Moteur Physique pour les jeux vidéo

5ème GamiX

Chapitre 3 : Le lancer de rayons

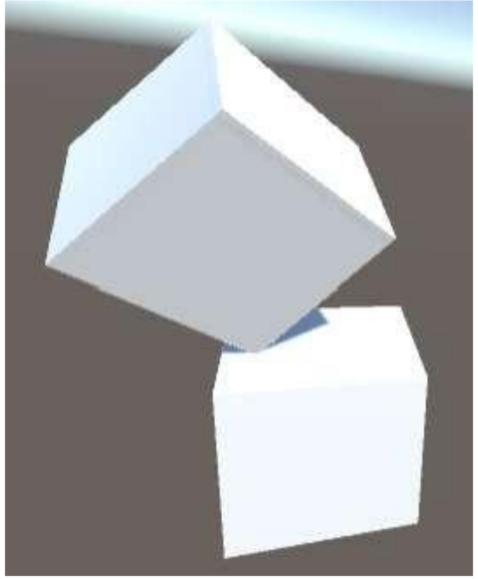
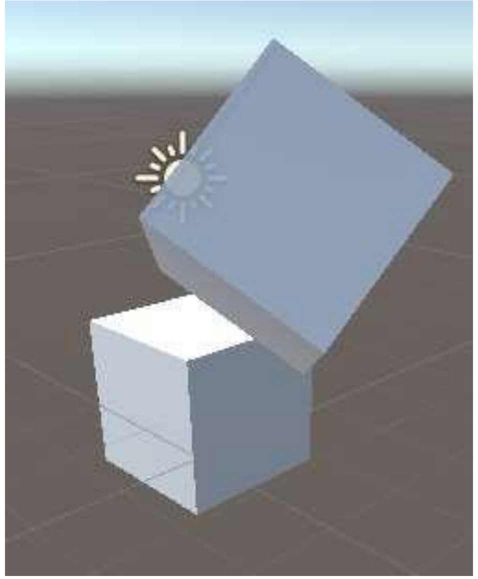
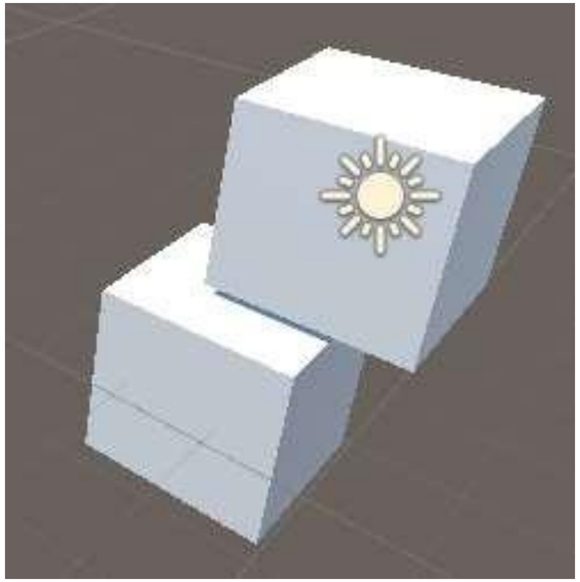
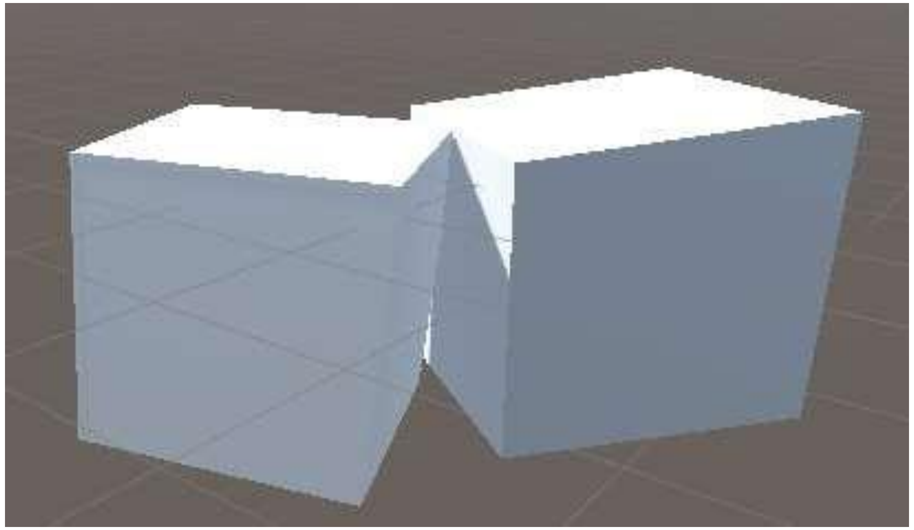
2025 – 2026

AHMED AMMAR

Objectif

Se doter d’outils mathématiques qui permettrons d’aborder la détection des collisions.

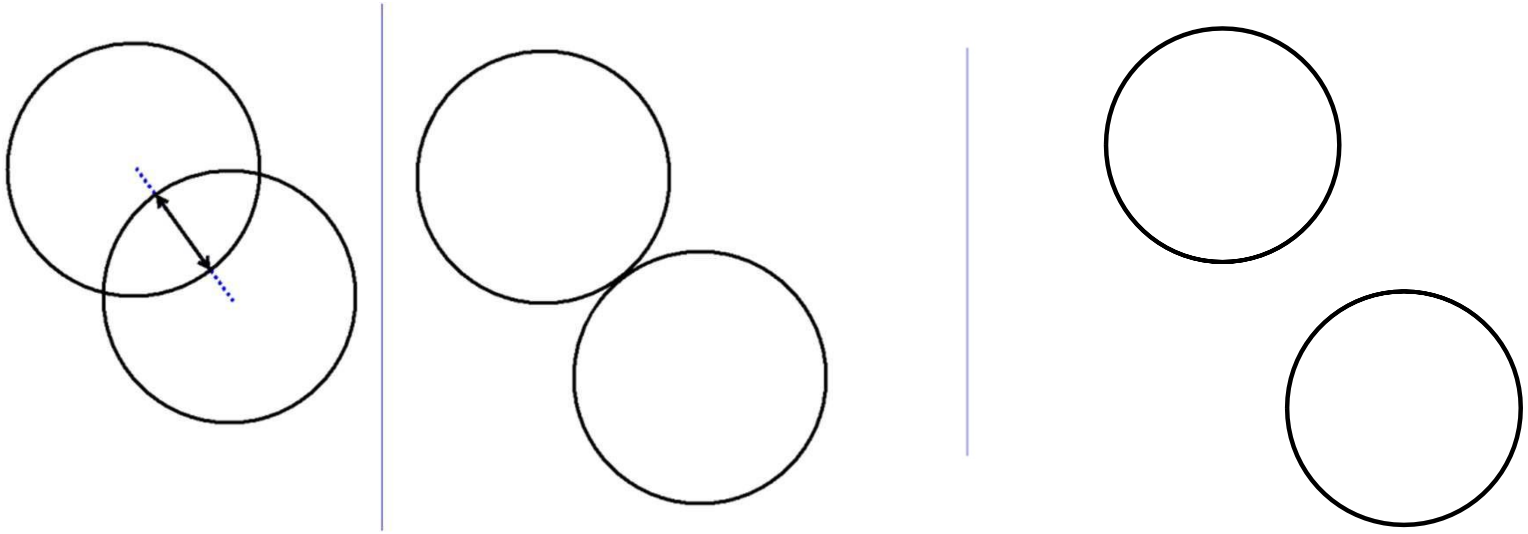
Objectif

Compétences attendues

- Savoir décrire une droite, demi-droite, segment et rayon grâce à une équation mathématique.
- Savoir décrire un plan grâce à une équation mathématique.
- Être capable de déterminer l’intersection entre un rayon et un plan.
- Être capable de déterminer l’intersection entre un rayon et une primitive (Sphère, cylindre).
- Être capable de déterminer l’intersection entre un rayon et un polygone.

Comment détecter les collisions ?

