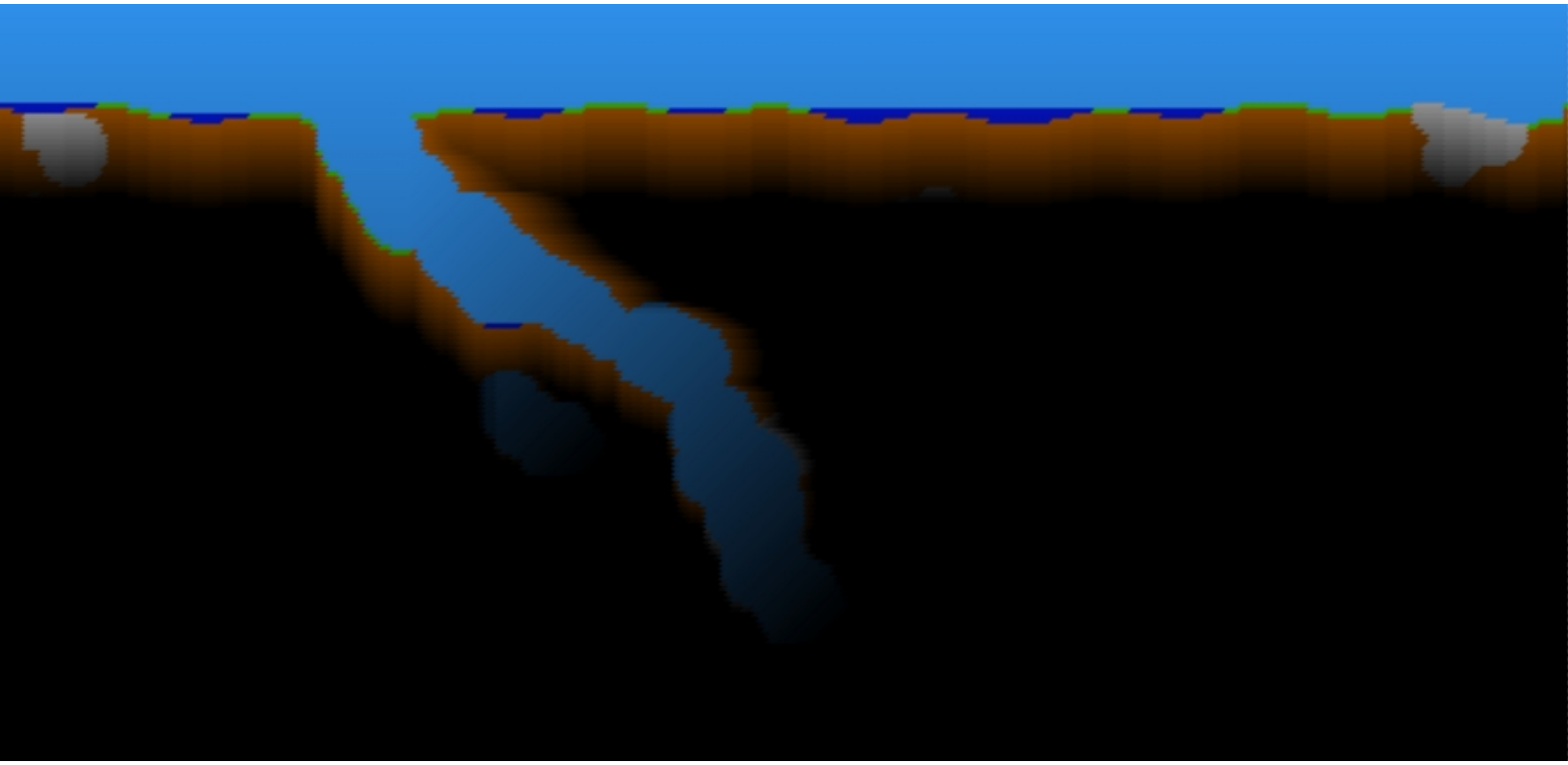
LES TROPHÉES NSI

Édition 2022

DOSSIER DE CANDIDATURE
PRÉSENTATION DU PROJET



Ce document est l'un des livrables à fournir lors du dépôt de votre projet : 4 pages maximum (hors documentation).

