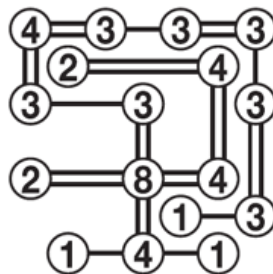


Projet Théorie des graphes: Hashiwakakero



Réalisé par :
Yacine BOUHALA

Responsable pédagogique :

Mme : **Carla Selmi**

Université des Sciences de Rouen

Table des matières

1 Introduction.....3

2 Définition du jeux.....3

3 Conception du jeux.....3

 3.1 Interfaces graphique3

 3.2 Partie back(algorithmique).....5

4 Bilan.....7

 4.1 Difficulté.....7

 4.2 Efficacité.....7

5 Conclusion.....7

6 Références8

1 Introduction

Dans ce projet nous essayons de réaliser une application solution du jeu Hashiwakakiro sous format numérique, le but de cette application est de permettre au Roi(le joueur) de relier tout son archipel d'îles par des ponts soit simples ou doubles pour pouvoir avoir accès à toutes ses îles, sans ce casser la tête, il aura juste à positionner les îles qui lui reviennent et l'application se chargera de trouver une solution de liaison entre les îles, si la solution existe bien sûr.

2 Définition du jeu

Le *Hashiwokakero* est apparu dans le magazine Puzzle en septembre 1990, bien qu'une version antérieure ait été publiée en 1989 par le japonais NIKOLI, ce jeu se joue sur une grille qui représente la superficie de l'archipel du Roi. On y retrouve des nombres de 1 à 8 inclusivement. Ils sont généralement encerclés et nommés *îles*. Le but du jeu est de relier toutes les îles en un seul groupe en créant une série de *ponts* entre les îles.

- Tout pont débute et finit sur une île.
- Deux îles ne peuvent pas être reliées par plus de deux ponts.
- Aucun pont ne peut en croiser un autre.
- Tous les ponts sont en ligne droite, à l'horizontale ou à la verticale.
- Le nombre de ponts qui passent sur une île est le nombre indiqué sur l'île.
- Toutes les îles doivent être reliées entre elles.

3 Conception du jeu

3.1 Interfaces graphique

Nous avons mis en œuvre une interface graphique qui sera la plateforme du jeu et qui représentera le point d'interaction avec l'utilisateur, lors du lancement de l'application une fenêtre comportant un champ texte et un menu déroulant, le champ texte nommé « **Number of Islands** » représentant le nombre d'îles que le joueur ou le Roi possède, il devra donc renseigner ce champ afin que le nombre d'îles soit sauvegardé dans le système et pour que le Roi ne se trompe pas lors du positionnement de ses dernières sur l'interface, car le système lui permettra de positionner seulement le nombre d'îles entré en paramètre. Quand au menu déroulant, ce dernier correspond à la superficie de l'archipel, par convention et pour faciliter les traitements on suppose que l'archipel a une forme carrée, le Roi choisira alors la superficie correspondant à son archipel, si le joueur se trompe quand il a fait rentrer ses informations, un bouton intitulé « **Reset** » est mis à sa disposition pour remettre les champs aux valeurs initiales et recommencer la saisie des données. Si l'utilisateur a fini la saisie il devra cliquer sur « **Confirm** » pour transmettre les données saisies au programme, si tout va bien et que le nombre d'îles est inférieur ou égal à la superficie de l'archipel, à noter que la superficie de l'archipel est calculée par L^2 , tel que L représente le nombre d'îles qu'on peut positionner sur la colonne et la ligne de notre carré qui représente une matrice par sa superficie, donc si c'est le cas le jeu commencera si non une erreur qui demandera à l'utilisateur de rectifier sa saisie surgira.

EXEMPLE :

Si l'utilisateur sélectionne 2X2 dans la liste déroulante et qu'il a dit qu'il a 5 îles, cela générera une erreur car on ne peut pas positionner 5 éléments dans une matrice 2X2, c'est impossible le nombre d'îles maximum que peut contenir un archipel de cette taille est seulement 4.

je vous laisse le soin de découvrir a quoi ressemble cette fenêtre ainsi que la gestion d'erreur lui correspondant dans les figures suivantes :



