

# DESCRIPTION D'UNE SITUATION PROFESSIONNELLE

<b>PARCOURS</b>	<b>SISR</b> <input type="checkbox"/>	<b>SLAM</b> <input checked="" type="checkbox"/>
<b>Lieu de réalisation</b>	Campus Montsouris	 
<b>Période de réalisation</b>	Du : 4/04/23	Au : 18/04/23
<b>Modalité de réalisation</b>	<b>SEUL</b> <input checked="" type="checkbox"/>	<b>EN EQUIPE</b> <input type="checkbox"/>
<b>Intitulé de la mission</b>	Création d'un démineur	
<b>Description du contexte de la mission</b>	Mettre en place le jeu du démineur	
<b>Contraintes &amp; Résultat</b>	Ressources fournies / contraintes techniques / Résultats attendu Ressources fournies : C# Contraintes Techniques : Résultats attendus : Faire un démineur fonctionnel	
<b>Productions associées</b>	Liste des documents produits et description 	
<b>Modalités d'accès aux productions</b>	Identifiants, mots de passe, URL d'un espace de stockage et présentation de l'organisation du stockage 	

<b>Description détaillée de la situation professionnelle retenue et des productions réalisées</b> en mettant en évidence la démarche suivie, les méthodes et les techniques utilisées Lancer Visual Studio Ouvrir le projet du démineur Lancer le projet Jouer au démineur
---

```

1 references
public static void afficheTab(string[] tab) // Affiche les éléments d'un tableau à une dimension
{
    for(int i = 0; i < tab.Length; i++) {
        Console.WriteLine(i);
        if (i % 10 == tab.Length - 1) {
            Console.WriteLine("");
        }
        Console.WriteLine(tab[i]);
    }
}

2 references
public static void initialiserTableau(ref string[] tableau) { // initialise un tableau à une dimension avec des " "
    for (int i = 0; i < tableau.Length; i++) {
        tableau[i] = " ";
    }
}

/* public static void Remplitab(ref string[] tableau, int nombreCases) {...}

3 references
public static int verifValeur (string texte) { // Vérifie que la valeur saisie est bien un nombre
    int valeur = 0;
    bool valideValeur = false;
    do {
        Console.WriteLine();
        Console.WriteLine(texte);
        try {
            valeur = int.Parse(Console.ReadLine());
            valideValeur = true;
        } catch (Exception e) {
        }
    } while (!valideValeur);
    return valeur;
}

```

```

1 references
public static int verifValeur (string texte) { // Vérifie que la valeur saisie est bien un nombre
    int valeur = 0;
    bool valideValeur = false;
    do {
        Console.WriteLine();
        Console.WriteLine(texte);
        try {
            valeur = int.Parse(Console.ReadLine());
            valideValeur = true;
        } catch (Exception e) {
        }
        Console.WriteLine("Veuillez saisir un nombre");
        valideValeur = false;
    } while (!valideValeur);
    return valeur;
}

2 references
public static void rempliMines(ref string[] tab, int taille) { // Remplit le tableau avec des mines à des positions aléatoires
    int mines = 0;
    for (taille-->0) {
        mines = 2;
    } else {
        mines = 4;
    }
    Random aléatoire = new Random();
    int position = 0;
    int i = 0;
    while (i < mines) {
        position = aléatoire.Next(0, tab.Length);
        if (tab[position] == " ") {
            tab[position] = "M";
        }
    }
}

```

```

public static void rempliMines(ref string[] tab, int taille) { // Remplit le tableau avec des mines à des positions aléatoires
    int mines = 0;
    if (taille==10) {
        mines = 2;
    } else {
        mines = 4;
    }
    Random aléatoire = new Random();
    int position = 0;
    int i = 0;
    while (i < mines) {
        position = aléatoire.Next(0, tab.Length);
        if (tab[position] == " ") {
            tab[position] = "M";
            i++;
        }
    }
}

1 references
public static void minesAdjacentes (ref string[] tab) // Calcule le nombre de mines adjacentes et ajoute le nombre dans le
{
    int res = 0;
    for(int i = 0; i < tab.Length; i++) {
        if (tab[i] != "M") {
            if(i!=0) {
                if (tab[i-1] == "M") {
                    res++;
                }
            }
            if (i!=tab.Length-1) {
                if (tab[i+1] == "M") {
                    res++;
                }
            }
        }
    }
}

