



Université Mohammed V de Rabat Ecole Nationale
Supérieure d'Informatique et d'Analyse des Systèmes



Rapport Projet C

SUDOKU POUR ENFANTS

Abdelkabir Ghazal | Ahmed Reda Eladnani

Encadrant: Mr A.EL FAKER | année universitaire
2018/2019

Remerciements

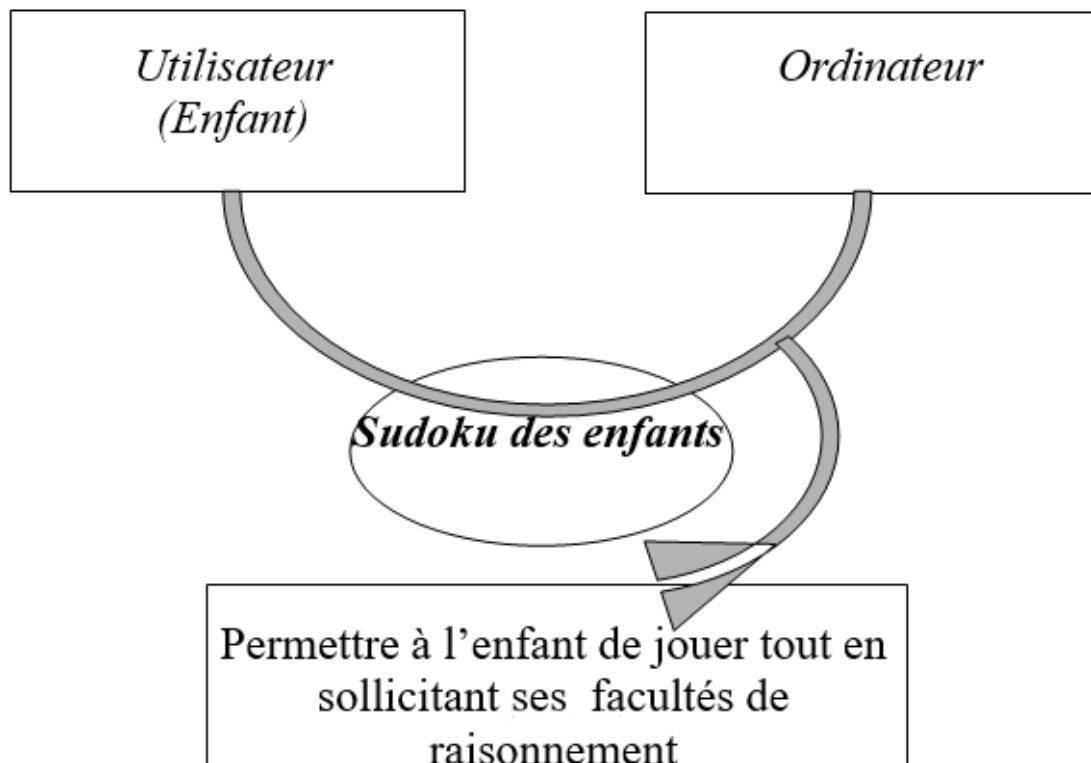
Nos remerciements vont à notre encadrant et professeurs du module de « Structures de données » : Monsieur Abdellatif EL FAKER pour son entière disponibilité, également à nos professeurs des modules « Algorithmique » : Monsieur Ahmed ETTALBI pour le savoir qu'il nous a transmis et de « Techniques de programmation » : Monsieur Hatim GUERMAH et Madame Safae ELHAZMIR pour les outils et les méthodes de programmation dispensées. Enfin, nos remerciements s'adressent aux professeurs et au corps administratif de l'Ecole Nationale Supérieure de l'Informatique et de l'Analyse des Systèmes-ENSIAS.

