Rapport SNAKE

Clément JANNAIRE

Clémence DUCREUX

Sommaire :

1. Présentation…………………………………………… 3
2. Fonctionnalité………………………………………….4
3. Découpage……………………………………………….6
4. Serpent…………………………………………………….8
5. Conclusion personnelle…………………………….10
6. Présentation

Le jeu du Snake, également connu sous le nom de Serpent, est un jeu classique d'arcade

Le joueur contrôle un serpent qui se déplace à l'intérieur d'une zone de jeu, souvent représentée par un espace rectangulaire. L'objectif principal est de faire grandir le serpent en mangeant des objets, généralement représentés par des points ou des fruits, dispersés aléatoirement sur l'écran. Au fur et à mesure que le serpent consomme ces objets, sa longueur augmente.

Cependant, le défi réside dans le fait que le serpent ne doit pas entrer en collision avec les bords de l'écran ni avec son propre corps en croissance. Si cela se produit, le joueur perd la partie. Le jeu devient progressivement plus difficile car le serpent devient plus long, augmentant ainsi le risque de collision.

Le Snake a été l'un des premiers jeux vidéo populaires et reste un classique apprécié par de nombreuses générations de joueurs

Une image contenant capture d’écran, Rectangle, texte

Description générée automatiquementUne image contenant capture d’écran, diagramme, Rectangle, texte

Description générée automatiquement

1. Fonctionnalité

Timer : Une des fonctionnalités est le timer, pour concevoir le timeron définit un cycle 1000000UL parce que dans une 1seconde il y a 1 000 000 microsecondes, on a fait une boucle infinie et pour trouver le temps en secondes on a fait le temps en microsecondes –le temps du start tout diviser par le cycle c’est-à-dire1 000 000, pour afficher les minutes on a divisé les secondes par 60.

(Nous avons deux versions du timer car nous avons un problème a l’affichage que l’on n’a pas réusis à résoudre, donc nous avons fait un autre timer avec time.h pour que l’affichage soit plus sympathique avec un timer)

Une image contenant texte, Police, Graphique, logo

Description générée automatiquement

Les Pomme (pèche) : La fonctionnalité des pommes permet au serpent de se nourrir et de grandir lorsqu'il atteint leur position sur le terrain. À chaque mouvement du serpent, la collision avec une pomme est détectée, augmentant ainsi sa taille. Après avoir mangé une pomme, sa position est réinitialisée à une nouvelle position aléatoire, assurant une apparition continue d'objets à consommer. Cette mécanique non seulement ajoute un aspect de croissance dynamique au serpent, mais crée également un défi stratégique pour le joueur qui doit chercher et atteindre les pommes pour maintenir la croissance du serpent dans le jeu.

Une image contenant Caractère coloré, vert, rose, dessin humoristique

Description générée automatiquement

PixelNoir : La fonctionnalité des pixels noirs consiste à générer aléatoirement des blocs sur la zone de jeu. Lorsque le serpent entre en collision avec l'un de ces pixels noirs, le jeu se termine, et le joueur perd la partie. La collision est détectée en vérifiant les coordonnées du serpent par rapport aux coordonnées des pixels noirs. Si une collision est détectée, le jeu s'arrête, indiquant ainsi la fin de la partie.

Une image contenant vert, Rectangle, Caractère coloré, capture d’écran

Description générée automatiquement

Score : La fonctionnalité score est afficher en bas à droite de l’écran. Le score dans le jeu est incrémenté chaque fois que le serpent mange une pomme, ce qui augmente la longueur du serpent. Chaque pomme mangée ajoute un nombre fixe de points au score. L'objectif du joueur est généralement d'obtenir un score aussi élevé que possible avant de rencontrer une condition de fin de jeu, telle que la collision avec un pixel noir ou avec le corps du serpent lui-même ou encore avec le bord de la zone de jeu.

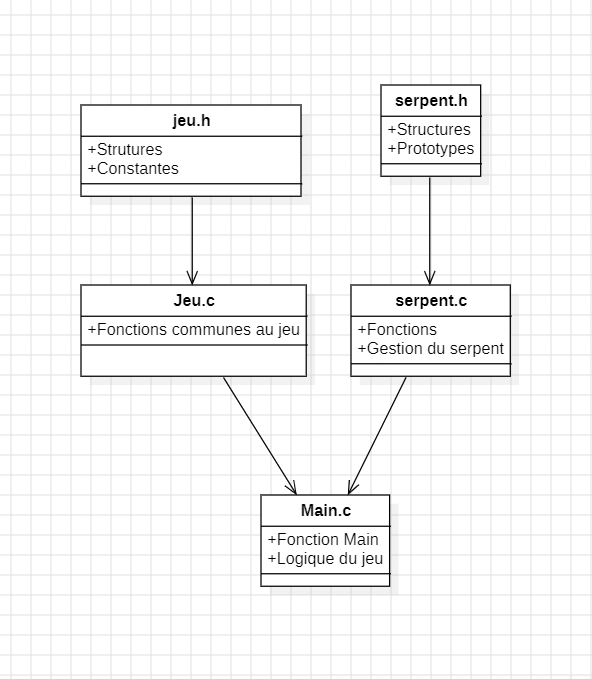
Une image contenant texte, Police, Graphique, noir

