Introduction

Dans ce TP, on va abandonner les cercles pour réaliser une height map.

Reprenez le dernier TP.

Debug

La première chose à faire est d'activer DirectX en débug. Pour cela lancer le control panel que vous trouverez dans les utilities du sdk de DirectX, accessible depuis le menu démarrer.

Cocher la case de Debug et glisser le slider de niveau d'output sur le 3^{e} cran par exemple.

Lancer votre exécutable et corriger les erreurs ou warnings de DirectX. Elles seront affichées dans l'output de Visual Studio.

Pour ceux qui ont regardé pour gérer les touches je vous conseille fortement de faire un mode wireframe (qui se fait avec un device->SetRenderSate(D3DRS_FILLMODE, D3DFILL_WIREFRAME)). Pour ceux qui n'ont pas regarder vous pouvez compiler avec quand vous en avez besoin

Height Map

On va maintenant réaliser la grille de la height map avec une hauteur fixe (les coordonnées Y étant la hauteur pour la suite du tp).

Attention

Il se peut que les cartes graphique des rack ne supporte que des petits vertex buffer. Testez avec une petit taille de grille(150 par 150 par exemple, il me semble que 200 par 200 passe aussi). Quand vous utiliserez le fichier raw que je fournis il faudra bien faire attention à pas dépasser cette limite et à en tenir compte dans vos boucle de remplissage, ainsi que pour des coordonnées de texture.

Vertex Buffer

Créez le Vertex Buffer et remplissez le, un vertex par point de la grille

Nombre de vertex du Vertex Buffer = TailleX * TailleZ

En deux boucles for imbriqué vous devriez avoir aucun soucis pour faire le remplissage.

Index Buffer

Créez l'Index Buffer. Taille de l'Index Buffer = 6 * (TailleX -1) * (TailleZ – 1)

Petit indice (ouch le jeu de mots :D) :

Les indices dans l'index buffer (en supposant que vous utilisez un triangle list) pour le i^e quadrilatère (formé par deux triangles) seront de la forme i, i + a, i + b, i + c, i + a, i + b. A vous de déterminez les valeur de a, b et c. Rien ne vaut une feuille de brouillon et un petit dessin.

Remplissez le et faites le draw call.

Hauteur

On va maintenant s'attaquer à la hauteur !!

Voici une fonction qui vous permet de charger mes fichiers raw. Mettez un include de <string> dans le stdafx.h

```
bool LoadRAW (const std::string& map)
{
	FILE  *file;

	fopen_s(&file, map.c_str (), "rb");
	if (!file)
		return false;


	fread(&m_sizeX, sizeof(unsigned short), 1, file);
	fread(&m_sizeZ, sizeof(unsigned short), 1, file);

	unsigned int size = m_sizeX * m_sizeZ;
	unsigned char *tmp = new unsigned char[size];
	m_height = new float[size];

	fread(tmp, sizeof(unsigned char), size, file);

	fclose(file);

	int i = 0;

	for (unsigned short z = 0; z < m_sizeZ; ++z)
		for (unsigned short x = 0; x < m_sizeX; ++x, ++i)
			m_height[i] = float ((m_maxY * tmp[i]) / 255.0f);

	delete[] tmp;
	return true;
}
```

Il vous suffit ensuite d'utiliser ce que vous avez chargé depuis ce fichier (`m_height (float *)`, `m_sizeX (unsigned short)`, `m_sizeZ(unsigned short))` c'est à dire utiliser m_sizeX pour la taille en X de la height map, m_sizeZ pour la taille Z et m_height pour les valeurs des hauteurs de chaque vertex.

`m_maxY(float)` étant la hauteur maximum que vous souhaitez avoir pour votre terrain, c'est donc à vous de la définir.

Pensez à bien tout Release sinon vous aurez des leaks et les leaks c'est le mal.

Vous pouvez vous amusez à changer la couleur en fonction de la hauteur.

Utiliser PIX

Tester un peu PIX (cliquer partout toussa).

Pour lancer une expérience, faites File->New Experiment

Dans Program Path mettez le chemin vers l'exécutable.

Pour capturer que une image cocher la 2e case

Dans More Options vous pouvez changer la toucher pour capturer l'image ainsi que le working directory(Startup folder)

NVPerfHud

Pour ceux qui le souhaite et ont une carte NVidia vous pouvez tester NVPerfHud.

Première étape, le télécharger.

Ensuite il faut rajouter ce code avant la création du Device et se servir de num_card en 1er argument et DeviceType en 2e argument de CreateDevice.

```
D3DDEVTYPE DeviceType = D3DDEVTYPE_HAL;

// Look for 'NVIDIA PerfHUD' adapter
// If it is present, override default settings
for (unsigned int adapter = 0; adapter < m_pD3D->GetAdapterCount ();
adapter++)
{
	D3DADAPTER_IDENTIFIER9 identifier;
	HRESULT Res;
	Res = m_pD3D->GetAdapterIdentifier (adapter, 0, &identifier);
	if (strstr (identifier.Description, "PerfHUD") != 0)
	{
		num_card = adapter;
		DeviceType = D3DDEVTYPE_REF;
		break;
	}
}
```