
PLAN DE COURS: SGPU

Djihane BABAHENINI

24/03/2019

Tables des matières

I. Informations sur le cours.....	1
II. Présentation du cours	2
III. Contenu.....	4
IV. Pré-requis.....	5
V. Visées d'apprentissage.....	5
VI. Modalités d'évaluation des apprentissages	6
VII. Modalités de fonctionnement.....	7
VIII. Ressources d'aide.....	8

I. Informations sur le cours

Faculté: Sciences Exactes et Sciences de la Nature et de la vie

Département: Informatique

Public cible : 1^{ère} année Master, spécialité Image et Vie Artificielle (IVA)

Intitulé du cours : Shaders sur les GPU (SGPU)

Crédit:04

Coefficient:02

Durée : 16 semaines

Horaire: Lundi: 08h00-11h00

Salle: 02

Enseignant :

Cours et TP: Dr. Djihane BABAHENINI

Contact : par mail au djihane.babahenini@univ-biskra.dz.

Disponibilité :

Au laboratoire : Lundi, Mercredi de 08h30 -12h00, Mardi de 13h00-16h3

Réponse par mail : Je m'engage à répondre par mail dans 24 heures qui suivent la réception du message, sauf en cas des imprévus.

II. Présentation du cours

La synthèse d'image a connu une évolution importante ces dernières années et ses applications sont très variées, ce qui nécessite un large calcul des images de rendu. En effet, le CPU de la machine permet de générer des images dans un temps de calcul et un coût mémoire très élevé. Depuis une dizaine d'années, la synthèse d'image prend une direction vers l'utilisation des cartes graphiques pour accélérer le temps de calcul d'une part et produire des images qualifiées réalistes d'autre part. Un shader est un programme exécuté non pas par le processeur de la machine mais plutôt par la carte graphique, cette dernière qui nous aide à produire des images ou des scènes complexes dans un temps de calcul réduit.

La programmation avec les shaders est très utilisée aujourd'hui en synthèse d'images, plus particulièrement dans les domaines qui nécessitent un temps de calcul interactif tels que le domaine des jeux vidéo, la reconstruction 3D, et la réalité virtuelle.

