

RAPPORT DE PROJET

Simulation de collision d'une particule sur une surface appliquée au golf



Visualisation d'une partie de golf sur le jeu

Bernadet Pierre



I. Introduction

Pour ce projet, j'ai souhaité mettre en place une simulation interactive sous la forme d'un jeu de golf. Ayant commencé le golf cette année, ce thème m'a paru particulièrement pertinent et motivant. Il m'a offert l'occasion de combiner un intérêt personnel récent avec des concepts d'informatique graphique abordés en cours. Pour cela, j'ai d'abord mis en place l'environnement représentant un parcours de golf. Je me suis ensuite attaqué à la modélisation physique du mouvement de la balle, avant d'ajouter plusieurs extensions afin de rendre le jeu plus agréable et ludique.

II. Initialisation de l'environnement

L'environnement est d'abord formé par un terrain texturé rectangulaire pour lequel une fonction de hauteur est définie en chaque point. Cette fonction repose sur la combinaison de plusieurs "collines" centrées en des positions spécifiques, à l'aide de gaussiennes paramétrées, ainsi que sur l'ajout d'un bruit de Perlin pour créer un aspect naturel et irrégulier. Le bruit est appliqué de manière contrôlée : il est atténué progressivement dans certaines zones, notamment autour de l'axe horizontal $y=0$, afin de conserver des surfaces plus lisses et praticables pour le jeu.

Cette atténuation est rendue possible grâce à une fonction de lissage basée sur le cosinus, qui permet de faire varier en douceur l'amplitude du bruit en fonction de la position sur le terrain. Par ailleurs, une dépression circulaire est ajoutée pour simuler un green, zone plane entourant le trou, dont l'effet s'applique également de façon progressive à l'aide d'une fonction *smoothstep*, évitant ainsi des transitions abruptes dans le relief. L'ensemble du terrain est discrétisé en sommets répartis sur une grille régulière, chaque sommet possédant une position 3D, une coordonnée de texture et une normale calculée par différentiation numérique centrale sur la fonction de hauteur. La normale sera utilisée lors de la collision de la balle au terrain.

Une surface d'eau est également ajoutée à la scène pour enrichir l'environnement. Elle est générée à partir d'une grille similaire à celle du terrain, avec une hauteur définie par un bruit de Perlin de faible amplitude, ce qui permet de créer un effet de vagues. Afin d'obtenir un rendu réaliste, la transparence est activée lors du rendu grâce au blending OpenGL (`glEnable(GL_BLEND)`).

Une skybox est également utilisée afin de simuler le ciel. Elle est construite à partir d'une image panoramique découpée en six faces, puis mappée sur un cube englobant la scène. Son rendu se fait sans écriture dans le *depth buffer*, ce qui garantit qu'elle reste toujours en arrière-plan, derrière les autres objets de la scène.

Des éléments végétaux sont ajoutés pour enrichir visuellement le parcours. L'herbe est représentée par de simples quadrilatères texturés, orientés dynamiquement en fonction de la caméra pour un effet de "billboard", et distribués aléatoirement sur le terrain. Les arbres, quant à eux, sont composés de plusieurs maillages (tronc, branches et feuillage) chargés à partir de fichiers .obj, et placés manuellement en des points précis du terrain. Le feuillage utilise un shader de transparence.

Un drapeau est placé au niveau du trou pour indiquer visuellement la cible. Il est composé d'un mât cylindrique et d'un quadrilatère texturé animé par un shader personnalisé. Ce shader applique une déformation sinusoïdale sur les sommets en fonction de leur coordonnée horizontale et du temps, simulant ainsi un flottement au vent plus marqué vers l'extrémité du drapeau grâce à un facteur d'intensité progressif (*smoothstep*). Le trou est représenté par un disque noir, centré dans une zone plus large texturée pour simuler le green.

La balle, représentée par une sphère claire, est initialisée à une position de départ fixe, tandis qu'une flèche cylindrique indique dynamiquement la direction et la puissance du prochain tir. Ces éléments visuels servent de point d'ancrage pour la partie centrale du projet : la simulation physique du mouvement de la balle, que nous détaillons dans la section suivante.

