WebGL

S. Choplin

WebGL
Licence Sciences pour l'Ingenieur 1ère année
Université de Picardie J. Verne

Présentations

- Sébastien Choplin, (sebastien.choplin@u-picardie.fr)
  - Enseignant chercheur en informatique à l'UPJV.
  - Pas expert WebGL mais
    - enseignements en initiation à l'infographie
    - formation universitaire en mathématiques/informatique
    - bonnes connaissances des concepts du web
    - expérience en OpenGL
- Vous ?
  - math ?
  - HTML, JavaScript, ... ?
  - 3D ?
- WebGL démos
  http ://www.chromeexperiments.com/webgl/

Présentations

- Sébastien Choplin, (sebastien.choplin@u-picardie.fr)
  - Enseignant chercheur en informatique à l'UPJV.
  - Pas expert WebGL mais
    - enseignements en initiation à l'infographie
    - formation universitaire en mathématiques/informatique
    - bonnes connaissances des concepts du web
    - expérience en OpenGL
- Vous ?
  - math ?
  - HTML, JavaScript, ... ?
  - 3D ?
- WebGL démos
  http ://www.chromeexperiments.com/webgl/

Présentations

- Sébastien Choplin, (sebastien.choplin@u-picardie.fr)
  - Enseignant chercheur en informatique à l'UPJV.
  - Pas expert WebGL mais
    - enseignements en initiation à l'infographie
    - formation universitaire en mathématiques/informatique
    - bonnes connaissances des concepts du web
    - expérience en OpenGL
- Vous ?
  - math ?
  - HTML, JavaScript, ... ?
  - 3D ?
- WebGL démos
  http ://www.chromeexperiments.com/webgl/

Ce cours

- 4 jours de 2x3h $\Rightarrow$ 24h
  - un peu de manipulations de bases
  - un peu de présentation des concepts
  - un maximum de mise en pratique
- évaluation : un projet à rendre plus tard
  - en groupe
  - sujet en fonction de l'avancement
- support et exercices disponibles ici :
  http ://home.mis.u-picardie.fr/ choplin/enseignement/webgl/

Complexe mais pas inabordable si

- vous êtes un peu organisé
- vous ne perdez pas le fil
- ça vous amuse

Ce cours est une première pour moi :

- n'hésitez pas à inter-agir ! ! !

Ce cours

- 4 jours de 2x3h $\Rightarrow$ 24h
  - un peu de manipulations de bases
  - un peu de présentation des concepts
  - un maximum de mise en pratique
- évaluation : un projet à rendre plus tard
  - en groupe
  - sujet en fonction de l'avancement
- support et exercices disponibles ici : *http ://home.mis.u-picardie.fr/ choplin/enseignement/webgl/*

Complexe mais pas inabordable si

- vous êtes un peu organisé
- vous ne perdez pas le fil
- ça vous amuse

Ce cours est une première pour moi :

- n'hésitez pas à inter-agir ! ! !

Ce cours

- 4 jours de 2x3h $\Rightarrow$ 24h
  - un peu de manipulations de bases
  - un peu de présentation des concepts
  - un maximum de mise en pratique
- évaluation : un projet à rendre plus tard
  - en groupe
  - sujet en fonction de l'avancement
- support et exercices disponibles ici :
  http ://home.mis.u-picardie.fr/ choplin/enseignement/webgl/

Complexe mais pas inabordable si

- vous êtes un peu organisé
- vous ne perdez pas le fil
- ça vous amuse

Ce cours est une première pour moi :

- n'hésitez pas à inter-agir ! ! !

Ce cours

- 4 jours de 2x3h $\Rightarrow$ 24h
  - un peu de manipulations de bases
  - un peu de présentation des concepts
  - un maximum de mise en pratique
- évaluation : un projet à rendre plus tard
  - en groupe
  - sujet en fonction de l'avancement
- support et exercices disponibles ici :
  http ://home.mis.u-picardie.fr/ choplin/enseignement/webgl/

Complexe mais pas inabordable si

- vous êtes un peu organisé
- vous ne perdez pas le fil
- ça vous amuse

Ce cours est une première pour moi :

- n'hésitez pas à inter-agir ! ! !

Ce cours

- 4 jours de 2x3h $\Rightarrow$ 24h
  - un peu de manipulations de bases
  - un peu de présentation des concepts
  - un maximum de mise en pratique
- évaluation : un projet à rendre plus tard
  - en groupe
  - sujet en fonction de l'avancement
- support et exercices disponibles ici :
  http ://home.mis.u-picardie.fr/ choplin/enseignement/webgl/

Complexe mais pas inabordable si

- vous êtes un peu organisé
- vous ne perdez pas le fil
- ça vous amuse

Ce cours est une première pour moi :

- n'hésitez pas à inter-agir ! ! !

Références

- Vous même
  Vos cours de mathématiques de 3^{me}
- http ://www.khronos.org/webgl/
  http ://www.khronos.org/webgl/
- T. Parisi.
  WebGL : Up and Running.
  O'Reilly Media, 2012.
- Learning WebGL
  http ://learningwebgl.com

Lignes directrices

Introduction

Supposé connu :

- HTML / CSS / Javascript
- un peu de maths

Lignes directrices

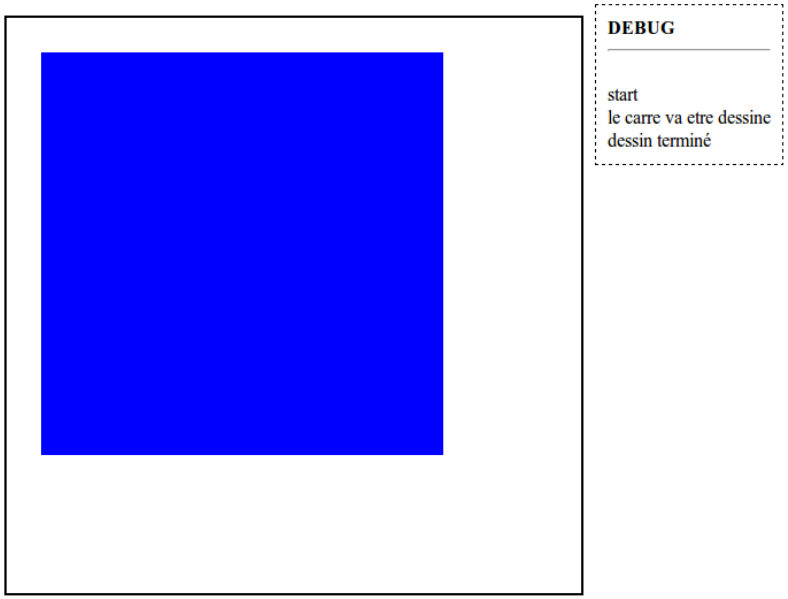
Canvas HTML 5

Depuis HTML5 : composant permettant de dessiner avec interactions Javascript.

```
<html>
<head></head>
<body>
<canvas id="moncanvas" width="500" height="500">
  Affichage d'un texte pour les navigateurs qui ne supportent
</canvas>
<script type="text/javascript">
  var exemple = document.getElementById('moncanvas');
  var ctx = exemple.getContext('2d');
  ctx.fillStyle = 'blue';
  ctx.fillRect(30, 30, 50, 50);
</script>

</body>
</html>
```

Exercice 1

Dessiner un cube dans un canvas HTML



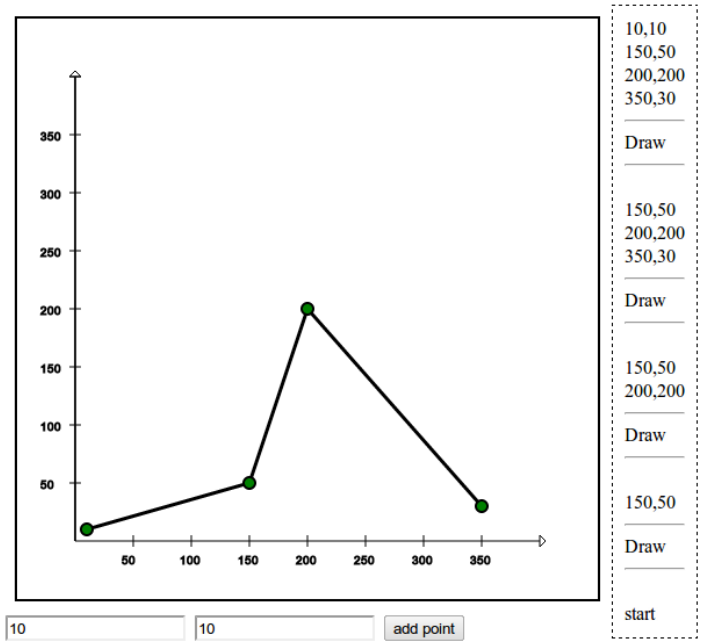
A utiliser

- js : `fillRect`, `fillStyle`, `getElementById`, ...
- CSS : `border`

- dessiner la bordure du contour du canvas
- ajouter un élément HTML et une fonction javascript pour afficher des informations de débuggage.

Exercice 2

Interaction Javascript, dessiner un graphique



A utiliser

- js : tableau, `beginPath`, `moveTo`, `lineTo`, `strokeText`, `arc`, `onSubmit`, ...
- HTML : `<form>`, `<inputtext>`

- dessiner un cercle sur chaque point
- dessiner un trait entre chaque point
- mettre 2 champs texte pour ajouter un point de manière dynamique

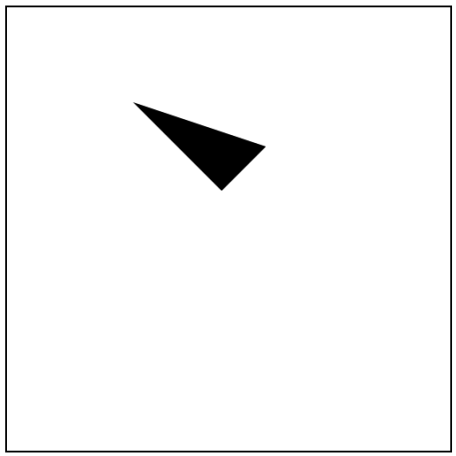
Exercice 3

Animation, faire bouger un triangle

- faire bouger un triangle suivant une direction donnée
- modifier la direction pour 'faire rebondir' le triangle quand il touche un bord

A utiliser

- js : `clearRect`,




```
window.requestAnimFrame =
 (function(callback) {
 window.setTimeout(callback
     ,1000/60);
});
function animate() {
  //... modification et dessin ...
 requestAnimFrame(animate);
}
```

Lignes directrices

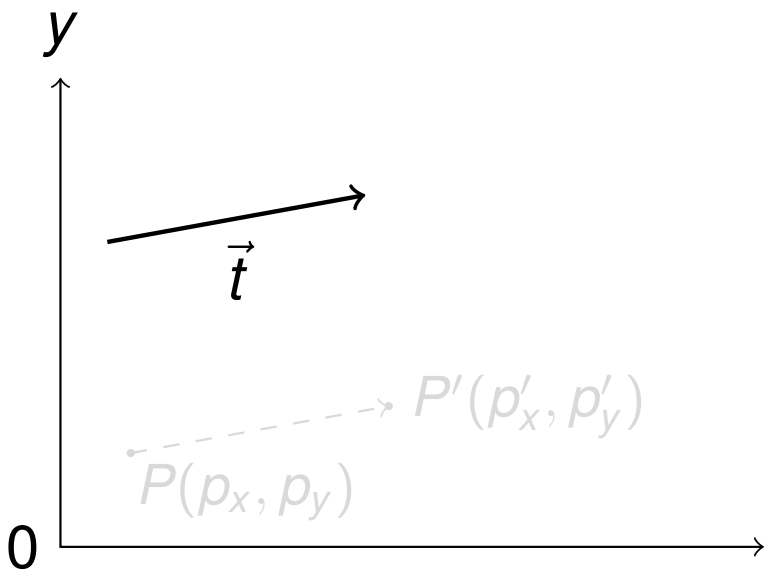
Représentation des objets

- points par points
- par des polygones (triangles, ...)
- par des courbes paramétriques,
- ...

Il faut aussi gérer l'empilement des objets.

Transformations : translation

translation de vecteur $\vec{t} = (t_x, t_y)$

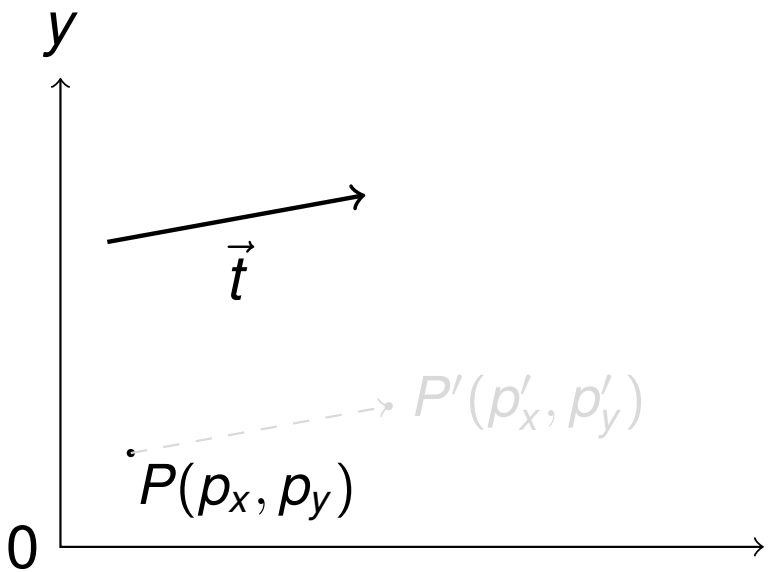


fonction de P : Coordonnées de P' en

$$\begin{cases} p'_x = p_x + t_x \\ p'_y = p_y + t_y \end{cases}$$

Transformations : translation

translation de vecteur $\vec{t} = (t_x, t_y)$

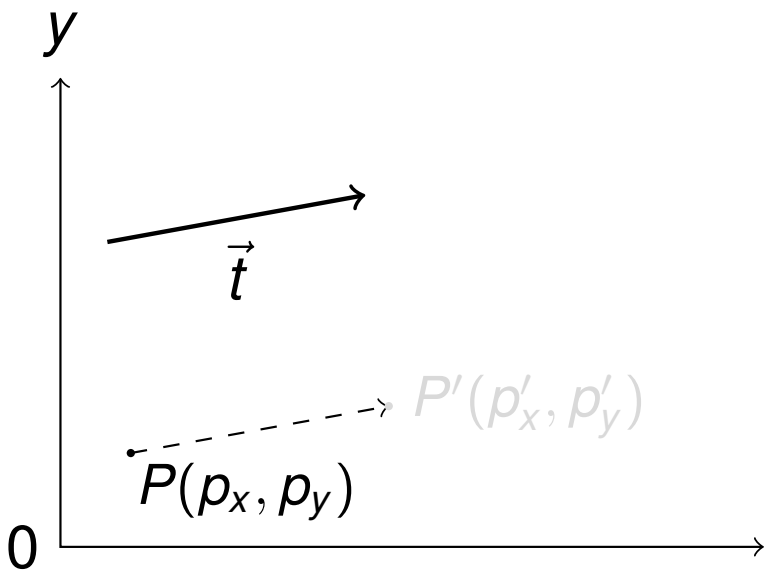


fonction de P : Coordonnées de P' en

$$\begin{cases} p'_x = p_x + t_x \\ p'_y = p_y + t_y \end{cases}$$

Transformations : translation

translation de vecteur $\vec{t} = (t_x, t_y)$

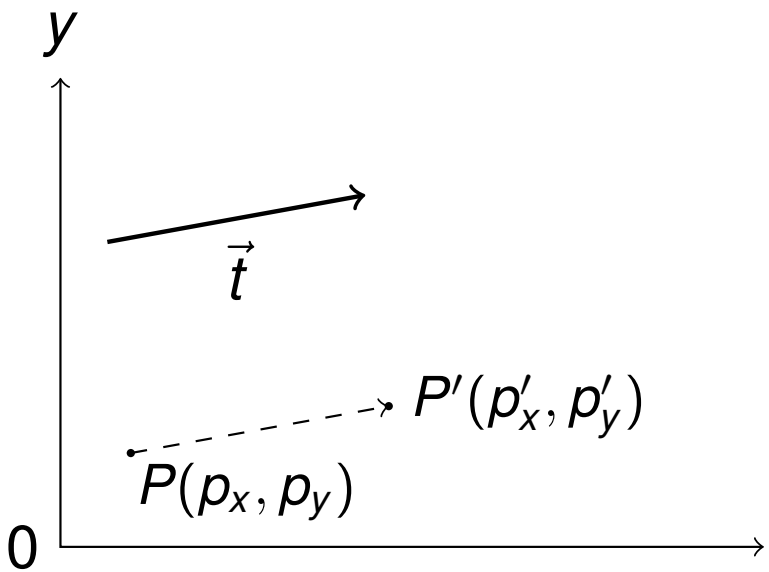


Coordonnées de P' en fonction de P :

$$\begin{cases} p'_x = p_x + t_x \\ p'_y = p_y + t_y \end{cases}$$

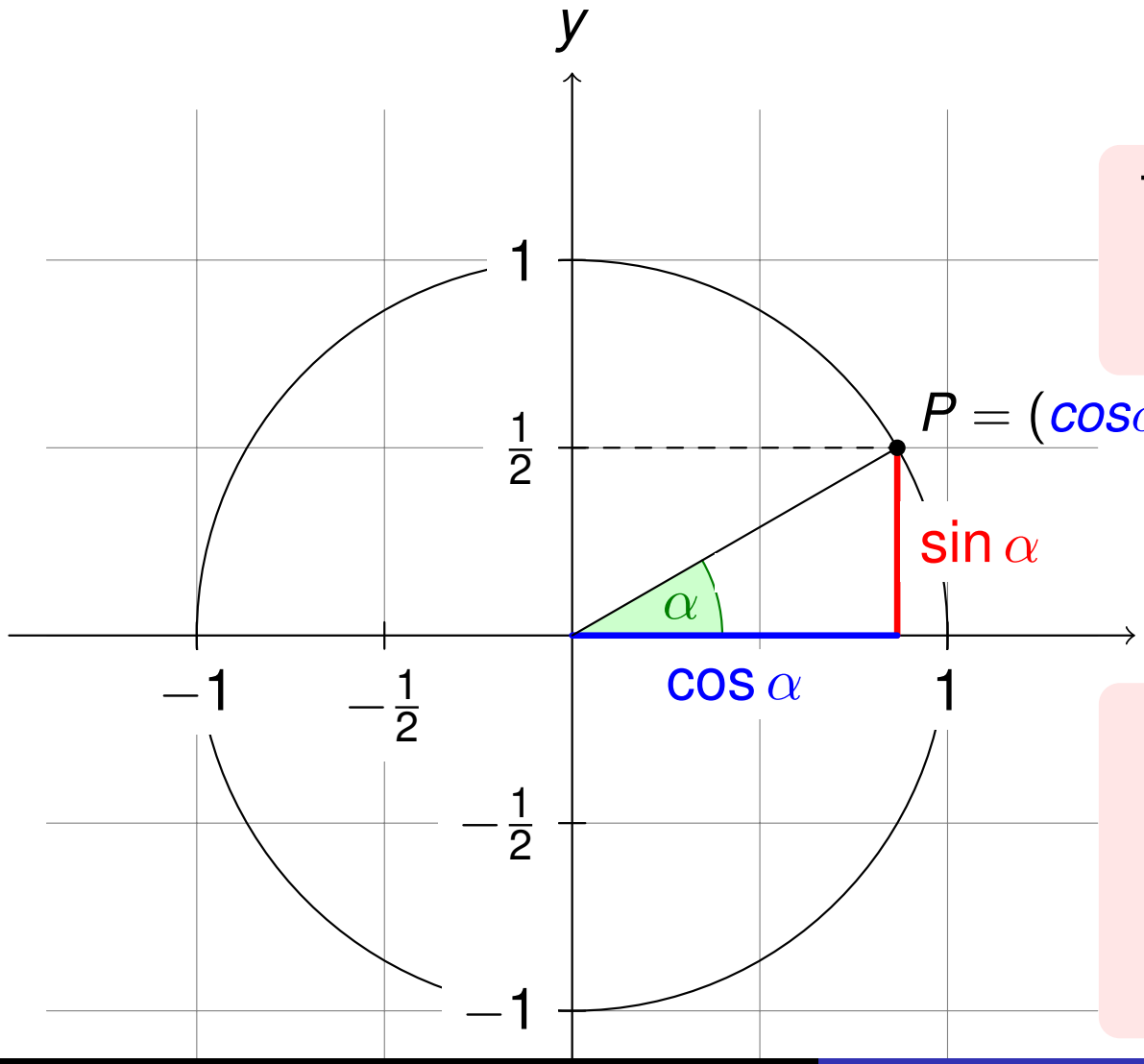
Transformations : translation

translation de vecteur $\vec{t} = (t_x, t_y)$



fonction de P : $\begin{cases} p'_x = p_x + t_x \\ p'_y = p_y + t_y \end{cases}$ Coordonnées de P' en

