

INF585 - COMPUTER ANIMATION

- Là-Haut !

Jean-Peïc Chou & Pierre Fernandez

MOTIVATIONS

Après le MODAL Modélisation expressive pour la fabrication 3D [1], c'est avec enthousiasme et nostalgie (avouée) que nous avons décidé de nous inspirer une nouvelle fois du film d'animation Pixar : *Là-Haut*. Notre objectif a cette fois-ci été d'obtenir un rendu plus réaliste et de mettre en œuvre de nouvelles techniques d'animation vues en cours. Si un souci de fidélité au film [2] nous a suivi lors de la conception de la maison, nous en avons également profité pour proposer de nouvelles fonctionnalités, pour se déplacer notamment.



Image tirée du film d'animation "Là-Haut"



DESIGN LOGICIEL

Nous avons travaillé en parallèle sur chaque partie de l'animation avant d'attacher le tout à la maison. L'optimisation de chaque pièce a été nécessaire afin d'avoir un rendu final correct.

Ballons, collisions, fils

Les ballons comprennent plusieurs niveaux de modélisation :

- Leur forme a d'abord été donnée par un cône surmonté d'une boule, mais cela donnait lieu à une ombre malvenue à l'intersection des deux parties. Nous avons par la suite utilisé un objet plus réaliste (rondeur, légère asymétrie, nœud...) exporté sous Blender.
- Pour la partie physique, les ballons sont représentés par une classe nommée *particle_structure* donnant notamment leur position, leur vitesse et les forces s'appliquant sur ceux-ci. La poussée d'Archimède permet ainsi de faire voler les ballons malgré la gravité. Lorsqu'un ballon s'éloigne de la cheminée d'une certaine distance, à savoir la longueur du fil attribué, nous retenons celui-ci en le remplaçant à bonne distance et en lui donnant une légère impulsion dans l'autre sens proportionnelle à sa vitesse pour obtenir une simulation réaliste. Nous avons également implémenté la gestion des collisions entre les ballons en les représentant par deux boules l'une au-dessus de l'autre. Nous aurions pu obtenir une gestion plus fine des collisions en considérant N boules mais avons considéré que le résultat était esthétiquement optimal.



Les fils sont simulés par N (par défaut 10) particules invisibles liées par des ressorts, soumises à la gravité et à des frottements importants. Ils sont représentés en traçant des segments entre les particules qui se suivent. Les conditions aux limites sont les suivantes :

- La première particule est attachée à la cheminée (dans le code, la position d'attache est donné par `fil_structure.fix`)
- La dernière particule est attachée au ballon

Les collisions entre les fils et les ballons ne sont pas prises en compte. Bien que cela n'ait pas été plus compliqué, la simulation aurait demandé trop de calculs. Pour cette même raison, les fils ne sont simulés que par deux particules lorsqu'ils sont stables : la première et la dernière sont conservées, les autres sont supprimées.

Lorsque qu'un ballon est lâché (bouton *remove*), la première particule est libérée. Le ballon peut désormais s'envoler librement. Il est retiré de la simulation quelques secondes après qu'il a été lâché. Les ballons sont ajoutés grâce au bouton *add*.

Au-delà d'une dizaine de ballons, la simulation devient trop lente pour être visualisée correctement (<15 fps). La gestion des collisions implique une complexité en $O((\text{nombre de ballons})^2)$ et chaque fil ajoute N particules à la simulation.

Propulseurs

Les "propulseurs" ou "tuyaux d'échappement" correspondent simplement à des cylindres placés derrière la maison. Les propulseurs comprennent deux modes de fonctionnement :

- Le mode *motor_off* : de la fumée sort périodiquement des tuyaux comme lorsqu'une voiture est à l'arrêt (mais démarrée). La fumée est représentée par des billboards. Afin d'obtenir une animation expressive, le cylindre est déformé périodiquement lors du passage de la fumée dans le tuyau selon la formule d'une onde de déformation.
- Le mode *motor_on* ou *boost* : des flammes jaillissent des tuyaux et la maison se déplace si elle est dans les airs. Le mode est activable à partir des touches fléchées. Le tuyau adopte alors la forme d'un pavillon (de trompette). Les "flammes" sont représentées par les mêmes billboards, colorés cette fois-ci. Ces derniers sont projetés aléatoirement dans un cône. Plus la fumée est au centre, plus elle devient jaune (plus chaude qu'à l'extérieur). Une simulation physique des flammes aurait été trop coûteuse en calculs.



La fumée a une certaine espérance de vie (celle de la fumée en mode *boost* est plus courte). Elle disparaît après une durée égale à celle-ci.

Rideaux



Les rideaux ont été simulés à l'aide d'un système masse-ressorts. Pour ce modèle, la toile est subdivisée en une grille $N \times N$ où sont échantillonnées les particules.

