

Informatique graphique interactive avancée Projet

Animation(marche) d'un robot bipède avec keyframing

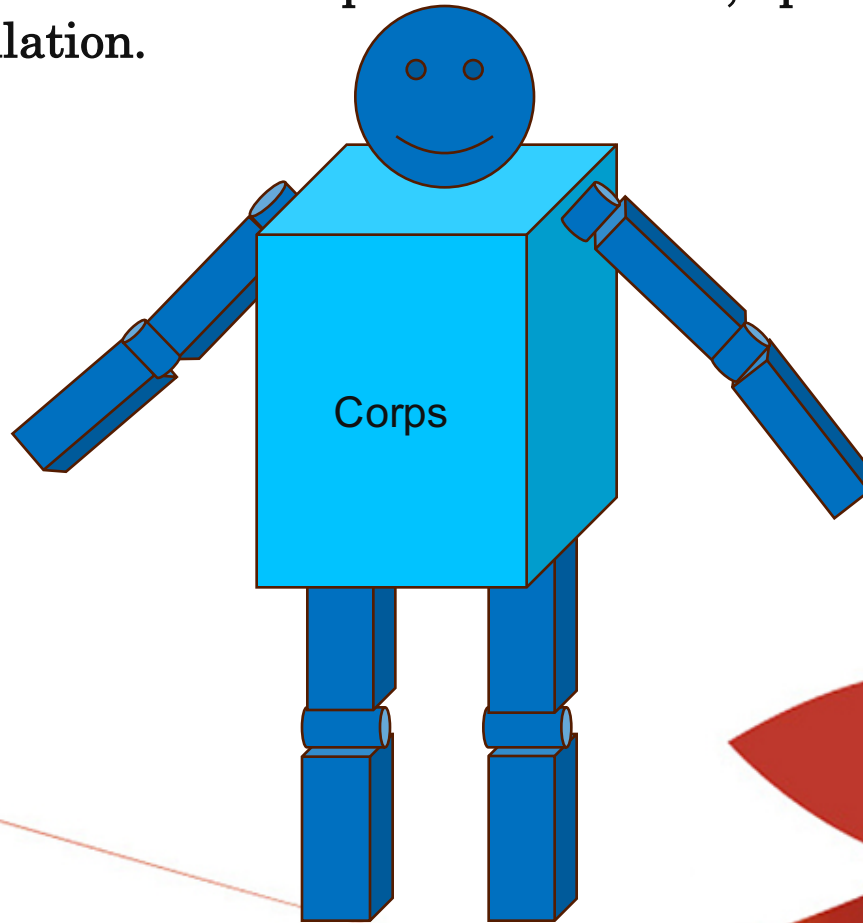
I. L'idée de départ

Par rapport à ce sujet, nous trouvons qu'il y a trois grandes parties dans ce projet :

1. Dessiner un robot entier et bones avec OPENGL.
2. Bouger le bras et jambe avec la méthode cinématique inverse et directe.
3. Réaliser l'animation avec interpolation linéaire et non-linéaire.