Cercle trigonométrique



Cercle de rayon 1

Théorème de Pythagore

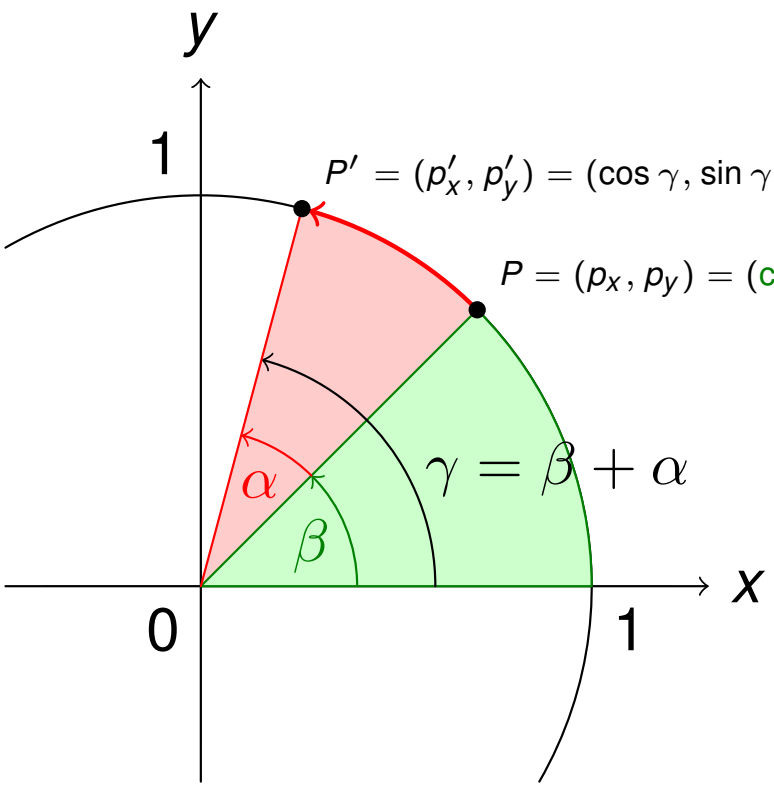
$$\cos^2\alpha + \sin^2\alpha = 1$$

Conversion deg / rad

degré	radian
α	$\alpha\pi/180$
$180\alpha/\pi$	α

Transformations : rotation

Rotation d'angle α de centre $(0, 0)$.



Formules Trigo. :

$$\begin{aligned} \cos(a+b) &= \cos(a)\cos(b) - \sin(a)\sin(b) \\ \sin(a+b) &= \cos(a)\sin(b) + \sin(a)\cos(b) \end{aligned}$$

$$\begin{cases} p'_x &= p_x\cos(\alpha) - p_y\sin(\alpha) \\ p'_y &= p_x\sin(\alpha) + p_y\cos(\alpha) \end{cases}$$

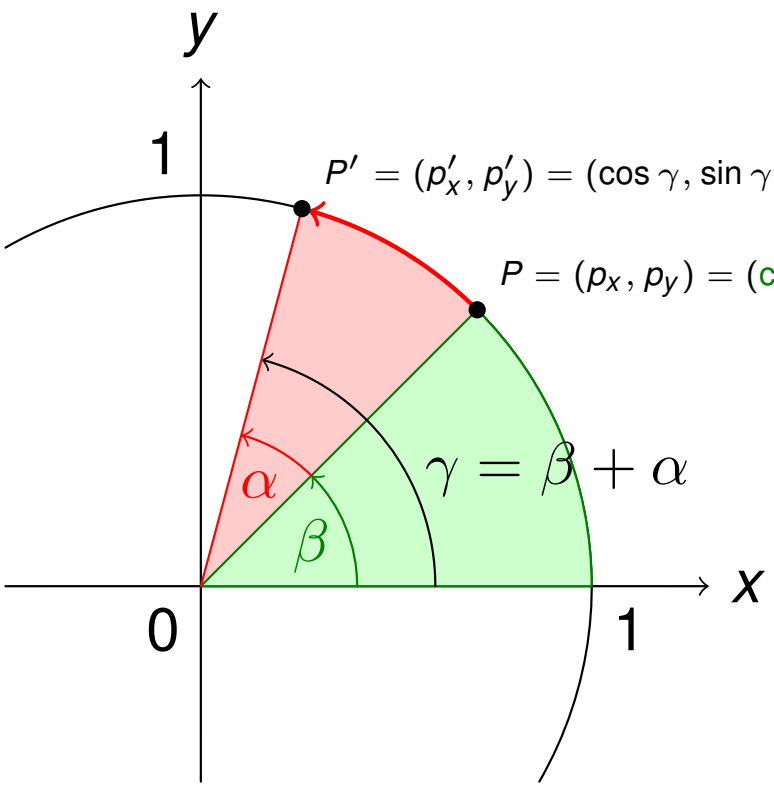
Ecriture matricielle :

$$\begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

$$\begin{bmatrix} x & y \end{bmatrix} \quad \begin{bmatrix} x\cos(\alpha) - y\sin(\alpha) & x\sin(\alpha) + y\cos(\alpha) \end{bmatrix}$$

Transformations : rotation

Rotation d'angle α de centre $(0, 0)$.



Formules Trigo. :

$$\begin{array}{rcl} \cos(a+b) & = & \cos(a)\cos(b) - \sin(a)\sin(b) \\ \sin(a+b) & = & \cos(a)\sin(b) + \sin(a)\cos(b) \end{array}$$

$$\begin{cases} p'_x & = & p_x\cos(\alpha) - p_y\sin(\alpha) \\ p'_y & = & p_x\sin(\alpha) + p_y\cos(\alpha) \end{cases}$$

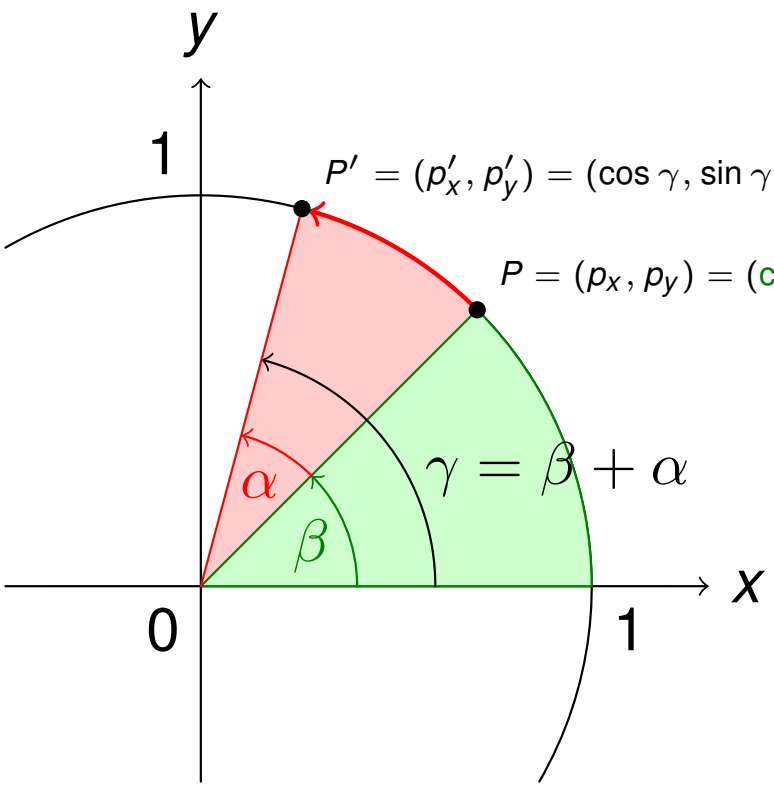
Ecriture matricielle :

$$\begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

$$\begin{bmatrix} x & y \end{bmatrix} \quad \begin{bmatrix} x\cos(\alpha) - y\sin(\alpha) & x\sin(\alpha) + y\cos(\alpha) \end{bmatrix}$$

Transformations : rotation

Rotation d'angle α de centre $(0, 0)$.



Formules Trigo. :

$$\begin{array}{rcl} \cos(a+b) & = & \cos(a)\cos(b) - \sin(a)\sin(b) \\ \sin(a+b) & = & \cos(a)\sin(b) + \sin(a)\cos(b) \end{array}$$

$$\begin{cases} p'_x & = & p_x\cos(\alpha) - p_y\sin(\alpha) \\ p'_y & = & p_x\sin(\alpha) + p_y\cos(\alpha) \end{cases}$$

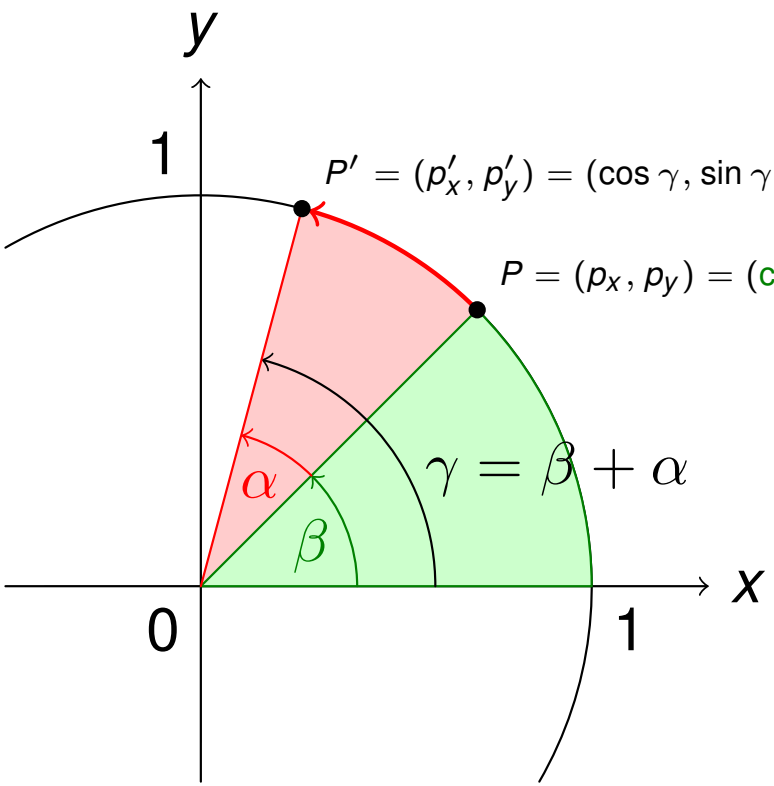
Ecriture matricielle :

$$\begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

$$\begin{bmatrix} x & y \end{bmatrix} \quad \begin{bmatrix} x\cos(\alpha) - y\sin(\alpha) & x\sin(\alpha) + y\cos(\alpha) \end{bmatrix}$$

Transformations : rotation

Rotation d'angle α de centre $(0, 0)$.



Formules Trigo. :

$$\begin{aligned} \cos(a+b) &= \cos(a)\cos(b) - \sin(a)\sin(b) \\ \sin(a+b) &= \cos(a)\sin(b) + \sin(a)\cos(b) \end{aligned}$$

$$\begin{cases} p'_x &= p_x\cos(\alpha) - p_y\sin(\alpha) \\ p'_y &= p_x\sin(\alpha) + p_y\cos(\alpha) \end{cases}$$

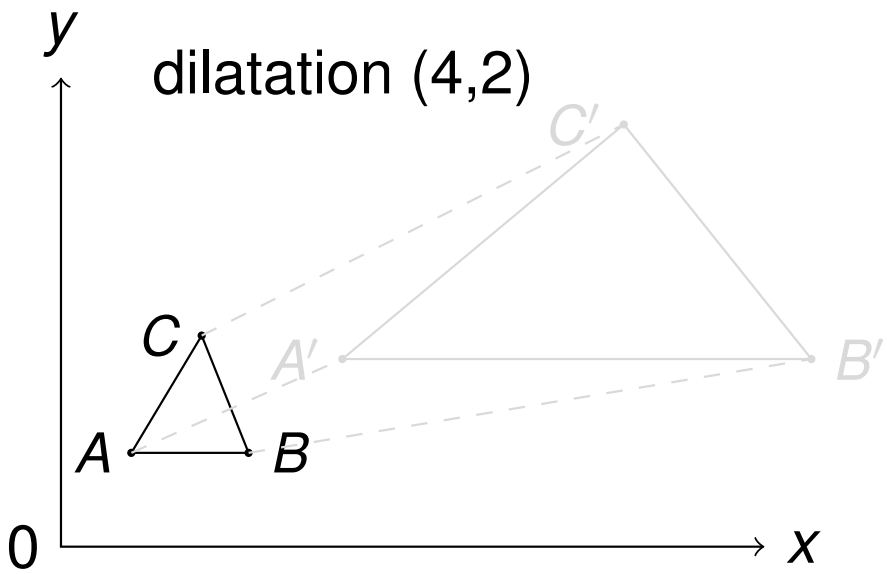
Ecriture matricielle :

$$\begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

$$\begin{bmatrix} x & y \end{bmatrix} \quad \begin{bmatrix} x\cos(\alpha) - y\sin(\alpha) & x\sin(\alpha) + y\cos(\alpha) \end{bmatrix}$$

Transformations : dilatation

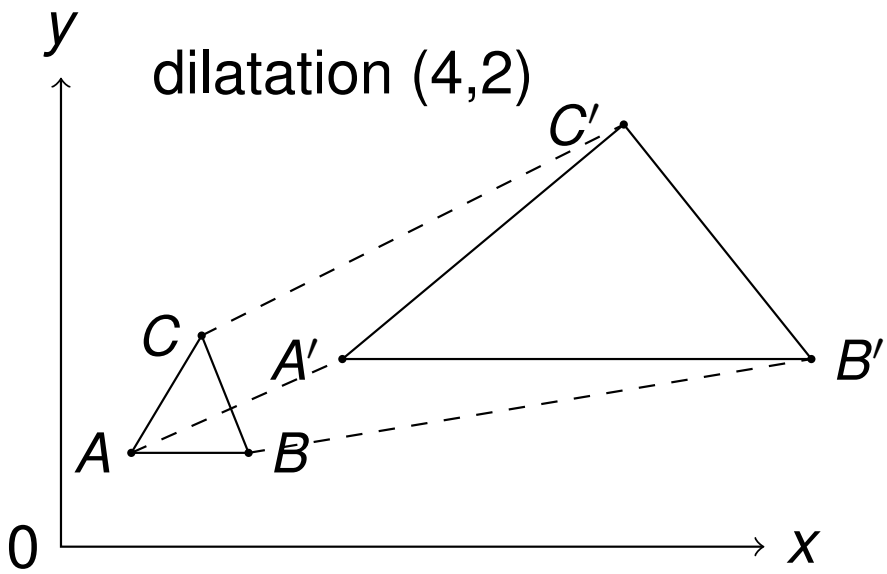
Dilatation ou Etirement (d_x, d_y)



$$\begin{cases} x' & = & d_x \times x \\ y' & = & d_y \times y \end{cases}$$

Transformations : dilatation

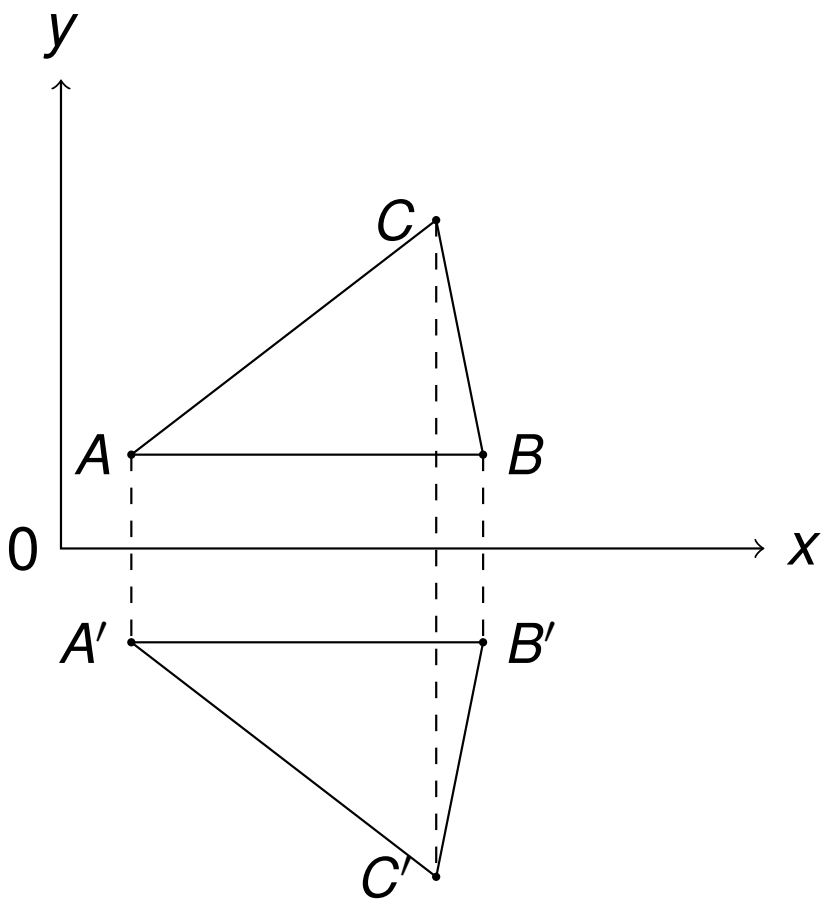
Dilatation ou Etirement (d_x, d_y)