Chaque particule, de masse m , est soumise à :

- son poids mg
- du frottement $-\mu v$
- une force due au vent (ou plutôt friction due à l'air) : $S \cdot (\vec{w} \cdot \vec{n}) \cdot \vec{n}$, qui dépend de la surface d'une parcelle de la grille S et de la vitesse de la maison (on fixe $\vec{w} = -\vec{v}$). Cette force est dirigée selon la normale à la surface de parcelle.
- des forces de ressorts issues des voisins les plus proches avec une constante de raideur et une longueur à vide considérées constantes pour un type d'interaction.

Puis la solution est approchée à chaque pas de temps à l'aide d'une méthode d'Euler semi-implicite, i.e. :

$$\begin{cases} v_{t+dt} = v_t + dt \times F \\ p_{t+dt} = p_t + dt \times v_{t+dt} \end{cases}$$

Les paramètres à choisir sont donc le nombre de particules du système et les paramètres présents dans le calcul des forces. Ces paramètres ont été choisis de façon à simuler un tissu relativement rigide et ne demandant pas trop de ressources (faible nombre de particules $N = 12 \times 12$).

Enfin le tissu est attaché par à une tringle. À chaque pas de temps, certaines positions sont donc imposées au niveau de la grille du rideau (ceci se traduit dans le code par une mise à jour des *positional_constraints* du rideau à chaque étape de la boucle).

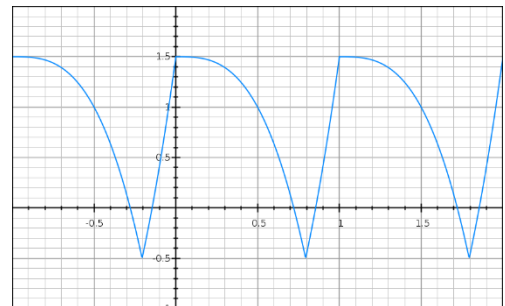
Oiseaux

L'animation des oiseaux se fait à deux niveaux :

a. Le battement d'ailes

Pour réaliser le battement d'ailes, une option aurait été d'utiliser un squelette, mais le processus de création du squelette et de rigging est un processus long et fastidieux. Une façon plus facile d'accès est l'interpolation de formes. À partir d'un modèle d'oiseau, un autre modèle a été créé avec les ailes en position haute. Pour simuler un battement d'aile, il suffit alors d'interpoler chaque vertex entre les positions qu'il occupe dans chacune des deux formes.

Pour améliorer l'animation, l'interpolation n'est pas linéaire mais cubique, ainsi la fonction qui donne le facteur d'interpolation en fonction du temps est : $-e + (1 + 2 * e) * |2 * ((t - [t])^3 - 0.5)|$, où e est un facteur d'extrapolation, permettant de s'éloigner un peu des formes d'origine et t est le temps. (la fonction est représentée à droite)



b. Le comportement des oiseaux

Le comportement des oiseaux en vol a été simulé à l'aide d'un système inspiré des boids [3] (classe *bird* dans le code). Ce système se caractérise par 3 types forces :

- Cohésion : les oiseaux ont tendance à rester cohérent, ceci se traduit par une force : $f_{cohes} = \lambda_{cohes} \times (c_p - p)$, où c_p est le barycentre de toutes les positions et p est la position de l'oiseau.
- Séparation : des oiseaux trop proches se repoussent, ce qui se traduit par une force répulsive $f_{separ} = \lambda_{separ} \times (p - c_{p-close})$, où $c_{p-close}$ est le barycentre des positions des oiseaux considérés comme proches (à une distance $d < \delta_{close} = 5$ par défaut), et p est la position de l'oiseau. Pour éviter d'avoir un changement trop brutal de forces, les voisins plus lointains sont aussi considérés mais avec un poids plus faible.
- Alignement : des oiseaux proches ont tendance à s'aligner en eux, la force résultante est : $f_{align} = \lambda_{align} \times (c_{v-close} - v)$, où $c_{v-close}$ est le barycentre des vitesses et v la vitesse des oiseaux.

Les oiseaux suivent la maison, à ces forces nous avons donc ajouté une force d'attraction-répulsion de la maison définie par : $\log\left(\frac{d}{d_{carac}}\right) + \left(\frac{d}{d_{carac}}\right)^2$ et une force d'alignement avec la maison qui pousse les

oiseaux à adopter la même vitesse que la maison, similaire à f_{align} . La force d'attraction répulsion a été choisie de telle sorte qu'elle soit répulsive lorsque les oiseaux sont trop près ($d < d_{carac} = 6$) et de plus en plus attractive dans le cas contraire. Enfin afin d'éviter que les oiseaux rentrent dans le sol, nous avons ajouté une force de répulsion due au sol : cette force est inversement proportionnelle à la hauteur des oiseaux. Ces forces seront discutées dans la partie suivante.