Pour accéder à la liste complète des éléments à fournir, consultez la page **Préparer votre participation**.

Vous avez des questions sur le concours ? Vous souhaitez des informations complémentaires pour déposer un projet ? Contactez-nous à **info@trophees-nsi.fr**.

NOM DU PROJET : GÉNÉRATION PROCÉDURALE

Site web du dossier de projet :

http://generation-procedurale.tk/

(alternative) :

https://stxrm-cc.github.io/generation-procedurale-final/

> PRÉSENTATION GÉNÉRALE :

Pour commencer ce projet est une reconstitution d'une génération procédurale en 2 dimensions (il existe des exemples maintenant culte dans l'histoire du jeu vidéo en 3 dimensions comme Minecraft ou Muck mais aussi en 2 dimensions comme Terraria). Ce projet s'inscrit dans l'ère du temps où la mode ce dirige vers des jeux vidéos à ''monde ouvert'' et où la créativité est maître, c'est une des raisons qui nous a poussés à choisir ce sujet. De plus, pour voir ce projet d'un œil plus ''scolaire'', cela nous permettais de traiter plusieurs parties du programme de Première comme

« l'algorithme des k plus proches voisins », une simulation (très simplifié) d'un fluide ou même le faite de résoudre des problèmes qui nous semblent impossible sans essayer. Enfin, pour notre développement personnel avec l'autonomie (presque complète), pour répartir les taches ou encore pour la gestion du temps. Notre objectif était de faire un monde qui ce génère automatiquement et avec divers paramètres gérant la génération.

> ORGANISATION DU TRAVAIL :

Notre équipe ce compose de 2 personnes : Dimitris Kaltsas et Nils Viollet, touts deux en classe de Première au lycée Paul Émile Victor à Champagnole. Nous nous sommes répartie le travail de manière à ce que la partie Python soit pour Nils Viollet et que toutes la partie HTML, CSS, JavaScript soit pour Dimitris Kaltsas. Pour nous partager le travail, cela était plus simple étant donnée que nous sommes dans la même chambre d'internat, mais nous nous passions le code via Git-Hub et Discord (que nous utilisions aussi pour communiquer). En dehors de l'établissement nous travaillions environs 5 heurs par semaines ce qui a été suffisant.

> LES ÉTAPES DU PROJET :

Au début nous n'avions que l'idée, nous l'avons développé pour avoir un monde fait de terre et de pierre (il n'y avais pas de ciel) puis, nous avons ajoutés des grottes et le ciel, ensuite nous avons rajoutés de l'eau (ce qui fut plutôt compliqué) et de l'herbe. Quand nous sommes arrivés là, nous nous sommes dit que nous pouvions nous arrêtés et donc finir. Mais nous avons eu alors une idée : Pourquoi ne pas ajoutés des ombres ? C'est donc ce que nous avons fait, nous avons tout d'abord utilisé un algorithme qui gérerais chaque source de lumière et son impact autour d'elle. Mais

cette méthode n'était pas très performante et donc nous avons opté pour l'algorithme des plus proche voisins. En parallèle nous avons fait le dossier de projet avec HTML/CSS/JavaScript et nous avons commencer par l'accueil et, en copiant le model de celui-ci, nous avons fait les 4 autres pages. Nous avions dès le départ l'envie de mettre un interpréteur Python dans notre page HTML et c'est donc ce que nous avons fait.

