

Shadercomp

Compilation et optimisation de shaders

SOMMAIRE

- ▶ Présentation
- ▶ Objectifs du projet et choix techniques
- ▶ Présentation de nos travaux
- ▶ Démonstration
- ▶ Perspectives d'évolutions

Introduction

- ▶ PFEE en collaboration avec Ubisoft
- ▶ Création de shaders génériques



Présentation

- ▶ Jeux vidéos et les images de synthèse actuels demandent une forte puissance de calculs
- ▶ Utilisation massive des GPUs
- ▶ Nécessité de programmer les GPUs
- ▶ Shaders complexes, différents langages



Matthieu BOURNAT, Antoine
CHESNEAU, Romain GUYOT de la
HARDROUYERE

16/01/2014

Présentation

- ▶ Solution: **ShaderComp**
- ▶ Interface de programmation de shaders
- ▶ Interface nodale
- ▶ Shaders génériques:
 - Possibilité de générer dans différents langages
- ▶ Shaders mathématiques de base disponibles

Présentation

- ▶ Evolutivité:
 - Gestion des plugins shaders en python
 - Gestion des plugins printers en python
- ▶ Génération du shader:
 - Avec aperçu
 - Dans le langage désiré
- ▶ Guide de démarrage rapide détaillé
- ▶ Documentation complète

