

Synthèse d'Images

Informatique Graphique

G.-P. Bonneau

Planning

- Cours (10 séances)
 - Mardi de 10h15 à 11h45
 - distanciel
 - Georges-Pierre.Bonneau AT inria.fr
- TD machine (10 séances)
 - distanciel
 - voir ADE

Métiers de l'Informatique Graphique

Films d'Animation

- 2006: achat de Pixar par Walt Disney pour 7,4 Mds de \$; 850 salariés
- Dreamworks (Shrek), CA 725 Mo \$; 1940 salariés



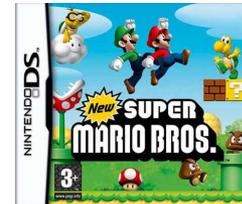
Glendale, Californie



Emeryville, Californie

Jeux Vidéos

- Fabricants: Nintendo, Sony, Microsoft...
- Ventes 2010
 - 42,4 millions de PlayStation 3
 - 46,4 millions de Xbox 360
 - 77,7 millions de Nintendo Wii
- Editeurs: EA, Ubisoft, Atari...



Photographie numérique

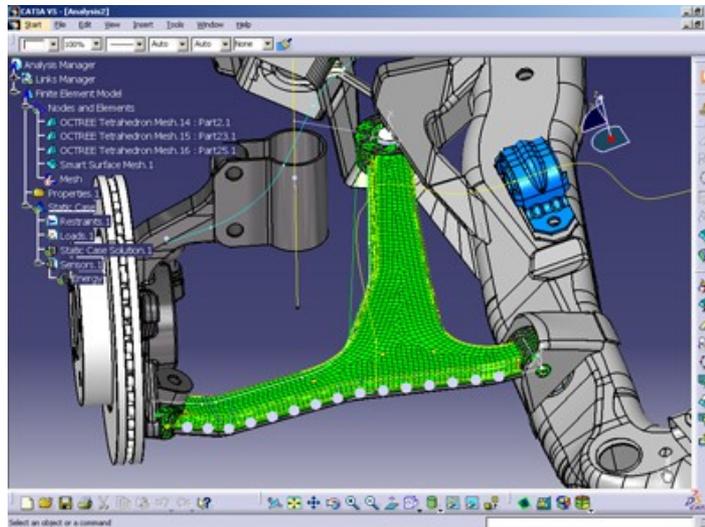


Images HDR



C.A.O.

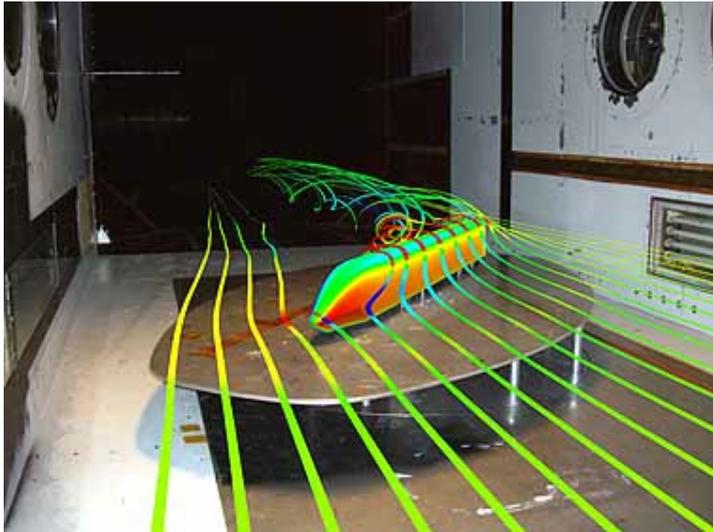
- CATIA, Dassault Systèmes, CA 1,3 Mds d'Euros



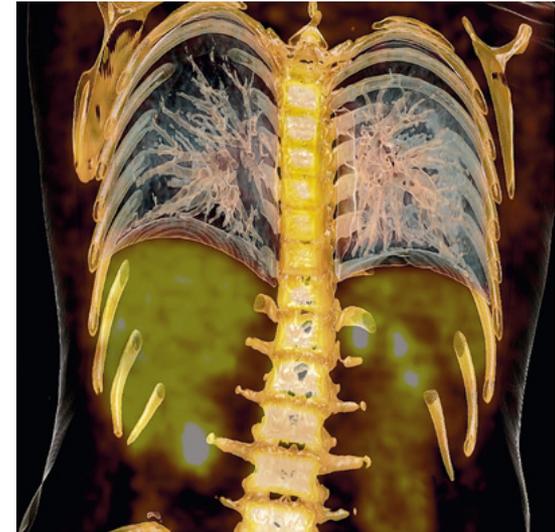
ECP

Visualisation

- Simulation, Observation, Ingénierie, Médecine

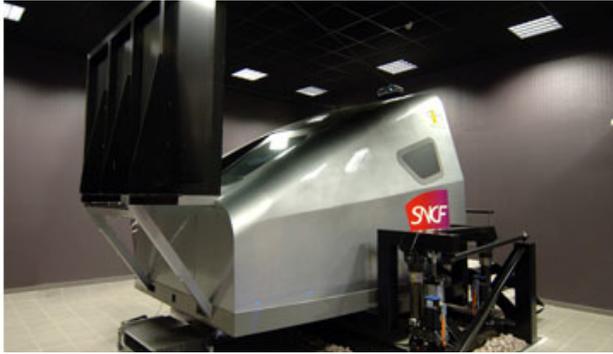


DLR

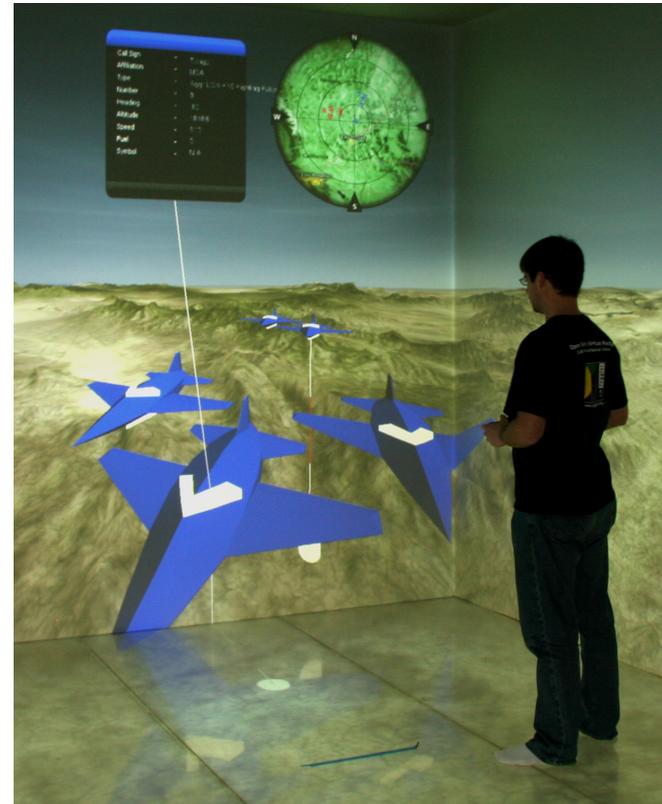
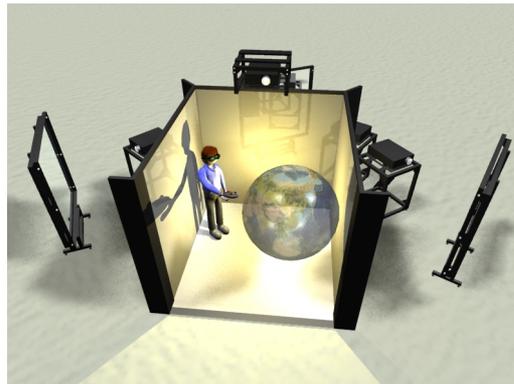


