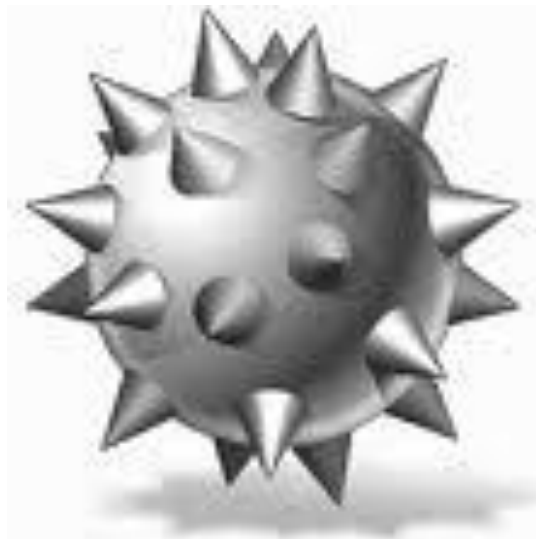


UNIVERSITE CHEIKH ANTA DIOP DE DAKAR



ECOLE SUPERIEURE POLYTECHNIQUE DEPARTEMENT GENIE INFORMATIQUE

NIVEAU DIC1



Développé par :

Cheikh Sidya

CAMARA

Ndèye Awa

GUEYE

Guirguélia Flore Providence SADIO

Juin 2010

Sommaire

PRESENTATION DU DEMINEUR 3

TYPES 3

FONCTIONS..... 4

PRESENTATION DU DEMINEUR

Le **démineur** est un jeu de réflexion dont le but est de localiser des mines cachées dans un champ virtuel avec pour seule indication le nombre de mines dans les zones adjacentes. Et il nous a été demandé d'en développer un en mode texte.

Pour ce faire nous avons scindé le programme en deux parties:

- ✚ Types : regroupant l'ensemble des constantes définies et des types utilisés.
- ✚ Fonctions : constituer de l'ensemble des fonctions élaborées pour le jeu.

TYPES

<i>Nom Type</i>	<i>Explication</i>
<pre>typedef struct str_champs{ int nb_erreurs_restant; int nb_lignes; int nb_colonnes; parcelle *tab; } str_champs; typedef str_champs *champs;</pre>	Défini le champ virtuel et contient les informations suivantes : <ul style="list-style-type: none">• Le nombre d'erreur restant• Le nombre de lignes• Le nombre de colonnes• Un pointeur vers un tableau de nb_lignes* nb_colonnes
<pre>struct parcelle{ int mine; /* 1 s'il y a une mine, 0 sinon */ int status; /* DEJAVU si déjà vu, PASVU sinon */ }; typedef struct parcelle parcelle;</pre>	Type parcelle composé de la mine et du statut de la parcelle.

FONCTIONS

<i>Nom Fonction</i>	<i>Explication</i>
int indice (champs C, int i, int j)	Retourne l'indice de la parcelle dont les coordonnées sont i et j.
int getX(champs C, int k)	Retourne le numéro de la ligne correspondant à l'élément d'indice k.
int getY(champs C, int k)	Retourne le numéro de la colonne correspondant à l'élément d'indice k.
void genererMine(champs C)	Permet de générer des mines comprises entre 0 et le nombre de parcelles. Tout de même on vérifie à chaque fois si l'emplacement généré n'est pas déjà miné.
champs init(int nb_lignes, int nb_colonnes, int nb_mines)	Construit et retourne un champ de taille nb_lignes sur nb_colonnes contenant nb_mines le nombre de mines. Les mines devront être placées aléatoirement.
int nb_mines_voisins(champs C, int i, int j)	Détermine le nombre de mines qui entourent une parcelle de coordonnées i et j.
void affichage(champs C)	Affiche l'état du champ C.
void affichage_triche(champs C)	Permet de dévoiler l'emplacement des mines.
int reste_parcelle_nonMinee(champs C)	Détermine le nombre de parcelles non minées restant.
int joueur(champs C, int i, int j)	Permet au joueur de jouer un coup en donnant les coordonnées de la parcelle qu'il veut découvrir.
void recupXY(char* chaine, int* x, int* y)	Traite la chaine saisie et renvoie les coordonnées x et y de la parcelle désignée.
int sauvegarder(char *fichier, p_champs C)	Permet de sauvegarder un état du jeu dans un fichier.
champs lire(char *fichier)	Permet de restaurer un jeu qui a été sauvegardé dans un fichier.
int fin_jeu(champs C)	Indique si le jeu est fini ou non.
int a_gagne(champs C)	Indique si le joueur a gagné ou non.
void menuDefault(champs A)	Est affiché lorsque l'utilisateur ne fournit aucun paramètre d'entrée. Ce menu permet alors à l'utilisateur de choisir un niveau de difficulté.
void commencerJeu(champs C)	Après avoir initialisé le champ on appelle cette

fonction pour commencer le jeu.

Tant que le jeu n'est pas fini elle affiche l'état du jeu et lit ce que le joueur a saisi.

Il peut soit :

- Entrer **Q** : pour quitter le jeu ;
- Entrer **S** pour enregistrer le jeu en cours dans ce cas il doit donner le nom du fichier dans lequel il veut l'enregistrer ;
- Entrer **a** pour tricher, on lui affiche tout le contenu du champ ;
- i, j (les coordonnées de la parcelle).