie.

2.2 Travail demandé

Partie 1 : la visualisation de surface et l'utilisation des nuanceurs de tessellation

On demande de réaliser un programme permettant d'afficher une surface 3D définie par la fonction $z = \text{Fonc}(x, y)$ pour $(x, y) \in [-2, 2] \times [-2, 2]$. Cette surface est d'abord affichée en utilisant les sommets de deux triangles représentant un carré plat dans le plan xy . La première étape sera d'utiliser des GL_PATCHES et d'activer les nuanceurs de tessellation afin d'augmenter le nombre de triangles (voir Figure 1). La seconde étape sera de calculer la hauteur en z dans un nuanceur en utilisant la fonction $\text{Fonc}()$. Ce déplacement en z sera modulée par un facteur modifiable interactivement par des touches au clavier et on utilisera deux plans de coupe (`glClipDistance[]` calculée dans un nuanceur) afin de restreindre l'affichage de la fonction en hauteur dans l'intervalle $z \in [-2, 2]$ (voir Figure 1).

La surface sera éclairée par une source de lumière soit ponctuelle, soit directionnelle dont la position et l'orientation sont contrôlées interactivement. Dans le nuanceur de tessellation, il faudra donc calculer la normale à chaque nouveau sommet en utilisant la dérivée de la fonction $\text{Fonc}()$:

$$\vec{N}(x, y) = \left[\frac{\partial \text{Fonc}(x, y)}{\partial x}, \frac{\partial \text{Fonc}(x, y)}{\partial y}, -1 \right]$$

Ce calcul sera effectué par une méthode numérique en utilisant (avec $\epsilon = 0.01$) :

$$\vec{N}(x, y) = \left[\frac{(\text{Fonc}(x + \epsilon, y) - \text{Fonc}(x - \epsilon, y))}{(2 * \epsilon)}, \frac{(\text{Fonc}(x, y + \epsilon) - \text{Fonc}(x, y - \epsilon))}{(2 * \epsilon)}, -1 \right]$$

(N'oubliez pas de normaliser le vecteur \vec{N} ainsi calculé avant de l'utiliser.)

Enfin, de façon semblable à la source de lumière qui est ponctuelle ou directionnelle, on utilisera la position réelle de l'observateur (si `localviewer=true`) ou une direction constante (0.0, 0.0, 1.0) vers l'observateur (si `localviewer=false`) pour l'illumination.

