

Pièges !

Table des matieres :

Organisation du travail	3
Guide d'utilisation du programme	4
Avancement du projet	4
structures de données	5
Informations sur le code	5
Remarques sur le projet	5

Organisation du travail :

J'ai organisé mon travail en deux fichiers , game.py pour les fonctions de base du jeu et pieges .py pour l'interface graphique et le moteur du jeu

J'ai commencé par aller sur game.py et a marquer en commentaire toutes les informations essentielles, regles du jeu ,fonctionnement de chaque tour .

Ensuite,j'ai listé les structures de données nécessaires ,ainsi que toutes les fonctions essentielles , comme par exemple generer_tirettes,placer_billes,déplacer,generer_tirettes etc....

Après cela, j'ai crée 3 variables globales ,DIM(pour dimension) lui affectant la valeur de 7(car notre plateau sera de dimension 7x7), et ensuite VERTICAL et HORIZONTAL , leur affectant respectivement la valeur de 0 et 1 ,ce qui me sera utile par la suite pour connaître l'orientation de mon tableau.

J'ai ensuite organisé le code en 3 classes ,Plateau,Tirettes,Joueur dans lesquels j'ai mis en place les méthodes essentielles au fonctionnement du jeu afin de pouvoir par exemple déplacer() dans la classe Tirette , placer_bille() dans la classe Joueur etc.

Après avoir élaboré game.py , j'ai ajouté le fichier pieges.py pour créer l'interface graphique et le moteur du jeu à l'aide du module fltk

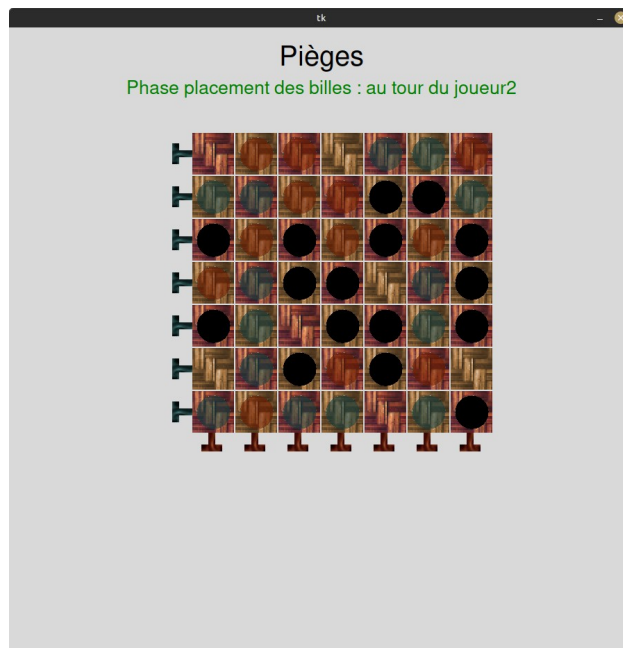
J'ai fais le travail seul (convenu avec Mme Van Den Bogaard), ce qui m'a donné sans doute beaucoup plus de travail que les autres , mais cela m'a permis de me surpasser , de découvrir de nouvelles façons de coder , comme avec la poo (Programmation Orientée Objet) qui a permit de rendre beaucoup plus lisible mon code , et de me retrouver dans mon code facilement car avec un projet qui dure plusieurs semaines , on peut potentiellement se perdre ou oublier pourquoi on a codé une ligne de telle ou telle manière.

J'ai beaucoup consulté le forum Stackoverflow , en particulier pour la poo Python, j'ai trouvé des renseignements intéressants et je pense que cela a renforcé mon autodidaxie .

Guide d'utilisation du programme :

pour lancer le jeu , on exécute pieges.py, l'interface graphique s'affiche avec le plateau , on peut désormais jouer.

(1)



La Phase de placement des billes commence , chaque joueur doit placer ses 5 billes en faisant un clic Gauche sur une case de son choix,une fois que cela est fait , la phase de manipulation des tirettes commence , chaque joueur manipule une tirette de son choix chacun son tour,pour tirer une tirette , on doit faire un clic gauche dessus ,pour pousser une tirette , on doit faire un clic droit, le dernier joueur a avoir au moins une bille sur le plateau gagne.

(2)



Avancement du projet :

structures de données :

- La position des billes sur la grille
 - liste des billes (tuples x,y) ✓
- liste des cases pour les tirettes (taille 7×7) (O(1))
 - lorsque trou True
 - sinon false False ✓
- La position des billes sur la grille ✓
- La position des tirettes et la position de leurs trous ✓
- Une fonction qui déplace une tirette ✓
- Une fonction qui permet de placer une bille d'un joueur sur la grille (pour la phase préliminaire) ✓

Bugs connus :

il peut arriver très rarement qu'il n'y ait plus d'une bille qui est sur une case sans trous , empêchant le jeu de se terminer .

Améliorations éventuelles :

non.

Informations sur le code :

J'ai choisi d'utiliser la poo python (programmation orientée objet), pour ses avantages de clarté et de structure , en effet , la poo python est très pratique pour se repérer dans son code , .

j'ai également utilisé des variables globales :

-DIM, VERTICAL, HORIZONTAL dans le fichier game.py
-LARGEUR_FENETRE, HAUTEUR_FENETRE, OFFSET_X ,OFFSET_Y,
DIM=game.DIM, VERTICAL = game.VERTICAL,
HORIZONTAL = game.HORIZONTAL, PLACEMENT, TIRETTES, FIN dans le fichier piege .py

Remarques sur le projet :

Ce projet m'a plu car j'ai appris à utiliser la poo Python et à être autodidacte , de plus coder un jeu me plaît beaucoup , et l'avantage est que le debuggage est plus attrayant car nous avons un résultat visuel ce qui n'est pas le cas dans les debuggages de code lambda .