

Shing Shang

— Les fonctionnalités implémentées.

- Affichage d'un plateau
- Choix de l'équipe qui commence
- Sauvegarde du plateau et du joueur en train de jouer
- Choix de reprendre une sauvegarde ou de créer une nouvelle partie
-

— organisation du programme : découpage en fonctions, rôle des fonctions principales que vous avez utilisées, explications de votre démarche.

- 1 fichier utiliser pour la fonction main
- 4 fichier utiliser pour la déclaration des fonctions

Les fonctions principaux sont pour :

- L'affichage
- La création des pions et du plateau
- La saisie des cases de départ et d'arrivée
- La sauvegarde
- Rejouer
- Déplacer le pion

— l'organisation et la répartition des tâches au sein du binôme durant la durée du projet

— 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.

trucs à essayer d'ajouter:

“ laisser la machine choisir “ (random pour le joueur qui commence)

mettre des noms de joueurs

commenter le programme

mieux compiler le programme (modifier les #include dans chaque .c)

vérifier qu'il existe un fichier sauvegarde avant

ajouter un truc qui donne un lien vers les règles du jeu au début de la partie - Je sais pas comment faire

fin de partie:

joueur gagnant

nb de coups totaux dans la partie

nb de pions sautés

nb de pions restants pour chaque joueur

oublie pas d'ajouter dans le dossier ce que t'ajoutes dans le programme !!

(je deteste l'économie)

Je te comprend =)

Projet Tutoré

Dut informatique 1ère année

Shing Shang

Bossard Valentin

Melo da Silva Alexis

TP2.2

Sommaire

- 1) Fonctionnalités implémentées
- 2) Organisation du programme
- 3) Répartition des tâches
- 4) Bilan qualitatif du travail
- 5) Mode d'emploi illustré
- 6) Ressentis personnels
- 7) Annexes

Introduction

Au cours de notre première année de DUT Informatique nous avons été amenés à la réalisation d'un projet tutoré en binôme, qui consiste en la programmation d'un jeu de plateau multijoueur : Le Shing-Shang. Le dossier que voici va permettre de se rendre compte de l'organisation pour la réalisation du projet au sein de notre binôme, de comprendre le jeu en lui même et d'apporter nos ressentis à la sortie de cette expérience.

Les fonctionnalités implémentées

Le programme a été entièrement codé en C et est fonctionnel sur le terminal sous Linux. Il inclut les bibliothèques Linux `stdio.h`, `stdlib.h` et `string.h`.

Notre programme possède toutes les fonctionnalités suivantes :

- Un affichage du plateau à chaque tour avec un terminal propre qui est nettoyé à chaque tour (commande "clear")
- Une fonctionnalité multijoueur local
- Un choix pour déterminer le joueur qui commence
- La possibilité de sauvegarder une partie (position des pions et le tour du joueur)
- Un choix entre la création d'une nouvelle partie ou bien la reprise d'une sauvegarde existante

Organisation du programme

Pour clarifier notre programme nous avons décidé de le couper en plusieurs fichier .c accompagnés de leurs header, le tout sera compilé ensemble et le jeu se lancera depuis un programme principal contenant le `main()`.

Programme principal - `shing_shang.c`

On y définit les variables et les pointeurs qui seront utilisés par toutes les fonctions du programme. C'est aussi dans le programme principal que l'on fait appel aux

fonctions d'initialisation du plateau, de la création des pions et toutes les fonctions permettant de jouer. Une boucle permet d'enchaîner les tours et se termine quand la fonction "gagner " atteint la valeur 1, c'est à dire qu'un dragon se pose sur un drapeau ou qu'il n'y a plus de dragons dans une équipe. Une fois sorti de cette boucle un petit message de fin s'affiche et la partie est terminée.

