

PROJET C#

Introduction du projet.....	Page 2
Screens interface + détails.....	Page 3-4
Screens code C#.....	Page 5-7
Screen fichier vide et rempli.....	Page 8

INTRODUCTION

L'application est un extrait du système de matchmaking du jeu vidéo Minecraft qui prend en compte les prénoms, leurs nombre de morts ainsi qu'un joueur qu'ils ont tués lors de leur partie.

L'application consiste à enregistrer des données sur des joueurs qu'on pourra enregistrer et sauvegarder dans un fichier "minecraft.txt". Après avoir réalisé ceci, nous pouvons cliquer sur des boutons qui exécuteront des événements différents comme afficher un classement, afficher la liste des joueurs avec leur nombre de morts et enfin afficher le joueur le plus et moins morts.

Pour réaliser cette application j'ai utilisé ChatGPT pour faire le classement des joueurs. Sinon pour tout le reste j'ai utilisé les cours comme exemple.

SCREEN INTERFACE + DÉTAILS

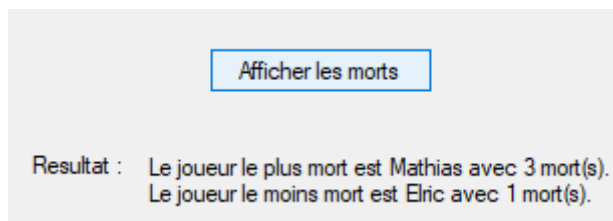
Ci-dessous, une capture d'écran de l'interface du projet C# entier.

The screenshot shows a Windows application window with a light gray background. On the left side, there are several controls: a text box labeled 'Pseudo :' containing 'Sofiane', a button 'Afficher', a text box labeled 'Nombre de morts :' containing '3', a radio button group labeled 'Joueur tués :' with 'Nino' and 'Stanley' (selected), and two buttons 'Enregistrer' and 'Sauvegarder'. On the right side, there is a button 'Afficher les morts', a label 'Resultat : label6', and a button 'Afficher le classement'. In the center, there is a large empty rectangular box. At the bottom right, there is another empty rectangular box.

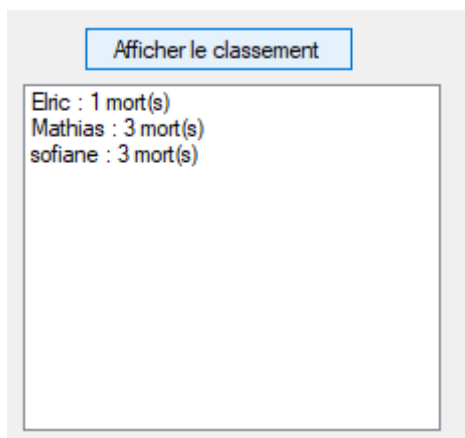
Voici, deux nouvelles captures d'écran où nous pouvons voir un message quand on effectue une sauvegarde en cliquant sur le bouton correspondant (capture d'écran de gauche). Il y a aussi un aperçu du résultat quand on clique sur le bouton afficher lorsque que les données nécessaires ont été saisis auparavant (capture d'écran de droite).

The left screenshot shows a small dialog box with a title bar, a close button (X), and the text 'Sauvegarde dans le fichier effectué'. At the bottom, there is an 'OK' button. The right screenshot shows a larger dialog box with a title bar and a close button (X). It contains a text area with the following text: 'Eric a tué Nino', 'Mathias a tué Stanley', and 'sofiane a tué Stanley'. Above the text area is a button labeled 'Afficher'.

Sur la capture d'écran ci-dessous nous pouvons voir un aperçu du résultat que ça donne lorsque l'on clique sur le bouton "Afficher les morts".



Enfin, ci-dessous il y a une capture d'écran qui affiche le classement des joueurs en fonction du nombre de morts lorsque l'on clique sur le bouton "Afficher le classement".



CODE C#

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Projet
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            // Déclaration des tableaux pour stocker les informations des joueurs
            string[] TabPseudo = new string[10];
            int[] TabNbMorts = new int[10];
            string[] TabJoueurTues = new string[10];

            // variable pour compter le nombre de joueurs enregistrés
            int nbJoueur;

            private void Form1_Load(object sender, EventArgs e)
            {
                // initialisation de la variable de comptage à 0 au lancement
                nbJoueur = 0;
            }
            private void BTenregistrer_Click(object sender, EventArgs e)
            {
                // Enregistrer les informations du joueur dans les tableaux correspondants
                TabPseudo[nbJoueur] = TBPseudo.Text;
                TabNbMorts[nbJoueur] = Convert.ToInt16(UDnbmorts.Value);
            }
        }
    }
}
```

```

if (RBNino.Checked)
    TabJoueurTues[nbJoueur] = "Nino";
else
    TabJoueurTues[nbJoueur] = "Stanley";

// Augmenter le compteur de joueurs enregistrés
nbJoueur++;

```

```

}

```

```

private void BTAfficher_Click(object sender, EventArgs e)
{
    // Afficher les joueurs qui ont tué Nino ou Stanley dans la listBox
    listBoxKill.Items.Clear(); //Cela permet de vider la listBox pour éviter de tout
réafficher à chaque fois
    for (int i = 0; i <= nbJoueur - 1; i++)
        if (TabJoueurTues[i] == "Nino")
            listBoxKill.Items.Add(TabPseudo[i] + " a tué Nino");
        else
            listBoxKill.Items.Add(TabPseudo[i] + " a tué Stanley");
}

```