> FONCTIONNEMENT ET OPÉRATIONNALITÉ :

Dans notre projet, nous avons fait tout ce que nous voulions faire, c'est à dire une génération de grottes, une génération du ciel, une simulation de l'eau et de la lumière, ainsi que l'ajout d'herbe. Nous avons remplie nos objectifs, et nous avons donc fini le projet. Mais cela n'a pas été sans péripétie, pour trouver les bugs, nous nous pausions tout d'abord une question : Quels sont les cas spéciaux et es-ce que le programme les traites ? Malheureusement, nous ne sommes pas infaillibles… Mais à chaque découverte d'un bug, nous l'avons identifier (en le reproduisant plusieurs fois pour en définir la cause) et nous l'avons réglé. Pour donner des exemples de bug que nous avons rencontré, il y aurais la génération du ciel de l'autre coté : le ciel, au lieu de ce générer en haut, ce générait en bas… Cela nous donnais une île flottante sans limite… Pour résoudre ce problème, nous avons donc inversé le monde (au lieu de commencer de a gauche et en haut nous avons commencer en bas à droite). Cela nous a demander de réécrire l'entier-té du code… En autre problème nous avons eu la simulation de la lumière : au lieu de décroître avec la distance, celle-ci augmentait de manière exponentiel. Pour réglé ce problème cous avons changé la façon de générer celle-ci en passant de nombre entier à chaîne de caractère mais, suite à cela nous avons changé de système de simulation (dû à la lenteur du précédent). Ensuite nous avons rencontré un bug qui faisait que l'image n'était pas bien dessinée : l'espace entre les pixel aurais du être nul mais il y en avais bien un… Ce bug était dû à une erreur d'arrondie, mais cela nous a quand même pris 1 à 2 heures avant de trouver où était le problème. Enfin nous avons rencontré un problème qui n'était pas un bug : comment avoir la position des pixels dans un cercle à partir d'un rayon ? Pour résoudre ce problème, nous avons choisi une approche sur feuille où nous essayions de trouver une solution sans programmer… Et nous sommes arrivé avec cette solution : le programme part d'un point et tourne autour en carrés de plus en plus grand, en vérifiant la distance avec le centre pour chaque pixel, et si la distance entre ce pixel et le centre est plus petite que le rayon, alors on l'ajoute, sinon on ne le prend pas.

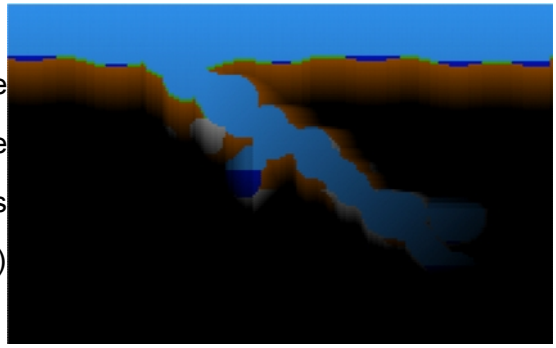
> OUVERTURE :

Il y a plein de manière différente pour amélioré notre projet, on pourrais par exemple ajoutés différent type de bloc (du bois, du fer, du sable…) ou différent type de liquides (lave, acide…) mais on peut aussi ajouter la génération de structures comme des maisons ou des mines. On peut aussi amélioré la simulation de l'ombre pour ajouter des sources de lumières comme des torches ou du feu et pour le rendu visuel on pourrait ajouter des textures aux blocs pour les rendre plus beau et plus reconnaissable. Enfin on pourrait aller plus loin et ajouter un personnage se déplacent dans le monde. Si nous devions diffusé notre projet nous utiliserions de la pub sur ds plateforme comme YouTube avec comme public des jeunes développeurs mais aussi des gens voulant apprendre de nouvelles choses et ceux qui pourraient s'orienter vers l'informatique (mais soyons sérieux une minute, personne n'utiliserait ce programme en sachant qu'il y a des méthodes plus performantes et plus professionnelles). Enfin, si nous devrions le refaire, nous ajouterions quelque fonctionnalité cité ci-dessus et nous n'utiliserions pas le module « turtle » voire même nous ne le ferions pas en Python (peut-être en C++ ou en Java).

