

Rapport animation Nicolas Pronost

Barros Mikel-ange

I) Présentation du TP

L'objectif de ce tp est de faire tenir un personnage en équilibre le plus longtemps possibles et ce malgré des évènements visant à le faire tomber tels que la rotation de la plateforme sur laquelle il se tient ou l'envoi de balle sur son corps. Afin de mettre en place cela nous définissons notre personnage comme étant un corps rigide articulé composé d'un ensemble de membres reliés entre eux par des articulations. Chacun des membres du personnage aura son propre poids et ses propres dimensions. Ainsi à partir de ses différentes informations et des forces appliquées sur notre personnage, nous pouvons définir des moments cinétiques pour chacune des articulations ce qui lui permettra de rester au sol. Le moment cinétique est une force appliquée dans le cadre d'une rotation et qui va avoir les mêmes propriétés que la quantité de mouvement dans le cadre d'une translation. Il va ainsi permettre d'appliquer une résistance aux différentes forces perçues par le personnage et ainsi lui permettre de rester debout. La formule du moment cinétique en un point M par rapport à un point O est décrite comme étant le produit scalaire entre la quantité de mouvement du point et le vecteur OM.

II) Introduction

Que ce soit pour représenter des évènements climatiques ou juste le comportement d'un objet ou d'un animal, la physique a toujours été un point essentiel de l'animation. Et il en va de même pour l'animation de personnages réalistes et notamment pour ce qui concerne le contrôle de mouvement. En effet, dans les jeux vidéo, le personnage est animé par deux grands contrôleurs : le premier va lui faire prendre des positions prédéfinies telles que la marche ou le saut, alors que le deuxième va le faire réagir en fonction des différentes actions qui se passent autour de lui. Ces deux contrôleurs agissent ensemble afin de donner aux jeux vidéo un rendu le plus réaliste possible. Dans ce document nous nous concentrerons exclusivement sur le deuxième contrôleur, celui réagissant aux forces. Comme dit dans la présentation du TP, dans notre cas ce contrôleur a donc pour but de maintenir notre personnage le plus droit possible. Afin de tester le contrôleur et de l'améliorer nous pouvons rajouter divers éléments à notre personnage ou à la simulation physique. Je vais détailler les différentes améliorations choisies et leur fonctionnement dans les deux parties suivantes.

III) Choix des améliorations

III.1) Ajout d'une tête

Afin de rendre notre personnage plus réaliste le premier ajout que j'ai décidé de faire a été de prolonger son corps afin qu'il ai une tête. Cette première amélioration avait pour but de voir comment l'ajout d'un élément pouvait influencer le comportement de notre personnage vis-à-vis des forces qui l'entourent.

III.2) Ajout d'un bras

Une fois que la tête a été ajoutée et que nous sommes parvenu à un personnage stable, un deuxième ajout logique m'a semblé être l'ajout de bras. En effet, dans la réalité chez l'être humain les bras ont un rôle de stabilisateurs ce qui lui permet de tenir bien plus longtemps dans une situation difficile et l'objectif ici était de reproduire ce comportement.

IV) Mise en place des améliorations

IV.1) Ajout d'une tête

Afin d'ajouter une tête j'ai dû créer un nouvel élément de corps, puis lui assigner une nouvelle articulation. La difficulté ici a été de trouver les bons paramètres pour que la tête soit stable et ne fasse pas chuter mon personnage.

IV.2) Ajout d'un bras

J'ai créé le bras de la même manière que la tête. La principale différence a été qu'ici je voulais que le bras serve à stabiliser la structure. Et c'est pourquoi je lui ai donné un poids bien plus important que la tête ainsi qu'un mouvement extrêmement fort. Ainsi, le bras est un élément influençant grandement le mouvement du personnage et permettant une meilleure stabilité à la structure.

