

JEDI TOURNAMENT 2016

Par :

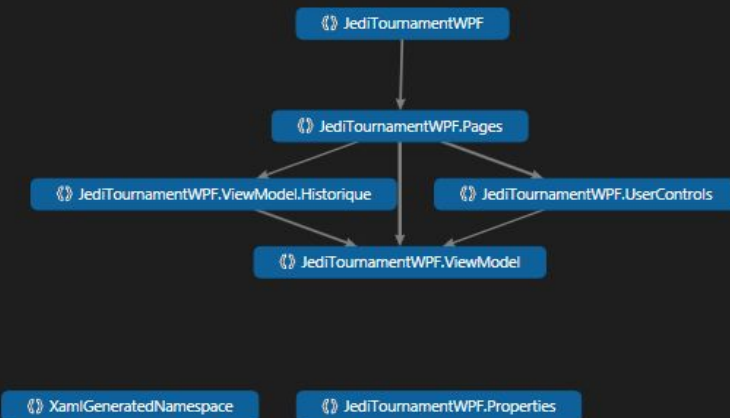
Adrien Pierreval

Oumaima El Amrani

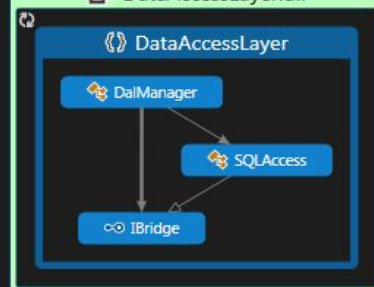
Nicolas Chabaliier

Vincent Mizoules

JediTournamentWPF.exe



DataAccessLayer.dll

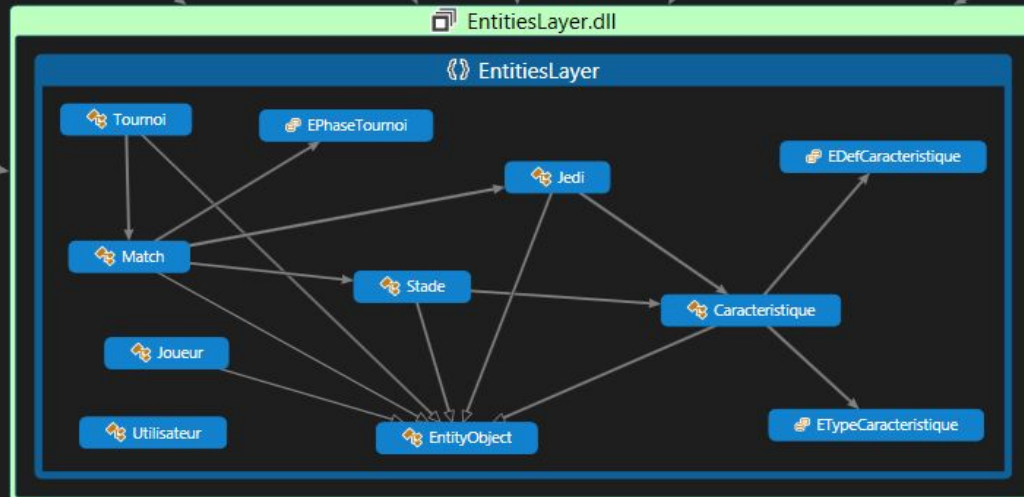


BusinessLayer.dll

StubDataAccessLayer.dll

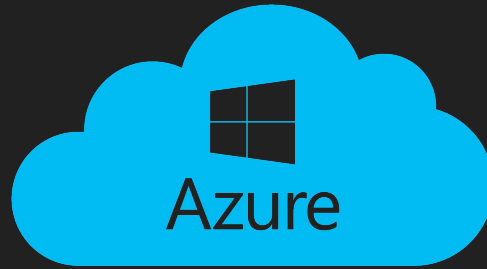
DataAccessLayerTest.dll

EntitiesLayer.dll



JediTournamentConsole.exe

Moyens



Gestion Projet

Découpage en parties :

Adrien [*chef projet*] : Déroulement des tournois (mode automatique/manuel)

