

**Dossier de programmation 2 :**  
**SONTOT Alexis**

# Sommaire

A) Compétences

B) Structure des données

C) Explication des algorithmes :

1. La classe Joueur
2. Menu MDI
3. Form nouveau joueur
4. Form voir joueur
5. Form changer mail
6. Form modifier avatar
7. Form Evoluer
8. Form points

D) Jeu d'essai

E) Conclusion

## A) Compétences :

A4.1.6 Gestion d'environnements de développement et de test.

A4.1.7 Développement, utilisation ou adaptation de composants logiciels.

A4.1.8 Réalisation des tests nécessaires à la validation d'éléments adaptés ou développés.

A4.1.9 Rédaction d'une documentation technique.

## B) Structure des données :

Classe Globale :

```
class Global
{
    public static int nbMaxNiveaux = 42 ;
    public static int plafondScore = 666;
    public static ClassJoueur leJoueurTest;
}
```

Classe Joueur :

```
class ClassJoueur
{
    //attributs privés
    private string nom;
    private string prenom;
    private string email;
    private string pseudo;
    PictureBox avatar;
    private int niveau;
    private int score;
}
```

## C) Explication des algorithmes :

### 1. Classe Joueur :

La Classe Joueur contient des getters permettant de récupérer les attributs privés de la classe.

Elle contient aussi des setters qui permettent de modifier les attributs privés de la classe.

De plus elle a 3 procédures :

```
public void NiveauxSupp()
{
    this.niveau += 1;
}

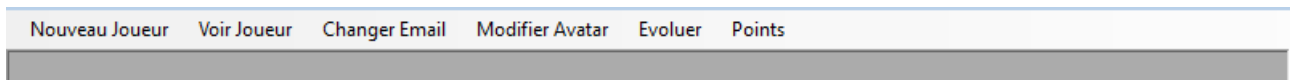
public void NiveauxReinit()
{
    this.niveau =1;
}

public void ScorePlus()
{
    this.score += 111;
}
```

NiveauxSupp permet l'augmentation du niveau d'un, NiveauxReinit permet de réinitialisé le niveau à 1 et ScorePlus permet l'augmentation du niveau de 111.

## 2. Menu MDI :

Création d'un menu MDI qui contient des options comme la création d'un nouveau joueur ( option obligatoire pour avoir accès aux autres options) et d'autre option tel que :



