



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 : Spicy Journey – 香り旅 – Voyage Épicé

> PRÉSENTATION GÉNÉRALE :

Spicy Journey est un jeu d'exploration dans un monde généré de manière procédurale et rempli de forêts luxuriantes. Avec son point de vue isométrique et son style pixel art, c'est un jeu rétro-visuel qui ne manquera pas de vous captiver. Spicy Journey offre des possibilités infinies d'exploration et de découverte. Il n'y a pas deux jeux identiques, vous pouvez donc y revenir encore et encore pour une nouvelle expérience. Si vous cherchez un jeu qui vous aide à vous détendre et à vous relaxer, ne cherchez pas plus loin que Spicy Journey. Avec ses magnifiques forêts, son charmant style pixel art et ses possibilités d'exploration infinies, c'est le jeu idéal pour tous ceux qui cherchent à s'évader dans un monde paisible et immersif.

L'histoire :

L'histoire se déroule dans un monde fantastique rempli de magie et de mystère. Le personnage que vous contrôlez est une jeune femme japonaise nommée Hana, née dans un petit village isolé niché dans une forêt luxuriante. Hana est une cuisinière passionnée et a hérité de sa grand-mère un livre de recettes traditionnelles japonaises. Les ingrédients dont elle a besoin pour cuisiner ces plats sont dispersés dans le monde entier. Elle décide de partir à l'aventure pour les collecter et les cuisiner afin de perpétuer les traditions familiales et de rendre hommage à sa grand-mère. Elle se lance dans un incroyable voyage à travers la nature sauvage, découvrant des secrets mystiques sur le monde qui l'entoure. En chemin, Hana rencontre des personnages fascinants qui l'aident dans sa quête, comme des marchands qui lui vendent des ingrédients utiles pour ses recettes (pas encore implémenté).

Le monde :

La projection isométrique est une méthode de représentation visuelle d'objets tridimensionnels en deux dimensions dans les dessins techniques et d'ingénierie. Il s'agit d'une projection axonométrique dans laquelle les trois axes de coordonnées apparaissent également raccourcis et l'angle entre deux d'entre eux est de 120 degrés.

La conception isométrique est une technique de conception qui a résisté à l'épreuve du temps, puisqu'elle est utilisée depuis plusieurs décennies. Utilisée pour la première fois dans les jeux vidéo en 1981 avec les débuts de Zaxxon de Sega, elle a rapidement gagné en popularité grâce à sa capacité à créer des environnements tridimensionnels sans avoir recours à des logiciels de modélisation 3D complexes. Le nom de la technique, "isométrique", vient du fait que toutes les lignes du dessin sont tracées à un angle de 30 degrés, créant ainsi une illusion de profondeur et de perspective.

Cette technique s'est avérée incroyablement polyvalente et est encore largement utilisée aujourd'hui dans divers domaines de la conception, notamment la conception de jeux vidéo et la conception graphique. La capacité de créer facilement des environnements complexes et visuellement attrayants a fait de la conception isométrique l'une des préférées des concepteurs du monde entier. En outre, cette technique a été adaptée et affinée au fil du temps, ce qui a conduit à la création de nombreuses variantes, chacune présentant des caractéristiques et des avantages uniques.

Le bruit de Perlin est un type de générateur de texture procédural inventé par Ken Perlin en 1985. Il s'agit d'une fonction qui génère une apparence pseudo-aléatoire, ce qui signifie que le résultat semble aléatoire, mais qu'il est en fait déterministe et peut être facilement contrôlé en modifiant les paramètres d'entrée ou la "graine".

Le bruit de Perlin est couramment utilisé pour générer divers types de textures, telles que des nuages, des terrains et des marbres. Il est également utilisé dans de nombreux jeux vidéo pour créer des paysages réalistes et dynamiques, ainsi que dans l'animation et les effets spéciaux.

cinématographiques. L'algorithme est basé sur une grille de vecteurs de gradient qui sont utilisés pour interpoler entre les points de treillis afin de produire le résultat final.

