

Atelier Programmation

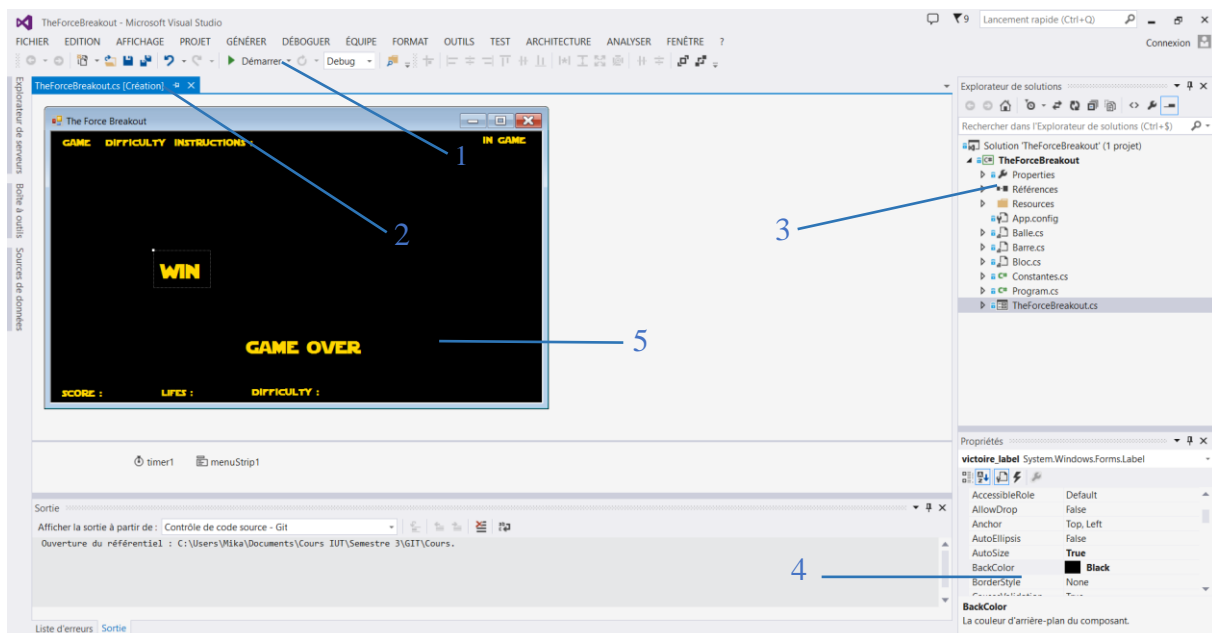
Jeu : The Force Breakout

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

The Force Breakout est un simple jeu de casse-briques : une raquette, une balle et des briques à casser. En le découvrant, vous verrez qu'il faudra le compléter et même y apporter quelques améliorations. 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. **Ouvrez le dossier nommé « TheForceBreakout » et double-cliquez sur le fichier de même nom pour ouvrir Visual Studio.**

Voici l'environnement sous lequel nous allons 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 de la fenêtre

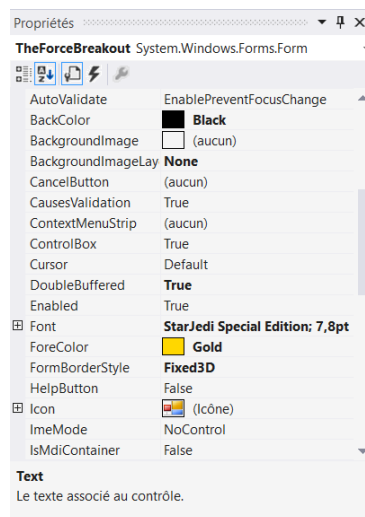
1. Pour débiter

Tout d'abord, vous allez **lancer l'exécutable fourni à la racine du dossier**. Pour lancer le jeu, pressez la touche **Entrée** ou cliquez. Pour relancer la balle après la perte d'une vie, pressez la barre d'espace. Pour **changer le niveau du jeu**, cliquez sur « **Level** » puis choisissez un niveau. Vous avez 3 vies.

