Paul Goux
Jérémy Puertolas

Rapport de projet :

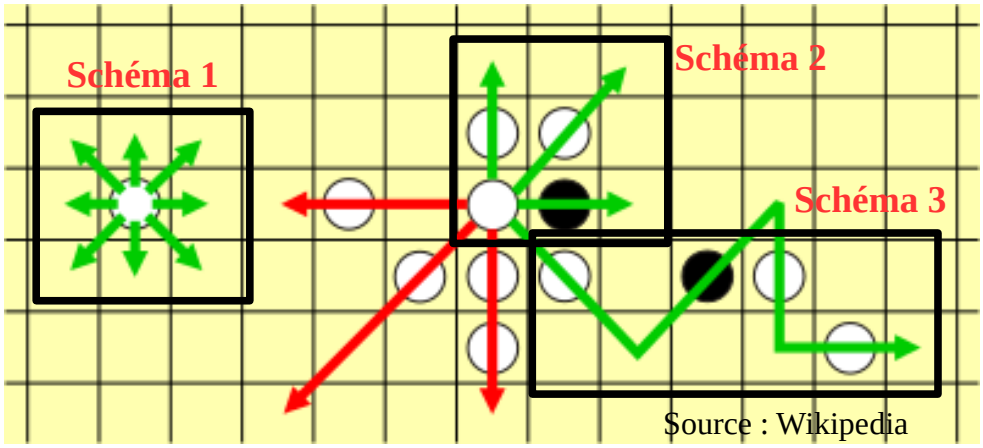
Notre projet porte sur le jeu de plateau « Halma » qui se joue à deux joueurs (dans notre cas). Nous avions pour objectif de créer ce jeu sur machine en langage C. Nous avons utiliser les logiciels de développement Emacs et Xcode.

Fonctionnalités implémentées :

Déplacement :

Chaque pion peut être simplement déplacé par les joueurs de case en case (Cf. Schéma 1). Le joueur choisit un pion qui lui appartient, et renseigne ensuite là où il souhaite que ce dernier se rende. Un joueur ne peut que bouger ses propres pions. Le saut est autorisé si un pion nous sépare d'une case libre à 2 cases de distance (Cf. Schéma 2). Il est possible de faire plusieurs saut de suites (Cf. Schéma 3).

Dans cette illustration, on imagine que nous sommes l'équipe aux pions blancs . Les flèches vertes indiquent les déplacements possibles, et les rouges ceux qui sont impossibles.



Source : Wikipedia

Changement de tour :

Lorsque qu'un joueur ne peut plus ou ne veut plus jouer, c'est au tour de l'autre joueur. A la fin de chaque tour (lorsque les deux joueurs ont effectué leurs coup), on passe au tour suivant.

Sauvegarde :

Après chaque tour, la partie est automatique sauvegardée. Tout les cinq tours, on demande au joueur 2 (avec l'accord du joueur 1 dans l'idéal) s'il veut quitter la partie. Si oui, le jeu s'arrête et ils pourront reprendre la partie ultérieurement.

Conditions de victoire :

La partie s’arrête lorsque l’un des deux joueurs réussi à placer tous ses pions dans la zone adverse. C’est ce dernier qui remporte la partie.

Organisation du programme :

Explication de notre démarche :

Nous avons choisi de créer plusieurs fonctions avec un rôle bien précis. Chaque fonction n’a que peu d’utilité si on la sort du programme. Cela rend le code plus simple à comprendre.

Fonction d’initialisation :

La fonction "initialiser" va ajouter des bordures au tableau servant à limiter les déplacements des pions sur le plateau. Une fois le plateau créé, on demande aux joueurs s’il veulent charger leur dernière partie. Si oui, les pions reprennent les positions qu’ils occupaient à la partie précédente. Sinon, les pions prennent les positions par défauts dans le coin supérieur gauche, et le coin inférieur droit.

Fonction d’Affichage :

La fonction "afficher" permet d'afficher le plateau sur le terminal. Pour cela elle traverse le tableau et affiche soit un espace (case vide), un & (pion bleu), un $ (pion rouge) ou un ‘.’(case où peut aller un pion). De plus, la fonction affiche le pion sélectionné en gras. Cette fonction affiche également le numéro du tour en cours, le joueur qui a la main, ainsi que les délimitations des différentes cases. On retrouve autour du plateau les indices de colonnes et de lignes.

Fonction du Pion de départ :

La fonction "saisir_pion_départ" demande au joueur de sélectionner un pion de sa couleur en saisissant ses coordonnées. Si les coordonnées sont incorrectes ou que le pion ne peut pas se déplacer, on demande au joueur de recommencer la saisie. Une fois un pion valide saisit, on renseigne un 2 dans les cases où il peut faire un pas et un -2 dans les cases où il peut faire un saut.