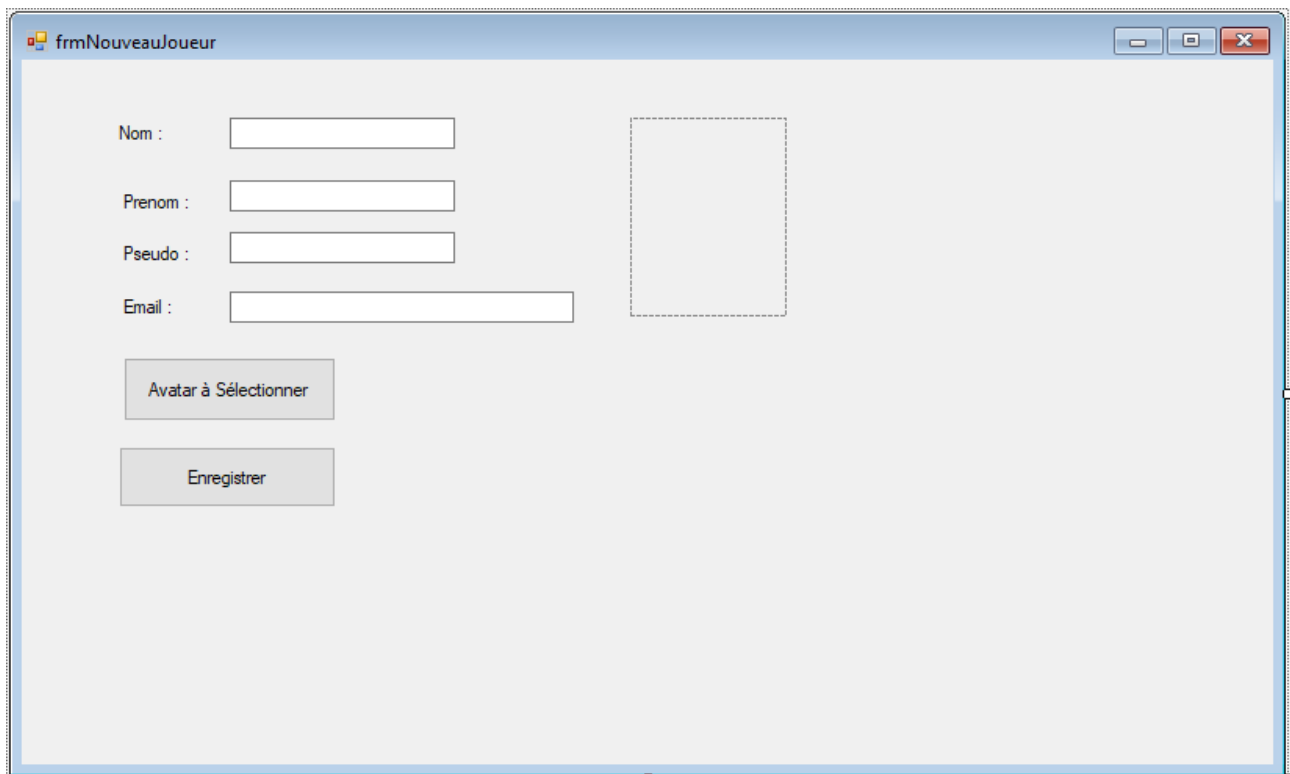
Pour ouvrir les forms enfant voici le code :

```
private void créerNouveauJoueurToolStripMenuItem_Click(object sender, EventArgs e)
{
    frmNouveauJoueur newform = new frmNouveauJoueur();
    newform.MdiParent = this;
    newform.WindowState = FormWindowState.Maximized;
    newform.Show();
}
```

## 3. Form nouveau joueur :

Le Form nouveau joueur permet d'instancier un objet ClassJoueur qui sera par la suite mis en Globale :

- Le visuel du form :

A screenshot of a Windows form titled 'frmNouveauJoueur'. The form has a light grey background and a blue border. It contains four text input fields labeled 'Nom :', 'Prenom :', 'Pseudo :', and 'Email :'. To the right of these fields is a dashed rectangular box. Below the input fields are two buttons: 'Avatar à Sélectionner' and 'Enregistrer'. The form has standard Windows window controls (minimize, maximize, close) in the top right corner.

Le code du form dans lequel on instancie l'objet avec les valeurs des textbox ainsi que de l'avatar :