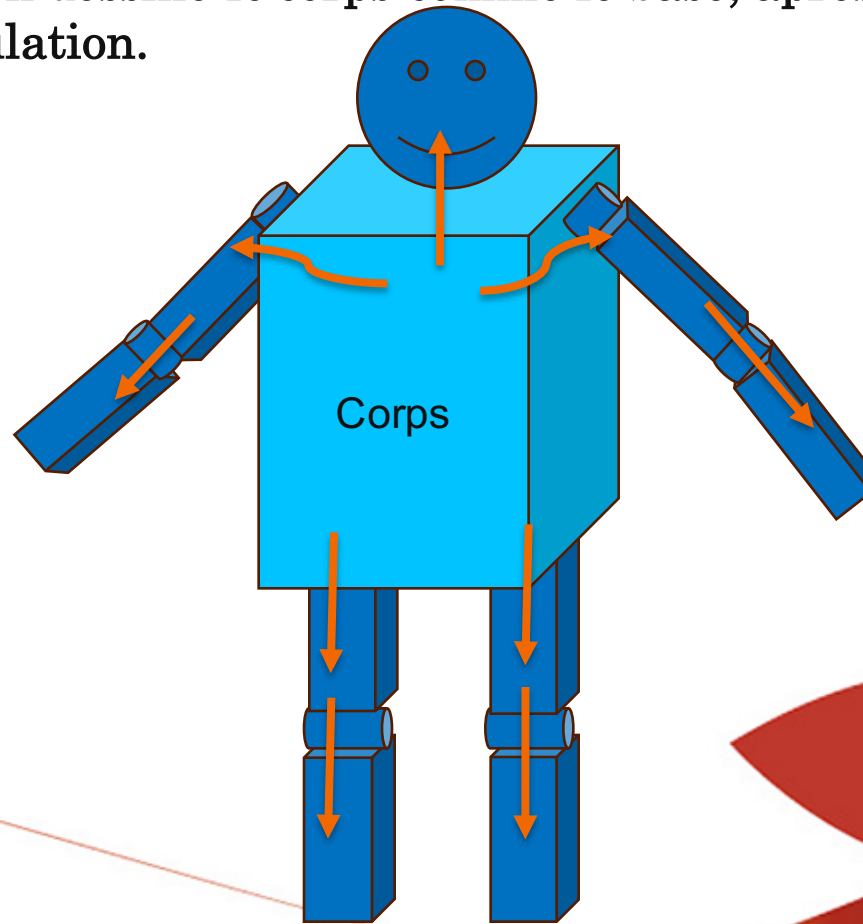
II. Dessiner un robot entier avec OpenGL.

Pour dessiner un robot entier qui peut bouger symétriquement le bras et jambe, d'abord on dessine le corps comme la base, après on dessine le bras, la jambe et l'articulation.



II. Dessiner un robot entier avec OpenGL.

Pour dessiner un robot entier qui peut bouger symétriquement le bras et jambe, d'abord on dessine le corps comme la base, après on dessine le bras, la jambe et l'articulation.



III. Système de bones

Le système de bones contient trois bones:

1. UpLeg. 2. Lowleg. 3. Effector.

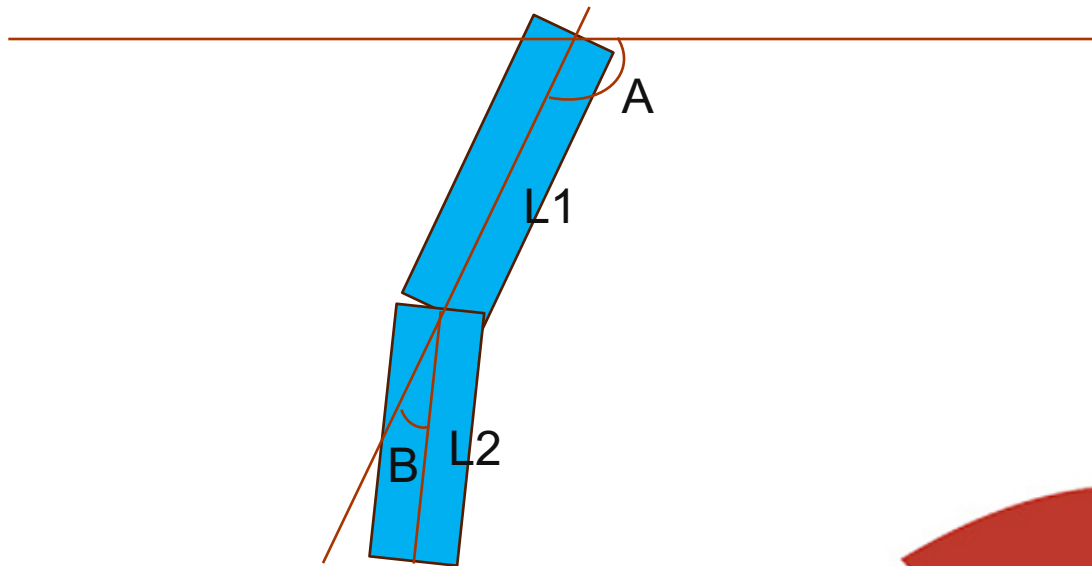
On fait l'initialisation de position et orientation de chaque bone, l'effector est la point extrémité de lowleg, il est utilisé pour faire cinématique inverse.



