

SPECIFICATION

ETUDIANTS : DIALLO ALPHA OUMAR BINTA / JANG YE CHAN

NIVEAU : Licence 2 Informatique Paul Sabatier

SUJET : Le jeu des Amazones

FONCTIONNALITÉS DU JEU :

DEFINITION DES PIONS :

Pour cela nous définirons comme suit les pions utilisés durant le jeu à l'aide des define de la bibliothèques du langage C :

Nombre de pions noirs NB_NOIR qui équivaut à 4
Nombre de pions blancs NB_BLANC qui équivaut à 4
Les pions noirs P_NOIR 'N'
Les pions blancs P_BLANC 'B'
La case vide VIDE 'O'
La flèche FLECHE 'X'

STRUCTURE UTILISÉES :

On définira une structure coordonnées dans laquelle on conservera les coordonnées du pion à déplacer. Cette structure comportera une abscisse et une ordonnée. On l'implantera de la manière suivante :

```
struct {  
    int abs;  
    int ord;  
}coord_amazones;
```

Pour le nombre de case on définira N 10

On définira une matrice de type char dans laquelle on va stocker les différents types de pions au cours du jeu,
elle sera défini de la manière suivante:

```
typedef char MATRICE[N][N];
```

En ce qui concerne les fonctions et les tests, nous allons expliquer en détails en quatre parties :

FONCTIONS ET TESTS POUR TOUTES LES VERSIONS :

On fera une fonction affichage qui affichera le jeu a chaque tour .

On testera si les pions peuvent se déplacer conformément au sujet (c'est a dire en horizontales, verticale, diagonale).

On testera qu'on ne peut pas déplacer de pion s'il y a quelque chose (pions,barrage) ni la sauter.

On testera de même pour les barrages.

On devra faire en sorte que le déplacement du barrage se fait en fonction du coup joué pour le pion.

On testera que l'ordinateur joue correctement et aléatoirement(c'est a dire sans débordement, sans sauter de pion).

On implantera une fonction qui compte et affiche a chaque tour le nombre de cases disponibles.

On pourra mettre en place aussi une fonction qui affiche a la fin du jeu le nombre de tour joué jusqu'à la fin du jeu.

On implantera une fonction booléenne qui vérifie a chaque tour la fin du jeu c'est a dire qu'il compte le nombre de coups possible qu'il reste pour chaque joueur, et si il n'en reste plus , la fonction renverra vrai.

FONCTIONS DE TESTS POUR LA VERSION 1:

On mettra en place 2 modes , contre un joueur ou contre un ordinateur en forme de menu.

On testera si chaque déplacement est possible, c'est a dire si la case ou on veut placer le pion est vide et si les cases intermédiaires sont toutes libres(vides).

On testera de même pour les barrages .

Pour cela on mettra en place une fonction qui calcule et renvoie a partir de la position initiale du pion, toutes les dernières cases libres pour les directions NORD , SUD , EST , OUEST , NORD-EST , NORD-OUEST , SUD-EST, SUD-OUEST.

On implantera une fonction de sauvegarde et de chargement pour la sauvegarde de la partie et du chargement d'un niveau, pour que l'utilisateur puisse revenir sur les coups déjà joués et d'y naviguer à sa guise.

FONCTIONS ET TESTS POUR LA VERSION 2:

Pour cette version , on doit vérifier avant tout , la bonne inclusion de la bibliothèque ncurses.h.

Il faudra vérifier que notre structure dynamique pour les fonctions de sauvegarde et de chargement marchent bien pour pouvoir permettre à l'utilisateur de revenir en arrière dans une partie et d'y naviguer à sa guise dans les coups joués précédemment.

Il faudra vérifier que la fonction de sauvegarde est bien implanté pour la sauvegarde d'une partie, et toute l'historique dans un fichier texte.

FONCTIONS ET TESTS POUR LA VERSION 3:

Sur cette version on doit tester que les joueurs peuvent être informés que les amazones sont isolées.

Vérifier que la machine ne joue plus au hasard mais selon une stratégie.

On essaiera d'afficher en mode graphique le jeu en utilisant la bibliothèque SDL.