Siemens

Réalité Virtuelle, Simulateurs



Corys, société Grenobloise



Affichage immersif (CAVE)

Introduction à OpenGL

- Bibliothèques graphiques: OpenGL, GLFW, GLEW, GLM ...
- API en C++ et SHADER GLSL
- Interface entre
 - langage haut-niveau
 - (pilote) carte graphique

Documentation OpenGL

- www.opengl.org
- Pleins de cours/tutoriaux sur Internet!
 - Tuto: <http://antongerdelan.net/opengl/>
 - Tuto: <http://www.opengl-tutorial.org>

Cours Computer Graphics

- Stanford: <http://cs148.stanford.edu>
- Berkeley: <http://www-inst.cs.berkeley.edu/~cs184/fa16/schedule.html>
- MIT: <http://ocw.mit.edu> puis rechercher Computer Graphics

Matériel et Logiciel en salle F202

- PC
 - Carte Nvidia Quadro FX 580
 - 512 Mb
 - 32 cœurs
 - Ecrans DELL 19", 1280x1024
- Logiciel
 - OS Linux, compilateur gnu gcc
 - Librairie GLFW

Compiler les exemples du cours (Linux)

- Installer les dernières version de
 - build-essential, libglfw3-dev, libglew-dev, libglm-dev, mesa-utils,
 - `sudo apt-get install <nameOfPackage>`
- Télécharger les exemples du cours
- Compiler et exécuter les exemples
 - `mkdir build; cd build; cmake ..; make`
 - `./si_info4`

Compiler les exemples du cours (OSX)

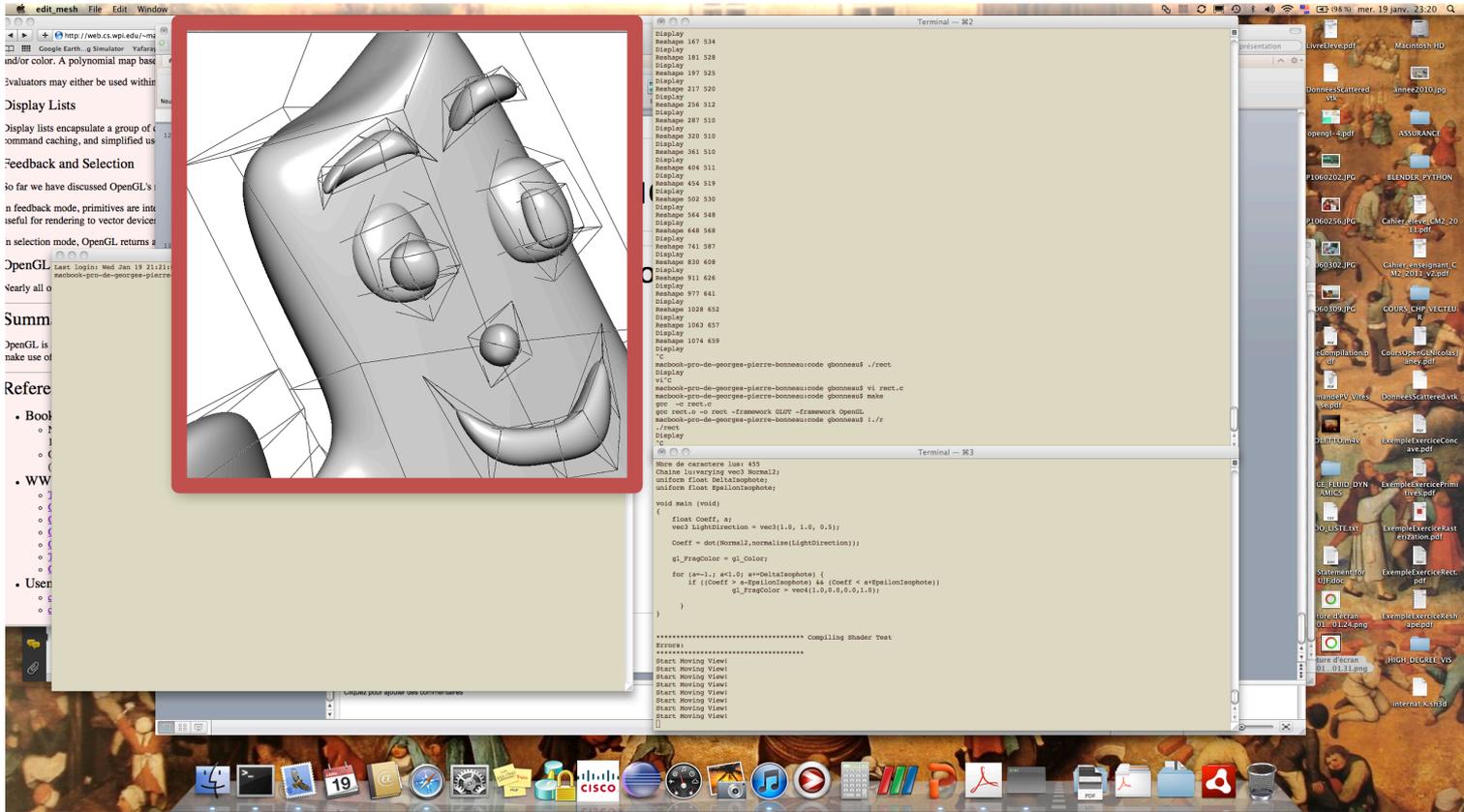
- Installer Homebrew [<https://brew.sh/>]
- Installer glfw3, glew, glm, cmake, pkg-config:
`brew install glfw glm make pkg-config`
- Télécharger les exemples du cours
- Compiler et exécuter les exemples
 - `mkdir build; cd build; cmake ..; make`
 - `./si_info4`

IDE Conseillée pour C++/OpenGL

– Visual Studio Code, avec les extensions:

- C/C++
- Cmake
- CMakeLists.txt
- GLSL lint

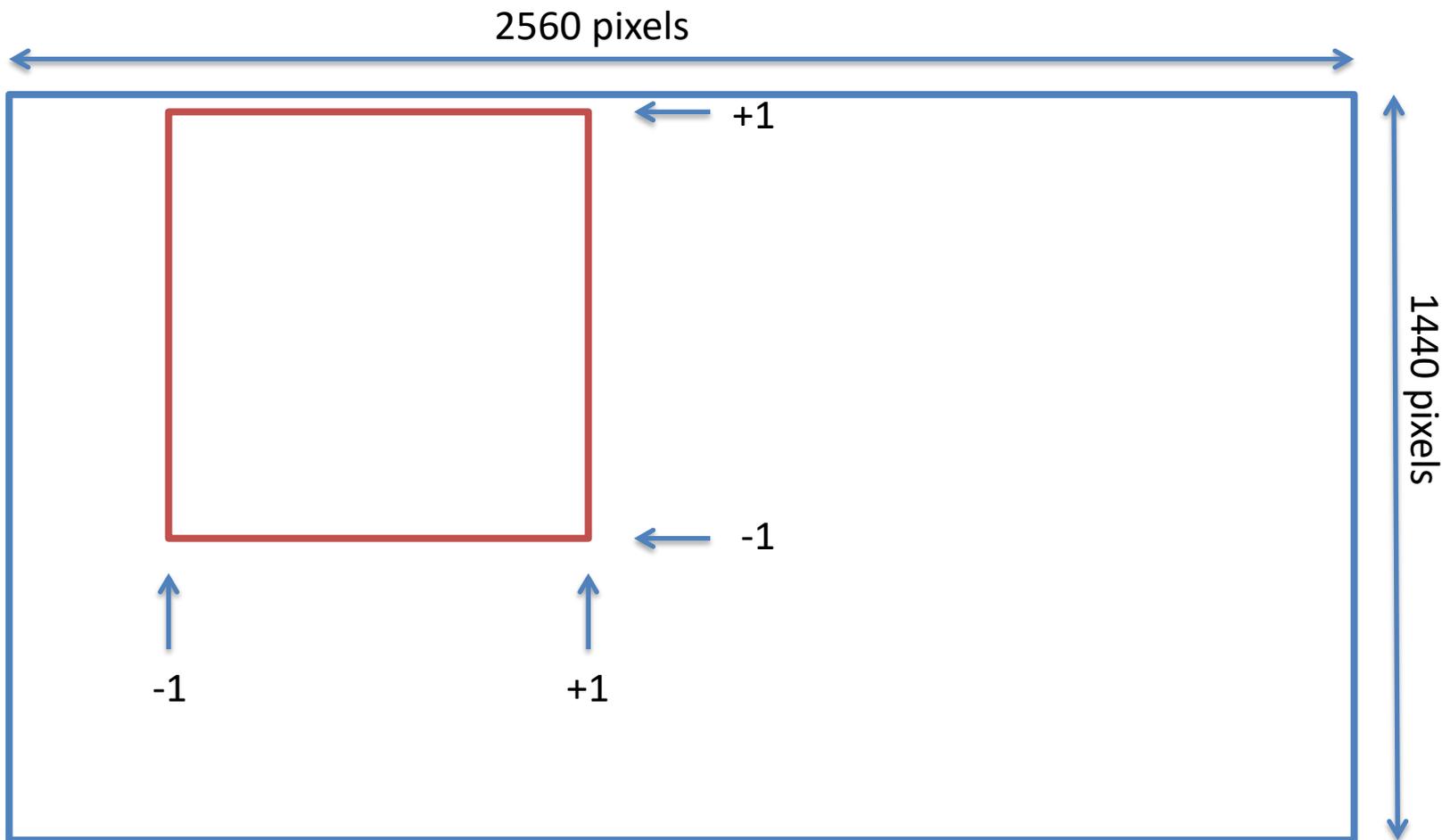
2560 pixels



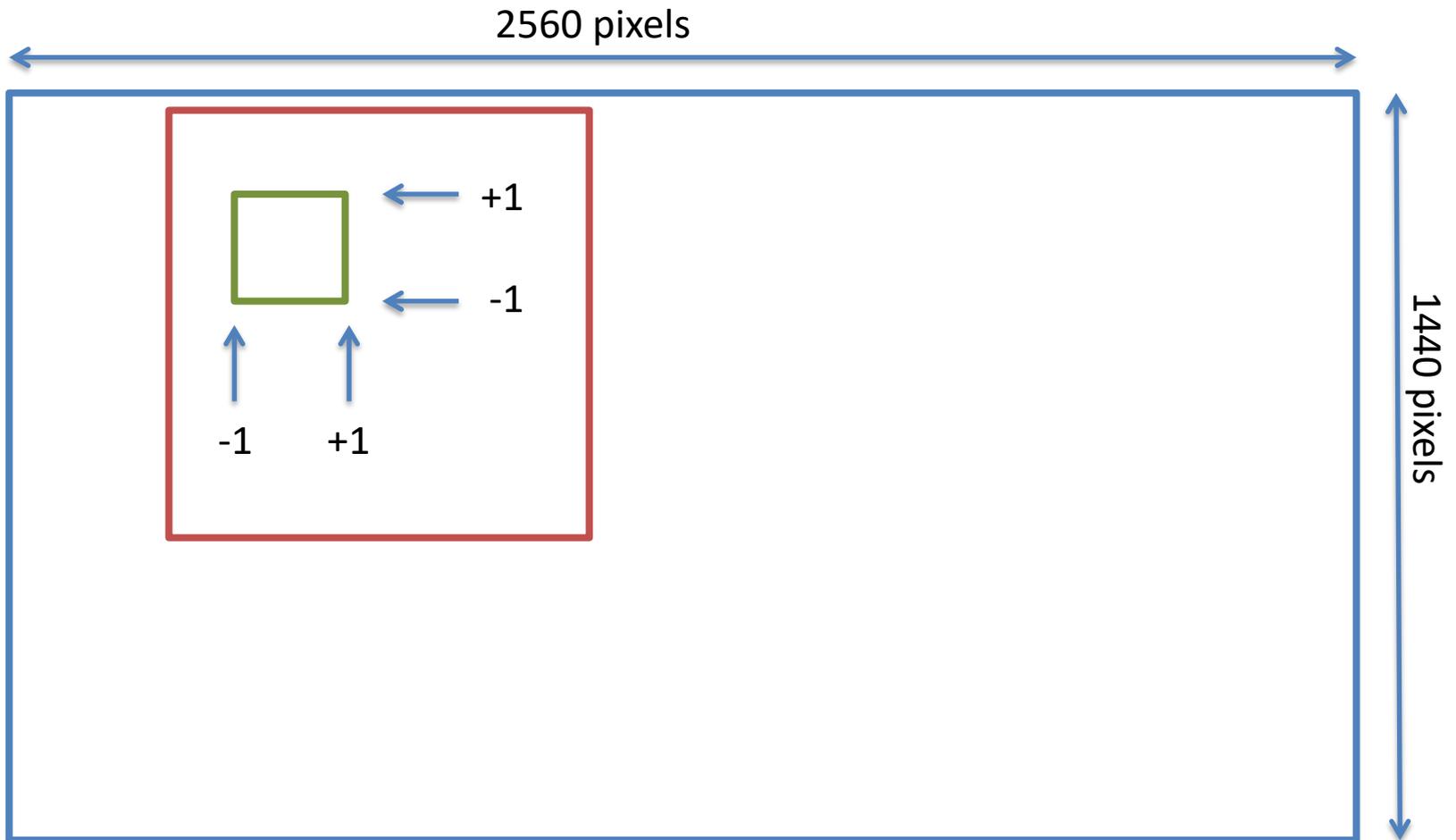
1440 pixels



Fenêtre active, 800x800 pixels



Fenêtre active, 800x800 pixels



Fenêtre active, 800x800 pixels



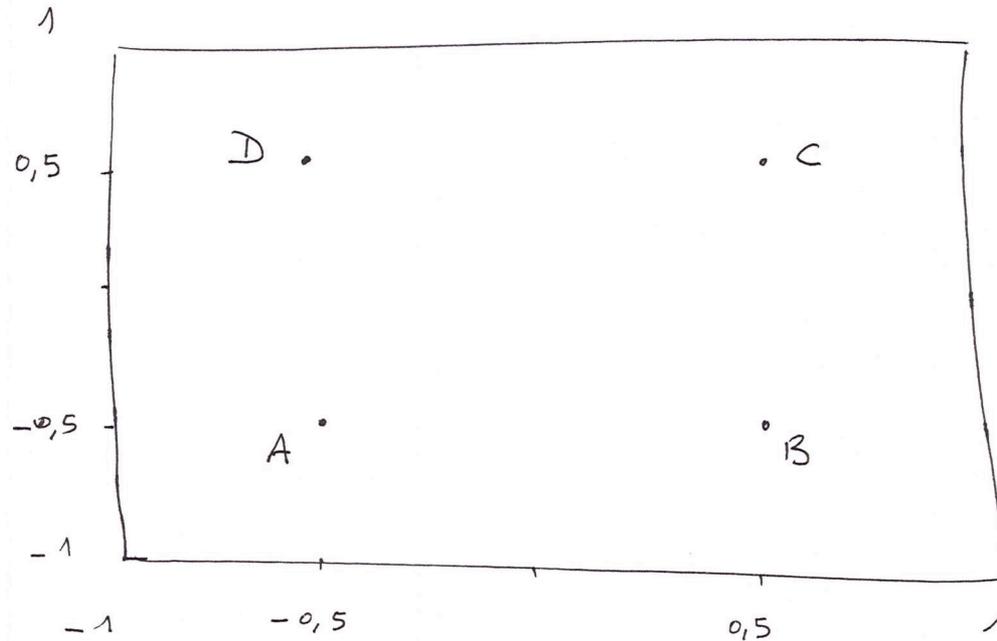
Viewport, 200x200 pixels

Un premier exemple

- Démo dessin d'un carré dans une fenêtre rectangulaire
 - Transformation nécessaire pour que le carré apparaisse comme un carré
 - Gestion des transformations quand la fenêtre change de taille
 - Le carré reste un carré, sa taille s'adapte à celle de la fenêtre
 - Le carré reste un carré de taille constante