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 « **TheForceBreakout.cs** » dans l'explorateur de fichiers pour accéder au **fichier en mode Création**.

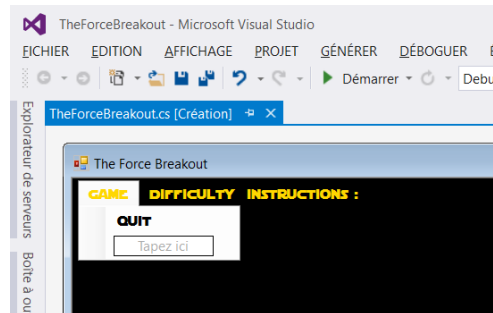
Modification des informations de la fenêtre

Modifiez le titre de la fenêtre du programme : sélectionnez la fenêtre principale par un simple clic dessus, puis dans la fenêtre de propriétés (voir ci-dessous), cherchez l'attribut **Text** et modifiez son champ d'origine « **The Force Breakout** » (voir l'image ci-dessous). Vous pouvez aussi **par exemple ajouter un logo à l'application en cliquant sur le champ de l'attribut Icon** puis en sélectionnant l'icône présent dans le dossier nommé « **Resources** » du projet.



Nouvel élément de barre d'outils

Ajoutez un nouvel item « **Quit** » dans le menu « **Game** » : Pour cela, cliquez sur le bouton « **Game** » directement dans la fenêtre du jeu à gauche, puis sur « Tapez ici » pour rajouter un item à ce menu déroulant. Ecrivez « **Quit** », puis validez avec la touche Entrée (voir l'image ci-dessous).



Comme vous le voyez, la couleur de la police est noire et le fond du bouton est gris par défaut. Pour changer cela, cliquez sur votre nouveau bouton et changez les propriétés BackColor et ForeColor. Par exemple, placez-vous dans la rubrique « Web et sélectionnez « black » pour BackColor et « gold » pour ForeColor.

Exécutez votre programme avec la petite flèche en haut de l'écran, ou en appuyant sur la touche « F5 ».

Vous voyez votre nouveau bouton créé. Cependant si vous cliquez dessus il ne fonctionne pas. Fermez votre programme.

Pour que ce nouveau bouton fasse effet, rendez-vous à nouveau dans Visual Studio. Double-cliquez sur ce dernier. Le logiciel vous place alors immédiatement au milieu du code qu'il aura généré. Ce code gère donc l'événement Click de «Quit», c'est-à-dire le cas où l'utilisateur clique sur le bouton «Quit».

Ecrivez simplement « `this.Close();` » comme ci-dessous.

```
private void quitToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.Close();
}
```

Exécutez le programme pour vérifier que votre programme fonctionne correctement.

Créer la balle

Comment jouer si la base du jeu n'existe pas ? Je vous le demande... Nous allons construire la balle, un des éléments principaux du jeu.

Pour cela, rendez-vous dans l'explorateur afin de sélectionner le fichier « TheForceBreakout » disponible en dépliant le volet « TheForceBreakout.cs ». En fait, lorsqu'on double-clic simplement sur « TheForceBreakout.cs », Visual Studio ouvre le mode « création » qui nous permet d'interagir graphiquement avec le programme. Cependant il nous faut accéder au mode « code » du fichier. Le fichier de code est normalement déjà ouvert depuis que vous avez implémenté le bouton « quit ». Vous devez voir dans la barre d'onglets le fichier « TheForceBreakout.cs » qui correspond au fichier en mode « code » et le fichier « TheForceBreakout.cs [Création] » qui correspond au mode fichier en mode « création ».

Maintenant que vous êtes dans le fichier de code, vous voyez que le code est découpé en plusieurs régions. Ces régions sont simplement utilisées pour rendre le code plus clair.