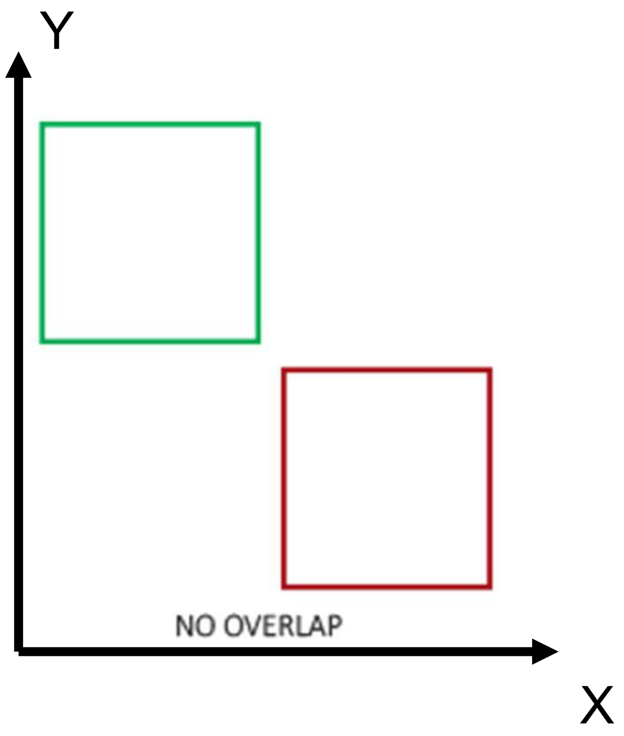


INTERDIT pour un Rigid Body

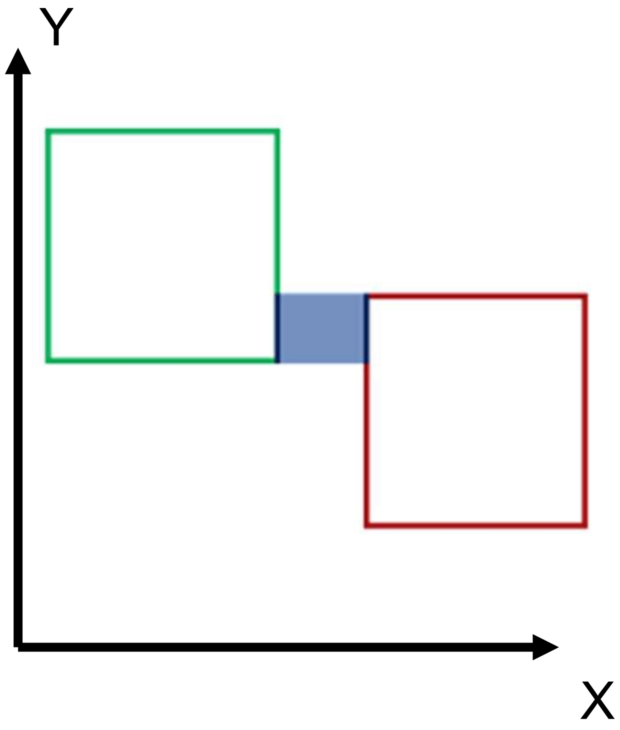
Collision

Pas de Collision

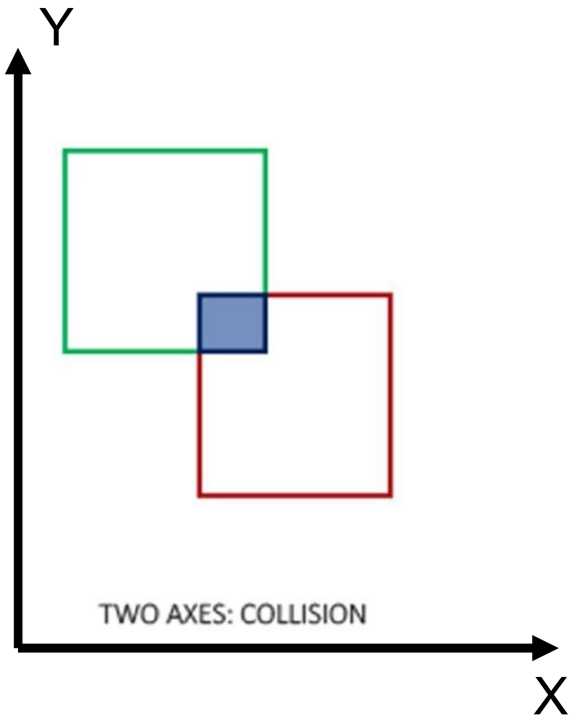
Comment détecter les collisions ?



Pas de Collision

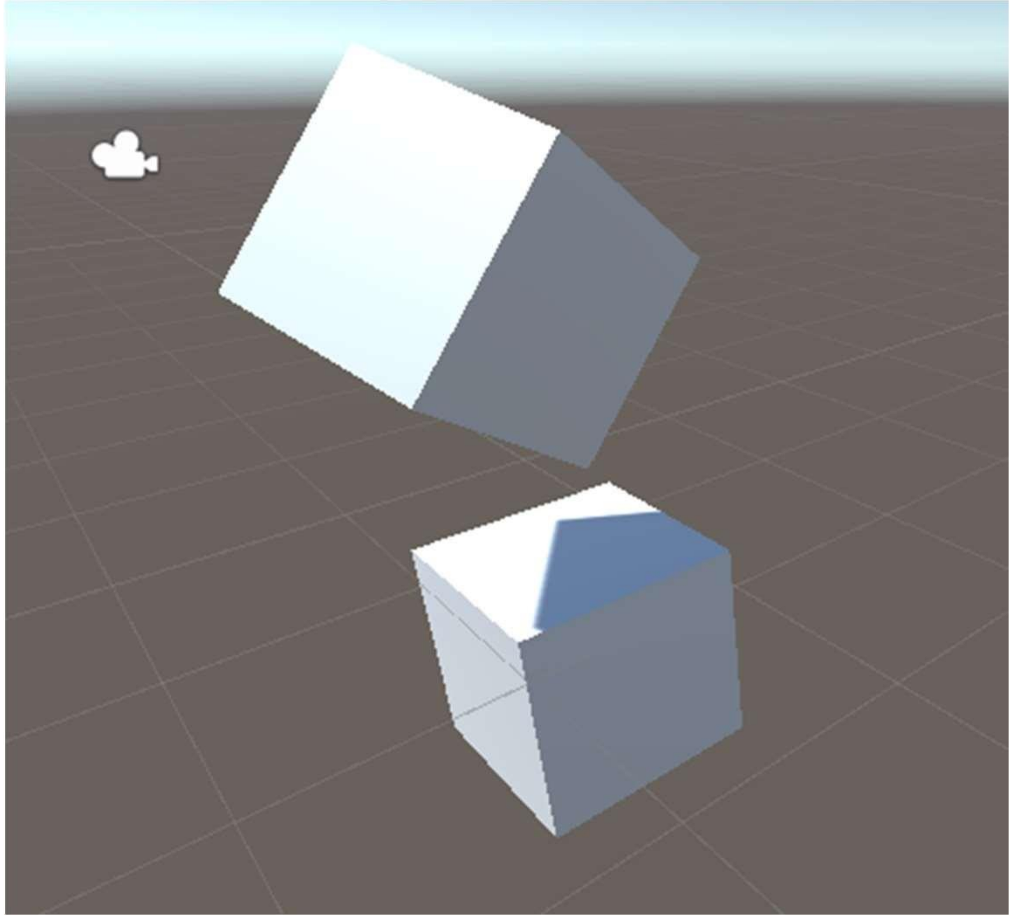


Pas de Collision



Collision

Comment détecter les collisions ?



Solides indéformables:

Quels contacts possibles lors de la collision ?

- Point-plan
- Arrête-Plan
- Arrête - Arrête

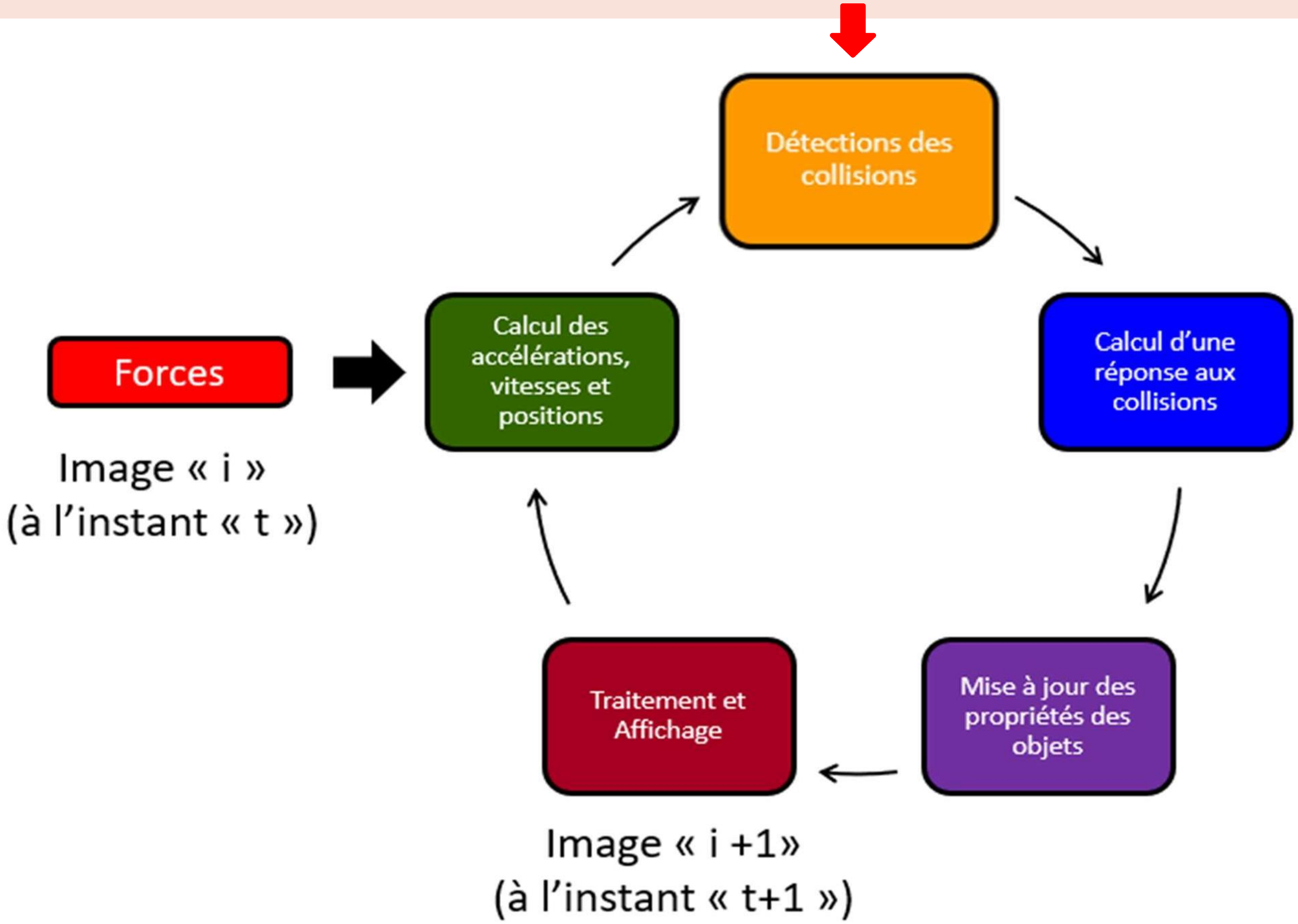
Comment les détecter ?

