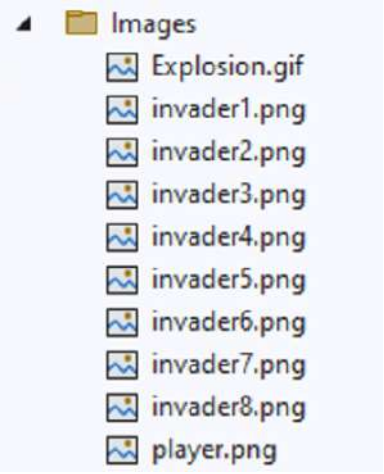
SEANCE 1 – SPACE INVADERS

OBJECTIFS

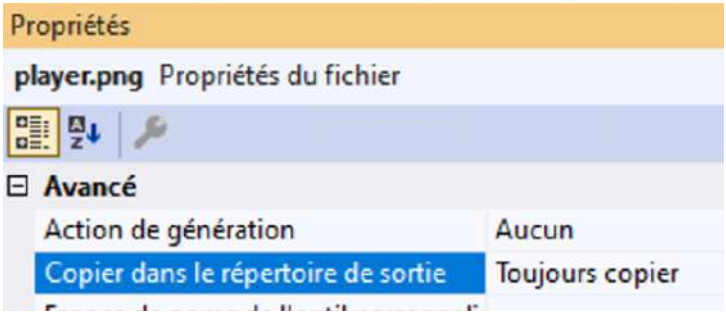
- Mettre en œuvre le jeu Space Invaders vu en cours
- Apporter des améliorations au jeu

PARTIE 1 – PREPARATION DU PROJET

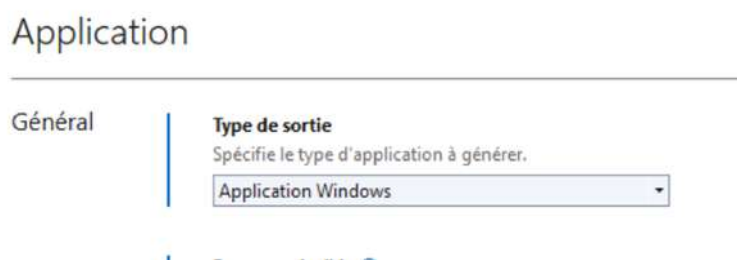
- Créer un nouveau projet WPF SpaceInvaders
- Créer un répertoire images et importer les images sur le réseau



Modifier les propriétés des images en « Toujours copier » :



- Modifier si nécessaire le projet en application Windows :



Remarque : vous pouvez basculer en mode console si vous désirez avoir une fenêtre console en parallèle (utile pour debugger par exemple)

- Modifier l'icône de votre application.
- Vérifier que le code compile

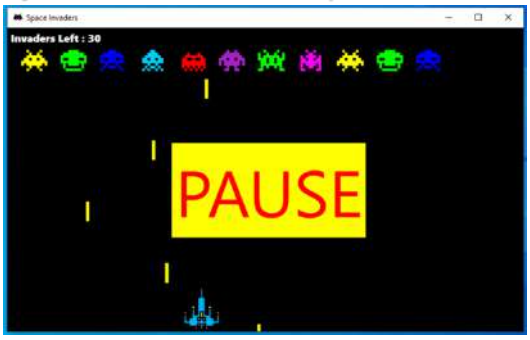
PARTIE 2 – IMPLEMENTATION DU CODE

- Implémenter le code et vérifier qu'il fonctionne
- Améliorer la lisibilité du code en créant des méthodes à appeler dans GameEngine : Structurer de cette manière :

```
1 référence
private void GameEngine(object sender, EventArgs e)
{
    // création d'un rectangle joueur pour la détection de collision
    Rect player = new Rect(Canvas.GetLeft(player1), Canvas.GetTop(player1), player1.Width, player1.Height);

    // complète l'affichage du nombre d'ennemis tués
    enemiesLeft.Content = "Invaders Left : " + totalEnemies;

    MovePlayer();
    MakeBulletTimer();

    // parcours de la liste des rectangles d'objets du canvas
    {
        foreach (Rectangle x in myCanvas.Children.OfType<Rectangle>())
        {
            MoveAndTestBulletPlayer(x);
            MoveEnemiesAndTestCollisionPlayer(x, player);
            MoveAndTestEnemiesBullet(x, player);
        }
    }
    RemoveItemsToRemove();
    TestWin();
}
```