```

private void BTSave_Click(object sender, EventArgs e)
{
    // Sauvegarder les pseudos des joueurs dans un fichier texte
    StreamWriter monFichier = new StreamWriter(@"minecraft.txt");
    for (int i = 0; i <= nbJoueur - 1; i++)
    {
        monFichier.WriteLine(TabPseudo[i]);
    }
    monFichier.Close();
    MessageBox.Show("Sauvegarde dans le fichier effectué");
}

```

```

private void button1_Click(object sender, EventArgs e)
{
    // Trouver le joueur qui a le plus de morts et celui qui en a le moins
    int MAX, MIN;
    string nul, fort;
    MAX = TabNbMorts[0];
    MIN = TabNbMorts[0];
    fort = TabPseudo[0];
    nul = TabPseudo[0];
}

```

```

for (int i = 1; i < nbJoueur; i++)
{
    if (TabNbMorts[i] > MAX)
    {
        MAX = TabNbMorts[i];
        fort = TabPseudo[i];
    }

    if (TabNbMorts[i] < MIN)
    {
        MIN = TabNbMorts[i];
        nul = TabPseudo[i];
    }
}

// Afficher le résultat dans le label
labelResultat.Text = "Le joueur le plus mort est " + fort + " avec " + MAX + "
mort(s)." + "\nLe joueur le moins mort est " + nul + " avec " + MIN + " mort(s).";
}

private void BTClassement_Click(object sender, EventArgs e)
{
    // Créer une liste de tuples pour stocker les pseudos et le nombre de morts
    correspondant
    List<(string, int)> joueurs = new List<(string, int)>();

    // Ajouter les pseudos et le nombre de morts correspondant à la liste
    for (int i = 0; i < nbJoueur; i++)
    {
        joueurs.Add((TabPseudo[i], TabNbMorts[i]));
    }

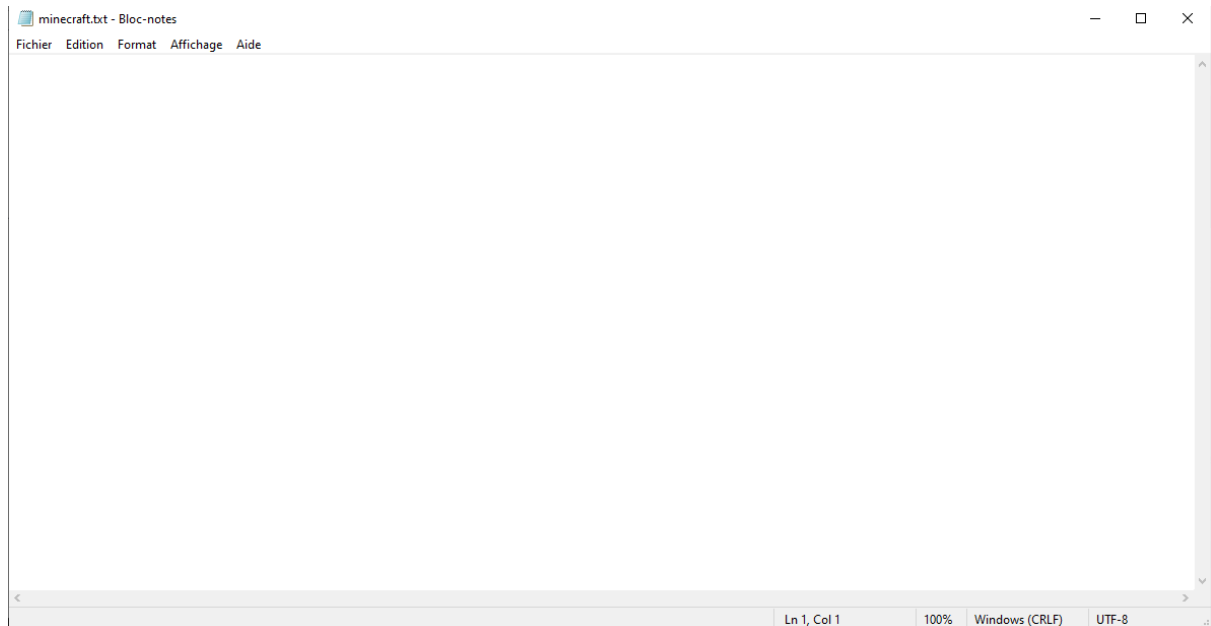
    // Trier la liste par ordre croissant du nombre de morts
    joueurs.Sort((x, y) => x.Item2.CompareTo(y.Item2));

    // Afficher les pseudos dans la listBox par ordre croissant du nombre de
    morts
    listBoxClassement.Items.Clear();
    foreach (var joueur in joueurs)
    {
        listBoxClassement.Items.Add($"{joueur.Item1} : {joueur.Item2} mort(s)");
    }
}
}
}

```

SCREENS FICHIER VIDE + REMPLIS

Ci-dessous il y a une capture d'écran du fichier "minecraft.txt" qui sera utilisé pour le projet C#.



Voici un exemple du fichier "minecraft.txt" lorsqu'on exécute le programme (capture d'écran ci-dessous).

