



nom de votre projet :	Sudokool
membres de l'équipe :	Clovis PACCOU
membres de l'équipe :	Evan PORTRAIT
membres de l'équipe :	Danny MASSOU
Niveau d'étude :	Première Générale
établissement scolaire :	Lycée Charles de Gaulle, Muret, OCCITANIE

enseignante/enseignant de	M. Jeronimo
NSI :	

> PRÉSENTATION GÉNÉRALE :

Notre projet consiste en un jeu de Sudoku intégrant des fonctionnalités pratiques et esthétiques dans l'objectif d'améliorer l'expérience utilisateur par rapport à une application de sudoku classique.

Ce projet est né lorsque nous nous sommes donné le défi de trouver un algorithme permettant de résoudre une grille de Sudoku quelconque. Ainsi, lorsque notre professeur de NSI nous a parlé de ce concours, nous avons profité de l'occasion pour mettre sur pied ce projet, et d'en faire une application jouable à part entière.

Cette expérience nous a permis d'améliorer nos compétences en programmation, que ce soit pour la gestion de l'interface graphique, ou pour la partie davantage technique concernant la résolution et la génération des grilles.

Ainsi, le logiciel que nous proposons permet de jouer sur des grilles de dimensions 4×4 , 9×9 et 16×16 .

Il propose de charger et de sauvegarder une grille à tout moment, permettant ainsi de finir la grille ultérieurement par exemple. L'enregistrement a lieu dans un fichier, qui peut ainsi être copié, partagé, etc.

Ce jeu permet de résoudre une grille quelconque, et de générer des grilles uniques et de difficulté modulable.

Une option (désactivable) permet d'afficher les valeurs déjà présentes sur la même ligne, colonne ou carré que la valeur qui vient d'être saisie.

L'utilisateur peut décider de verrouiller une case, ce qui a pour effet de bloquer la case contre une modification accidentelle. L'utilisateur peut bloquer une case lorsqu'il est sûr de sa valeur.

Pour ce jeu, deux modes existent : le mode *joueur*, permettant de jouer au Sudoku normalement, et un mode *éditeur* permettant de modifier toutes les cases d'une grille, et donc de recopier des grilles papier vers le logiciel.

Pour finir, notre logiciel permet de choisir entre différents packs de textures, ce qui offre une variété et une originalité graphique à l'utilisateur, option dont les autres applications classiques trouvables sur internet sont dépourvues.

> ORGANISATION DU TRAVAIL :

Nous sommes trois à avoir contribué au projet, et avons tous eu un rôle spécifique lors de la réalisation de celui-ci.

Tout d'abord : Evan, l'un des deux programmeurs. Il s'est occupé principalement de la partie interface graphique de l'application, c'est-à-dire la partie visible de l'interface. Cela comprend le placement des boutons, de la grille, des images de fond, de l'intégration de la musique, et la partie fonctionnelle de l'interface, comprenant la liaison entre les actions du joueur sur la fenêtre avec les différentes fonctions du code, telle que le placement d'un chiffre sur une case, ou l'activation des différents boutons.

De plus, il s'est occupé de la structure générale du programme, c'est-à-dire de l'organisation du code en différents fichiers, classes et fonctions, afin respecter l'architecture modèle-vue-contrôleur (MVC), permettant ainsi un code clair et organisé.

Ensuite : Clovis, le deuxième programmeur. Il s'est quant à lui occupé principalement de la partie non visible du code. Il est le principal créateur de l'algorithme de résolution et de génération des grilles. La fonction de résolution est une fonction récursive, permettant un code compact qui assure de trouver systématiquement et de manière optimisée la solution d'une grille, si elle existe.

De surcroît, Il s'est aussi occupé de la partie communication du projet, comprenant la création de l'ensemble des documents textuels pour la présentation de ce projet, cela comprend ce document, la notice du jeu, et le sondage pour les élèves du lycée (une diffusion du jeu a eu lieu sur l'ENT de notre lycée afin d'encourager le test et les retours sur le jeu)

Enfin : Danny, dernier membre de ce trio. Il n'est pas développeur, et s'est occupé de la vidéo de présentation du projet, de la composition de la bande sonore du jeu et de la création des graphismes (images des packs de textures). Il a tout de même codé le fichier "main.py" du projet afin qu'il soit inclus dans la création du code.

Son aide a aussi été précieuse pour trouver des axes d'amélioration, identifier des bugs, et relever des optimisations d'ergonomie possibles.

Nous remercions également nos camarades du groupe de NSI ainsi que toutes les personnes ayant testé et émis des retours pour leurs conseils et leurs avis constructifs qui ont permis le développement de la version finale du projet.

Pour l'organisation du travail, nous nous sommes mis d'accord au tout début sur la structure générale du projet, en faisant quelques schémas pour avoir les idées claires.

Ensuite nous avons débuté le développement du projet, en utilisant GitHub pour mettre le code en commun : chacun des deux programmeurs créait une partie différente du jeu pour ne pas entrer en conflit. La répartition des tâches s'est faite plus ou moins naturellement, en fonction des compétences et des envies de chacun.

Nous relisons toutefois régulièrement le code de l'autre afin de comprendre ce qu'il avait fait, et éventuellement demander des précisions, puis potentiellement d'essayer d'améliorer le code et/ou l'optimiser en apportant un point de vue différent. Cette relecture d'un point de vue extérieur a permis d'ajouter des commentaires plus constructifs et pertinents (pour nous-même mais aussi pour vous, relecteurs de ce projet).