Maison, assemblage et animation globale

La maison est représentée par un objet importé que nous avons trouvé sur 3D Warehouse, puis modifié sur Blender pour le rendre plus simple. Les fils des ballons, les propulseurs et les rideaux sont attachés à la maison. Leur position sont mises à jour à partir de celle de la maison (`house_boid` dans le code).

L'ensemble se déplace selon des principes physiques simples. À partir d'un certain nombre de ballons, la maison s'envole et se stabilise à une certaine hauteur (atmosphère à densité décroissante). Bien sûr, la simulation n'est pas réaliste car il faudrait bien trop de ballons pour faire voler une maison, ce qui aurait demandé trop de calculs. Lorsque la maison est en vol, elle peut se déplacer grâce aux touches fléchées qui activent le moteur et confèrent une force de propulsion à la maison. Les rideaux et les ballons, attachés à la maison, se mettent naturellement en mouvement. Le sol est un parallélépipède plat simple sur lequel repose la maison. Celle-ci ne peut pas s'écraser en dessous lorsque les ballons sont lâchés.

Décor, modèles et textures

Même si le but premier du projet était l'animation, nous avons beaucoup travaillé à rendre les modèles esthétiques et à faire en sorte que les textures soient cohérentes. Pour cela, il a fallu prendre en main Blender, un outil assez complexe mais qui a permis d'exporter et de modifier les fichiers .obj obtenus sur internet. Au niveau de la texture de la maison, les coordonnées uv ont été définies à partir d'une projection carrée dont les axes coïncident avec ceux de la maison, et dont la surface correspond à peu près à un cinquième de la face avant de la maison. Ainsi, les coordonnées uv s'échelonnent de -2 à 3, et la partie manquante de texture est complétée par répétition de la texture (qui a été choisie seamless pour répondre à ce besoin). Il en a été de même pour le parallélépipède représentant le sol, dont les coordonnées uv ont dû être définies.

Enfin, un dernier élément de décor ajouté est la skybox, une boîte à grande échelle dont la texture a été téléchargée sur internet et dont le modèle provient du cours INF443 [4].



RESULTATS & DISCUSSIONS

Le résultat final de l'animation (disponible en vidéo) est plutôt satisfaisant, tout élément confondu. Le comportement des oiseaux est cohérent, celui des voiles et des ballons tout autant. Cependant, quelques remarques peuvent être faites donnant lieu à de possibles améliorations.

L'animation des ballons est très coûteuse dès que le nombre de ballons est important, rendant une modélisation semblable à celle du film impossible. Une façon de procéder aurait été de regrouper des ballons en billboard dès que leur nombre aurait dépassé une certaine limite, ou de considérer une base fixe de ballons. Sur le même sujet, l'intégration du mouvement des fils donne parfois lieu à des divergences dans leur animation, qui ont été réduites en empêchant un mouvement trop brutal. Toutefois,

il arrive que des oscillations s'installent lorsque le nombre de ballons devient trop important : c'est un point d'amélioration de l'animation.

Concernant l'animation des oiseaux, on remarque quelquefois des comportements anormaux d'oiseaux qui restent sur place et se retournent brutalement. Ceci est un point mis en avant dans le papier original de Reynolds [3]. Pour avoir un comportement plus réaliste, l'étape suivante est de prendre en compte plutôt qu'une force de répulsion, une action de « steer-to-avoid », qui présente l'avantage de n'avoir pas l'effet de retournement brutal.



CONCLUSION

Alors que le projet effectué l'an dernier en ThreeJS visait à développer une modélisation expressive de la maison et de ses ballons, ce projet a permis l'élaboration de modélisations plus complexes, prenant en compte les collisions entre ballons, la présence de rideaux qui paraissent tout à fait réalistes, la fumée sortant des propulseurs ou même d'oiseaux artificiels suivant la maison.



Il était très intéressant de trouver des manières de représenter des phénomènes assez complexes, comme le mouvement d'un rideau ou les interactions entre ballons, sans simulation physique trop complexe ou lourde en calcul. En résumé, ce que nous retenons du projet est que le plus important en computer animation est de trouver une façon simple de modéliser en temps réel des phénomènes complexes.



REFERENCES & LIENS UTILES

- [1] Projet de MODAL réalisé en mars 2019, <https://github.com/pierrefernandezgenestier/MODAL-Up>
- [2] Scène de décollage dans le film Pixar *Up* <https://www.youtube.com/watch?v=LBqE6KdqJ20>
- [3] Flocks, Herds, and Schools: A Distributed Behavioral Model, Craig W. Reynolds
<https://team.inria.fr/imagine/files/2014/10/flocks-hers-and-schools.pdf>
- [4] INF443, Marie-Paule Cani, <https://moodle.polytechnique.fr/course/view.php?id=5295>
- [5] INF585, Damien Rohmer, <https://moodle.polytechnique.fr/course/info.php?id=5645>