Comptes-rendus d’expérience

**Sujet : Delta jeu de dame**

**Cours:** RAC **Etudiants: Aebischer, Salomon**

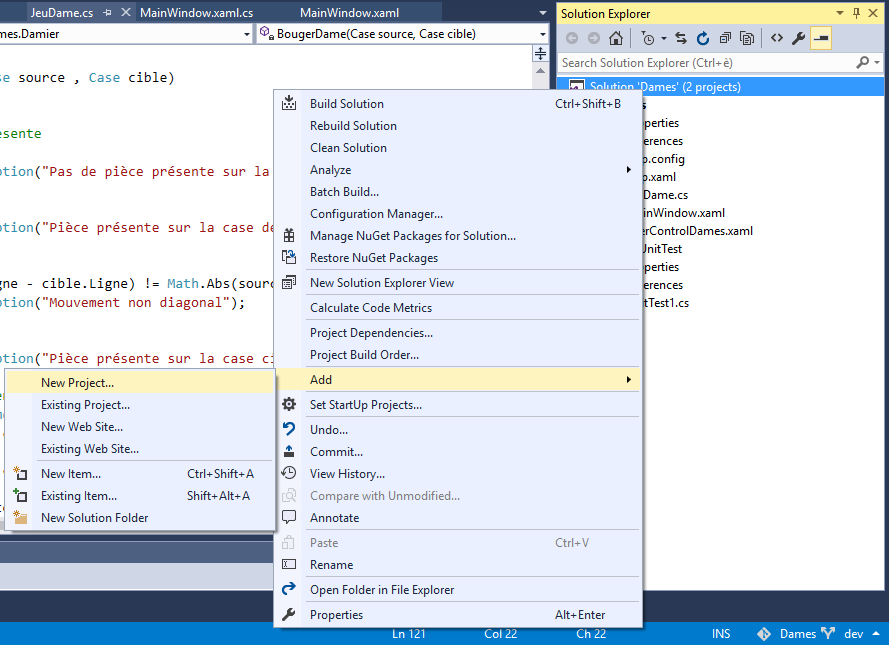
# Objectifs du travail

Créer un projet de test unitaire en C# (UnitTest) permettant de tester toutes les classes du jeu de dames. Ces classes sont décrites dans un diagramme UML.

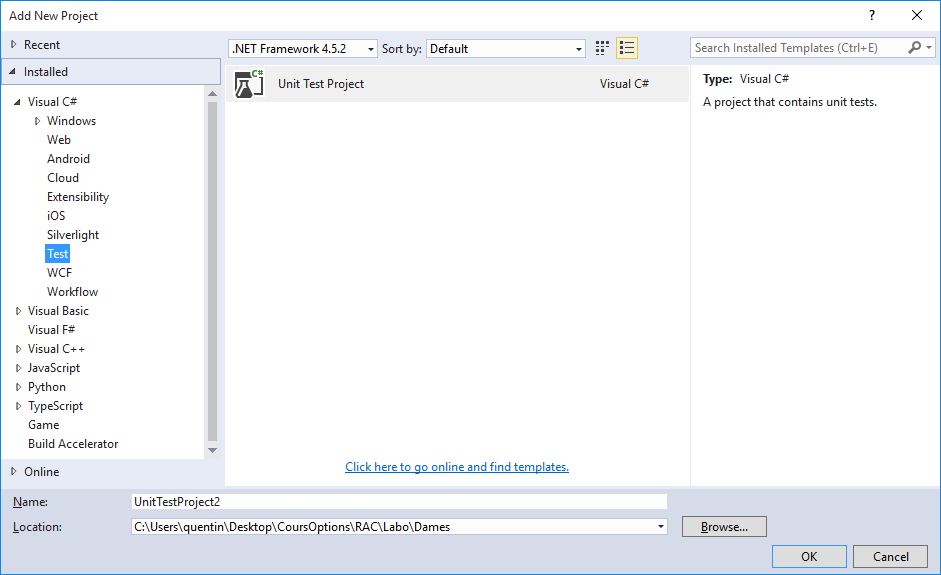
Pour tester, il faut implémenter le code testant les toutes les fonctionnalités, puis exécuter celui-ci.