Fonction du pion d’arrivé :

La fonction "saisir_coord_arrive" permet à l'utilisateur de saisir les coordonnées où il souhaite déplacer son pion. La fonction vérifie que les coordonnées sont valides et que le pion puisse se rendre sur la case souhaité. Pour cela, la fonction regarde si dans la case souhaitée se trouve un 2 (pas) ou un -2 (saut).

Fonction de jeu:

La fonction "jouer" déplace le pion de sa case de départ vers sa case d’arrivé. Pour cela elle met le numéro du pion dans la case d'arrivé puis elle met un 0 (vide) dans la case de départ.

Fonction de victoire :

La fonction "tester_victoire" regarde le tableau et compte le nombre de pion du joueur en cours dans la zone adverse. Si ce nombre est 15 (soit tous les pions), elle retourne une valeur indiquant quel joueur a gagné. Sinon, elle retourne 0.

Fonction de sauvegarde :

La fonction "sauvegarder_partie" permet de sauvegarder la partie en cours dans le fichier "sauvegarde_halma.txt". Il sera ainsi possible de reprendre la partie ultérieurement, grâce à la fonction "charger_partie".

Organisation et répartition des tâches :

Durant nos heures de travail sur le projet Halma, nous répartissions les tâches de la manière suivante :

- 5/10 minutes pour savoir quoi faire et comment en tout début de session (« quelles fonctions ? modification dans le code existant ? ... »).
- moins de 5 minutes pour la répartition des tâches de manière équilibrée en fonction de la quantité de travail demandée et des envies.
- 15/30 minutes avant la fin : regroupement et intégration du travail dans le code source afin d'avancer, et nous terminions par un rapide débogage du programme (si bogs ils y avaient).

Cette répartition a permis a chacun d'avancer des parties du projet tout en restant connecté au reste du code global.

Bilan Qualitatif :

Difficultés rencontrées :

Nous avons eu des difficultés à faire fonctionner certains morceaux de code avec des procédés qui n’ont pas beaucoup été traités en cours, comme les « getchar », et la sauvegarde. Ces difficultés ont cependant été surmontées à l’aide de sites internet spécialisés et divers forums.

Points intéressants :

Le travail en groupe est un point positif, il nous a permis de travailler sur différentes parties du programme à deux. Le fait d'aller chercher de l'information par soit même était aussi très intéressant (notamment pour les « getchar », la mise en couleur des caractères, et la sauvegarde). C'était aussi très pertinent de voir que à peu près chaque groupe utilisaient une solution différente pour faire fonctionner son programme.

Mode d'emploi :

Après avoir lancé le programme avec la commande : prompt> ./Halma
un premier message apparaît.

```
MacBook-Air-de-Paul:Halma paulgoux$ ./Halma
voulez vous recommencer l'ancienne partie ? (O/N)
```

Illustration 1

En répondant oui ('O') à se message, le programme relancera l'ancienne partie. Il est à noté que lorsque l'on joue pour la première fois, répondre oui lancera l'application avec les pions à leurs places d'origines.

Après avoir répondu au message, le plateau s'affiche.

