

Cahier des charges

SUDOKU - Groupe 6

Projet du premier semestre - DevWeb II

A. Glossaire :

- *MVP* :

Minimum Viable Product ou produit minimum viable. Version d'un produit qui permet d'obtenir un maximum de retours client avec un minimum d'effort (source: Wikipedia)

- *EPHEC* :

Ecole Pratique des Hautes Etudes Commerciales

- *Console/bash/shell* :

Programme qui reçoit des instructions informatiques par un utilisateur et qui les exécute dans le système d'exploitation.

- *Scripts* :

Programme qui exécute une action prédéfinie dans un fichier exécutable

- *Pycharm, Visual Studio Code, IDLE*

Éditeur de code spécialisé pour le langage Python.

Éditeur de code intelligent qui peut compléter automatiquement pendant le codage.

- *GUI* :

Graphic User Interface, interface graphique permettant à l'utilisateur d'interagir dans la fenêtre de jeu.

1. Présentation du client

Il n'y a pas de client à proprement dit pour la réalisation de ce travail. Nous sommes, Jean-Michael Tang, Geoffrey Brogniet, nos propres clients.

2. Présentation du projet

Notre projet consiste en la création d'un jeu mondialement connu qui est le sudoku. Nous aimerions réaliser le jeu de A à Z, mais aussi la création d'une base de données qui puisse enregistrer ce qu'on veut, comme les utilisateurs ou les scores. On ne veut donc pas ressembler à tous les sudokus qui se trouvent sur le marché, mais nous voulons nous différencier en ajoutant nos petites touches personnelles.

3. Objectif du client

Nos objectifs sont les suivants, proposer un jeu jouable, simple, clair pour les joueurs, mais aussi nous voulons que les joueurs puissent enregistrer leur partie en cours, leurs parties finies et pouvoir garder, dans leurs comptes personnels, le temps de jeu de chaque partie, mais aussi le temps total de jeu et qu'ils puissent avoir la possibilité d'enregistrer dans un fichier PDF tout ce qui est cités ci-dessus. Nous aurons donc, pour cela, besoin d'une base de données.

4. Intervenants

Les intervenants dans ce projet sont, d'abord, nous les clients, programmeurs, développeurs du jeu, mais aussi nos correcteurs, coaches, professeurs de notre section d'informatique de l'EPHEC qui sont Virginie Van Den Schriek, Jonathan Noël et Xavier Dubruylle.

5. Cibles/Utilisateurs

Ce projet, ce jeu est destiné à tout le monde connaissant le sudoku et sachant y jouer. Mais bien sûr, ce jeu peut cibler les personnes ne connaissant pas les règles du jeu vu qu'elles seront explicitement visibles dans notre jeu.

6. Demandes fonctionnelles

Nos demandes fonctionnelles vont être réparties en plusieurs points afin de voir plus clairement toutes les demandes. Nous avons décidé de séparer les demandes fonctionnelles de notre MVP et de notre projet final.

Voici les demandes fonctionnelles de notre MVP :

- Le jeu va demander à l'utilisateur d'écrire son nom ou son pseudo s'il veut afin de savoir qui joue.
- Le joueur souhaite pouvoir choisir sa difficulté de jeu (Facile, Moyen, Difficile, Expert)
- Le jeu va créer une grille de nombres de manière aléatoire. Selon la difficulté, le nombre de cases vides va augmenter.

- Pour chaque cases vides, et tant qu'elles ne sont pas remplies, le jeu va demander, via une ligne de commande, où le joueur souhaite placer un chiffre dans une case. Par exemple, si la case de la ligne 2, colonne 3 est vide et que le joueur souhaite la remplir, il devra indiquer 2 pour la ligne, 3 pour la colonne et le chiffre qu'il pense être bon.
- Dès que le joueur va faire une faute, le jeu lui indiquera cela et va comptabiliser les fautes.
- Le jeu à la fin de la partie va analyser la grille et arrêter le temps de jeu. Après ça, il va calculer le nombre de fautes réalisées durant la partie et va additionner le nombre de secondes en plus au temps de jeu, ce qui donnera le temps final.

Voici les demandes fonctionnelles du projet final :

- Le jeu, quand il va être lancé, ouvrira une fenêtre de connexion. L'utilisateur pourra donc se connecter ou alors choisir de créer un compte. S'il fait ça, le jeu va ouvrir une nouvelle fenêtre qui sera la fenêtre d'inscription. Lorsque ça sera fait et que tout sera convenablement complété, le jeu va revenir sur la page de connexion.
- Lorsque le joueur sera connecté, une nouvelle fenêtre va s'afficher et il aura deux choix, soit commencer une nouvelle partie, soit voir son compte et ses parties précédentes (s'il en a). Le jeu va donc soit créer une nouvelle partie, soit ouvrir une nouvelle fenêtre avec le compte du joueur.
- Si le joueur choisit de créer une nouvelle partie, le jeu va ouvrir une nouvelle fenêtre et le jeu va demander au joueur de choisir une difficulté (il aura le choix entre Facile, Moyen, Difficile, Expert). Lorsque ça sera choisi, le jeu va créer la grille selon la difficulté et va démarrer le temps.
- Pendant la partie, si le joueur fait une faute, le jeu retiendra le nombre de fautes faites pour faire le calcul à la fin du jeu.
- À la fin du jeu, le jeu va enregistrer le temps sans les fautes, va analyser la grille, va comptabiliser les fautes totales de la partie, va additionner ce total de fautes (1 faute = 10 secondes) au temps retenu pour donner un temps final et va annoncer via un pop-up au joueur son score final avec temps et nombres de fautes.
- Après, quand le joueur aura fermé le pop-up, le jeu va enregistrer les données du jeu comme la date de la partie, l'image de la grille, le temps final pour le mettre dans le compte du joueur.

7. Contraintes

Nous souhaitons, mais nous devons également développer ce projet dans le langage de programmation Python. Nous utiliserons la version 3.8 de Python qui est la dernière en date. Python permet d'avoir un code clair, net et correctement indenté car, c'est là son principal atout, l'indentation.

Nous devons aussi nous assurer que le code puisse être facilement "réparé" en cas de panne ou de problèmes. Nous devons pour cela mettre en place des tests unitaires pour vérifier que notre code ne se rentre pas en conflit lui-même mais, nous devons aussi créer des scripts, shell pour régler des problèmes plus rapidement.

Nous devons aussi faire attention à ce que l'application soit disponible sur ordinateur via une fenêtre de jeu, via la console de l'ordinateur ou via un smartphone.

Autre contrainte est l'agenda donné par nos professeurs où nous devons tenir un œil régulier et fournir des parties de projet régulièrement également.

8. Charte graphique / Ergonomie

Nous ne devons pas fournir un effort, prendre du temps sur la partie graphique de notre projet, il n'y a pas d'exigence au niveau de cela.

Le jeu doit pouvoir également se jouer en console, c'est-à-dire que via un programmeur comme PyCharm ou Visual Studio Code, nous pouvons y jouer sans devoir créer une fenêtre de jeu. Mais bien sûr ce jeu doit être joué sur une console Python ou en fenêtre GUI.

9. Enveloppe budgétaire

L'enveloppe budgétaire de projet est inexistante. Il n'y a pas de rémunération derrière ce projet vu qu'il s'agit d'un travail d'école.

10. Planification

Le projet est prévu sur une période de 4 mois, par itérations d'une semaine. Chaque semaine aura lieu un état des lieux, un état d'avancement et une démonstration de l'avancée et du développement de l'application.