SUPERNOVA

Paul Leflon - Aryelle Pocholle - Chloé Xu - Lévina Sun

EN QUOI CONSISTE LE PROJET

SUPERNOVA est un jeu inspiré du T. Rex Game de Google Chrome. On y trouve un joueur qui doit se déplacer verticalement pour éviter des obstacles avançant de plus en plus vite et attraper des pièces. Par défaut, ce joueur est un astronaute et les obstacles des météorites, mais il est possible d'acheter des skins avec les pièces amassées. L'objectif est simple : survivre le plus longtemps possible pour avoir le meilleur score et attraper un maximum de pièces.

MODE D'EMPLOI

Arrivé sur le menu principal, 3 choix s'offrent à nous : on peut lancer une partie, consulter l'inventaire où on trouve nos skins possédés, les skins à acheter et le nombre de pièces que l'on possède, ou voir les statistiques de nos parties.

Trois touches suffisent pour jouer :

- Les flèches du clavier (haut et bas) pour déplacer le joueur
- LShift pour se déplacer plus rapidement

CHOIX TECHNIQUES

Dans un premier temps, nous avons créé un prototype en Python avec Pygame. Ça a été laborieux pour un résultat décevant. Nous avons donc décidé d'abandonner cette bibliothèque pour quelque chose de plus simple à utiliser. Nous voulions un jeu final complet dans son interface et n'avons pas voulu prendre le risque de prendre trop de temps à réaliser cela avec Pygame. Le plus simple a alors été de s'orienter vers les technologies web, HTML/CSS/JavaScript. Idéal pour créer une interface et pour le jeu, nous n'avions pas de grandes ambitions à la base pour les visuels donc de simples éléments HTML qui bougent suffisaient largement. Le fait de pouvoir utiliser des fichiers .gif, ce que ne propose pas pygame, était aussi un avantage. Les performances ont été une inquiétude mais finalement le jeu s'est montré fluide même sur des ordinateurs assez lents à la base. Nous avons également choisi de ne pas nous aider

d'un quelconque libraire externe, simple fierté d'avoir écrit l'intégralité du code nous-même. Il est vrai que cela aurait pu améliorer les performances ou la qualité du code mais les choses sont déjà largement satisfaisantes ainsi.

LE CODE

Le projet est écrit en HTML, CSS et JavaScript. Chaque langage a quelques règles de style pour conserver un code uniforme entre les différents fichiers.

En HTML

- *Kebab-case* pour les noms de classes et d'attributs
- Guillemets doubles

En CSS

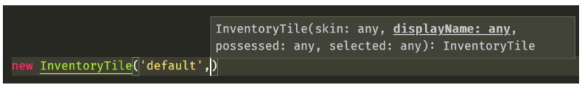
- Chaque classe JavaScript affichable (*Sprite, Coin, Obstacle, Player*) doit avoir un fichier CSS correspondant
- Les règles sont ordonnées selon ces consignes
- Les blocs sont tous séparés d'une ligne vide

En JavaScript

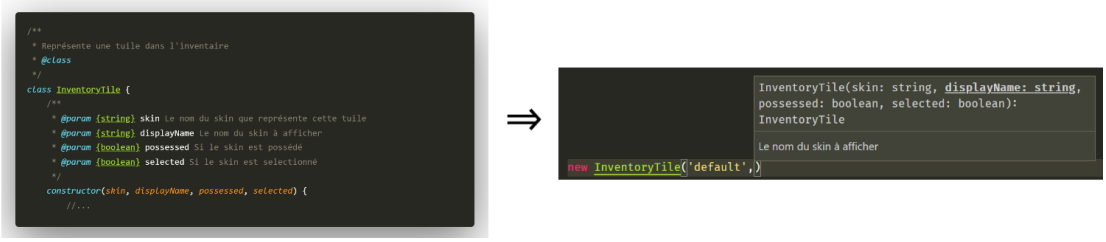
- On utilise le point-virgule en fin de ligne
- On utilise des guillemets simples pour les strings (ou des apostrophes inversées pour les template strings)
- On n'utilise pas d'accolades autour des blocs conditionnels d'une seule ligne
- Chaque méthode de classe est espacée d'une ligne vide d'une autre et est documentée avec **JSDoc**