Nos inspirations :

Ghibli :

Pour créer Spicy Journey, nous nous sommes inspirés des films du Studio Ghibli, connus depuis toujours pour leur univers enchanteur et leur atmosphère magique. Pour ce faire, ils ont voulu respecter plusieurs règles essentielles qui caractérisent les films de Ghibli comme la nature, le lien familial (avec l'histoire du jeu) et l'attention au détail. Voici quelques liens qui nous ont aidé :

- [The Immersive Realism of Studio Ghibli](#)
- [Hayao Miyazaki | The Mind of a Master](#)
- [Le Réalisme Immersif du Studio Ghibli](#)

Minecraft :

Dans la première version de notre générateur de cartes, nous nous demandions comment nous pouvions créer un monde aussi réaliste. C'est alors que nous nous sommes inspirés de Minecraft, un jeu bac à sable connu pour sa liberté de jeu et son monde infini généré de manière procédurale. Cette génération est une caractéristique clé du jeu qui permet aux joueurs d'explorer un monde qui ne se termine jamais et qui en fait donc un jeu sans limites. Après de nombreuses recherches, nous avons trouvé une vidéo qui nous a tout expliqué : [Minecraft terrain generation in a nutshell](#)

Et d'autres à retrouver sur le [Wiki du Github](#) !

> ORGANISATION DU TRAVAIL :

Il y a quelques mois, deux passionnés de développement - [@Zecyl](#) (alias Ulysses) et [@ImSumire](#) (alias Galaad) - ont décidé de travailler ensemble sur un projet de jeu. Leur objectif était de créer un jeu d'exploration isométrique qui serait à la fois beau et relaxant à jouer.

En travaillant en tandem, les deux développeurs ont réussi à diviser leur travail de manière efficace. Zecyl s'est chargé de la mise en place du **multijoueur**, ce qui a impliqué l'installation du **Raspberry Pi** en tant que **serveur**, la gestion des **packets** et la **synchronisation** des données entre les différents joueurs. Il a également travaillé sur les premières versions de l'algorithme de **génération du monde**.

De son côté, Sumire a travaillé sur la création du **moteur isométrique** et du monde, s'est occupé de la gestion globale de l'**interface utilisateur**, des **déplacements** des personnages et des **musiques** et **ambiances** du jeu. Tout en travaillant sur le projet, Sumire a également veillé à l'**optimisation** pour que le jeu fonctionne rapidement et sans problème.

Parmi les logiciels les plus populaires utilisés figurent [Github](#), [Aseprite](#) et [Visual Studio Code](#). **GitHub** est utilisé pour héberger le code source, permettant ainsi de travailler sur le même projet en même temps. **Aseprite** est utilisé pour créer des sprites pour le jeu, tandis que **Visual Studio Code** est l'éditeur de code de choix pour le développement.

En plus de ces outils, l'équipe utilise également [itch.io](#) pour trouver des sprites, ainsi que [freesound.org](#) pour trouver des effets sonores et des musiques. Pour créer une boucle propre pour les pas du joueur, nous avons utilisé [drumbot.com](#)

[Discord](#) est également utilisé comme moyen de communication principal, nous permettant de rester en contact et de partager nos avancés et nos idées. Nous faisons des réunions tous les jours (au lycée et/ou en appel) et nous travaillons au lycée (CDI) et surtout chez nous.

> PRÉSENTATION DES ÉTAPES DU PROJET :

L'histoire a commencée l'année dernière où il fallait présenter un projet, nous avons donc choisie de faire une application de visualisation de cartes générées procéduralement, nous avons utilisé [Tkinter](#) (vu pendant l'année de première) pour l'affichage et le [Simplex Noise](#) (qui avait été entièrement refait pour être compatible avec le projet). Voici le lien du commit Github de l'ancienne version : [ici :D](#)

Quand nous avons appris pour le concours nous nous sommes mis dessus. Le projet était à l'origine **Atlas** et devait être une simple visualisation d'un monde. Cependant, au fur et à mesure, lorsque l'affichage a été fait et que tout marchait, nous avons décidé d'aller plus loin en implémentant un joueur et les mouvements, et petit à petit de nouvelles idées ont commencées à émerger et nous les implémentions. Nous avons donc renommé le jeu **Spicy Journey** qui veut littéralement dire **Voyage Épicé**.