Sous programmes

structure.h

Il sert à définir les structures du plateau et des pions et à définir les énumérations des types de pion et des couleurs

deplacement.c et .h

Il sert à définir les fonctions qui déplacent les pions et qui les suppriment, ainsi que les fonctions permettant de rejouer.

verif.c et .h

Il sert à définir toutes les fonctions qui vont vérifier si les déplacements sont autorisés ou non, c'est à dire vérifier que le pion de départ est valide, que la case d'arrivée est valide et que le saut est possible. Il permet aussi de vérifier qu'il existe un fichier de sauvegarde dans l'hypothèse où le joueur souhaiterait reprendre une partie existante. Pour faciliter le code on a décidé de découper les vérifications de déplacement entre chaque type de pion (singe, lion, dragon), pour éviter les répétition on vérifie au tout départ que la case d'arrivée est vide et dans le cas échéant si c'est un portail et que le pion est un dragon adverse.

initialisation.c et .h

Il permet de créer le plateau en insérant chaque pion dans chaque cellule au départ. Il permet aussi de créer les pions, de leur associer une couleur. On y a placé une fonction void qui permet d'afficher le plateau. Les fonctions qui gèrent la sauvegarde et continuer une partie déjà existante à partir d'un fichier sont aussi placées ici.

tour.c et .h

Il contient toutes les fonctions nécessaires pour la saisie de l'utilisateur ; le pion de départ, la case d'arrivée, les affichages de questions qui demandent quel joueur commence, si on veut continuer une partie, rejouer. Les fonctions de fin de partie

sont aussi placées ici en plus de la fonction permettant de changer de joueur à chaque tour.

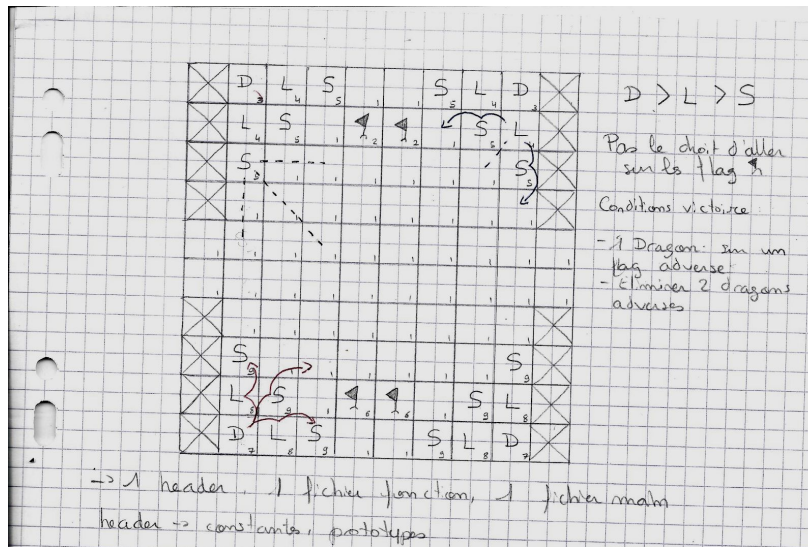
Pour éviter de tout compiler en écrivant toutes les lignes à chaque fois on s'est empressé d'écrire un petit script qui contient toutes les lignes nécessaires à la compilation, cela nous permet de seulement écrire " sh run.sh " (run étant le nom du script) depuis un terminal se trouvant dans le dossier du programme, pour faciliter le debugging et lancer le programme plus rapidement.

Répartition des tâches

Au début quand le projet nous a été donné nous sommes partis sur des tentatives de travail chacun de notre côté en répartissant les tâches entre nous et en essayant plusieurs logiciels permettant de coder sur des fichiers partagés en ligne tels que <https://codeshare.io/> ou <http://collabedit.com/>. Au final travailler à deux sur le même fichier était trop compliqué et essayer de répartir les tâches était tout aussi compliqué car il fallait s'accorder sur beaucoup de points comme le nom des fonctions, variables et d'autres choses pour garder une homogénéité du code. De plus on ne pouvait pas tester les fonctions et le programme car il fallait mettre en commun à chaque fois. On a donc essayé de travailler en binôme sur le même ordinateur et nous avons remarqué que nous étions beaucoup plus efficaces comme ça alors la majorité du travail a été faite ensemble à l'IUT en dehors des heures de cours. Lorsque nous rencontrions un problème que nous n'arrivions pas à résoudre nous travaillions chez nous le soir individuellement et nous faisons un point le lendemain ou du moins la prochaine fois où nous travaillions sur le projet pour essayer de proposer les solutions qu'on avait trouvé et les comparer. La plupart du temps le travail était donc fait en duo sans réelle répartition du travail car les problèmes et les solutions nous venaient au fur et à mesure et travailler ensemble sur les mêmes choses au même moment permettait d'avoir deux avis différents et détecter les erreurs plus rapidement, plutôt que de travailler séparément et revenir ensuite sur des morceaux de programme isolés pour les améliorer.