IV. Cinématique directe.

Cinématique directe :

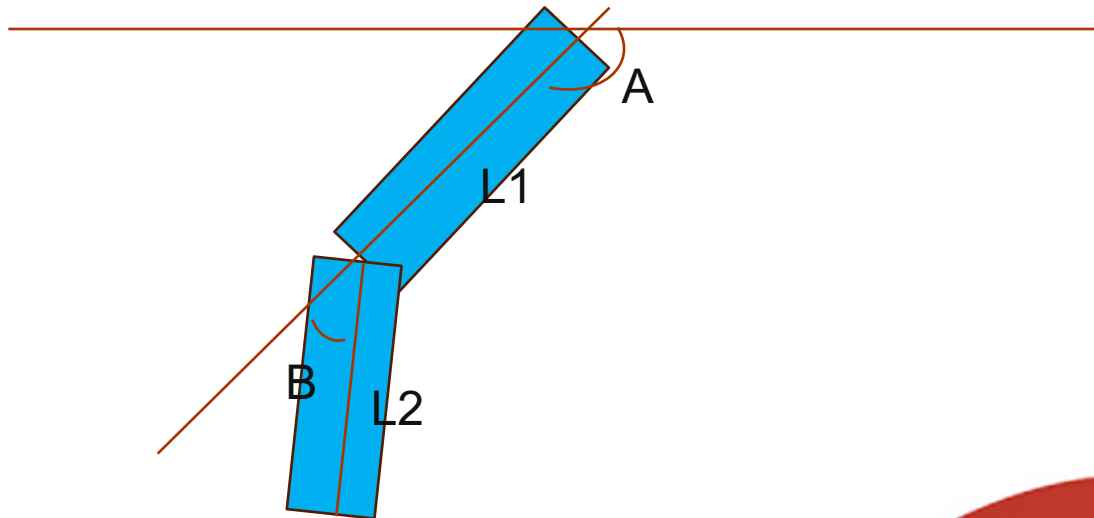
On change directement l'angle A de cuisse et l'angle B de jambe.



IV. Cinématique directe.

Cinématique directe :

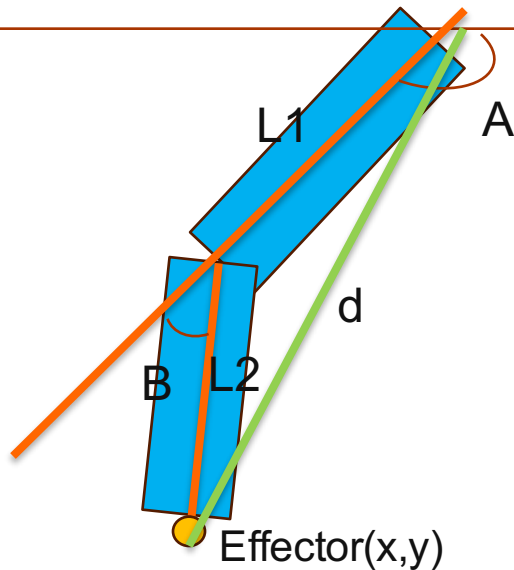
On change directement l'angle A de cuisse et l'angle B de jambe.



IV. Algorithme de cinématique inverse.

Cinématique inverse :

On utilise les formules de cinématique inverse, ensuite on utilise la coordonnée d'effector, la longueur L1 de cuisse et la longueur L2 de jambe pour calculer l'angle A et l'angle B en changeant la position d'effector, à la fin on change les attributs de rotation d'angle A et d'angle B.



