Réalisation d'un jeu vidéo d'escalade afin de valoriser un travail de recherche en animation procédurale

François Boschet, Thomas Giraudeau,
Delphine Millet, Paul Rivière
Kevin Thek

Encadrant : Steve Tonneau

Résumé

Ce rapport présente le déploiement de notre projet d'études pratiques. L'objectif de ce projet est de mettre en avant un travail de recherche en animation procédurale. Il consiste en la conception et l'implémentation d'un algorithme de cinématique inverse dans un jeu vidéo d'escalade à concevoir également.

1 Introduction

Les problématiques d'animation dans les milieux du jeu vidéo et du dessin animé sont des enjeux importants, car extrêmement perceptibles. En effet, dans la plupart des grands projets de ce type, c'est une armée d'artistes et de techniciens qui œuvrent afin de créer des mouvements toujours plus réalistes.

Cependant, malgré l'évolution constante dans ce domaine et le niveau de qualité atteint aujourd'hui, les techniques d'animation actuelles demeurent insatisfaisantes. Par exemple, certaines configurations de l'environnement peuvent prendre en défaut l'animation créée, comme le déplacement d'un objet que prend le héros d'un jeu vidéo (voir fig.1), ou bien la modification d'une scène dans la création d'un film d'animation. Autant de situations qui complexifient la création de médias.
Parmi les pistes proposées pour résoudre cette problématique, l'animation procédurale fait partie des solutions les plus intéressantes, et c'est ce que nous avons implémenté dans un jeu vidéo, nommé « Kat'dventures » . Nous reviendrons sur l'animation procédurale dans une première partie, avant d'expliquer le gameplay du jeu. Nous évoquerons les technologies utilisées pour créer le jeu en lui-même, puis nous reviendrons brièvement sur l'exécution du projet.

2 Animation procédurale

Parmi les techniques d'animation nous retrouvons deux méthodes principales pour créer les mouvements :

— L'animation procédurale que nous allons développer plus bas.
— La *Motion Capture*, qui consiste en la création du mouvement par des cascadeurs munis de capteurs.

Mais pourquoi utiliser des techniques d'animation procédurale, plutôt que la *Motion Capture* ? En réalité, la *Motion Capture*, même si elle permet d'obtenir des

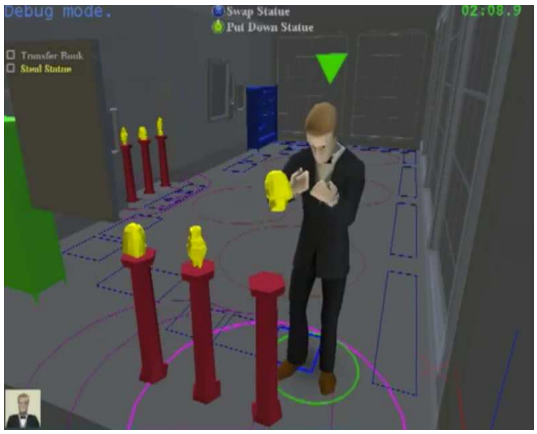

FIGURE 1 – Lorsque l'environnement change, le mouvement n'est pas toujours adapté.

mouvements réalistes, car obtenus directement depuis l'homme, pose le problème de la réutilisation des mouvements. En effet, un mouvement issu de *Motion Capture* est contextuel à un environnement spécifique, et ne peut donc pas être transposé à un autre environnement.

Dans notre cas, nous allons travailler sur les techniques d'animation procédurale. Ce mouvement sera issu d'algorithmes mathématiques, et devra également respecter les contraintes inspirées du monde réel (gravité, limites articulaires ou encore confort de la position). Les moyens de mettre en place pratiquement l'animation procédurale s'appellent les méthodes procédurales.

2.1 Cinématique inverse

Parmi les différentes familles de méthodes procédurales, nous trouvons la cinématique inverse. Le but de cette méthode est simple : une extrémité de membre doit être le plus près possible d'une cible fixée. Le déplacement de l'extrémité du membre va entraîner un déplacement des autres membres reliés avec elle, jusqu'à obtenir une position pour chaque membre.

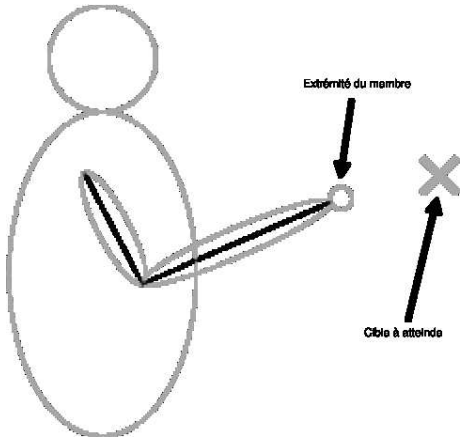

FIGURE 2 – Schéma d'illustration de la cinématique inverse

Pour obtenir ces positions, différents algorithmes mathématiques sont dispo-

nibles : *Cyclic Coordinate Descent* (CCD), *Forward And Backward Reaching Inverse Kinematics* (FABRIK), l'utilisation de la matrice jacobienne, etc. Parmi tous ceux-là, nous avons choisi d'implémenter CCD, car il est plus accessible et plus économe en calcul ce qui est un point important dans un jeu vidéo, afin de garder une fluidité des mouvements correcte.