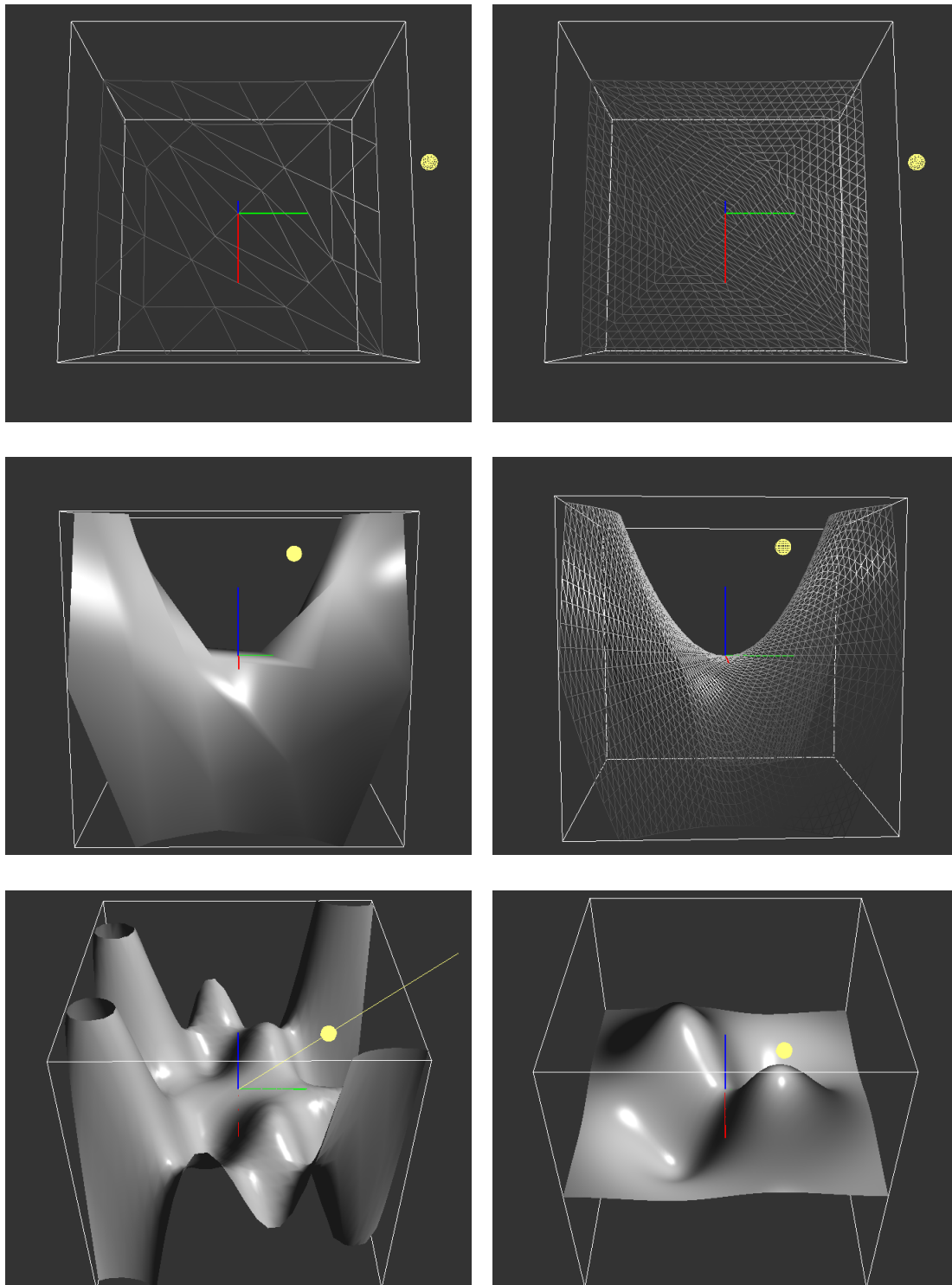


FIGURE 1 – (en haut) La surface plane avec une basse ou haute tessellation ; (au milieu) déplacement de la surface selon une fonction, avec tessellation basse ou haute ; (en bas) déplacements selon différentes fonctions

Partie 2 : le déplacement avec une texture et l’affichage en stéréo

Afin de bien comprendre les utilisations possibles ou alternatives des textures, on déformera le carré selon une texture plutôt que selon une fonction. On utilisera l’intensité (la moyenne des RGB) des texels pour modifier la hauteur des sommets (voir Figure 2).

