



**POLYTECHNIQUE
MONTRÉAL**

LE GÉNIE
EN PREMIÈRE CLASSE

INF2705 Infographie

Spécification des requis du système

Travail pratique 6

Déplacement de surface

Département de génie informatique et génie logiciel

École polytechnique de Montréal

Hiver 2015

Historique des modifications du document

Date	Version	Description	Auteur
	1.0	Version initiale	Benoît Ozell

Table des matières

1	Introduction	3
1.1	But	3
1.2	Portée	3
1.3	Références	3
2	Description globale	4
2.1	But	4
2.2	Travail demandé	4
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	8
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 des sommets dans le nuanceur. Ce travail pratique lui permettra de se familiariser avec le dessin d'une grille, le calcul des normales et les fonctions d'illumination (`glLightfv()`, `glEnable(GL_LIGHTING)`), et l'affichage en stéréoscopie. Ce TP permettra aussi de mettre en pratique le déplacement de sommets à partir de textures en utilisant les nuanceurs.

2.2 Travail demandé

Partie 1 : la visualisation de surface (déplacement dans nuanceur de sommets) en stéréo

On demande de réaliser un programme permettant d'afficher une surface 3D définie par la fonction $z = Fonc(x, y)$ pour $(x, y) \in [-2, 2] \times [-2, 2]$. Cette surface est affichée en déformant un carré plat représenté par une grille de $N \times N$ quadrilatères dans le plan xy et dont la hauteur en z sera calculée *dans le nuanceur de sommets* en utilisant la fonction `Fonc()`. Cette valeur de déplacement z sera aussi modulée par un facteur modifiable interactivement par les touches `+` / `-`.

La scène pourra être vue en mono, en stéréoscopie en mode anaglyphe ou en mode double. On utilisera des plans de coupe (`glClipPlane`) 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 directionnelle, soit positionnelle dont la position et l'orientation sont contrôlées interactivement. Il faudra donc calculer la normale à chaque sommet *dans le nuanceur de sommets* en utilisant la dérivée de la fonction `Fonc()` selon x et y :

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

Ce calcul sera effectué par une méthode numérique (plutôt qu'analytique) en utilisant :

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

(N'oubliez pas de normaliser la normale \vec{N} ainsi calculée dans le nuanceur.)