# Essais réalisés

Pour créer un test unitaire, il faut faire un clic droit sur la solution 🡺 Add 🡺New Project…



Puis Ajouter un unit test en c# :



Une fois le test unitaire créer il ne reste plus qu’à coder les tests que nous voulons exécuter.

Pour avoir les classes à dispositions dans le fichier UnitTest.cs, il faut ajouter le using contenant les classes au début du fichier. Dans notre cas cela donne : “using Dames;”

Pour le jeu de dames, nous avons fait des tests rapide pour :

* La création du damier
* La création de cases
* L’occupation des cases créées (les cases ont une propriété « Occupe »)
* La création de pion et l’assignation à une case
* Le déplacement d’une pièce

Voici le code permettant :

using System;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using Dames;

namespace DameUnitTest

{

[TestClass]

public class UnitTest1

{

[TestMethod]

public void TestMethod1()

{

var dame = new Damier();

Assert.IsNotNull(dame);

Case c1 = dame.getCase(0, 0);

Assert.IsNotNull(c1);

Assert.IsFalse(c1.Ocuppe);

//Ajout d'une pièce en 0,0

Pion p1 = new Pion(couleur.BLANC);

c1.MettrePiece(p1);

Assert.IsTrue(c1.Ocuppe);

Assert.IsFalse(dame.getCase(1, 1).Ocuppe);

//Bouger la pièce en 1,1

try

{

dame.BougerPiece(0, 0, 1, 1);

}

catch(DameException e)

{

Assert.Fail(e.Message);

}

Assert.IsFalse(c1.Ocuppe);

Assert.IsTrue(dame.getCase(1, 1).Ocuppe);

//Pose pièce en 2,2

Pion p2 = new Pion(couleur.NOIR);

dame.getCase(2, 2).MettrePiece(p2);

//Bouger pion de 1,1 en 3,3 pour manger le pion en 2,2

try

{

dame.BougerPiece(1, 1, 3, 3);

}

catch (DameException e)

{

Assert.Fail(e.Message);

}

Assert.IsFalse(dame.getCase(2, 2).Ocuppe);

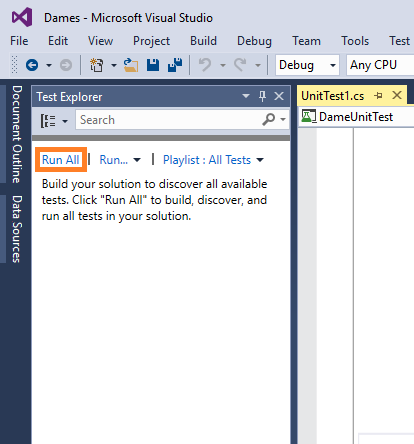
}

}

}

Après cela, tous les tests possibles n’ont pas étés fait, mais ceux-là suffisent pour ce que nous avons à faire par la suite. Nous avons aussi corrigé quelques bugs par la suite lors de la partie graphique du jeu de dames.

Pour l’exécution du code test, il n’y a pas besoin de lancer l’application complète. Il est possible d’exécuter que l’UnitTest à l’aide de la fenêtre « Test explorer » et le bouton « Run All »:



# Résultats et conclusion

Le test unitaire nous a permis de tester toutes nos classes et de pouvoir coder l’interface du jeu de dames en sachant que tout fonctionne. Ceci permettant de savoir que si une erreur surgit elle vient du code de l’interface et non du jeu de dames.

Date: 09.06.2016 Signatures : Aebischer, Salomon

HESSO.HEIG-VD.iAi.JDZ, 13 mars 2014