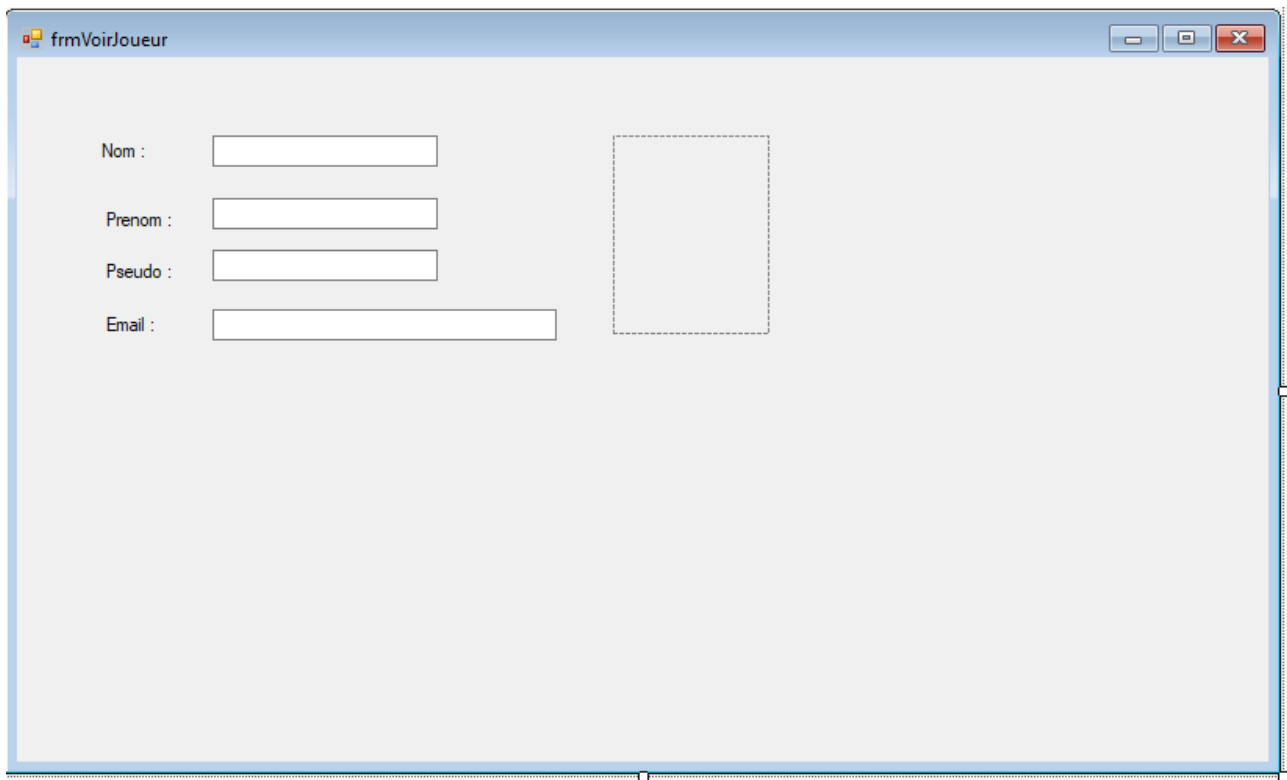
```
private void button1_Click(object sender, EventArgs e)
{
    string avatar = "";
    OpenFileDialog imageD = new OpenFileDialog();
    imageD.Filter = "jpg files (*.jpg |*.jpeg) PNG files (*.png)|*.png";
    imageD.ShowDialog();
    avatar = imageD.FileName;
    pbAvatar.ImageLocation = avatar;
}

private void btEnregistrer_Click(object sender, EventArgs e)
{
    try
    {
        if (tbNom.Text != "" && tbPrenom.Text != "" && tbEmail.Text != "" && tbPseudo.Text != "")
        {
            Global.leJoueurTest = new ClassJoueur(tbNom.Text, tbPrenom.Text, tbPseudo.Text, tbEmail.Text, pbAvatar );
            MessageBox.Show("Joueur Créer", "info", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        else
        {
            MessageBox.Show("Erreur entrer un email valide, ou/et un nom/prenom valide", "error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
    catch
    {
        MessageBox.Show("Erreur entrer un email valide, ou/et un nom/prenom valide", "error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

#### 4. Form voir joueur :

Le Form voir joueur nous montre les informations du joueur dans des textbox ainsi que son avatar dans une picturebox :

Le visuel :



The screenshot shows a Windows application window titled "frmVoirJoueur". Inside the window, there are four text boxes arranged vertically on the left side, each with a label to its left: "Nom :", "Prenom :", "Pseudo :", and "Email :". To the right of these text boxes is a large, empty rectangular area with a dashed border, which serves as a placeholder for an avatar image. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

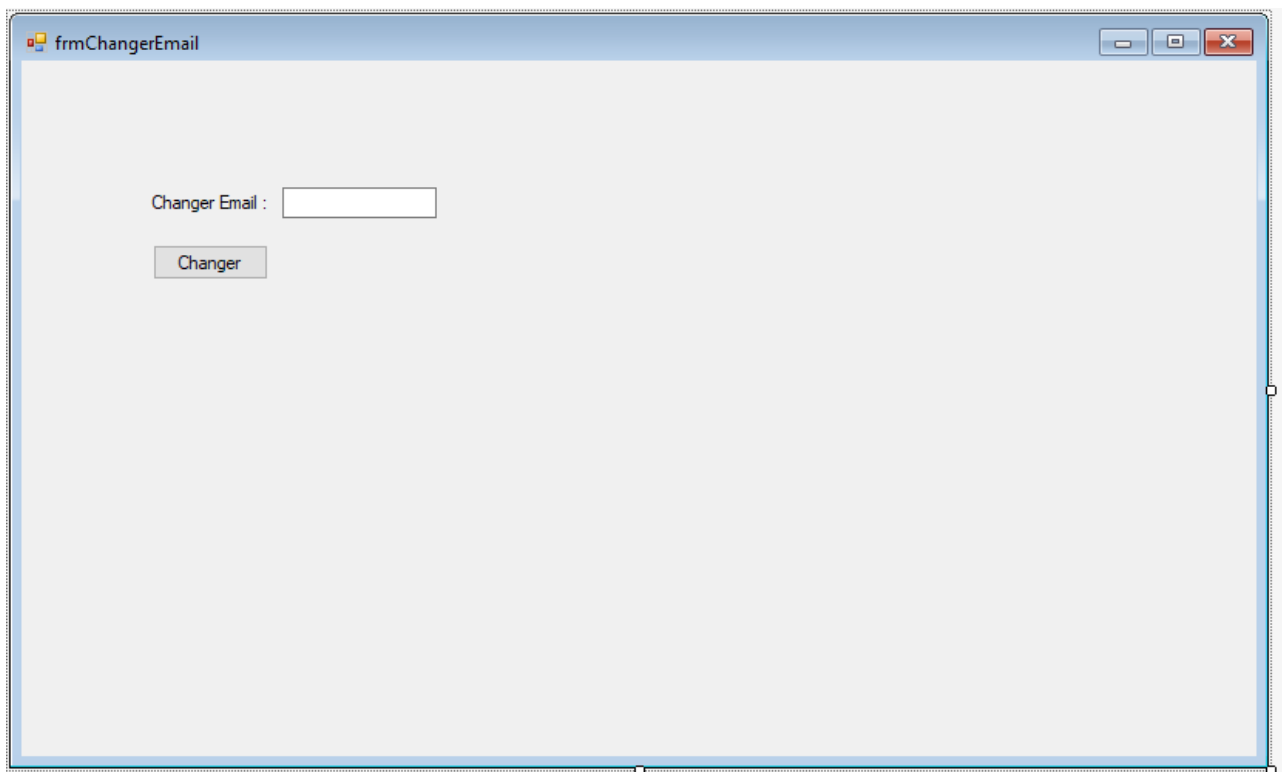
Le code :