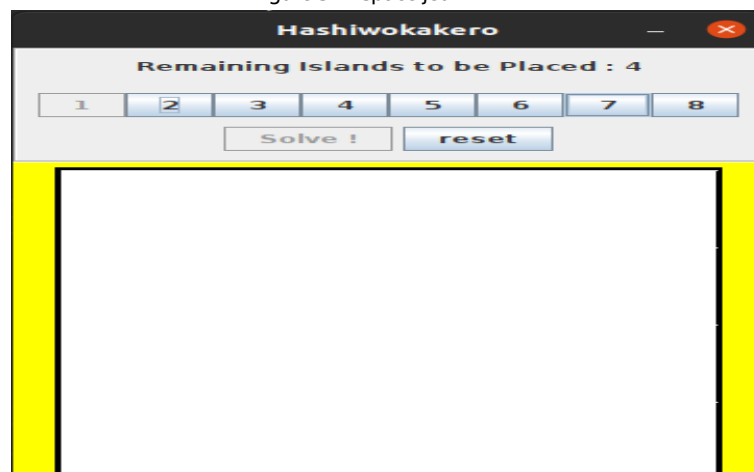
figure2 : gestion d'erreur



figure 1 : Accueil et choix

Dans le cas ou tout s'est bien passé et que le joueur ne s'est pas trompé dans la saisie de données, il sera automatiquement rediriger sur la plateforme du jeux une fois qu'il a transmis les donnée via le bouton dédié a cet effet qui est « Confirm », dans cet espace en haut de la fenêtre qui représente le nombre d'îles restant a placer ,ce nombre correspond initialement au nombre d'îles entré en paramètre dans la fenêtre prétendante, ce dernier décroît au fur et a mesure qu'une île est placé sur la map, juste en dessous une liste de bouton numéroté de 1 a 8, chaque bouton représente une île et le numéro sur le bouton représente le nombre de ponds que cette île peut prendre en charge, et enfin le joueur trouvera un espace blanc qui est la superficie de l'archipel et la ou il devra positionner ses îles en cliquant sur sur une position dans cette surface, il trouvera aussi deux boutons « Solve ! » qui sert a donner une solution si celle-ci est possible, et « Reset » qui ramènera le joueur a la première fenêtre et il devra recommencer le processus a nouveau. Je vous laisse le soin de découvrir cette fenêtre dans la figure suivante :

figure 3 : Espace jeux



Pour jouer, l'utilisateur devra en premier lieux sélectionner une île sur la liste de boutons pressente sur l'interface, une fois cela fait il devra cliquer sur une position de son choix sur l'espace blanc afin de positionner cette île, un cercle représentant l'île avec son numéro de pond se dessinera alors sur cette position, une fois cela fait vous remarquerez que le nombre d'îles restantes a positionner a diminuer de 1, l'utilisateur devra répéter ce processus un nombre de fois correspondant au nombre d'îles entré en paramètre, une fois qu'il a fini le positionnement de ses îles, le bouton « Solve ! » qui était initialement désactivé sera activé pour permettre a l'utilisateur de demander au programme de lui trouver une solution en

cliquant sur ce dernier, la figure suivante vous donnera un aperçu sur le processus décrit ce dessus :