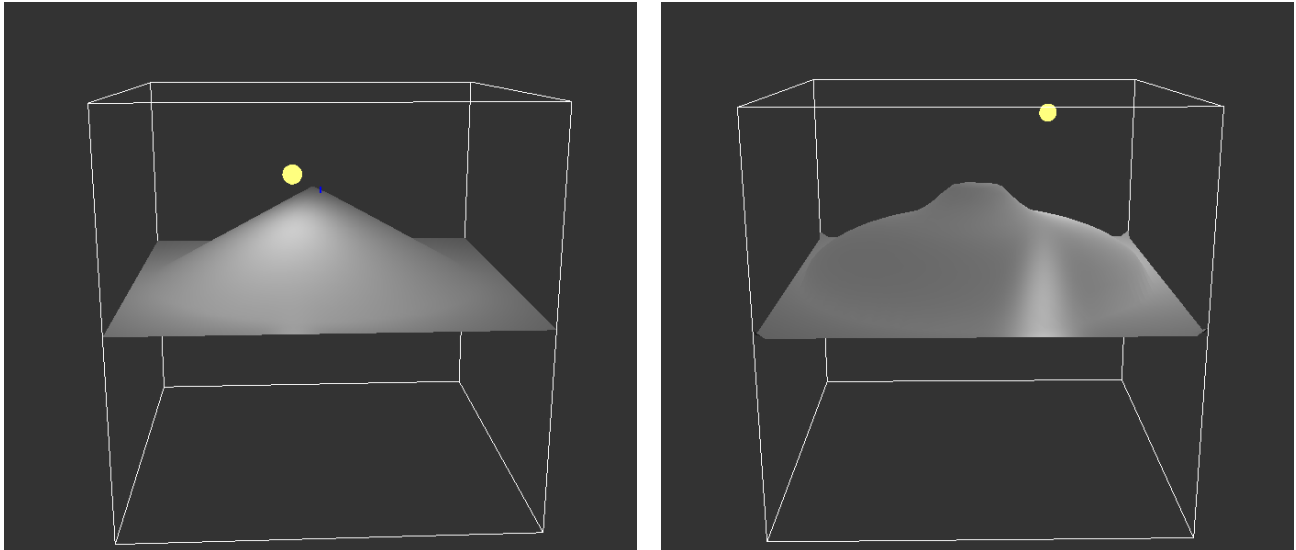


FIGURE 2 – Placage de déplacement avec deux textures différentes

Une seconde texture pourra être appliquée afin de « colorer » la surface déformée. Comme dans la partie 1, le programme permettra de moduler l’effet du déplacement et il sera évidemment aussi possible de changer les textures utilisées en cours d’exécution. (voir Figure 3).

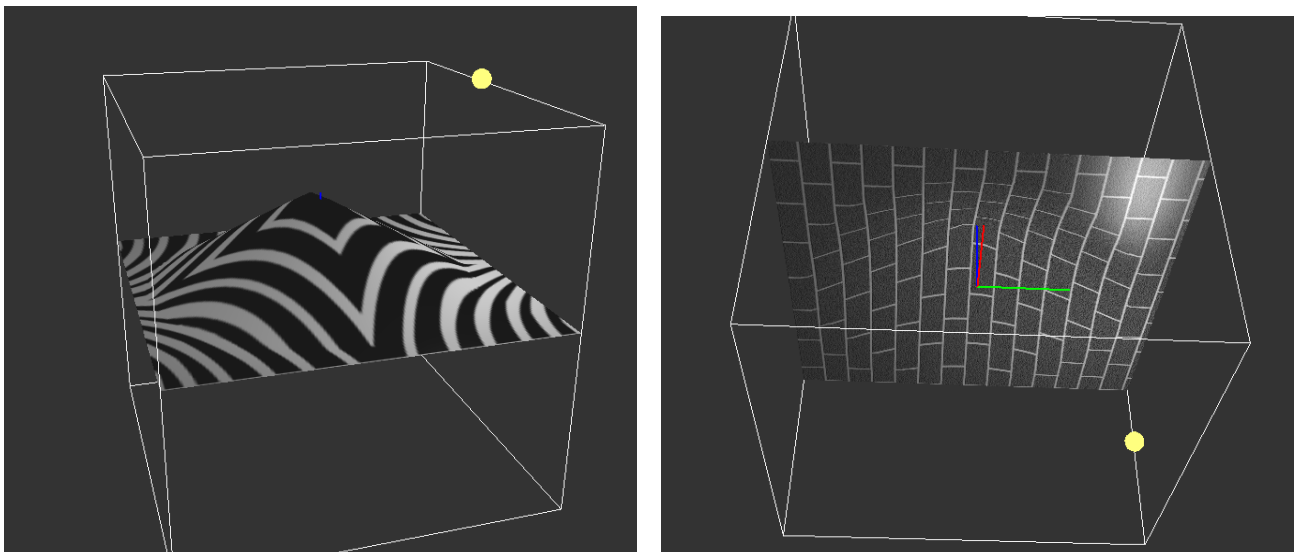


FIGURE 3 – Placage de déplacement avec deux textures différentes

Afin de mieux percevoir la profondeur, la scène pourra être vue en stéréoscopie en mode anaglyphe ou en mode double (voir Figure 4).