Il y a un mois nous sommes arrivé à savoir précisément ce à quoi le projet ressemblerait ! Finalement, avec beaucoup d'efforts, de détermination et d'organisation, nous avons fini le projet. C'est donc avec un travail acharné dont nous sommes fiers. Leur voyage a été long et difficile, mais cela en valait la peine. Le jeu est un bel exemple de ce qu'il est possible de réaliser avec du travail acharné et de la persévérance.

> FONCTIONNEMENT ET OPÉRATIONNALITÉ :

Malgré que le jeu comporte les parties les plus importantes, nous n'avons pas terminé le projet à 100%, il nous reste seulement le fait de pouvoir cuisiner et quelques autre idées. Nous sommes en train d'implémenter le multijoueur et le système de cuisine.

Notre projet a été rigoureusement testé pour **vérifier l'absence de bugs**. Nous avons utilisé le framework de test **Pytest** pour exécuter des tests automatisés et manuels pour s'assurer que chaque partie du code fonctionne comme prévu. Voici leur site : <https://docs.pytest.org/en/7.3.x/>

En plus des tests, nous avons également fait attention à la **qualité du code** et la **lisibilité** en utilisant le **formateur Black** pour garantir que le code était clair et facile à lire. Black a automatiquement formaté notre code en utilisant les conventions recommandées par la communauté Python et est basé sur **PEP8**. Voici le lien du projet Black : <https://github.com/psf/black>

Nous avons également utilisé **Pylint** pour vérifier la **qualité de notre code**. Pylint est un outil de linting qui permet de détecter les erreurs de syntaxe, les problèmes de performance, les erreurs de style et les erreurs logiques dans le code Python. En utilisant Pylint, nous avons pu détecter des erreurs potentielles dans notre code et de rendre le code plus sain et plus facile à comprendre. Voici le lien du projet Pylint : <https://github.com/pylint-dev/pylint>

De plus, des optimisations ont été faites, Sumire a beaucoup utilisé de techniques pour éviter le surplus de calculs inutiles ou répétitifs. Afin d'optimisé, nous avons lu beaucoup de documents et utilisé des bibliothèques comme **Numba** (traduit les fonctions en code machine optimisé au moment de l'exécution) ou **Threading** et en utilisant du cache/de la mémoïsation.

En fin de compte, notre engagement envers les tests rigoureux et la qualité du code a permis d'assurer la stabilité et la fiabilité de notre projet. Nous sommes convaincus que notre projet est prêt pour une utilisation saine et une lecture agréable, grâce à la combinaison de Pytest, Black et Pylint.

Afin de faciliter l'utilisation du projet nous avons mis les bases de l'installation du projet sur le [README.md](#) du [Github](#). Pour commencer il suffit de télécharger le code, d'exécuter ces commandes :

```
~$ pip install -r requirements.txt
~$ pip install --upgrade pip
```

Et de démarrer le fichier [main.py](#) avec Python; sinon vous pouvez télécharger la dernière version de l'exécutable dans la page Releases.

Nous avons eu quelques difficultés au début pour bien gérer les coordonnées, nous avons eu des difficultés à bien trier les sprites pour que le joueur puisse passer derrière et devant les éléments (*juste une mauvais histoire d'offset venu de je ne sais où*). Finalement, Pylint nous a pris du temps mais est maintenant valide et ça rend le code vraiment mieux !