Comme je l'ai dit, il faut « construire » la balle donc allez dans la région nommée « FONCTIONS DE CONSTRUCTION » et trouvez l'action suivante :

```
// Cette action sert à mettre en place la balle
public void miseEnPlaceDeLaBalle()
{
}
}
```

Pour construire la balle, il suffit de la créer et de l'ajouter à ce que Visual Studio nomme « Controls ». En fait, ceci correspond simplement à notre programme.

Ajoutez les lignes suivantes entre les accolades pour créer la balle :

```
balle = new Balle();
this.Controls.Add(this.balle);
```

Créer la barre

De manière extrêmement similaire, si vous voulez jouer, il va falloir vous munir d'une raquette. Pour cela rien de plus simple... Comme précédemment, rendez-vous dans l'action :

```
// Cette action sert à mettre en place la barre
public void miseEnPlaceDeLaBarre()
{
}
}
```

Puis rajouter les deux lignes nécessaires afin de créer une barre et de l'ajouter au programme.

Changer la bordure des blocs

Ici, on veut changer un peu l'apparence des blocs. Pour cela, il faut modifier le constructeur d'un bloc. Pour cela, double-cliquez sur Bloc.cs dans l'explorateur. Cliquez ensuite sur « cliquez ici pour passer en mode code ». Il ne vous reste plus qu'à repérer le constructeur d'un bloc et de changer le style « FixedSingle » par « None » ou « Fixed3D ».

```
public Bloc(int i, int j, Image image_bloc)
{
    this.BackgroundImage = image_bloc;
    this.BackgroundImageLayout = ImageLayout.Stretch;
    this.BorderStyle = BorderStyle.FixedSingle;

    this.Location = new Point(largeur * j, hauteur * i + 28);

    this.Size = new Size(largeur, hauteur);
}
```

Déplacer la balle

Bon vous allez me dire, c'est bien beau tout ça. J'ai une balle, une raquette, des blocs mais je joue comment ? Elle ne bouge pas ma balle ! Je vous réponds : tranquille, ça arrive...

Rendez-vous dans le fichier « TheForceBreakout.cs » en mode code jusqu'à la région nommée « GESTION DES EVENEMENTS ». Trouvez l'action qui gère les événements lorsque le jeu est en marche. Normalement vous n'aurez pas trop de mal à la trouver sachant qu'elle se nomme « public void jeuEnMarche() » Je dis bien normalement... Mais je ne doute pas de vos compétences. 😊

```
// Cette action permet d'afficher ce qu'il faut lorsque le jeu est en état de marche
public void jeuEnMarche()
{
    if (etat_du_jeu == Etat.JOUE)
    {
        this.fin_label.Text = "score : " + score;
        this.etat_label.Text = "in game";
        this.instructions_label.Text = "press escape to pause";

        /*for (int i = 0; i < vitesse_balle; i++)
        {
            balle.bouger();
            collisionBlocs(balle);
        }*/
    }
}
```

Ici, rien à écrire c'est déjà fait, il suffit de décommenter le code car pour le moment, il n'est pas pris en compte par le programme. Effacez les caractères « /* */ ». Je vais vous expliquer brièvement ces 5 lignes décommentées par vos soins.

Ce qui va réellement faire bouger la balle est `balle.bouger()`; . Cependant je veux que la balle ne se déplace pas d'un à un pixel mais de plusieurs pixels à la fois. Pour cela, j'utilise ce que l'on nomme une boucle « for ». Cela permet simplement de répéter tout ce que je veux, le nombre de fois que je veux. Ici je fais bouger la balle d'un pixel avec `balle.bouger()`; et je vérifie si la balle est en collision avec un

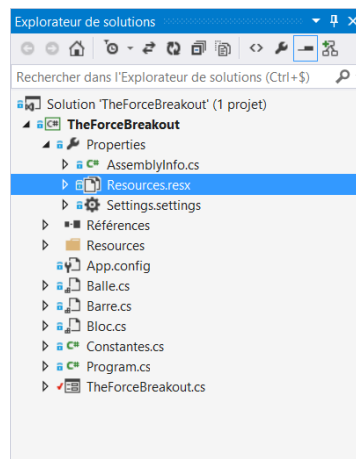
bloc par `collisionBlocs(balle);`. Et je répète ces opérations du nombre de fois que je le veux c'est-à-dire pour `i < vitesse_balle` soit `vitesse_balle - 1`.