Bilan qualitatif du travail

Au début du projet nous avons des difficultés pour se lancer dans l'écriture du code sur machine directement et comprendre les règles du jeu (surtout les déplacements) alors on a fait quelques croquis sur papier pour les fonctions, les déplacements, et pour visualiser à quoi devait ressembler le jeu au final.



Par exemple la première chose qu'on a fait c'est de dessiner le plateau avec les déplacements de pions possibles et on avait tout le temps la feuille sous les yeux pour mettre les nouvelles règles dans un coin au fur et à mesure

On a eu du mal avec la notion de structure qui nous a été introduite après que le projet soit lancé. On a donc essayé d'en incorporer dans notre programme même si on ne savait pas trop comment cela fonctionnait au début. On a eu pas mal de soucis pour allier les structures et les pointeurs, on arrivait pas à aller chercher les données des structures avec leurs pointeurs, mais au fur et à mesure que l'on travaillait sur le projet et grâce aux TDs nous y sommes arrivés.

On a aussi eu des difficultés quand il fallait vérifier si le déplacement de pions était possible car ils ont tous des paternes différents donc on a décidé de découper la vérification des déplacements en trois fonctions, une pour chaque type de pion.

Mode d'emploi illustré

Notre jeu respecte la majorité des règles standards du shing shang mais nous avons du effectuer des choix pour des particularités du jeu qui n'étaient pas spécifiées dans les règles de base et sur lesquelles les professeurs ne s'accordaient pas toujours.

Les règles de base et le principe du shing shang globalement est expliqué ici : http://jeuxstrategieter.free.fr/Shing_shang_complet.php

Voici les spécificités de notre version du jeu :

- Les couleurs des joueurs ne sont pas blanc et noir mais bleu et rouge
- Les pions sont représentés par les initiales de leur type (ex : L pour Lion)
- On ne peut pas deviner la “force” d’un type de pion grâce à sa taille donc il faut connaître ce détail avant de jouer.

Préparation :

- Chaque joueur possède 12 bushis
- Ces bushis sont :
 - a. 2 Dragons
 - b. 4 Lions
 - c. 6 Singes
- Ces bushis ont des puissances différentes, le dragon est plus puissant que le lion qui est lui-même plus puissant que le singe

Déroulement du jeu :

1. Un bushi peut :

- Se déplacer aussi bien en avant qu’en arrière
- Se déplacer horizontalement, verticalement, en diagonale
- Sauter un autre bushi si celui-ci est plus petit ou de puissance égale
- Sauter un autre bushi si celui-ci est sur une case contiguë et si la case suivante est vide
- Enchaîner plusieurs sauts

2. Déplacement des bushis :

- Les singes peuvent se déplacer de une ou deux cases
- Les lions peuvent se déplacer de une case
- Les dragons ne peuvent que sauter

3. Saut

Si pendant un tour, on réalise un Shing Shang (un enchainement de sauts), et que l’on saute par-dessus un bushi adverse, celui-ci est retiré du plateau puis le joueur à le droit de rejouer avec un autre de ses bushi.