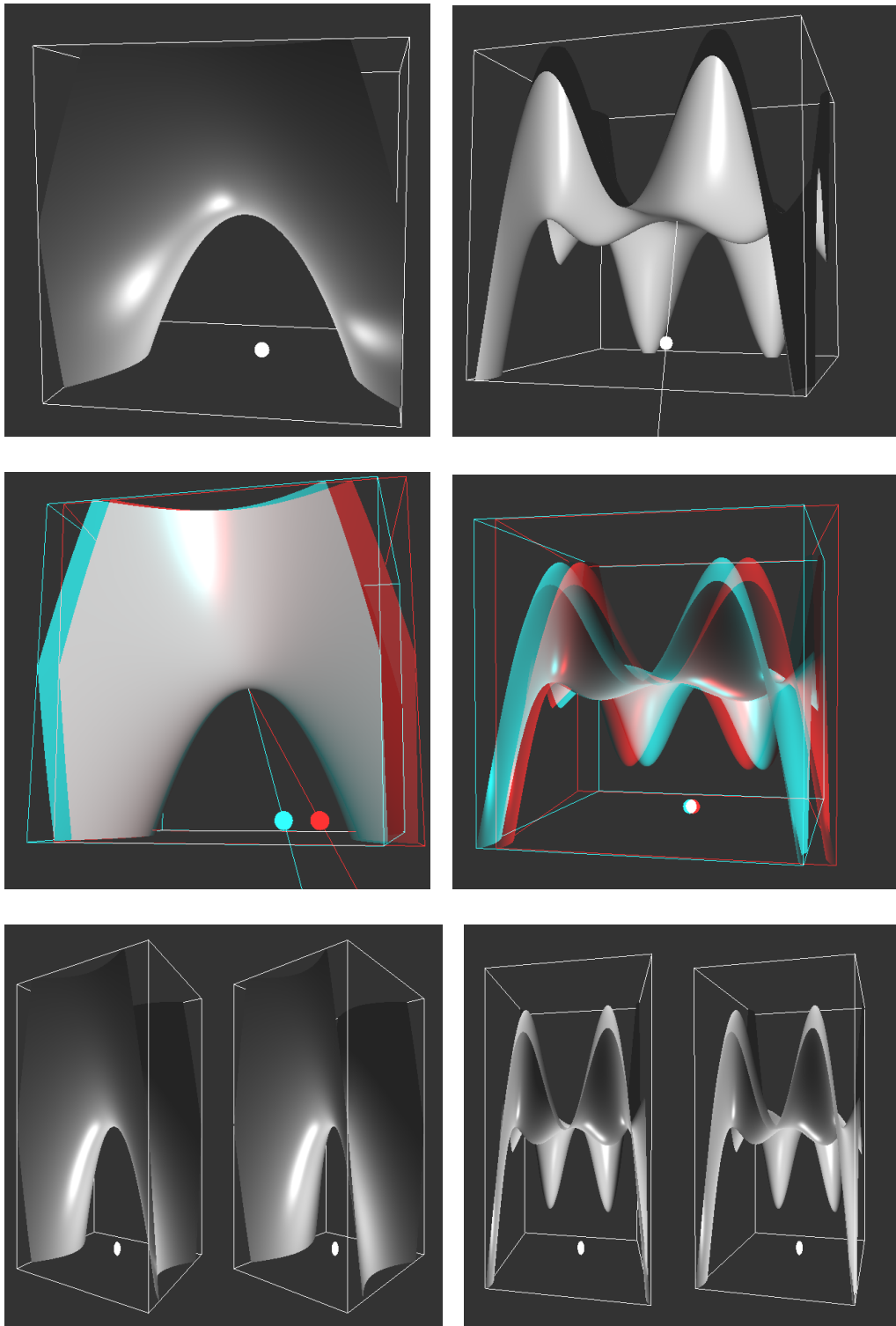


FIGURE 1 – Déplacement de la surface selon une fonction : en mono, en anaglyphe, en mode double

Partie 2 : le déplacement avec une texture

(Attention : je modifierai peut-être un peu les requis pour cette partie d'ici la seconde séance de laboratoire afin d'ajouter l'utilisation d'une seconde texture pour la couleur. Rien de trop difficile, mais intéressant !)

Le programme permettra d'afficher un carré sur lequel une texture (parmi plusieurs) est appliquée. Le programme permet de changer d'une texture à l'autre en cours d'exécution.

En plus d'afficher la texture sur le carré, on demande d'effectuer un déplacement des sommets du carré à l'aide de la texture, ce qui implique d'utiliser les valeurs de la texture pour modifier la hauteur des sommets constituant le carré. Le carré plat sera donc déformé et présentera un relief (Figure 2).

Une fois le placage de déplacement réalisé, le programme doit également permettre de moduler l'effet du placage de déplacement ; c'est-à-dire de permettre de modifier la hauteur des points du carré selon une proportion uniforme aux données de textures.

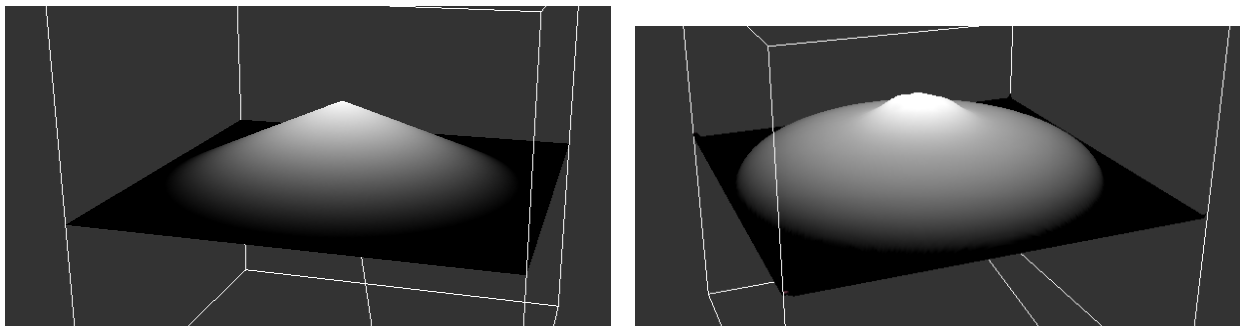


FIGURE 2 – Placage de déplacement avec deux textures différentes

2.3 Fichiers fournis

Le code fourni présente un quadrilatère en mode perspective. Les rotations de la caméra sont déjà implémentées. Des textures (Figure 3) sont fournies avec le code de base. Le code pour initialiser, charger et lancer les nuanceurs est aussi fourni.

