

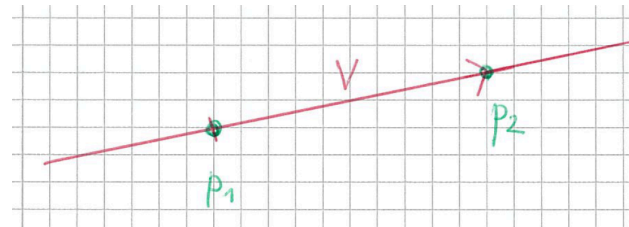
Points, Vecteurs Perspective

Point, Vecteur

```
typedef point float[2];  
typedef vecteur float[2];
```

```
p1, p2: point;
```

```
V: vecteur;
```



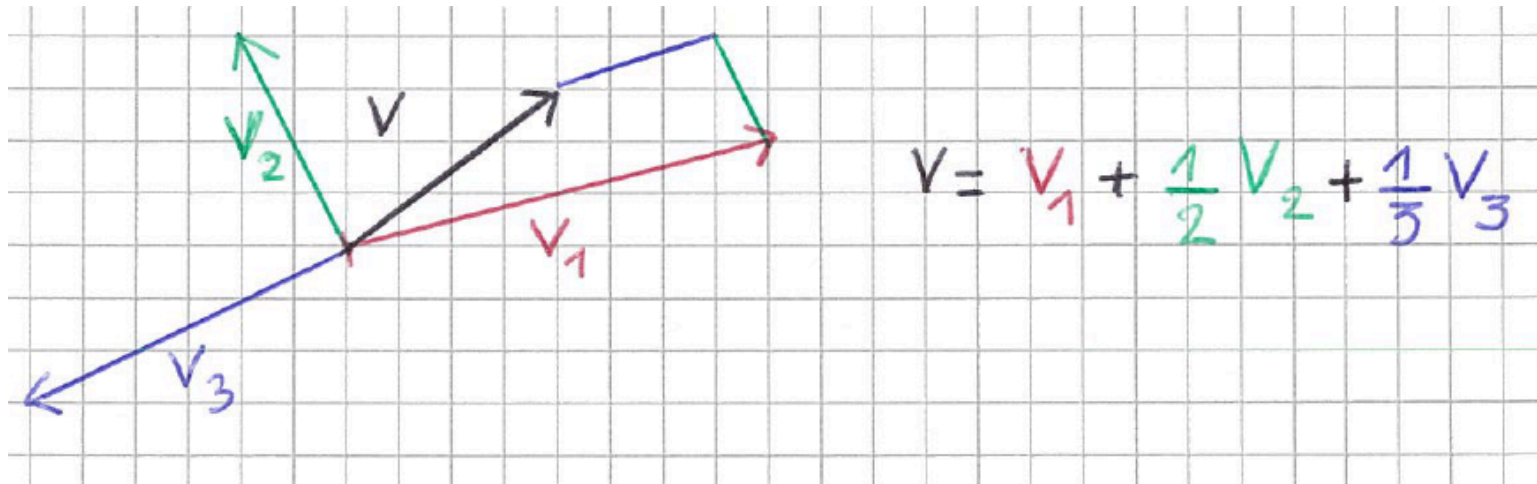
```
V = p2 - p1; // V[0] = p2[0] - p1[0]
```

```
           // V[1] = p2[1] - p1[1]
```

```
p2 = p1 + V;
```

