

Atelier Programmation

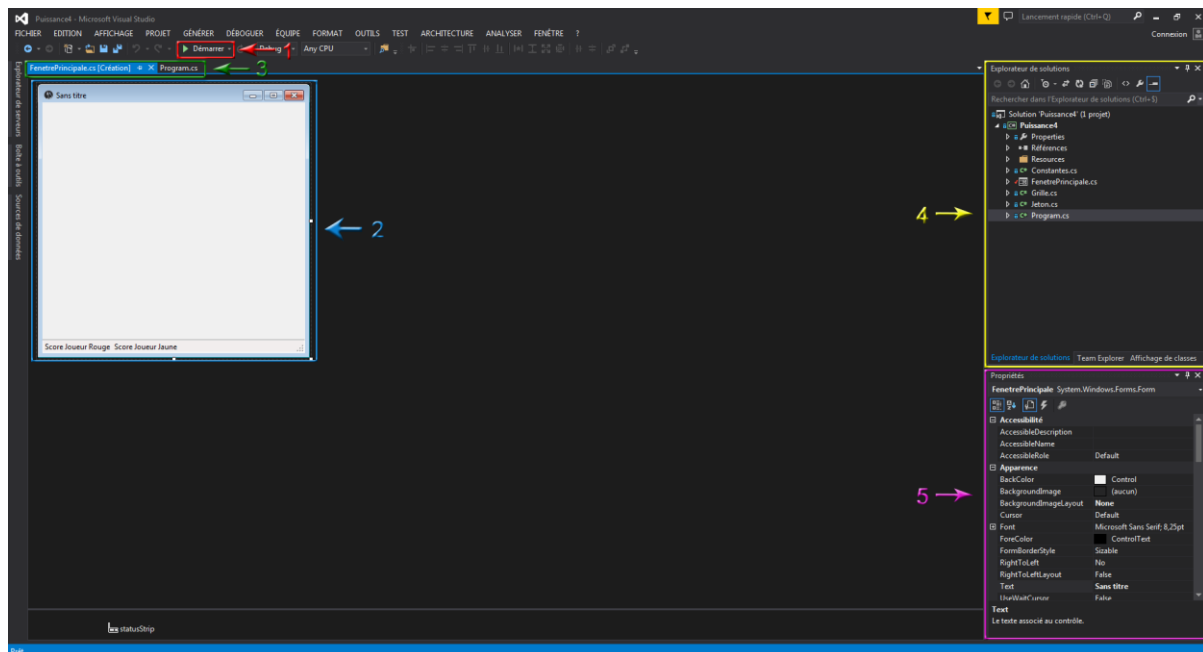
Jeu : Puissance 4

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. Aperçu du programme
3. Onglets des fichiers ouverts
4. Explorateur des fichiers constituant le logiciel
5. Propriétés de l'objet sélectionné

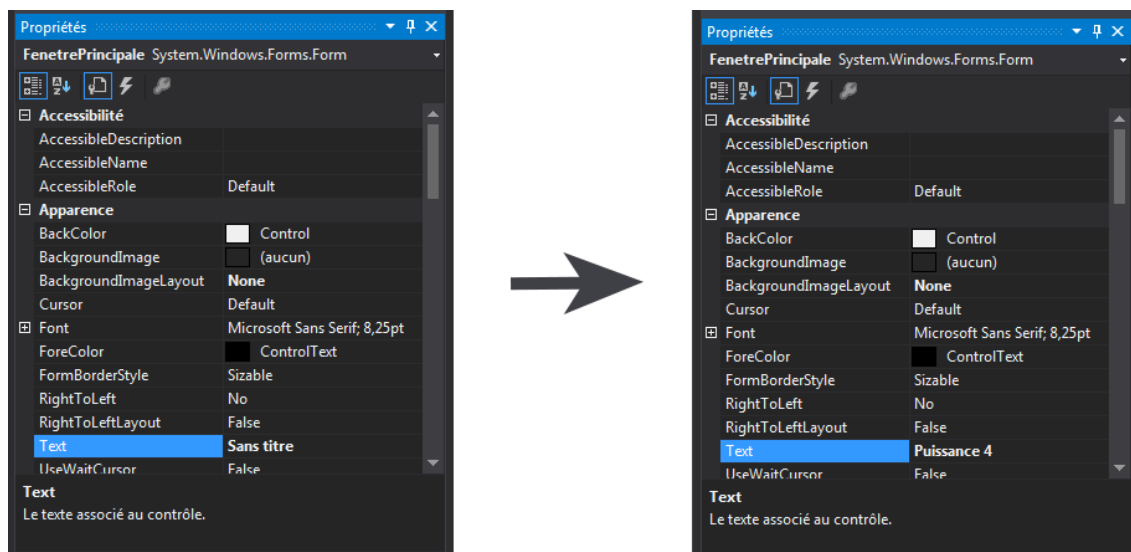
1. Pour débiter

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.