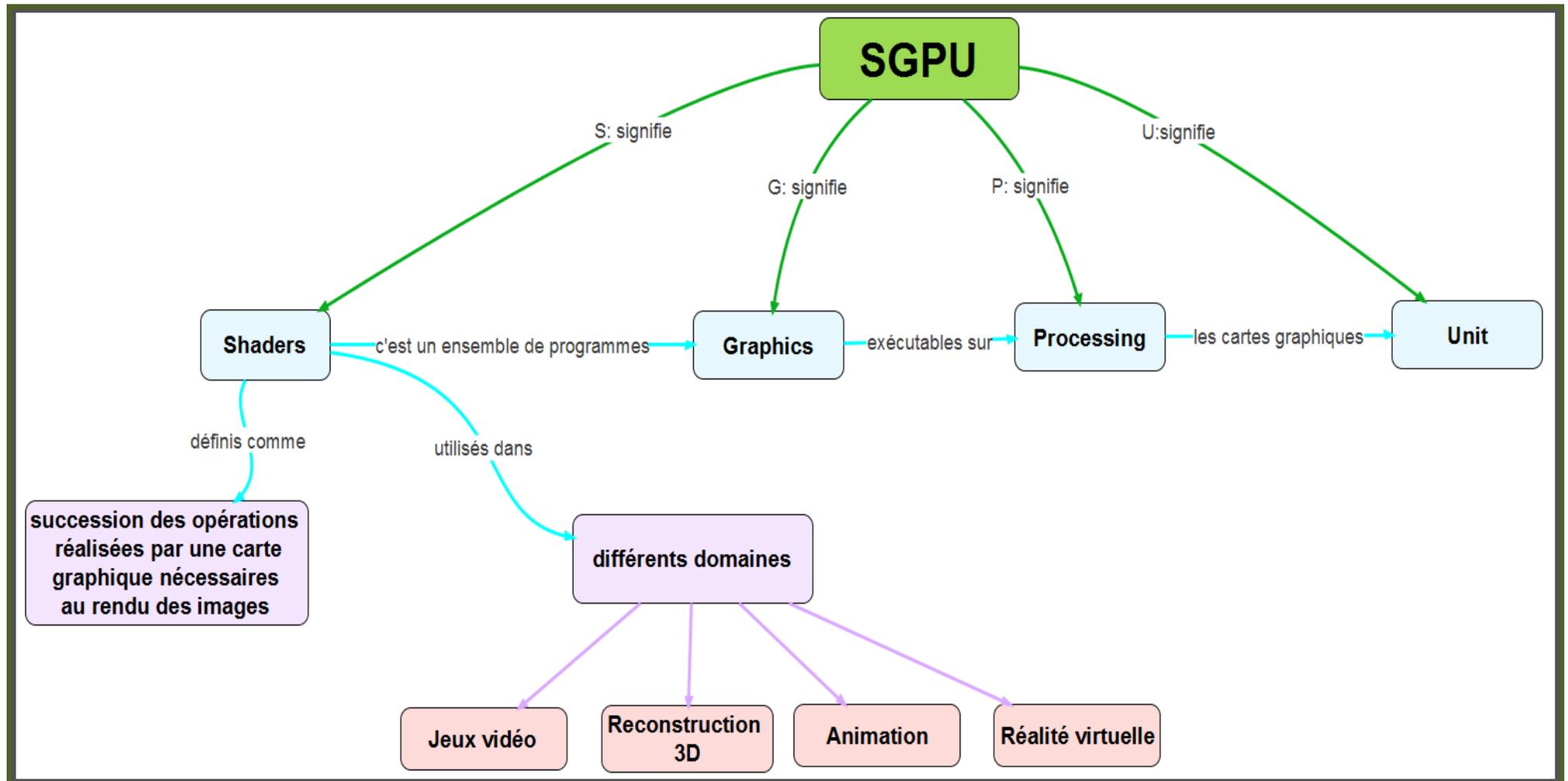
Ce cours intitulé le « SGPU » permet de vous donner un aperçu sur les GPU (architecture et programmation du pipeline graphique), il traite aussi les shaders dans le domaine de la synthèse d'images

Le cours est scindé en un ensemble d'unités d'apprentissage qui vous permettent d'acquérir des compétences en matière d'utilisation des shaders pour le rendu des scènes 3D complexes en synthèse d'images. Il permet de vous familiariser avec des nouvelles techniques avancées de synthèse d'images vues en 1^{er} semestre.

Il vous permet, également, la programmation avec le langage GLSL (OpenGL Shading Language)

La figure 1 montre la définition de SGPU et les domaines d'utilisation des shaders:

Figure 1- définition de SGPU et domaines d'utilisation des shaders



III. Contenu

Le cours est scindé en quatre unités d'apprentissages, chaque unité d'apprentissage est traitée à travers des séquences pédagogiques permettant l'accompagnement de l'étudiant à bien comprendre les notions de shaders, GPU, synthèse d'images,....., cette accompagnement est consolidée par des activités d'apprentissages et des travaux pratiques où ces notions sont mises en œuvre, c'est une des forces de ce cours. Ce polycopié pourra constituer un instrument efficace pour l'étudiant. L'ensemble des unités d'apprentissage sont décrites au-dessous:

a. Présentation générale de la synthèse d'image

Cette unité est considérée comme un rappel sur les notions de base de la synthèse d'image qui sont étudiées dans le premier semestre de l'année courante. Ces notions concernent l'historique de la synthèse d'image, comment construire une scène 3D?, les transformations géométriques, et le système de projection.

b. Le pipeline graphique

Le pipeline graphique de rendu est une succession d'étapes exécutables sur les GPUs permettant l'accélération de temps de rendu. Il existe deux types de pipeline: pipeline classique avec des étapes statiques et pipeline programmable où le programmeur peut interagir avec certaines de ses étapes.