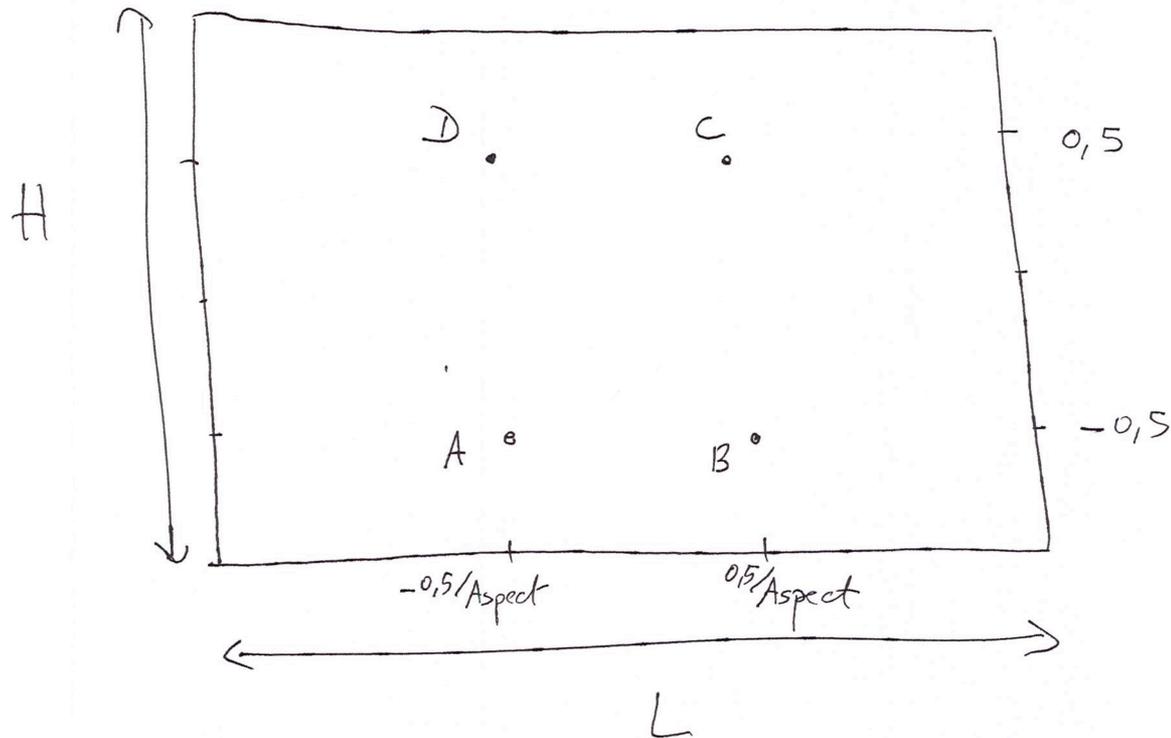
Coordonnées d'un carré dans une fenêtre rectangulaire

"Carre"
 $A = \begin{pmatrix} -1/2 \\ -1/2 \end{pmatrix}$ $B = \begin{pmatrix} 1/2 \\ -1/2 \end{pmatrix}$ $C = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}$ $D = \begin{pmatrix} -1/2 \\ 1/2 \end{pmatrix}$