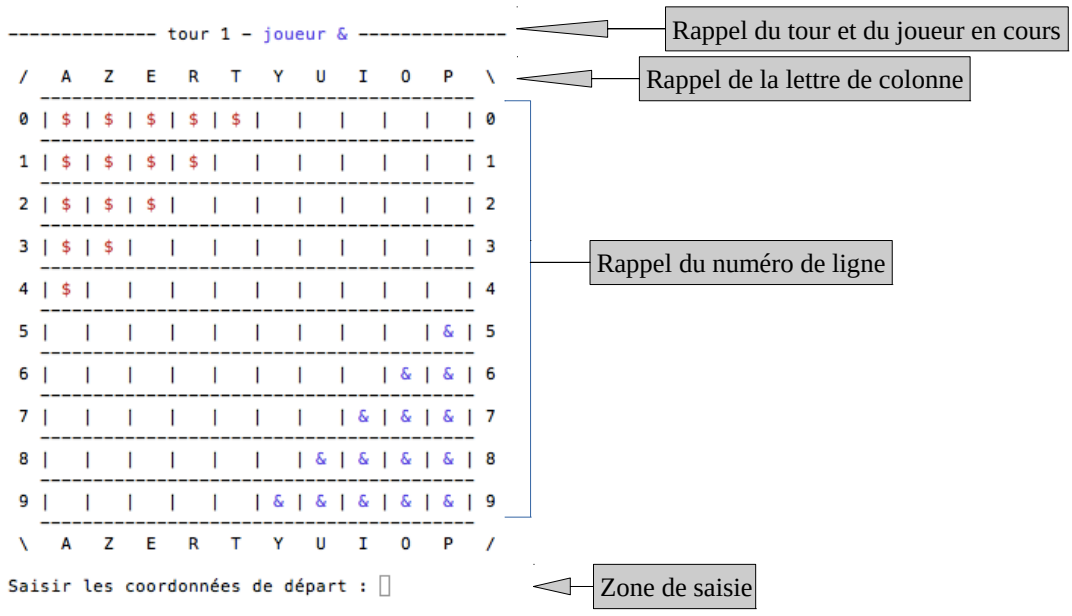


Illustration 2: plateau de départ

Au tour du joueur, celui-ci doit commencer par saisir les coordonnées du pions qu'il souhaite déplacer. Pour cela il renseigne dans le champ en bas les coordonnées du pion. Il est possible de renseigner d'abord la ligne puis la colonne, ou inversement. Pour valider tous les choix, il suffit d'appuyer sur entrée.

```
Saisir les coordonnées de départ : 9Y
```

Illustration 3: saisit des coordonnées de départ

Une fois le pion sélectionné, celui-ci devient gras sur le plateau. De plus, des points apparaissent dans les cases où peut se déplacer le pion choisit. Il suffit alors de renseigner les coordonnées de l'un des points.

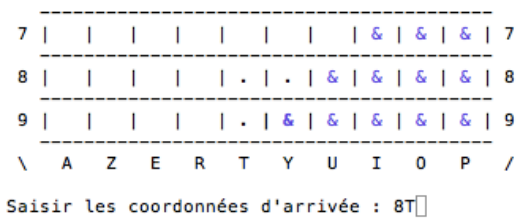


Illustration 4: saisit des coordonnées de départ

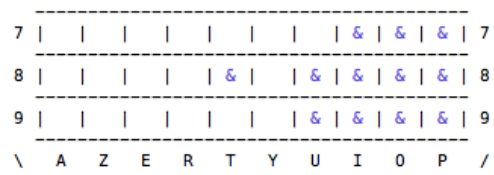


Illustration 5: déplacement du pion par un pas

Il est aussi possible de faire des sauts.

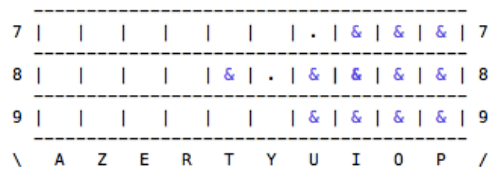


Illustration 4: saisit du pion

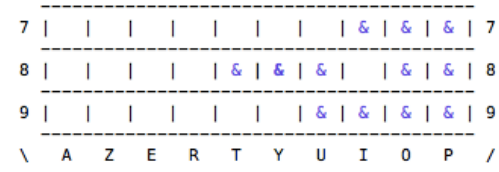


Illustration 7: déplacement du pion

Après un saut, le joueur peut continuer à jouer s'il le souhaite. Un message apparaît donc lui demandant s'il souhaite continuer à jouer.

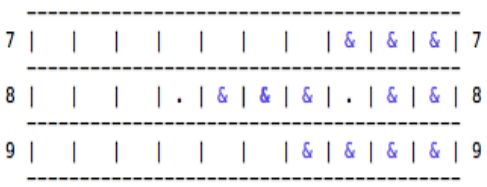


Illustration 8: 2e déplacement du pion

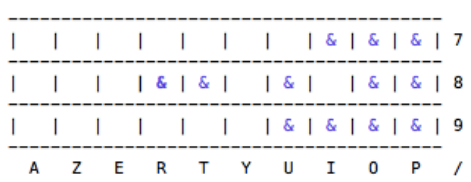


Illustration 9: arrivé du pion après son 2e saut

Tous les cinq tours, il sera demandé aux joueurs s'ils souhaitent ou non quitter la partie. Celle-ci sera enregistrer. Notez que la partie est aussi enregistrée à la fin de chaque tour.

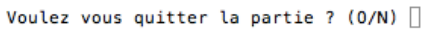


Illustration 10: quitter la partie en cours

Il sera possible de reprendre la partie ultérieurement grâce au premier message (cf Illustration 1).

Le jeu continue ainsi jusqu'à ce qu'un joueur place tous ses pions dans le camp adverse.

Illustration 11: le joueur & remporte la partie

Conclusion :

La réalisation d'un projet dans sa totalité est une nouveauté pour nous. Certains points techniques ont été intéressants à réaliser, comme la sauvegarde. Néanmoins, certains problèmes techniques (les « getchar » par exemple) ont nuit à l'avancement du projet. Travailler en groupe n’est pas forcément très simple, mais dans notre cas, nous avons essayer de nous organiser, et tout s’est bien déroulé.