$$\begin{cases} x' & = & d_x \times x \\ y' & = & d_y \times y \end{cases}$$

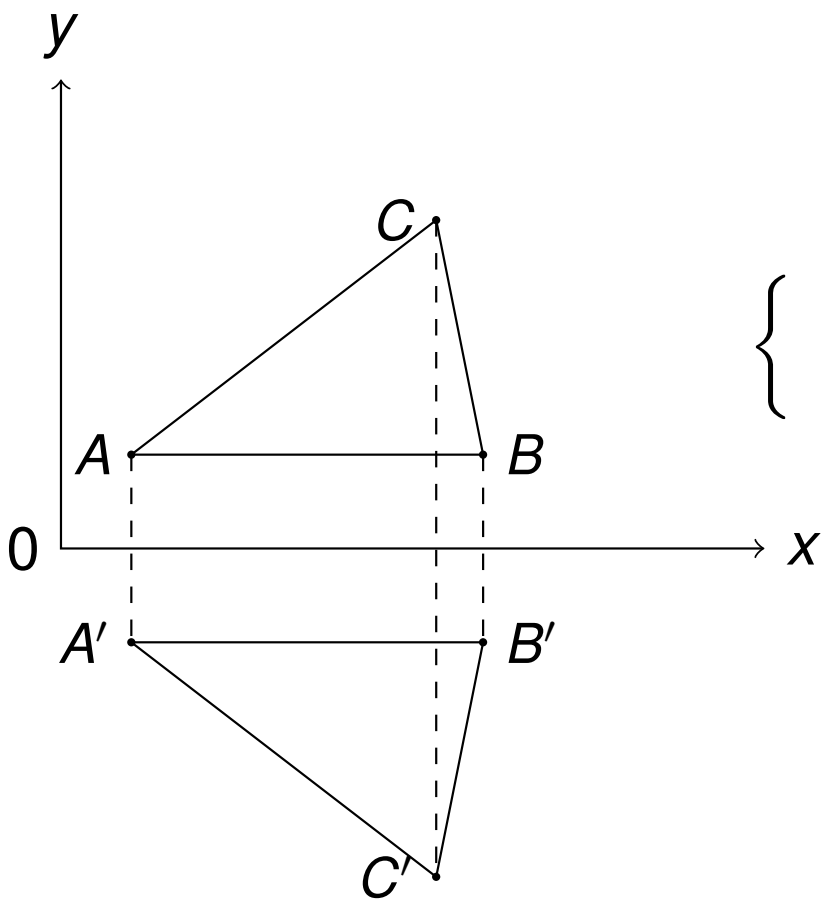
Transformations : symétrie

Symétrie par rapport à l'axe des abscisses



Transformations : symétrie

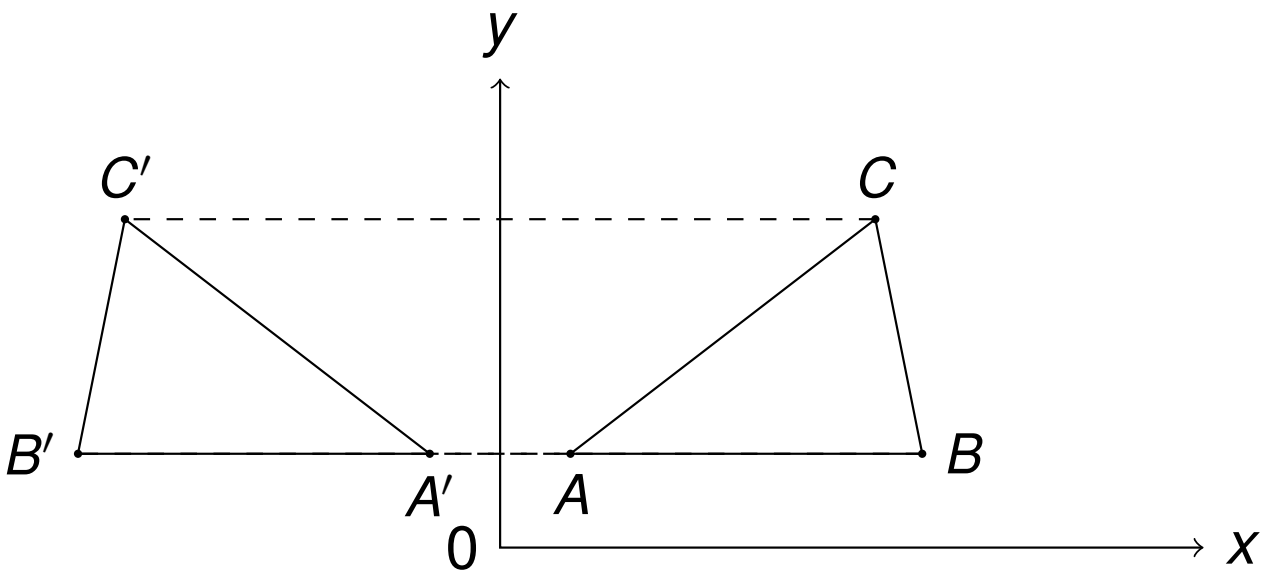
Symétrie par rapport à l'axe des abscisses



$$\begin{cases} x' = x \\ y' = -y \end{cases}$$

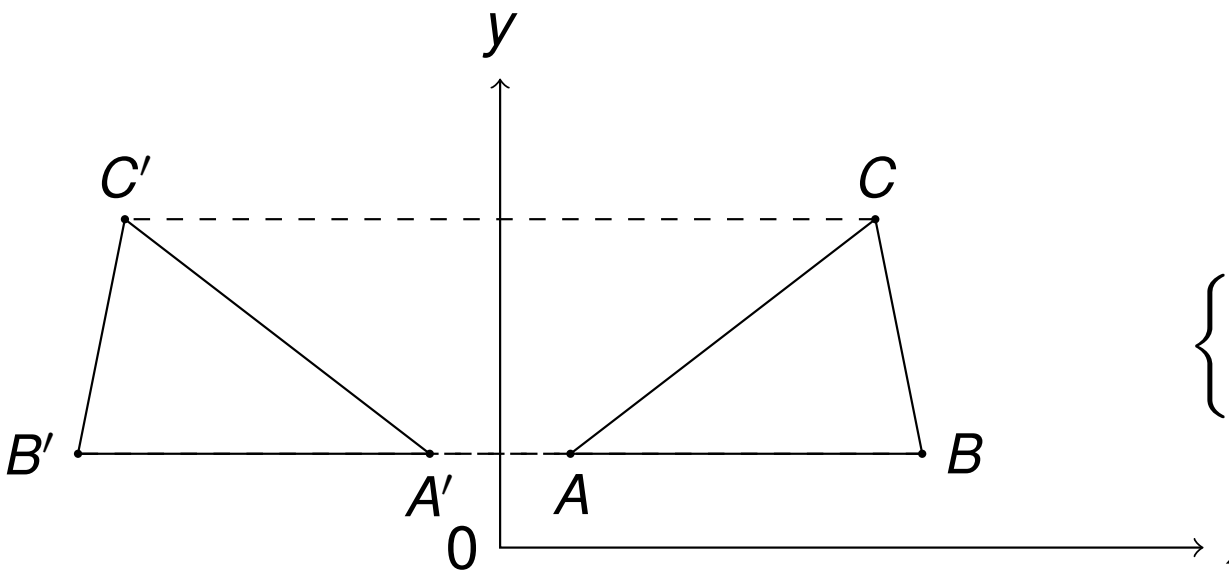
Transformations : symétrie

Symétrie par rapport à l'axe des ordonnées



Transformations : symétrie

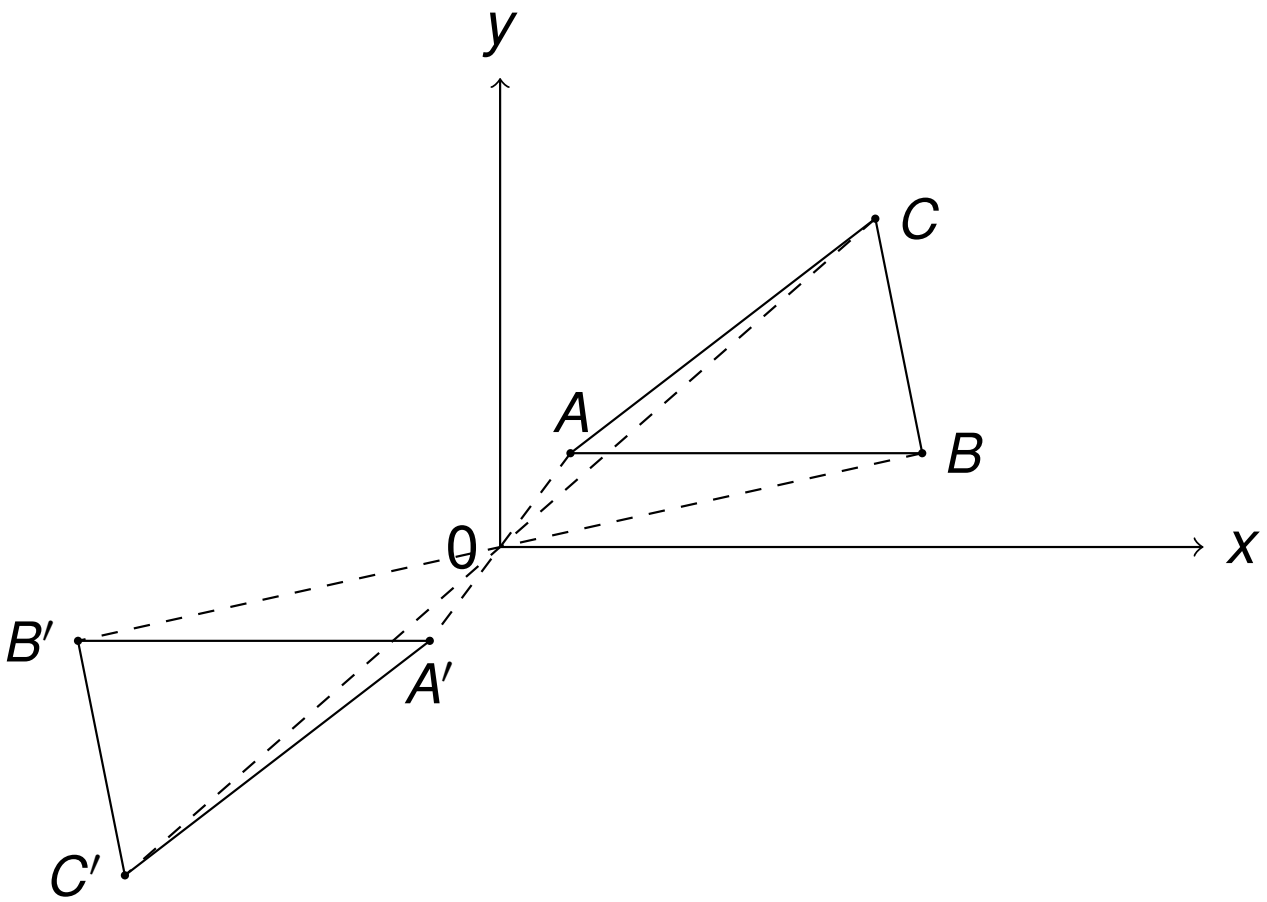
Symétrie par rapport à l'axe des ordonnées



$$\begin{cases} x' = -x \\ y' = y \end{cases}$$

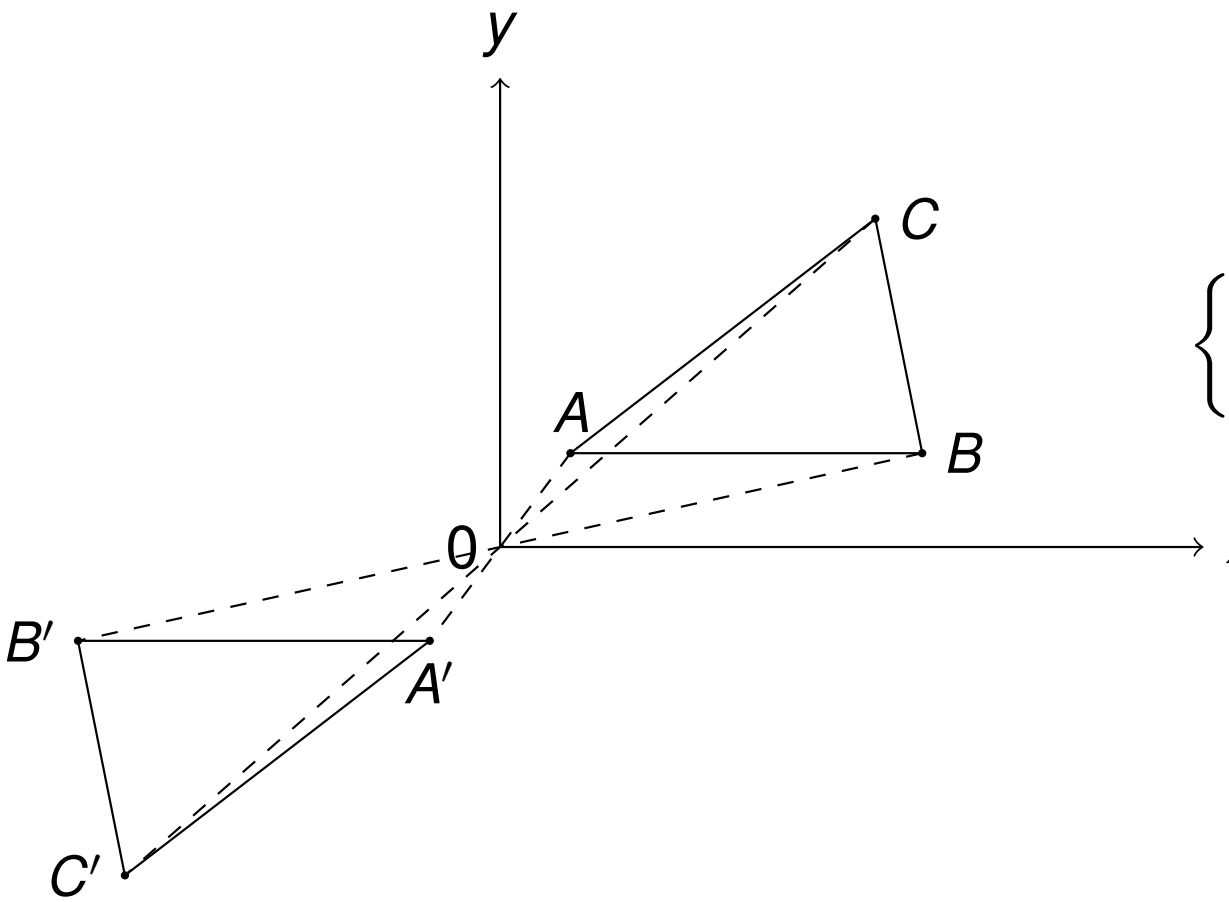
Transformations : symétrie

Symétrie par rapport à l'origine



Transformations : symétrie

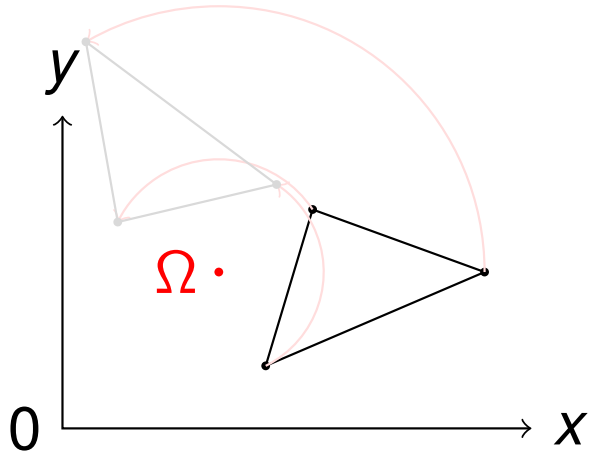
Symétrie par rapport à l'origine



$$\begin{cases} x' = -x \\ y' = -y \end{cases}$$

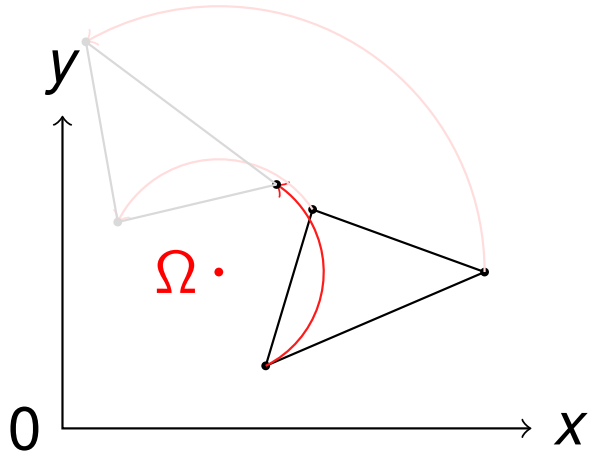
Compositions de transformations

Comment faire une rotation de centre $\Omega(1,1)$, d'angle 120° ?



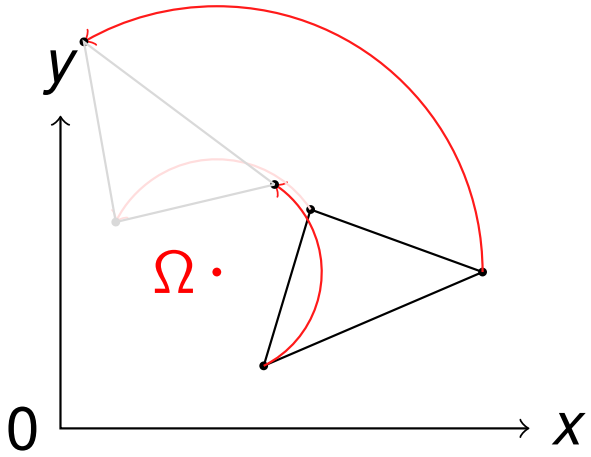
Compositions de transformations

Comment faire une rotation de centre $\Omega(1,1)$, d'angle 120° ?



