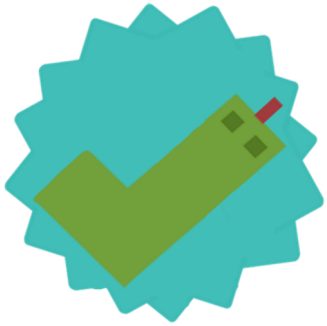
DESCRIPTION DU PROJET
SNAKE MÉGA OFFICIEL



PRÉSENTATION GÉNÉRALE

Evelyn, Fanny et Doriane vous présentent leur jeu Snake (méga officiel).
Le joueur va pouvoir se challenger en dirigeant le serpent vers la nourriture placée aléatoirement sur la grille de jeu, faisant ainsi monter son score et grandir le serpent. Il va néanmoins devoir faire attention à ne pas se cogner dans les murs ou à se manger lui-même, sous peine de "Game Over" !

ÉTAPES DU PROJET

Concernant le programme que nous avons codé, nous l'avons avant tout commencé ensemble lors des étapes ci-dessous :

- La fenêtre de jeu : ses dimensions, le titre, la couleur de base du fond,...
- La structure du jeu : la boucle "while", les évènements de base à détecter dans la boucle (ex : fermer la fenêtre quand l'utilisateur appuie sur la touche "échap" ou qu'il clique sur la croix), le nombre de répétitions de la boucle par seconde,...

La charge de travail a ensuite été répartie équitablement entre les membres.

ORGANISATION DU TRAVAIL

Evelyn a codé la majorité de ce qui se rapporte au serpent lui-même :
→ Ses possibilités : l'arrêt de la partie quand le serpent cogne un mur ou se rentre dedans (check_fail()), la croissance du serpent lorsqu'il mange un fruit (check_elements()),...Tout cela passe par des évènements à détecter en plus dans la boucle principale "while".
→ Son graphisme : taille (adaptée à la grille de jeu), couleur, yeux,...Cela a été mûrement réfléchi afin de conserver un design du serpent simple mais agréable à jouer

→ Ses mouvements : l'option des vecteurs a ici été utilisée pour représenter sous Python les déplacements du serpent guidé par le joueur grâce aux flèches du clavier. De plus, cette partie du programme est faite pour qu'un vecteur soit ajouté, soit une partie du corps en plus, lorsque le serpent mange un fruit.

→ Le logo Snake et son animation dans la vidéo de présentation.

Fanny a codé tout ce qui concernait la nourriture du serpent (les fruits) :

→ Le design : 3 fruits (pomme, orange et myrtille) ont été dessinés à la main pour offrir une diversité des graphismes au joueur.

→ Le positionnement : la nourriture apparaît aléatoirement sur l'une des cases de la grille, en prenant soin qu'elle n'apparaisse pas à un endroit déjà occupé par le serpent (fruit_aleatoire()).

→ L'alternance des fruits : La sorte de fruit apparaissant est choisi aléatoirement lors du programme, afin que l'ordre ne soit pas toujours le même (position_fruit()).

Doriane a codé les options supplémentaires au jeu :

→ La grille de jeu : l'arrière plan en damier a été choisi pour former le plateau sur lequel le serpent évolue afin d'ajouter de l'esthétique au jeu. Les couleurs choisies restent claires pour ne pas masquer le serpent lors de la partie.

→ Le score : cette option, affichée en haut à gauche, a été ajoutée afin de permettre au joueur de voir sa progression, battre ses records et s'amuser entre amis (score())

→ La fenêtre "Game Over" : elle s'affiche dès lors que le serpent se cogne contre un mur ou lui-même. Elle est simple et indique le score réalisé par le joueur lors de cette partie. Elle propose également au joueur de rejouer directement en pressant la touche "R", ajoutant un évènement à détecter dans la boucle principale "while" (game_over()).

VALIDATION DU FONCTIONNEMENT

ça marche.

IDÉES D'AMÉLIORATION

De nombreuses améliorations et évolutions du jeu sont possibles pour ce jeu :

→ Nous pourrions tout d'abord penser à un écran d'accueil, évitant ainsi le lancement immédiat du jeu lorsqu'on charge le programme. Il pourrait également contenir plusieurs nouvelles options : musique de jeu qui pourrait être activée ou désactivée, choix de la couleur du serpent, choix d'un niveau de difficulté (changeant par conséquent la vitesse et pouvant ajouter des fruits empoisonnés)

→ Nous pourrions ensuite proposer un court tutoriel sur les règles du jeu, disponible avec l'écran d'accueil ou s'affichant automatiquement au lancement du jeu.

→ Nous avons penser à ajouter un mode multijoueur où deux joueurs pourraient jouer ensemble sur la même grille de jeu, avec deux serpents. Cela permettrait de jouer ensemble en coopérant pour aller le plus loin possible en faisant monter le score !

→ Une intelligence artificielle qui simule une partie.

DOCUMENTATION TECHNIQUE

-Pour réaliser le Snake, nous avons utilisé une bibliothèque python nommée Pygame, permettant la création de jeux. En effet, Pygame crée une fenêtre qui reste affichée de par la boucle while du code.

-De plus, nous avons utilisé les CLASSES. Bien qu'elles appartiennent au programme de terminale, nous les avons étudiés chez nous. Elles servent à catégoriser les fonctions et à rendre le code plus lisible. Pour utiliser ces dernières, il faut préciser à quelle classe elles appartiennent.

Ex: snake.ajoute_bloc()
→ ajoute_bloc() est la fonction appartenant à la classe snake
Si la fonction est utilisée dans sa propre classe, il faut le préciser aussi.
Ex : self.position_fruit()
→ position_fruit() est la fonction utilisée dans la classe fruit. Lorsqu'elle est utilisée dans sa classe, on le précise avec «self».