Campo, Thauvin et Bazire
16 janvier 2020

Rapport

Projet Labyrinthe

```
203         elif direction=='E':
204             labyrinthe['carte_amovible'] = decalageLigneADroite(plate
205
206 def tournerCarte(labyrinthe,sens='H'):
207     """
208     tourne la carte à jouer dans le sens indiqué en paramètre (H hora
209     paramètres: labyrinthe: le labyrinthe considéré
210                 sens: un caractère indiquant le sens dans lequel tou
211     Cette fonction ne retourne pas de résultat mais mais à jour le
212     """
213     if sens == 'H':
214         tournerHoraire(labyrinthe['carte_amovible'])
215     if sens == 'A':
216         tournerAntiHoraire(labyrinthe['carte_amovible'])
217
218 def getTresorCourant(labyrinthe):
219     """
220     retourne le numéro du trésor que doit cherche le joueur courant
221     paramètre: labyrinthe: le labyrinthe considéré
222     resultat: le numéro du trésor recherché par le joueur courant
223     """
224     return tresorCourant(labyrinthe['liste_joueurs'])
```

python™

Répartition du Travail

Au début nous avons commencé par nous répartir le travail avec les fichiers cartes pour Simon, joueur pour Maël et listejoueurs pour Fabrice.

Maël a ensuite géré les décalages du fichier matrice pendant que Fabrice a commencé Plateau et Simon labyrinthe.

Maël ayant fini à écrire matrice assez rapidement nous a rejoins et a écrit quelques fonctions sur chacun des fichiers plateau et labyrinthe.

Fabrice a ensuite écrit LabyrintheTexte et a fait une grande partie de la conversion Orienté Objet du projet.

Maël et Simon ont ensuite traduit quelques fonctions en Orienté Objet.

Et nous sommes enfin entrés tout les 3 dans une grande phase de débeugage.

Tests

Pour ce qui s'agit des tests, pour les 3 premiers programmes, nous avons utilisés les fichiers de tests fournis.

Et pour le reste des fichiers nous avons simplement utilisé des print pour voir si le programme affichait un affichage conforme à notre attente.

Je m'explique avec un exemple pour plateau, nous avons écris une fonction d'affichage d'un plateau pour pouvoir voir un peu plus clairement si en créant un plateau, on avait bien une matrice d'ordre 7 qui contenait bien 49 cartes avec les murs présents sur les bords du plateau, les joueurs dans les coins et enfin des trésors différents sur chaque carte.

Choix des structures de données

Joueur

Pour joueur nous avons décidé de prendre un couple avec le nom du joueur et la liste des trésors (une liste d'entiers).

Carte

Pour carte, nous avons opté pour un dictionnaire car il y avait trop d'informations à stocker pour utiliser un tuple. Il y a donc 6 clés, une pour chacune des directions nord, ouest, sud, est, une clé pour les trésors et une pour les pions. En ce qui concerne les valeurs, pour les murs, nous avons mis des booléens qui valent True si on a un mur dans la direction, un entier pour trésor et enfin une liste d'entier pour pions.

ListeJoueurs

La liste des joueurs est une liste à deux éléments (et pas un couple car avec un couple on ne pourrais pas modifier le joueur courant). Le second est un entier représentant le joueur courant et le premier est un dictionnaire dont les clés sont les numéro des joueurs et les valeurs sont des joueurs . (voir plus haut)

Plateau

Plateau est aussi une liste à deux éléments. Le second représente la carte amovible du plateau, celle que l'on va pouvoir insérer pour jouer. Le premier élément est une matrice à deux dimensions (liste de liste) de cartes (voir plus haut).

Labyrinthe

Le labyrinthe est un dictionnaire, encore une fois pour la même raison que précédemment, à savoir : il y a trop d'informations à stocker pour faire une liste ou un tuple. Il comporte 5 clés : plateau (la matrice stocké dans la premier élément de la liste de Plateau), listes_joueurs (la liste des joueurs sous la structure ListeJoueurs), carte_amovible (qui contient sous la structure Carte la carte avec laquelle on va jouer (comme vu dans plateau)), phase (un entier qui indique si on est en phase 1 ou en phase 2 du jeu et dernier coup (un couple initialisé à (None,None)).

Algorithmes implémentés et bugs

Nous avons réussi à implémenter tout ce qui était demandé dans le projet.

A savoir les 5 fichiers python nommés dans la partie précédente mais aussi le fichier labyrintheTexte.py

Mais nous avons encore pas mal de bugs connus, en réalité, nous n'avons même pas réussi a faire fonctionner correctement grand chose.

Que ce soit en mode graphique ou en mode texte on peut lancer le labyrinthe, configurer la partie (donc créer les joueurs et attribuer le nombre de trésors à chaque joueurs).

Pour chacun des modes , le labyrinthe s'affiche et le menu de choix est fonctionnel (pour le mode texte). On peut faire tourner la carte amovible autant qu'on veut par contre l'insertion d'une carte ne fonctionne pas. On voit bien au lancement du labyrinthe, les pions à chaque coins du labyrinthe. Et dans le mode graphique on voit bien aussi le nombre de trésors attribué à chaque joueur, ce nombre est bon.