

Rapport Projet Informatique

Table des matières

sujet	1
graphique.....	4
Partie HTML	4
partie JavaScript	5
partie brython.....	5
autre.....	6
Source utilisée	6

Sujet :

Mon sujet était le nonogramm.

Mon programme devait être capable d'en résoudre une grille.

Approche du problème :

Pour résoudre ce problème, j'ai opté pour le langage de programmation python pour la résolution et du HTML, JavaScript et brython (browser python) pour la partie graphique.

J'ai utilisé la méthode de programmation de POO permise par python.

```
nonogramm.py > ...
1  import copy
2  from random import randint
3
4
5  > class Type: ...
8
9
10 > class Case: ...
56
57
58 > class Grille: ...
60
```

J'ai pour cela créé 3 classes, la première, Type, peut être comparée à une énumération en java, elle sert à stocker les noms de variables sans se mélanger.

La deuxième, Case, contient les données d'une classe, elle m'a servi à pouvoir avoir des variables globales pour les grilles, elle est organisée en dictionnaire dans la classe Grille.

Et la dernière classe, Grille, est celle contenant toutes les cases et faisant tous les calculs nécessaires à la résolution d'un nonogramm.

C'est la classe contenant le plus de méthodes.

Chaque méthode commençant par '_' est une méthode privée n'ayant pas d'utilité en dehors de la classe.

J'ai créé 3 méthodes de résolution différente :

La méthode « humaine » qui va résoudre et réfléchir comme pourrais le faire un être humain, c'est à dire en utilisant la logique.

La méthode brute, cette méthode consiste à chercher en premier lieu toutes les combinaisons possibles pour une ligne grâce à un produit scalaire, puis à tester toutes les combinaisons possible jusqu'à trouver la grille qui est rempli

Ex :

[■,X,X,■] [X,■,X,■] et [■,X,■,X] sont les 3 combinaisons possibles pour l'indice [1,1] sur une ligne de 4

Et la dernière méthode utiliser est la méthode du back tracking,
Cette méthode consiste à remplir une à une les lignes de d'une grille tant que les indices peuvent être possibles, cette méthode est plus efficace que la méthode brute car elle permet d'éliminer nombre de combinaisons avant d'avoir tout rempli. C'est la méthode utiliser par la partie graphique car elle est la méthode la plus efficaces des 3.

Graphique :

J'ai choisi de représenter les grilles graphiques en HTML et JavaScript/brython, car je trouvais ça plus simple que de le faire grâce à des imports graphiques python que je ne connais pas.

Brython est une implémentation python 3 pour HTML 5.

Il permet la même chose que JavaScript dans une page web.

J'ai seulement utilisé du JavaScript parce que j'aime bien ce langage.

J'ai utilisé principalement le JavaScript pour créer la grille à l'écran, puis le brython pour les comportements en lien avec la classe Grille, importé depuis le fichier python présent dans le même dossier.

Partie HTML :

Le code en html est principalement présent pour organiser la structure du site,

Partie JavaScript :

Le code en javascript sert seulement à faire des modifications sur la page html,

Il pourrait être complètement remplacer par le code brython

Partie brython :

Le code brython est l'essentiel de du programme graphique, il fait les import du fichier python, recupere les données, fais les calculs permettant la resolution des grilles, et modifie la page en fonction des choix de l'utilisateur

Autre :

J'ai aussi utilisé un Git pour pouvoir revenir en arrière en cas de mauvaise manipulation de ma part. Avec un remote sur GitHub, d'ailleurs ce fichier étant dans le même dossier que le code, il fait aussi partie du Git.

[lien du Git](#)

Source utilisée :

Merlin et ChatGPT (comme documentation et recherche d'erreur)

<https://brython.info/>

(Pour les fonctions propres à brython)