Dans cette partie les différentes unités de conception sont présentées avec des activités d'apprentissage qui vous permettent l'assimilation des différents concepts.

c. Programmation GPU par les shaders

Dans cette partie, il s'agit de présente les différents notions et types de shaders:

- Qu'est ce qu'un shader?
- Langage de programmation des shaders
- Les shaders en GLSL

IV. Pré-requis

Avant de commencer le cours des shaders sur GPU, les étudiants doivent avoir une connaissance approfondie de la synthèse d'image. ils peuvent rafraîchir ces connaissances par la lecture des documents sur l'infographie et la modélisation. Afin de pouvoir tirer le maximum de ce cours il faut connaître :

- ✓ Les notions de base des mathématique, vecteurs, matrices, intégrales,....
- ✓ Les Algorithmes et l'Architecture Parallèle.
- ✓ Le langage de programmation C++ et la librairie OpenGL.

V. Visées d'apprentissage

La compétence visée par ce cours, dans son ensemble, est « *d'être capable de concevoir, et d'implémenter des applications en relation avec le rendu et la synthèse d'image en vous appuyant sur les shaders et le langage GLSL* ».

- ✓ En termes de connaissances,
 - définir les notions de base de la synthèse d'image conduit l'étudiant à bien suivre le cours.
 - apprendre les notion de pipeline graphique, et des shaders permettant la compréhension de fonctionnement des shaders en côté théorique.
- ✓ En termes de savoir-faire,
 - savoir utiliser les shaders pour accélérer le temps de calcul de rendu.
 - savoir implémenter des programmes avec le langage C++ et GLSL.
- ✓ En termes de savoir-être,
 - savoir réaliser des projets collectifs (présentation, consultation avec l'enseignant).
 - savoir communiquer avec ses collègues et avec l'enseignant à travers la formation à distance.

VI. Modalités d'évaluation des apprentissages

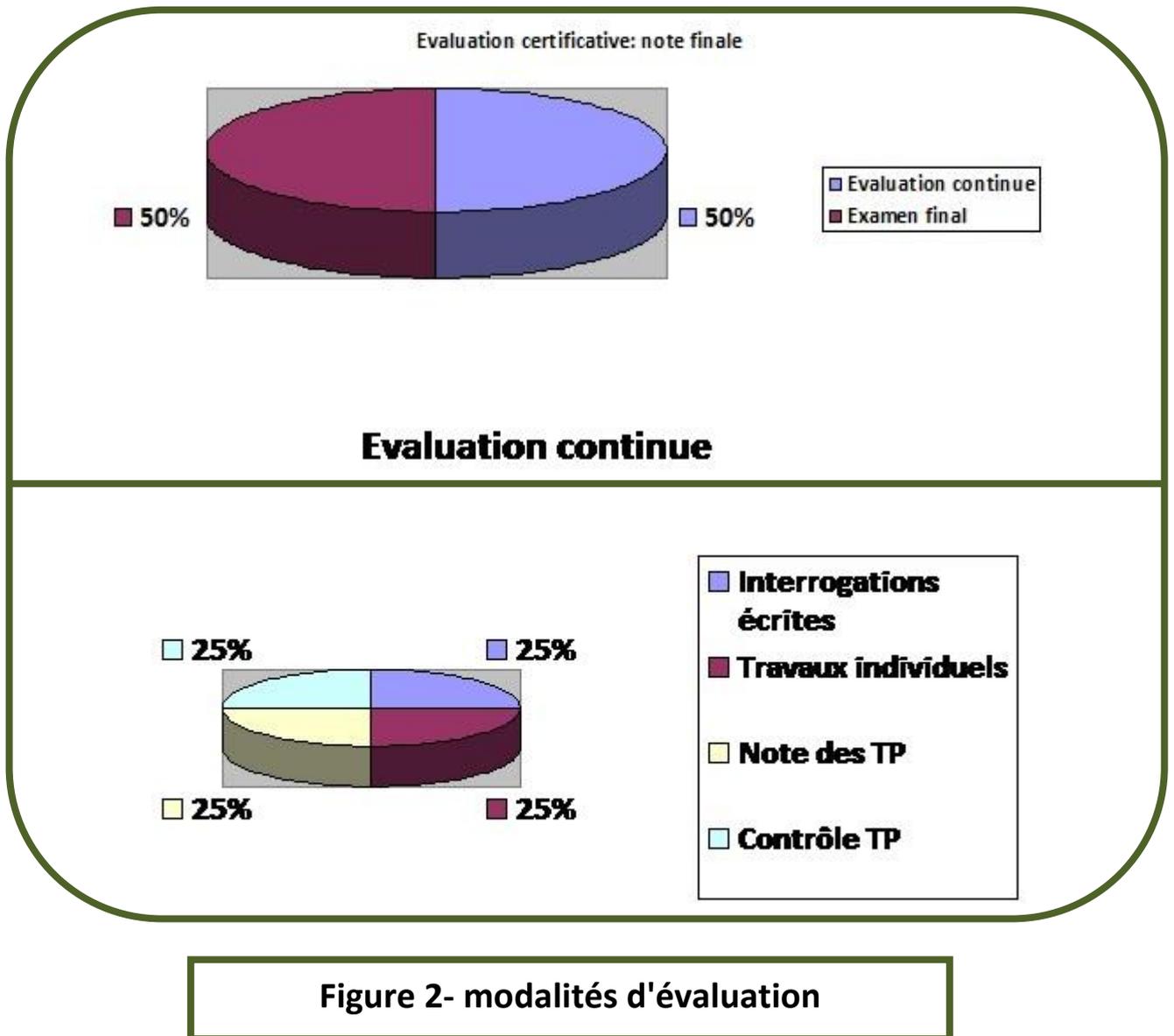
L'évaluation finale se fait à travers:

- a. Un examen final sur table et qui porte sur tout ce que vous avez vu dans ce cours pendant le semestre. L'examen compte pour 50% de la note finale (voir figure 2), vous aurez

- ✓ À résoudre des exercices proches aux exercices traités lors des séances de cours, des TPs et des interrogations.
- ✓ À répondre à des questions de cours

- b. Évaluation continue et régulières de 50% restant, elle est divisée en deux parties: une note de TD de 25% et une autre de TP de 25%.

- Note de TD: il s'agit de la moyenne des notes des interrogations écrites, et des travaux individuels.
- Note de TP: il s'agit de la moyenne des notes des TP et d'un contrôle TP.



VII. Modalités de fonctionnement

Le cours est organisé en :

- ❖ Séances théoriques afin de vous familiariser avec les concepts théoriques de la synthèse d'image et des shaders. Les initiations de ce cours sont déjà présentées en premier semestre (cours *Introduction à la synthèse d'image*). Ces séances seront enrichies par des exercices et des activités d'apprentissage avant de mieux comprendre le cours.
- ❖ En séances de travaux pratiques afin de concevoir et de mettre en œuvre quelques types d'éclairage et de rendu des scènes 3D, tous en utilisant les shaders.

VIII. Ressources d'aide

Voici quelques ressources pour vous aider à un bon suivi du cours:

1. Moya, C. Gonzalez, J. Roca, A. Fernandez and R. Espasa, "Shader performance analysis on a modern GPU architecture," *Microarchitecture, 2005. MICRO-38. Proceedings. 38th Annual IEEE/ACM International Symposium on*, Barcelona, 2005, pp. 10 pp.-364.
doi: 10.1109/MICRO.2005.30
2. McCool, M., Du Toit, S., Popa, T., Chan, B., & Moule, K. (2004, August). Shader algebra. *ACM Transactions on Graphics (TOG)* (Vol. 23, No. 3, pp. 787-795). ACM.
- Upstill, Steve. *The RenderMan Companion: A Programmer's Guide to Realistic Computer Graphics*. Addison-Wesley. ISBN 0-201-50868-0.
3. Ebert, David S; Musgrave, F. Kenton; Peachey, Darwyn; Perlin, Ken; Worley, Steven. *Texturing and modeling: a procedural approach*. AP Professional. ISBN 0-12-228730-4.
4. Fernando, Randima; Kilgard, Mark. *The Cg Tutorial: The Definitive Guide to Programmable Real-Time Graphics*. Addison-Wesley Professional. ISBN 0-321-19496-9.