> OUVERTURE :

Pendant le développement, du jeu par notre grande organisation, nous avons tenu une liste de choses à faire (la fameuse To-do list) que nous avons mis à jour tellement souvent... Actuellement, nous nous concentrons sur l'ajout de fonctionnalités telles que la possibilité de **cuisiner** des plats délicieux comme dis ci-dessus, mais aussi d'introduire une **génération plus naturelle** avec des biomes variés, des montagnes majestueuses, des rivières sinueuses et des structures abandonnées pour une immersion totale. L'ajout de la **pêche** pour passer du bon temps aurait été sympathique, puis le **cycle jour/nuit** pour rajouter du réalisme et une continuité dans le monde et l'implémentation d'un **marchant ambulant** pour rajouter plus d'ingrédients et une économie. Nous avons également envisagé de permettre aux joueurs de **communiquer entre eux en jeu**, ce qui ferait de notre jeu une expérience communautaire unique. Mais aussi rajouter de la vie comme l'**implémentation d'animaux**.

Notre but n'est pas de faire le meilleur jeu de tous les temps (*car le dino de Google existe déjà*) mais plutôt de faire un jeu qui détend et est sympathique à jouer. Nous avons fait un peu de pub sur un serveur d'amis développeurs et nous attendons des demandes d'ajouts, nous sommes donc ouvert à d'autres propositions.

Si nous devions le refaire, nous aurions pris en compte plusieurs éléments qui n'ont pas été considérés auparavant. Tout d'abord, nous aurions commencé plus tôt pour nous donner suffisamment de temps pour réaliser toutes les fonctionnalités que nous avons prévues. Nous aurions également mis en place une organisation encore plus rigoureuse pour optimiser notre temps.

De plus, nous aurions veillé à intégrer des fonctionnalités supplémentaires qui étaient initialement prévues, mais que nous n'avons pas eu le temps de développer. Cela aurait pu inclure une meilleure interface utilisateur, des options de personnalisation pour les utilisateurs, une intégration intuitive du multijoueur avec le hosting sur la machine du joueur. Et finalement nous aurions pu rendre le projet plus professionnel avec un [site web](#) utilisant [Django](#) (qui est en développement) et une meilleure installation avec un environnement grâce à [Poetry](#).

DOCUMENTATION

Nous avons mis en place un README.md sur Github. Mais voici les étapes à suivre :

1. Installez la dernière version de Python : <https://www.python.org/downloads/>
Versions vérifiées : 2.7, 3.8, 3.9 et 3.10
2. Vérifiez que vous avez pip
3. Téléchargez le code sur Github : <https://github.com/ImSumire/Spicy-Journey>
Le bouton vert avec « Code » > Download ZIP
4. Ouvrez le terminal à partir du fichier racine (avec `cd` si vous n'arrivez pas à l'ouvrir à partir du dossier actuel)
5. Tapez : `pip install --upgrade pip`
6. Tapez : `pip install -r requirements.txt`
7. Tapez : `python3 main.py`
Ou démarrez avec votre IDE

Le jeu prend en général une à deux secondes à s'ouvrir, c'est dû aux chargements des données (images, musiques, sons, librairies, fichiers de data).

Librairie utilisées :

- `sys` : Pour les paramètres et fonctions propres aux systèmes d'exploitation
- `os` : Afin de récupérer des informations sur le systèmes d'exploitation
- `json` : Lire les fichiers de données avec comme extension le .json
- `time` : Récupérer les données sur le temps actuel
- `pygame` : Afficher le jeu et quelques autres outils comme les vecteurs
- `random` : Générer des nombres aléatoires
- `noise` : Pour le Simplex Noise, qui est extrêmement rapide
- `subprocess` : Pour lancer des commandes dans le terminal
- `math` : Outils de base de mathématiques rapides
- `threading` : Pour gérer plusieurs processus en même temps
- `numba` : Compilateur Just-In-Time qui optimise le code
- `socket` : Interface de mise en réseau de bas niveau
- `pickle` : Sérialisation d'objets Python
- `traceback` : Afficher et récupérer une trace de la pile