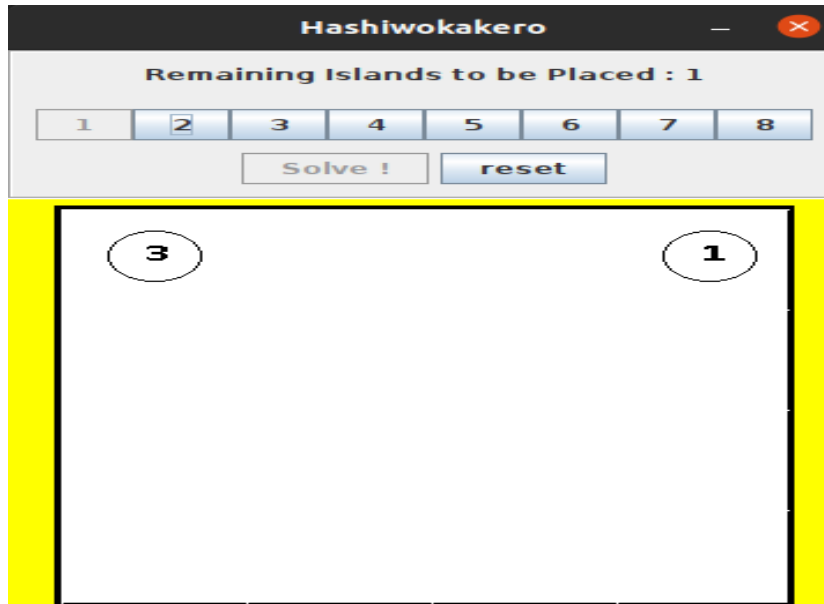


figure 4 : placement d'île

3.2 Partie back(algorithmique) :

une fois que toutes les îles positionnés et que l'utilisateur a cliqué sur « **Solve !** » le processus de résolution commence, l'algorithme du programme commence d'abord par parcourir toute la map un par un, et pour chaque île remplir la liste de ses voisins, et si par exemple on a une île qui supporte 8 pond et qu'elle a un nombre de voisins inférieur a 4, le programme ressortira directement et dira alors qu'aucune solution ni possible, dans le cas ou ce test est positif on rempli la liste des pond possible pour cette île si la somme des pond possible pour cette île est inférieur au nombre de ponds que cette île doit avoir cela arrêtera le programme il nous dira qu'aucune solution n'a était trouver si non mettre tout les ponds possible en versifiant que le pond rajouté ne crée pas un croisement avec un autre pond si c'est le cas ne pas ajouter ce pond, en règle générale j'ai utilisé le processus du backTraking, ce processus consiste a commencer son chemin sur une île voir tout ses voisins faire une configuration c'est a dire relier cette île a un de ses voisins puis aller sur cette île la relier a un de ses voisin puis quand on arrive a une île ou il n'y a plus de voisins vérifier si toute les ils sont parcouru et si elles sont relier avec le nombre de pond adéquat dans le cas contraire on reviens en arrière pour voir si y a une île dont on a pas relier tout ses ponds, on met un des ponds manquant et on continue sur l'île suivante, et on recommence le prouesses, si a l'issu de ce processus la grille n'est pas solutionné, on fait une autre configuration a l'île-1 si cela est possible, si non on supprime les pends on reviens a l'île-2 on lui fait une autre configuration, si toute les configuration de cette île mènent a une impasse on supprime aussi ses ponds et on passe a l'île-3 pour lui donner une nouvelle configuration et ainsi de suite.... Ce processus se répète récursivement jusqu'à ce que l'on trouve une solution, et dans ce cas une fenêtre va s'afficher pour dire que le programme a trouvé une solution a la grille, et les îles seront alors relié par des pond adéquats. Dans le cas ou l'algorithme du backTraking arrive a l'île de départ et il a fait toutes les configuration possible (combine de ponds) mais sans succès, l'application informera que la grille ne comporte pas de solution possible.