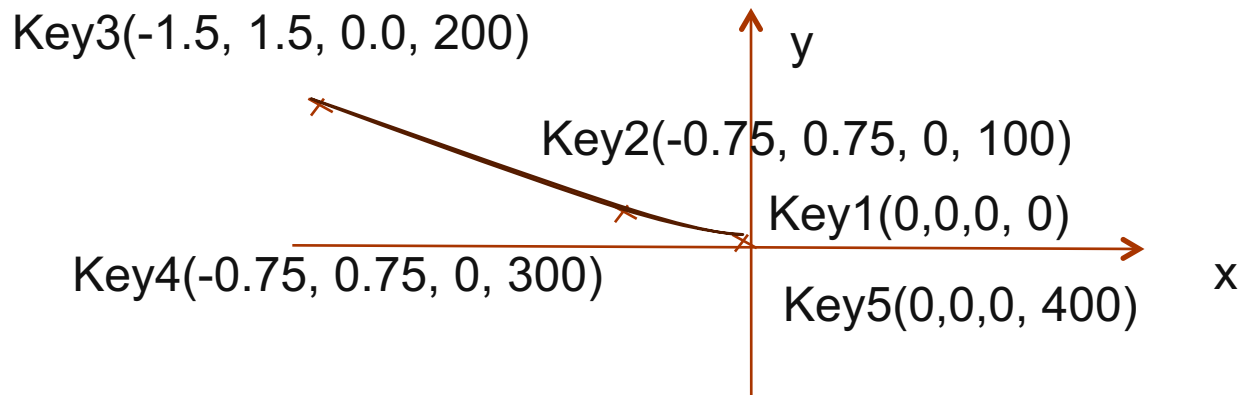
$$d = x^2 + y^2$$

$$\cos B = \frac{d^2 - L1^2 - L2^2}{2 * L1 * L2}$$

$$\angle A = \arctan\left(\frac{y}{x}\right) - \arctan\left(\frac{L2 * \sin B}{L1 + L2 * \cos B}\right)$$

Animation -- Keyframing

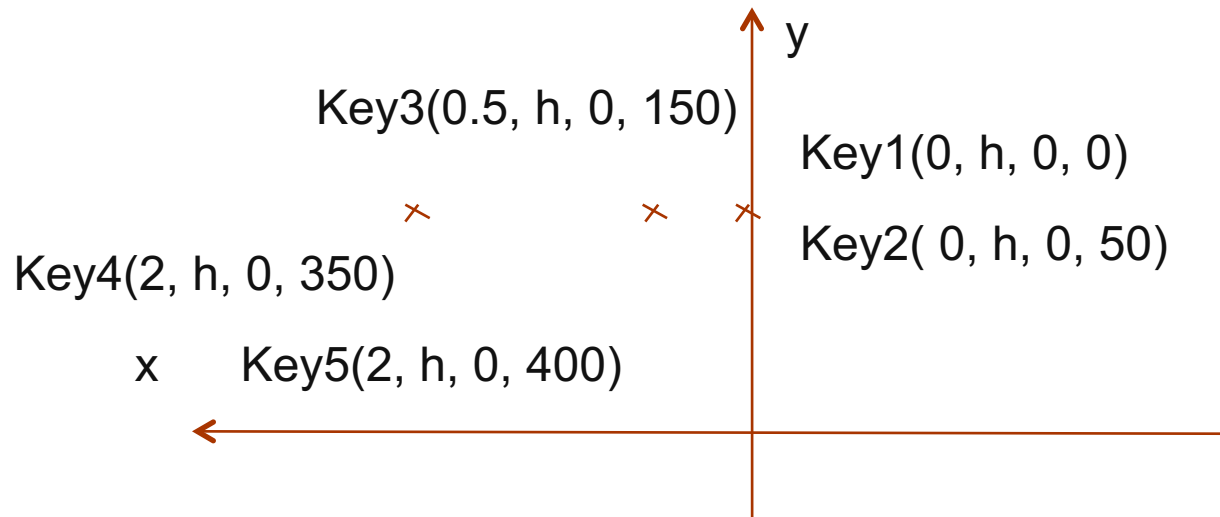
- Pour animer le robot, nous utilisons la méthode de Keyframing.
- Le jambe avancé: interpolation Hermite