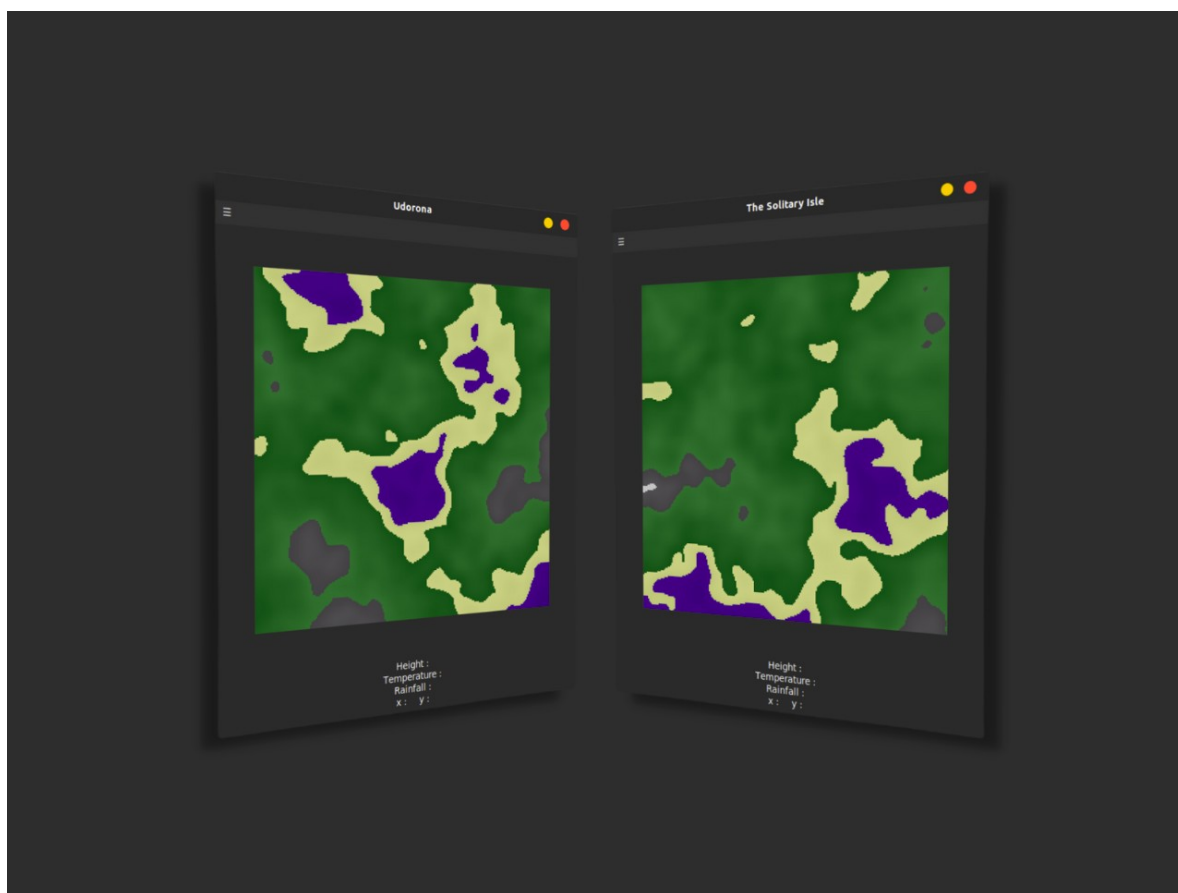
Matériel : [Raspberry Pi](#)

Formats de stockage des données :

- **JSON** : JavaScript Object Notation est un format de données textuel dérivé de la notation des objets du langage JavaScript.
- **CSV** : Comma-separated values est un format texte ouvert représentant des données tabulaires sous forme de valeurs séparées par des virgules.
- **LANG** : Lang est un format texte de données de traduction prenant un id et une valeur séparés d'un signe d'égalité.

Images :

IMAGES



À quoi ressemblerait l'application de génération de carte



Le menu de Spicy Journey (香り旅 signifie littéralement « Voyage d'odeurs agréables »)



Screenshot du gameplay, c'est bo :)



Screenshot du gameplay avec le visuel sur l'inventaire (haut-droite)



Logo du jeu fait par Sumire