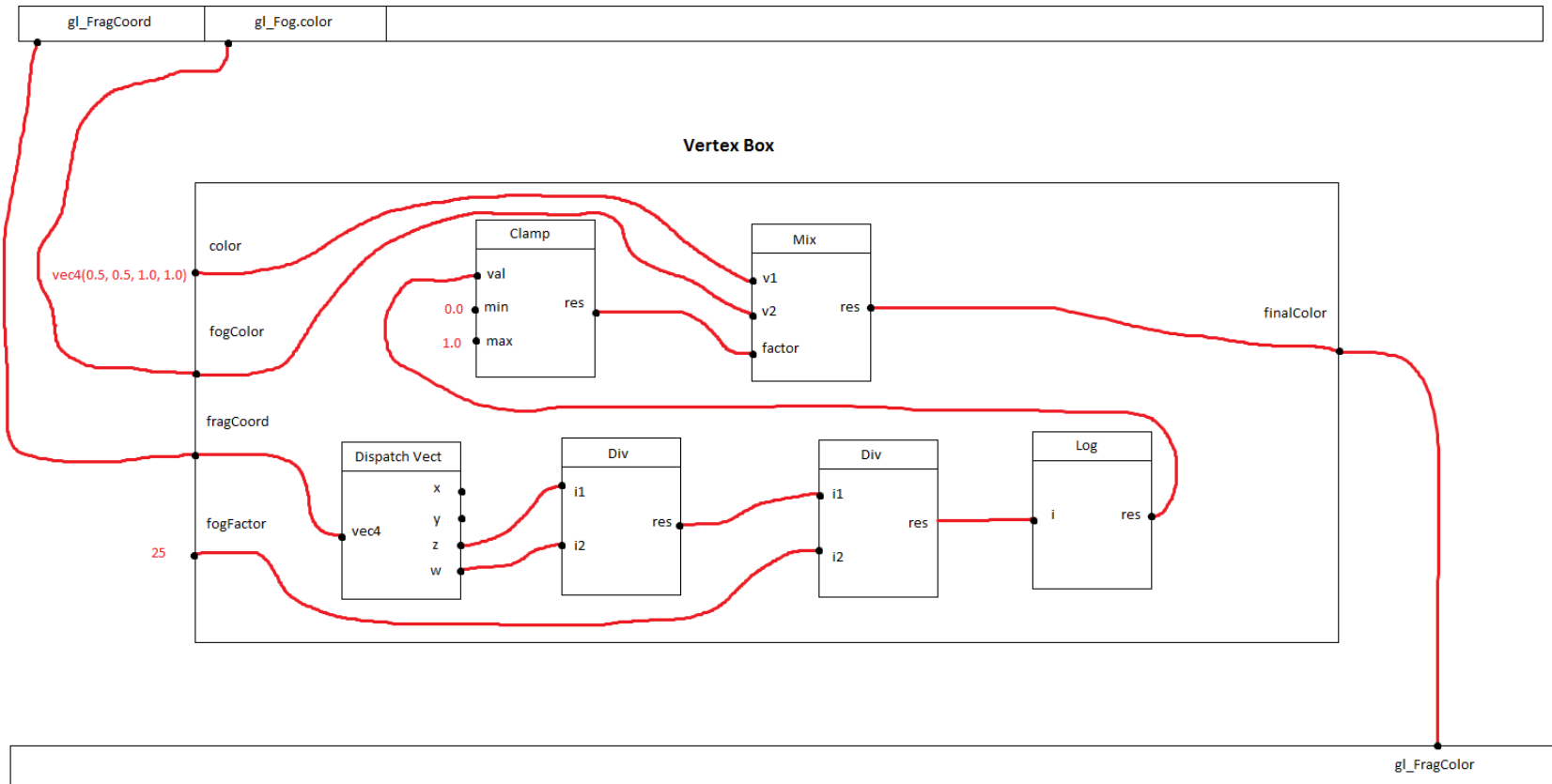
Objectifs du projet

- ▶ Assembler différents shaders de base en un seul shader composé
- ▶ Générer le code de ce shader pour différentes plateformes
- ▶ Recherche d'une structure interne efficace
- ▶ Forte modularité
- ▶ Approfondissement des connaissances des processus d'affichage graphique

Choix techniques

- ▶ Python:
 - Communauté très active
 - De nombreuses bibliothèques
 - Déjà maîtrisé
 - Multi plateforme
- ▶ Structure nodale
- ▶ Un Printer de GLSL:
 - Multi plateforme
 - Déjà connu

Choix techniques



Présentation de nos travaux

- ▶ Une API qui propose trois possibilités:
 - Un aspect utilisateur
 - Un aspect développeur de plugins Shader
 - Un aspect développeur de plugins Printer
- ▶ Tutoriel détaillé pour chacun des aspects

Présentation de nos travaux

Aspect utilisateur

- ▶ Une seule classe comme interface
- ▶ Création d'un réseau de nœuds permettant la réalisation du shader
- ▶ Aperçu du résultat à l'issue de la réalisation

User tutorial Part 1: Your first project

In this tutorial, you have an overview of how the library is working, and you will learn how to use it to create a bootstrap project using the API.

Présentation de nos travaux

Aspect programmeur de plugins Shader

- ▶ Développement en deux étapes:
 - Une classe d'interface
 - Hérite d'une classe générique
 - Une partie générateur par langage cible
 - Génère le code spécifique au langage

ShaderComp

User tutorial ▾

Programmer tutorials ▾

Programmer Tutorial Part 1: Create your own shader!

If you previously read the three parts of the [User Tutorial](#), you may think that the API allows to build some complex networks using built-in shader nodes, but the amount of different nodes may be quite poor for your needs. Don't worry, leave you alone with that. In this tutorial, you will learn how to create plug-ins to extend the features of the library !