Pour plus de détail sur l'algorithme CCD, nous vous renvoyons à la documentation technique.

2.2 Gestion des contraintes articulaires

Cependant, contrairement à d'autres algorithmes de cinématique inverse comme FABRIK, CCD ne gère pas les contraintes articulaires. Il s'agit de prendre en compte les butées articulaires naturelles du squelette du personnage, afin de générer des mouvements contraires aux conditions du monde réel. L'enjeu est d'éviter les positions aberrantes du personnage, rendant impossible le mouvement. Ainsi, nous avons dû gérer les contraintes articulaires manuellement, en parallèle de l'algorithme CCD. Là encore, une explication plus détaillée est disponible dans la documentation technique.

2.3 Choix des cibles

Nous avons vu que la cinématique inverse a besoin de cibles sur lesquelles doivent pointer les extrémités des membres. Là encore une problématique se dessine : comment choisir la meilleure cible possible en fonction de l'environnement, mais également du personnage, afin d'obtenir un mouvement réaliste ?

Afin de remédier à cela, nous avons utilisé la méthode introduite dans l'article *Task Efficient Computation of Contact Configurations* (TECCC) [1]. Cette méthode analyse le mouvement global voulu par l'utilisateur (monter, descendre, avancer,...) et adapte alors le point cible en fonction de ce qui entoure le personnage.

3 Gameplay

Kat'dventures est un jeu de plateforme en 3D, dont le but est d'aider un chat à atteindre une pelote de laine située au bout d'un niveau constitué d'obstacles. Face à certains obstacles, les capacités normales du chat ne seront pas efficaces. Il devra alors passer dans un mode lui offrant des super-pouvoirs !

3.1 Mode ordinaire

Le joueur commence le jeu avec un chat normal : il marche et saute comme un chat commun. Cela pourrait suffire pour franchir le niveau, mais certains obstacles restent infranchissables malgré tout.

3.2 Mode super-pouvoirs

C'est alors que le mode super-pouvoirs devient nécessaire. Il peut être déclenché en appuyant sur un bouton du clavier. Lorsque ce mode est activé, le chat se redresse et acquiert certaines aptitudes humaines, comme celle de pousser ou encore d'escalader des parois. Ces fonctionnalités permettent de mettre en avant les possibilités offertes par la cinématique inverse.

Pousser un cube Quand il est debout, la cinématique permet au chat de poser ses pattes sur un cube. Il peut donc déplacer le cube en avançant.

Escalader un mur La cinématique inverse associée à la méthode TECCC permet d'obtenir des mouvements globalement réalistes. Cette association permet de changer les positions et le nombre de prises sur le mur, permettant de concevoir plusieurs configurations différentes et offrant au final une variabilité quasi-infinie.

4 Technologies utilisées

4.1 Création du jeu - Unity3D

FIGURE 3 – Logo d'Unity3D

Afin de créer le jeu vidéo mettant en valeur ces techniques d'animation, nous avons choisi d'utiliser la plateforme Unity3D. Il s'agit d'un moteur de jeu, avec un IDE[1] intégré. Ce logiciel offre, même dans sa version gratuite, de vastes possibilités de manière accessible. Les jeux produits peuvent être compilés et compatibles aussi bien sur ordinateur (Windows, Mac OS, Linux) que sur appareil mobile (iOS, Android), navigateur web ou encore console de jeux (PlayStation, Xbox, Wii).

Unity3D propose ainsi des objets graphiques intégrés, des textures, et beaucoup d'autres éléments. Pour avoir des comportements plus personnalisés, des scripts en `Javascript`, `C#` ou `Boo` peuvent être associés aux éléments de jeu par un simple glisser-déposer. Enfin, le logiciel bénéficie d'une très grande communauté, permettant d'aider au mieux les débutants.

Ce logiciel nous a été fortement recommandé par notre encadrant car il est facile à prendre en main, nous soulageant ainsi de la difficulté de maîtrise technique d'un moteur du jeu. Cela nous a permis de nous concentrer sur l'élaboration des scripts de cinématique inverse et sur la création de l'environnement.

4.2 Création du chat - Blender

Le héros du jeu, un chat, a dû être modélisé à partir d'éléments trouvés sur Internet, proposés gratuitement au téléchargement par des artistes. La mise en place du squelette du chat a demandé une étape de création supplémentaire sur le logiciel de modélisation et d'animation 3D, gratuit et open-source Blender. Un membre du groupe s'est formé parallèlement à la manipulation de ce logiciel. La création du squelette de l'animal a été une étape importante, car elle conditionne une partie non négligeable de la mise en place de l'algorithme de cinématique inverse, en particulier pour les contraintes articulaires. De plus, les animations de base du chat (en mode normal) ont été réalisées également sur Blender.

