Atelier Programmation

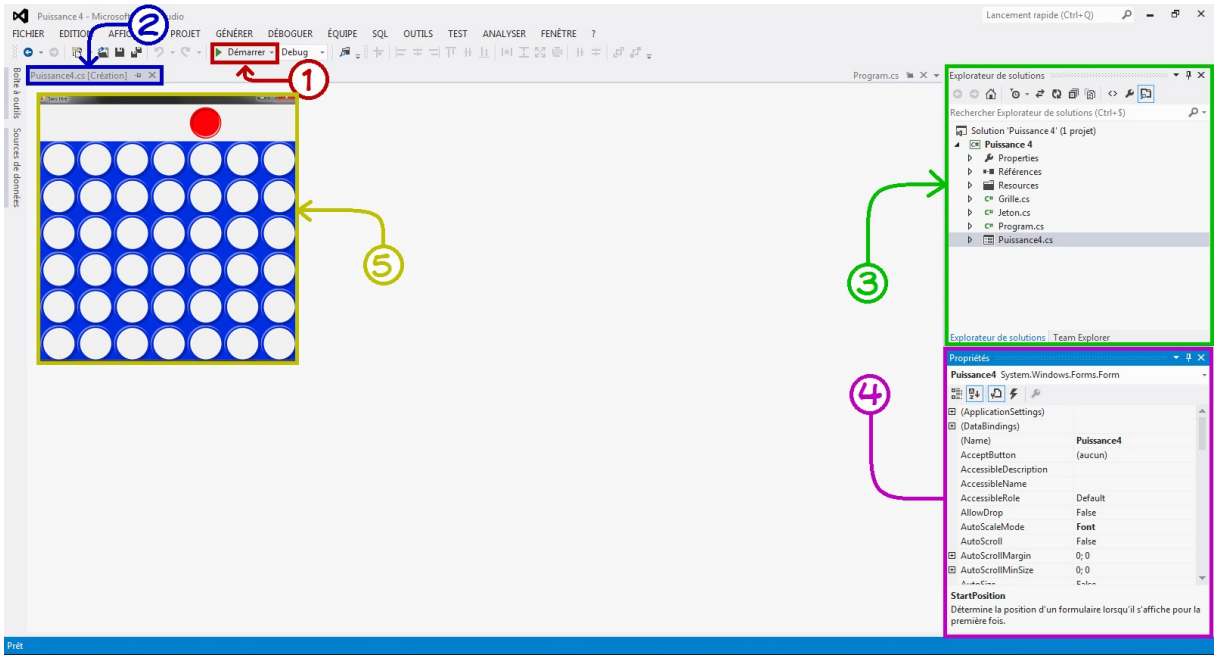
Jeu : Puissance 4

En cas de problème, n'hésitez pas à demander de l'aide !

Puissance 4 est un jeu simple, où vous devez aligner 4 jetons de la même couleur avant votre adversaire. En le découvrant, vous verrez qu'on peut y apporter quelques améliorations, mais aussi qu'il y a quelques problèmes à corriger. Votre mission aujourd'hui sera de le rendre meilleur, et ainsi comprendre le fonctionnement de la programmation. Pour cela, nous utiliserons Microsoft Visual Studio, un logiciel de développement pour Windows.

Voici l'environnement sous lequel vous allez passer les 20 prochaines minutes :

Vue globale de Visual Studio



1. **Compiler et exécuter le programme**
2. **Onglets des différents fichiers**
3. **Explorateur des différents fichiers**
4. **Propriétés de l'objet sélectionné**
5. **Aperçu du programme**