Présentation de nos travaux

Aspect programmeur de plugins Printer

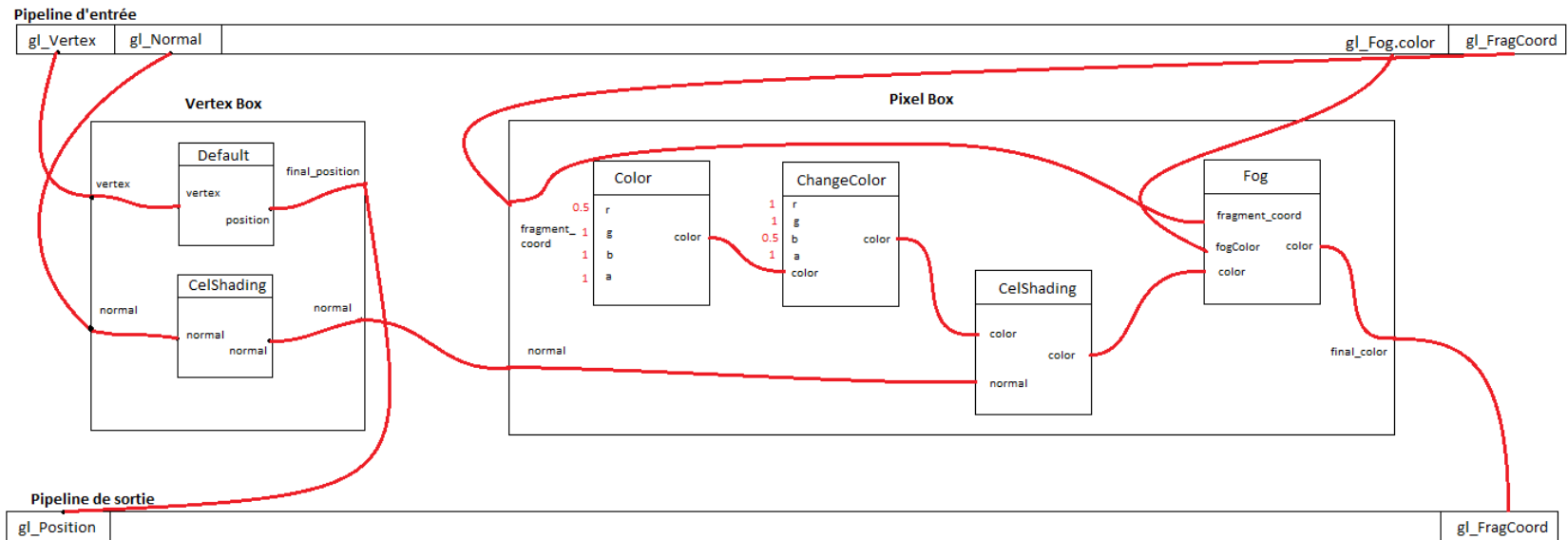
- ▶ Réécriture des procédés de génération spécifiques à un langage
- ▶ Réécriture des générateurs de tous les shaders

Programmer Tutorial Part 2: What about a new platform ?!

The ShaderComp API is created to allow shader programmer to create their shaders independently of the platform the shader will be executed on. That's why, when computing a project, you need to specify the printer to use. In this part of the tutorial, you will see how to create new printers, wich is the second way to extend the features of the library.

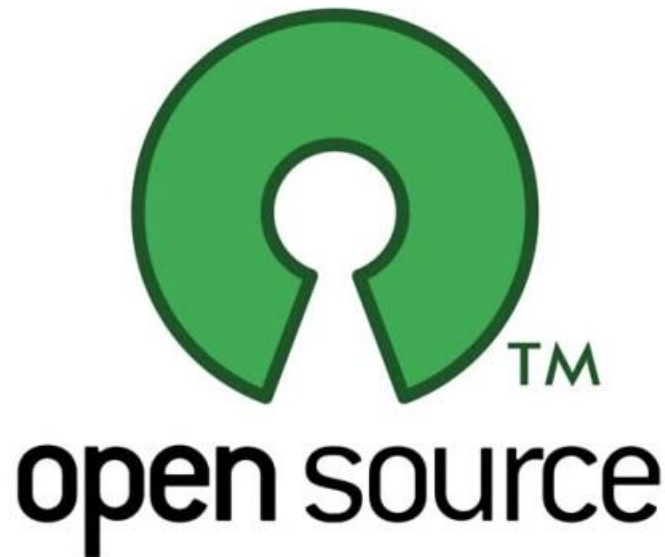
Démonstration

► Interface graphique



Perspectives d'évolution

- ▶ Interface graphique
- ▶ Multiplication des plugins
 - Grâce aux tutoriaux : Communauté OpenSource?
- ▶ Amélioration de l'expérience utilisateur



Conclusion

- ▶ Expérience enrichissante :
 - Connaissances approfondies des shaders et de la programmation GPU
 - Travail en collaboration avec une entreprise
- ▶ De nombreux défis à relever.
- ▶ Au départ, mieux définir les directions du projet

Questions

