

TP 2 – PokemonTournament WPF

Objectif du TP :

Manipuler des fichiers xaml, créer des windows, réutiliser la bibliothèque de classe du TP n°1.
Créer une smart application WPF attrayante et séduisante.

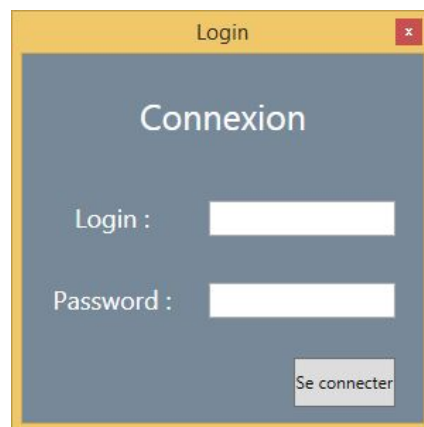
Etape 1 : Création du projet

Ouvrir Visual Studio et votre ancienne solution du TP1. Fichier => nouveau projet, Choisir Visual C# => Application WPF. La nommer « PokemonTournamentWPF ».

Dans la solution nouvellement créée, ajouter la référence aux dlls « BusinessLayer », et « EntitiesLayers » créées dans le TP précédent.

Etape 2 : Ecran de connexion

Créer la fenêtre de connexion : Le champ password doit être de type « PasswordBox ».



Créer une nouvelle classe dans EntitiesLayer « Utilisateur » (avec nom, prenom, login et password) et créer une méthode GetUtilisateurByLogin() dans StubDataAccessLayer. La vérification du mot de passe devra se faire dans la couche BusinessLayer.

Le code derrière le bouton de connexion devrait ressembler à ça :

```

void btnConnexion_Click(object sender, RoutedEventArgs e)
{
    if (BusinessManager.CheckConnexionUser(txtLogin.Text.ToLower(), txtPassword.Password))
    {
        MainWindow win = new MainWindow();
        win.Show();

        this.Close();
    }
}

```

Enfin, pour que cet écran de login se lance au démarrage de votre application en lieu et place de la mainWindow vous devez modifier le fichier « app.xaml ».

Etape 3 : Gestion des erreurs

La gestion des erreurs en WPF est assez simple : tout se passe dans la classe app.cs :

```

/// <summary>
/// Logique d'interaction pour App.xaml
/// </summary>
public partial class App : Application
{
    protected override void OnStartup(StartupEventArgs e)
    {
        base.OnStartup(e);
        DispatcherUnhandledException += App_DispatcherUnhandledException;
    }

    void App_DispatcherUnhandledException(object sender, System.Windows.Threading.DispatcherUnhandledExceptionEventArgs e)
    {
        //ici votre gestion d'erreur
    }
}

```

A vous d'imaginer une gestion efficace...

Etape 4 : Main Windows

Créer la fenêtre principale pour qu'elle permette l'accès aux 4 fonctionnalités qu'elle doit satisfaire. Choisir le container qui convient le mieux à une représentation aérée et customisable. Le résultat pourrait ressembler au visuel ci-dessous :

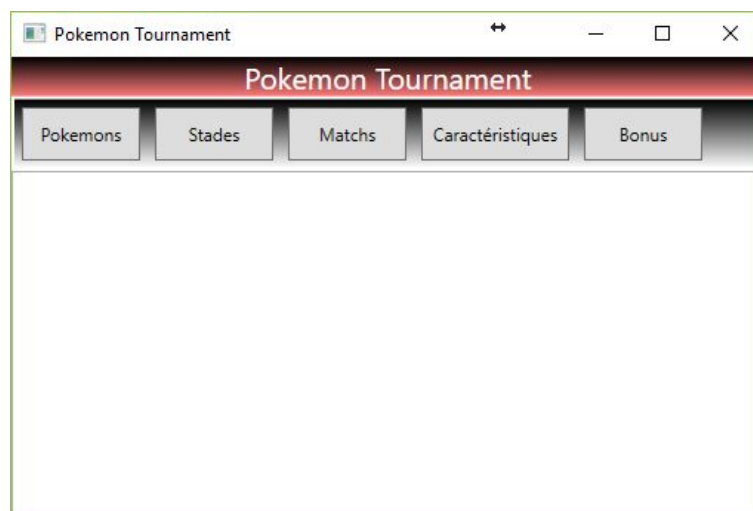


Image non contractuelle : faite comme vous voulez !

Lors du 1^e lancement, la fenêtre doit s'ouvrir au centre de l'écran.

Etape 5 : Fonctionnalités

Le but de l'application est de permettre d'afficher les méthodes faite dans le TP1 mais en plus sympathique, c'est à dire :

- Affichier la liste des Pokemons
- Afficher la liste des Stades
- Afficher la liste des Matches
- Afficher la liste des caractéristiques
- Afficher la liste de votre choix

Créer un bouton pour chaque option et afficher le résultat dans un conteneur.

Etape 6 : Sérialisation

Ajouter un bouton "exporter les pokemons" qui exportera la liste complète des Pokemons dans un fichier XML. L'utilisateur doit pouvoir choisir le nom et l'emplacement du fichier de destination.

A la fin de ce TP, votre application permet d'afficher le contenu des collections issues de la base de données qui sont, pour le moment, « stubées ».