```
private void frmVoirJoueur_Load(object sender, EventArgs e)
{
    try
    {
        tbNom.Text = Global.leJoueurTest.getNom();
        tbPrenom.Text = Global.leJoueurTest.getPrenom();
        tbPseudo.Text = Global.leJoueurTest.getPseudo();
        tbEmail.Text = Global.leJoueurTest.getEmail();
        pbAvatar = Global.leJoueurTest.getavatar();
    }
    catch
    {
        MessageBox.Show("Veuillez créer un Joueur", "Erreur", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

#### 5. Form changer mail :

Tout est dans son nom il permet a l'aide d'une textbox et d'un bouton de changer l'adresse mail :

Le visuel :



Le code :

```
private void btChanger_Click(object sender, EventArgs e)
{
    try
    {
        Global.leJoueurTest.setEmail(tbChangerEmail.Text);
    }
    catch
    {
        MessageBox.Show("Veuillez créer un Joueur", "Erreur", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

## 6. Form modifier avatar :

Le Form modifier avatar reutilise le code de nouveau joueur pour changer son avatar grace au setter SetAvatar()

```
private void btChangerAvatar_Click(object sender, EventArgs e)
{
    try
    {
        string avatar = "";
        OpenFileDialog imageD = new OpenFileDialog();
        imageD.Filter = "jpg files (*.jpg |*.jpeg) PNG files (*.png)|*.png";
        imageD.ShowDialog();
        avatar = imageD.FileName;
        pbAvatar.ImageLocation = avatar;