3 Exigences

3.1 Exigences fonctionnelles

Partie 1 :

- E1. On utilise des plans de coupe pour limiter l'affichage en Z.
- E2. Le logiciel affiche chacune des surfaces définies par la fonction *Fonc()* dans le nuanceur de sommets.
- E3. L'affichage peut être en stéréoscopie en mode anaglyphe ou en mode double.
- 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 positionnelle que directionnelle.
- E6. Le logiciel utilise correctement les touches listées à l'annexe A.

Partie 2 :

- E7. Les neuf textures peuvent être affichées sur le carré.
- E8. Les paramètres des textures sont bien initialisés (avec la fonction *glTexCoordPointer()*).
- E9. Le placage de déplacement est implémenté à l'aide de nuanceurs.
- E10. 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
x	Passage entre le mode pipeline fixe et utilisation des nuanceurs
v	Recharge les fichiers des nuanceurs et recrée le programme
g	Passage fil de fer ou plein
l	L'observateur est local à la scène ou placé au loin (<i>localviewer</i>)
p	Changement entre lumière positionnelle et directionnelle
s	Passage de l'affichage mono et stéréo anaglyphe et stéréo double
0	Utilisation de <i>Fonc()</i> pour le déplacement
.	Passer d'une définition de <i>Fonc()</i> à une autre
1, 2, ..., 9	Utilisation de la texture 1, 2, ..., 9 pour le déplacement
+ / -	Ajuster l'effet de la fonction de déplacement ou du placage de déplacement
Gauche / droit	Changement du vecteur direction de la lumière selon les X
Bas / haut	Changement du vecteur direction de la lumière selon les Y
PageSuiv / PagePrec	Changement du vecteur direction de la lumière selon les Z
ESC	Arrêt de l'application

B Figures supplémentaires

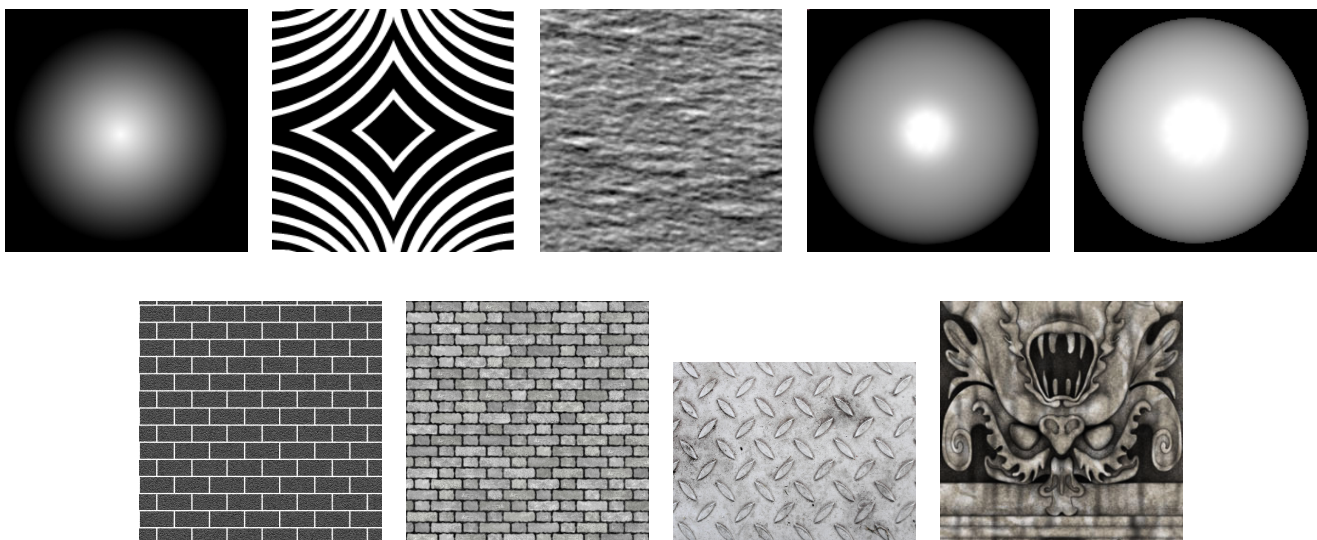


FIGURE 3 – Les textures utilisées

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.
4. Calculer une nouvelle normale pour chaque sommet afin d'illuminer la surface en relief.