Opérations entre vecteurs

- Toute combinaison linéaire

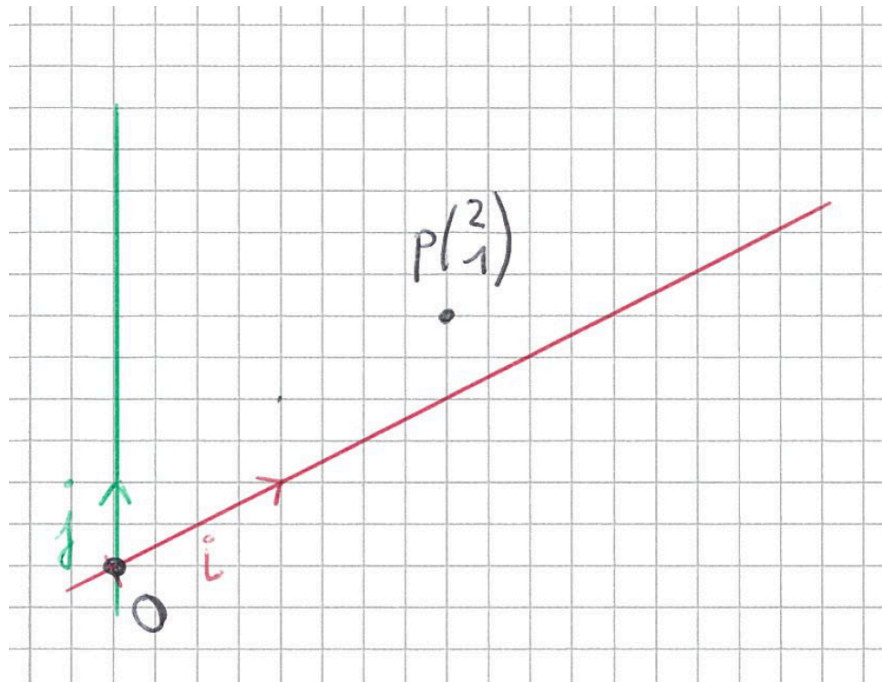


$$V = \alpha_1 V_1 + \cdots + \alpha_n V_n$$

$\alpha_1, \cdots, \alpha_n$ n réels

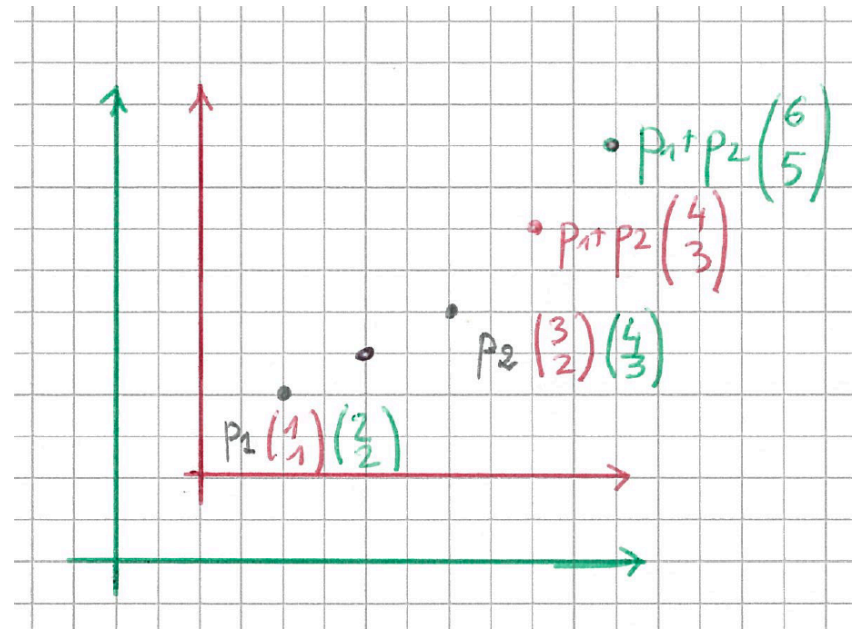
Coordonnées cartésiennes

- Repère: un point O (origine), deux vecteurs i, j
- $P = O + x * i + y * j$; // $P=[x,y]$



Point \neq Vecteur

- Opérations interdites = opérations dépendantes du repère cartésien
- Exemple: $P = P_1 + P_2$



Opérations entre points

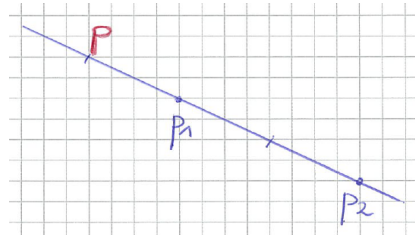
- Toute combinaison barycentrique

$$p = \alpha_1 p_1 + \cdots + \alpha_n p_n$$

$$\alpha_1 + \cdots + \alpha_n = 1$$

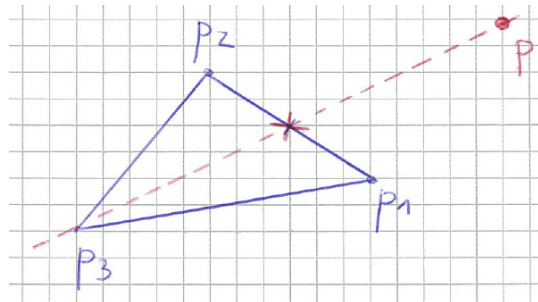
- Combinaison barycentrique de 2 points:

$$p = \frac{3}{2}p_1 - \frac{1}{2}p_2$$



- Combinaison barycentrique de 3 points:

$$p = p_1 + p_2 - p_3$$



Transformations affines

- Transformation des points de l'espace telle que
 - un point devient un point
 - Une ligne devient une ligne
 - Un plan devient un plan
 - Le parallélisme est conservé

Ecriture matricielle

- Transformation affine des points:

$$P_2 = M P_1 + V$$

M	Matrice 2x2
V	Matrice 2x1, vecteur
P_1, P_2	Matrice 2x1, points

- Pas valable pour transformer des vecteurs
 - Les vecteurs doivent être inchangés par les translations

Ecriture matricielle homogène

- Ajout d'une coordonnée
 - Point : $[x,y] \Rightarrow [x,y,1]$
 - Vecteur: $[x,y] \Rightarrow [x,y,0]$

$$\begin{pmatrix} x_2 \\ y_2 \\ 0 \text{ ou } 1 \end{pmatrix} = \begin{bmatrix} m_{11} & m_{12} & v_x \\ m_{21} & m_{22} & v_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 0 \text{ ou } 1 \end{pmatrix}$$

- OpenGL: matrices 4x4

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & v_x \\ m_{21} & m_{22} & m_{23} & v_y \\ m_{31} & m_{32} & m_{33} & v_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transformations en OpenGL

Librairie glm OpenGL Mathematics

```
glm::mat4 MaMatrice;
```

- `glm::translate(vx, vy, vz);`
 - Translation du vecteur $V = [vx, vy, vz]$

```
glm::vec3 AxeDeRotation( vx, vy, vz);
```

```
glm::rotate( angle, AxeDeRotation );
```

- Rotation autour de l'axe passant par l'origine et porté par le vecteur $[vx, vy, vz]$. `angle` en degrés.

- `glm::scale(sx, sy, sz);`
 - Echelle sx, sy, sz le long des axes Ox, Oy, Oz