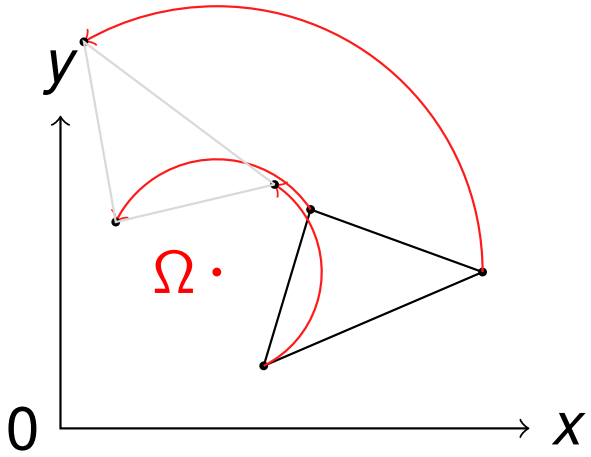
Compositions de transformations

Comment faire une rotation de centre $\Omega(1,1)$, d'angle 120° ?



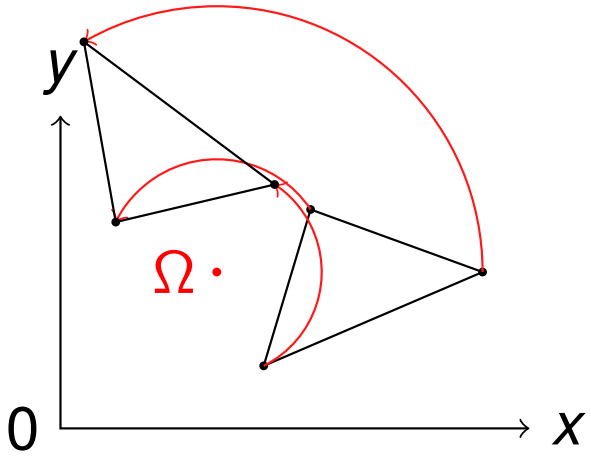
Compositions de transformations

Comment faire une rotation de centre $\Omega(1,1)$, d'angle 120° ?



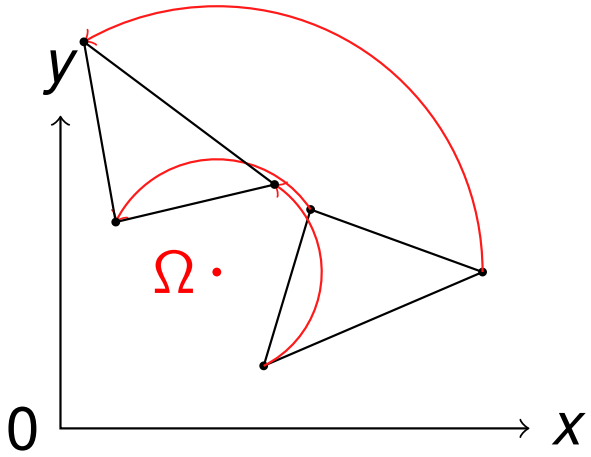
Compositions de transformations

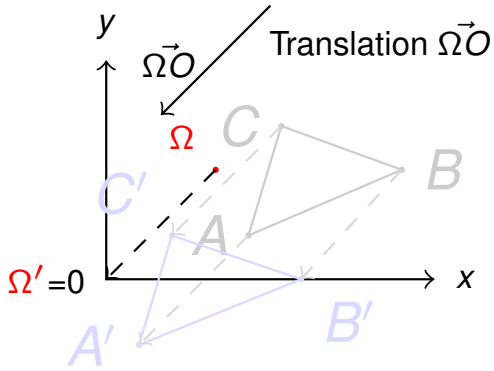
Comment faire une rotation de centre $\Omega(1,1)$, d'angle 120° ?



Compositions de transformations

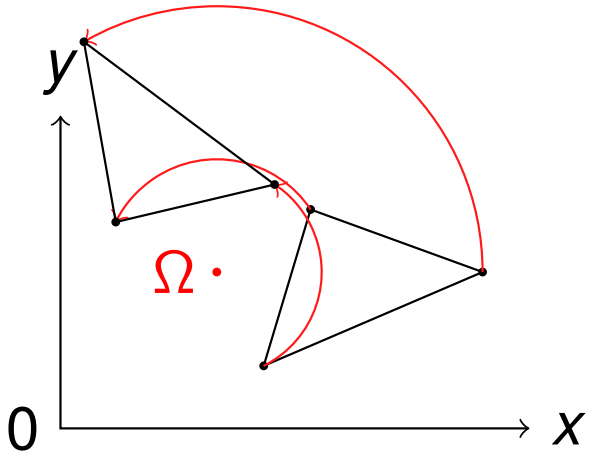
Comment faire une rotation de centre $\Omega(1,1)$, d'angle 120° ?

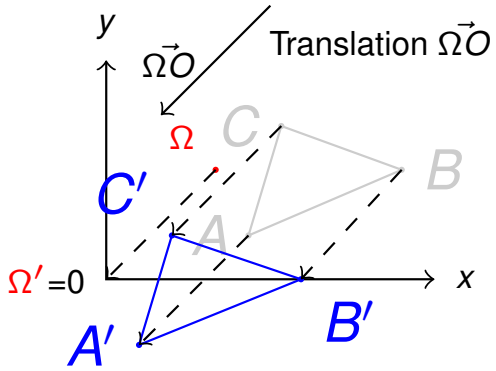




Compositions de transformations

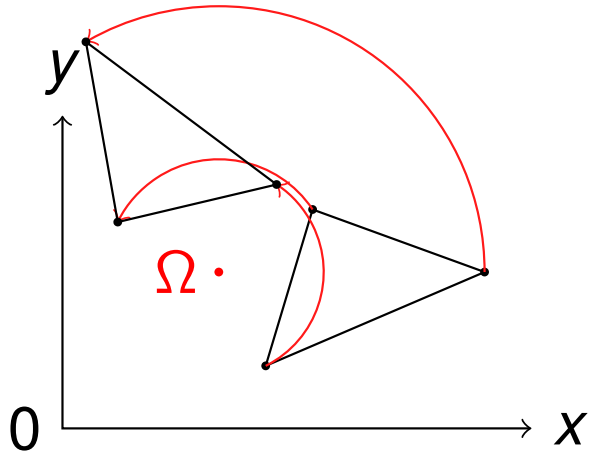
Comment faire une rotation de centre $\Omega(1,1)$, d'angle 120° ?

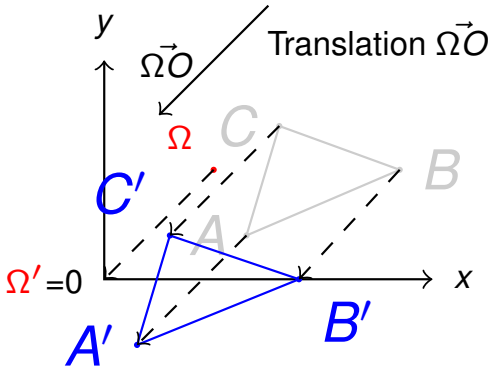


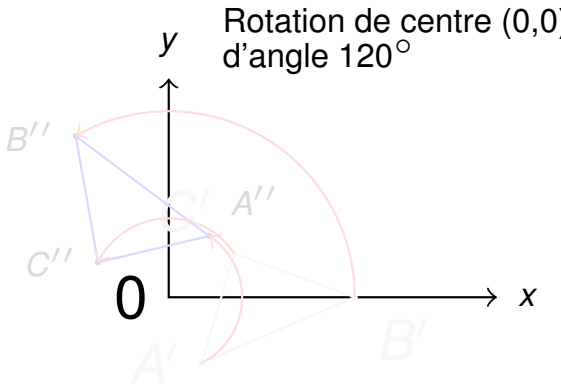


Compositions de transformations

Comment faire une rotation de centre $\Omega(1,1)$, d'angle 120° ?

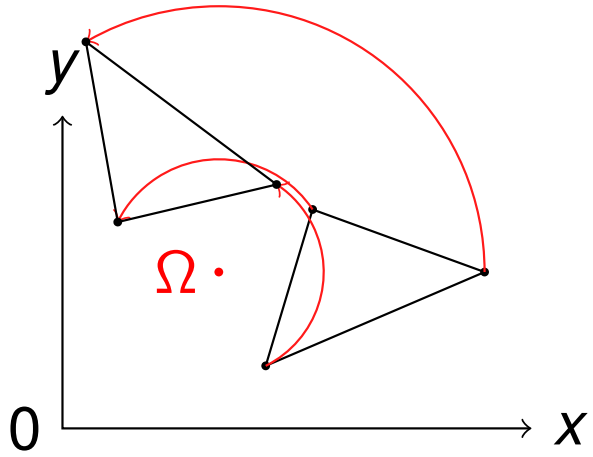


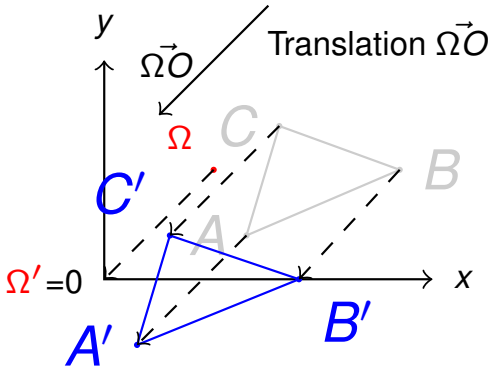


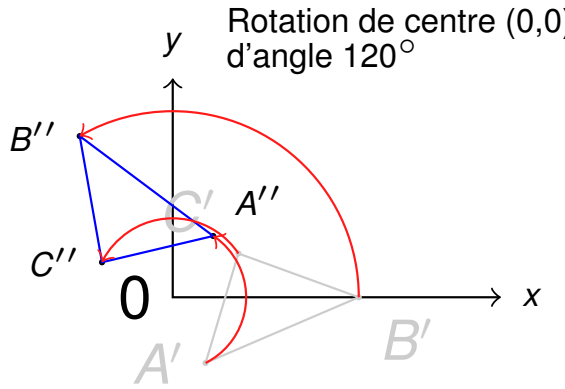


Compositions de transformations

Comment faire une rotation de centre $\Omega(1,1)$, d'angle 120° ?

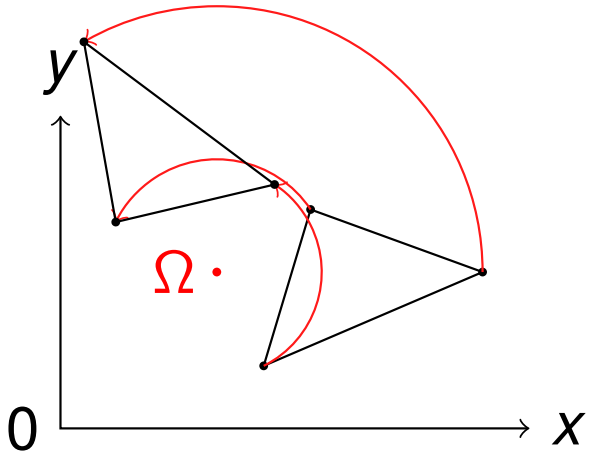


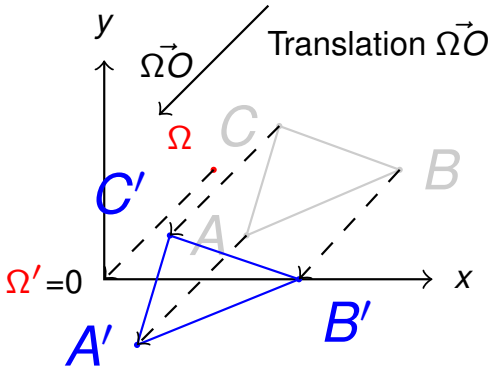


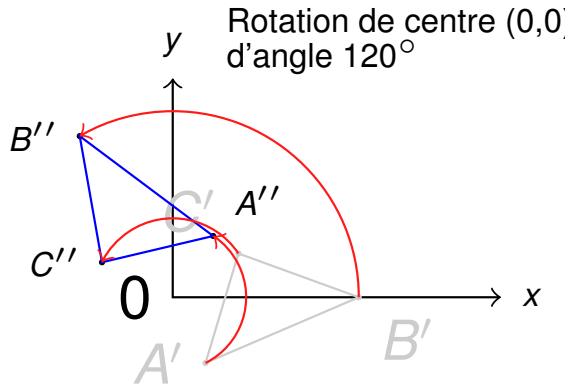


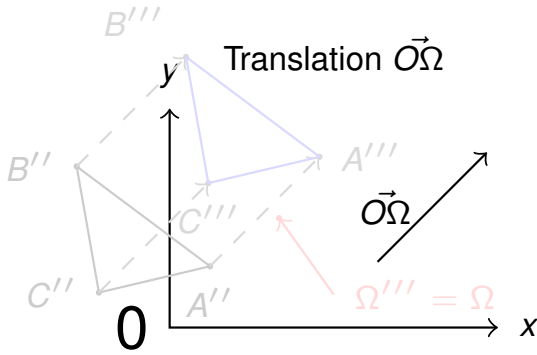
Compositions de transformations

Comment faire une rotation de centre $\Omega(1,1)$, d'angle 120° ?



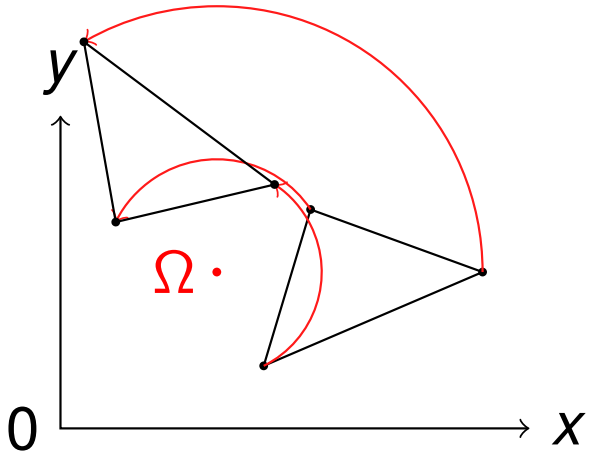


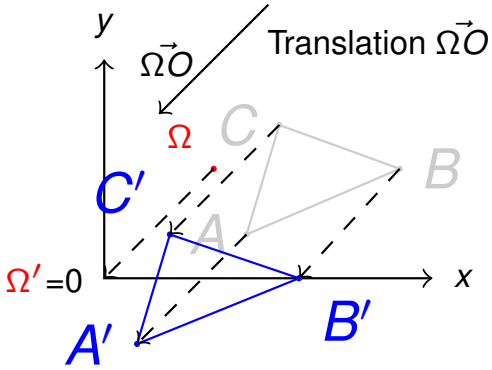


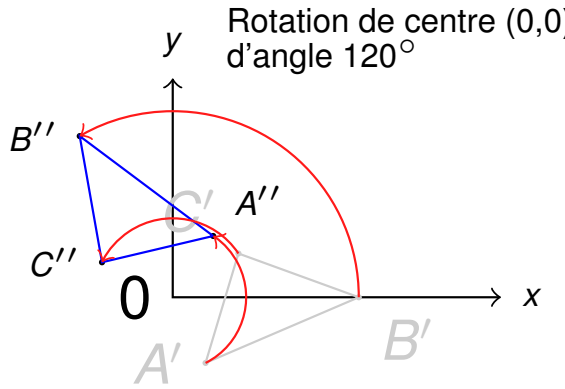


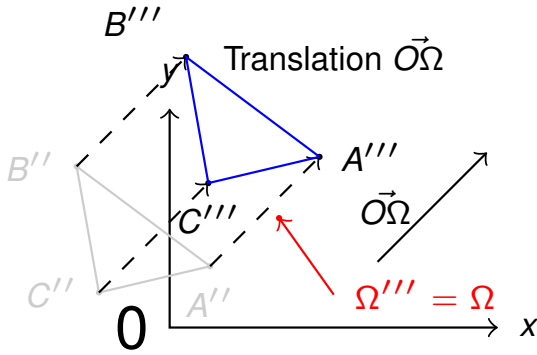
Compositions de transformations

Comment faire une rotation de centre $\Omega(1,1)$, d'angle 120° ?









Compositions de transformations

$x''' = ?, y''' = ?$

Translation de vecteur $(-1, -1)$

$$\begin{cases} x' &= x - 1 \\ y' &= y - 1 \end{cases}$$

Rotation de centre $(0, 0)$, d'angle $120°$

$$\begin{cases} x'' &= (x-1)\cos(120) - (y-1)\sin(120) &= -\frac{x-1}{2} - \frac{\sqrt{3}(y-1)}{2} \\ y'' &= (x-1)\sin(120) + (y-1)\cos(120) &= \frac{\sqrt{3}(x-1)}{2} - \frac{y-1}{2} \end{cases}$$

Translation de vecteur $(1, 1)$

$$\begin{cases} x''' &= -\frac{x-1}{2} - \frac{\sqrt{3}(y-1)}{2} + 1 &= -\frac{1}{2}x - \frac{\sqrt{3}}{2}y + \frac{3+\sqrt{3}}{2} \\ y''' &= \frac{\sqrt{3}(x-1)}{2} - \frac{y-1}{2} + 1 &= \frac{\sqrt{3}}{2}x - \frac{1}{2}y + \frac{3-\sqrt{3}}{2} \end{cases}$$

Compositions de transformations

$x''' = ?, y''' = ?$

Translation de vecteur $(-1, -1)$

$$\begin{cases} x' &= x - 1 \\ y' &= y - 1 \end{cases}$$

Rotation de centre $(0, 0)$, d'angle $120°$

$$\begin{cases} x'' &= (x-1)\cos(120) - (y-1)\sin(120) &= -\frac{x-1}{2} - \frac{\sqrt{3}(y-1)}{2} \\ y'' &= (x-1)\sin(120) + (y-1)\cos(120) &= \frac{\sqrt{3}(x-1)}{2} - \frac{y-1}{2} \end{cases}$$

Translation de vecteur $(1, 1)$

$$\begin{cases} x''' &= -\frac{x-1}{2} - \frac{\sqrt{3}(y-1)}{2} + 1 &= -\frac{1}{2}x - \frac{\sqrt{3}}{2}y + \frac{3+\sqrt{3}}{2} \\ y''' &= \frac{\sqrt{3}(x-1)}{2} - \frac{y-1}{2} + 1 &= \frac{\sqrt{3}}{2}x - \frac{1}{2}y + \frac{3-\sqrt{3}}{2} \end{cases}$$

Compositions de transformations

$x''' = ?, y''' = ?$

Translation de vecteur $(-1, -1)$

$$\begin{cases} x' &= x - 1 \\ y' &= y - 1 \end{cases}$$

Rotation de centre $(0, 0)$, d'angle $120°$

$$\begin{cases} x'' &= (x-1)\cos(120) - (y-1)\sin(120) &= -\frac{x-1}{2} - \frac{\sqrt{3}(y-1)}{2} \\ y'' &= (x-1)\sin(120) + (y-1)\cos(120) &= \frac{\sqrt{3}(x-1)}{2} - \frac{y-1}{2} \end{cases}$$