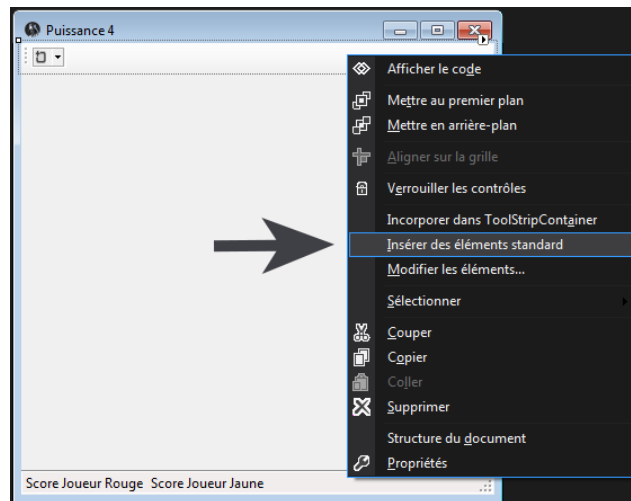


2. Implémentations et modifications

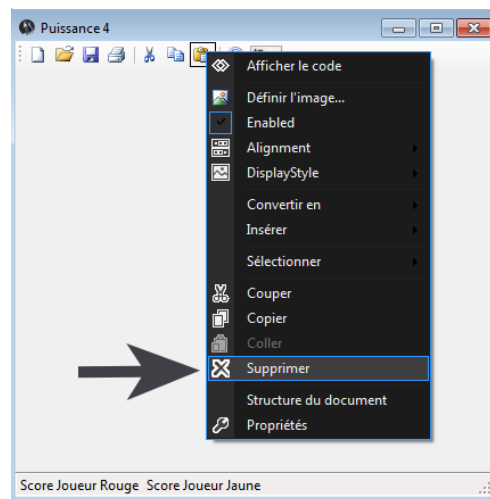
Dans cette partie, vous allez créer un menu qui permettra de lancer une nouvelle partie et de créer une fenêtre « à propos », que vous personnaliserez vous-même.

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.



Supprimez des icônes afin de ne garder que les boutons « Nouveau » et « Aide ». Faites un clic droit sur chaque bouton inutile puis «supprimer».



Modifiez le nom de l'icône pour lancer une nouvelle partie à l'aide de son attribut (`Name`) et le texte affiché lorsque l'on survole le bouton avec la souris grâce à l'attribut `Text`. Par exemple, mettez « nouveauBouton » pour l'attribut (`Name`) et « Nouvelle partie » pour l'attribut `Text`.

Pour le deuxième bouton donnez la valeur « aProposBouton » à l'attribut (`Name`) et la valeur « A propos » à l'attribut `Text`.

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` se trouvant dans le fichier *Constantes.cs* (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 nouveauBouton_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Voulez-vous commencer une nouvelle partie ?", "Nouvelle partie", MessageBoxButtons.YesNo) == DialogResult.Yes)
    {
        initBarreScores();
        initVariables();
    }
}
```

La fonction Aide

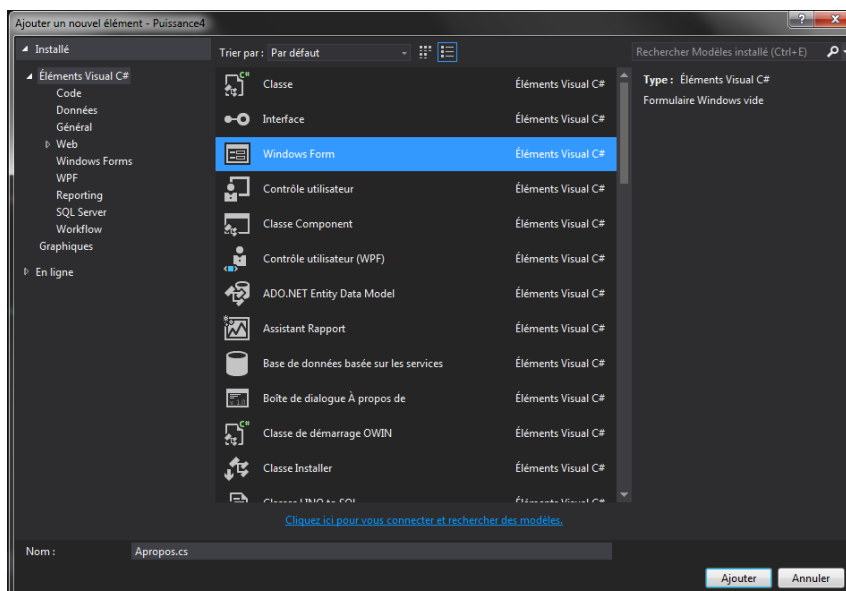
Retournez sur la fenêtre en mode *Création* et double-cliquez sur l'icône Aide du `ToolStrip`. De même que précédemment, implémentez cette fonction :

```
private void aProposBouton_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 ».