Table des matières :

Remerciements.....	1
Table des matières	2
Introduction au jeu.....	4
-Bête à corne.	
-Description du fonctionnement du jeu.	
Organisation du projet (code).....	9
-Les fichiers sources.	
-Les fichiers headers	
-L'exécutable.	
Fonctionnement du code source.....	10
-Méthode de résolution du Sudoku.	
-Les principales fonctions.	
Difficultés trouvées.....	13
-Manipulation de la SDL.	
-Organisation du Projet.	
-Erreur de compilation.	
-Contrainte du temps.	
Ressources.....	15
Conclusion et perspectives.....	16

Introduction au jeu :

Bête à corne :

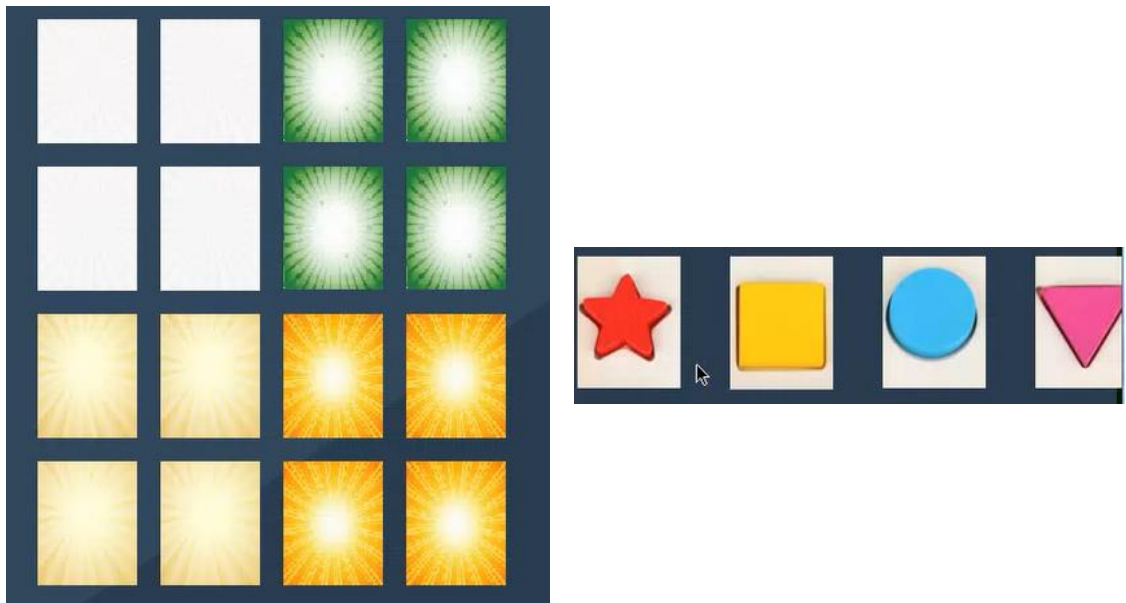


- Le jeu est destiné aux enfants, et prend la forme d'une grille de **taille 4×4** composée de **4 sous-grilles de 2×2** , différenciée par la couleur, appelées «régions». Les cases sont supposées contenir des formes géométriques. Le but est de remplir les cases vides en une durée minimale.

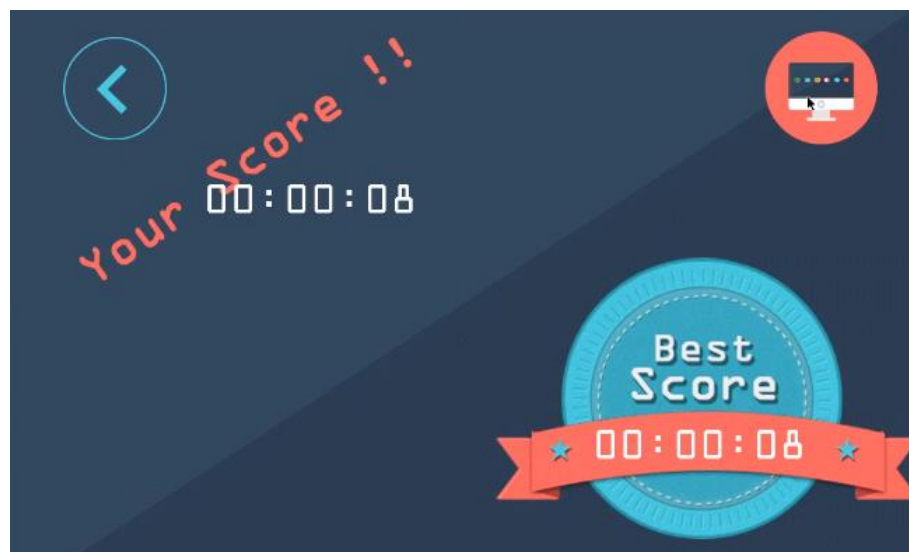
Un chrono est lancé à chaque début de partie.