Fin du jeu :

Un joueur gagne quand :

- Un de ces dragons arrive sur le portail adverse
- OU
- Il capture les deux dragons de son adversaire

Comment jouer :

- Ouvrir un terminal depuis le répertoire du programme
- Exécuter le programme principal en écrivant " sh run.sh "
- Choisir si on veut reprendre une partie sauvegardée ou commencer une nouvelle partie (les choix se font en écrivant 0 ou 1)
- Dans le cas où on choisi une nouvelle partie, on défini quel joueur commence
- Si on veut sauvegarder le jeu il suffit d'écrire les coordonnées 0,0
- Le plateau s'affiche

	0	1	2	3	4	5	6	7	8	9
0		D	L	S	x	x	S	L	D	
1		L	S	x	P	P	x	S	L	
2		S	x	x	x	x	x	x	S	
3		x	x	x	x	x	x	x	x	
4	x	x	x	x	x	x	x	x	x	x
5	x	x	x	x	x	x	x	x	x	x
6		x	x	x	x	x	x	x	x	
7		S	x	x	x	x	x	x	S	
8		L	S	x	P	P	x	S	L	
9		D	L	S	x	x	S	L	D	

Les numéros représentent les coordonnées, les D les dragons, L lions, S singes, P portails et les X les cases libres où les pions peuvent se déplacer.

- Il faut entrer les coordonnées du pion à déplacer en écrivant la coordonnée x, appuyer sur entrée, puis entrer celle en y. Pareil pour la case d'arrivée. Le programme tourne en boucle tant que les coordonnées ne sont pas valides
- Pour savoir quel joueur doit jouer il suffit de regarder la couleur de la ligne qui demande les coordonnées
- Quand la partie est terminée un petit message apparaît affichant le vainqueur et quittant le jeu

Conclusions personnelles

Conclusion de Valentin :

J'ai apprécié travailler sur ce projet, car c'est la première fois que je programme un jeu en groupe et cela apporte beaucoup, car on a pas forcément la même vision des choses, la même façon de penser la logique du programme et cela permet souvent de progresser et de trouver un moyen de réaliser une fonction quand l'un ou l'autre est bloqué et ne voit pas comment la construire. De plus, réaliser ce projet m'a permis d'apprendre davantage sur le langage C et la programmation en général car dans un projet on a besoin de réutiliser toutes les notions vues en TD, ce qui nous les fait réviser et mieux les comprendre si on a dû mal avec une notion, ce qui était mon cas avec les structures, je ne voyais pas du tout comment les utiliser. Enfin, réaliser un projet comme celui-ci est toujours enrichissant, car on ne code pas pour réaliser une fonction comme en TD, on aboutit à un programme final, on comprend à quoi toutes les notions apprises peuvent nous servir, ce qui n'était pas forcément le cas avant.

Conclusion d'Alexis :

Travailler sur ce projet m'a apporté beaucoup de choses et m'a permis de me rendre compte des difficultés que pose le fait de travailler sur un projet informatique à deux. On s'est assez vite rendu compte qu'il était difficile de travailler à deux séparément sur le projet donc on a fait le choix de travailler ensemble et en même temps sur les mêmes choses, même si pour les projets à venir que ce soit à l'IUT ou en entreprise il faudra impérativement réussir à séparer les tâches. Valentin et moi avons un niveau en programmation très inégal, alors j'ai beaucoup apprécié qu'il me prenne sous son aile pour ce projet comme une sorte de "tuteur", cela m'a permis d'en apprendre beaucoup sur la programmation et sur le travail d'équipe. Pour compenser le fait que je sois en dessous au niveau des compétences et connaissances, j'essayais tout de même d'apporter des solutions quand il y avait des problèmes, de prendre le clavier pour coder les choses essentielles et basiques afin de manipuler, je m'occupais du debugging en testant tous les déplacements de pion possible pour trouver des erreurs, et je me suis attelé à la rédaction du dossier pour laisser Valentin terminer le programme en détail quand la deadline approchait. Travailler avec quelqu'un d'autre, surtout quelqu'un plus à l'aise avec la programmation, m'a poussé à redoubler d'effort lors des séances de travail en duo pour ne pas le ralentir et être au niveau, j'ai beaucoup aimé travailler de la sorte. Le développement du programme en lui-même a aussi permis d'aborder de nouvelles notions utiles du langage C comme les structures, pointeurs ou la compilation de plusieurs fichiers qui seront utiles pour le futur.