Vincent : Mise en place de la création du Tournoi + Design graphique

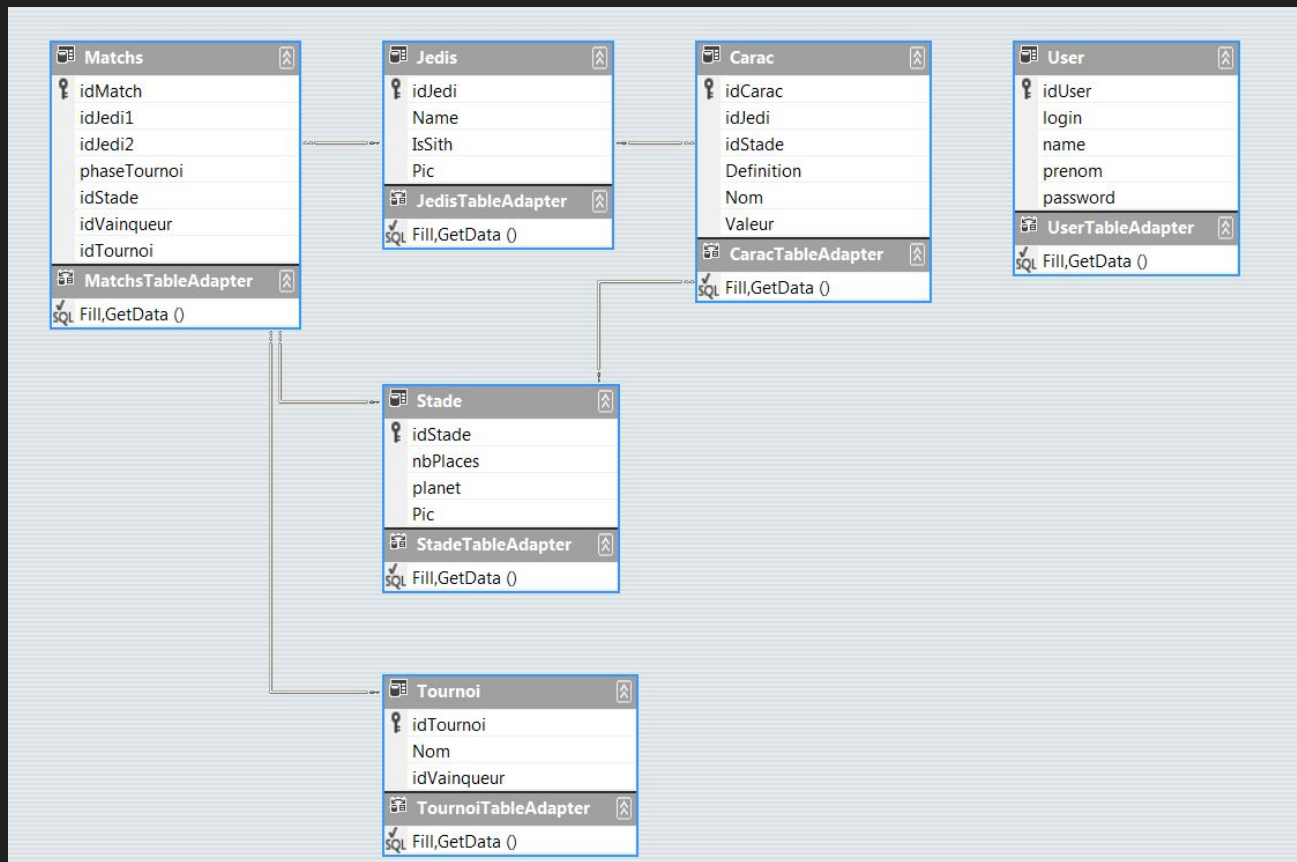
Oumaima : Implémentation des bases de données

Nicolas : Implémentation de la gestion des Jedis et des Stades + Test unitaires

Couche données

BDD

Tests unitaire



Stub

```
public List<Stade> getAllStades()
{
    List<Stade> listStades = new List<Stade>();

    // Caractéristiques éventuelles des stades
    List<Caracteristique> caracsTatooine = getAllCaracs().Where(s => s.Type == ETypeCaracteristique.Stade && s.ID == 13).ToList();
    List<Caracteristique> caracsHoth = getAllCaracs().Where(s => s.Type == ETypeCaracteristique.Stade && (s.ID == 12 || s.ID == 13)).ToList();
    List<Caracteristique> caracsCoruscant = getAllCaracs().Where(s => s.Type == ETypeCaracteristique.Stade && s.ID == 14).ToList();

    // Crée la liste des stades
    listStades.Add(new Stade(1, 150, "Jakku", null));
    listStades.Add(new Stade(2, 2000, "Tatooine", caracsTatooine));
    listStades.Add(new Stade(3, 10000, "Hoth", caracsHoth));
    listStades.Add(new Stade(4, 110000, "Coruscant", caracsCoruscant));

    return listStades;
}
```