PARTIE 3 – AMELIORATION DU CODE

- Supprimer le messageBox de fin ou de gagne pour le remplacer par un label au milieu de l'écran
- Ajouter la fonctionnalité pause. Pour simplifier pour pourrez utiliser la touche P et R



- Vous allez créer une première page pour choisir la difficulté du jeu

- Créer un stackpanel contenant les composants à afficher.
- Référencer les composants avec des noms pour pouvoir les afficher/cacher dynamiquement en behind (code C#).
- Coder la méthode correspondant au clic sur le bouton jouer

Voici les règles de difficulté :

	Vitesse ennemis	Vitesse tir ennemi	Vitesse joueur	Vitesse tir ennemi	Limite Timer tir ennemi
Facile	3	3	20	3	90
Moyen	6	5	10	5	60
Difficile	8	10	10	10	30

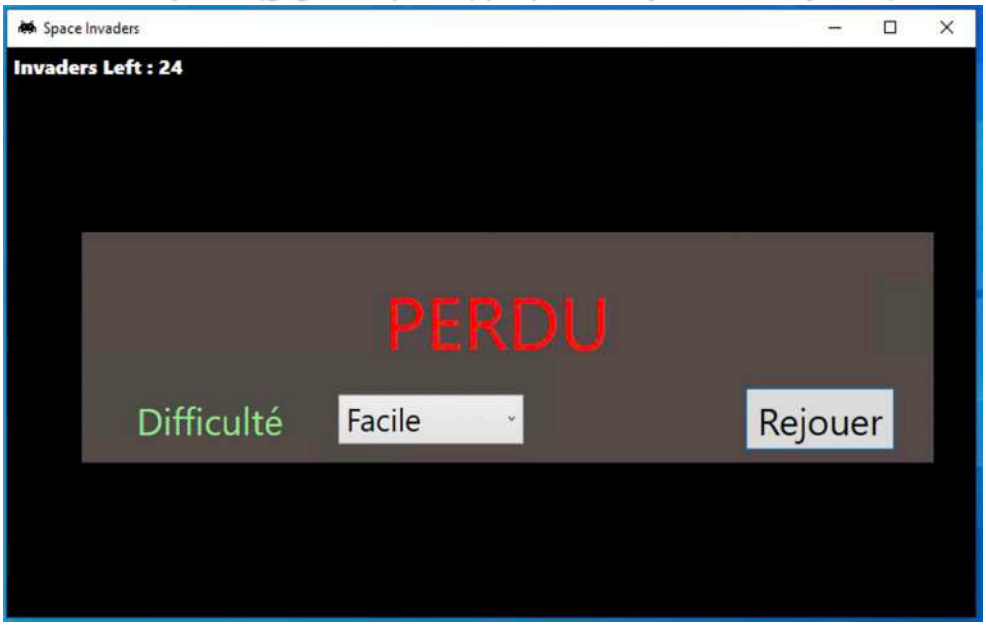
PARTIE 4 – EXPLOSION ET REJOUER

- Lorsque vous perdez afficher un gif animé d'explosion

Vous pourrez utiliser le package nugget :