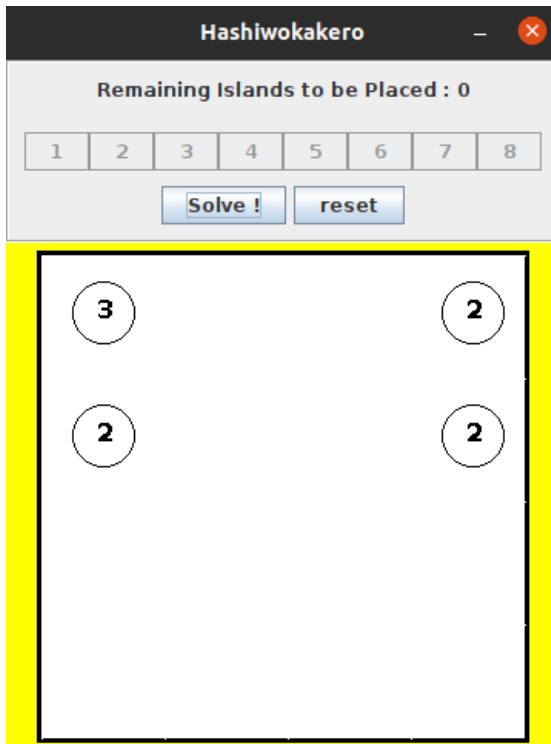


Figure 5 : exemple grille sans solution

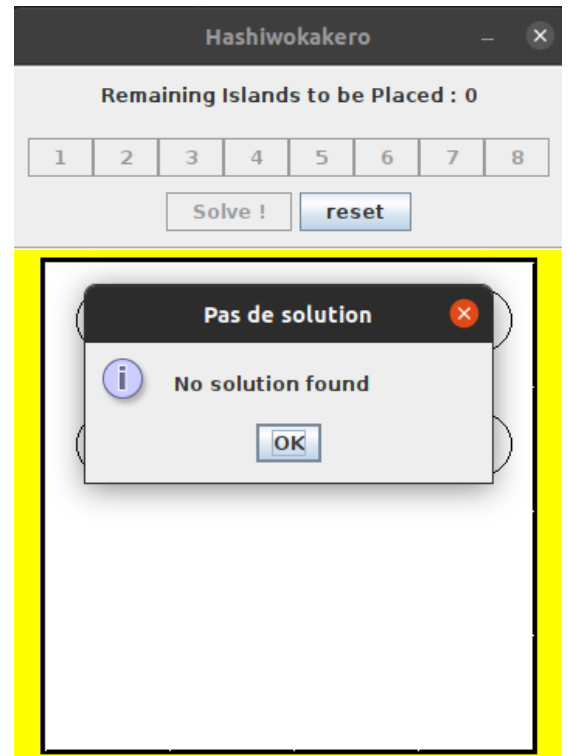


figure 6 : résultat grille sans solution

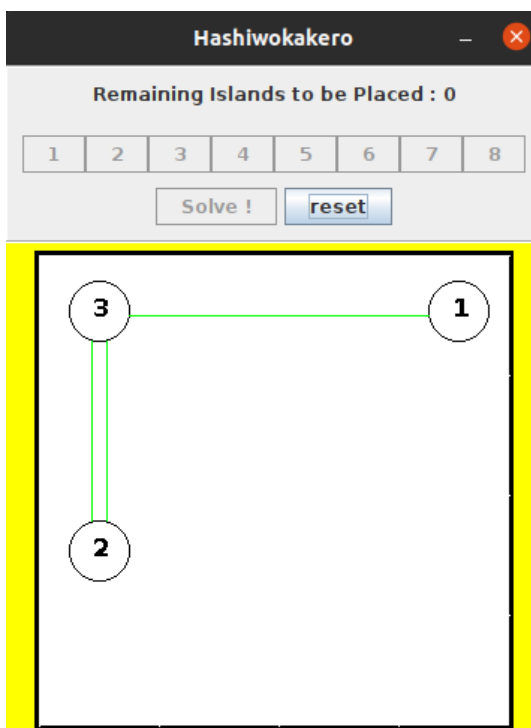


figure 7 : exemple grille avec solution

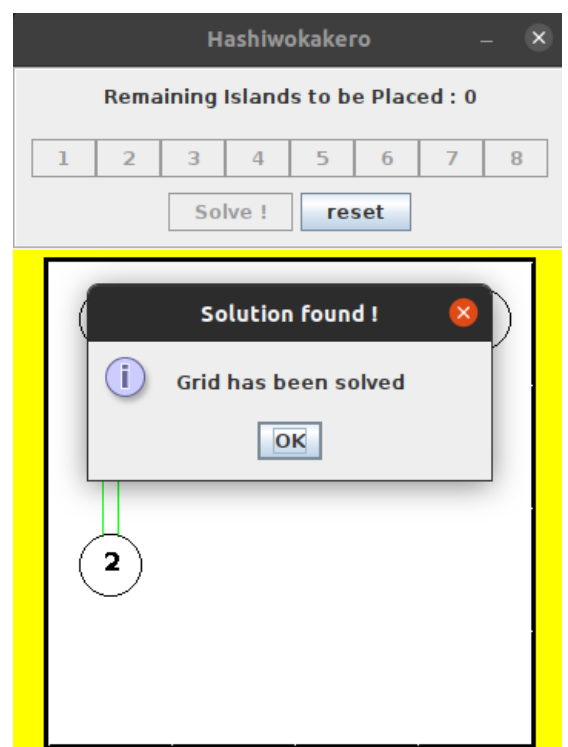


figure 8 : résultat grille avec solution

4 Bilan

4.1 Difficulté

La plus grande difficulté de ce projet réside sur le fait de trouver l'idée de base avec la quelle ce problème devrai être résolut la mise en point de l'algorithme et la logique du jeux qui fait appelé a la récursivité.

4.2 Efficacité

Le programme fonction parfaitement, et l'application repend au besoin client(projet), mais y a un petit problème de performance des fois le programme prend un peut de temps pour afficher les île lors du clic, il se peut donc que vous positionnez les îles et elle s'afficheront après un petit laps de temps ou une fois appuyé sur le bouton « solve !».

5 Conclusion

J'ai eu un réel plaisir à mettre en place cette application ,qui ma permis de mettre en pratique les notions vu dans le cour de Théorie des graphes notamment le parcours des graphes, il m' aussi appris a passer de la théorie a la pratique, comme il m'a appris a me surpasser, a me casser la tête pour trouver des solutions, il m'a aussi appris a chercher des repenses a mes question, mais le plus important dans tout ça s'est qu'il m'a appris a gérer mon stresse, car entre la préparation au examens, le stage et le boulot a coté, j'ai réussi a garder la tête froide, a sacrifier de mon temps de repos pour avancer dans ma vie, se sont la les bons points que je peux tirer de ce travail.

6 Références :

<https://en.wikipedia.org/wiki/Hashiwokakero>
<https://en.wikipedia.org/wiki/Backtracking>