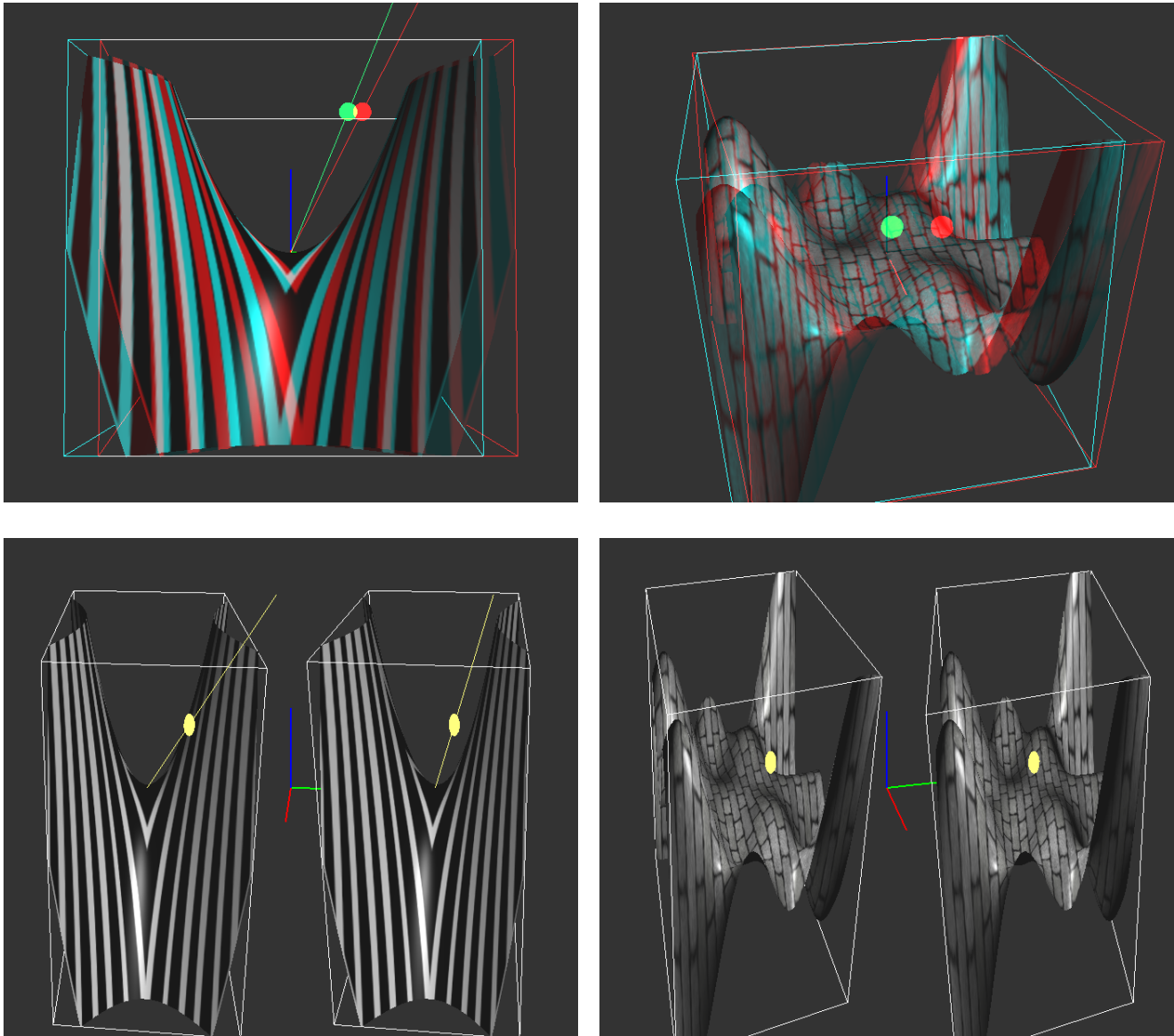


FIGURE 4 – Affichage en anaglyphe (pour des lunettes) ou en mode double (pour un écran autostéréoscopique)

2.3 Fichiers fournis

Le code fourni présente un quadrilatère en mode perspective. Les rotations de la caméra sont déjà implémentées. Des textures (Figure 5) sont fournies avec le code de base.

3 Exigences

3.1 Exigences fonctionnelles

Partie 1 :

- E1. Le surface carrée est subdivisée en triangles en utilisant les nuanceurs de tessellation.
- E2. On utilise des plans de coupe pour limiter l'affichage en Z.
- E3. Le logiciel affiche chacune des surfaces définies par la fonction *Fonc()*.
- E4. Le calcul des normales est fait correctement et la surface est bien illuminée.
- E5. Le calcul de l'illumination est correct, autant pour la lumière ponctuelle que directionnelle.
- E6. On peut utiliser la position réelle de la caméra ou un vecteur dans sa direction (selon `localviewer`).
- E7. Le logiciel utilise correctement les touches listées à l'annexe A.

Partie 2 :

- E8. L'affichage peut être en stéréoscopie en mode anaglyphe ou en mode double.
- E9. Les neuf textures peuvent être affichées sur le carré.
- E10. Le déplacement par la texture est implémenté à l'aide de nuanceurs.
- E11. Le coloration par la texture est implémentée à l'aide de nuanceurs.
- E12. Il est possible d'ajuster l'effet du placage de déplacement.