La contrainte à satisfaire est la suivante: chaque case doit contenir une forme géométrique, de telle sorte qu'elle ne se répète pas encore une fois, et dans la même ligne et colonne, et dans la même zone.



- L'utilisateur s'identifie avec son **nom/pseudo nom** et un **mot de passe**. Il a le choix entre **2 niveaux** différents du degré de difficulté (débutant et difficile). S'il réussit à résoudre le sudoku, une affiche de félicitation apparaît ainsi qu'un audio.
- Chaque utilisateur a son meilleur score personnel enregistré.



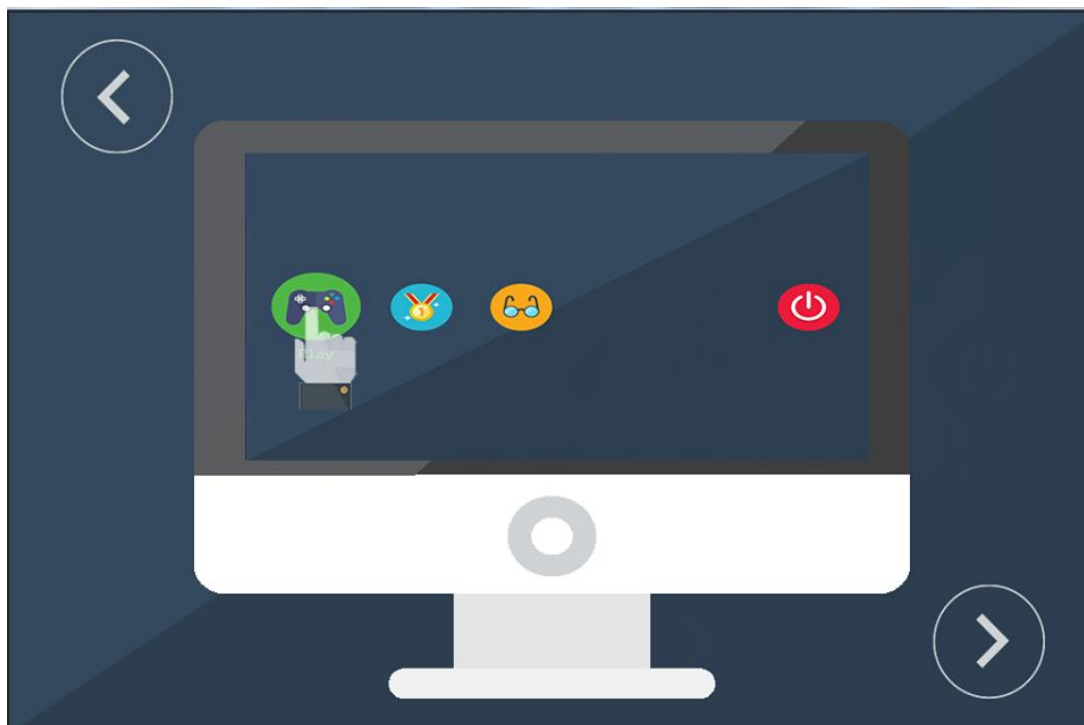
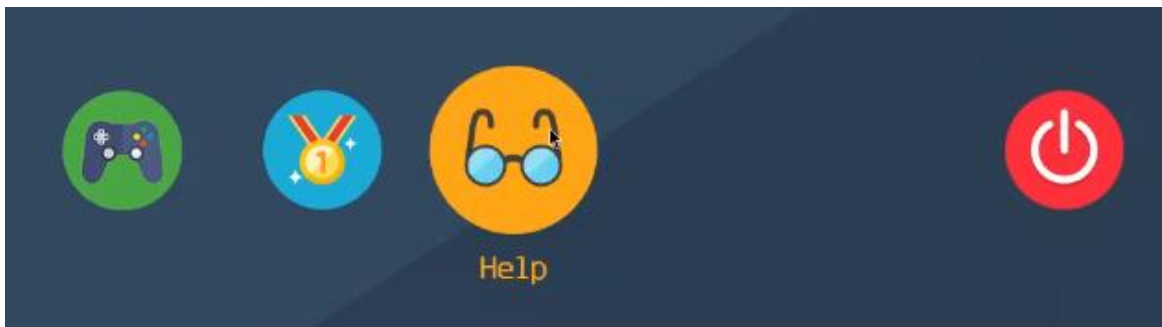
- Le jeu peut être recommencé en cliquant sur le bouton «[Reset](#)».