        Global.leJoueurTest.setAvatar(pbAvatar);
    }
    catch
    {
        MessageBox.Show("Veuillez créer un Joueur", "Erreur", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

## 7. Boutons niveau :

Appelle les procédures NiveauSupp et NiveauReinit grâce à deux boutons de plus, il faut vérifier si le niveau n'est pas égal au niveau maximum dans le cas contraire on ne fait rien et informe l'utilisateur qu'il a atteint le niveau maximum.

```
private void btNivSupp_Click(object sender, EventArgs e)
{
    try
    {
        if (Global.leJoueurTest.getNiveau() < Global.nbMaxNiveaux)
        {
            Global.leJoueurTest.NiveauxSupp();
            MessageBox.Show("Niveau Augmente le niveau actuel est de " + Global.leJoueurTest.getNiveau(), "info", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        else
        {
            MessageBox.Show("Niveau Maximum Atteint ", "info", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
    catch
    {
        MessageBox.Show("Veuillez créer un Joueur", "Erreur", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void btReinit_Click(object sender, EventArgs e)
{
    try
    {
        Global.leJoueurTest.NiveauxReinit();
        MessageBox.Show("Niveau Réinitialisé ", "info", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    catch
    {
        MessageBox.Show("Veuillez créer un Joueur", "Erreur", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```



## 8. Form points :

Le form point permet de voir les points actuels grâce à une textbox en readonly, elle appelle aussi la procédure ScorePlus :

```
private void btScorePlus_Click(object sender, EventArgs e)
{
    try
    {
        if (Global.leJoueurTest.getScore() < Global.plafondScore)
        {
            Global.leJoueurTest.ScorePlus();
            int Score = Global.leJoueurTest.getScore();
            tbScore.Text = Score.ToString();
        }
        else
        {
            MessageBox.Show("Le score max a été atteint niveau supérieur", "info", MessageBoxButtons.OK, MessageBoxIcon.Information);
            if (Global.leJoueurTest.getNiveau() < Global.nbMaxNiveaux)
            {
                Global.leJoueurTest.NiveauxSupp();
                MessageBox.Show("Niveau Augmente le niveau actuel est de " + Global.leJoueurTest.getNiveau(), "info", MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
            else
            {
                MessageBox.Show("Niveau Maximum Atteint ", "info", MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
            Global.leJoueurTest.setScore(0);
            Global.leJoueurTest.ScorePlus();
            int Score = Global.leJoueurTest.getScore();
            tbScore.Text = Score.ToString();
        }
    }
    catch
    {
        MessageBox.Show("Veuillez créer un Joueur", "Erreur", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void frmScore_Load(object sender, EventArgs e)
{
    try
    {
        int Score = Global.leJoueurTest.getScore();
        tbScore.Text = Score.ToString();
    }
    catch
    {
        MessageBox.Show("Veuillez créer un Joueur", "Erreur", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

## E) Jeu d'essai :

1	Ouverture des différentes form et menus via le menu MDI	Valider
2	Form nouveau joueur fonctionnelle entièrement ainsi que ses tests	Echec : Les tests ne sont pas effectués
3	Form voir joueur fonctionnel	Valider
4	Form changement de mail fonctionnel	Valider
5	Form changement d'avatar fonctionnel	Valider
6	Form changement de points fonctionnels	Valider
8	Bouton évoluer et réinitialiser fonctionnelles	Valider
9	Message d'erreur en cas de tentative d'ouverture d'une option du mdi autre que nouveau joueur avant la création d'un nouveau joueur	Echec : Le message d'erreur apparaît cependant le formulaire s'ouvre quand même après l'apparition du message

## F) Conclusion :

Pour conclure le menu Mdi est dans l'ensemble fonctionnelle cependant le projet aurait pu être amélioré et être intégré au dpl et utiliser une collection pour les joueurs afin d'en enregistrer plusieurs et donc de pouvoir choisir le joueur à modifier ainsi que choisir le niveau max et le plafond de score.