```

```

public static void MinesAdjacentes (ref string[] tab) // Calcule le nombre de mines adjacentes et ajoute le nombre dans le tableau à la position calculée
{
    int res = 0;
    for(int i = 0; i < tab.Length; i++) {
        if (tab[i] != "M") {
            if(i!=0) {
                if (tab[i-1] == "M") {
                    res++;
                }
            }
            if (i!=tab.Length-1) {
                if (tab[i+1] == "M") {
                    res++;
                }
            }
            tab[i] = Convert.ToString(res);
            res = 0;
        }
    }
}

2 references
public static bool tabMinesAdjacentes (ref string[] tab, int position) { // Vérifie si la position contient une mine ou pas
    bool verif = false;
    if (tab[position] == "M") {
        verif = true;
    }
    return verif;
}

3 references
public static bool tabComplet(ref string[] tab, string[] tableau) { // Vérifie si le tableau de jeu est complet (si toutes les cases sauf celles des mines ont été découvertes)
    for(int i = 0; i < tableau.Length; i++) {
        if (tableau[i] == " ") {
            tabComplet = false;
        }
    }
}

```

```

1 references
public static bool tabComplet(ref string[] tab, string[] tableau) { // Vérifie si le tableau de jeu est complet (si toutes les cases sauf celles des mines ont été découvertes)
    bool tabComplet = true;
    for(int i = 0; i < tableau.Length; i++) {
        if (tableau[i] == " ") {
            tabComplet = false;
        }
    }
    return tabComplet;
}

2 references
public static void jouer () // Met en place la routine de jeu
{
    Console.WriteLine("Bienvenue sur le jeu de mines");
    Console.WriteLine("Choisissez la taille du tableau avec lequel vous voulez jouer (10 ou 16 cases)");
    string taille = "vous voulez jouer avec quelle taille de tableau? (10 ou 16 cases)";
    int nombreCases = verifValeur(taille);
    while(nombreCases < 10 || nombreCases > 16) {
        Console.WriteLine("Erreur de saisie, choisissez 10 ou 16");
        nombreCases = verifValeur(taille);
    }
    string[] tab = new string[nombreCases];
    initialiserTableau(ref tab);
    rempliMines(ref tab, nombreCases);
    minesAdjacentes(ref tab);
    string[] tableau = new string[nombreCases];
    initialiserTableau(ref tableau);
    afficheTab(tab);
    int position = 0;
    do {
        Console.Clear();
        afficheTab(tab);
        taille = "vous voulez jouer avec quelle taille de tableau? (10 ou 16 cases)";
        position = verifValeur(taille);
        if (verifValeur(tab, position)) {
            tab[position] = "M";
        }
    } while (true);
}

```

```

public static void jouer () // Met en place la routine de jeu
{
    Console.WriteLine("Bienvenue sur le jeu de mines");
    Console.WriteLine("Choisissez la taille du tableau avec lequel vous voulez jouer (10 ou 16 cases)");
    string taille = "vous voulez jouer avec quelle taille de tableau? (10 ou 16 cases)";
    int nombreCases = verifValeur(taille);
    while(nombreCases < 10 || nombreCases > 16) {
        Console.WriteLine("Erreur de saisie, choisissez 10 ou 16");
        nombreCases = verifValeur(taille);
    }
    string[] tab = new string[nombreCases];
    initialiserTableau(ref tab);
    rempliMines(ref tab, nombreCases);
    minesAdjacentes(ref tab);
    string[] tableau = new string[nombreCases];
    initialiserTableau(ref tableau);
    afficheTab(tab);
    int position = 0;
    do {
        Console.Clear();
        afficheTab(tab);
        taille = "vous voulez jouer avec quelle taille de tableau? (10 ou 16 cases)";
        position = verifValeur(taille);
        if (verifValeur(tab, position)) {
            tab[position] = "M";
        }
    } while (true);
}

```

```

}
afficheTab(tabJeu);
Console.WriteLine();
Console.WriteLine("Dommage, vous avez perdu! Vous gagnerez peut-être la prochaine fois.");
}

0 references
public static void Main(string[] args) { // Lance la routine de jeu
    //int taille = tailleTableau();
    //string[] tableau = new string[taille];
    //initialiserTableau(ref tableau);
    //int nombreCases = positionMines(tableau);
    //remplirTab(ref tableau, nombreCases);
    //afficheTab(tableau);
    jouer();
}
}

```