

Guide de l'utilisateur

Dans notre application appelée ShootingSIM, l'utilisateur peut simuler des tirs de balles de type G1 dépendamment de différents paramètres comme le vent et la distance de la cible jusqu'à une distance de 800 mètres. Dans l'application, on peut déplacer la cible pour la placer à une distance de 100 à 800 mètre par incrément de 100 mètres. Il y a donc 8 distances possibles.([Figure 1.](#)) L'utilisateur peut aussi rajouter un vent dans la direction et la force voulue qui permet de plus ou moins faire dévier la trajectoire de la balle.([Figure 2.](#)) Pour compenser l'action du vent, il est possible à l'utilisateur de changer l'angle horizontal de l'arme jusqu'à 3 degrés à droite ou à gauche. Il y a aussi possibilité d'ajuster l'angle vertical de l'arme jusqu'à un maximum de 0.65 degrés parce qu'avec ce type d'arme, un angle plus grand ne permettrait pas une distance bien plus grande en raison de la traînée.([Figure 3.](#)) Parlant de traînée, l'application prend en compte la résistance de l'air pour réduire la vitesse de la balle de plus en plus à mesure de son temps de vol. Pour mieux visualiser les distances, il y a des panneaux à droite placés à chaque 50 mètres. Pour aider aussi à ajuster son tir, l'utilisateur peut se servir de deux vues, une de côtés([Figure 5.](#)) et une de dessus([Figure 6.](#)), qui permettent d'analyser la trajectoire de la balle selon l'axe des Z et des Y. La cible est d'ailleurs présente sur ces vues pour permettre de voir la distance à laquelle la balle est passée des cibles. À la pression du bouton "fire in the hole"([Figure 4.](#)), un tir sera effectué qui sera accompagné d'un bruit de coup de feu pour ajouter un peu d'immersion. Ensuite, une fenêtre de dialogues apparaît après chaque tir pour indiquer à l'utilisateur s'il a réussi, ou non, à toucher la cible. Enfin, l'utilisateur a une certaine liberté avec la taille de la fenêtre qui peut s'adapter à différentes grandeurs d'écrans pour voir les objets 3d comme on le devrait(Java gère mal la 3D et le redimensionnage de fenêtre ensemble).

Pour les fonctionnalités non implémentées, il y a tout d'abord le fait que l'utilisateur ne voit pas l'arme avec laquelle il tire. On a eu trop de problème avec le rafraichissement d'objet devant la caméra qui faisait disparaître d'autres objets. On avait aussi prévu de rajouter d'autre force en jeu lors d'un tir de balle comme l'effet Magnus ou encore la portée. On avait aussi pensé implémenter d'autres types de balle mais le coefficient de traînée étant dur à trouver pour chaque type, nous avons décidé de nous concentrer sur le type G1 en particulier, la .38 Super. On voulait aussi avoir une interface plus clémentine pour les yeux mais la difficulté d'importer des modèles et des textures dans Java nous a découragé. On a aussi réduit la précision que l'utilisateur pouvait entrer des valeurs pour une utilisation plus axée sur la facilité d'utilisation plutôt que sur la profondeur. Nous avons donc seulement des "Sliders" pour gérer les angles et la force du vent. Finalement, l'affichage en temps réel de la trajectoire de la balle est passé à la trappe en raison de la complexité de gérer un mouvement de l'espace 3D de Java et du fait que les vues sont suffisantes pour analyser la trajectoire du projectile.

Bugs :

- Beaucoup d'objet 3d affiche en noir(alors qu'elle devrait avoir des couleurs).
- L'interface peut disparaître derrière les objets.
- Changer la taille de la fenêtre peut engendrer des bugs d'interface
(conseillé : 1920x1080p)
- Le message qui indique si on touche ou non la cible peut afficher derrière la fenêtre
(ce qui donne l'effet que le programme crash, mais il suffit d'aller la chercher à l'arrière).
- Le son de coup de feu peut ne pas fonctionner.

