Introduction

En continuant sur ma lancée astronomique du projet 1, j’ai voulu dans ce projet explorer les banques de données ouvertes de la NASA afin de réaliser un système de visualisation de données en rapport aux exoplanètes dont nous avons confirmé l’existence.

Banque de données

J’ai commencé avec la banque de données de tous les systèmes planétaires du *Nasa Exoplanet Science Institute* (<https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=PS>). Ces bases de données étant extrêmement exhaustives, j’ai eu beaucoup de difficulté à comprendre et à choisir quels paramètres utiliser. Mon idée de base était de seulement utiliser les planètes découvertes en 2023, mais j’ai rapidement manqué d’idées. J’ai finalement sélectionné quelques paramètres qui seraient selon moi intéressants à visualiser, soit la masse, le rayon, la température, l’excentricité (un ratio qui détermine à quel point la forme de la planète diffère d’un cercle) ainsi que la distance. J’ai ensuite téléchargé un fichier CSV avec ces paramètres pour toutes les planètes ayant été confirmées hors de notre système solaire.

Le premier problème que j’ai rencontré avec la base de données est que les planètes y étaient listées plusieurs fois si plusieurs publications les référenciaient. Pour pallier à cela, j’ai effectué une *loop for* qui ne copierait les informations d’une planète que si celle-ci apparaissait pour la première fois dans le tableau. De plus, les paramètres qui m’intéressaient n’étaient pas recensés pour toutes les planètes découvertes. J’ai donc choisi de n’utiliser que celle qui avaient tous les paramètres que je voulais visualiser. Pour ce faire, j’ai également utilisé une *loop for* qui ne copiait les paramètres dans un nouveau tableau que si aucune des colonnes n’était vide.

Planètes

Je me suis ensuite créé un tableau par paramètre en utilisant des opérateurs  *select.* Avec ces tableaux, j’ai créé une sphère que, avec la technique d’instanciation par GPU, j’ai répliqué en utilisant mes paramètres comme opérateurs d’instanciation. J’ai choisi d’utiliser le rayon pour la taille, en multipliant cette dernière par le facteur d’excentricité (que j’ai mis sur une échelle d’1 à 2, 1 étant une sphère) afin de me créer deux channels différents, que j’ai pu utiliser afin de créer de différentes formes plus ou moins près de la sphère parfaite. J’ai ensuite utilisé la température pour déterminer la couleur des planètes. En inversant le canal de la température, j’ai obtenu deux canaux, soit un rouge et un bleu. Plus la planète est chaude, plus la planète est rouge. J’ai ensuite créé un *noise* pour mon canal vert, qui lui a une très faible amplitude afin de créer une variation de couleur très minime. Finalement, j’ai utilisé la distance afin de placer mes planètes. J’ai ensuite utilisé la méthode de données en continu afin de voyager à travers les distances. J’ai ensuite associé cette valeur à la position x de la caméra, en plus de l’envoyer dans un TOP *text* afin d’afficher cette distance. J’ai appliqué un filtre à ma valeur afin que le mouvement soit plus fluide. J’ai également appliqué ce mouvement à une lumière qui me permettrait de mettre un effet d’ombre aux planètes. De plus, j’ai essayé d’utiliser la masse pour la position en y. Or, je n’aimais pas l’effet que cela donnait et je ne trouvais pas que ça aidait la visualisation. J’ai donc ultimement décidé de ne pas utiliser ce paramètre, et de plutôt faire vaciller mes planètes selon un *noise*.

Étoiles

J’ai terminé en créant à nouveaux des étoiles pour mon arrière-plan. Je voulais que ces dernières aide à visualiser le mouvement dans l’espace, d’une planète à l’autre. Je voulais donc qu’elles soient statiques. Cette fois-ci, j’ai utilisé à nouveau la méthode d’instanciation par GPU sur un cercle que j’ai multiplié selon trois *noises*, me créant ainsi des positions x, y et z. J’ai mis une quantité de 8000 *samples* sur ces derniers, pour un total de 8000 étoiles. J’ai ensuite utilisé un matériel *phong* afin de leur faire émettre de la lumière.

Pour toutes ces composantes, j’ai effectué un *render*, puis j’ai tout composé en utilisant des méthodes *over*.