Bergeot Loïc & Depaoli Ugo

Jeu du Moulin

Rapport de la création du jeu du moulin sur python 3

Introduction

* Objectif :

L'objectif est de recréer, avec l'aide de python, le jeu du moulin, un ancien jeu populaire en Europe. C'est un jeu qui possède plusieurs phases de jeu avec des règles différentes.

Table des Matières

I - Guide d’utilisation

A - Lancement du Jeu

Le but du Jeu est de retirer les pions du joueur adverse à l’aide de moulins, lorsqu’un joueur tombe sous la barre des 3 pions ou lorsqu’un joueur est bloqué : il perd. Un moulin est un alignement de 3 pions sur une ligne, faire un moulin nous fait obtenir le droit de retirer un pion adverse en dehors d’un de ses moulins.

Au lancement du jeu il vous est demandé de choisir un nombre de pions. Ce nombre de pions détermine sur quelle variante de plateaux vous allez jouer. Ce jeu possède 4 versions de plateau et sur chacune des versions se joue un nombre bien précis de pions.

B - Comment jouer

Commandes : Clic gauche de la souris - sélection de pions/emplacement

Pour commencer à jouer une fois la variante du plateau créé, il faut commencer par placer ses pions, pour se faire, cliquez sur un nœud pour pour placer un pion. Une fois que le premier joueur à placer son pion, c’est au tour du deuxième joueur. Et ainsi de suite jusqu'à ce que la limite de pion soit atteinte.

Ensuite c’est la phase de jeu, objectif : faire descendre l’adversaire à moins de 3 pions ou restreindre ses mouvements.

II - Fonctionnement de l’équipe et organisation

A - Organisation

Pour l’organisation, nous avons décidé de tout faire pendant les séances de travaux pratiques jusqu’aux vacances scolaires. Ensuite lorsque les vacances sont arrivées, nous avons uniquement utilisé l'application discord pour travailler et expliquer les incompréhensions. Donc au sujet du code nous avons tous les deux travaillé dessus.

B - Répartition

Il restait alors à faire le rapport et la partie graphique avec le module fltk. Nous nous sommes donc réparti le travail, comme Ugo est bien plus à l'aise avec le module fltk que Loïc, c’est donc Ugo qui s'est chargé de cette partie, Loïc s'est donc chargé de la rédaction du rapport.

Donc pour la répartition du travail en termes de pourcentages on a :

- Loïc : 40%
- Ugo : 60%

C - Difficultés

La seule difficulté rencontrée est au niveau du code car l’organisation n’était pas à plaindre. Pour le code, nous avons rencontré plusieurs problèmes, nous amenant à le changer en partie, voire totalement.

III - Avancement du projet

A - Avancement des tâches

Au début du projet nous avons cherché toutes les structures de données que l'on pensait adaptées. Les deux représentations les plus cohérentes étaient celles dites en "3D" et celles dites en "2D". Ne parvenant pas à nous décider et voyant une utilité différente aux deux représentations, nous pensions pouvoir utiliser la représentation 3D dans le code et la représentation 2D dans la partie graphique, nous nous sommes vite rendu compte que ça allait être plus compliqué qu'autre chose.

Nous avons donc opté pour la représentation 3D.

Après cela, nous avons donc commencé par la fonction « moulin » (avec l'aide d'une ancienne fonction que l'on utilisait dans la représentation 2D), qui consiste à détecter si un moulin se trouve autour du point que l'on vient de poser.

Suite à ça, il vient logiquement la fonction qui permet de prendre un pion après la création d'un moulin (pour rappel, un moulin consiste à avoir 3 pions alignés), c'est donc la création de la fonction « retirer pion ».

Nous avons ensuite continué la création des différentes fonctions, la fonction « voisins » qui consiste à renvoyer une liste de toutes les coordonnées des voisins d'un pion. Puis la fonction « perdu » qui regarde le « plateau » pour voir si un joueur a moins de trois pions ou qu'un des joueurs est bloqué. Ensuite nous avons créé la fonction « déplacement » qui sert simplement à déplacer un pion avec l'aide de la fonction « voisin » pour voir si le coup est possible. Après cela, il ne manquait plus que, selon nous, la fenêtre graphique fltk et la partie principale du code.

Ensuite nous sommes arrivés à un problème en créant la fenêtre graphique fltk, pour créer la fonction « pixel_vers_case », fonction qui consiste à faire correspondre la case dans la liste par rapport aux coordonnées du clic. Seulement, nous n'avons pas trouvé d'autre moyen que de

lister tous les cas possibles, ce qui était très long. Nous avons donc opté pour la représentation 2D et donc la modification du code entier.

Pour la plupart des fonctions ce n'était pas très compliqué. Après ce changement, nous avons donc pu continuer à avancer sur la fenêtre graphique fltk et la terminer quelque temps plus tard. Ensuite, il manquait que le "main" qui nous a posé quelques problèmes, résolus assez facilement.

B - Choix de structure des données

Parlons maintenant de notre choix de structure. Nous avons donc choisis une structure en « 2D », en terme de code, la structure 2D se traduit par une "liste de listes", dans la liste se trouve des « True » qui signifie qu'un point existe à cet endroit et donc que l'on peut y jouer en fonction de l'avancement de la partie. Dans les autres cases des listes se trouvent des « False » qui signifie que la case n'existe pas et donc que l'on ne pourra jamais jouer sur cette case. (La version 3D évoquée plutôt était une liste de listes de listes)

Voici l'exemple d'un de ne tableau de jeu :

[**[True**, False, **True**, False, **True**],

[False, **True**, **True**, **True**, False],

[**True**, **True**, False, **True**, **True**],

[False, **True**, **True**, **True**, False],

[**True**, False, **True**, False, **True**]]

C - Impressions

Plusieurs choses nous ont plus dans ce projet, tout d’abord, le fait que ce ne soit pas un jeu avec des règles simples, effectivement, ce jeu comporte donc trois phases les unes différentes des autres, donc on commence par créer notre code pour la phase une seulement lorsque l’on veut l’adapter pour la phase deux, on rencontre plusieurs problèmes ce qui nous pousse à réfléchir d’une autre façon et donc à changer le code régulièrement, ce qui a rajouté de la difficulté.

Ensuite ce qui nous a plu mais à la fois déplu, c’est que le choix de la structure de données était libre ce qui est plutôt bien car notre code est plus ou moins unique. Cependant ce côté-là nous a aussi déplu car avec toutes les structures de données différentes, nous avions souvent l’impression d’avoir fait le mauvais choix, que notre structure n’était pas la bonne ou qu' il y avait un meilleur choix plus pratique, plus rapide et plus facile. De plus, le fait que ce jeu soit plus ou moins dur à coder nous a permis d’utiliser tout ce que l’on a pu voir dans le premier semestre ce qui est plutôt encourageant, car on remarque que certaines choses que l’on a appris ont une utilité que l’on ne soupçonnait pas.