Si vous avez compris tant mieux, sinon ne vous en faites pas... Vous aurez largement le temps de comprendre cela en cours lors de votre premier semestre à nos côtés !

Prenez l'habitude de refermer les régions lorsque vous avez fini votre affaire.

Rajouter un fond d'écran

Actuellement, le jeu est doté d'un seul fond d'écran. Rendez-vous dans l'explorateur. Dépliez le volant « Propriétés » et double-cliquez sur « Resources.resx » afin d'ajouter une image de fond d'écran à exploiter.



Vous voyez maintenant toutes les images utilisées et déjà présentes dans le jeu. Pour rajouter une nouvelle image, cliquez sur la petite flèche à droite du bouton « Ajouter une ressource » et sélectionnez « Ajoutez un fichier existant ». Allez maintenant dans le dossier du jeu « TheForceBreakout » et dans le dossier des ressources « Resources ». Vous voyez qu'il y a déjà un autre background nommé « background_2 » prédestiné à être sélectionné par vos soins. Quelle chance !... Sélectionnez-le et cliquez sur le bouton « Ouvrir ». Votre image est prête à être utilisée. Maintenant il faut dire au programme d'utiliser soit le premier, soit le second. Rendez-vous dans le fichier « TheForceBreakout.cs » en mode code.

Dépliez la région nommée « FONCTIONS DE CONSTRUCTION » et repérez l'action « `public void miseEnPlaceDuBackground()` » comme ci-dessous.

```
// Cette action sert à mettre en place l'image d'arrière-plan du jeu parmi les fonds d'écran disponibles dans les ressources.
public void miseEnPlaceDuBackground()
{
    Random rnd = new Random(DateTime.Now.Millisecond);
    int x = rnd.Next(1, 1);
    switch (x)
    {
        case 1:
            this.BackgroundImage = global::TheForceBreakout.Properties.Resources.background_1;
            break;
    }
}
```

Ici, un nombre entier nommé x est tiré aléatoirement entre 1 et 1 ce qui donne... 1 ! Il faut donc choisir un nombre entre 1 et 2 comme ceci :

```
int x = rnd.Next(1, 3);
```

Votre nombre tiré, il faut maintenant qu'il prenne effet. Rajoutez un nouveau cas à la suite du premier pour que votre nouveau background soit pris en compte :

```
case 2:
    this.BackgroundImage = global::TheForceBreakout.Properties.Resources.background_2;
    break;
```

2. Pour aller plus loin

Activer un bonus multi-balles

Si vous lisez ces lignes mon jeune padawan, c'est que vous avez assez de talents pour passer les épreuves et devenir chevalier Jedi. Je vous propose ainsi d'écrire un peu de code.

Par défaut, j'ai décidé que ce bonus s'activera lorsque l'on appuie sur la touche 'M' du clavier.

Pour cette fonctionnalité, il va falloir prouver votre maîtrise de la force. Trouvez l'emplacement adéquat dans la région FONCTIONS DE CONSTRUCTION, et créez deux autres balles que vous nommerez balle2 et balle3. Comme vu précédemment, il faut construire les balles et les ajouter toutes les deux au programme.

La difficulté ici est de pouvoir différencier les trois balles. En effet, si on construit simplement les deux nouvelles balles, on ne pourra les différencier car elles seront au même endroit et bougeront à la même vitesse. Une solution simple consiste à inverser le déplacement en X de la deuxième balle et de multiplier le déplacement en Y par 2 de la troisième balle. De cette façon les balles partiront toutes dans des directions différentes.

