

# Module 1106 – PROJET

## ALGORITHMIQUE-PROGRAMMATION

### JEU DES PETITS CHEVAUX

Samir Loudni, IUT CAEN – Département Informatique (Version 1.1)

4 novembre 2018

## 1 Description du projet

Les Petits chevaux est un jeu qui se joue sur un plateau de  $15 \times 15$  cellules, constitué d'une piste de jeu, d'un dé, de quatre écuries de couleurs différentes et de 16 pions-chevaux (4 par couleur), comme indiqué par la Figure 1.

**Préparation :** Chaque joueur choisit sa couleur, et pose ses quatre pions dans l'écurie de sa couleur. Le premier joueur est tiré au sort. Les autres suivent, chacun à son tour, dans le sens des aiguilles d'une montre.

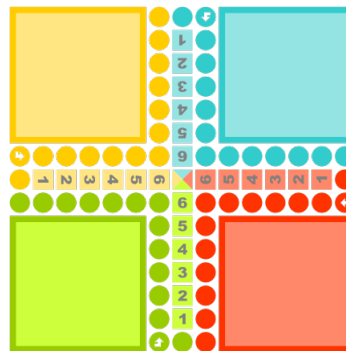


FIGURE 1 – Plateau de jeu des petits chevaux (URL : [https://fr.wikipedia.org/wiki/Jeu\\_des\\_petits\\_chevaux](https://fr.wikipedia.org/wiki/Jeu_des_petits_chevaux)).

**Déroulement du jeu :** Pour sortir un pion de l'écurie et venir le placer sur la première case de son parcours, le joueur doit faire un 6 avec le dé. S'il n'a pas fait 6, il passe son tour. Pendant la partie, il faudra obligatoirement faire un 6 pour mettre un nouveau pion sur la piste. Les règles ci-dessous définissent le déplacement des pions sur le plateau.

- Une fois un pion sorti, le joueur le fait avancer du nombre de cases égal aux pions du dé. S'il a plusieurs pions sortis, le joueur peut faire avancer, au choix, n'importe lequel de ses pions en piste.
- Un joueur qui fait 6 rejoue automatiquement une deuxième fois et peut, au choix, faire sortir un cheval ou avancer de 6 cases.
- Un pion dont la route est barrée par un autre pion adverse ne peut se déplacer et passera son tour si les points du dé l'amène au-delà du barrage.
- En revanche, si le tirage au dé permet à un joueur de placer son pion exactement sur une case déjà occupée par un pion adverse, il renvoie le pion de l'adversaire à l'écurie et prend sa place. Celui qui retourne à l'écurie repart donc de zéro et doit obtenir de nouveau un 6 pour sortir. Toutefois, deux pions de la même couleur peuvent se poser sur la même case.

**Placement des pions dans les cases numérotées :** A la fin d'un tour complet du jeu, les pions se placent sur les cases numérotées de 1 à 6. Chaque pion avance du nombre de cases égal aux points du dé sur la case numérotée correspondante, à condition que celle-ci soit libre. Il est également possible de déplacer un pion à l'intérieur des cases numérotées pour libérer de la place et faire entrer un autre pion. Si aucun déplacement n'est possible, le pion passe son tour.

**Fin de la partie :** Le premier joueur qui réussit à ranger ses quatre pions dans les 4 cases numérotées le plus près du centre du plateau de jeu gagne la partie.

## 2 Notes de mise en œuvre

**Votre programme devra permettre de visualiser une séquence de jeu en affichant à chaque tour : le plateau de jeu, les déplacements des pions du joueur.**

Voici quelques indications des points à réaliser :

- initialisation du plateau de jeu
- gestion du début d'un tour de jeu (déterminer qui commence)
- lancé du dé et choix du pion à déplacer
- validation d'un déplacement (est-il valide ? sinon retour en saisie)
- déplacer le pion sur la case associée, retirer le cas échéant les pions mangés et retour en saisie, ou début du tour pour le joueur suivant
- vérification de fin de partie

Dans un premier temps, définissez les principales structures de données dont vous allez avoir besoin pour représenter les éléments du jeu : plateau de jeu, informations sur les joueurs et gestionnaire de jeu (contenant les informations globales concernant l'état du jeu courant). Pour chaque structure de données, définissez les fonctions associées.