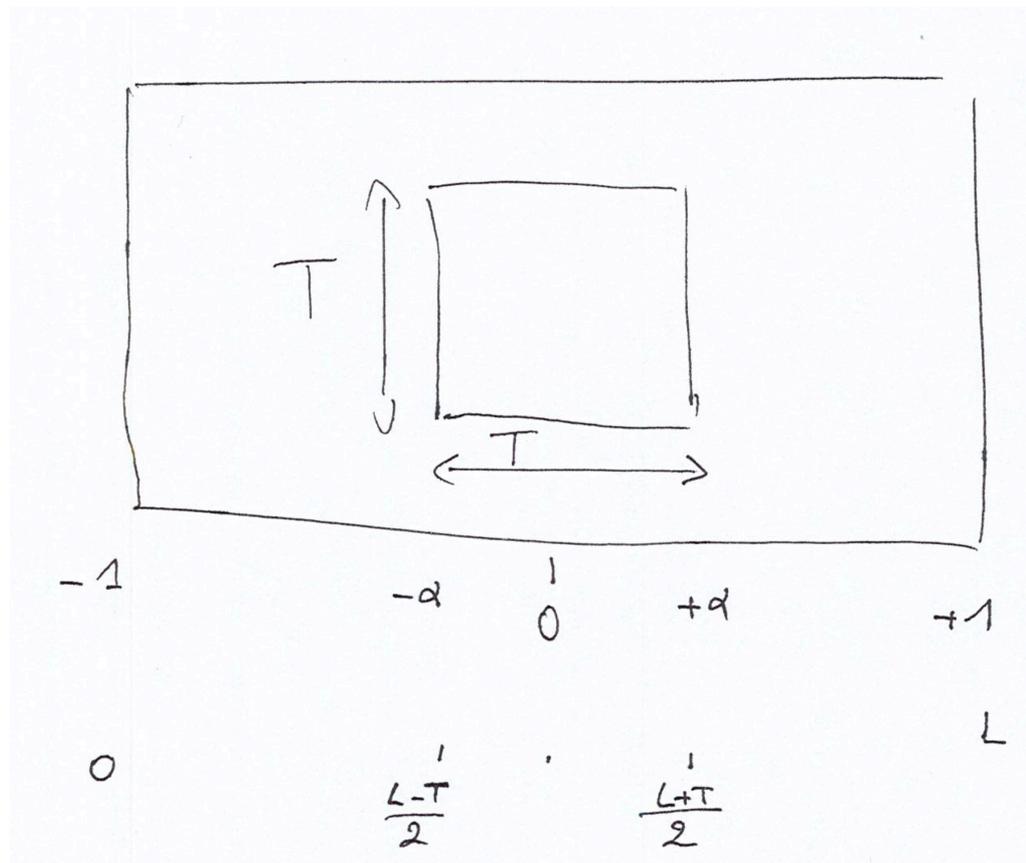


Le carré apparaît comme un carré

$$\text{Aspect} = \frac{L}{H}$$



Carré de taille constante



Continu/Discret

Monde mathématique

Géométrie

Point

Ligne

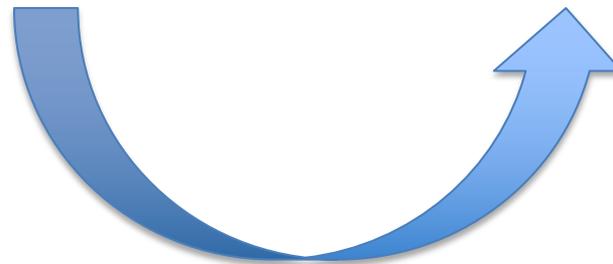
Polygone

Coordonnées réelles $\pi/3, \sqrt{5}$

Traduction à l'écran

Fragment, pixel

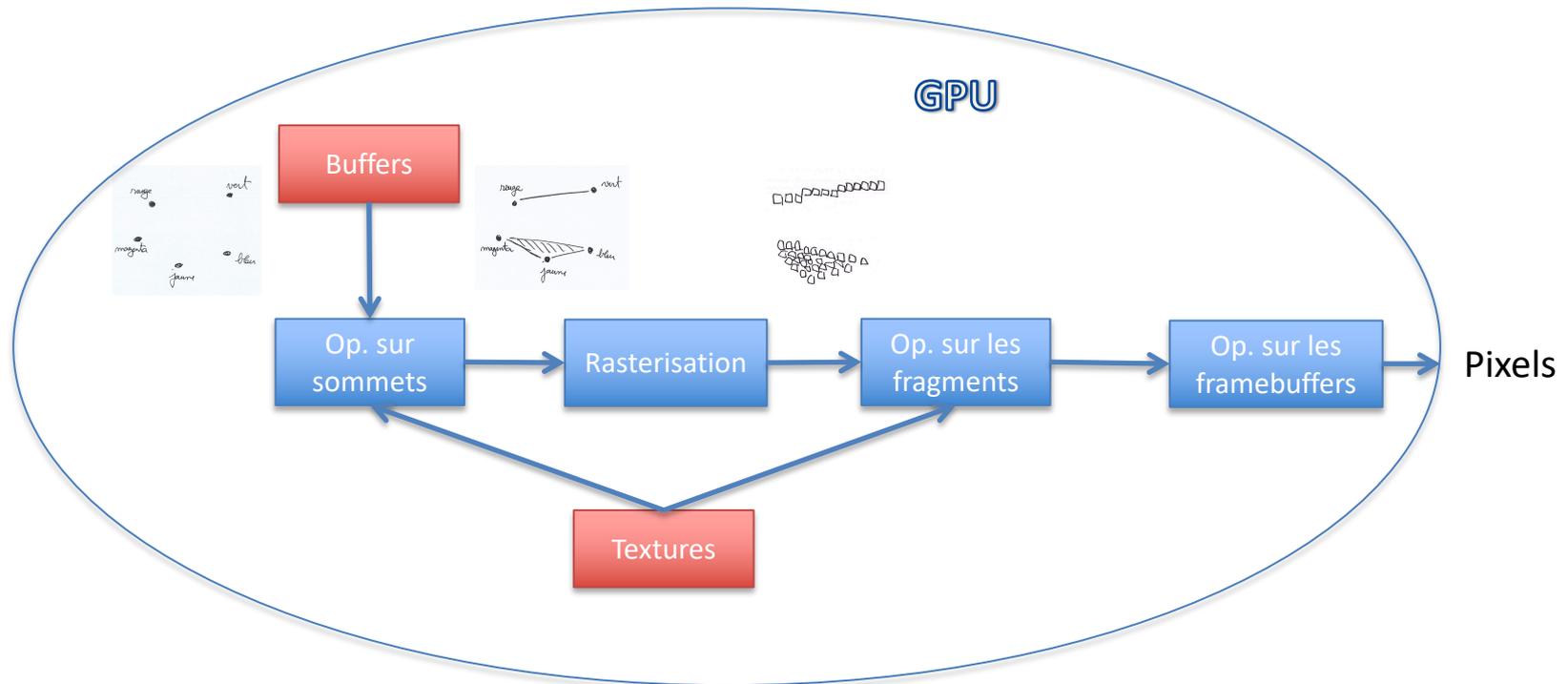
Coordonnées entières 13, 211