On accède aux déplacements des balles de cette façon : `balle2.DeplacementX = quelque chose`.

Une autre difficulté nous fait face. Comme nous avons créé les deux autres balles et qu'on les a ajoutés au programme, elles vont s'afficher dès le lancement du jeu. Or, nous voulons qu'elles s'affichent seulement lorsqu'on le veut.

Il faut alors cacher les nouvelles balles jusqu'à l'activation du bonus.

On paramètre la visibilité d'un élément de cette façon : `balle2.visible = true` ou `false`.

Une fois cela écrit, fonctionnera le multi-balles.

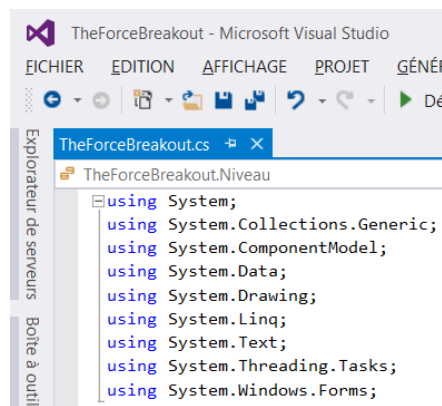
Ajouter de la musique

Afin de ressentir le pouvoir de la force, je vous propose un dernier challenge. Vous allez ajouter la musique de votre film de science-fiction préféré.

Pour cela, ajoutez aux ressources la musique de la même façon que vous avez ajouté le second background toute à l'heure. Lorsque vous êtes dans l'explorateur Windows pour rechercher la musique dans le dossier « Ressources » du projet, veillez à sélectionner tous les types de fichier et non pas que les images sinon la musique ne s'affichera pas. La partie intéressante arrive maintenant.

Nous avons tout d'abord besoin de créer une instance correspondant à notre musique. Pour cela, rendez-vous dans le fichier « TheForceBreakout.cs » toujours en mode code.

Vous l'aurez peut-être remarqué, ou peut-être pas.... Mais on a besoin d'inclure certaines bibliothèques afin de disposer de code déjà écrit qu'il n'est pas nécessaire de recoder à chaque nouveau programme. Ces bibliothèques sont utilisées de cette manière :



Notre musique aura besoin de la bibliothèque « System.Media » pour fonctionner. Je vous demande de l'inclure. Ensuite rendez-vous dans la région LES VARIABLES DU JEU afin de créer notre instance.

Le type de la musique est public `SoundPlayer`. Vous la nommez comme vous le voulez.

Il va maintenant falloir construire notre musique. Pour cela, allez dans la région FONCTIONS DE CONSTRUCTION et construisez votre musique toujours comme vu précédemment dans le constructeur

```
public TheForceBreakout() :
```

Une petite précision s'impose. En effet, vous venez de construire votre musique. Très bien et je vous en félicite. Mais vous n'avez pas précisé quelle musique votre programme doit charger. Il faut préciser dans les parenthèses le chemin vers votre musique. Comme vous l'avez ajouté précédemment aux ressources de votre programme, il faut préciser le chemin de la manière suivante :

```
audio = new SoundPlayer(global::TheForceBreakout.Properties.Resources.StarWars_Theme);
```

Il ne vous reste plus qu'à lancer votre superbe musique de la manière suivante :

```
audio.Play();
```

Trivial non ?

Je vous félicite, vous êtes arrivé à la fin de cette session. Vous connaissez maintenant les secrets de la force et je vous décerne le statut de chevalier Jedi. Vous en voulez encore pour espérer devenir un jour peut-être maître Jedi ? Inscrivez-vous à l'IUT et nous vous apprendrons toutes les subtilités et les secrets du pouvoir du code... 😊

Merci à vous et à bientôt !