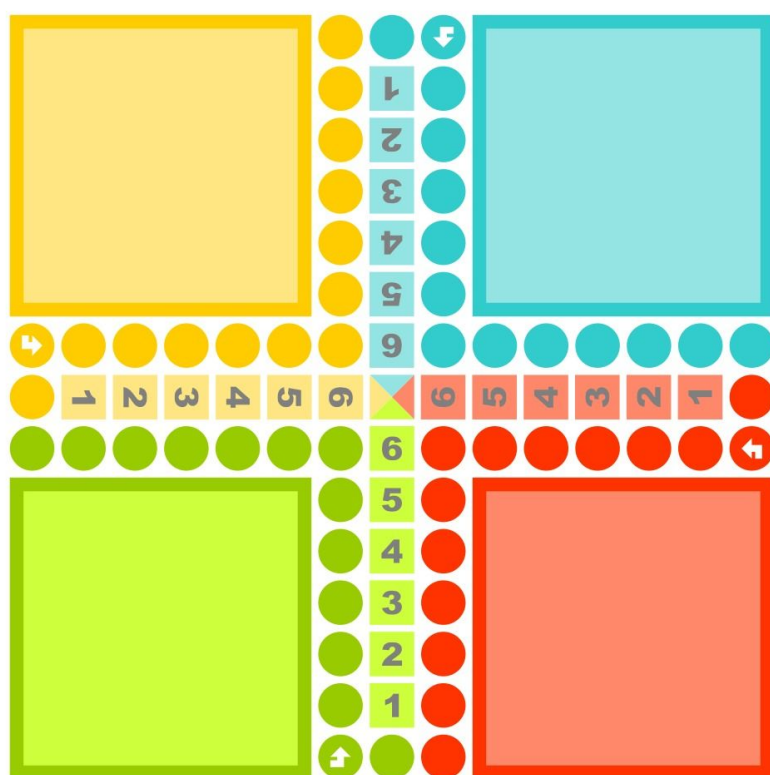


PROJET D'ALGORITHMIQUE

Petits Chevaux 2018/2019



LEPREVOST Loïck / LE CORFF Pierre

05/01/2019

Fonctionnalités du jeu:

Pour réaliser ce projet, nous avons suivi à la lettre les règles évoquées dans le document expliquant le but du projet et lorsque le document nous paraissait flou, nous avons tenté de coller au maximum au règle du véritable jeu des petits chevaux. Toutes les règles que nous appliquons sont expliquées dans le menu de démarrage dans lequel on peut également retrouver une présentation du plateau de jeu que nous avons voulu la plus fidèle possible et une option permettant soit de lancer une partie soit de quitter le programme.

Ainsi, notre jeu peut se jouer de 2 à 4 joueurs avec les écuries s'affichant uniquement en fonction de la sélection des joueurs. Nous avons également voulu que l'utilisateur ait la plus grande liberté d'action possible, c'est-à-dire que c'est lui qui choisit, via l'invité de commande, quel pion il veut faire évoluer sur le plateau de jeu, nous ne le forçons pas à jouer un pion précis.

Nous avons également établi une version graphique du lancé de dé remplaçant l'affichage classique textuel afin de rendre cela plus visuel et plus intuitif.

Organisation du code:

Notre code peut se découper en cinq parties majeures regroupant chacune plusieurs fonctions. Ces parties sont l'initialisation, le déroulement du jeu, la saisie des utilisateurs, l'affichage, qu'il s'agisse du plateau ou du dé et les fonctions gérant l'avancée du jeu.

La fonction principale dans le déroulement du jeu est la fonction **Jeu** qui liste les différents cas de figure dans lequel le jeu peut se dérouler tel que le tirage d'un 6, le déroulement du jeu si aucun pion n'est sorti ou au contraire si tous les pion sont sortis... Ainsi de cette fonction se découle deux autres fonctions, **Jeu_direct** et **Sortie** qui se composent de différentes autres fonctions et qui résument les deux seules actions que l'utilisateur peut effectuer c'est-à-dire soit jouer directement un pion soit sortir un nouveau pion.

Une autre fonction importante du projet est la fonction **printTab** mais sur un plan graphique dans le cas présent, il s'agit de la fonction définissant les couleurs ainsi que les symboles à afficher en fonction de la lettre/valeur contenu sur la case de la matrice. Tout

cet affichage est réalisé à partir de printf et par le biais de define qui représente spécifiquement un affichage de fond ou d'écriture de texte différent. Cette fonction se compose des fonctions *printTab4* et *affichage_plateau* qui permettent respectivement de définir des lettres précises aux cases contenant des pions pour pouvoir les afficher d'une couleur précise et de définir quelles écuries ont été sélectionnées et ainsi connaître quelles couleurs il faut afficher sur le plateau de jeu.

