

# Module 1106 : PROJET

## ALGORITHMIQUE-PROGRAMMATION

### SHING SHANG

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

8 novembre 2015

## 1 Description du projet

Le Shing Shang est un jeu de stratégie qui se joue sur un damier de 84 cellules avec  $2 * 12$  pièces (Bushis).

**Préparation :** Chaque joueur une armée de 12 Bushis en une couleur (rouge ou noir). Cette armée est composée de 3 groupes de Bushis : 2 Dragons, 4 Lions et 6 Singes. La différence entre les groupes est indiquée par sa grandeur (les Singes étant les plus petits Bushis, les Lions sont les Bushis moyen et les Dragons correspondent aux plus grands Bushis). Chaque joueur place son armée dans les coins du damier (voir le schéma à l'URL : [http://jeuxstrategieter.free.fr/Shing\\_shang\\_complet.php](http://jeuxstrategieter.free.fr/Shing_shang_complet.php)).

**Déroulement du jeu :** Les joueurs jouent à tour de rôle. Lors de son tour de jeu un joueur doit déplacer l'un de ses Bushis de deux façons :

- Glisser vers une case suivante vide horizontalement, verticalement ou en diagonal, aussi bien en avant qu'en arrière.
- Sauter au dessus d'un Bushi si celui-ci est de la même grandeur ou plus petit que le sauteur. Pour pouvoir sauter, le Bushi doit se trouver sur une case contiguë à une case occupée par un Bushi rouge ou noire, il peut sauter la pièce, verticalement, horizontalement ou en diagonale, à condition que la case suivante soit vide. On peut enchaîner plusieurs sauts au cours d'un même tour. Cet enchaînement de sauts s'appelle un Shing Shang.

Les règles suivantes complètent celles indiquées au-dessus :

- **Les singes peuvent** se déplacer de une ou deux cases dans n'importe quelle direction, horizontalement, verticalement ou en diagonal mais sans changer de direction au cours du tour.
- **Les Lions peuvent** se déplacer de une case dans n'importe quelle direction, horizontalement, verticalement ou en diagonal.
- **Les dragons ne peuvent** se déplacer qu'en sautant.

Si au cours d'un Shing Shang on saute par dessus un Bushi adverse, on doit s'arrêter et le Bushi de l'adversaire est retiré du plateau. Toutefois, on a le droit à un tour de jeu supplémentaire avec un autre Bushi. Voir les exemples ci-dessous :

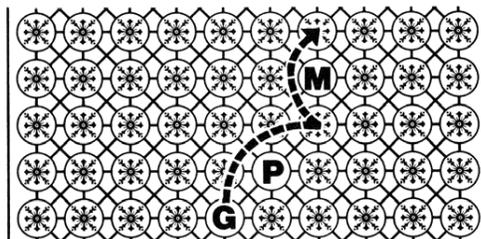


FIGURE 1 – Le grand Bushi saute au-dessus 2 Bushis à soi. Fin du tour de jeu.

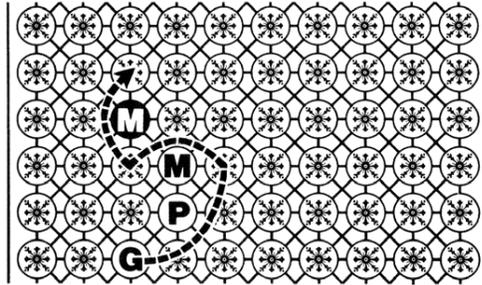


FIGURE 2 – Le grand Bushi saute au-dessus 2 Bushis à soi et au-dessus d'un Bushi de l'adversaire. Le joueur a droit à un autre tour de jeu avec un autre Bushi.

**Fin du jeu :** Un joueur gagne la partie lorsqu'il parvient à amener l'un de ses dragons sur l'un des portails (i.e. deux cases spéciales) de son adversaire ou qu'il capture les deux dragons de son adversaire.

**Votre programme devra permettre de visualiser une séquence de jeu en affichant à chaque tour : le plateau de jeu, les déplacements et/ou sauts courant du joueur.**

Voici quelque indications des points à réaliser :

- placement des pièces ;
- gestion du début d'un tour de jeu (déterminer qui commence ?)
- saisie d'un coup (donc choix d'une pièce puis séquence de déplacements)
- validation d'un coup (est-il valide ? sinon retour en saisie)
- déplacer les pièces, retirer les pièces mangées et retour en saisie, ou début du tour de l'adversaire si pas de pièce mangée
- vérification de fin de partie

## 2 Notes de mise en œuvre

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, définir les données correspondantes et les fonctions associées.

Dans un deuxième temps, définir la boucle de contrôle globale du jeu qui assurent 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.

**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 graphique de votre programme. Il est toutefois préférable dans un premier temps de définir une version non graphique afin de bien séparer les fonctions associées au jeu et l'affichage graphique du jeu.

## 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 Filex à l'adresse suivante : <http://ent.unr-runn.fr>.

La procédure pour déposer votre projet est la suivante :

- sélectionner Université de Caen Basse Normandie puis renseigner les identifiant 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 2 janvier 2016 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 réalisée.