LAURENT Maëva, LELONG Chloé - Groupe TD-TP 4

AP1 - Mini-projet :

SNAKE

Introduction :

Snake est un jeu très populaire, dont il existe de nombreuses versions différentes à travers le monde. Dans ce jeu, le joueur prend le contrôle d'un serpent qu'il peut déplacer sur un terrain où apparaitront petit à petit des pommes. Le but du jeu est d'avaler le plus de pommes possibles, mais attention : chaque pomme rend le petit serpent de plus en plus grand, si bien qu'il en devient difficile à manier. Si le serpent se mord lui-même ou s'il heurte les côtés du terrain, la partie est perdue. Beaucoup de nouvelles règles ont été créées au fur et à mesure des années, rendant chaque version de *Snake* unique, et nous en avons nous aussi tenté l'expérience.

Documentation technique :

Modifications de fonctions déjà existantes et définitions des nouvelles fonctions :

- **affiche_serpent :** Nous avons modifié cette fonction de sorte que, pour chaque élément faisant partie de la liste corps, les coordonnées de chaque élément soient converties par la fonction case_vers_pixel, qui permettra en suite de créer un cercle vert correspondant à une partie du corps du serpent.

- **change_direction :** En reprenant le début de fonction déjà écrit, nous avons continué dans ce même sens avec les 3 autres directions possibles. Nous avons aussi ajouté une condition en plus afin que le serpent ne puisse pas faire demi-tour, c'est-à-dire que si une touche est pressée alors que la direction initiale est l'opposée de la direction associée à cette touche, alors la direction ne changera pas. La fonction retourne la direction du serpent sous forme de couple d'entiers.

- **deplacement :** Cette nouvelle fonction prend en argument les coordonnées de serpent ainsi que le couple correspondant à la direction choisie par l'utilisateur (à l'aide de la fonction change_direction), et additionne les deux couples entre eux. Elle retourne les nouvelles coordonnées du serpent après le déplacement.

- **lapomme :** Cette fonction est celle qui se charge de générer les coordonnées valides d'une nouvelle pomme. On entend par « valide » que cette pomme n'apparaitra pas sur le serpent ou hors du terrain. Elle retourne ainsi le couple d'entiers représentant la position de cette nouvelle pomme.

Programme principal :

Initialisation du jeu :

Avant même que le jeu démarre, certains facteurs sont déjà déterminés, tels que la direction (le serpent se dirigera vers le haut au début du jeu), la position de la tête du serpent et du reste de son corps (au milieu du terrain de jeu), le score du joueur est initialisé à 0, et la première pomme du jeu est créée et ajoutée à la liste pommes vide contenant les coordonnées des pommes à venir. Le timer timerpomme, qui servira à déterminer quand faire apparaitre de nouvelles pommes, est également initialisé à 0. La variable **jouer** est aussi égale à False, et ce jusqu'à ce que la partie commence.

Note : Nous avons choisi de "séparer" le serpent et de faire une distinction entre son corps et sa tête. Sa tête correspond à la variable serpent et son corps correspond à la liste corps, dont le premier élément est serpent. Nous trouvions cela plus clair et plus simple à exploiter qu'une liste contenant le serpent entier.

Interface d'accueil :

L'interface d'accueil est le menu du jeu, on y trouve 3 boutons : **PLAY**, **HELP** et **EXIT**. Le premier permet de lancer une partie, le deuxième affiche les règles du jeu, et le troisième ferme instantanément la fenêtre. Pour quitter la page avec les règles du jeu, un bouton **BACK** est présent et permet de revenir sur le menu du jeu. Le bouton **PLAY** est celui qui lance la partie, et qui fait prendre la valeur True à **jouer**. Ainsi, un décompte est lancé et la partie peut commencer !

Affichage du terrain de jeu :

Sur un fond noir, nous avons créé une ligne blanche horizontale au niveau supérieur de la fenêtre, afin de pouvoir afficher le score dans un coin sans qu'il n'empiète sur le terrain de jeu. La première pomme apparait aussi, en même temps que le serpent, qui part dès la fin du décompte.

Déroulement de la partie et ses différents événements :

Direction que prend le serpent :

La partie continue tant que **jouer** a pour valeur True. Au moment où **jouer** devient False, la phrase **« Perdu ! »** apparait sur l'écran, et il n'y a d'autre choix que de fermer la fenêtre en appuyant sur

n'importe quelle touche. Dès l'affichage du terrain, le serpent se déplace en suivant les indications de l'utilisateur, dont la direction est déterminée par la fonction change_direction, qui impose aussi ses contraintes (pas de demi-tour ou de recul possible).

