Kat'dventures – Documentation technique

1 Mouvements dans le mode normal

Dans le mode normal, nous utilisons des animations créées avec le logiciel Blender. Vu que le sol est plat, le mouvement reste réaliste. Les animations concernées sont : l'attente, la marche et le saut.

2 Cinématique inverse

L'algorithme de cinématique inverse CCD a été implémenté dans le jeu sous la forme du script `CCD3d.cs`.

Description de l'algorithme CCD

Nomenclature

— p_i est la position d'une articulation q_i du bras ;
— p_e est la position de l'effecteur (c'est-à-dire du bout du bras) ;
— p_t est la position de la cible à atteindre par l'effecteur.

Déroulement de l'algorithme

Point de départ : l'articulation la plus proche de l'effecteur (q_{n-1}).

— On calcule α l'angle formé entre les deux vecteurs $q_i\vec{p}_e$ et $q_i\vec{p}_t$
— On fait une rotation de centre p_i et d'angle α autour de l'axe $q_i\vec{p}_e \wedge q_i\vec{p}_t$, ce qui permet d'aligner $q_i\vec{p}_e$ et $q_i\vec{p}_t$
— Si la distance entre l'effecteur et la cible (c'est-à-dire $||p_e\vec{p}_t||$) devient plus petite par rapport à un nombre ϵ fixé, l'objectif est atteint.
— Dans le cas contraire, deux choix :
  — Si on n'est pas arrivé à la racine du membre ($i > 0$), on itère une autre fois avec l'articulation q_{i-1}.
  — Si $i < 0$, cela veut dire qu'on est arrivé à la racine du membre. On recommence alors avec q_{n-1}.

La figure 1 montre un exemple d'application de l'algorithme.

3 Passage d'un mode à l'autre

Le choix de passage d'un mode à l'autre est dicté par la pression d'une touche par le joueur. À ce moment-là, le script `EnablePush.cs` s'active, et procède au redressement du chat, de telle sorte qu'il soit dans une posture humaine.

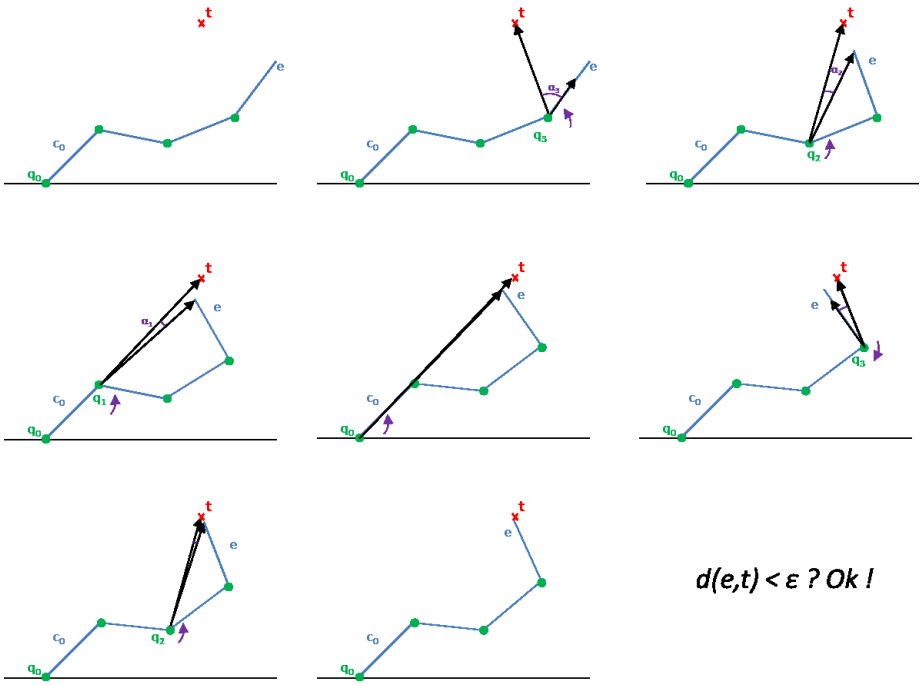

FIGURE 1 – Application de l'algorithme CCD

4 Mouvements en mode super-pouvoirs

Pour faire bouger le chat en position debout, nous nous sommes appuyés sur la bibliothèque *Locomotion System* développé par Rune Skovbo Johansen [1]. Cet ensemble de scripts utilise la cinématique inverse pour obtenir un mouvement réaliste et similaire à celui d'un humain.

5 Choix des prises

Nous utilisons la méthode de notre encadrant, implémenté ici sous le nom de `CatManipulatibility.cs`. Nous avons utilisé ce script comme une boîte noire, car nous l'avons reçu tard dans le projet, et notre encadrant a considéré que son implémentation serait trop prenante. Il est à noter que la méthode complète permet d'obtenir une configuration complète (c'est-à-dire la cible et les positions/rotations des articulations), mais nous nous en servons que pour trouver la cible adéquate.

La fonction qui nous intéresse est `ftr`, qui prend en paramètre un `Transform` et la direction que nous voulons donner au chat, et renvoie un indicateur de "réalisme" de la configuration. Cela nous permet de trouver la meilleure prise. Ensuite, les déplacements des articulations sont effectués par les opérations de `CCD3d.cs`.

L'algorithme utilisé est le suivant :

1. On choisit les prises à proximité (écart d'un ϵ défini)
2. Pour chaque prise p :
   (a) On crée un clone du membre, et on lui fait décrire l'algorithme CCD pour qu'il trouve une configuration

1. Le package *Locomotion System* est disponible sur le Unity Asset Store : *https ://www.assetstore.unity3d.com/en/# !/content/7135*

(b) On appelle la fonction `ftr` du script `CatManipulability` avec la direction voulue de l'utilisateur.

(c) Si la valeur renvoyée par `ftr` est plus grande que pour la prise d'avant, on retient la prise comme étant la meilleure.

6 Intégration des scènes

Notre mode de développement consistait en la création d'une scène par partie de parcours. Ainsi le mur d'escalade ou encore le cube ont été fait chacun dans une scène différente.

7 Menu et Interface Utilisateur

Les menus sont composés d'une image de fond et de boutons. La position des boutons est fixée manuellement, relativement à la résolution de l'écran. La taille des boutons est gérée par le script `AdaptiveFont.cs`. Lorsque l'utilisateur passe la souris au dessus des boutons, la taille de ceux-ci augmente d'un facteur 1,2. Lorsque l'utilisateur clique sur un bouton, il déclenche un événement qui charge une scène.