Rasterisation

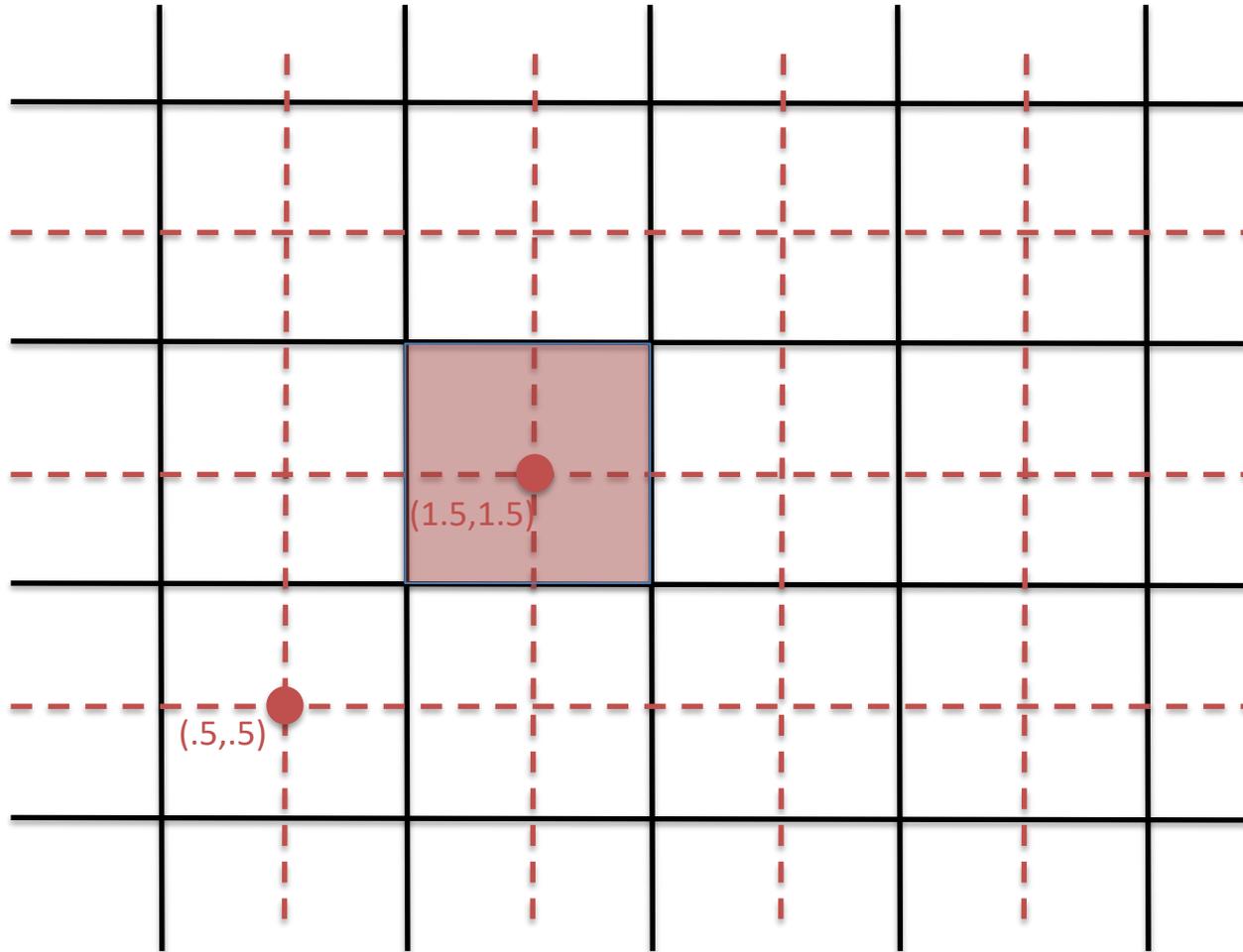
Pipeline OpenGL simplifié

De la **PRIMITIVE GRAPHIQUE** (polygones, lignes, points, image) au **PIXEL**



CPU: initialise les buffers, les textures et les calculs sur le GPU

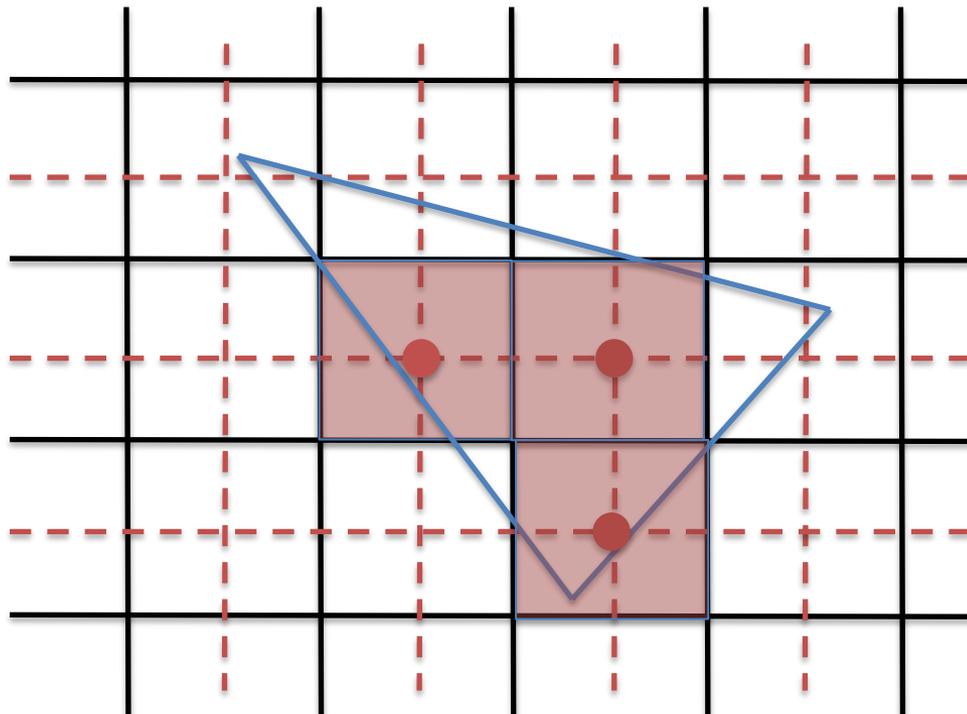
Coordonnées des Pixels



En OpenGL, les centres des pixels ont des coordonnées $i + 0.5$, i entier.