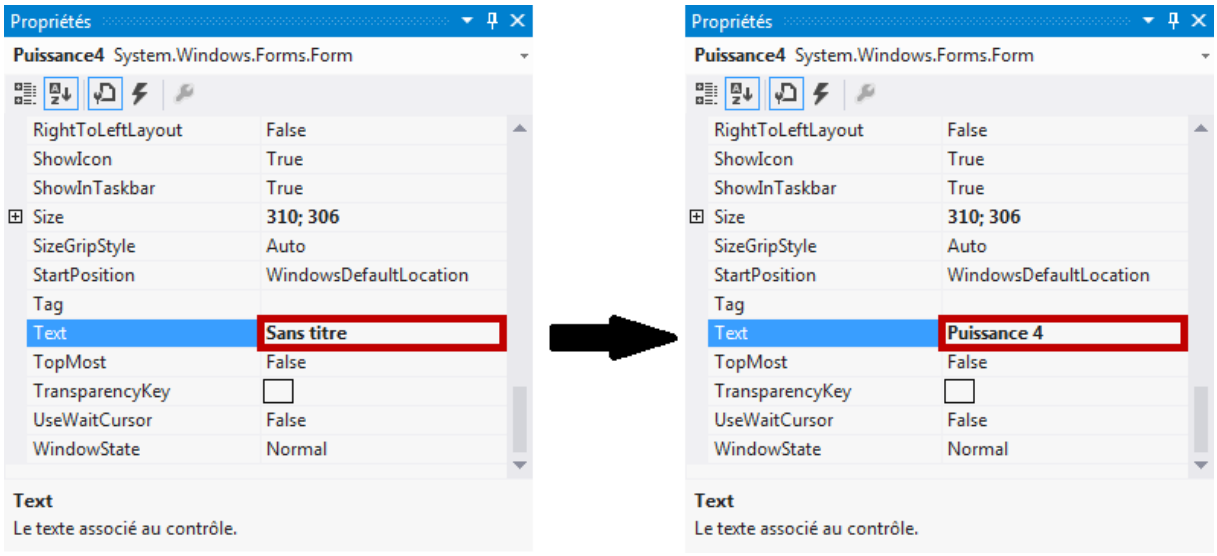
1. Pour débuter

Tout d'abord, vous allez exécuter le programme : cliquez sur la flèche verte en haut ou appuyez sur F5, et jouez un peu avec le programme pour le découvrir.

Une fois que vous vous serez familiarisé avec le jeu, quittez-le et revenez à la fenêtre de Visual Studio. Si la vue centrale est vide, double-cliquez sur *Puissance4.cs* dans l'explorateur de fichiers pour accéder au mode Design.

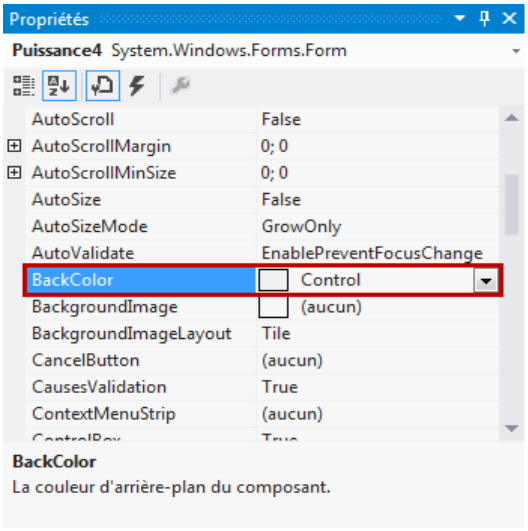
Le titre de la fenêtre

Cliquez sur la fenêtre pour la sélectionner, puis, dans la fenêtre des propriétés, modifiez la valeur de l'attribut `Text` afin de renommer la fenêtre.



L'arrière-plan de la fenêtre

Changez la couleur d'arrière-plan de la fenêtre en modifiant, de la même manière que précédemment, la variable BackColor dans les propriétés de la fenêtre. Vous pouvez sélectionner la couleur que vous voulez mettre.

