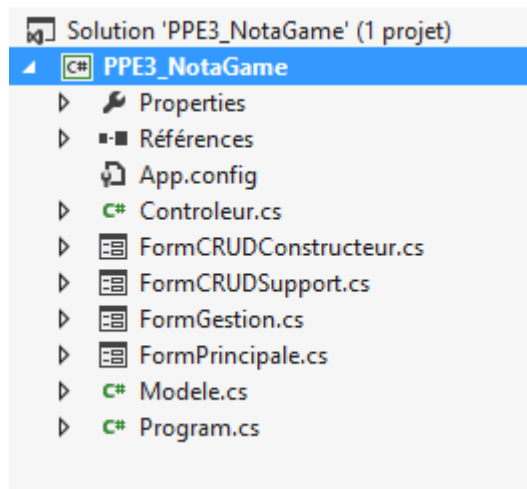


PPE3 : NOTAGAME

Explications Partie Application C#

- Code fourni pour l'application C# existante : https://github.com/autret/PPE3_NotaGame_C-
- une **application C#** en M.V.C et en mode connecté pour la base de données
 - qui permet de faire le CRUD (Create Read Update Delete) sur les tables CONSTRUCTEUR et SUPPORT de la base de données.
- Documentation : **PPE3_NotaGame.XML** dans bin/debug
- Structure de l'application existante :



- Rappel de la **Base de données commune** aux 2 applications :
 - table JeuxVideos (**idJV**, nomJV, anneeSortie, classification, editeur, description) : jeux videos
// classification correspond à un âge minimum conseillé
 - table Support (**idS**, nomS, anneeSortie, caractéristiques, idC#) : console de jeux
 - table Compatible (**idJV#**, **idS#**) : permet de vérifier la compatibilité d'un jeu en fonction d'une console
 - table Constructeur (**idC**, nomC) : liste des constructeurs de supports
 - table Users (**idU**, email, pseudo, communauté) : liste des utilisateurs qui ont posté un commentaire. la communauté est une information pouvant être (geek, joueur occasionnel, etc...)
 - table Commentaire (**idJV#**, **idU#**, libelle) : liste des commentaires mis sur les jeux par les utilisateurs.
- Les données en connexion avec la BD sont gérées dans des DataAdapter et DataTable via des collections dans la classe MODELE :

```
// les DataAdapter et DataTable seront gérés dans des collections avec pour chaque un
indice correspondant :
// indice 0 : récupération des noms des tables
// indice 1 : Table Constructeur
// indice 2 : Table Support avec jointure pour récupérer tous les libellés
```

```
// indice 3 : Table Support

// collection de DataAdapter
private List<MySqlDataAdapter> dA = new List<MySqlDataAdapter>();

// collection de DataTable récupérant les données correspond au DA associé
private List<DataTable> dT = new List<DataTable>();
```