Translation de vecteur $(1, 1)$

$$\begin{cases} x''' &= -\frac{x-1}{2} - \frac{\sqrt{3}(y-1)}{2} + 1 &= -\frac{1}{2}x - \frac{\sqrt{3}}{2}y + \frac{3+\sqrt{3}}{2} \\ y''' &= \frac{\sqrt{3}(x-1)}{2} - \frac{y-1}{2} + 1 &= \frac{\sqrt{3}}{2}x - \frac{1}{2}y + \frac{3-\sqrt{3}}{2} \end{cases}$$

Compositions de transformations

$x''' = ?, y''' = ?$

Translation de vecteur $(-1, -1)$

$$\begin{cases} x' &= x - 1 \\ y' &= y - 1 \end{cases}$$

Rotation de centre $(0, 0)$, d'angle $120°$

$$\begin{cases} x'' &= (x-1)\cos(120) - (y-1)\sin(120) &= -\frac{x-1}{2} - \frac{\sqrt{3}(y-1)}{2} \\ y'' &= (x-1)\sin(120) + (y-1)\cos(120) &= \frac{\sqrt{3}(x-1)}{2} - \frac{y-1}{2} \end{cases}$$

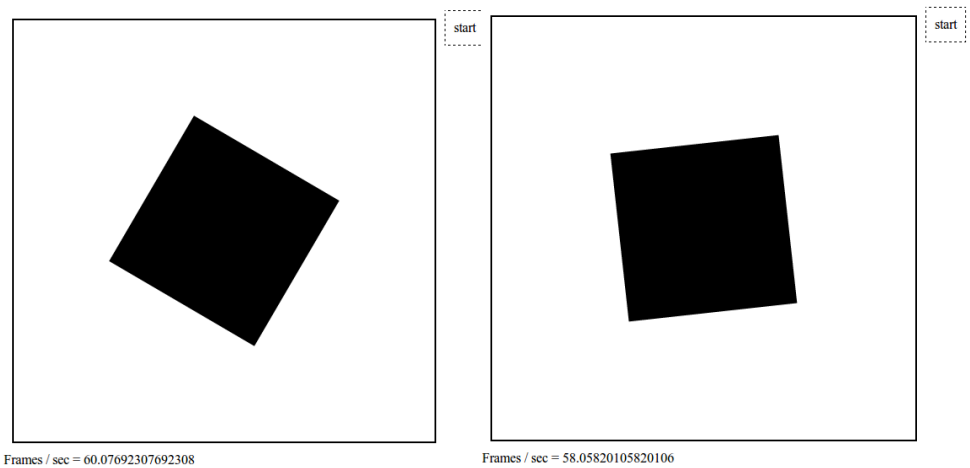
Translation de vecteur $(1, 1)$

$$\begin{cases} x''' &= -\frac{x-1}{2} - \frac{\sqrt{3}(y-1)}{2} + 1 &= -\frac{1}{2}x - \frac{\sqrt{3}}{2}y + \frac{3+\sqrt{3}}{2} \\ y''' &= \frac{\sqrt{3}(x-1)}{2} - \frac{y-1}{2} + 1 &= \frac{\sqrt{3}}{2}x - \frac{1}{2}y + \frac{3-\sqrt{3}}{2} \end{cases}$$

Exercice 4

Faire une animation qui fait tourner un carré autour de son centre

1. sans utiliser les méthodes JS `rotate`,`translate` ...
2. avec les méthodes JS `rotate`,`translate` ...



- Ajouter un compteur du nombre d'image par seconde.

Les Matrices

Au lieu d'écrire :

$$\begin{cases} x' &= a \times x + b \times y \\ y' &= c \times x + d \times y \end{cases}$$

On écrit :

$$\begin{bmatrix} a & c \\ b & d \end{bmatrix}$$

Produit :

$$\begin{bmatrix} x & y \end{bmatrix} \times \begin{bmatrix} a & c \\ b & d \end{bmatrix} = \begin{bmatrix} ax + by & cx + dy \end{bmatrix}$$

Attention au sens du produit Si on écrit le vecteur (x, y) en colonne, le produit est :

$$\begin{bmatrix} a & b \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax + by \end{bmatrix}$$

Les Matrices

Au lieu d'écrire :

$$\begin{cases} x' &= a \times x + b \times y \\ y' &= c \times x + d \times y \end{cases}$$

On écrit :

$$\begin{bmatrix} a & c \\ b & d \end{bmatrix}$$

Produit :

$$\begin{bmatrix} x & y \end{bmatrix} \times \begin{bmatrix} a & c \\ b & d \end{bmatrix} = \begin{bmatrix} ax + by & cx + dy \end{bmatrix}$$

Attention au sens du produit Si on écrit le vecteur (x, y) en colonne, le produit est :

$$\begin{bmatrix} a & b \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax + by \end{bmatrix}$$

Les Matrices

Au lieu d'écrire :

$$\begin{cases} x' &= a \times x + b \times y \\ y' &= c \times x + d \times y \end{cases}$$

On écrit :

$$\begin{bmatrix} a & c \\ b & d \end{bmatrix}$$

Produit :

$$\begin{bmatrix} x & y \end{bmatrix} \times \begin{bmatrix} a & c \\ b & d \end{bmatrix} = \begin{bmatrix} ax + by & cx + dy \end{bmatrix}$$

Attention au sens du produit Si on écrit le vecteur (x, y) en colonne, le produit est :

$$\begin{bmatrix} a & b \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax + by \end{bmatrix}$$

Ecriture matricielle des transformations

Rotation d'angle α, de centre $(0, 0)$:

$$\begin{cases} x' &= x\cos(\alpha) - y\sin(\alpha) \\ y' &= x\sin(\alpha) + y\cos(\alpha) \end{cases}$$

$$\begin{bmatrix} x & y \end{bmatrix} \times \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

$$= \begin{bmatrix} x\cos(\alpha) - y\sin(\alpha) & x\sin(\alpha) + y\cos(\alpha) \end{bmatrix}$$

Ecriture matricielle des transformations

Translation de vecteur (t_x, t_y) :

$$\begin{cases} x' = x + t_x \\ y' = y + t_y \end{cases}$$

Ecriture matricielle des transformations

Translation de vecteur (t_x, t_y) :

$$\begin{cases} x' = x + t_x \\ y' = y + t_y \end{cases}$$

Problème : impossible d'écrire sous forme matricielle

Ecriture matricielle des transformations

Translation de vecteur (t_x, t_y) :

$$\begin{cases} x' = x + t_x \\ y' = y + t_y \end{cases}$$

Problème : impossible d'écrire sous forme matricielle

$\Rightarrow$**utilisation des coordonnées homogènes.**

Coordonnées homogènes

$$P(x,y) : [x \quad y \quad 1] = [x/w \quad y/w \quad w]$$

permet d'exprimer les transformations sous forme matricielle.

Exemple : **translation** de vecteur $\vec{t} = (t_x, t_y)$: $\begin{cases} x' = x + t_x \\ y' = y + t_y \end{cases}$

Matrice de la transformation :

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \times ? = \begin{bmatrix} x + t_x & y + t_y & 1 \end{bmatrix}$$

Coordonnées homogènes

$$P(x,y) : [x \quad y \quad 1] = [x/w \quad y/w \quad w]$$

permet d'exprimer les transformations sous forme matricielle.

Exemple : **translation** de vecteur $\vec{t} = (t_x, t_y)$: $\begin{cases} x' = x + t_x \\ y' = y + t_y \end{cases}$

Matrice de la transformation :

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \mathbf{t_x} & \mathbf{t_y} & 1 \end{bmatrix} = \begin{bmatrix} x + t_x & y + t_y & 1 \end{bmatrix}$$

Ecriture matricielle des transformations

Rotation de centre $(0, 0)$, d'angle α :

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} x\cos(\alpha) - y\sin(\alpha) & x\sin(\alpha) + y\cos(\alpha) & 1 \end{bmatrix}$$

Composition des transformations avec les matrices

Rotation de centre (c_x, c_y), d'angle α :

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -c_x & -c_y & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

$$= \begin{bmatrix} x' & y' & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

$$= \begin{bmatrix} x'' & y'' & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

$$= \begin{bmatrix} x''' & y''' & 1 \end{bmatrix}$$

Composition des transformations avec les matrices

Rotation de centre (c_x, c_y), d'angle α :

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -c_x & -c_y & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

$$= \begin{bmatrix} x' & y' & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

$$= \begin{bmatrix} x'' & y'' & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

$$= \begin{bmatrix} x''' & y''' & 1 \end{bmatrix}$$

Composition des transformations avec les matrices

Rotation de centre (c_x, c_y), d'angle α :

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -c_x & -c_y & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

$$= \begin{bmatrix} x' & y' & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

$$= \begin{bmatrix} x'' & y'' & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

$$= \begin{bmatrix} x''' & y''' & 1 \end{bmatrix}$$

Composition des transformations avec les matrices

Rotation de centre (c_x, c_y), d'angle α :

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -c_x & -c_y & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

$$= \begin{bmatrix} x' & y' & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

$$= \begin{bmatrix} x'' & y'' & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

$$= \begin{bmatrix} x''' & y''' & 1 \end{bmatrix}$$

Composition des transformations avec les matrices

$$\begin{bmatrix} x''' & y''' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -c_x & -c_y & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

Le produit de matrices est “associatif” :

$$AxBxC = (AxB)xC = Ax(BxC) \quad ,$$

$\Rightarrow$ il suffit de multiplier les matrices correspondant aux transformations successives.

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ -c_x\cos(\alpha) + c_y\sin(\alpha) & -c_x\sin(\alpha) - c_y\cos(\alpha) & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ -c_x\cos(\alpha) + c_y\sin(\alpha) + c_x & -c_x\sin(\alpha) - c_y\cos(\alpha) + c_y & 1 \end{bmatrix}$$

Composition des transformations avec les matrices

$$\begin{bmatrix} x''' & y''' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -c_x & -c_y & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

Le produit de matrices est “associatif” :

$$AxBxC = (AxB)xC = Ax(BxC) \quad ,$$

$\Rightarrow$ il suffit de multiplier les matrices correspondant aux transformations successives.

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ -c_x\cos(\alpha) + c_y\sin(\alpha) & -c_x\sin(\alpha) - c_y\cos(\alpha) & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ -c_x\cos(\alpha) + c_y\sin(\alpha) + c_x & -c_x\sin(\alpha) - c_y\cos(\alpha) + c_y & 1 \end{bmatrix}$$

Composition des transformations avec les matrices

$$\begin{bmatrix} x''' & y''' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -c_x & -c_y & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

Le produit de matrices est “associatif” :

$$AxBxC = (AxB)xC = Ax(BxC) \quad ,$$

$\Rightarrow$ il suffit de multiplier les matrices correspondant aux transformations successives.

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ -c_x\cos(\alpha) + c_y\sin(\alpha) & -c_x\sin(\alpha) - c_y\cos(\alpha) & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ -c_x\cos(\alpha) + c_y\sin(\alpha) + c_x & -c_x\sin(\alpha) - c_y\cos(\alpha) + c_y & 1 \end{bmatrix}$$

Composition des transformations avec les matrices

$$\begin{bmatrix} x''' & y''' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -c_x & -c_y & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

Le produit de matrices est “associatif” :

$$AxBxC = (AxB)xC = Ax(BxC) \quad ,$$

$\Rightarrow$ il suffit de multiplier les matrices correspondant aux transformations successives.

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ -c_x\cos(\alpha) + c_y\sin(\alpha) & -c_x\sin(\alpha) - c_y\cos(\alpha) & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ -c_x\cos(\alpha) + c_y\sin(\alpha) + c_x & -c_x\sin(\alpha) - c_y\cos(\alpha) + c_y & 1 \end{bmatrix}$$

Composition des transformations avec les matrices

But : faire des transformations en réutilisant les transformations de base

- soit en appliquant une à une les transformations élémentaires
- soit en calculant la matrice correspondante et en appliquant une seule transformation
  voir :
  http ://www.html5canvastutorials.com/advanced/html5-canvas-custom-transform/

Attention avec les matrices

- le produit de matrices n'est pas commutatif

$$A \times B \neq B \times A$$

- si on fait le produit dans l'autre sens : on écrit le vecteur en colonne et il faut "transposer" les matrices de transformation :

$$\begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \begin{bmatrix} ax + by \\ cx + dy \\ 1 \end{bmatrix}$$

et faire les produits de composition de transformation à

Attention avec les matrices

- le produit de matrices n'est pas commutatif

$$A \times B \neq B \times A$$

- si on fait le produit dans l'autre sens : on écrit le vecteur en colonne et il faut "transposer" les matrices de transformation :

$$\begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \begin{bmatrix} ax + by \\ cx + dy \\ 1 \end{bmatrix}$$

et faire les produits de composition de transformation à

Attention avec les matrices

- le produit de matrices n'est pas commutatif

$$A \times B \neq B \times A$$

- si on fait le produit dans l'autre sens : on écrit le vecteur en colonne et il faut "transposer" les matrices de transformation :

$$\begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \begin{bmatrix} ax + by \\ cx + dy \\ 1 \end{bmatrix}$$

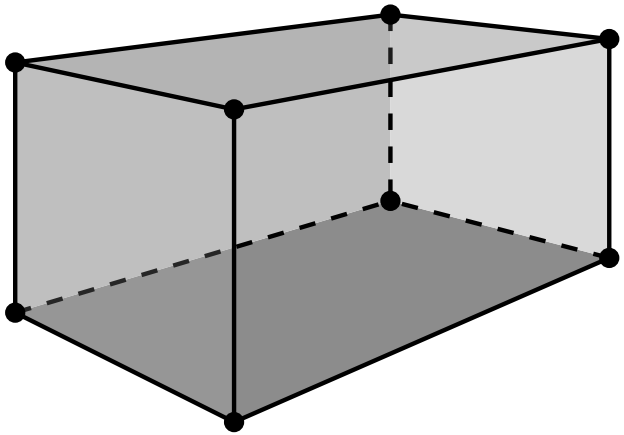
et faire les produits de composition de transformation à

Exercice 5

Calculer la matrice de la rotation de centre $(3, 4)$, d'angle $\pi/4$

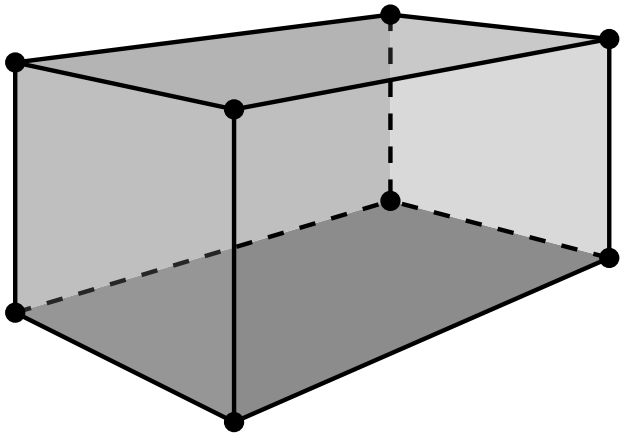
Lignes directrices

3D



Comme la 2D avec une composante en plus ?

3D



Comme la 2D avec une composante en plus ?
non :

- on affiche toujours en 2D
- la quantité de calculs augmente

Coordonnées homogènes

$$P(x,y,z) : [x \quad y \quad z \quad \mathbf{1}] = [x/w \quad y/w \quad z/w \quad w]$$

Exemple : **translation** de vecteur $\vec{t} = (t_x, t_y, t_z)$: $\begin{cases} x' = x + t_x \\ y' = y + t_y \\ z' = z + t_z \end{cases}$

$$[x \quad y \quad z \quad 1] \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix} = [x + t_x \quad y + t_y \quad z + t_z \quad 1]$$

Homogénéiser les coordonnées

$\Rightarrow$ écrire le vecteur avec $w = 1$.

Coordonnées homogènes

$$P(x,y,z) : [x \quad y \quad z \quad \mathbf{1}] = [x/w \quad y/w \quad z/w \quad w]$$

Exemple : **translation** de vecteur $\vec{t} = (t_x, t_y, t_z)$: $\begin{cases} x' = x + t_x \\ y' = y + t_y \\ z' = z + t_z \end{cases}$

$$[x \quad y \quad z \quad 1] \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix} = [x + t_x \quad y + t_y \quad z + t_z \quad 1]$$

Homogénéiser les coordonnées

$\Rightarrow$ écrire le vecteur avec $w = 1$.

Coordonnées homogènes

$$P(x,y,z) : [x \quad y \quad z \quad \mathbf{1}] = [x/w \quad y/w \quad z/w \quad w]$$

Exemple : **translation** de vecteur $\vec{t} = (t_x, t_y, t_z)$: $\begin{cases} x' = x + t_x \\ y' = y + t_y \\ z' = z + t_z \end{cases}$

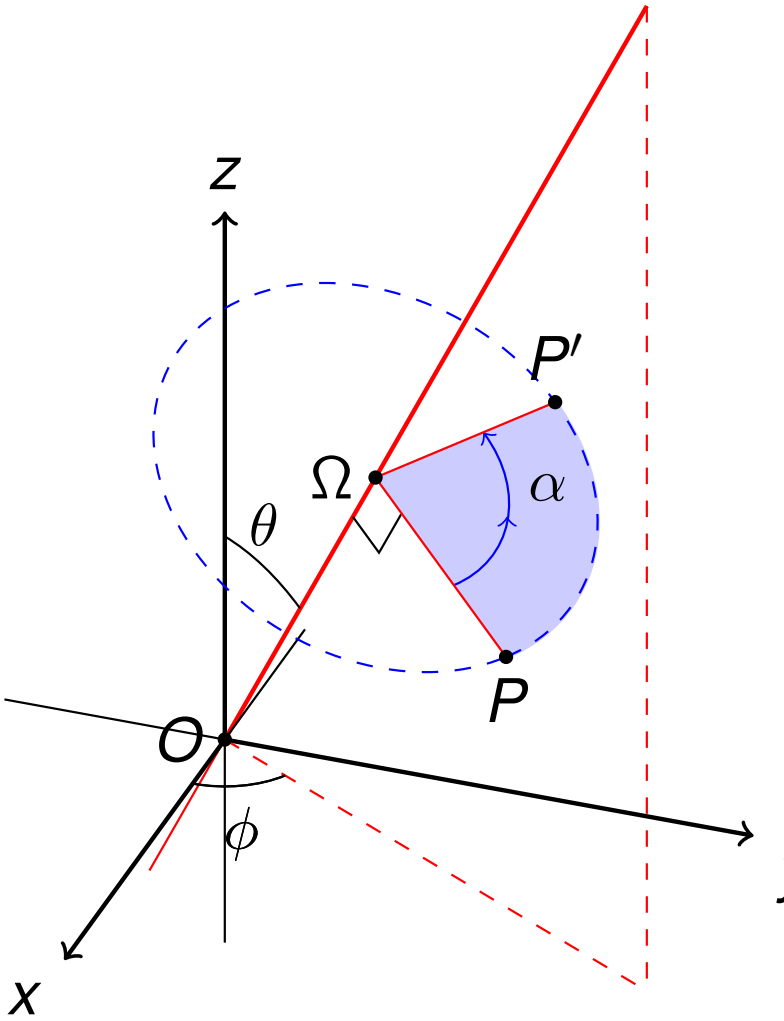
$$[x \quad y \quad z \quad 1] \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix} = [x + t_x \quad y + t_y \quad z + t_z \quad 1]$$

Homogénéiser les coordonnées

$\Rightarrow$ écrire le vecteur avec $w = 1$.