Avant de l'afficher à l'utilisateur

Comment les traiter ?

Eviter l'interpenetration

Comment détecter les collisions ?



Droite et segment

On définit la droite passant par deux points P_1 et P_2 de l'espace par l'équation :

$$\boxed{P(t) = (1 - t).P_1 + t.P_2}$$

Où le paramètre « t » prend TOUTES les valeurs appartenant à R

Droite et segment

On définit le segment $[P_1, P_2]$ de l'espace par l'équation :

$$\boxed{P(t) = (1-t).P_1 + t.P_2}$$

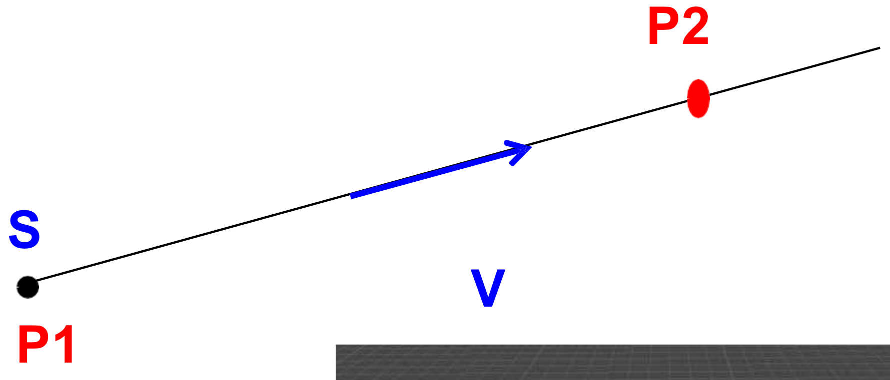
Où $t \in [0,1]$

Demi-droite (Rayon)

Une demi-droite (un rayon) est définie par un point de départ **S** d’un cotes et continuant à l’infini de l’autre cotes dans un direction **V**.

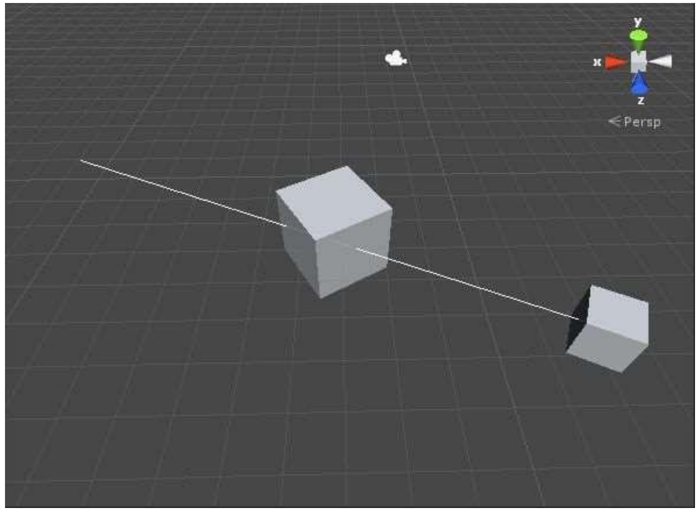
$$\boxed{P(t) = S + t.V}$$

$$t \geq 0$$

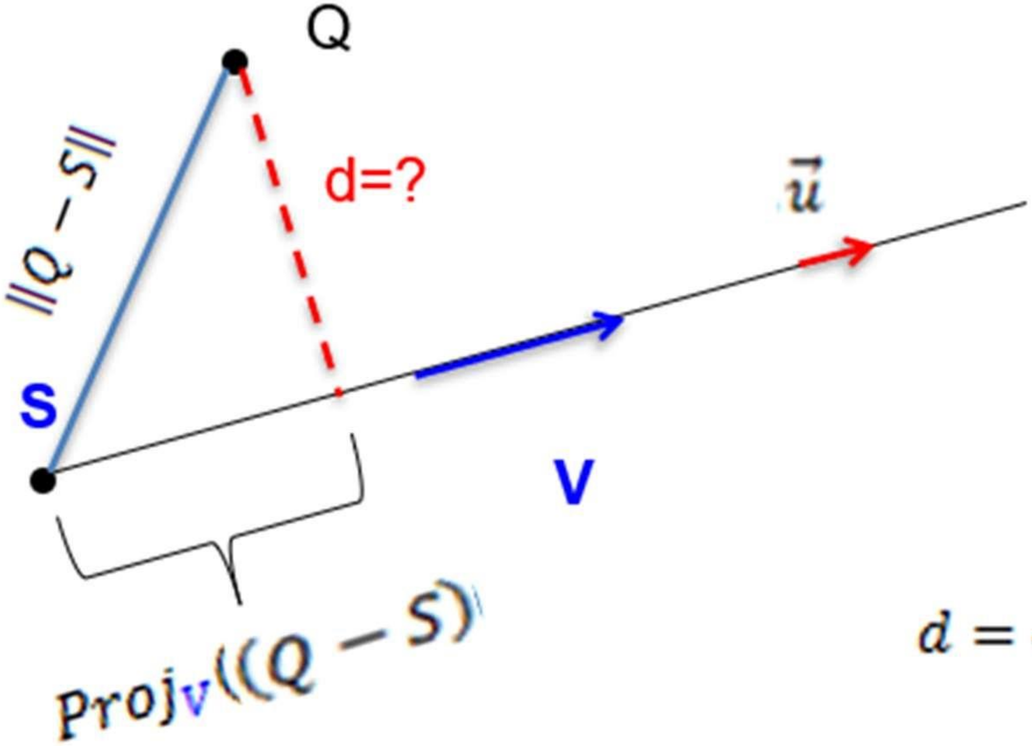


Remarque :