L'édition du code a été réalisée sur Visual Studio Code pour Evan, et sur PyCharm Community pour Clovis.

Enfin, nous avons communiqué principalement au Lycée, lorsque nous en avons l'occasion, et autrement par messagerie instantanée (Whatsapp). À ce propos, la très grande majorité du projet s'est faite en dehors du lycée, les heures de NSI étant dédiées au suivi du cours, et les heures de temps libre au Lycée n'étant pas suffisantes pour un avancement satisfaisant du projet.

Il est très difficile d'estimer le temps passé à la réalisation du projet, car celui-ci s'est étalé sur quasiment 5 mois (de novembre à Mars 2024), mais nous l'estimons à une cinquantaine d'heures chacun, soit un total de ± 150 heures.

LES ÉTAPES DU PROJET :

Après avoir trouvé l'idée du projet nous avons commencé par créer les différents fichiers code qui allait composer notre jeu. Nous avons créé en priorité les classes Game, Sudoku et Graphism. Ces dernières sont les piliers de notre code. Au début elles étaient plutôt vides, elles contenaient des méthodes basiques et/ou incomplètes, elles servaient à visualiser plus facilement la structure générale du code afin d'en faciliter la complétion

Nous avons commencé par créer une fenêtre rudimentaire, la possibilité de la redimensionner, d'afficher quelques éléments, etc. Pendant longtemps la fenêtre se composait d'un fond noir avec une grille basique. Nous voulions une interface graphique que pour se représenter grossièrement à quoi ressemblerait le jeu, nous permettant ainsi de créer les fonctionnalités principales dont notre projet allait avoir besoin.

Nous avons ensuite fait les fonctions basiques pour placer et supprimer des valeurs, et les méthodes pour ouvrir et enregistrer des sudokus sont arrivées rapidement. Ces dernières permettaient de faire des tests et des débogs avec une variété de grilles sans avoir à recopier une grille manuellement à chaque fois.

Une fois ces méthodes testées, vérifiées et approuvées, nous avons commencé la création de la méthode de résolution, il s'agit de la partie qui nous a motivé à créer le jeu. Puis nous avons ajouté la méthode de génération, qui nous a posé quelques problèmes par ailleurs.

La partie graphique complète est arrivée très tardivement, comprenant l'ajout des différents menus, boutons, et la possibilité de choisir différents pack de textures.

> FONCTIONNEMENT ET OPÉRATIONNALITÉ :

Nous n'avons pas rencontré beaucoup de difficultés, si ce n'est celle de vouloir ajouter trop de fonctionnalités à notre jeu par rapport au temps imparti. Nous avons dû en abandonner une partie pour garder les plus importantes à nos yeux.

Pour trouver et identifier les bugs, nous effectuons régulièrement des tests sur les nouvelles fonctionnalités implémentées. Nous avons aussi fait jouer nos proches et rendu une version disponible sur l'ENT du lycée, accompagnée d'un sondage pour recueillir des avis et identifier des bugs.

Un important travail d'optimisation a eu lieu pour permettre la résolution rapide actuelle. En effet, la résolution d'une grille prend en moyenne 1 seconde, contre plus d'une minute avant les optimisations apportées.

Au moment du dépôt, toutes les fonctionnalités que nous désirions implémenter sont parfaitement fonctionnelles, et nous avons fourni le code original, le code dépourvu de tout les test et messages consoles, et une version exécutable du logiciel (.exe).

> OUVERTURE :

Comme nous l'avons annoncé précédemment, un certain nombre de fonctionnalité ont dû être abandonnées en cours de route à cause du manque de temps. Nous voulions par exemple permettre à l'utilisateur d'ajouter des notes, c'est-à-dire les différentes valeurs potentielles pour chaque case, et ajouter une nouvelle texture pour les symboles de la grille lorsque la grille est entièrement complétée, puisqu'actuellement rien n'indique qu'une grille a été remplie.

De plus, la fonction de génération que nous proposons actuellement ne nous satisfait qu'à moitié, car elle ne permet pas de générer des grilles avec moins de 50 % de cases originelles, et prend un temps considérable à s'exécuter pour une grille en 16×16 , allant de 6 secondes pour une grille facile à plus de 15 minutes pour une grille difficile.

Enfin nous aurions aimé créer un menu à part pour sauvegarder une grille, charger une grille, et sélectionner un pack de texture, au lieu de passer par les fonctions pré-existantes et peu esthétiques de tkinter.

En outre, à moyen/long terme nous aimerions ajouter un mode « campagne », c'est-à-dire un mode permettant au joueur de tenter de résoudre une suite de sudokus de plus en plus durs en interdisant les options de "triche" (résolution automatique et affichage des erreurs).

Si le projet était à refaire nous nous organiserions mieux pour la gestion du temps, afin de ne pas être sous pression, surtout pour les dernières fonctionnalités ajoutées peu de temps avant la finalisation du projet.

De surcroît, grâce ce projet nous avons appris à travailler en équipe pour la création et le maintien d'un jeu. Nous avons des idées différentes, parfois incompatibles, et nous avons souvent dû faire du tri pour en faire le maximum, sans pour autant s'éparpiller pour au final se retrouver avec des implémentations partielles et/ou buggées.

Notre projet permet donc d'accéder facilement et efficacement au jeu connu du sudoku, car facile d'accès et d'utilisation. L'utilisateur peut se mettre au défi, avec des sudokus de difficulté modulable. Notre jeu propose une expérience utilisateur agréable et relativement complète, par rapport à ce qui existe déjà sur le sujet.