Rotations 3D

Les rotations sont autour d'un axe

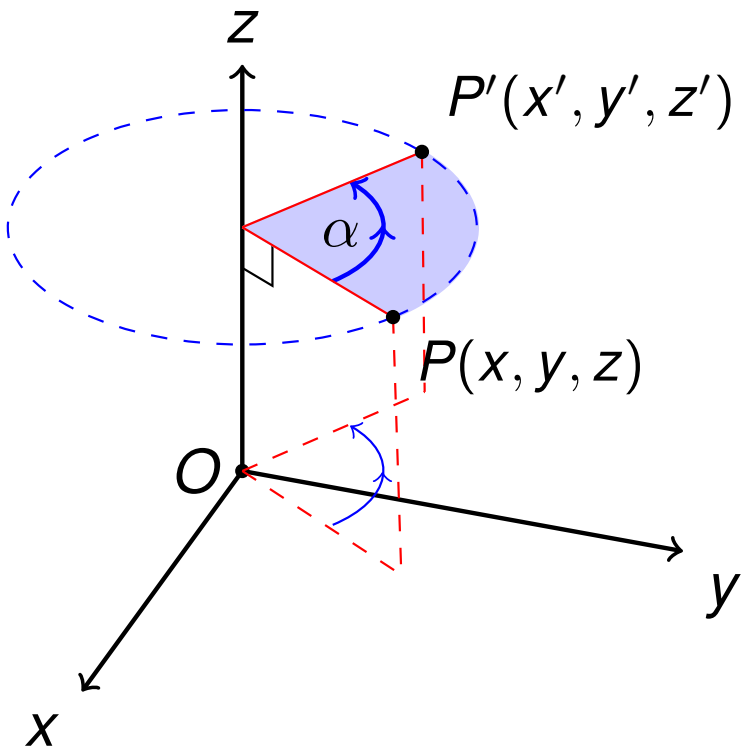


Ω est le point tel que $\vec{O\Omega} \perp \vec{\Omega P}$
Comme en 2D, il faut décomposer en transformations simples :

- "amener" le centre Ω à l'origine du repère (translation)
- aligner l'axe de rotation avec un axe du repère
- faire la transformation simple
- reorienter l'axe de rotation
- "ramener" Ω à sa place.

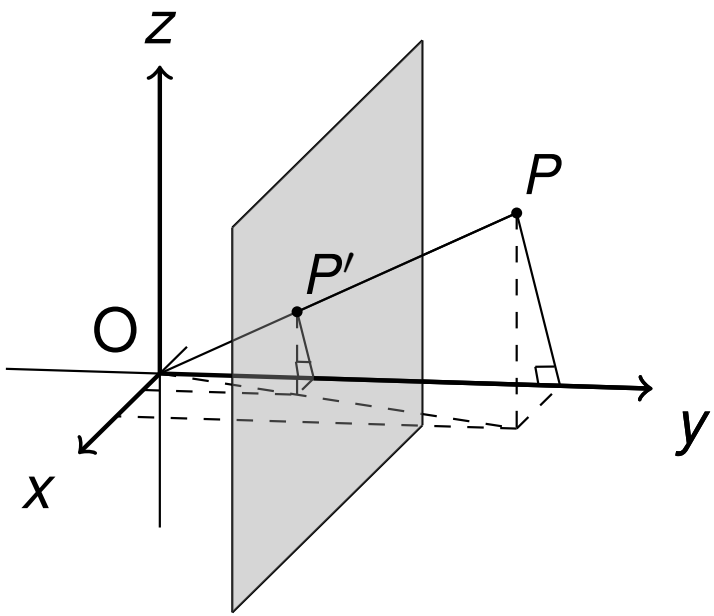
Rotations 3D

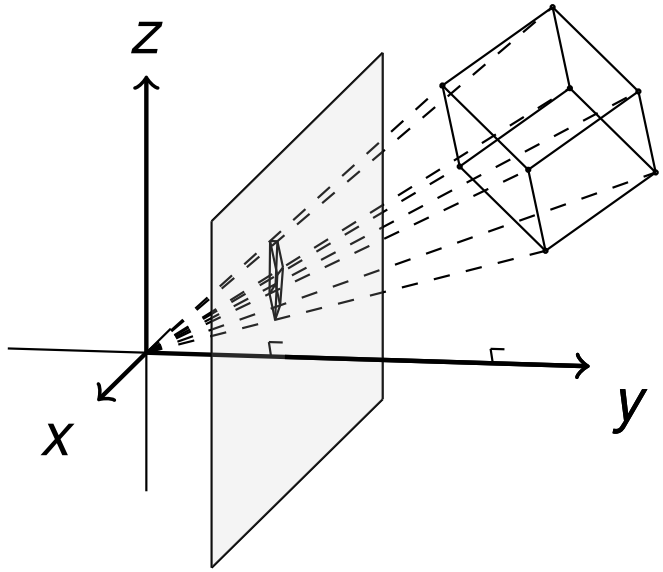
Rotations autour de l'axe Oz

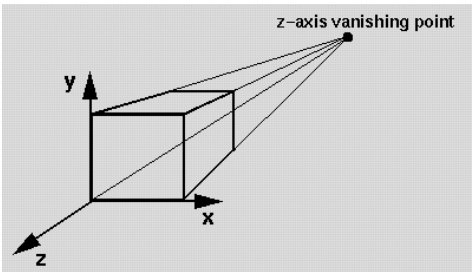


Donner la matrice de rotation autour de l'axe Oz.

Projections : Projection perspective

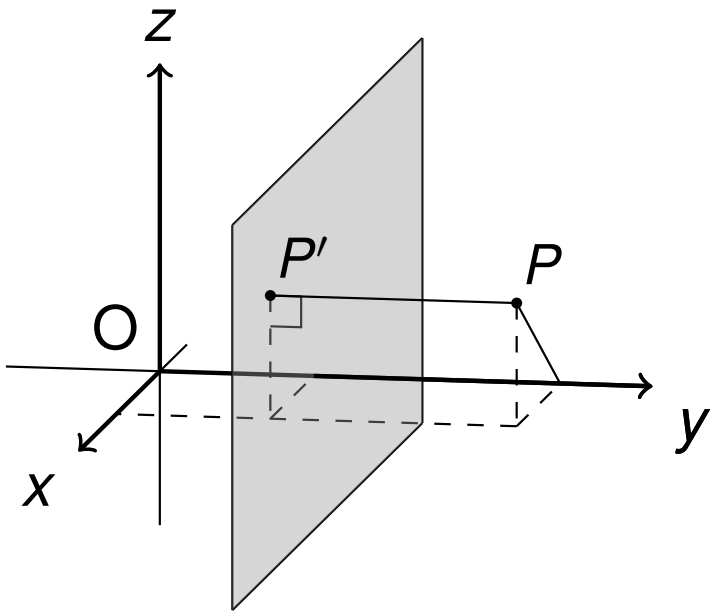


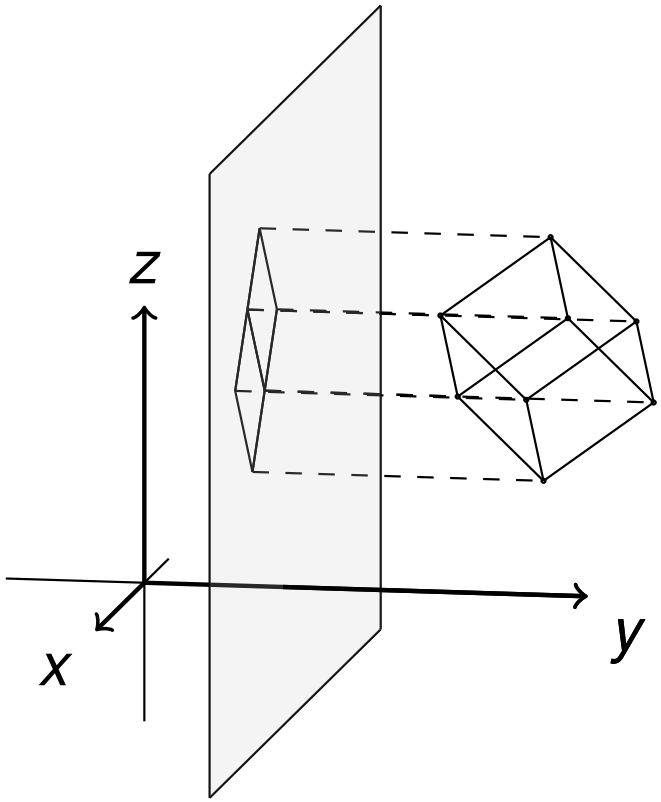




Exercice : Calculer la matrice de la projection perspective avec $(0, 0, 0)$ comme point de fuite, dans le plan $y = 10$.