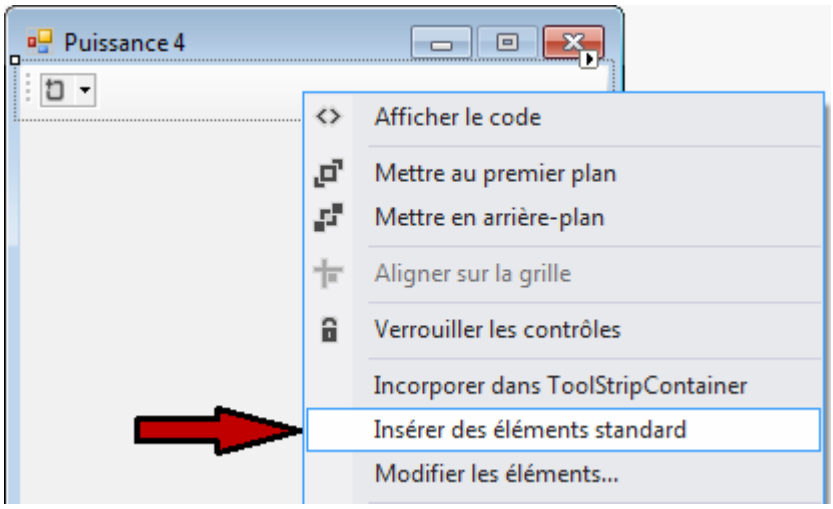


2. Implémentations et modifications

Dans cette partie, vous allez créer un menu qui permettra de lancer une nouvelle partie et de créer une fenêtre d'aide ou un « à propos », que vous personnaliserez vous-même.

La barre de menus

A partir de la boîte à outils (la bande bleue sur le côté gauche de l'écran), insérez par glisser-déposer un objet de type `ToolStrip` en haut de votre fenêtre. Ensuite, insérez-y les éléments standard par un clic droit sur la barre. Enfin, supprimez des icônes afin de ne garder que les boutons « Nouveau » et « Aide ».



Exécutez le programme. Vous pouvez constater que le jeton que l'on déplace est caché par la barre de menus, il faut donc décaler le jeu vers le bas. Pour ce faire, affichez le code de *Puissance4.cs* et modifiez la valeur de la constante `MARGIN_TOP` (essayez de trouver la valeur la plus adéquate).

La fonction Nouveau

Retournez à l'onglet *Puissance4.cs [Création]*, et double-cliquez sur l'icône Nouveau du `ToolStrip`. Visual Studio vous emmène au milieu de la fonction qu'il vient de générer, il ne vous reste plus qu'à l'implémenter :

```
private void nouveauToolStripButton1_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Voulez-vous commencer une nouvelle partie ?",
                        "Nouvelle partie",
                        MessageBoxButtons.YesNo) == DialogResult.Yes)
    {
        init();
    }
}
```

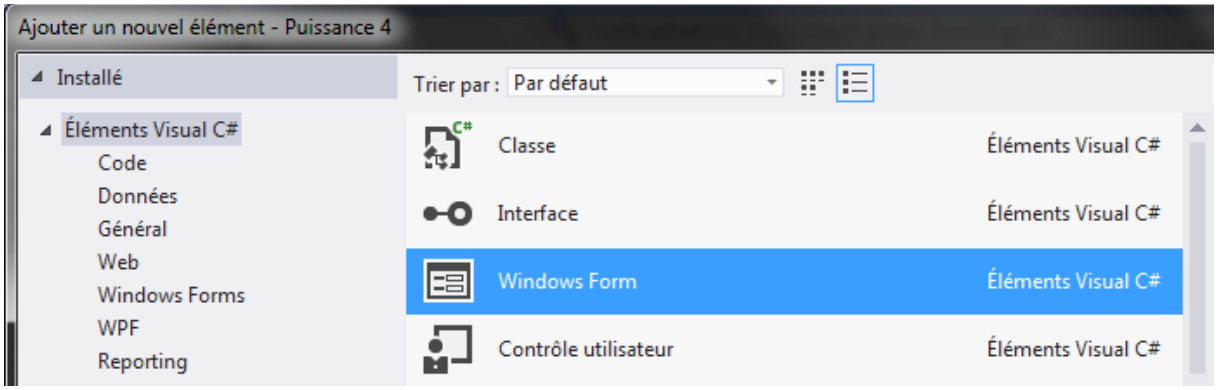
La fonction Aide

Retournez sur la fenêtre en mode *Création* et double-cliquez sur l'icône Aide du `ToolStrip`. De même que précédemment, implémentez cette fonction :