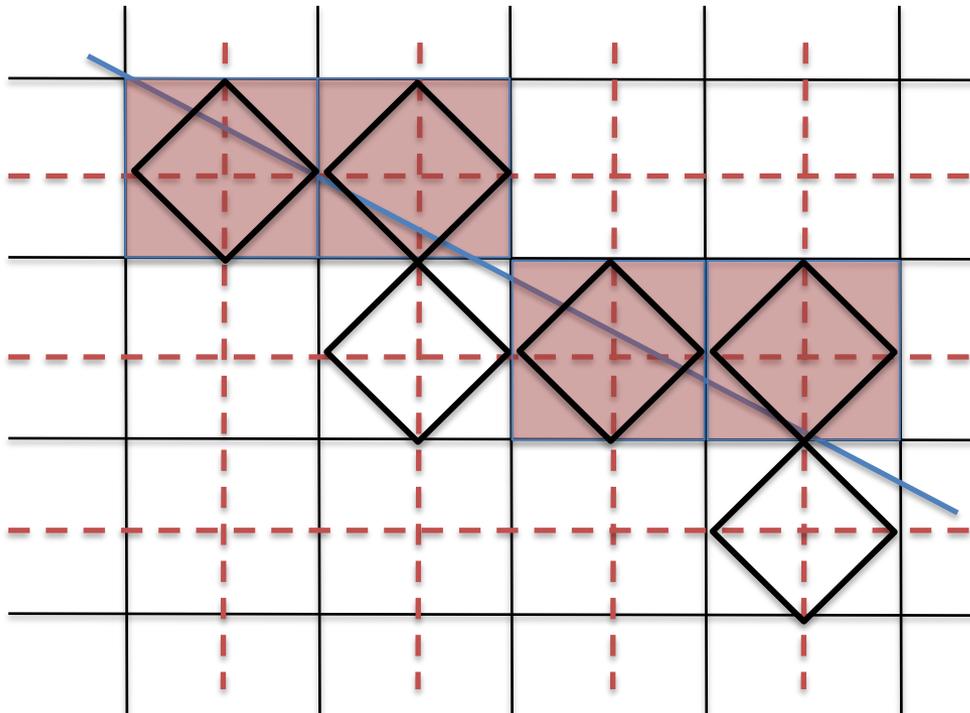
Remplissage de polygones

- Un fragment est généré si le centre du pixel est à l'intérieur de la région

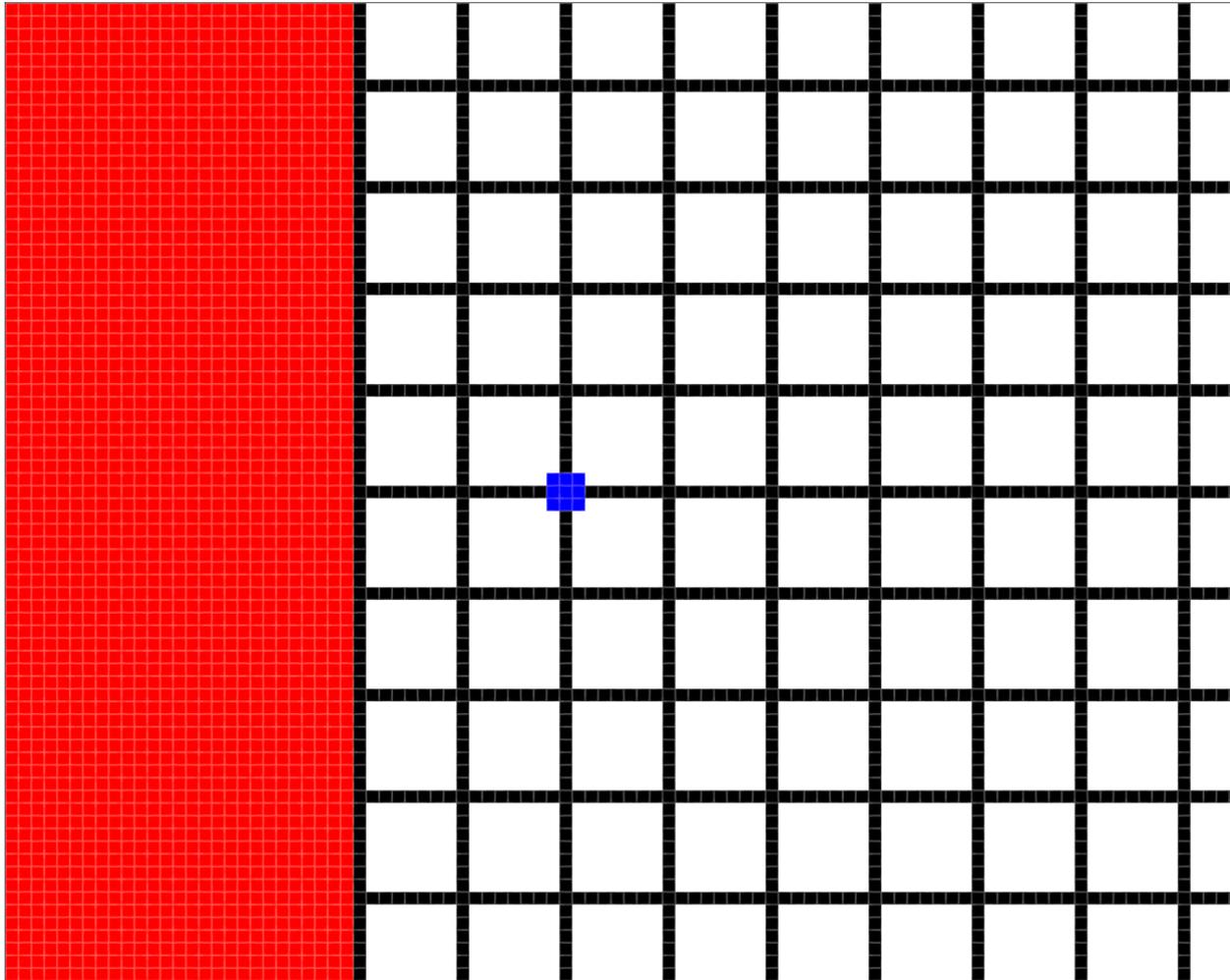


Dessin de lignes

- Un fragment est généré si la ligne intersecte le losange



Demo 1 règles de rasterisation



Point bleu (400.5, 256.5)
Point d'épaisseur 3
⇒ 9 fragments pour ce point

Ligne de taille 1, passant par
les centres des pixels

Polygone carre rouge dans les
sommets ont des coordonnées
entières

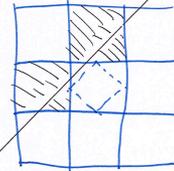
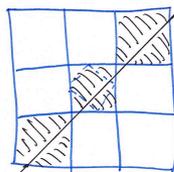
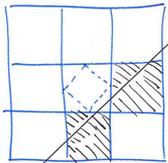
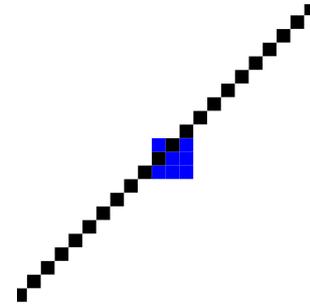
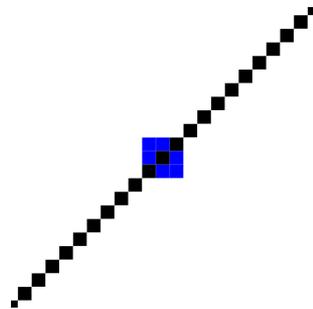
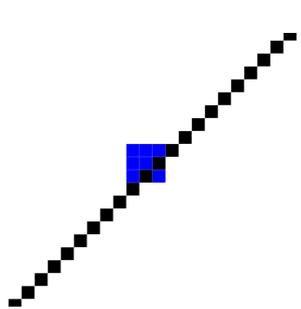
Demo 2 règles de rasterisation

Point bleu (255.5, 255.5)

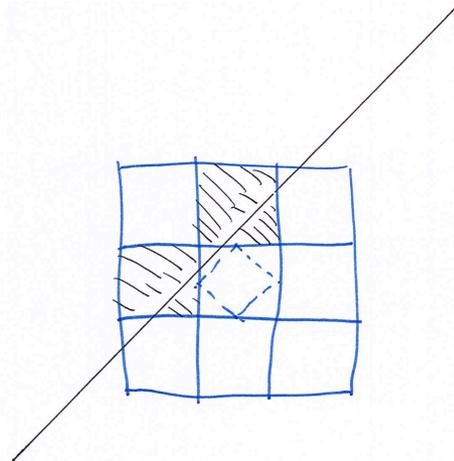
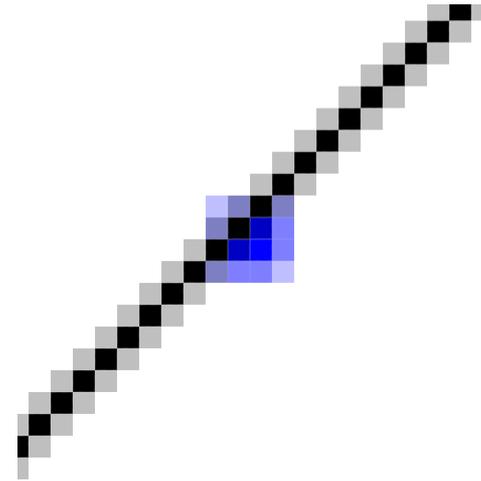
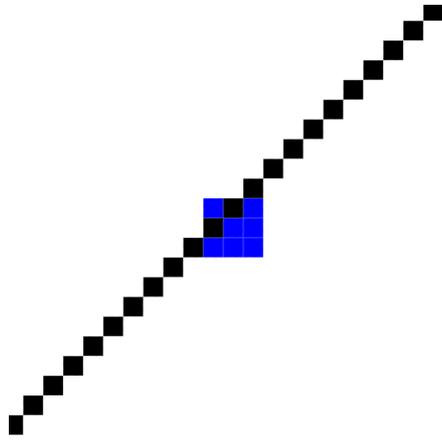
Point d'épaisseur 3