Avec $\begin{cases} P_1 = S \\ P_2 - P_1 = V \end{cases}$



Distance point-droite

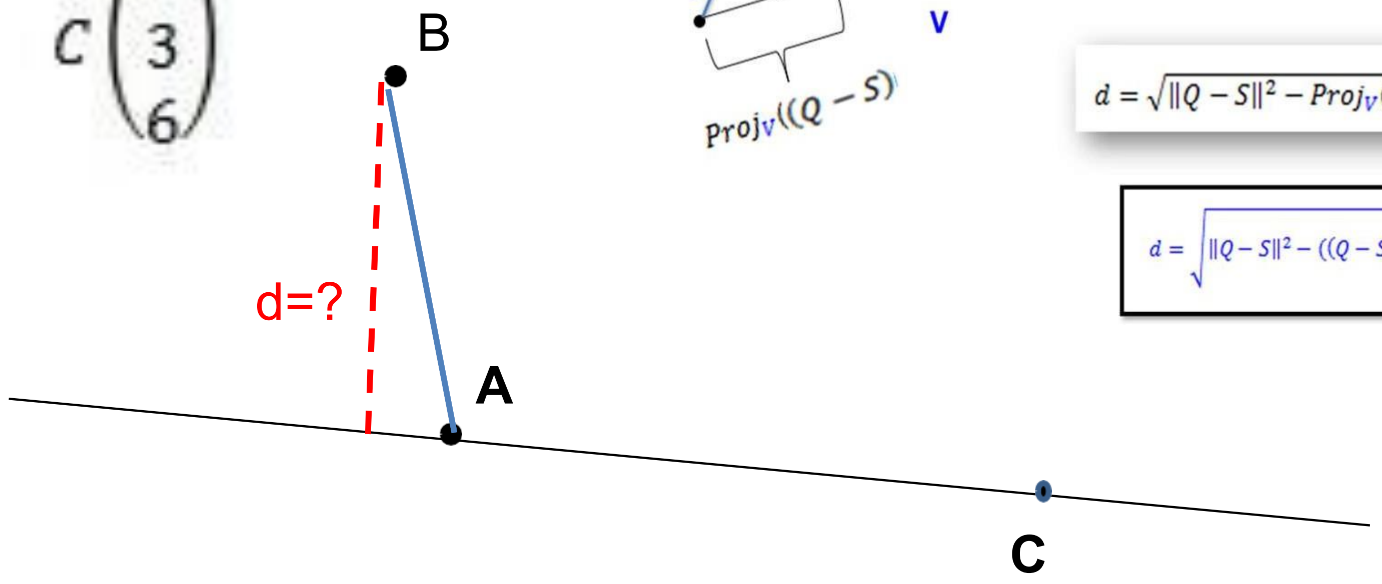


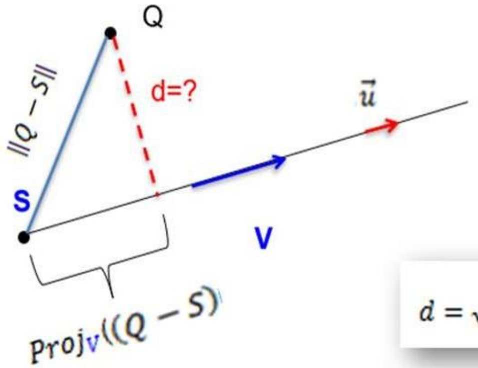
$$d = \sqrt{||Q-S||^2 - Proj_V((Q-S))^2}$$

$$\boxed{d = \sqrt{||Q-S||^2 - ((Q-S).\frac{V}{||V||})^2}}$$

Distance point-droite : Exercice

$$A\begin{pmatrix}1\\4\\-5\end{pmatrix} \quad B\begin{pmatrix}2\\-3\\-8\end{pmatrix} \quad C\begin{pmatrix}2\\3\\6\end{pmatrix}$$





$$d = \sqrt{\|Q-S\|^2 - Proj_V((Q-S))^2}$$

$$\boxed{d = \sqrt{\|Q-S\|^2 - \left((Q-S).\frac{V}{\|V\|}\right)^2}}$$

Distance point-droite : Exercice

Solution

Données

- $A(1, 4, -5)$
- $B(2, -3, 8)$
- $C(2, 3, 6)$

Étape 1 : Calcul des vecteurs

- Vecteur $\overrightarrow{BA} = B - A$

$$\overrightarrow{BA} = (2-1, -3-4, 8+5) = (1, -7, 13)$$

- Vecteur $\overrightarrow{V} = C - A$

$$\overrightarrow{V} = (2-1, 3-4, 6+5) = (1, -1, 11)$$

Étape 2 : Projection de $\overrightarrow{BA}$ sur $\overrightarrow{V}$

La projection de $\overrightarrow{BA}$ sur $\overrightarrow{V}$ est donnée par la formule :

$$\mathrm{Proj}_{\overrightarrow{V}}(\overrightarrow{BA}) = \frac{\overrightarrow{BA} \cdot \overrightarrow{V}}{\overrightarrow{V} \cdot \overrightarrow{V}} \overrightarrow{V}$$

- Produit scalaire $\overrightarrow{BA} \cdot \overrightarrow{V}$

$$\overrightarrow{BA} \cdot \overrightarrow{V} = 1 \cdot 1 + (-7) \cdot (-1) + 13 \cdot 11 = 1 + 7 + 143 = 151$$

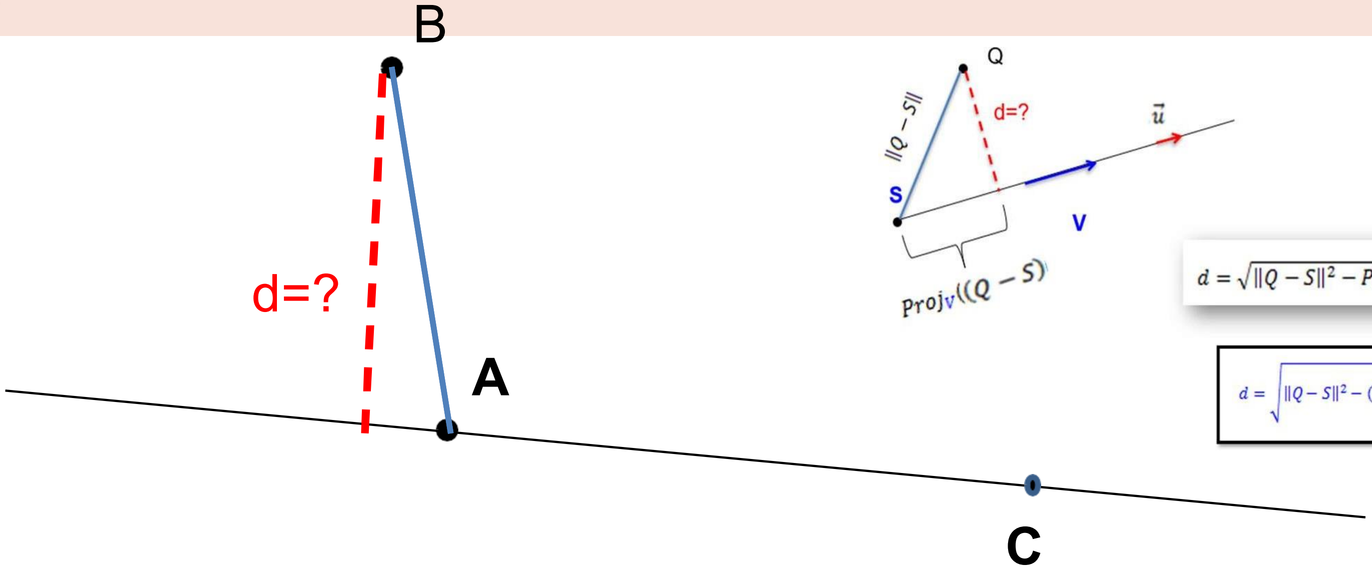
- Produit scalaire $\overrightarrow{V} \cdot \overrightarrow{V}$

$$\overrightarrow{V} \cdot \overrightarrow{V} = 1^2 + (-1)^2 + 11^2 = 1 + 1 + 121 = 123$$

- Projection :

$$\mathrm{Proj}_{\overrightarrow{V}}(\overrightarrow{BA}) = \frac{151}{123} \overrightarrow{V} = \frac{151}{123}(1, -1, 11)$$

$$\mathrm{Proj}_{\overrightarrow{V}}(\overrightarrow{BA}) \approx (1.23, -1.23, 13.51)$$



$$d = \sqrt{\|Q-S\|^2 - Proj_V((Q-S))^2}$$

$$\boxed{d = \sqrt{\|Q-S\|^2 - ((Q-S).\frac{V}{\|V\|})^2}}$$

Étape 3 : Calcul de la distance

La distance d entre le point B et la droite passant par A et C est donnée par :

$$d = \sqrt{||\overrightarrow{BA}||^2 - ||\mathrm{Proj}_{\overrightarrow{V}}(\overrightarrow{BA})||^2}$$

- Norme de $\overrightarrow{BA}$:

$$||\overrightarrow{BA}|| = \sqrt{1^2 + (-7)^2 + 13^2} = \sqrt{1 + 49 + 169} = \sqrt{219}$$

- Norme de la projection $\mathrm{Proj}_{\overrightarrow{V}}(\overrightarrow{BA})$:

$$||\mathrm{Proj}_{\overrightarrow{V}}(\overrightarrow{BA})||^2 = (1.23)^2 + (-1.23)^2 + (13.51)^2 \approx 1.51 + 1.51 + 182.52 = 185.54$$