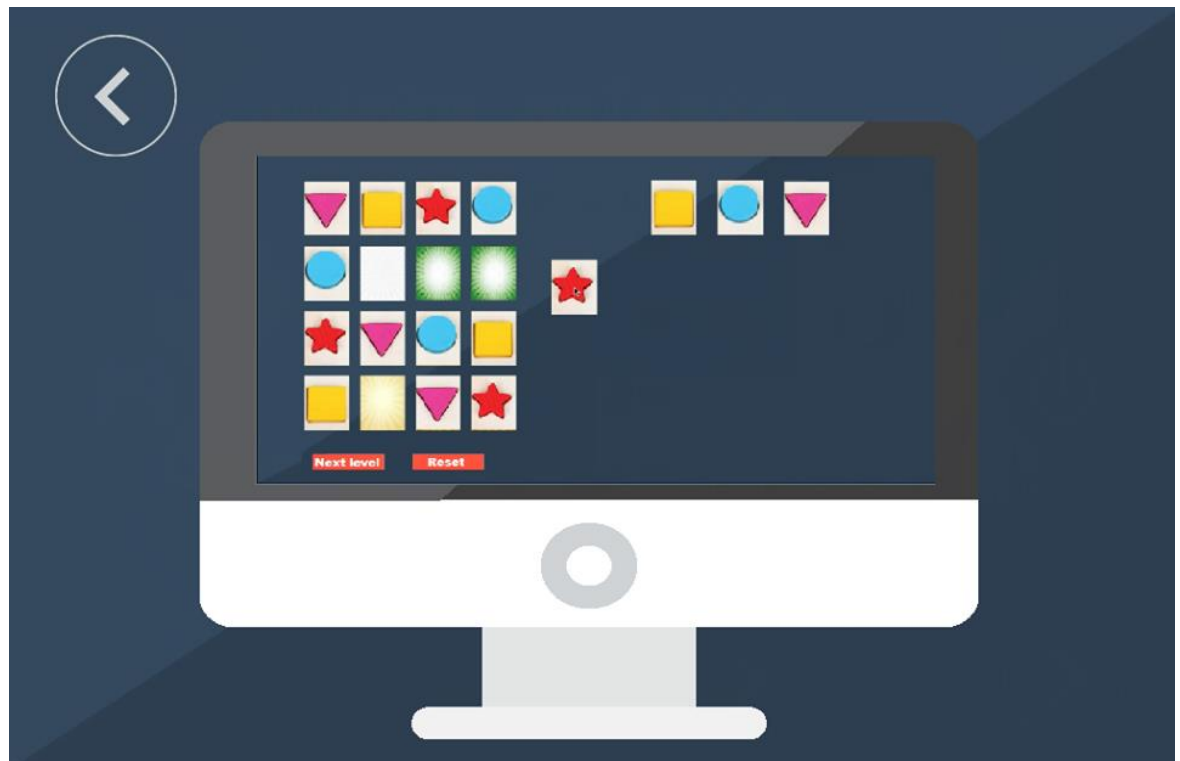
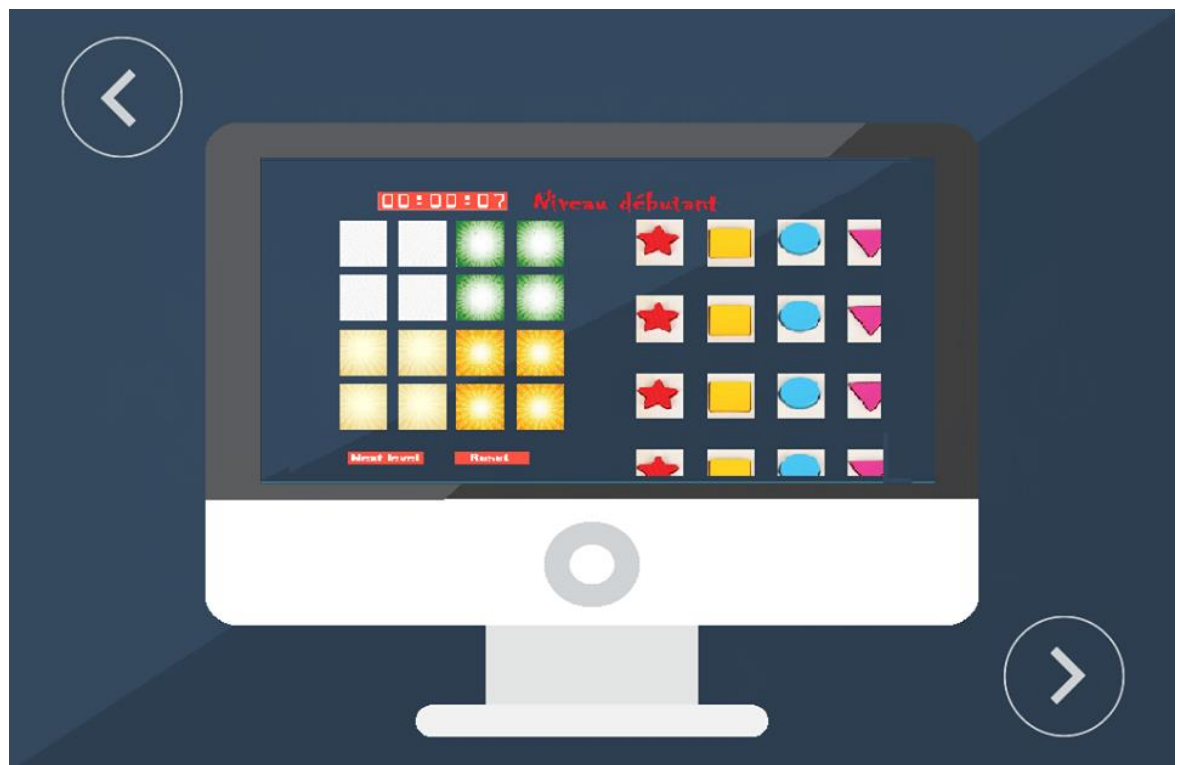