⇒ 9 fragments pour ce point

3 lignes de même pente

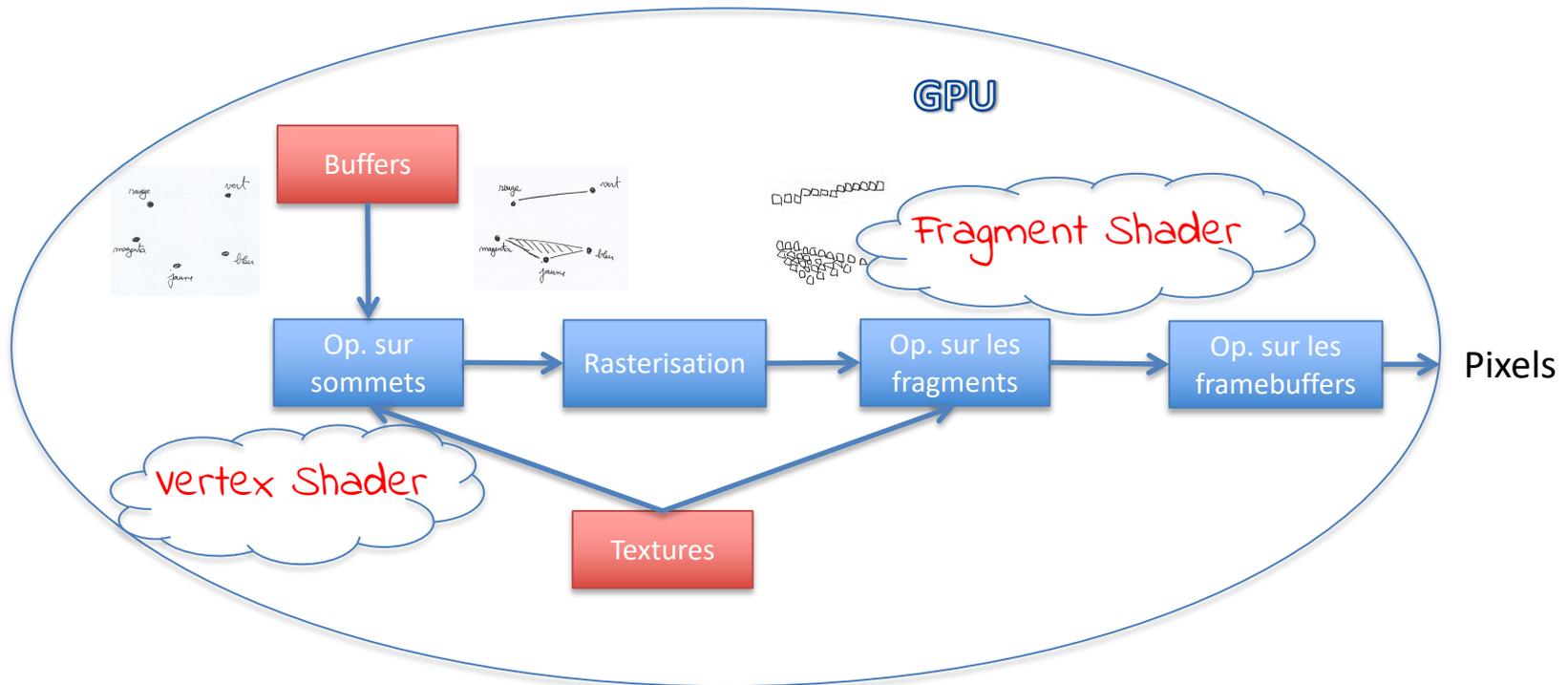


Demo 2 avec Anti-Aliasing



Pipeline OpenGL simplifié

De la **PRIMITIVE GRAPHIQUE** (polygones, lignes, points, image) au **PIXEL**



CPU: initialise les buffers, les textures et les calculs sur le GPU

Shader GLSL

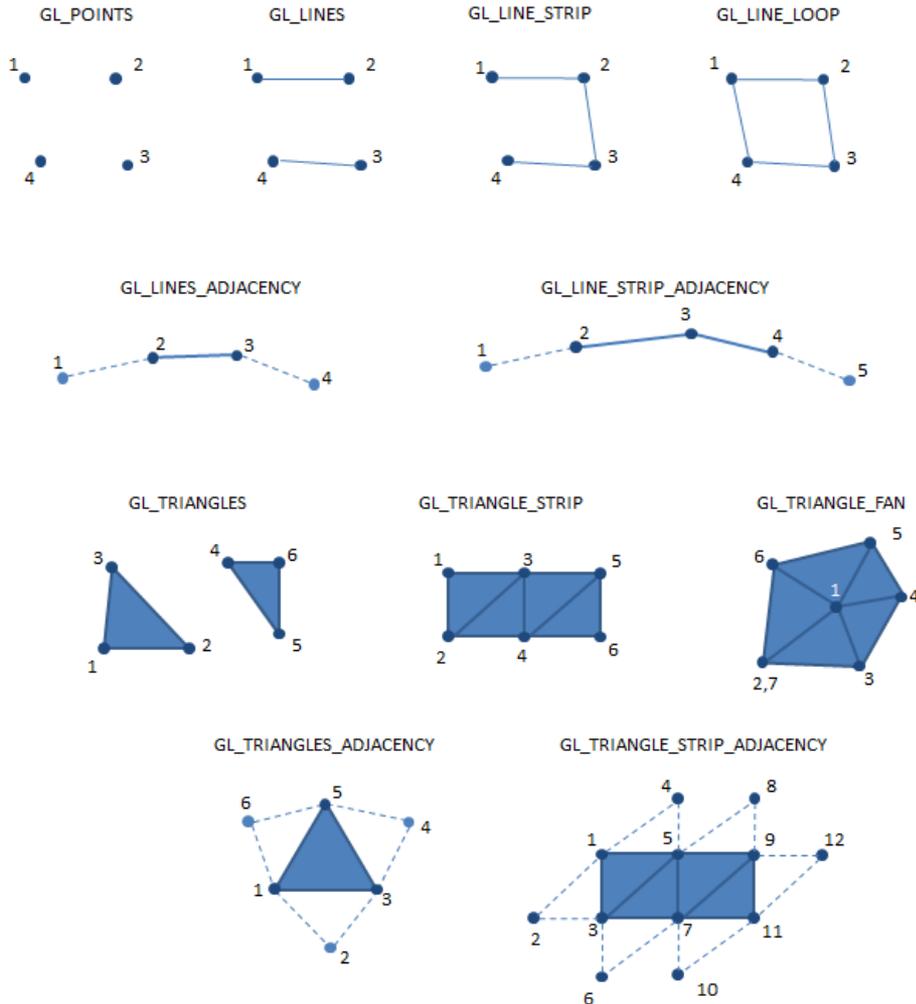


GTX 970 Engine Specs:
CUDA Cores

1664

- Variables Uniformes
`uniform mat4 MaMatrice;`
- Variables dépendantes du sommet/fragment
`layout(location = 0) in vec4 MonSommet;`
`layout(location = 1) out vec4 MaCouleur;`
- Passage de variables entre Shader:
 - Interpolation des variables par le Rasteriser

11 + 1 Primitives OpenGL

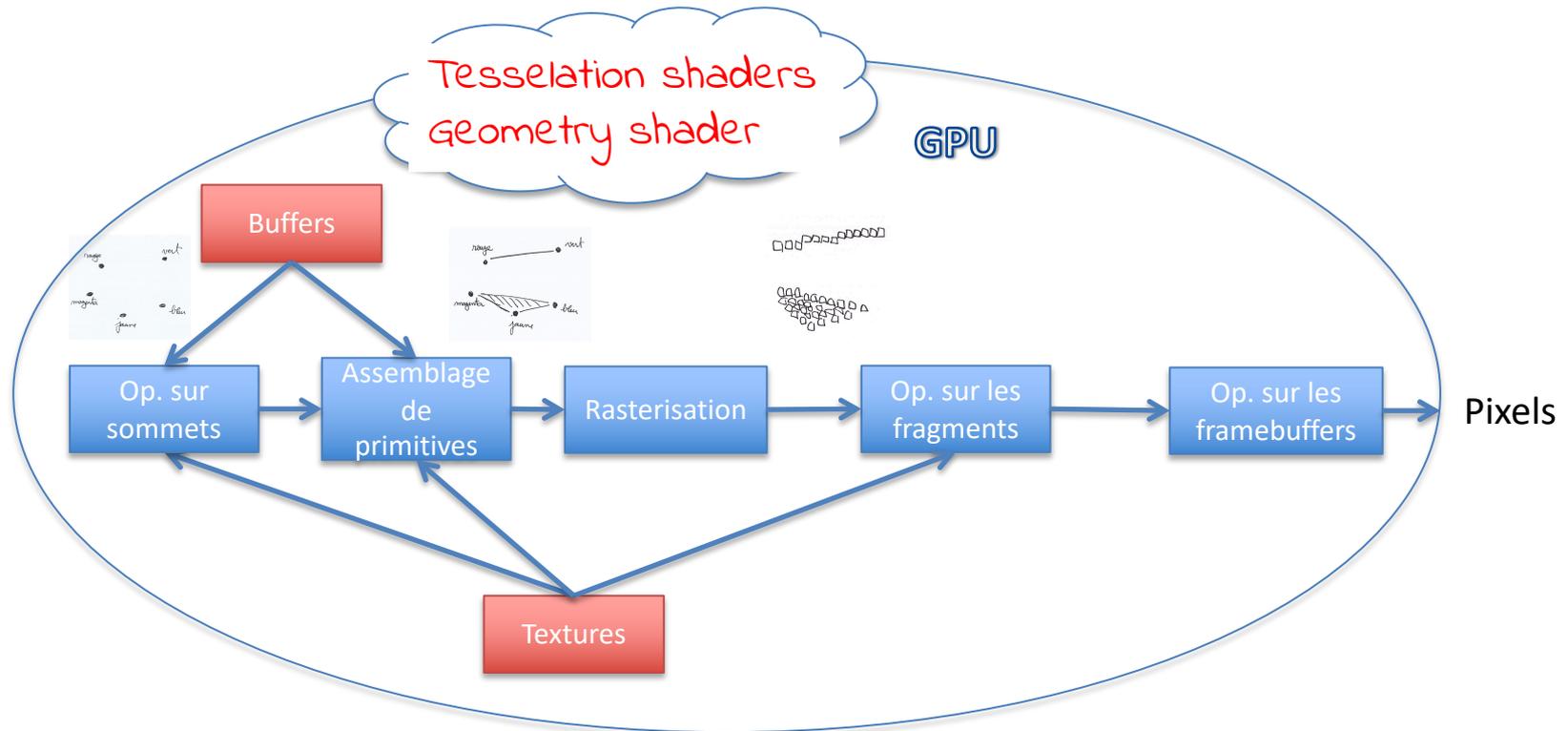


Texture

Formes

Pipeline OpenGL (moins) simplifié

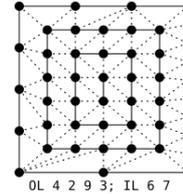
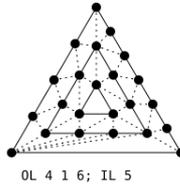
Du **SOMMET** au **PIXEL**



CPU: initialise les buffers, les textures et les calculs sur le GPU

Tessellation

- Notion de patch abstrait:



- (tessellation) Control shader
 - Contrôle la finesse de la tessellation sur les bords et à l'intérieur
- (tessellation) Evaluation shader
 - Calcule les sommets à partir des sommets du vertex shader et des paramètres des sommets abstraits
- Geometry shader
 - Modifie en dernier ressort les primitives transmises au Rasteriser