DOCUMENTATION

• *Illustrations, captures d'écran, etc*

Quel sont les bibliotheques utilisés ?



Dans ce projet nous avons utilisé 4 modules : le module « math », le module « random », le module « time » et le module « turtle ». Notez que tout ces modules n'ont (en fonction de votre version de Python) pas besoin d'être installer.

Comment lancer le programme et le dossier de projet?

Pour lancer le programme de notre projet, il vous faudra ouvrir le fichier « generation-procedural-final » (ou le télécharger directement dans la page «Le Projet» du site web de notre dossier de projet) avec votre interpréteur Python puis lancer l'exécution depuis celui ci, ensuite toutes les actions à faire vous seront spécifié.

- Si vous avez déjà lancé le programme -

- notez que la touche « r » active des actions dans la consoles et non des actions -
- visibles dans la fenêtre de turtle -

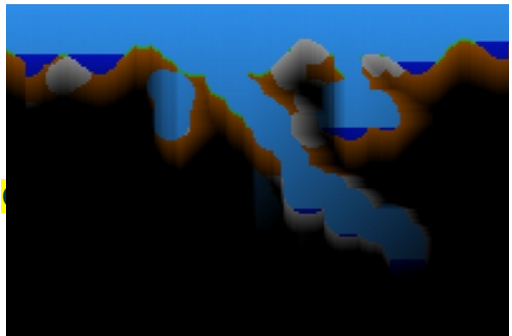
Pour le dossier de projet, il vous faudra lancer le fichier « index.html » qui vous lancera automatiquement une page web (la qualité du rendu se fera en fonction de votre navigateur web). Dans celui-ci la résolution de votre écran aura un impact (que nous avons essayé de minimiser). Le menu en haut vous permettra de naviguer entre différente partie du site.

Pour ce projet nous nous avons utilisé une structure qui nous a parus simple : une seule fonction pour les commander toutes… En effet, il y a une seule fonction principale qui gère toutes les autres, c'est elle qui ‘‘dit'' quelle action ce fait et quand. On pourrais la représenter comme un arbre avec les branches et le tronc principale. Comme dit précédemment c'est le langage interprété Python que nous avons choisi. Nous l'avons choisi car il est simple malgré sa lenteur et l'impossibilité de faire des interfaces acceptable.

Pour le déroulement des étapes, c'est très simple : on pourrait presque le dessiner littéralement sur une feuille avec le même procédé que le programme. Pour commencer, le programme « dessine » (il fait une liste composé uniquement de 1 et de 2) avec deux tiers de pierre et le reste de terre en ajoutant de la terre dans la partie de la pierre et inversement. Ensuite le programme prend un stylo avec une taille aléatoire et « dessine », en changeant la taille du stylo régulièrement, des zigzagues dont la trajectoire est, elle aussi, aléatoire. Puis, le programme « dessine » le ciel en utilisant des fonctions « sin » et « cos » pour donner des plaines ou des montagnes. Après cela le programme fait une simulation sommaire d'une source d'eau en simulant le mouvement de chaque particule en fonction des autres. Pour finir le programme ajoute de l'herbe sur la terre en surface qui n'est pas sous l'eau. Enfin on simule de la lumière avec une résistance de 10 pour l'air, de 50 pour l'eau et de 100 pour le reste (c'est à dire les blocs physique). Pour ce faire on part de tout en haut et on descend en regardant les voisins les plus proches pour définir la luminosité du bloc. Et voila c'est tout pour la génération, mais maintenant il faut afficher le monde… Pour cela nous avons utilisé le module turtle qui permet de dessiner sur l'écran. Donc chaque bloc a une luminosité (défini lors de la simulation de l'ombre) qui est afficher à l'écran.

Vu que le programme fait tout dans un ordre défini, il est donc possible de sauter des étapes. Il est naturellement impossible de sauter toutes les étapes, donc voici celles qui ne sont pas nécessaire :

- la génération des grottes ;
- l'apparition de l'eau ;



- et l’ajout de l’herbe.