- Calcul de la distance :

$$d = \sqrt{219 - 185.54} = \sqrt{33.46} \approx 5.78$$

Distance point-droite : Exercice

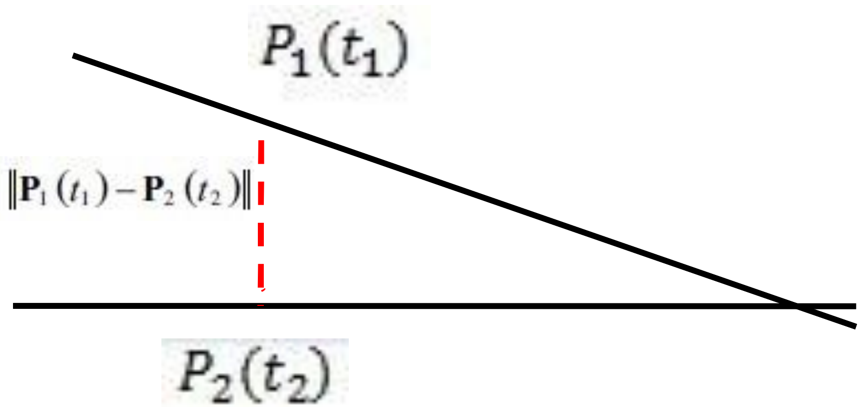
Solution (C#)

```csharp
using UnityEngine;

public class ProjectionDistance : MonoBehaviour
{
    // Points A, B, et C
    private Vector3 pointA = new Vector3(1, 4, -5);
    private Vector3 pointB = new Vector3(2, -3, 8);
    private Vector3 pointC = new Vector3(2, 3, 6);

    void Start()
    {
        // Calcul du vecteur BA et du vecteur V (AC)
        Vector3 BA = pointB - pointA;
        Vector3 V = pointC - pointA;

        // Projection de BA sur V
        Vector3 projectionBAonV = ProjectVector(BA, V);

        // Distance entre B et la droite passant par A et C
        float distance = CalculateDistance(BA, projectionBAonV);

        // Afficher les résultats dans la console
        Debug.Log("Projection de BA sur V: " + projectionBAonV);
        Debug.Log("Distance entre B et la droite passant par A et C: " + distance);
    }
```

```csharp
    // Méthode pour projeter un vecteur sur un autre
    Vector3 ProjectVector(Vector3 u, Vector3 v)
    {
        float dotProductUV = Vector3.Dot(u, v); // Produit scalaire u · v
        float dotProductVV = Vector3.Dot(v, v); // Produit scalaire v · v
        return (dotProductUV / dotProductVV) * v; // Projection de u sur v
    }

    // Méthode pour calculer la distance
    float CalculateDistance(Vector3 BA, Vector3 projection)
    {
        // Norme de BA
        float normBA = BA.magnitude;

        // Norme de la projection
        float normProjection = projection.magnitude;

        // Distance entre le point et la droite
        return Mathf.Sqrt(normBA * normBA - normProjection * normProjection);
    }
}
```

Distance entre 2 droites

On se propose de chercher la distance minimale entre ces droites



Droite 1 ⇨ $P_1(t_1) = S_1 + t_1.V_1$

Droite 2 ⇨ $P_2(t_2) = S_2 + t_2.V_2$

Quelles sont les valeurs de t1 et de t2 qui minimisent $\|P_1(t_1) - P_2(t_2)\|$?

$f(t_1, t_2) = \|P_1(t_1) - P_2(t_2)\|^2$ Ensuite : $\frac{\partial f}{\partial t_1} = 0$ et $\frac{\partial f}{\partial t_2} = 0$ Pour minimiser

Distance entre 2 droites

Développer l'expression de la distance :

$$f(t_1, t_2) = \|(S_1 - S_2) + t_1 V_1 - t_2 V_2\|^2$$

En développant le produit scalaire :

$$f(t_1, t_2) = (S_1 - S_2) \cdot (S_1 - S_2) + 2t_1(S_1 - S_2) \cdot V_1 - 2t_2(S_1 - S_2) \cdot V_2 + t_1^2 V_1 \cdot V_1 + t_2^2 V_2 \cdot V_2 - 2t_1 t_2 V_1 \cdot V_2$$

Nous dérivons ensuite par rapport à t_1 et t_2, et posons ces dérivées égales à zéro pour obtenir deux équations :

1. **Dérivée par rapport à t_1 :**

$$\frac{\partial f}{\partial t_1} = 2(S_1 - S_2) \cdot V_1 + 2t_1 V_1 \cdot V_1 - 2t_2 V_1 \cdot V_2 = 0$$

2. **Dérivée par rapport à t_2 :**

$$\frac{\partial f}{\partial t_2} = -2(S_1 - S_2) \cdot V_2 - 2t_1 V_1 \cdot V_2 + 2t_2 V_2 \cdot V_2 = 0$$

Nous pouvons réécrire ces équations sous forme de système linéaire :

$$\begin{aligned} t_1 V_1 \cdot V_1 - t_2 V_1 \cdot V_2 &= -(S_1 - S_2) \cdot V_1 \\ -t_1 V_1 \cdot V_2 + t_2 V_2 \cdot V_2 &= (S_1 - S_2) \cdot V_2 \end{aligned}$$

Formulation matricielle

Nous pouvons maintenant écrire ce système sous forme de matrice :

$$\begin{pmatrix} V_1 \cdot V_1 & -V_1 \cdot V_2 \\ -V_1 \cdot V_2 & V_2 \cdot V_2 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} = \begin{pmatrix} -(S_1 - S_2) \cdot V_1 \\ (S_1 - S_2) \cdot V_2 \end{pmatrix}$$

Solution par inversion de matrice

Pour résoudre ce système, nous devons inverser la matrice :

$$A = \begin{pmatrix} V_1 \cdot V_1 & -V_1 \cdot V_2 \\ -V_1 \cdot V_2 & V_2 \cdot V_2 \end{pmatrix}$$

Et résoudre pour :

$$\mathbf{t} = A^{-1}\mathbf{b}$$

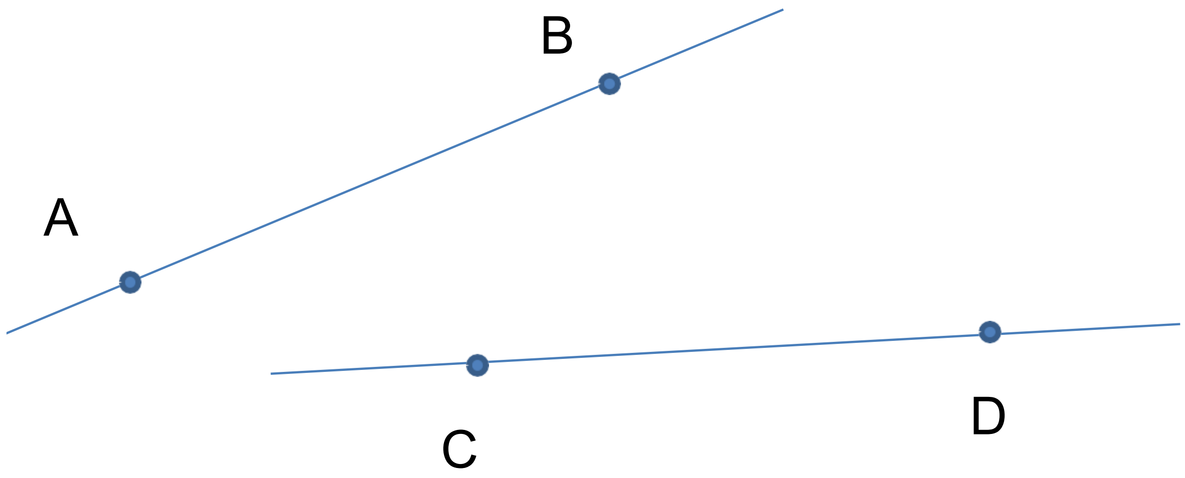
où

$$\mathbf{b} = \begin{pmatrix} -(S_1 - S_2) \cdot V_1 \\ (S_1 - S_2) \cdot V_2 \end{pmatrix}$$

On réinjecte ensuite ces valeurs de t1 et t2 dans l'expression de $f(t_1, t_2)$

Exercice

$A\begin{pmatrix}1\\0\\4\end{pmatrix}; B\begin{pmatrix}-1\\1\\3\end{pmatrix}; C\begin{pmatrix}0\\5\\4\end{pmatrix}$ et $D\begin{pmatrix}1\\2\\2\end{pmatrix}$



chercher la distance minimale entre les droites (AB) et (CD)?

$$\begin{pmatrix}t_1\\t_2\end{pmatrix} = \begin{pmatrix}V_1^2 & -V_1V_2\\ V_1V_2 & -V_2^2\end{pmatrix}^{-1}\begin{pmatrix}(S_2-S_1).V_1\\(S_2-S_1).V_2\end{pmatrix}.$$

Exercice (Solution)

L'objectif est de trouver la distance minimale entre les droites (AB) et (CD), où les droites sont définies par les points $A(1,0,4)$, $B(-1,1,3)$, $C(0,5,4)$, et $D(1,2,2)$.