- Les scores de chaque utilisateurs sont enregistrés sur un fichier texte associé nommé «[players](#)». Les meilleurs scores sont affichés dans la classe «[Ranquing](#)» :



- L'utilisateur peut retrouver des consignes d'utilisation dans la classe «[Help](#)» :





Organisation du projet (code) :

Les fichiers du projet sont répartis comme suit :

- Des fichiers sources (.c) :
 - 1- Fonctions.c : contient les corps de toutes les fonctions utilisées.
 - 2- Main.c : contient la fonction main du projet (initialisation de la SDL, lancement de la fenêtre...)
- Des fichiers Headers (.h) :
 - 1- Fontions.h : contient les prototypes de toutes les fonctions et la déclaration des structures utilisées.
 - 2- Main.h : contient la déclaration des constantes utilisées dans la fonction main (la longueur et largeur de l'écran par exemple), ainsi que les includes nécessaires.
- Un fichier exécutable (.exe) :
 - 1- Sudoku.exe : l'exécutable du jeu.
 - 2- Contient aussi les bibliothèques SDL (SDL_image.dll , SDL_ttf.dll ,etc)

Fonctionnement du code source :

- **Méthode de résolution du sudoku :**

On a pensé à fixer 2 niveaux, chacun avec 3 distributions fixées vérifiant les règles du sudoku.

Quand l'utilisateur choisit un niveau, une des distributions est affectée aléatoirement à la grille.

Une fois que l'utilisateur remplit les cases, le programme vérifie si ses choix sont identiques à la distribution déjà fixée. Si oui, l'utilisateur gagne, sinon il reprend.

Cette méthode est certainement plus facile à coder que celle du 'Backtracking', mais fait l'affaire pour un jeu d'enfants.