Projections : Projection Orthogonale

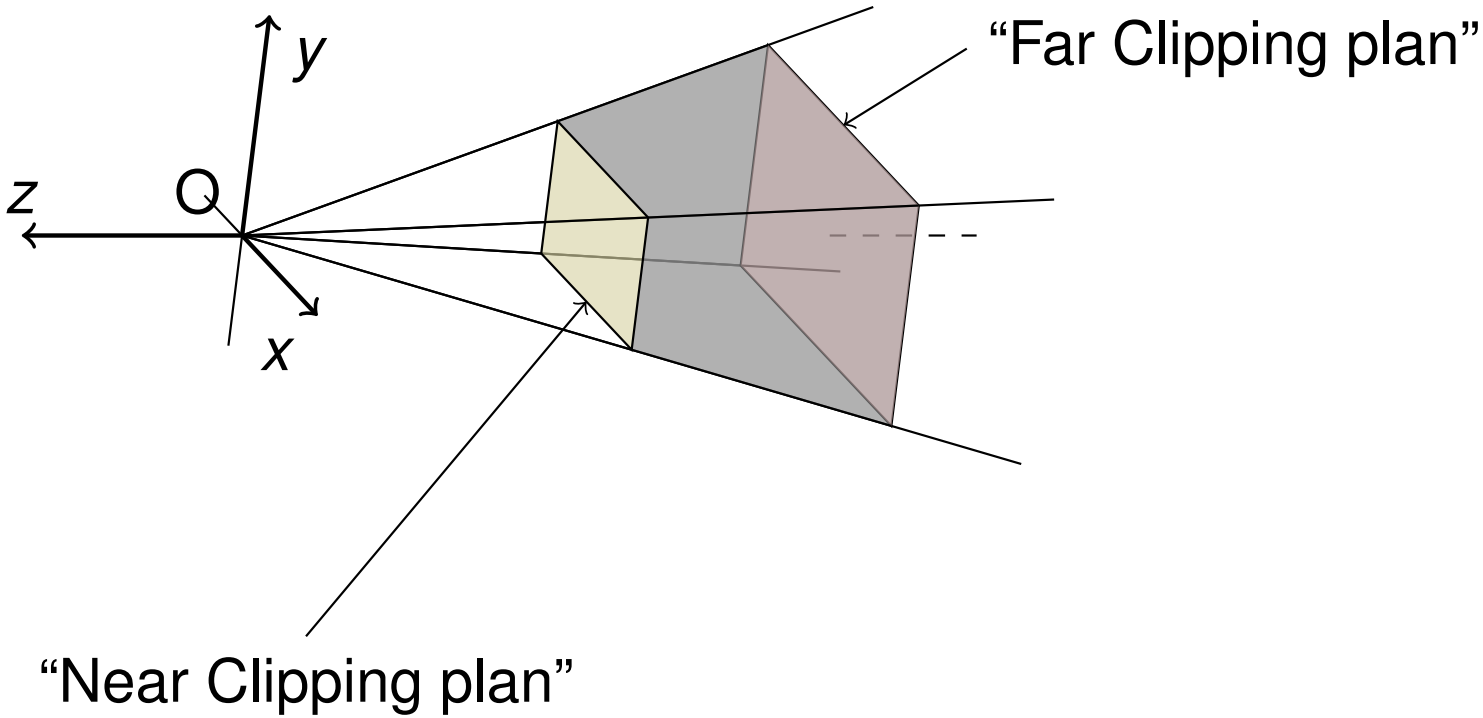




$$\begin{cases} x' = x \\ y' = 10 \\ z' = z \end{cases}$$

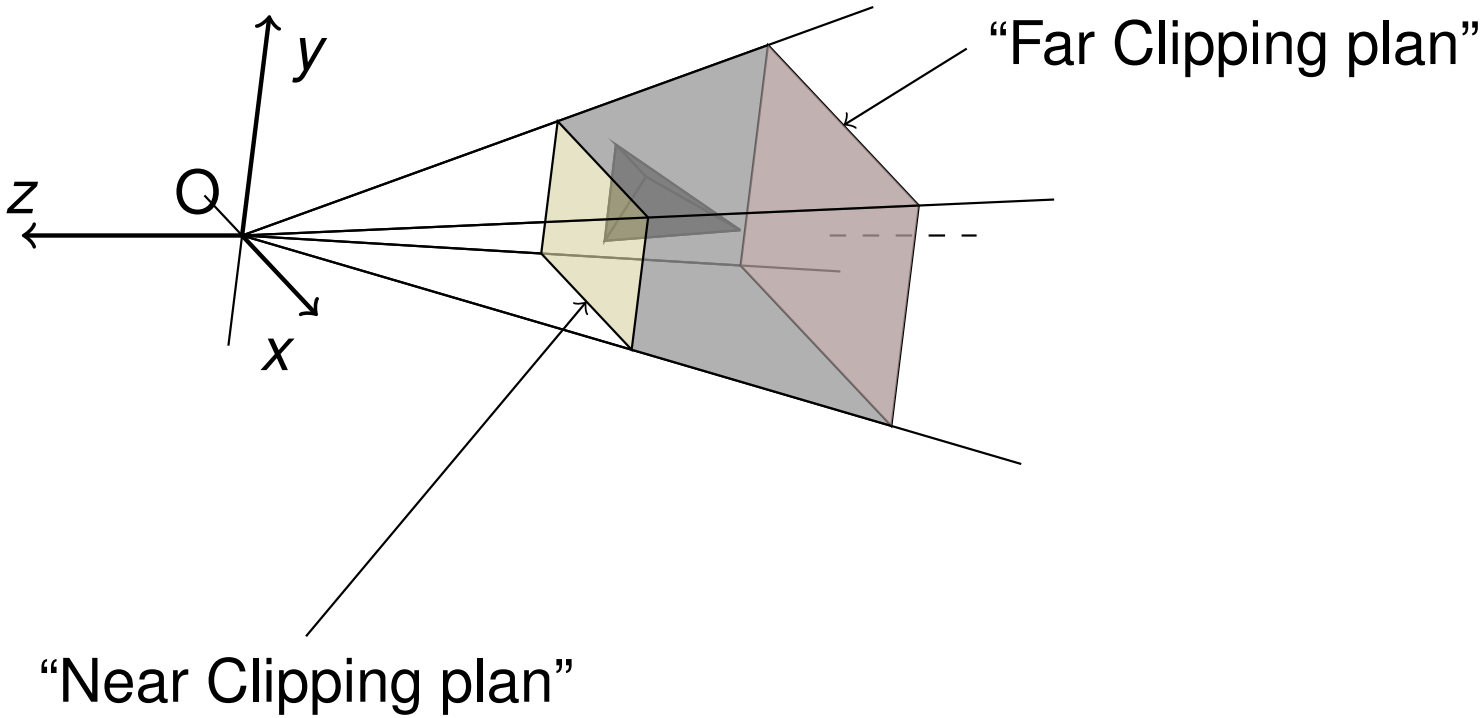
Projections : Vue Camera + Clipping

On utilise généralement la projection perspective



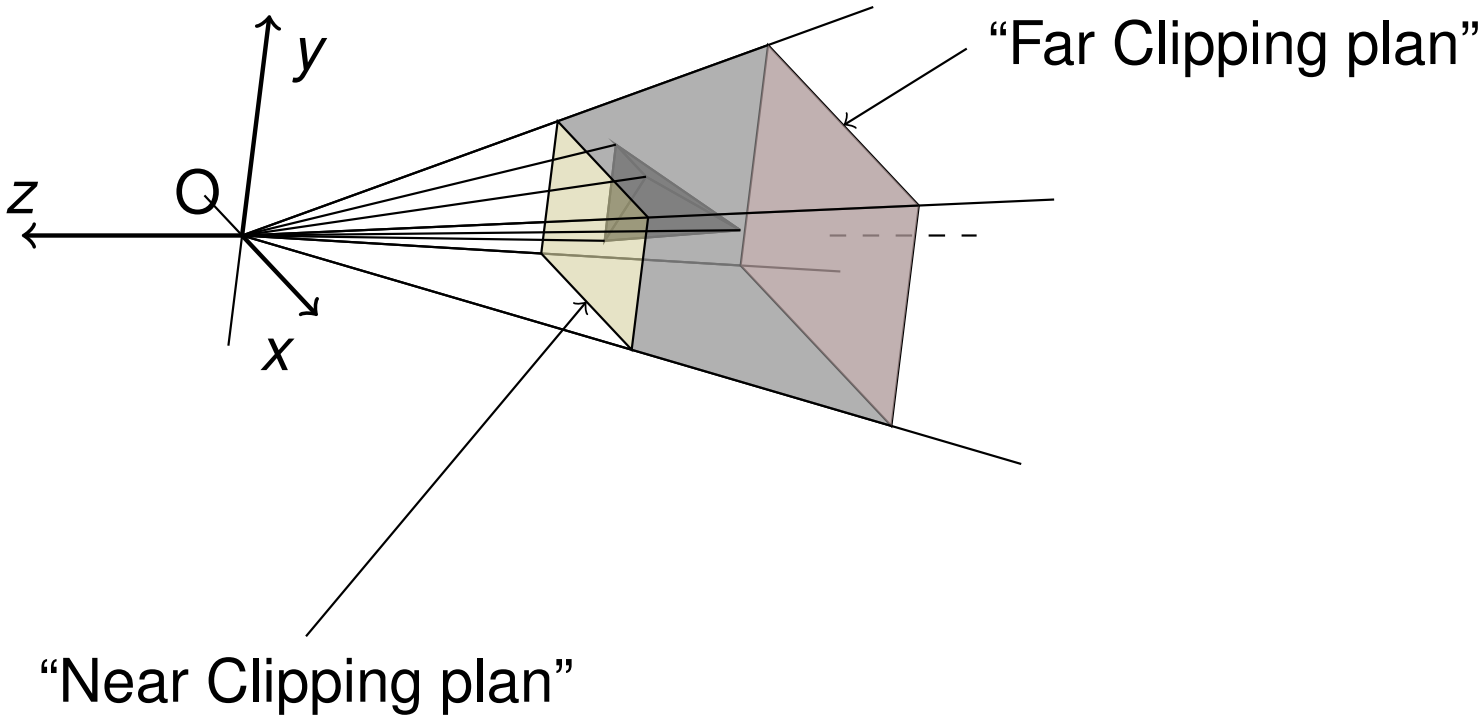
Projections : Vue Camera + Clipping

On utilise généralement la projection perspective



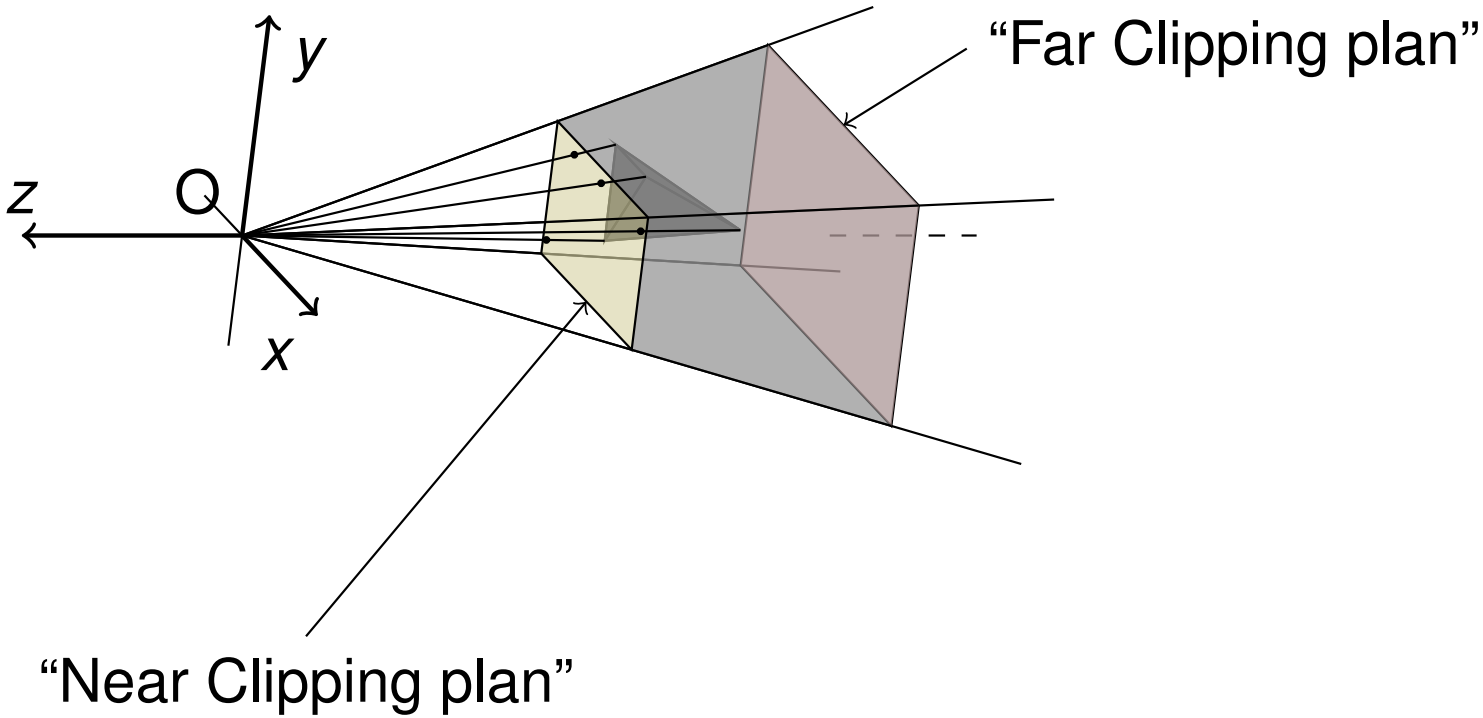
Projections : Vue Camera + Clipping

On utilise généralement la projection perspective



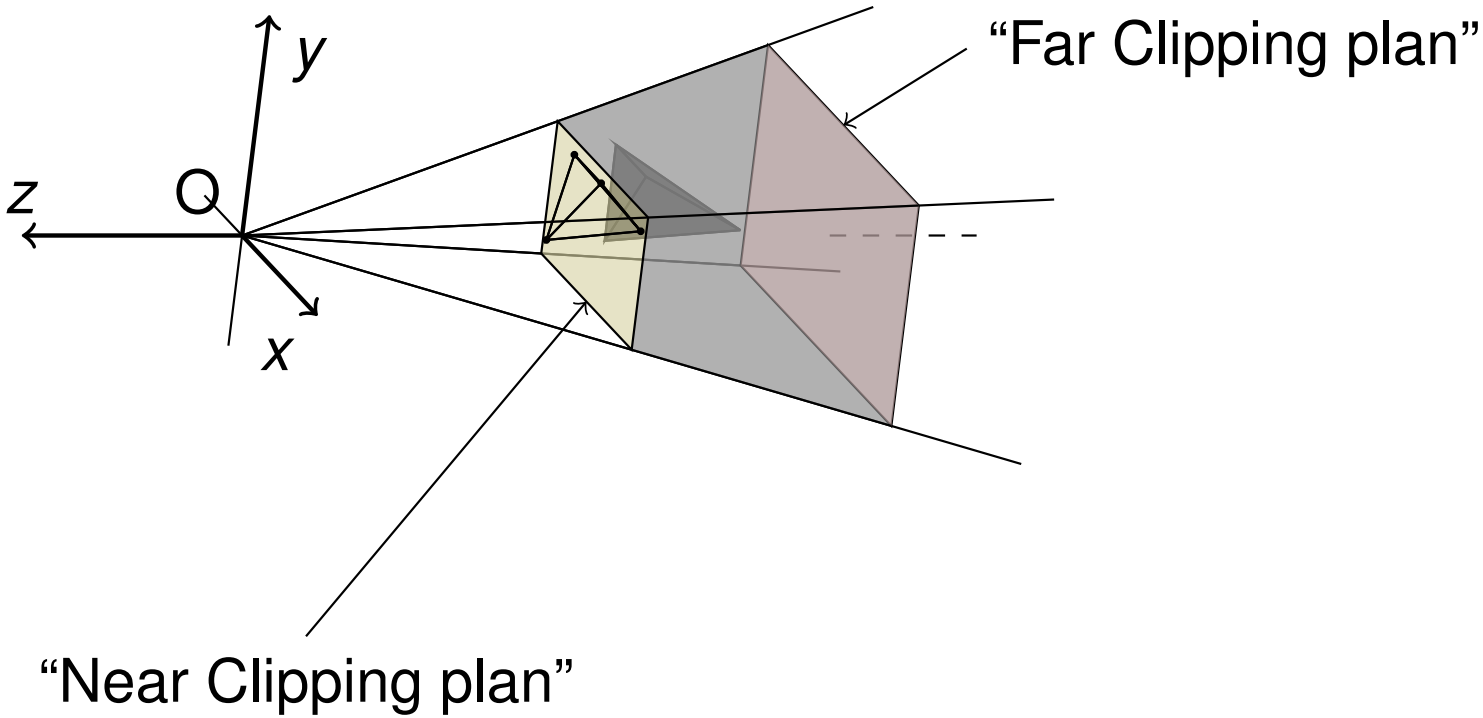
Projections : Vue Camera + Clipping

On utilise généralement la projection perspective

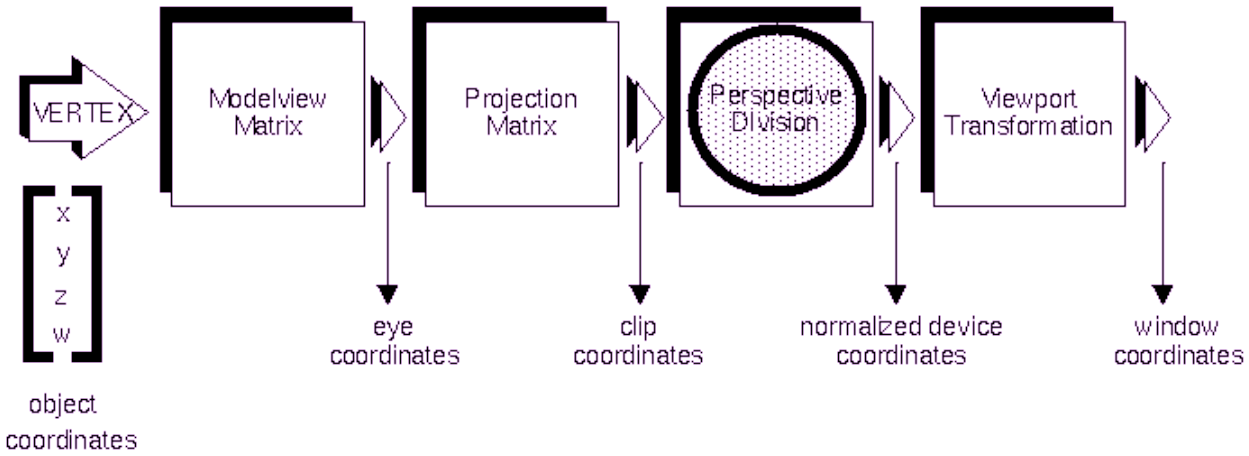


Projections : Vue Camera + Clipping

On utilise généralement la projection perspective

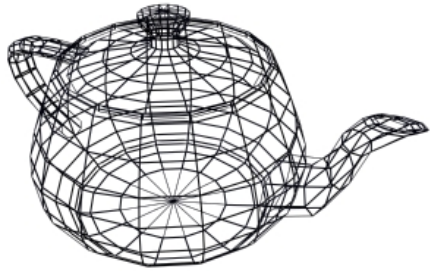


Pipeline 3D

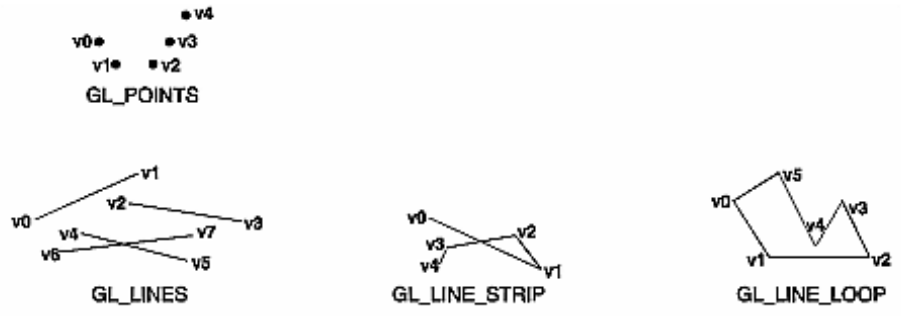

VERTEX
x
y
z
w
object coordinates
Modelview Matrix
eye coordinates
Projection Matrix
clip coordinates
Perspective Division
normalized device coordinates
Viewport Transformation
window coordinates

Les objets 3D : mode fil de fer

Définit par des lignes, le plus simple :

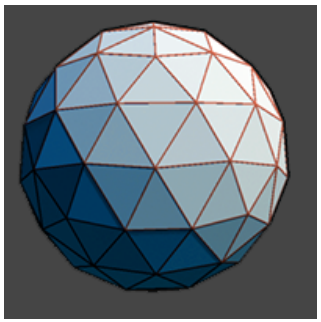


en OpenGL, avec GL_POINTS, **GL_LINES**, GL_LINE_STRIP, L_LINE_LOOP.

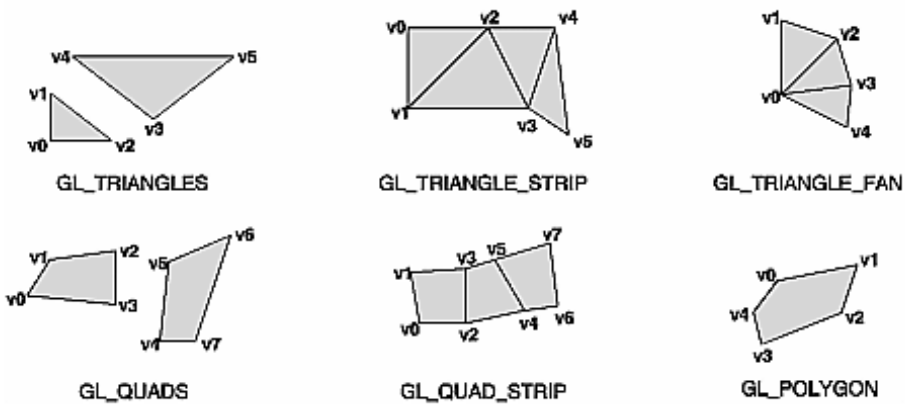


Objets 3D : polygones

Définit par des facettes (polygones) :



en OpenGL, primitives de polygones : GL_TRIANGLES, GL_QUADS, GL_TRIANGLE_STRIP, GL_TRIANGLE_FAN



Lignes directrices

OpenGL

http ://www.opengl.org/

- API multi-plateforme pour les images 3D
- primitives géométriques simples
- interprétables par le hardware (cartes graphiques)

http ://www.khronos.org/webgl/

WebGL version d'OpenGL (ES)

- intégration en page web (HTML5 Canvas)
- interaction Javscript.
- libre de droit

WebGL : start

Elements importants :

- **buffer**, structure de données simple (pour stocker des points par exemple)
- **matrices** de projection et de transformation (model view)

```
var modelViewMatrix = new Float32Array(
 [1, 0, 0, 0,
  0, 1, 0, 0,
  0, 0, 1, 0,
  0, 0, -5, 1]); // translation de -5 en 'z'
var projectionMatrix = new Float32Array(
 [1, 0, 0, 0,
  0, 1, 0, 0,
  0, 0, 1, -1,
  0, 0, 0, 0]);
```

- **shader** permet de définir comment seront dessiner les

WebGL : Hello World

Récupérer le **context** par le `canvas`

```
var canvas = document.getElementById("mon-canvas");

var gl;

try {
  gl = canvas.getContext("experimental-webgl");
} catch (e) {
  var msg = "Error get WebGL Context:" + e.toString();
  alert(msg);
  throw Error(msg);
}
```

WebGL : Hello World

Définir l'espace visible : **viewport**

```
gl.viewport(0, 0, canvas.width, canvas.height);
```

Nettoyage :

```
// on met le fond en noir (opaque)
gl.clearColor(0.0, 0.0, 0.0, 1.0);
// nettoyage des buffers
gl.clear(gl.COLOR_BUFFER_BIT);
```

WebGL : Hello World

Définir un tableau de points (**buffer**)

```
// creation d'un buffer
 var vertexBuffer = gl.createBuffer();
// definition en tant que buffer courant
 gl.bindBuffer(gl.ARRAY_BUFFER, vertexBuffer);
// on definit un tableau de points
 var vertices = new Float32Array([ 1,1,1,-1,-1,-1 ]);
// on met ces valeurs dans le buffer courant
 gl.bufferData(gl.ARRAY_BUFFER, vertices,
    gl.STATIC_DRAW);
```

```
[ 1,1,1, -1,-1,-1 ]
```

Il faudra se souvenir que ce **buffer** correspond à

- 2 points
- chaque point a 3 coordonnées

WebGL : Hello World

Dessiner un objet, ici un trait avec **GL_LINES**

```
// place le buffer a dessiner dans le buffer courant
gl.bindBuffer(gl.ARRAY_BUFFER, vertexBuffer);
// definition du 'shader' a utiliser
gl.useProgram(shaderProgram);
// specification des sommets au 'shaderProgram'
//  --> chaque sommet a 3 composantes
gl.vertexAttribPointer(shaderVertexPositionAttribute,
     3, gl.FLOAT, false, 0, 0);
// matrice de projection,  matrice de transformation
gl.uniformMatrix4fv(shaderProjectionMatrixUniform,
    false, projectionMatrix);
gl.uniformMatrix4fv(shaderModelViewMatrixUniform,
    false, modelViewMatrix);
// on dessine en indiquant :
// - le mode (GL_LINES)
// - la position de depart dans le buffer (0)
```

WebGL : shader

3 étapes :

1. définition du code source C pour le **vertexShader** et le **fragmentShader**
2. association WebGL du code et **compilation** par le navigateur
3. **initialisation** et définitions pour utilisation dans le programme JavaScript

WebGL : shader

```
var vertexShaderSource = "attribute vec3 vertexPos;\n"+
 "uniform mat4 modelViewMatrix;\n" +
 "uniform mat4 projectionMatrix;\n" +
 "void main(void) {\n" +
 "gl_Position = projectionMatrix*modelViewMatrix*\n" +
 "vec4(vertexPos, 1.0);\n" +
 "}\n";

var shader = gl.createShader(gl.VERTEX_SHADER);
gl.shaderSource(shader, str);
gl.compileShader(shader);

if (!gl.getShaderParameter(shader, gl.COMPILE_STATUS))
    {
  alert(gl.getShaderInfoLog(shader));
  return null;
}
```

Shader

Assez indigeste, ce qu'il faut en retenir :