Déplacement visuel du serpent :

Pour calculer sur quelle case le serpent se trouvera, on utilise la fonction deplacement, mais pour le voir se déplacer, on fait en sorte d'ajouter un cercle sur la case déterminée précédemment, et d'enlever le dernier cercle qui compose le serpent. Pour cela, on ajoute les coordonnées de la case en question à corps, mais en première position (d'indice 0) afin que cela devienne les coordonnées de la tête du serpent. On enlève en suite le dernier cercle (le dernier élément de la liste) avec la commande « corps.pop() »

Apparition et disparition des pommes :

En dehors de l'initialisation de la première pomme du jeu, 2 conditions sont fixées pour faire apparaitre des pommes sur le terrain : s'il n'y a plus de pomme sur le terrain, soit dès que pommes est vide, ou si timerpomme atteint 100, sachant qu'il est réinitialisé à 0 dès qu'une nouvelle pomme apparait sur le terrain. On utilise alors la fonction lapomme, de la même façon que lors de l'initialisation de la première pomme du jeu.

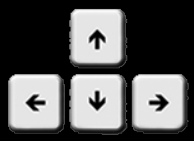
Les pommes disparaissent du terrain lorsque la tête du serpent passe dessus, soit lorsque les coordonnées de serpent sont identiques aux coordonnées d'un élément de pommes. A chaque pomme mangée, on additionne 1 au score, on supprime l'élément en question dans pommes, et le serpent gagne une nouvelle partie à son corps. Pour effectuer l'ajout du cercle, on utilise presque la même méthode que pour la visualisation du déplacement : on ajoute un cercle à l'avant qui représente sa nouvelle tête, sans supprimer le dernier cercle. C'est ce dernier cercle qui est vu comme le nouveau morceau gagné par le serpent.

Fautes éliminatoires :

Il existe deux conditions pour lesquelles jouer prend la valeur False, affiche le **message « Perdu ! »** et met fin à la partie :

- Si le serpent touche les bords : étant donné la ligne blanche créée et assignée en tant que bord du terrain, si l'abscisse ou l'ordonnée de serpent se trouvent respectivement en dehors des intervalles [0,39] et [3,32], le serpent touchera un des bords du terrain, et jouer prendra la valeur False.
- Si le serpent se mord lui-même : le serpent se mord lui-même dans le cas où les coordonnées du premier élément de corps (la tête du serpent), sont égales aux coordonnées d'un autre élément de corps. Suite à cela, jouer deviendra False.

Documentation utilisateur :



MENU DU JEU : Trois boutons s'offrent à vous lorsque le programme est lancé.

- ➔ **JOUER** : Mettre son curseur sur le bouton, et appuyer dessus à l'aide du clic gauche, permet de lancer une partie. Une fois avoir appuyé sur le bouton, un décompte s'affiche, la partie commence ensuite.
- ➔ **HELP** : Mettre son curseur sur le bouton, et appuyer dessus à l'aide du clic gauche, permet de prendre connaissance des règles du jeu. Celles-ci sont les suivantes : Vous dirigez un serpent à l'aide des touches directionnelles de votre clavier. (Toute autre touche ne fonctionnera pas) A savoir que votre serpent ne peut pas faire demi-tour sur lui-même, il avancera alors tout droit. Dirigez le serpent jusqu'aux pommes qui apparaitront sur l'écran. Il faut passer dessus pour gagner le point. Dans le cas où votre serpent s'enroule sur lui-même, la partie est perdue. Assurez-vous de rester dans la zone de jeu délimitée par la fenêtre de jeu, sinon c'est perdu. Pour quitter la fenêtre d'aide, appuyez sur le bouton "BACK", vous retournerez alors au MENU du jeu.
- ➔ **QUITTER** : Mettre son curseur sur le bouton, et appuyer dessus à l'aide du clic gauche, permet de fermer la fenêtre.

Conclusion :

Ce projet nous a permis de nous familiariser encore plus avec les travaux de groupe que lors des 3 premiers projets. En binôme, il a fallu nous départager le travail afin d'être le plus optimal possible, sans non plus totalement travailler de son propre côté. Nous avons beaucoup partagé de nos idées sur le projet, afin qu'il nous convienne à toutes les deux.

Globalement, nous n'avons pas rencontré de grandes difficultés. Chloé a eu du mal à faire le programme de son côté, mais après les explications de Maëva elle a pu rattraper son retard. Les doctest ont également constitué une certaine complication, puisque nous ne saisissons pas bien comment faire des doctest pour les fonctions d'affichage du serpent et des pommes.

Cependant, comme nous avons cherché à peaufiner notre programme jusqu'au bout, nous n'avons pas eu suffisant de temps pour arriver à mettre en place des améliorations possibles du jeu, ni même l'introduction du bouton *Rejouer*. Cela nous a tout de même mis en garde par rapport à la gestion du temps : la prochaine fois, il nous faudra passer moins de temps sur les fonctionnalités de base pour avoir l'occasion de rendre un projet encore plus complet.

Finalement, les retours sur ce projet sont très positifs, malgré la nouveauté du concept de jeu interactif, ce fut une expérience amusante et enrichissante que de créer son propre jeu de A à Z.