Dans un deuxième temps, définissez la boucle de contrôle global du jeu qui assure les différentes phases du jeu. **Il est conseillé d'avoir en permanence un programme qui compile et qui définit une version partielle du jeu.** Seul devra être réalisée une version textuelle de ce jeu. Le programme devra permettre de visualiser la partie en console.

**Aller plus loin.** Toute initiative personnelle (et justifiée) sera prise en compte et valorisée. Il existe de nombreux moyens d'étendre le projet. Une possibilité appréciable est de pouvoir interrompre une partie et la sauvegarder pour la reprendre ultérieurement. Un menu devra donc comporter les options "nouvelle partie", "sauvegarder la partie", "charger une partie en cours". Une seconde possibilité est de développer une version du jeu en partie rapide, en utilisant 2 dés au lieu d'un (voir l'URL <https://www.regles-de-jeux.com/variantes-des-petits-chevaux/> pour plus de détails.). Une troisième possibilité est d'ajouter une intelligence artificielle (c'est à dire la possibilité de jouer avec la machine).

## 3 Travail à rendre

Le projet est à réaliser en binômes (ou, avec l'accord de votre chargé de TP, en monôme). Le travail à rendre est un projet sous forme d'une archive zip à **déposer sur la plateforme Moodle**.

La procédure pour déposer votre projet est la suivante :

- sélectionner Université Caen Normandie puis renseigner les identifiants et mot de passe de votre messagerie universitaire.
- cliquer sur le lien Dépôt de fichiers. Télécharger depuis votre ordinateur le fichier à transmettre dans le champ Parcourir puis Envoyer.
- l'envoi du document génère une fenêtre avec un lien direct vers celui-ci. Copier ce lien pour l'insérer dans un mail adressé à votre enseignant chargé de TP.

Le nom de l'archive doit avoir la forme suivante : `Nom1Nom2.grTP.zip` ou `Nom1.grTP.zip` où `Nom1` et `Nom2` sont les noms de famille des membres des polynômes et `grTP` est le nom du groupe de TP auquel ils appartiennent (1.1, 1.2, etc.).

**Note** : Lors des séances de TP, les enseignants chargés de TP suivront l'avancement de votre projet et pourront vous aider sur certains points difficiles.

### Ce qu'il faut rendre :

- Le code source complet de votre application en C (largement commenté) qui compile sous **Linux** avec sa documentation d'installation et d'utilisation.  
**ATTENTION** : si votre programme fait appel à d'autres bibliothèques externes (comme par exemple la bibliothèque SDL), il est impératif de les inclure pour les besoins de test.
- Un court rapport d'une longueur comprise entre 3 et 10 pages présentant :
  - les fonctionnalités implémentées (très brièvement).
  - organisation du programme : découpage en fonctions, rôle des fonctions principales que vous avez utilisées, explications de votre démarche.
  - l'organisation et la répartition des tâches au sein du binôme durant la durée du projet (brièvement).
  - bilan qualitatif du travail, difficultés rencontrées, points qui vous ont paru intéressants.
  - un mode d'emploi avec quelques illustrations (p. ex. capture d'écrans, scénario d'exécution...), destinées à montrer l'opérationnalité de votre application.
  - Une conclusion sur l'apport (ou non) du projet en termes technique, scientifique, humain.

**Le code source ne doit pas faire partie du rapport (sinon en annexe).**

## 4 Évaluation du projet et calendrier

Le projet est à rendre le **le samedi 5 janvier 2019 avant minuit** (tout retard conduira à des pénalités). L'évaluation sera réalisée en fonction des critères ci-dessous :

- *qualité technique du code* : découpage en fonctions (préférez les fonctions courtes avec un rôle précis), modularité, un style de codage clair et cohérent, gestion des erreurs lors de saisies (les entrées erronées de l'utilisateur ne doivent pas interrompre le jeu).
- *lisibilité du code* : présentation du programme (indentation), usage de variables et de fonctions ayant des noms explicites, commentaires des parties clés de votre code . . .
- *documentation fournie* : organisation du programme et son mode d'emploi, bilan.
- *présentation orale* : démonstration du programme et questions sur le travail réalisé.
- *qualité de la démo* : prévoir plusieurs scénarios permettant d'illustrer le bon fonctionnement de différentes séquences de jeu (déplacement valides/invalides des pions, déplacement d'un pion sur une case occupée, déplacement d'un pion dont la route est barrée, etc) ainsi que la fin de partie (cas de victoire).