V) Difficultés rencontrées

V.1) Améliorations abandonnées

V.1.a) Création d'un chien

L'objectif de cette amélioration ,que j'ai abandonné, était de remplacer le corps vertical de l'humain avec un seul point d'appui par un corps horizontal ressemblant à celui d'un chien et ayant deux points d'appui. En effet, dans la réalité les quadrupèdes ont un centre de gravité bien plus bas que les bipèdes et ont donc une bien meilleure stabilité. Cependant l'ajout d'un deuxième point d'appui rajoutait des contraintes et

des calculs de forces au niveau des moments à appliquer aux différentes parties du corps et je n'ai pas trouvé les valeurs adéquates au bon fonctionnement de ce quadrupède. Comme vous pouvez le voir (figure 1) il avait une jambe tordue qui le faisait tomber en permanence.

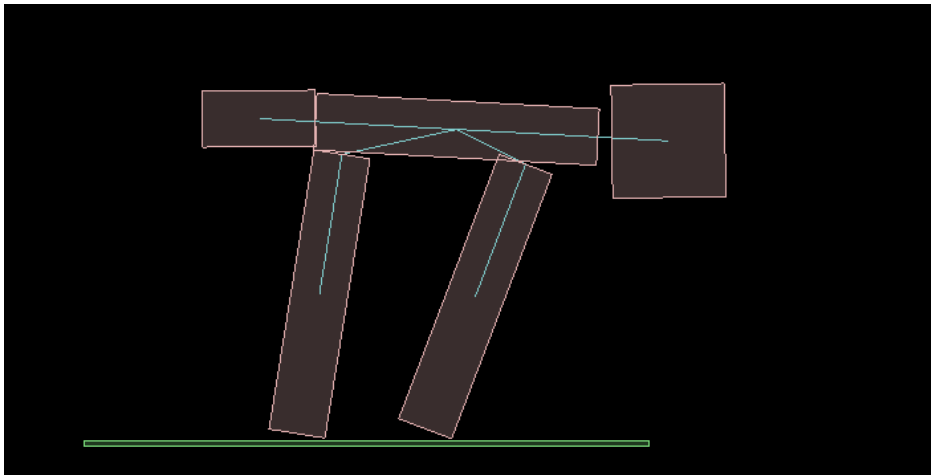


Figure 1 : quadrupède

V.1.b) Faire sauter la créature

Afin de faire sauter la créature bipède j'ai eu dans l'idée de modifier le fichier motion.txt afin de le faire sauter sur place. Malheureusement je me suis vite rendu compte que motion.txt n'autorisait que des mouvements à l'horizontale et ne trouvant pas un mouvement de balancier capable de faire sauter mon personnage j'ai aussi abandonné cette partie là.

V.2) Difficultés générales

La plus grosse difficulté que j'ai rencontrée dans ce tp fut la compréhension du code et de ce qui était attendu. En effet, au départ j'étais pas mal perdu pour comprendre le fonctionnement de la créature et il m'a fallu de nombreuses heures pour arriver à mes premières modifications.

VI) Conclusion

Ce Tp est une bonne approche de la physique appliquée à un personnage afin de lui faire avoir une attitude et des mouvements réalistes. De plus, il permet de découvrir et expérimenter sur un modèle plus ou moins simple comment sont appliqués et exploitées les différentes forces pouvant s'appliquer sur le corps. Cependant, il m'est vite apparu que pour obtenir des mouvements très réalistes et un fonctionnement cohérent du début à la fin de la simulation, de nombreux paramètres devaient être prédéfinis, connus et extrêmement précis ce qui rendait le travail en amont extrêmement compliqué et n'exclut pas la possibilité d'un bug ou d'un problème dans un monde aussi libre qu'un jeu vidéo par exemple. En effet, la complexité d'un modèle de contrôle de personnages évolue en fonction du nombre de force qui

s'applique dessus et pour un jeu vidéo cela peut très vite devenir compliqué à gérer. Et ce encore plus en considérant la quantité de calculs considérable à faire risquant de retarder l'exécution dans le cadre d'un projet en temps réel.