Application console

Menu - Jedi Tournament Console

- 1 - Afficher la liste des stades.
- 2 - Afficher la liste des matchs.
- 3 - Afficher la liste des matchs (Sith Vs Sith et Stade >200 places).
- 4 - Afficher la liste des siths.
- 5 - Afficher la liste des jedis avec plus de 3 de force et 50 points de vie.
- Q - Quitter.

Votre choix :

Liste des stades :

Stade de Jakku.
Stade de Tatooine.
Stade de Hoth.
Stade de Coruscant.

Application WPF - Login

Démarrage de l'application :

Identification puis arrivée sur la fenêtre principale



Application WPF - Gestions

UserControl :

-> JediWrite

-> CaracteristiquesWrite

-> CaracteristiqueWrite

MVVM

Opérationnel mais relié au stubData



Application WPF - Création de tournoi

UserControl :

-> MatchWrite

-> Jedi / Match en Read

MVVM

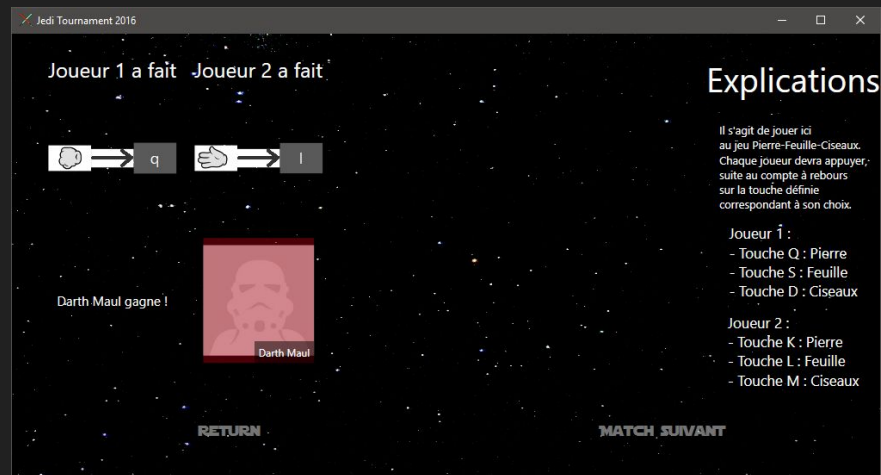
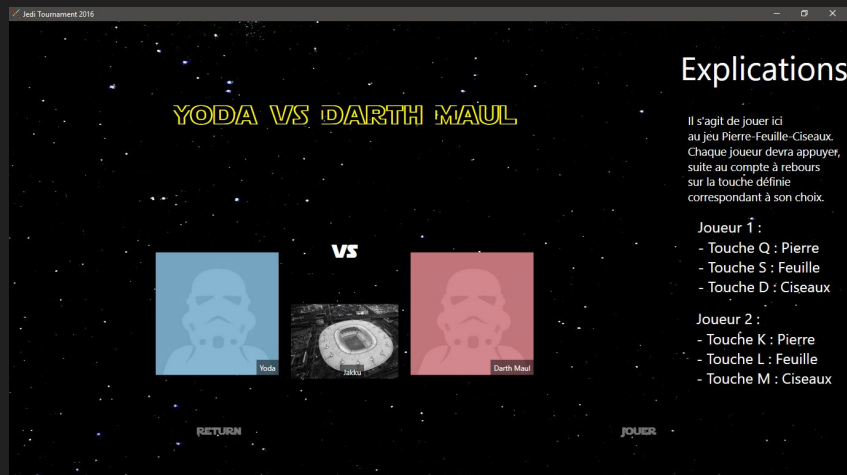
Matches générés/modifiable

Mode jeu Manuel/Automatique



Fonctionnement du jeu

Deux modes : automatique ou manuel



Conclusion

Résultat :

- Gestion Jedis, Stades
- Création des tournois et des matchs
- Mécanisme de jeu

Évolution souhaitées :

- Finir un historique pour afficher le résultat des tournois
- Utiliser la base de données à la place du Stub