



Projet C : Takuzu

KOCOGLU Lucas
MENIN Thibaut
Promo 2026

Groupe F
Groupe F

2021/2022



efrei
PARIS PANTHÉON-ASSAS UNIVERSITÉ

Table des matières

• Introduction du projet.....	3
• Présentation fonctionnelle du projet.....	3
➤ Fonctions proposées dans le sujet.....	3
• Présentation technique du projet réalisé	3
➤ Description principaux algorithmes.....	3
➤ Difficultés rencontrées et solutions apportées	5
• Conclusion	5
➤ Apprentissage sur l'organisation de travail.....	5
➤ Apprentissage sur le plan technique	5

• Introduction du projet

Le but de ce projet est de réaliser un programme informatique visant à recréer le jeu du Takuzu.

L'utilisateur a accès aux 3 grands domaines de ce projet :

- L'utilisateur joue sur une matrice en pouvant changer les cases visibles via le masque
- Le programme résout tout seul une grille de Takuzu
- Le programme génère une grille de Takuzu

• Présentation fonctionnelle du projet

➤ Fonctions proposées dans le sujet

Tout sauf génération_grilles (partie III).

• Présentation technique du projet réalisé

➤ Description principaux algorithmes

⇒ Séparer les fonctions dans différents dossiers donne plus de lisibilité au code.

▪ Les fichier « matrice » :



- Creation_mat_modele : Contient toutes matrices initialisées en dur dans la fonction. Utilisation de la fonction creation_matrice pour les mettre dans des tableaux dynamiques.
- Creation_matrice : Permet de créer un tableau dynamique avec comme argument la taille de la matrice à créer.
- Libere_matrice : Permet de libérer en toute sécurité l'espace mémoire utilisée par les matrices.
- Genreation_solution: Permet en fonction de la taille de générer une matrice solution pour la Partie 3.
- Créer_masque_aleat: Génère un masque aléatoire pour l'affichage de la matrice jeu.

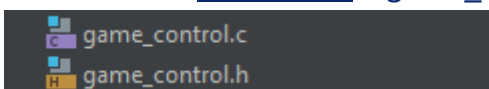
- Créer_masque_spe : permet de choisir les cases qui vont être visible dans la matrice jeu.
- Int_input_matrice : convertit la valeur des colonnes en chiffre et injecte la valeur choisie par l'utilisateur dans le masque.
- Takuzu_utilisateur : C'est la matrice visible par l'utilisateur. Elle est la combinaison de la matrice masque et de la matrice solution.

▪ Le fichiers « regles_taku » :



- Verification_cote : Vérifie lors du choix d'une valeur par l'utilisateur si la valeur respecte la règle de pas plus de valeur identiques côte-à-côte.
- Verification_lig_col : Vérifie lors du choix d'une valeur par l'utilisateur si la valeur respecte la règle que 2 lignes ou 2 colonnes ne peuvent pas être identiques.
- Verification_nb_iden : Vérifie s'il y a le même nombre de 1 et de 0 dans la ligne et dans la colonne.
- Verif_regles_taku : compile les trois fonctions de vérification des règles pour un soucis de visibilité.
- Grille_pleine : Condition d'arrêt quand la grille est pleine.

▪ Les fichiers « game_control » :



Contient la majorité des appels des fonctions pour faire la connexion entre-elles.

▪ Les fichiers « auto_resolve » :



Contient la fonction nécessaire à la résolution automatique des matrices. Il s'agit d'une fonction qui appelle les différentes fonctions réparties dans les fichiers spécialement créés pour la résolution automatique.

- Le fichier « main.c » :



C'est le fichier qui centralise toutes les fonctions et c'est sur celui-ci que l'on lance le programme.

➤ **Difficultés rencontrées et solutions apportées**

Ce semestre étant plus chargé l'organisation a été plus difficile et l'emploi du temps n'a pas toujours été respecté. Une meilleure organisation nous aurait permis de faire le projet dans de meilleures conditions.

• **Conclusion**

➤ **Apprentissage sur l'organisation de travail**

Cela a été l'occasion de découvrir le travail de groupe par l'intermédiaire de GitHub : la publication de code, de branches ; l'envoi d'un code modifié et la réception du travail du partenaire. Dans l'ensemble, c'est apprendre à travailler à plusieurs et fusionner les codes en gardant une trace écrite. Même si on ne maîtrise pas à totalement l'outil, cela nous a permis d'avoir une bonne prise en main qui nous facilitera la tâche pour les futurs travaux de groupe.

➤ **Apprentissage sur le plan technique**

Sur le plan technique, la gestion des données dynamique a été cruciale. Se répartir le travail et décentraliser les fonctions dans des fichiers spécifiques pour être le plus efficace possible a été une bonne décision.