

	BTS SIO Services Informatiques aux Organisations		
	Option	SLAM	
	Session	2019	

	Activité scolaire N°	1
--	-----------------------------	----------

NATURE DE L'ACTIVITE	Apprentissage du c : réalisation d'un démineur
Contexte	Dans le cadre de notre formation de développeur à l'école IMC, nous avons étudié le langage c avec notre professeur. Une des activités réalisées pour approfondir notre connaissance du c a été l'élaboration d'un démineur.
Objectifs	Faire un démineur en c, en utilisant des fonctions
Lieu de réalisation	École IMC, 5 rue Saint-Denis, 75001, Paris

DESCRIPTION DE LA SOLUTION RETENUE	
Conditions initiales	Demande de la part du professeur de faire un démineur en partant de rien.
Conditions finales	Le démineur a été fait et rendu dans les délais impartis.
Outils utilisés	CodeBlocks, c

CONDITIONS DE REALISATION	
Matériels	Poste informatique
Logiciels	CodeBlocks
Durée	1 heure
Contraintes	Le démineur devra être réalisé en langage c, en utilisant des fonctions. Il devra contenir un placement aléatoire de mines et permettre de se déplacer à l'intérieur de la grille pour pouvoir chercher les cases. Toute case dévoilée devra contenir un nombre qui correspondra à celui des bombes qui l'entoure.

COMPETENCES MISES EN OEUVRE POUR CETTE ACTIVITE PROFESSIONNELLE	
OBLIGATOIRE	Productions relatives à la mise en place d'un dispositif de veille technologique et à l'étude d'une technologie, d'un composant, d'un outil ou d'une méthode
A1.2.2	Rédaction des spécifications techniques de la solution retenue
A4.1.7	Développement, utilisation ou adaptation de composants logiciels
A4.1.9	Rédaction d'une documentation technique
A5.2.4	Étude d'une technologie, d'un composant, d'un outil ou d'une méthode

DEROULEMENT DE L'ACTIVITE

Contexte

Pour approfondir nos connaissances en c, que nous avons étudié lors de notre premier semestre de programmation en BTS, notre professeur nous a demandé de réaliser un démineur en application console. Les consignes étaient de réaliser le démineur à l'aide de fonctions, et de faire une interface utilisateur de 10 cases par 10 cases où le joueur pourrait se déplacer de case en case à l'aide du pavé directionnel. 10 bombes seraient placées de manière aléatoire sur la grille, et toute case dévoilée par l'utilisateur devrait contenir un nombre indiquant le nombre de bombes l'entourant. Mis à part le langage, la manière de construire le démineur était à la libre appréciation des élèves.

Le professeur désirait toute fois la réalisation d'une petite documentation technique, juste de quoi expliquer la manière dont le code avait été fais.

Exemple de code

Le jeu sera géré par la fonction main, qui crée les différents tableaux et appelle les fonctions qui vont se succéder pour mettre en place le jeu.

```
main()
{
    int i, j, tab[LINE_NB][COLUMN_NB], dec[LINE_NB][COLUMN_NB];
    int rep;
    int result;
    char touche;
    int x = 0, y = 0;
    srand(time(NULL));
    initMines(tab);
    initTab(tab);
    initDec(dec);
    afficheGrille(tab, dec);
    result = parcourir(tab, dec);
    switch (result)
    {
        case 0 : printf("BOOOOOOOOOOOOOOOOM !!!!! \nVous avez perdu !");
        break;
        case 1 : printf("Bravo ! Vous avez gagne le jeu !");
        break;
        case 2 : printf("Bon, vous abandonnez ... ");
    }
}
```

Nous avons choisi de faire deux tableaux : un qui initialise les numéros qui seront présents dans les différentes cases de la grille et un autre qui gère l'affichage et les conditions de victoire.

```
void initTab(int tab[LINE_NB][COLUMN_NB]) //initialisation du numéro qui sera présent dans les cases du tableau
{
    int i, j;
    for(i=1; i<LINE_NB; i++)
    {
        for(j=1; j<COLUMN_NB; j++){
            int nbbombes = 0, k, l;
            if (tab[i][j] != BOMB)
            {
                for (k = -1; k < 2; k++)
                {
                    for (l = -1; l < 2; l++)
                    {
                        if (tab[i+k][j+l] == BOMB)
                        {
                            nbbombes++;
                        }
                        tab[i][j] = nbbombes;
                    }
                }
            }
        }
    }
}

void initDec(int dec[LINE_NB][COLUMN_NB]) //tableau permettant de gérer l'affichage d'une case et les conditions de victoire
{
    int i, j;
    for(i=1; i<LINE_NB; i++)
    {
        for(j=1; j<COLUMN_NB; j++){
            dec[i][j] = 0;
        }
    }
}
```

Les bombes seront initialisées dans une autre fonction, et l'affichage de la grille à l'utilisateur de même.