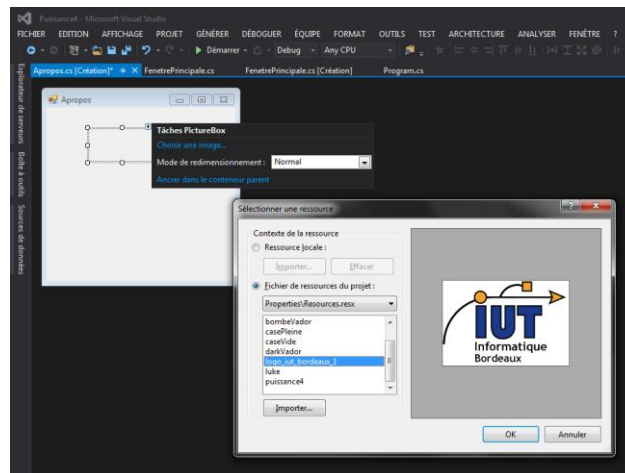
Pour remédier à ça, allez dans *Projet > Ajouter un Formulaire Windows*. Une nouvelle fenêtre a été ajoutée à votre projet, il s'agit de la fenêtre d'aide, ou « A propos ».

Dans le champ « Nom », modifier le texte pour avoir « Apropos.cs ».

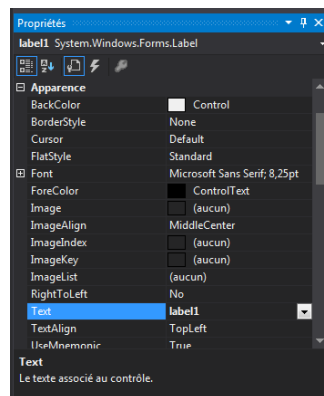


Exécutez le programme et cliquez sur le bouton d'aide. Vous pouvez remarquer qu'une fenêtre s'ouvre mais qu'elle est vide. Vous pouvez la modifier comme vous le souhaitez grâce à la Boîte à outils.

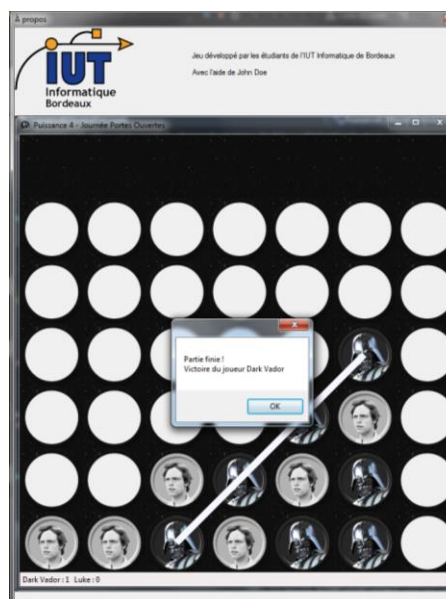
Cliquez sur « Boite à outils » puis faites glisser un objet de type « PictureBox » sur la fenêtre. Cliquez sur la flèche puis cliquez sur « Choisir une image » et enfin choisissez une image dans la liste.



Pour ajouter du texte, faites glisser un élément de type « Label » et modifiez son attribut « Text ».



Voici un exemple de ce que vous pouvez faire :



3. Pour aller plus loin

Dans cette partie, vous allez ajouter un bonus dans le jeu qui se déclenchera lorsque le joueur appuiera sur une touche sur son clavier.

Le bonus est géré dans la fonction `Puissance4_KeyPress()` se situant dans le fichier `FenetrePrincipale.cs`. Cette fonction est utile pour détecter tout appui sur une touche. En fonction de la condition, le programme pourra effectuer des actions différentes.

Pour commencer, vous devez trouver quelle touche active la bombe. Le mot « if » est appelé condition. Ici on teste si le joueur appuie sur une touche précise sur son clavier.

```
private void FenetrePrincipale_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar == 'b' || e.KeyChar == 'B')
    {
        // Code du bonus
    }
}
```

Ajoutez le code suivant qui modifie le jeton du joueur s'il appuie sur la touche de déclenchement des bombes.

```
private void Puissance4_KeyPress(object sender, KeyPressEventArgs e)
{
    if (e.KeyChar == 'b' || e.KeyChar == 'B')
    {
        if (joueur == "darkVador" && bombesVadorRestantes > 0)
        {
            bombesVadorRestantes--;
            joueur = "bombeVador";
            jeton = new Jeton(joueur, jeton.getPosition().X, jeton.getPosition().Y);
        }
        else if (joueur == "luke" && bombeslukeRestantes > 0)
        {
            bombeslukeRestantes--;
            joueur = "bombeLuke";
            jeton = new Jeton(joueur, jeton.getPosition().X, jeton.getPosition().Y);
        }
        Refresh();
    }
}
```

Voilà ! Vous avez fini notre petit atelier !