Représentation des points et des vecteurs :

- Soit $S_1 = A(1,0,4)$ et $S_2 = C(0,5,4)$.
- Les vecteurs directionnels sont :

$$V_1 = B - A = (-1-1, 1-0, 3-4) = (-2,1,-1)$$

$$V_2 = D - C = (1-0, 2-5, 2-4) = (1,-3,-2)$$

Calcul de $S_1 - S_2$:

$$S_1 - S_2 = (1-0, 0-5, 4-4) = (1,-5,0)$$

Calcul des produits scalaires :

- **Pour V_1 :**

$$V_1 \cdot V_1 = (-2)^2 + 1^2 + (-1)^2 = 4+1+1 = 6$$

- **Pour V_2 :**

$$V_2 \cdot V_2 = 1^2 + (-3)^2 + (-2)^2 = 1+9+4 = 14$$

- **Pour $V_1 \cdot V_2$:**

$$V_1 \cdot V_2 = (-2)(1) + (1)(-3) + (-1)(-2) = -2-3+2 = -3$$

- **Pour $(S_1 - S_2)$:**

$$S_1 - S_2 = (1,-5,0) \Rightarrow (S_1 - S_2)\cdot(S_1 - S_2) = 1^2 + (-5)^2 + 0^2 = 1+25 = 26$$

$$(S_1 - S_2)\cdot V_1 = (1)(-2) + (-5)(1) + (0)(-1) = -2-5+0 = -7$$

$$(S_1 - S_2)\cdot V_2 = (1)(1) + (-5)(-3) + (0)(-2) = 1+15+0 = 16$$

Formulation matricielle : En utilisant ces valeurs, nous pouvons écrire notre système d'équations :

$$\begin{pmatrix} V_1 \cdot V_1 & -V_1 \cdot V_2 \\ -V_1 \cdot V_2 & V_2 \cdot V_2 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} = \begin{pmatrix} -(S_1 - S_2)\cdot V_1 \\ (S_1 - S_2)\cdot V_2 \end{pmatrix}$$

En remplaçant avec nos valeurs :

$$\begin{pmatrix} 6 & 3 \\ 3 & 14 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} = \begin{pmatrix} 7 \\ 16 \end{pmatrix}$$

Inversion de la matrice : Pour résoudre ce système, calculons l'inverse de la matrice A:

$$A = \begin{pmatrix} 6 & 3 \\ 3 & 14 \end{pmatrix}$$

La déterminant est :

$$\det(A) = (6)(14) - (3)(3) = 84 - 9 = 75$$

L'inverse est donc :

$$A^{-1} = \frac{1}{75} \begin{pmatrix} 14 & -3 \\ -3 & 6 \end{pmatrix}$$

Exercice (Solution)

L'objectif est de trouver la distance minimale entre les droites (AB) et (CD), où les droites sont définies par les points $A(1,0,4)$, $B(-1,1,3)$, $C(0,5,4)$, et $D(1,2,2)$.

Résolution du système : Nous calculons :

$$\mathbf{t} = A^{-1}\mathbf{b} = \frac{1}{75}\begin{pmatrix} 14 & -3 \\ -3 & 6 \end{pmatrix}\begin{pmatrix} 7 \\ 16 \end{pmatrix}$$

Calculons le produit :

$$\begin{pmatrix} (14 \cdot 7) + (-3 \cdot 16) \\ (-3 \cdot 7) + (6 \cdot 16) \end{pmatrix} = \begin{pmatrix} 98 - 48 \\ -21 + 96 \end{pmatrix} = \begin{pmatrix} 50 \\ 75 \end{pmatrix}$$

Donc :

$$\mathbf{t} = \frac{1}{75}\begin{pmatrix} 50 \\ 75 \end{pmatrix} = \begin{pmatrix} \frac{2}{3} \\ 1 \end{pmatrix}$$

Distance Minimale

Les valeurs minimales de t_1 et t_2 qui minimisent la distance entre les deux droites sont :

$$t_1 = \frac{2}{3}, \quad t_2 = 1$$

Pour trouver la distance minimale, nous remplaçons ces valeurs dans l'expression de la distance $d(t_1, t_2)$:

1. **Calcul des points :**

$$P_1\left(\frac{2}{3}\right) = A + \frac{2}{3}V_1$$

$$P_1\left(\frac{2}{3}\right) = \begin{pmatrix} 1 \\ 0 \\ 4 \end{pmatrix} + \frac{2}{3}\begin{pmatrix} -2 \\ 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 - \frac{4}{3} \\ 0 + \frac{2}{3} \\ 4 - \frac{2}{3} \end{pmatrix} = \begin{pmatrix} -\frac{1}{3} \\ \frac{2}{3} \\ \frac{10}{3} \end{pmatrix}$$

$$P_2(1) = C + 1 \cdot V_2 = \begin{pmatrix} 0 \\ 5 \\ 4 \end{pmatrix} + \begin{pmatrix} 1 \\ -3 \\ -2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}$$

2. **Calculer la distance :**

$$d = \|P_1(t_1) - P_2(t_2)\| = \left\| \begin{pmatrix} -\frac{1}{3} - 1 \\ \frac{2}{3} - 2 \\ \frac{10}{3} - 2 \end{pmatrix} \right\| = \left\| \begin{pmatrix} -\frac{4}{3} \\ -\frac{4}{3} \\ \frac{4}{3} \end{pmatrix} \right\|$$

Calculons la norme :

$$d = \sqrt{\left(-\frac{4}{3}\right)^2 + \left(-\frac{4}{3}\right)^2 + \left(\frac{4}{3}\right)^2} = \sqrt{\frac{16}{9} + \frac{16}{9} + \frac{16}{9}} = \sqrt{\frac{48}{9}} = \frac{4\sqrt{3}}{3}$$

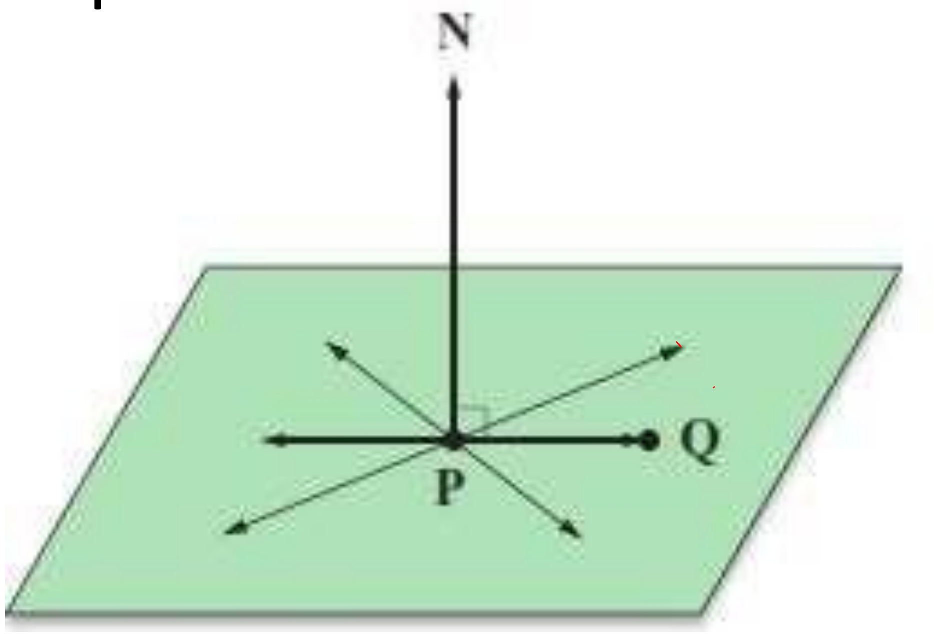
```
using UnityEngine;

public class MinDistanceBetweenLines : MonoBehaviour
{
    // Points A, B, C, D définissant les droites (AB) et (CD)
    public Vector3 A = new Vector3(1, 0, 4);
    public Vector3 B = new Vector3(-1, 1, 3);
    public Vector3 C = new Vector3(0, 5, 4);
    public Vector3 D = new Vector3(1, 2, 2);

    void Start()
    {
        // Vecteurs directionnels des droites
        Vector3 V1 = B - A; // Vecteur AB
        Vector3 V2 = D - C; // Vecteur CD

        // Vecteur entre les deux points d'origine
        Vector3 S1_minus_S2 = A - C;

        // Calcul des produits scalaires
        float V1_dot_V1 = Vector3.Dot(V1, V1);
        float V2_dot_V2 = Vector3.Dot(V2, V2);
        float V1_dot_V2 = Vector3.Dot(V1, V2);
        float S1_minus_S2_dot_V1 = Vector3.Dot(S1_minus_S2, V1);
        float S1_minus_S2_dot_V2 = Vector3.Dot(S1_minus_S2, V2);

        // Construction du système d'équations
        float[,] matrix = {
            { V1_dot_V1, -V1_dot_V2 },
            { -V1_dot_V2, V2_dot_V2 }
        };

        float[] constants = {
            -S1_minus_S2_dot_V1,
            S1_minus_S2_dot_V2
        };
```

C#

```
        // Résolution du système (utilisation d'une inversion manuelle de matrice)
        float det = matrix[0, 0] * matrix[1, 1] - matrix[0, 1] * matrix[1, 0];

        if (Mathf.Abs(det) < 1e-6)
        {
            Debug.LogError("Les droites sont parallèles ou très proches.");
            return;
        }

        // Calcul de l'inverse de la matrice
        float invDet = 1.0f / det;
        float[,] inverseMatrix = {
            { matrix[1, 1] * invDet, -matrix[0, 1] * invDet },
            { -matrix[1, 0] * invDet, matrix[0, 0] * invDet }
        };

        // Résolution pour t1 et t2
        float t1 = inverseMatrix[0, 0] * constants[0] + inverseMatrix[0, 1] * constants
        float t2 = inverseMatrix[1, 0] * constants[0] + inverseMatrix[1, 1] * constants

        // Calcul des points sur les droites correspondants aux valeurs t1 et t2
        Vector3 P1 = A + t1 * V1;
        Vector3 P2 = C + t2 * V2;

        // Calcul de la distance minimale
        float distance = Vector3.Distance(P1, P2);

        // Affichage du résultat dans la console Unity
        Debug.Log($"La distance minimale entre les droites est : {distance}");
    }
}
```

Les plans

Comment définir le plan ?