Enfin, la dernière fonction gèrera le déplacement de l'utilisateur sur la grille ainsi que les différentes actions qu'il peut faire en récupérant le code ascii des touches qu'il appuiera.

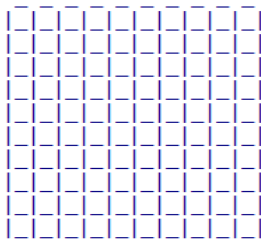
```
do{
    touche = getch();
    if(touche == -32){
        touche = getch(); // récupère la deuxième valeur de la touche directionnelle
        switch(touche){
            case 77: // droite
                if(j < 10){
                    j++;
                }
                break;
            case 75 : //gauche
                if(j > 1)
                {
                    j--;
                }
                break;
            case 80: //bas
                if(i < 10){
                    i++;
                }
                break;
            case 72: //haut
                if(i > 1) {
                    i--;
                }
                break;
        }
        //calcule x et y en fonction de i et j
        y = i;
        x = j * 2;
    }
}
```

Le code est disponible sur cet ordinateur, ainsi que la documentation technique si vous souhaitez plus de détails sur la réalisation du démineur.

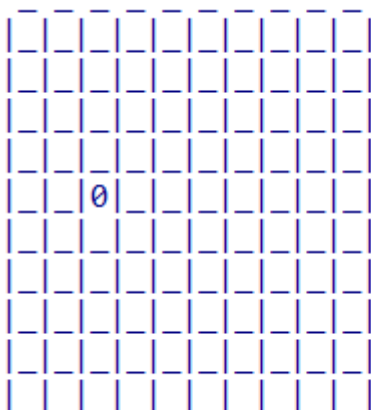
Le jeu terminé

Voici à quoi ressemble le jeu quand il est lancé :

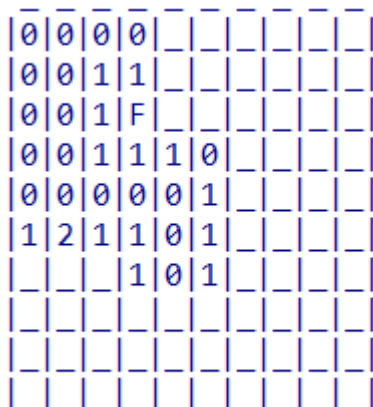
 "D:\Programme C\ExoDÚmineur.exe"



L'endroit où est situé l'utilisateur clignote, et il peut se déplacer grâce au pavé directionnel pour découvrir une des cases et commencer à jouer.



L'utilisateur pourra disposer des drapeaux, marqué d'un F, là où il pense trouver une bombe :



Quand vous réussissez à découvrir toutes les cases hormis celles contenant des bombes, le jeu s'arrête et vous informe de votre victoire :

```
|0|0|0|0|1|F|1|0|0|0|
|0|0|1|1|2|1|1|0|0|0|
|0|0|1|F|1|0|0|0|0|0|
|0|0|1|1|1|0|0|0|0|0|
|0|0|0|0|0|1|1|1|0|0|
|1|2|1|1|0|1|F|1|0|0|
|F|2|F|1|0|1|1|2|1|1|
|2|3|2|1|0|0|0|1|F|1|
|2|F|3|2|1|0|0|1|1|1|
|_|3|F|F|1|0|0|0|0|0|
```

Bravo ! Vous avez gagné le jeu !

Il en sera de même si vous tombez sur une bombe, matérialisée par le chiffre 9 :

```
|1|_|_|_|_|_|_|_|_|
|9|_|_|_|_|_|_|_|_|
|_|_|_|_|_|_|_|_|
|_|_|_|_|_|_|_|_|
|_|_|_|_|_|_|_|_|
|_|_|_|_|_|_|_|_|
|_|_|_|_|_|_|_|_|
|_|_|_|_|_|_|_|_|
|_|_|_|_|_|_|_|_|
|_|_|_|_|_|_|_|_|
```

B0000000000000000M !!!!!

Le jeu est disponible sur l'ordinateur si vous souhaitez le lancer vous-même et jouer une partie.

CONCLUSION

Le jeu a été rendu au professeur dans les temps impartis et a servi d'évaluation de nos capacités en c jusqu'alors. Cette tâche m'a permis d'approfondir mes connaissances en langage c et d'avoir une meilleure appréhension de la façon dont il faut réfléchir aux algorithmes lorsqu'on nous demande la réalisation d'un nouveau projet en code.