3.2 Exigences non fonctionnelles

La totalité du dessin peut se faire dans la fonction `afficherScene()` et dans les nuanceurs. Vous pouvez cependant ajouter des fonctions supplémentaires ainsi que d'autres classes et/ou fichiers si vous le jugez nécessaire.

ANNEXES

A Liste des commandes

Touche	Description
v	Recharge les fichiers des nuanceurs et recrée le programme
g	Passage fil de fer ou plein
i ou k	Incrémenter ou décrémenter le niveau de tessellation intérieure (INNER)
o ou l	Incrémenter ou décrémenter le niveau de tessellation extérieure (OUTER)
u ou j	Incrémenter ou décrémenter le niveau de tessellation intérieure ET extérieure
f	Passer d'une définition de surface à une autre pour le déplacement
+ ou -	Moduler l'effet du déplacement
Gauche ou droit	Changement du vecteur direction de la lumière selon les X
Bas ou haut	Changement du vecteur direction de la lumière selon les Y
PageSuiv ou PagePrec	Changement du vecteur direction de la lumière selon les Z
p	Changement entre lumière ponctuelle et directionnelle
w	On utilise la position réelle de l'observateur (<code>localviewer=true</code>) ou on utilise la direction vers l'observateur (0.0, 0.0, 1.0) (<code>localviewer=false</code>)
s	Passage de l'affichage mono et stéréo anaglyphe et stéréo double
c	Passer d'une texture à une autre pour la couleur
d	Passer d'une texture à une autre pour le déplacement
e	Passer d'une texture à une autre pour le déplacement ET la couleur
0	Revenir à la surface définie dans <i>Fonc()</i> pour le déplacement
ESC	Arrêt de l'application

B Figures supplémentaires

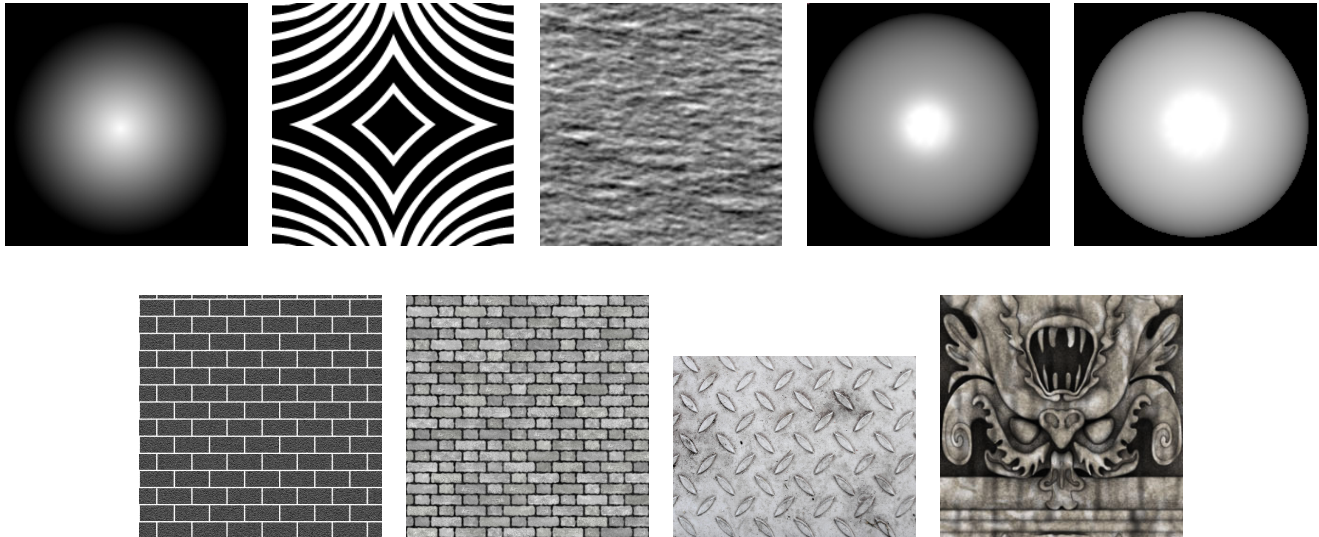


FIGURE 5 – Les textures utilisées

C Apprentissage supplémentaire

1. Utiliser différentes fonctions pour définir $Fonc(x, y)$.
2. Utiliser des textures différentes pour le déplacement et pour la couleur.
3. Implémenter le placage de déplacement pour des textures en couleur.