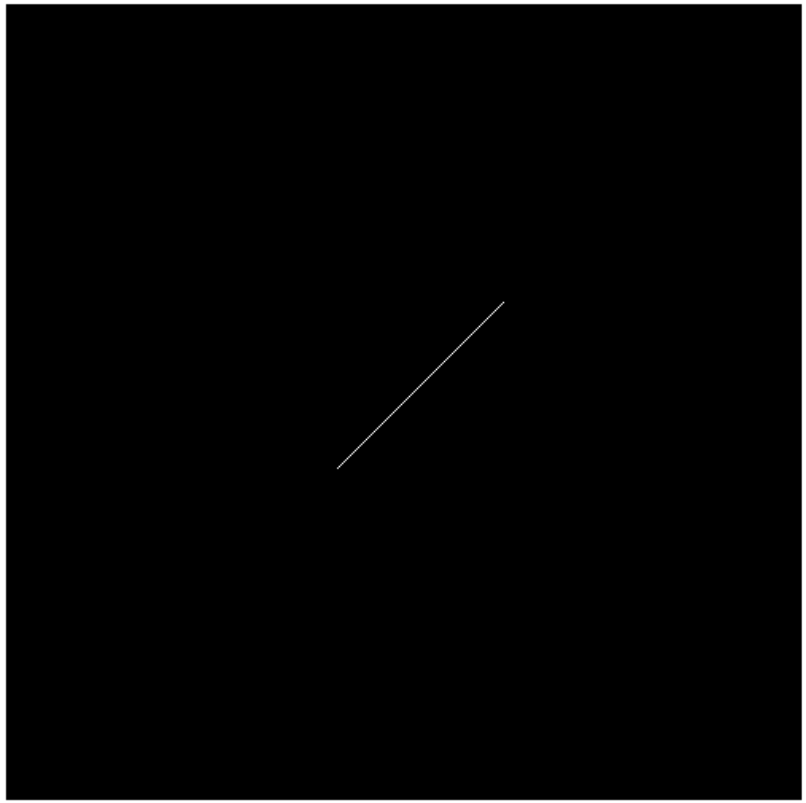
- définit comment doit être dessiner un sommet, une facette
- est compilé avant l'execution pour la rapidité
- à mettre dans des fonctions :

```
var vertexShaderSource = " ... ";
var fragmentShaderSource = " ... ";
function createShader(gl, str, type) {
  // ...
}
function initShader(gl) {
  // ...
}
```

http ://home.mis.u-picardie.fr/∼choplin/enseignement/webgl/shader.js

WebGL : un trait

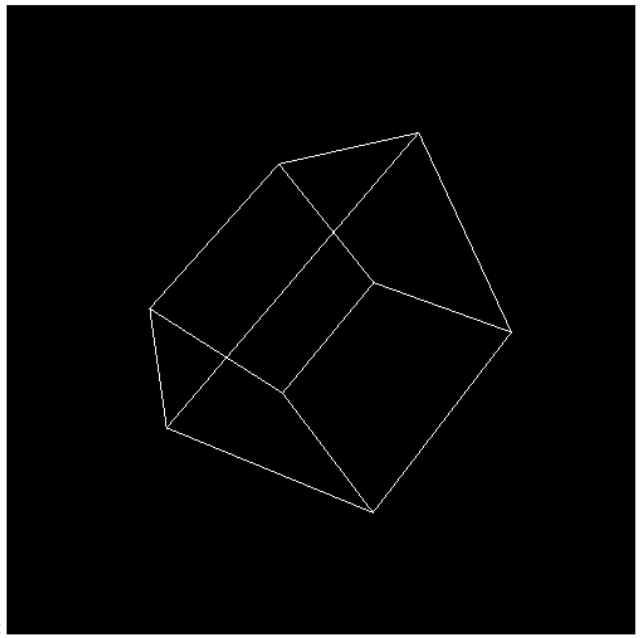
http ://home.mis.u-picardie.fr/∼choplin/enseignement/webgl/trait.html



canvas:

Logs
fini
set viewport
init shader
start onLoad

Premier Programme en WebGL



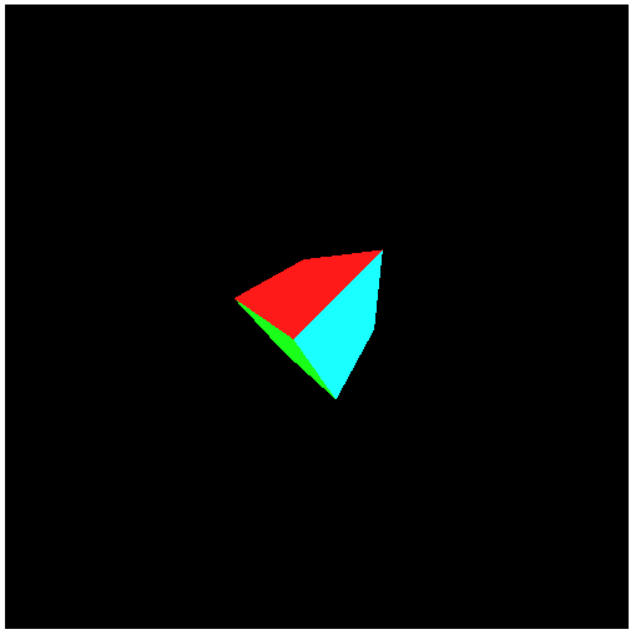
1. créer un cube en mode fil de fer
2. le faire tourner avec une animation

Utiliser la fonction `multiplication` pour faire le produit des matrices

http ://home.mis.u-picardie.fr/∼choplin/enseignement/webgl/matrice.js

Les facettes

Un cube avec des facettes en couleur



le canvas::
Logs
square created
got gl
got canvas
start

1. définir le cube avec des triangles
2. modifier le shader pour prendre en charge les couleurs
3. activer le **z-buffer**

```
gl.enable(gl.DEPTH_TEST);
```

Le shader pour les facettes

```
var vertexShaderSource =
  "attribute vec3 vertexPos;\n"+
  "attribute vec4 vertexColor;\n"+
  "uniform mat4 modelViewMatrix;\n" +
  "uniform mat4 projectionMatrix;\n" +
  "varying vec4 vColor;\n"+
  "void main(void) {\n"+
  "  gl_Position = projectionMatrix *\n"+
  "   modelViewMatrix * vec4(vertexPos, 1.0);\n" +
  "  vColor = vertexColor;\n"+
  "}";
var fragmentShaderSource =
  " precision mediump float;\n"+
  " varying vec4 vColor;\n"+
  " void main(void) {\n" +
  "  gl_FragColor = vColor;\n"+
  "}\n";
```

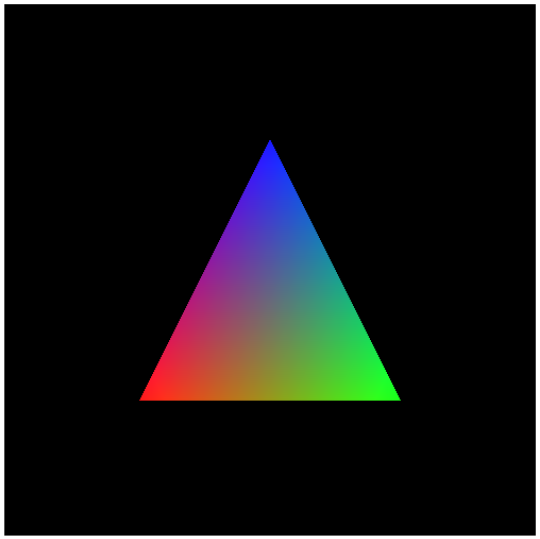
Le shader pour les facettes

Il faut indiquer à l'intialisation du shader, les nouveaux attributs C pour pouvoir les manipuler ensuite en JavaScript.

```
//... a l'initalisation du shader
    shaderVertexColorAttribute = gl.getAttribLocation(
        shaderProgram, "vertexColor");
    gl.enableVertexAttribArray(
        shaderVertexColorAttribute);
//... a la creation des sommets de l'objet,
//... on leur attribue une couleur
    var colorBuffer = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, colorBuffer);
    gl.bufferData(gl.ARRAY_BUFFER, new Float32Array(
        colors), gl.STATIC_DRAW);
//... pour l'affichage, on applique le buffer courant
    gl.vertexAttribPointer(shaderVertexColorAttribute,
        4, gl.FLOAT, false, 0, 0);
```

Couleur d'un triangle

La couleur d'un triangle est définie par la couleur aux trois sommets et une interpolation sur les autres points.



le canvas::
Logs
obj created
got gl
got canvas
start

Exercice : dessiner un triangle coloré en WebGL

le code du shader :
http ://home.mis.u-picardie.fr/∼choplin/enseignement/webgl/shader2.js

Mettre les shader dans le HTML

Une méthode un peu plus lisible pour définir les shader :

1. les mettre dans un élément du doc HTML,
2. aller les récupérer en JavaScript

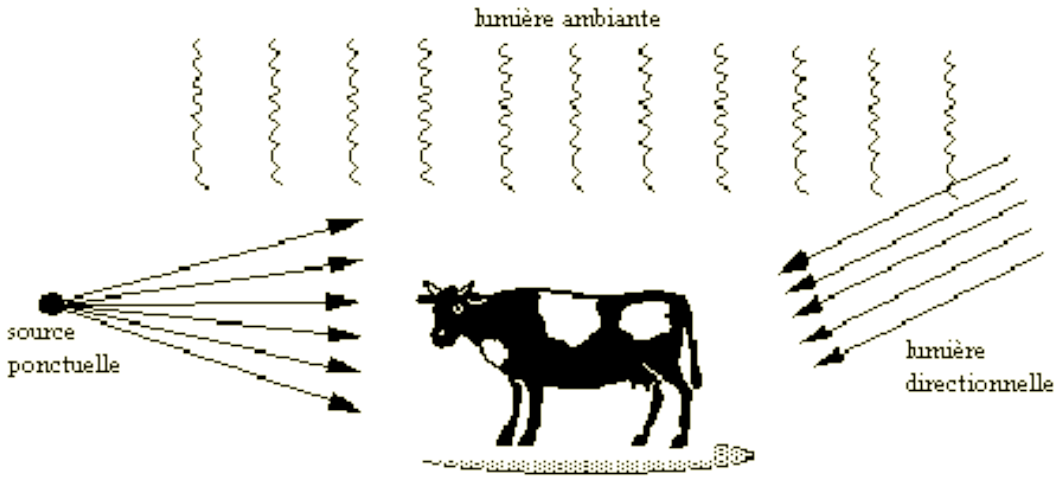
```
<html><head>
<script id="shader-vs" type="x-shader/x-vertex">
  attribute vec3 vertexPos;
  uniform mat4 modelViewMatrix;
  uniform mat4 projectionMatrix;
  void main(void) {
   gl_Position = projectionMatrix * modelViewMatrix * vec4(ver
  }
</script>
<script id="shader-fs" type="x-shader/x-fragment">
 void main(void) {
   gl_FragColor = vec4(1.0, 1.0, 1.0, 1.0);
 };
</script>
```

Mettre les shader dans le HTML

dans le JavaScript :

```
function getShader(gl, id) {
  // recupere l'element du doc HTML
  var shaderScript = document.getElementById(id);
  // recupere le contenu (le code source du shader)
  var theSource = shaderScript.firstChild.textContent;
  // cree le shader en donnant
  //    le code source et le type
  return createShader(gl,theSource,shaderScript.type);
}
```

Eclairage



Modèle de Phong sur les objets : $I = I_a + I_d + I_s$

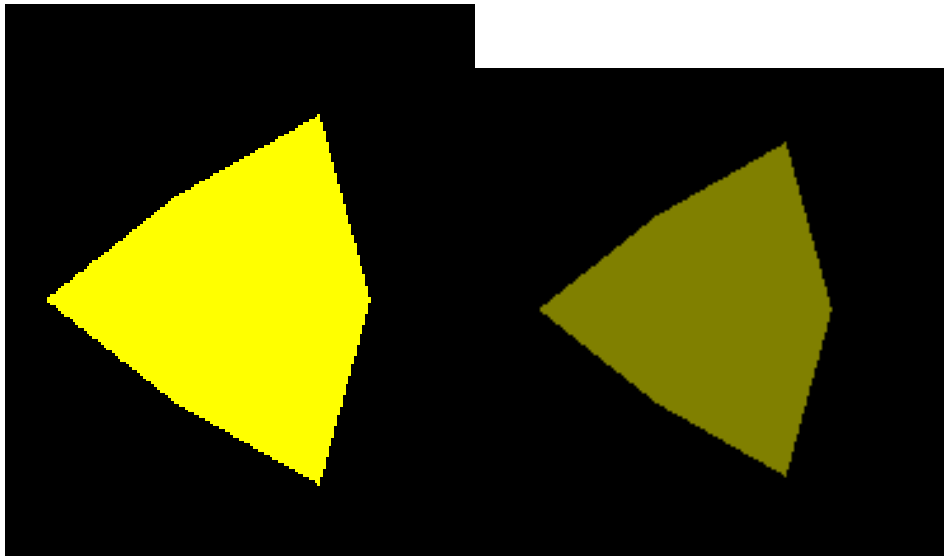
- I_a : lumière ambiante
- I_d : lumière diffusée à la surface
- I_s : lumière spéculaire / reflet

Calcul en chaque pointé : coûteux.

Eclairage : éclairage ambiant

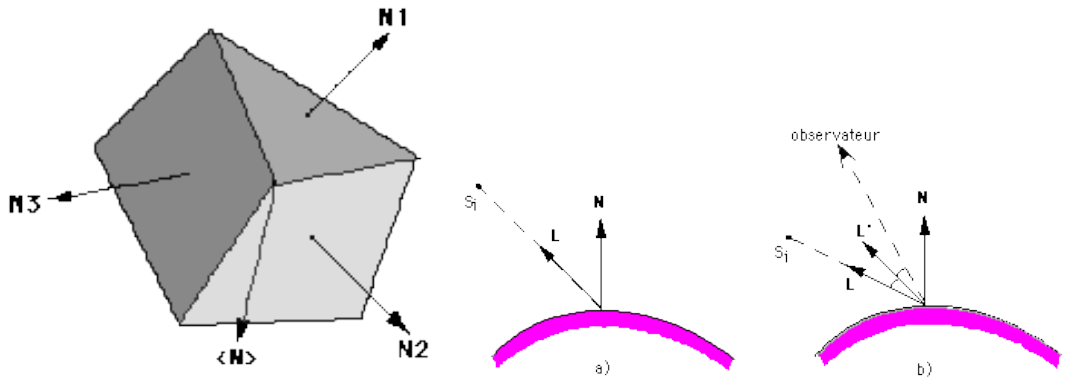
Principe : on multiplie l'intensité des composantes des couleurs par les données de la couleur, par exemple :

- Lumiere ambiante : $(0.5, 0.5, 0.5)$
- Couleur (en RGB) : $(1.0, 1.0, 0.0)$ (jaune)
- $\Rightarrow$ couleur affichée : $(0.5, 0.5, 0.0)$

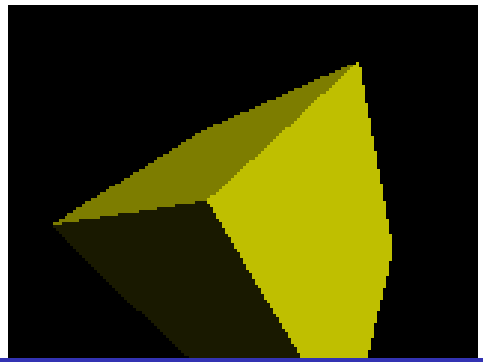


Eclairage : éclairage directionnel

Modèle de Lambert, tous les points de la surface ont la même intensité lumineuse
Principe : on mesure l'angle entre la normale à la surface et la direction de la lumière



L'intensité de la couleur est multipliée par le cosinus de l'angle (le produit scalaire si les vecteurs sont normalisés)



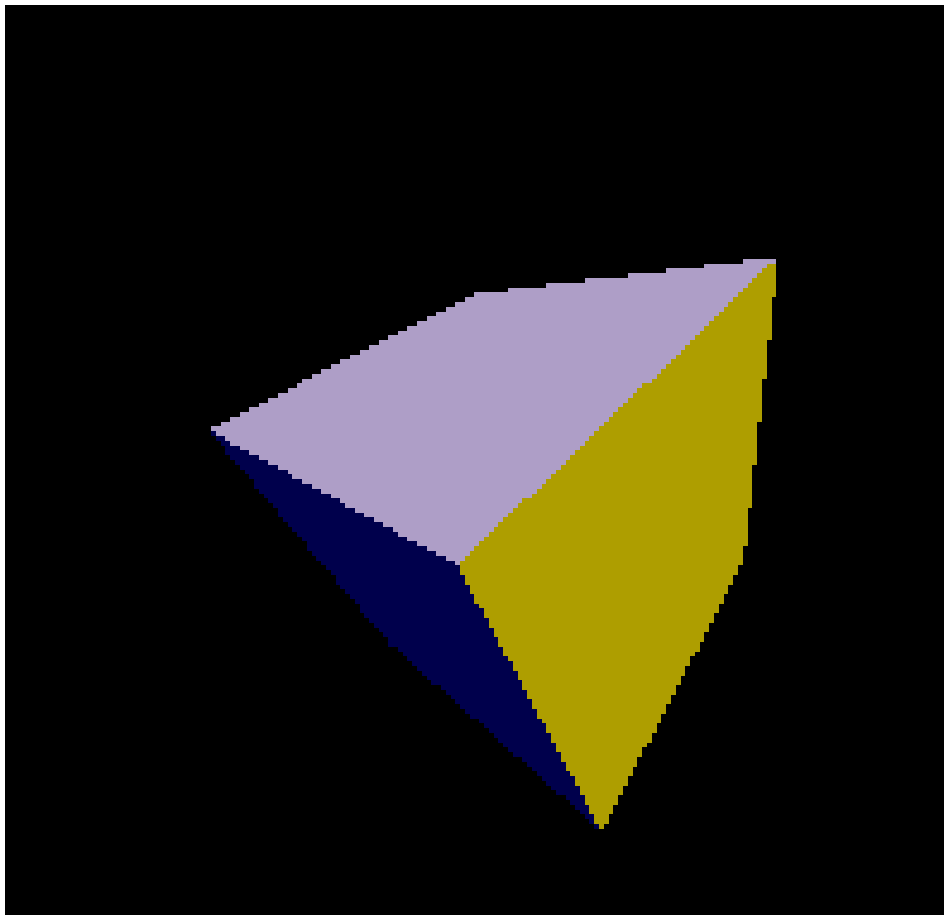
Eclairage : shader

```
<script id="shader-fs" type="x-shader/x-fragment">
 precision mediump float;
 varying vec3 vLighting;
 varying vec4 vColor;

 void main(void) {
  gl_FragColor = vec4(vColor.xyz*vLighting,vColor.a);
 }
</script>
```

Eclairage : shader

```
<script id="shader-vs" type="x-shader/x-vertex">
  attribute vec3 vertexNormal;
  varying vec3 vLighting;
  uniform mat4 uNormalMatrix;
  ...
  void main(void) {
    ...
    vec3 ambientLight = vec3(0.3, 0.3, 0.3);
    vec3 directionalLightColor = vec3(0.6,0.5,0.75);
    vec3 directionalVector = normalize(vec3(1,1,1));

    vec4 transformedNormal = uNormalMatrix * vec4(
        vertexNormal, 1.0);
    float directional = max(dot(transformedNormal.xyz,
        directionalVector), 0.0);
    vLighting = ambientLight + (directionalLightColor *
         directional);
```

Exercice : éclairer votre cube



Interaction clavier/souris

Comme en javascript ! !

```
document.onkeypress=function(e){
 var e=window.event || e;
 if ( 'y' == String.fromCharCode(e.charCode)){
   multiplication(modelViewMatrix,rotationY);
 }
 if ( 'z' == String.fromCharCode(e.charCode)){
   multiplication(modelViewMatrix,rotationZ);
 }
 ...
}

 element.onmousedown = function(ev){ ... };
 element.onmouseup = function(ev){ ... };
 element.onmousemove = function(ev){ ... };
```

`element` peut être un élement en dehors du canvas (pour faire

Exercice

Dessiner un trait pour représenter la direction de la lumière, ajouter un contrôle clavier pour modifier sa direction.

Textures