En ce qui concerne notre vision du jeu, nous avons voulu créer un tableau une dimension pour le chemin de jeu ce qui permet d'effectuer plus simplement les différents traitements en ce qui concerne les déplacements ou la vérification de la bonne application des règles. Nous avons ensuite relié ce tableau à une matrice en deux dimensions qui nous permet ainsi un affichage plus simple et avec plus de maîtrise selon nous.

L'aspect graphique et visuel était également très important pour nous, afin que le jeu soit le plus intuitif possible et facile à prendre en main en collant le plus possible au véritable jeu des petits chevaux.

Réalisation du code:

Pour réaliser notre projet, nous nous sommes répartis la tâche avec d'une part l'affichage du plateau et du dé et d'autre part toutes les fonctions correspondant au déroulement du jeu (déplacement, vérification de règles...). Cependant, il était impossible de ne pas travailler en lien puisque pour tester le bon fonctionnement du jeu, il est nécessaire d'avoir une interface graphique et inversement. C'est pourquoi nous avons travaillé en corrélation tout en s'entraînant chacun sur les problèmes que nous pouvions rencontrer.

Expérience personnelle:

Le jeu que nous proposons convient à nos attentes, nous avons réglé les différents bugs que nous avons rencontrés bien que d'autres soient sans nul doute présents. Nous sommes heureux de l'aspect global du projet puisque l'affichage est réussi et le jeu semble assez intuitif pour que l'utilisateur puisse apprécier d'y jouer.

Cependant nous pensons que nous avons un problème d'optimisation et cela est dû aux différentes versions du projet que nous avons entreprises puisqu'à la base nous n'avions pas du tout la même idée de réalisation du projet et par conséquent au cours du projet nous avons dû le modifier fortement entraînant une perte d'optimisation avec des

fonctions plus complexes que prévues.

Ce changement de réalisation du projet est notamment dû aux difficultés rencontrées lors de la réalisation de la fonction **Déplacement** puisque cette fonction était à l'origine composée de nombreux if imbriqués listant ainsi tous les cas de déplacements possibles. Cependant cette fonction était trop imposante c'est pourquoi Monsieur Héron nous a conseillé de passer par un tableau en une dimension qui décrirait le chemin de jeu et ainsi le déplacement consiste uniquement à faire déplacer le pion à travers ce plateau qui contient par avance toutes les coordonnées.

Par la suite, un autre problème auquel nous avons été confrontés, était l'affichage du plateau et principalement des couleurs puisque nous n'avions jamais géré l'affichage des couleurs en C sous Linux ce qui causa de nombreuses interrogations et souleva plusieurs problèmes que nous avons au final pu régler.

Les derniers problèmes que nous avons rencontrés, concernaient la fonction **Dépassement** puisque dans des cas très spécifiques, cette fonction autorisait des déplacements qui entraînaient le dépassement d'un pion par un pion d'une autre couleur. Pour résoudre ce problème, nous avons trouvé comme unique solution de créer des conditions très spécifiques pouvant être activées que lorsque le pion se trouve dans cette situation et ainsi un traitement est effectué sur le test du déplacement du pion.

Mode d'emploi:

Pour sortir un cheval il est nécessaire d'effectuer d'abord un 6:

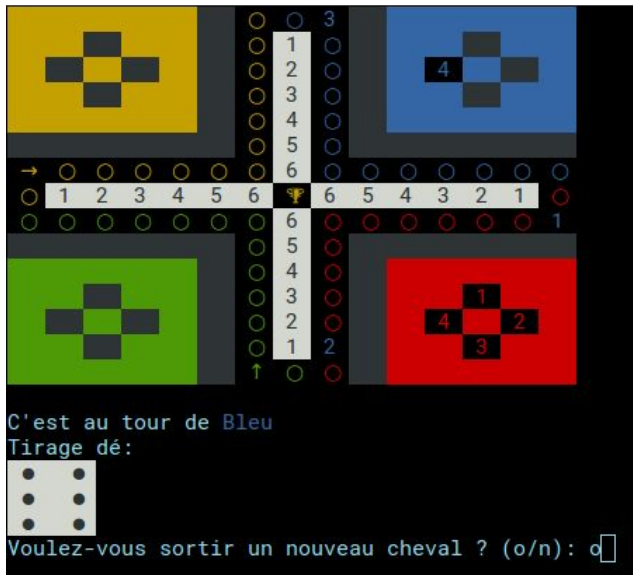


4



Cas particuliers:

Il n'est pas possible de sortir un cheval lorsqu'un cheval de même couleur est déjà sur la zone de départ:



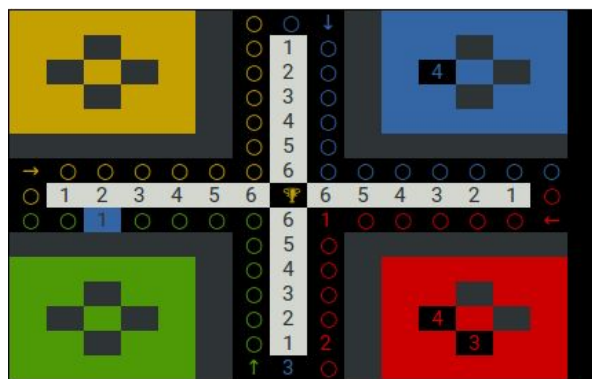
Deux chevaux d'une même couleur peuvent se dépasser:



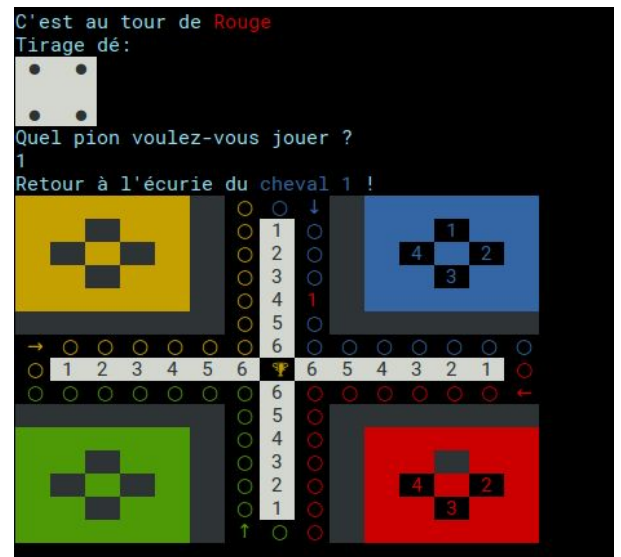
Cependant lorsque les chevaux ont une couleur différente, le déplacement est annulé:



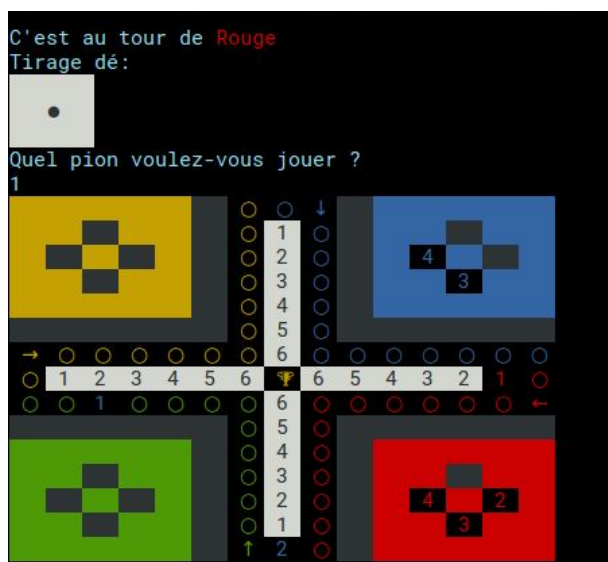
Deux chevaux de même couleur peuvent se superposer sauf sur l'échelle:



Cependant lorsqu'un cheval se déplace sur la case d'un cheval d'une autre couleur alors le cheval se déplaçant éjecte l'autre cheval qui retourne ainsi dans son écurie:



Pour gravir l'échelle, il faut tout d'abord effectuer un tour complet du plateau de jeu puis pour avancer sur l'échelle, il faut que le cheval fasse la valeur exacte de la case sur laquelle il veut se déplacer:



Conclusion:

Ce projet nous a permis d'en apprendre davantage sur la réalisation d'un projet en équipe puisque nous avons travaillé sur des parties séparées du code et nous avons eu besoin de réutiliser des éléments de notre binôme ce qui nous a obligé à communiquer pour comprendre à quoi servaient ces éléments, comment ils avaient été créés...

Cela nous également permis de nous rendre compte que ce type de projet est extrêmement chronophage et que par conséquent il est nécessaire de s'y prendre à l'avance puisqu'il est toujours possible qu'un contretemps survienne retardant ainsi toute la réalisation du projet.

Au niveau technique, nous avons beaucoup appris sur l'utilisation des pointeurs puisqu'ils composent l'ensemble de notre code. Nous avons également découvert l'affichage des couleurs dans la console permettant ainsi de faire un affichage graphique qui nous convient et enfin nous avons pu réellement utiliser la notion de Makefile.