Description générée automatiquement

Le mode pause est implémenté dans le jeu en utilisant une fonction dédiée appelée Pause(). Lorsque la barre d'espace est pressée, le jeu se met en pause, affichant un écran avec le message "Pause" et les instructions pour reprendre en appuyant à nouveau sur la barre d'espace. Pendant la pause, les positions actuelles des éléments tels que les pommes et les pixels noirs sont sauvegardées. En reprenant le jeu, ces positions sont restaurées, permettant au joueur de reprendre là où il s'était arrêté. (Nous avons quelque soucis avec la sauvegarde des postion des pixels et des pommes et du serpent)

1. Découpage

Malheureusement nous avons eu quelque souci avec le découpage du code. Car nous avons utiliser une mauvaise technique en codant tout dans un seul fichier puis le diviser à la fin. Nous aurions dû organiser le code avant de commencer à développer. Nous ferrions mieux la prochaine fois dans le découpage. Malgré tout en théorisé cela devrait donner ceci :



1. Serpent :

La fonction Update\_Serpent est conçue pour mettre à jour la position du serpent.

Affichage du Fond : La fonction affiche un élément visuel représentant le fond à la position du dernier segment du serpent. Cela pourrait être interprété comme un effet visuel de la traînée du serpent, ici la trainé sera de la même couleur que le fond pour que celle-ci ne d’affiche pas.

Affichage du Serpent : Elle affiche également le sprite du serpent à la position actuelle de sa tête, ce qui représente graphiquement la tête du serpent.

Mise à Jour des Positions : En utilisant une boucle, la fonction met à jour la position de chaque segment du corps du serpent en déplaçant chaque segment vers la position du segment précédent. Cela crée l'illusion du serpent se déplaçant.

Sauvegarde des Anciennes Positions : Les anciennes positions du serpent sont ensuite mises à jour pour être utilisées lors de la prochaine mise à jour. Cela garantit que le serpent conserve sa forme et son mouvement correctement. En résumé, la fonction s'occupe de l'affichage visuel du serpent en ajustant les positions de ses segments.

Gestion du Mouvement : La fonction commence par vérifier la valeur de la variable direction pour déterminer dans quelle direction le serpent doit se déplacer. Les différentes conditions (if) ajustent la position de la tête du serpent (pos\_x[0] et pos\_y[0]) en fonction de la direction choisie. Par exemple, si direction est égal à 1, la tête du serpent se déplace vers le haut en diminuant la coordonnée y.

Détection de Collision avec les Pommes : Ensuite, la fonction utilise une boucle for pour parcourir un certain nombre de pommes (pommex et pommey). Si la tête du serpent atteint la position d'une pomme, cela signifie qu'une collision s'est produite. Dans ce cas, le nombre de segments du serpent (segment) est augmenté de 2, et la position de la pomme est réinitialisée à une nouvelle position aléatoire.

Mise à Jour du Serpent : Après avoir géré le mouvement et les collisions, la fonction appelle Update\_Serpent() pour mettre à jour la position visuelle du serpent en fonction des changements apportés à ses coordonnées.

Affichage du Terrain : La fonction appelle également une fonction Terrain() pour mettre à jour l'affichage du terrain ou de l'environnement du jeu.

Une image contenant capture d’écran, vert

Description générée automatiquement

1. Conclusion personnelle :

Conclusion Clémence : La conception du Snake a été stimulant, mettant en lumière l'importance de la planification et de la résolution de problèmes. En surmontant les problèmes techniques, j'ai acquis une compréhension plus approfondie du développement de jeux et des principes fondamentaux de la programmation. Au-delà des lignes de code, ce projet a renforcé ma persévérance et ma capacité à transformer des idées abstraites en une réalité interactive. Je suis fier du résultat obtenu même s’il y a encore beaucoup de travaille pour m’améliorer qui m’attend.

Conclusion Clément : Pour le moi ce projet montre les difficultés à programmer un jeu. Donc je n’imagine même pas comment cela doit être compliqué en entreprise. Même si cela est super-intéressant de voir comment un jeu est codé, car quand on est joueur on ne voit pas forcément les « backgrounds » des jeux vidéo. Malgré le manque de partie graphique de notre part, ce concentré sur la partie technique a été un réé défie pour essayer de donner un jeu qui marche avec les contraintes données. Même si je pense que notre projet peut être plus optimisé, pour un début je suis assez fier de nous !