Le wiki se trouve ici :
https://github.com/XamlAnimatedGif/WpfAnimatedGif/wiki

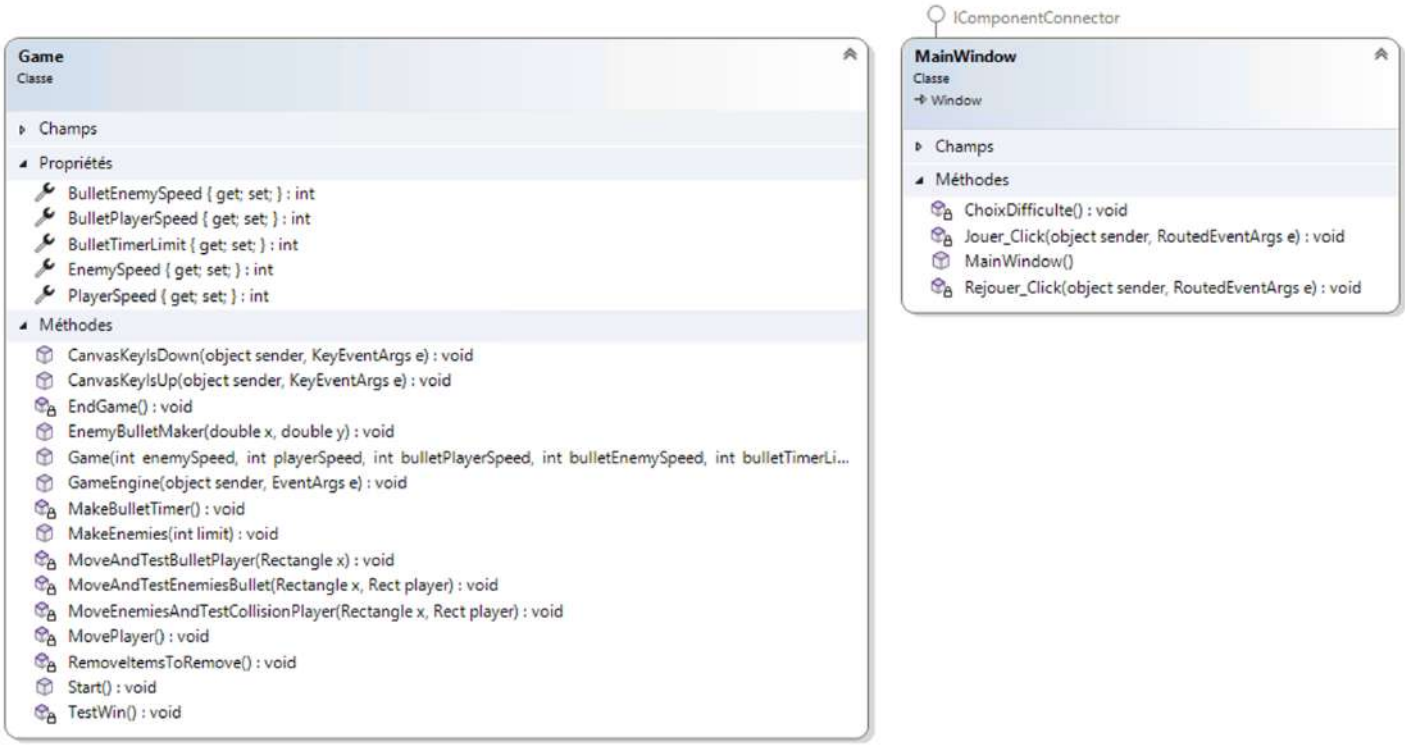
- A la fin de la partie (gagné ou perdu) proposer au joueur de rejouer (utiliser le stackpanel)



En profiter pour afficher le high score en haut à droite

PARTIE 5 – POUR LES PLUS RAPIDE

Créer une classe Game. Votre MainWindows instanciera un objet de type Game :



Remarque :

Si vous désirez accéder à des composants du MainWindow.xaml dans la classe Game, utiliser :

```
private MainWindow elementsMainWindow;
elementsMainWindow = ((MainWindow)System.Windows.Application.Current.MainWindow);
```