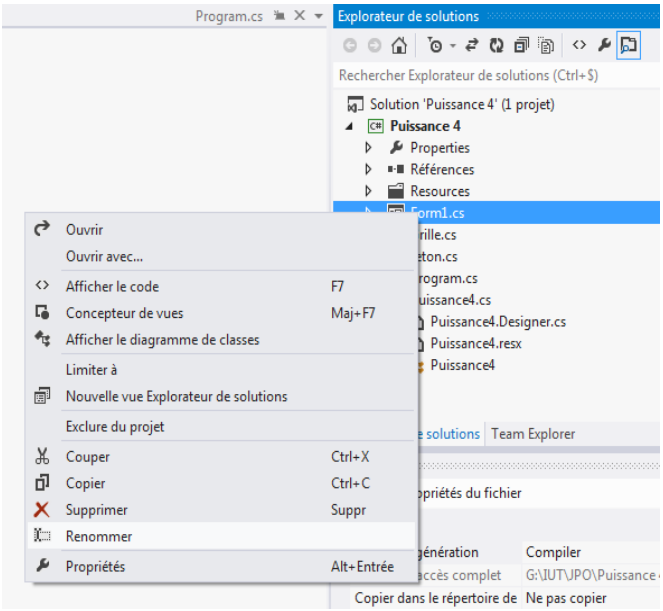
```
private void ToolStripButton1_Click(object sender, EventArgs e)
{
    Apropos about = new Apropos();
    about.showDialog();
}
```

En exécutant, vous pouvez remarquer que plus rien ne marche car il manque l'objet « A propos ».

Pour remédier à ça, allez dans Projet > Ajouter un composant… et sélectionnez Windows Form. Une nouvelle fenêtre a été ajoutée à votre projet, il s'agit de la fenêtre d'aide, ou « A propos ».



Dans l'explorateur de fichiers, renommez le fichier *Form1.cs* en *Apropos.cs*. Cliquez sur « Oui » pour renommer automatiquement tous les fichiers .

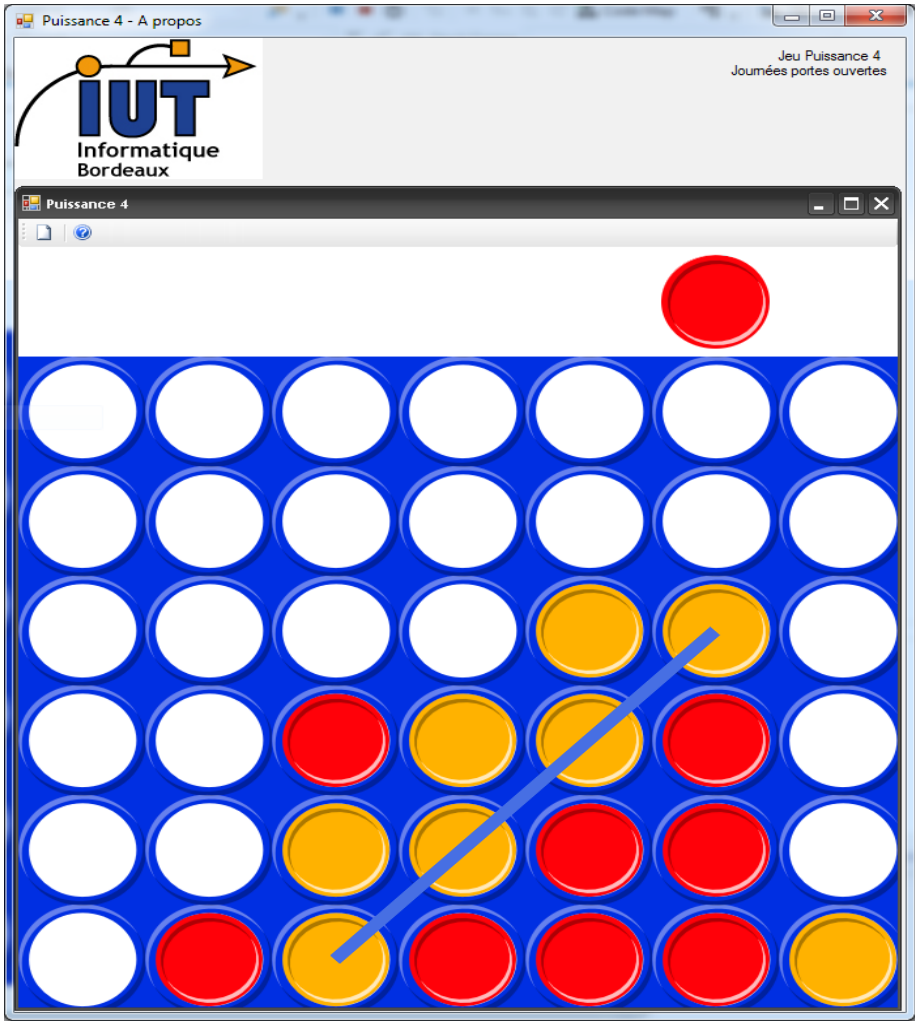


Enfin, dans l'onglet *Apopos.cs*, insérez ce code :

```
internal void showDialog()
{
    ShowDialog();
}
```