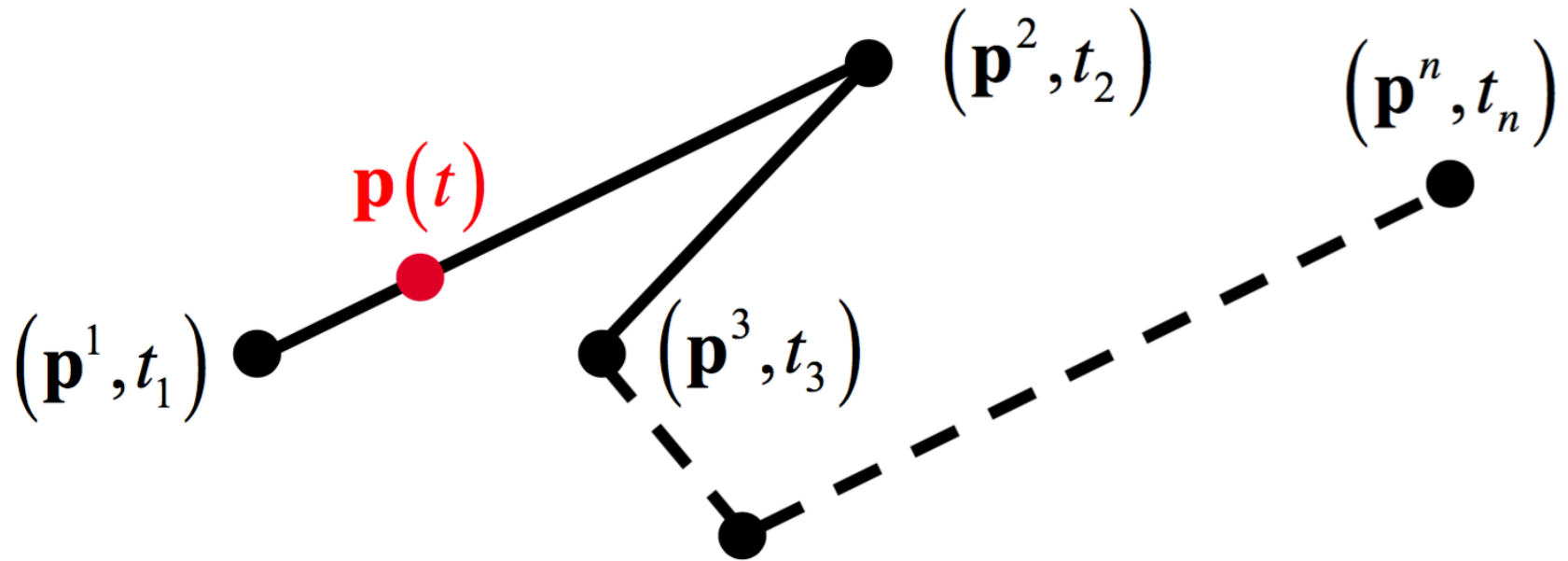
- Le jambe arrière: contrôlé par flag, left et upLeg(t_bone).

Animation -- Keyframing

- Pour avancer le robot, nous utilisons l'interpolation hermite pour `corps(t_bone)`



Keyframing—interpolation linéaire



$$\mathbf{p}(t) = \mathbf{p}(i, u) = (1 - u)\mathbf{p}^i + u\mathbf{p}^{i+1}$$

Pour $t_i \leq t \leq t_{i+1}$, avec $u = \frac{t - t_i}{t_{i+1} - t_i}$

Keyframing--interpolation non linéaire

- Blending Hermite Functions
- $H_0 = 2t^3 - 3t^2 + 1$
- $H_1 = -2t^3 + 3t^2$
- $H_2 = t^3 - 2t^2 + t$
- $H_3 = t^3 - t^2$

$$T_n = \frac{(1-T)(1+B)(1-C)}{2} * (P_0 - P_{n-1}) + \frac{(1-T)(1-B)(1+C)}{2} * (P_{n-1} - P_n)$$

et

$$T_{n+1} = \frac{(1-T)(1+B)(1+C)}{2} * (P_{n-1} - P_n) + \frac{(1-T)(1-B)(1-C)}{2} * (P_{n+2} - P_{n+1})$$

- $P(t) = H_0(t)P_n + H_1(t)P_{n+1} + H_2(t)T_n + H_3(t)T_{n+1}$

Difficulté rencontrées

- **sélection des points clés: influence la connuité de l'animation, encore un peu de saccadé lors du changement des jambes**
- **cohérence entre l'interpolation du corps et l'interpolation des jambes**

Merci à votre attention!