ESCARMANT Arsène - ROUSSEAU Mathieu – 1ère5

Compte-rendu : projet NSI 1er semestre 2019-2020

Sujet : reproduction du célèbre jeu 2048

1) Le projet initial

Le jeu 2048 consiste à ajouter des blocs contenant des puissances de 2 identiques dans une grille de 4x4 pour créer de plus grandes puissances. Une fois que le plateau de jeu est rempli et qu'on ne peut plus ajouter de blocs identiques, la partie se termine.

Une fonction spawn *ferait apparaître un 2 (ou occasionnellement un 4) après chaque coup, dans une case libre aléatoire.*
Une fonction move *déplacerait un bloc dans une direction donnée (à condition que le prochain bloc dans cette direction ne soit pas le même) jusqu'au prochain bloc ou au bord du plateau.*
Une fonction fusion *déplacerait un bloc dans une direction donnée et le fusionnerait avec le prochain bloc identique pour former un bloc dont la valeur vaut le double.*

Outils prévus : bibliothèque graphique Tkinter, images préenregistrées simplement à importer (plateau de jeu pour le fond, tuiles de jeu déplaçables).

2) Problème majeur : la bibliothèque Tkinter

Notre découverte de Tkinter fut longue et laborieuse. Entre les pages web consacrées au sujet, les tutos youtube de différents programmeurs, et les conseils occasionnels d'autres camarades bien plus expérimentés que nous dans ce domaine, nous avions des éléments qui auraient pu nous aider. Mais le problème venait de la diversité de la bibliothèque Tkinter et du peu d'informations contenues dans les longues vidéos youtube. Après de nombreuses tentatives pour créer un début de programme à l'aide du module Tkinter, nous avons décider de chercher si d'autres personnes avaient eu le même souci que nous. Nous avons donc consulté de nombreux forums sur internet, mais à notre grande surprise aucun d'eux ne comportait de réponses à nos questions, et il n'y avait pas une ligne de programme avec Tkinter. Essayant de comprendre les différents modules utilisés par les internautes, nous en avons découvert beaucoup, que nous ne pouvions malheureusement pas installer sur les ordinateurs du réseau du lycée. Parmi eux, *pygame* a retenu notre attention car il figurait dans de nombreux codes. Pourtant c'était une mauvaise piste, car malgré nos multiples tentatives, nos différents ordinateurs ne parvenaient pas à le télécharger. Nous avons alors abandonné Tkinter et envisagé d'utiliser *kandinsky*. Mais ce module

n'existant que sur la calculatrice Numworks, ce n'était pas non plus une bonne idée.

3) Solution au problème de la bibliothèque Tkinter

Plutôt que de s'éterniser sur des solutions trop complexes pour nous, nous avons préféré rédiger un programme avec un affichage et une interaction directement en console, quitte à avoir un premier programme pas très esthétique mais fonctionnel, et éventuellement le compléter plus tard s'il nous restait encore du temps. Nous avons donc travaillé sur un code interne très important avec un affichage négligeable sous forme de listes.
Voici le fonctionnement du programme :
- Tout d'abord, nous avons créé la grille puis la manière dont elle allait s'afficher, nous avons ensuite créé les cases de la grille.
- Puis, nous avons déterminé les conditions de la victoire et celles de la défaite, de l'échec.
- Ensuite, nous avons mis en place le déplacement, d'abord à droite et en bas puis en haut et à gauche. Nous avons également travaillé sur la restitution de la grille après le déplacement et sur la formation d'une nouvelle grille. Nous avons supprimé tous les 0 présents sur la grille avant et après chaque déplacement et nous avons appliqué la ou les modifications de la grille en fonction des différents déplacements. Après chaque déplacement, une fonction *spawn* qui fait apparaître un 2 (ou occasionnellement un 4), dans une case libre aléatoire. Puis pendant les déplacements, une fonction *move* déplace un bloc dans une direction donnée (à condition que le prochain bloc dans cette direction ne soit pas le même) jusqu'au prochain bloc ou au bord du plateau. Une fonction *fusion* déplace un bloc dans une direction donnée et le fusionne avec le prochain bloc identique pour former un bloc dont la valeur vaut le double.
- Enfin nous avons traité les situations possibles après l'échec d'une partie, soit le joueur souhaite recommencer et le programme se relance, soit le joueur souhaite quitter le jeu.

4) Conclusion

Après avoir envisagé l'utilisation de plusieurs modules de Python, nous avons écrit un programme en console avec un affichage en listes. Une possibilité s'est alors présentée : nous pourrions faire interagir le programme avec le joueur grâce à des commandes de *turtle* (listen, onkeypress…). Nous avons gardé la version en console de 2048 par manque de temps, mais néanmoins nous avons exploré les possibilités de *turtle* en amorçant un début de programme interactif qui fonctionnait, avec entre autres des couleurs de tuiles et des dominos au lieu des puissances de 2 (plus faciles à tracer).