Soit un point **P** de l'espace et un vecteur normal **n**.

➢ Le plan passant par **P** et normal à **n** est défini comme l'ensemble des points **Q** tel que :

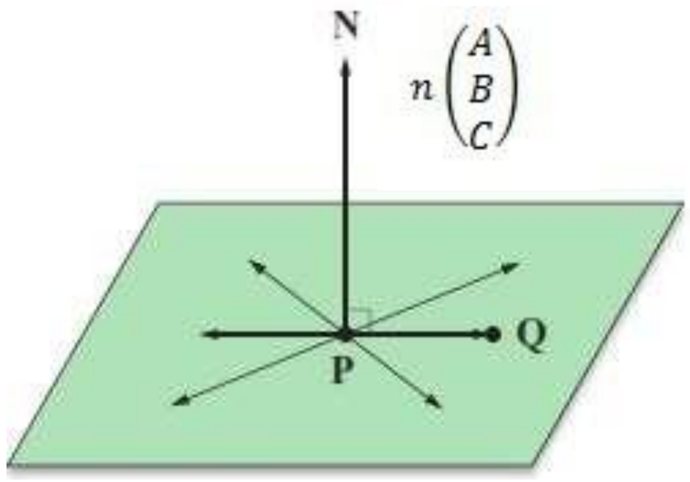
$$\mathbf{n}.(\mathbf{Q}-\mathbf{P})=0$$

Les plans

$\mathbf{n}.(\mathbf{Q}-\mathbf{P})=0 \Rightarrow n.Q-n.P=0$

vecteur

vertex



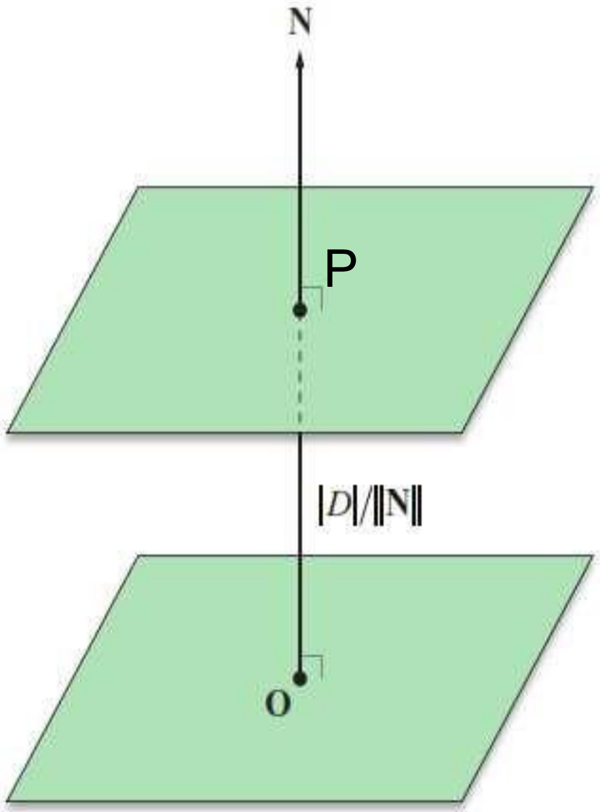
- L'équation du plan est donnée par :

$$\boxed{A.x + B.y + C.z + D = 0}$$

$$D = -n.P$$

Les plans

$$\boxed{A.x + B.y + C.z + D = 0}$$



$$|D|$$

$$\Downarrow$$

Distance séparant le plan d'un autre plan qui lui est parallèle et qui passe par l'origine.

Remarque

$$\boxed{A.x + B.y + C.z + D = 0}$$

On représente souvent le plan à l'aide d'un vecteur à 4 dimensions :

$$\boxed{L = < n, D >}$$

Le vecteur normal au plan

$$D = -n.P$$

(réel)

Distance d'un point à un plan

Soit un plan défini par l'équation suivante :

$$Ax + By + Cz + D = 0$$

Avec :

- A, B, C : composantes du vecteur normal $n(A, B, C)$ du plan.
- D : distance signée du plan à l'origine.

Soit un point $P(x_1, y_1, z_1)$ dans l'espace. Nous souhaitons calculer la distance entre ce point et le plan.

Formule de la distance d'un point à un plan

La distance perpendiculaire d'un point $P(x_1, y_1, z_1)$ au plan $Ax + By + Cz + D = 0$ est donnée par la formule :

$$d = \frac{|Ax_1 + By_1 + Cz_1 + D|}{\sqrt{A^2 + B^2 + C^2}}$$

Explication :

- **Numérateur :** $|Ax_1 + By_1 + Cz_1 + D|$ est la valeur absolue du plan calculée en $P(x_1, y_1, z_1)$.
- **Dénominateur :** $\sqrt{A^2 + B^2 + C^2}$ est la norme du vecteur normal $n(A, B, C)$.

- Si $D > 0$, le plan est **au-dessus** de l'origine dans la direction du vecteur normal.
- Si $D < 0$, le plan est **en dessous** de l'origine, toujours dans la direction du vecteur normal.
- Si $D = 0$, le plan **passe par l'origine**.

```
using UnityEngine;

public class DistancePointToPlane : MonoBehaviour
{
    // Définition du plan par son vecteur normal et sa distance D
    private Vector3 normal = new Vector3(1, 2, -1); // A = 1, B = 2, C = -1
    private float D = -5.0f; // Distance du plan à l'origine

    // Le point dont on veut calculer la distance au plan
    private Vector3 point = new Vector3(3, 2, 1); // P(x1 = 3, y1 = 2, z1 = 1)

    void Start()
    {
        // Calcul de la distance
        float distance = DistanceToPoint(point, normal, D);

        // Afficher la distance dans la console
        Debug.Log("Distance du point au plan : " + distance);
    }

    // Fonction qui calcule la distance d'un point à un plan
    float DistanceToPoint(Vector3 point, Vector3 normal, float D)
    {
        // On calcule le numérateur : |A*x1 + B*y1 + C*z1 + D|
        float numerator = Mathf.Abs(Vector3.Dot(normal, point) + D);

        // On calcule le dénominateur : sqrt(A^2 + B^2 + C^2)
        float denominator = normal.magnitude;

        // On retourne la distance
        return numerator / denominator;
    }
}
```

Intersection Rayon-plan

Droite

$$\boxed{P(t) = S + t.V}$$

Plan

$$\boxed{A.x + B.y + C.z + D = 0}$$

$$\boxed{n.P + D = 0}$$

Intersection ?

Le point P doit vérifier l'équation du plan !

$$n.(S + t.V) + D = 0$$

$$t = \frac{-(D + n.S)}{n.V}$$

On injecte « t »

$$\boxed{P(t) = S + t.V}$$

Intersection

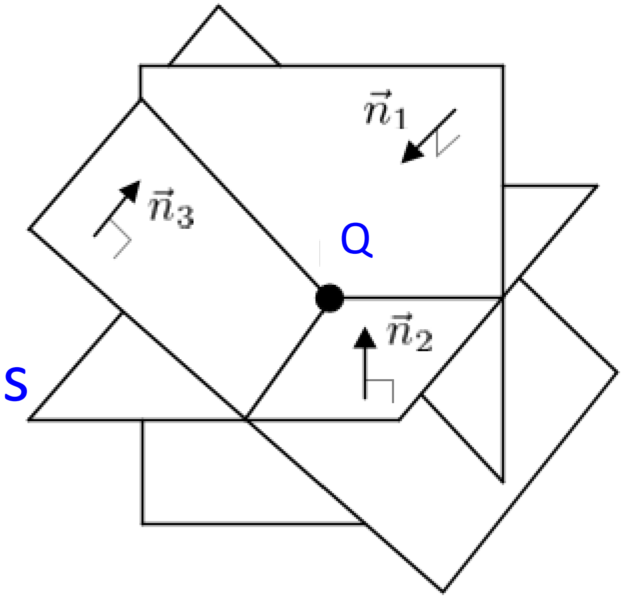
Remarque:

Si n.V=0 alors pas d'intersection

Intersection de 3 plans

Soit 3 plans

$$L_1 = < n_1, D_1 >$$
$$L_2 = < n_2, D_2 >$$
$$L_3 = < n_3, D_3 >$$



Pour trouver un point Q appartenant à l'intersection de ces 3 plans on doit résoudre le système:

$$\begin{aligned} n_1 Q + D_1 &= 0 \\ n_2 Q + D_2 &= 0 \\ n_3 Q + D_3 &= 0 \end{aligned} \Rightarrow \begin{aligned} n_1 Q &= -D_1 \\ n_2 Q &= -D_2 \\ n_3 Q &= -D_3 \end{aligned}$$

$$\begin{pmatrix} n1_x & n1_y & n1_z \\ n2_x & n2_y & n2_z \\ n3_x & n3_y & n3_z \end{pmatrix} . \begin{pmatrix} Q_x \\ Q_y \\ Q_y \end{pmatrix} = \begin{pmatrix} -D_1 \\ -D_2 \\ -D_3 \end{pmatrix} \Rightarrow \boxed{\begin{pmatrix} Q_x \\ Q_y \\ Q_y \end{pmatrix} = \begin{pmatrix} n1_x & n1_y & n1_z \\ n2_x & n2_y & n2_z \\ n3_x & n3_y & n3_z \end{pmatrix}^{-1} \begin{pmatrix} -D_1 \\ -D_2 \\ -D_3 \end{pmatrix}}$$

Remarque

- Si

$$\det\begin{pmatrix} n1_x & n1_y & n1_z \\ n2_x & n2_y & n2_z \\ n3_x & n3_y & n3_z \end{pmatrix} = 0$$

$\Downarrow$

La matrice n’est pas inversible

$\Downarrow$

Pas de point d’intersection en commun aux 3 plans

Recherche de zéros

Les racines d'un polynôme de degré « n » en « t » :

Surface plane	⇒ n=1
Surface quadratique (cylindre, sphère, etc)	⇒ n=2
Surfaces plus compliquées (torre, ect)	⇒ n=3 ou 4

Recherche de zéros

Polynômes de degré 2:

$$at^2 + bt + c = 0 \qquad \Delta = b^2 - 4ac \quad \begin{cases} \Delta = 0 \Rightarrow t = \dfrac{-b}{2a} \\ \Delta > 0 \Rightarrow t = \dfrac{-b \pm \sqrt{\Delta}}{2a} \end{cases}$$

Polynômes cubiques:

$$t^3 + at^2 + bt + c = 0$$

On peut faire disparaitre le terme quadratique en posant : $t = x - \dfrac{a}{3}$

$$\Rightarrow \quad x^3 + px + q = 0$$

$$p = b - \frac{1}{3}a^3$$

$$q = \frac{2}{27}a^3 - \frac{1}{3}ab + c$$

Recherche de zéros

$$x^3 + px + q = 0$$

Le discriminant d'un tel polynôme est donné par : $\Delta = -4p^3 - 27q^2$

$\Delta' = \dfrac{\Delta}{108}$ avec $p' = \dfrac{p}{3}$, $q' = \dfrac{q}{2}$ On pose $r = (-q' + \sqrt{-\Delta'})^{\frac{1}{3}}$, $s = (-q' - \sqrt{-\Delta'})^{\frac{1}{3}}$ Et on trouve 3 racines :

$\Delta' < 0$ $\Rightarrow$ $x_1 = r + s$ Seule racine réelle

$\Delta' = 0$ $\Rightarrow$ $r = s$ 2 racines réelles dont une double $x_1 = 2r$, $x_2 = x_3 = r(\rho + \rho^2) = -r$

$\Delta' > 0$ $\Rightarrow$ 3 racines complexes distinctes

où $\rho = -\dfrac{1}{2} + i\dfrac{\sqrt{3}}{2}$, $\rho^2 = -\dfrac{1}{2} - i\dfrac{\sqrt{3}}{2}$

Intersection de surfaces

$$\boxed{P(t) = S + t.V}$$

$$t \geq 0$$

Quelle intersection de ce rayon avec une surface ?

Remarque :

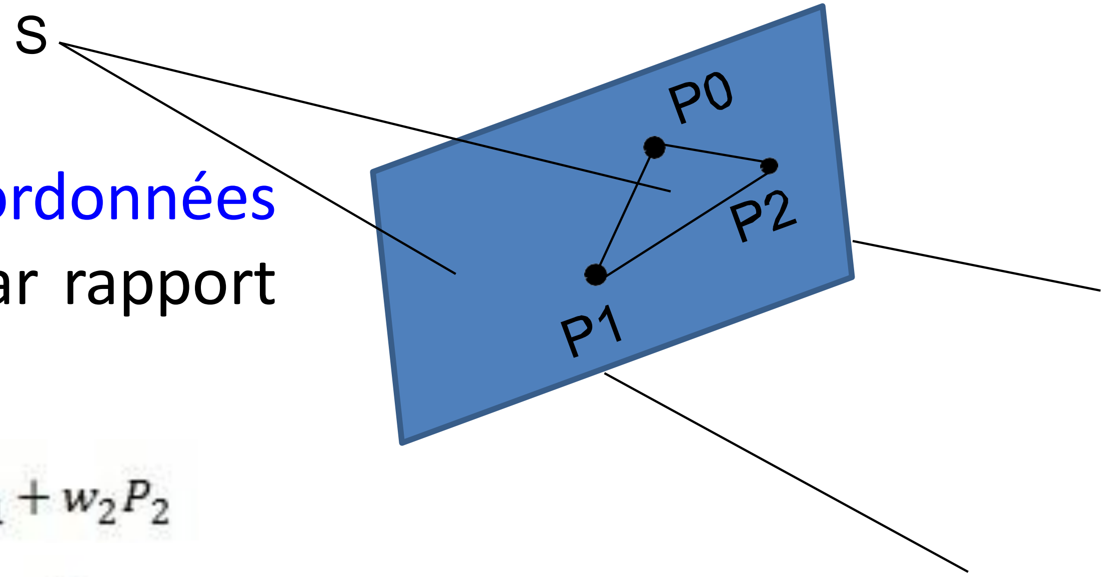
A l'exception du triangle, les intersections sont calculées dans l'espace objet

- Les intersections avec les différents objets passe donc tout d'abord par la transformation du rayon dans l'espace objet

- Une fois les coordonnées du pt d'intersection calculées elles ont ensuite, tout comme la normale au niveau de ce point, transformés dans l'espace réel.

Intersection rayon-triangle

QUESTION :



Il faut calculer les coordonnées barycentriques du point P par rapport aux points P0, P1 et P2

$w_0, w_1, w_2 = ?$ / $P = w_0 P_0 + w_1 P_1 + w_2 P_2$

et $w_0 + w_1 + w_2 = 1$

$w_0 = 1 - w_1 - w_2$ ⇒ $P = (1 - w_1 - w_2) P_0 + w_1 P_1 + w_2 P_2$

$$P - P_0 = w_1 (P_1 - P_0) + w_2 (P_2 - P_0)$$

On pose

$R = P - P_0$

$Q_1 = P_1 - P_0$

$Q_2 = P_2 - P_0$

⇒ $R = w_1 Q_1 + w_2 Q_2$

Intersection rayon-triangle

$$R = w_1 Q_1 + w_2 Q_2 \Rightarrow \begin{aligned} RQ_1 &= w_1 {Q_1}^2 + w_2 Q_1 Q_2 \\ RQ_2 &= w_1 Q_1 Q_2 + w_2 {Q_2}^2 \end{aligned}$$

$$\begin{pmatrix} {Q_1}^2 & Q_1 Q_2 \\ Q_1 Q_2 & {Q_2}^2 \end{pmatrix} \cdot \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} RQ_1 \\ RQ_2 \end{pmatrix}$$

$$\boxed{\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} {Q_1}^2 & Q_1 Q_2 \\ Q_1 Q_2 & {Q_2}^2 \end{pmatrix}^{-1} \begin{pmatrix} RQ_1 \\ RQ_2 \end{pmatrix}} \qquad \boxed{w_0 = 1 - w_1 - w_2}$$

Le point et à l’interieur du triangle ssi
w0, w1 et w2 tous >0

Travail individuel : Rendu dans 30’

Application:

- Créer une forme primitive (un cube)
- Créer une capsule
- Extraire les coordonnées des vertex du cube
- Choisir un polygone du cube
- Déterminer l’équation du plan formé par ce polygone
- Déterminer les coordonnées du point éventuel d’intersection du rayon émis du centre de la capsule dans une direction quelconque de vecteur v
- Implémenter la méthode et tester si le point d’intersection est contenu dans le polygone choisis