Exécutez le programme et cliquez sur le bouton d'aide. Vous pouvez remarquer qu'une fenêtre s'ouvre mais qu'elle est vide. Vous pouvez la modifier comme vous le souhaitez grâce au mode Designer.

Voici un exemple de ce que vous pouvez faire :

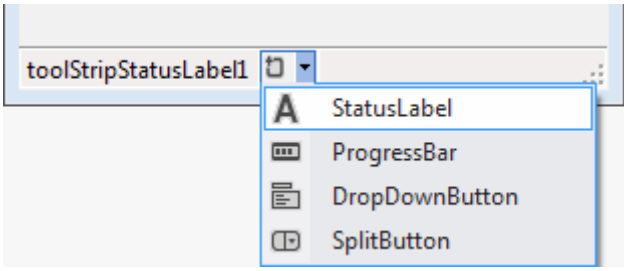


3. Pour aller plus loin

Cette partie consiste à rajouter une barre de statut en bas de la fenêtre pour visionner les scores des joueurs. De plus, la fin de partie est améliorée, car on connaît le vainqueur et le cas d'égalité est géré.

La barre de statut

De la même manière que pour la barre de menus, insérez un objet de type `StatusStrip` en bas de votre fenêtre. Insérez-y ensuite des `StatusLabel` par un clic droit sur l'icône déjà dans la barre.



Effacez les textes des 2 labels en modifiant leur propriété `Text`.

Une fois encore, vous pouvez constater si vous exécutez le programme que le bas du jeu est caché par la `StatusStrip`. Dans le code de *Puissance4.cs*, changez la valeur de la constante `MARGIN_BOTTOM` (encore une fois, trouvez la valeur la plus adéquate).

Implémentation du score

Dans le code de *Puissance4.cs*, modifiez la fonction `Puissance4_MouseClick` afin d'obtenir le code ci-dessous.

```
private void Puissance4_MouseClick(object sender, MouseEventArgs e)
{
    MouseClick

    jetons_gagnants = grille.jetonGagnant(i, j);
    if (jetons_gagnants != null)
    {
        Refresh();//Permet d'afficher quels jetons sont gagnants

        MessageBox.Show("Partie finie !\nVictoire du joueur " + joueur);

        if (joueur == "rouge")
        {
            joueurRouge++;
            toolStripStatusLabel1.Text = "Rouge : " + joueurRouge.ToString();
        }
        else if (joueur == "jaune")
        {
            joueurJaune++;
            toolStripStatusLabel2.Text = "Jaune : " + joueurJaune.ToString();
        }

        init();
    }
    else if (++nbJetons == NB_COLS * NB_ROWS)
    {
        MessageBox.Show("Egalité !");
        joueurRouge++;
        joueurJaune++;
        toolStripStatusLabel1.Text = "Rouge : " + joueurRouge.ToString();
        toolStripStatusLabel2.Text = "Jaune : " + joueurJaune.ToString();

        init();
    }
}
```

Modifiez également la fonction `nouveauToolStripButton1_Click` afin d'obtenir le code ci-dessous.

```
private void nouveauToolStripButton1_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Voulez-vous commencer une nouvelle partie ?",
                        "Nouvelle partie",
                        MessageBoxButtons.YesNo) == DialogResult.Yes)
    {
        joueurRouge = 0;
        joueurJaune = 0;
        toolStripStatusLabel1.Text = "Rouge : 0";
        toolStripStatusLabel2.Text = "Jaune : 0";
        init();
    }
}
```

Enfin, modifiez la fonction Puissance4, pour obtenir le code ci-dessous .

```
public Puissance4()
{
    Puissance4

    Refresh();

    toolStripStatusLabel1.Text = "Rouge : 0";
    toolStripStatusLabel2.Text = "Jaune : 0";
}
```

Voilà ! Vous avez fini notre petit atelier !