- **Les fonctions Principales :**

➤ `void saisir1` La fonction permettant la saisie du mot de passe.

➤ `void afficher_texte(int x, int y, int taille, char *ch, int R, int G,int B);`

La fonction permettant l’affichage du texte saisi dans l’écran : -ch : le texte saisi

-x, y : les coordonnées de la zone où l’affichage du texte est souhaitée.

-taille : la taille du texte.

-G, B et R: les paramètres de la couleur du texte.

➤ `void cas1 () ;` Fonction où les répartitions du niveau 1 sont déclarées, puis affectées aléatoirement.

➤ `void cas11 () ;` Fonction qui contient la distribution 1 du premier niveau.

➤ `void game () ;` Fonction qui englobe l’ensemble des opérations exécutées dans le jeu. Affecte une distribution aléatoire à la grille selon le niveau choisi, lance le chrono puis vérifie les choix d l’utilisateur une fois qu’il a terminé la partie.

- `void datesys(char ch[]);` La fonction responsable du chrono.
- `void menu();` Affiche le menu :
 - Play
 - Ranquing
 - Help
 - Quit
- `void ranking();` Affiche les meilleurs scores.

Difficultés trouvées :

- **Manipuler la SDL :**

Comme c'est la première fois qu'on travaille avec la SDL, on a fait face à plusieurs difficultés. La principale difficulté était d'insérer les bibliothèques SDL. Les bibliothèques SDL_image, SDL_ttf et SDL_mixer n'étaient pas disponibles dans le package d'installation, il fallait l'installer séparément, ce qui nous a coûté beaucoup de temps pour s'en rendre compte.

Il nous a fallu apprendre à manipuler les fonctions SDL, donc à chaque fois qu'on voulait exécuter une certaine tâche, il fallait chercher la fonction nécessaire puis comprendre son fonctionnement.

- **L'organisation du projet :**

Au début, on travaillait dans un seul fichier main.c dont il y avait les corps des fonctions et leurs prototypes, la déclaration de toutes les structures

et variables, ainsi que toutes les includes nécessaires etc. Cela a créé un désordre au début, jusqu'au moment où on s'est rendu compte qu'il fallait créer des dossiers Headers et bien organiser son projet.

- **Erreurs de compilation :**

Comme c'est notre premier projet en langage C, il y avait à chaque compilation des erreurs de différentes causes, ce qui était embêtant à un certain temps.

Par exemple :

- **Contrainte du temps**

- **Autres :**

-Lors de la saisie du mot de passe, il faut qu'il s'affiche des étoiles à l'écran, alors que le texte saisi doit être sauvegardé comme mot de passe. Ce n'était pas facile à réaliser vu la confusion qu'il y avait entre l'affichage et l'enregistrement de ce dernier.

-Au niveau difficile , et vu la méthode de résolution qu'on a suivie, il nous a fallu de traiter plusieurs permutations différentes dans chaque cas.

Ressources :

Pour réaliser se projet, on s'est basé, ou plutôt utilisé plusieurs sources parmi lesquelles :

- Travexa : un jeu suivant la même logique du sudoku. On s'est basé donc sur son code source pour mieux comprendre la logique de celui du sudoku. Une video youtube montre le fonctionnement du jeu <https://m.youtube.com/watch?v=UF4FV8wMFvM>
- Openclasserom
- Clipartlogo : site d'où on a prit la plus part des images utilisé dans le projet.
- Github : où on a trouvé le code source de Travexa.

Conclusion et perspectives

Notre projet a consisté à la réalisation d'un Sudoku pour les enfants.

Ce projet nous a permis d'approfondir nos connaissances théoriques, acquises tous le long de notre formation, par la pratique des nouvelles technologies. Cette expérience nous a permis de maîtriser le langage C ainsi que la bibliothèque utilisée qu'est SDL.

Nous sommes arrivés à développer la majorité des fonctionnalités du système dans les temps.

L'amélioration qu'on peut voir dans ce projet et le fait qu'il permet aux enfants de se divertir, jouer en améliorant leurs raisonnements logiques, ce qui manque dans la plus part des jeux des enfants.

Fin.