  JSDoc est un outil similaire au docstring en Python. Il permet en JavaScript de spécifier les types, les descriptions, (et bien plus) de nos fonctions, méthodes, propriétés, etc… De plus, l'éditeur de code utilisé (Visual Studio Code) intègre parfaitement ces descriptions et types dans son interface et permet une lecture plus rapide du code.

  SANS JSDOC



  AVEC JSDOC



  Le reste est noté en commentaire dans le code

CE QUI DEÇOIT

Le jeu final est satisfaisant mais il y a tout de même plusieurs points décevants.

Premièrement, les graphismes. Les dessins sont très beaux, mais on aurait pu faire des animations/effets sonores pour mieux les lier entre eux et fluidifier l'expérience. Cela se ressent notamment sur les pièces.

Une idée que nous avions eue pour rendre le jeu plus intéressant était d'ajouter des mini-jeux dans le jeu lui-même. À la manière de Mario Party ou les items de Mario Kart, l'idée était d'intégrer des éléments semblables à des pièces qui lanceraient un rapide mini-jeu offrant en cas de victoire un certain nombre de pièces au joueur. Malheureusement, vu tout ce qu'il y avait déjà à faire, nous n'avons pas réalisé cette idée.

Aussi nous aurions pu rendre le jeu jouable sur mobile. Le gameplay aurait été moins confortable mais parfaitement faisable. Nous n'avons cependant pas prévu ça à la base.

Enfin, le plus gros vice du jeu, les boîtes de collision (*Hitbox*) de nos éléments graphiques (*Sprite)*. Les *Hitbox* dans le jeu sont des rectangles. Lorsque ces rectangles se rencontrent, on détecte la collision entre les deux *Sprites*. Cela sert pour collecter des pièces et détecter la défaite quand le joueur rencontre un obstacle. Les assets que nous avons ne sont pas des rectangles mais des formes complexes. Ainsi, les *hitbox* ne décrivent pas parfaitement nos *Sprites* et des collisions peuvent être détectées alors que l'on voit clairement que les deux éléments ne se sont pas touchés.

Affichez les hitbox en jeu en mettant `debug: true` *dans l'appel* `super` *du constructeur du Sprite*

On le voit ici, les hitbox ne sont pas précises. Si le joueur monte et qu'un obstacle se trouvait au-dessus, on pourrait avoir une collision à cause de l'espace vide au-dessus de la guitare. Nous avons fait au mieux pour équilibrer les collisions mais il reste encore souvent des cas où une fausse collision survient et frustre. Un système de hitbox plus complet avait été initié mais nous n'avons pas eu le temps de le faire fonctionner assez bien pour qu'il vaille le coup et avons donc conservé ces *Hitbox* simples.

Tout ceci aurait pu être fait/amélioré, nous n'avons simplement pas eu le temps d'apprendre les techniques nécessaires et de les mettre en œuvre.

SOURCES · IMAGES

- Dessins : Lévina
- Explosion : Gifer
- Icônes : Font Awesome
    - Volume activé
    - Volume désactivé
    - Retour

SOURCES · POLICES

- NovaMono
- Wicked Mouse
- Komika Axis

SOURCES · MUSIQUES

- Musique du menu : *Banque de musiques libres dont on a perdu le nom*
- Musique en jeu : Andrew Huang – Drum & Bass
- Musique post-partie : Mighty Switch Force! OST - Tally Screen

SOURCES · EFFETS SONORES

- Explosion : Paul (Beepbox)
- Survol d'éléments du menu principal : Récupéré depuis le jeu osu!