III. Physique de la balle

La physique de la balle repose sur une intégration simple des équations du mouvement sous l'effet de la gravité, combinée à une gestion des collisions avec le terrain. À chaque image, la vitesse est modifiée par l'accélération gravitationnelle \vec{g} , puis utilisée pour mettre à jour la position. Lorsqu'une collision avec le sol est détectée (en comparant la position de la balle avec la hauteur du terrain local), une correction de la position est appliquée pour éviter que la balle ne pénètre dans le sol. Un rebond est ensuite simulé en inversant la composante normale de la vitesse, pondérée par un facteur de restitution.

En plus du rebond, des frottements sont introduits pour ralentir la balle. Dans le plan tangent au sol, la vitesse est amortie différemment selon que la balle se trouve sur le terrain ou sur le green (zone plus lisse). Si la vitesse devient suffisamment faible (norme < 0.15), la balle est considérée comme arrêtée.

Enfin, des conditions supplémentaires gèrent les cas limites : une balle hors du terrain est réinitialisée, et l'entrée dans le trou est détectée par une proximité en position et une faible vitesse.

$$\begin{array}{l} \vec{v} \leftarrow \vec{v} + \vec{g} \cdot \Delta t \\ \vec{p} \leftarrow \vec{p} + \vec{v} \cdot \Delta t \\ \text{si contact : } \vec{v} \leftarrow \vec{v} - (1 + e) \cdot (\vec{v} \cdot \vec{n}) \cdot \vec{n} \\ \vec{v}_{\text{tangent}} \leftarrow \vec{v}_{\text{tangent}} \cdot f \quad \text{avec } f < 1 \end{array}$$

Algorithme de mise à jour de la physique de la balle (LaTeX)

Lorsque la balle est immobile, une flèche cylindrique est affichée juste au-dessus d'elle pour visualiser la direction et la puissance du prochain tir. Elle est orientée dynamiquement selon

les angles de visée définis par l'utilisateur, et sa couleur varie progressivement du bleu au rouge en fonction de la vitesse sélectionnée, offrant ainsi un retour visuel intuitif. L'utilisateur peut ajuster l'angle horizontal et vertical de la visée à l'aide des touches A et D, puis W et S. La vitesse de tir peut être modifiée avec les touches Q et E. Lorsque la flèche est correctement orientée, l'utilisateur peut appuyer sur la barre d'espace pour déclencher le tir.

IV. Extensions

Plusieurs éléments ont été intégrés pour enrichir l'expérience de jeu au-delà de la simple simulation physique. Trois clubs différents sont proposés : un fer 7, un wedge et un putter. Chacun limite les plages de vitesse et d'angles de tir disponibles, ce qui permet de retrouver les caractéristiques distinctes associées à chaque type de club. Un mode de caméra dynamique permet à l'utilisateur de choisir entre une vue libre ou un suivi automatique centré sur la balle. Le nombre de coups effectués est comptabilisé automatiquement, et s'affiche dans une interface ImGui en temps réel. Lorsqu'un tir réussit à faire entrer la balle dans le trou, un message félicitant l'utilisateur apparaît à l'écran pendant quelques secondes.

Enfin, j'ai tenté d'ajouter des effets visuels supplémentaires en utilisant des imposteurs (billboards), pour générer une éclaboussure et une impacte lorsque la balle tombe dans l'eau ou frappe violemment le sol. Cependant, ces éléments n'ont pas pu être finalisés à temps et n'ont donc pas été intégrés à la version actuelle du projet.

V. Conclusion

Ce projet m'a permis de mettre en pratique les notions d'informatique graphique vues en cours à travers la réalisation d'un jeu de golf interactif. J'ai pu modéliser un environnement complet, simuler la physique d'une balle en interaction avec un terrain, et intégrer des éléments d'interface et de contrôle utilisateur. Certaines extensions, comme les effets d'impact, n'ont pas pu être finalisées, mais le projet dans son ensemble fonctionne de manière satisfaisante et offre une base solide pour d'éventuels développements futurs.