1. Integrated Development Environment

FIGURE 4 – Logo de Blender

5 Exécution du projet

5.1 Répartition des tâches

Sur les conseils de notre encadrant, nous nous sommes répartis les différentes tâches du projet :

- François et Delphine se sont chargés du level-design et de l'aspect graphique ;
- Thomas s'est chargé de la gestion du chat, notamment sa modélisation en 3D ;
- Kévin et Paul se sont chargés de la création des scripts du jeu.

Cette répartition est restée globalement fixe durant tout le projet.

Parmi les différents aspects à réaliser pour ce projet, nous avons dû implémenter le passage d'un mode à l'autre, en gérant les animations correspondantes ; nous avons également dû implémenter les butées articulaires qui ne sont pas gérées par l'algorithme CCD, ou encore intégrer la méthode TECCC.

5.2 Méthodologie

5.2.1 Encadrement

Notre encadrant nous a bien conseillé sur les technologies à mettre en place, ainsi que pour la création des scripts. Au début du projet les réunions ont été assez fréquentes afin d'être sûrs que nous avancions dans le bon sens. Certains TP nous ont été proposés pour appréhender l'interface d'Unity3D et la programmation en C# des méthodes de cinématique inverse. Quand le projet était bien commencé, nous avons espacé les rencontres afin d'avoir le temps de présenter des avancées intéressantes.

5.2.2 Utilisation d'outils

Pour faciliter la coordination de chacun et le travail collaboratif, nous avons mis en place un système de versions Git sur GitHub. Cela nous a permis d'appréhender et d'expérimenter sur le long terme cet outil. Chaque aspect du jeu avait sa branche dédiée, que nous fusionnions sur la branche `master` régulièrement.

Nous avons eu un peu de mal à gérer l'ensemble des fichiers générés par Unity3D pour chaque scène travaillée. En effet, le nombre impressionnant de fichiers cachés créés rendait l'utilisation de Git assez difficile. Nous nous sommes rendu compte qu'il était compliqué d'intégrer les différents types de fichiers, depuis les scènes Unity3D (en binaire) aux fichiers cachés, en passant par les fichiers de configuration

du projet ou ceux de l'utilisateur. Unity3D propose un système de gestion de version, mais uniquement réservé aux utilisateurs munis d'une licence *Team* (*Pro* + Add-on). Malgré son prix, ce logiciel pourrait être un réel plus pour gérer les projets sur Unity3D de manière simplifiée.

Pour la gestion du projet, nous avons commencé à communiquer principalement par les moyens traditionnels (mail, SMS), rendant le suivi compliqué. Nous avons donc utilisé les pages de Wiki offertes par la plateforme GitHub pour héberger les comptes-rendus et les pages rendant compte de l'avancement de chacun sur la partie le concernant. Ainsi fixer des objectifs intermédiaires était plus facile, et la communication interne au groupe facilité.

6 Idées d'amélioration

Le projet étant assez vaste, de nombreuses améliorations sont possibles. En plus de l'ajout d'autres niveaux, nous pourrions citer :

L'intégration d'un timer avec des objectifs de scoring : une échelle de score pourrait être ajoutée, en tenant compte du temps passé sur chaque niveau.

Ajout de puzzles à utiliser avec une WiiMote : Un rapide puzzle pourrait verrouiller la porte de fin de niveau. Leur résolution demanderait de faire bouger la patte du chat (à l'aide d'un pointeur laser) afin qu'elle appuie sur des boutons ou fasse bouger des objets. Pour le choix des puzzles, nous aurions pensé à un taquin, un cassage de blocs, etc.

Permettre d'autres mouvements au chat : l'élargissement des possibilités des mouvements du chat permettrait la création de nouveaux obstacles (des tuyaux à grimper ou des sauts de l'ange par exemple).

Améliorer les graphismes : le réalisme et le rendu des textures peuvent être perfectionnés.

7 Conclusion

Le projet Kat'dventures nous a permis d'appréhender la cinématique inverse, et de manière plus large les méthodes procédurales. Nous nous sommes rendu compte des possibilités que cela offre pour les créateurs de jeux vidéos, mais aussi le travail de recherche que cela demande pour créer les algorithmes.

Il s'agit bien de créer des animations les plus réalistes possible tout en garantissant l'adaptation du mouvement à l'environnement existant. C'est une bonne solution pour éviter les schémas répétitifs dans les jeux vidéos exploitant d'autres techniques comme la *Motion Capture*.

Références

[1] Steve Tonneau, Julien Pettré et Franck Multon : Task Efficient Contact Configurations for Arbitrary Virtual Creatures. *In Proceedings of Graphics Interface 2014 - To appear*. ACM, May 2014.