ISIMA – ZZ2

2 Mars 2018

Présentation du projet Services Web

Game Of Thrones

Encadré par:

- Anthony Brugère
- Raphael Gonçalves

Préparé par:

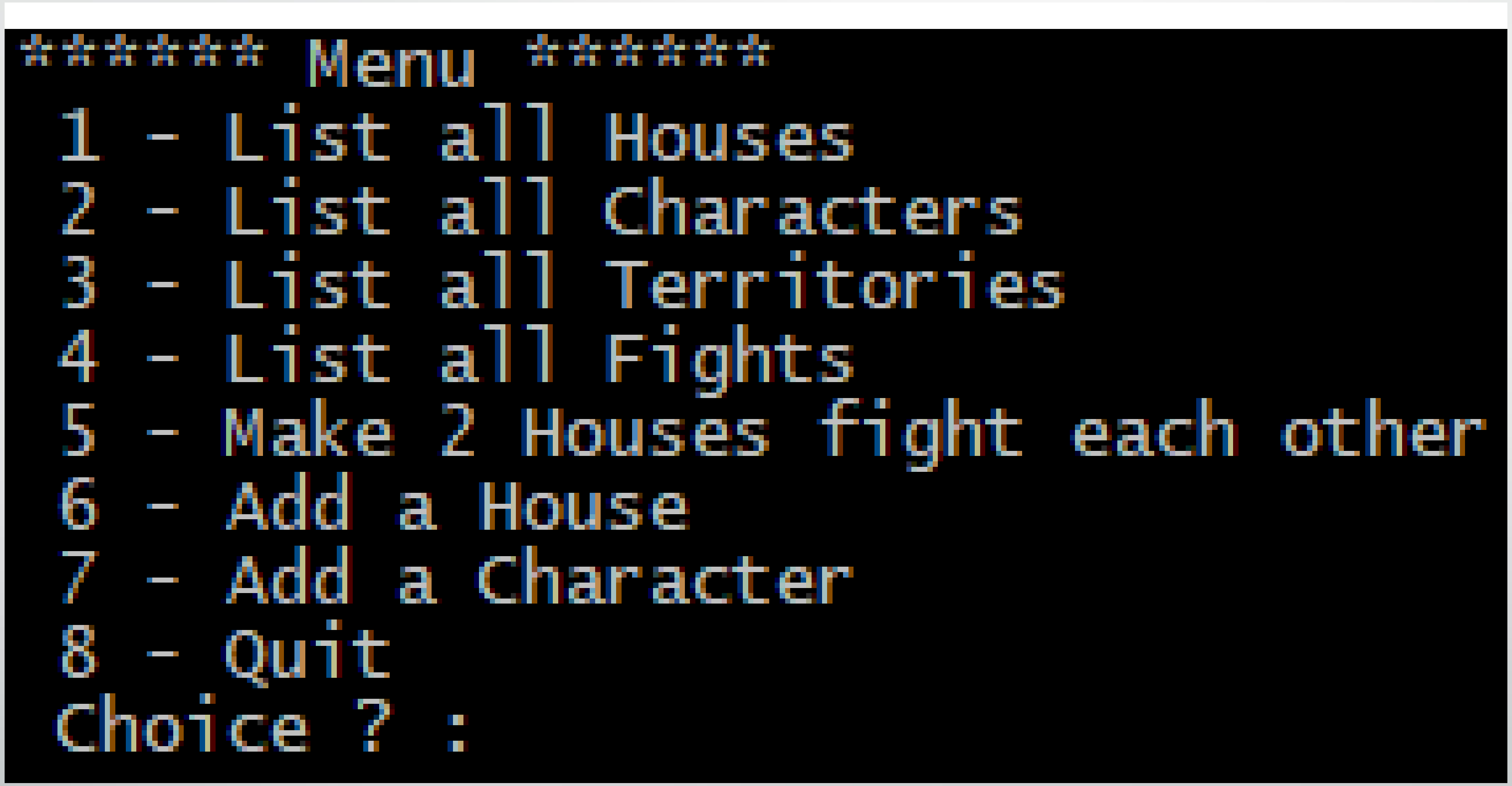
- Ravel Adamony
- Youssef Nidabrahim
- Hamza Bouhanni
- Said El Farkh
- Anass El Mkoudi

Plan

- Application Console
- Data Access Layer
- Architecture du projet
- Couche Présentation
- Règles du jeu
- Gestion de projet
- Démo

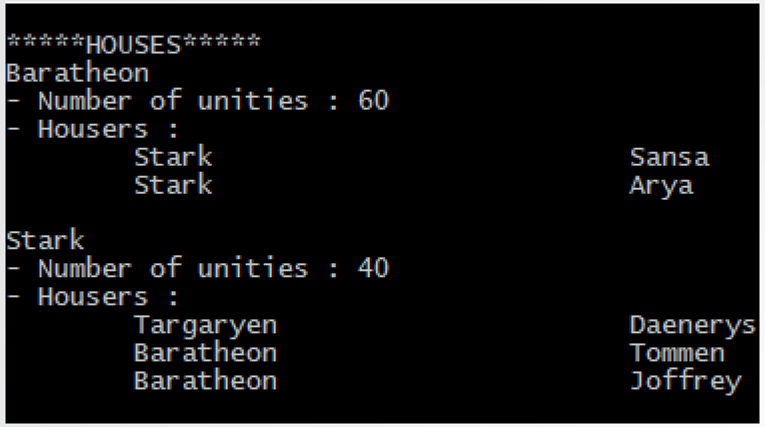
Application Console

Menu principal



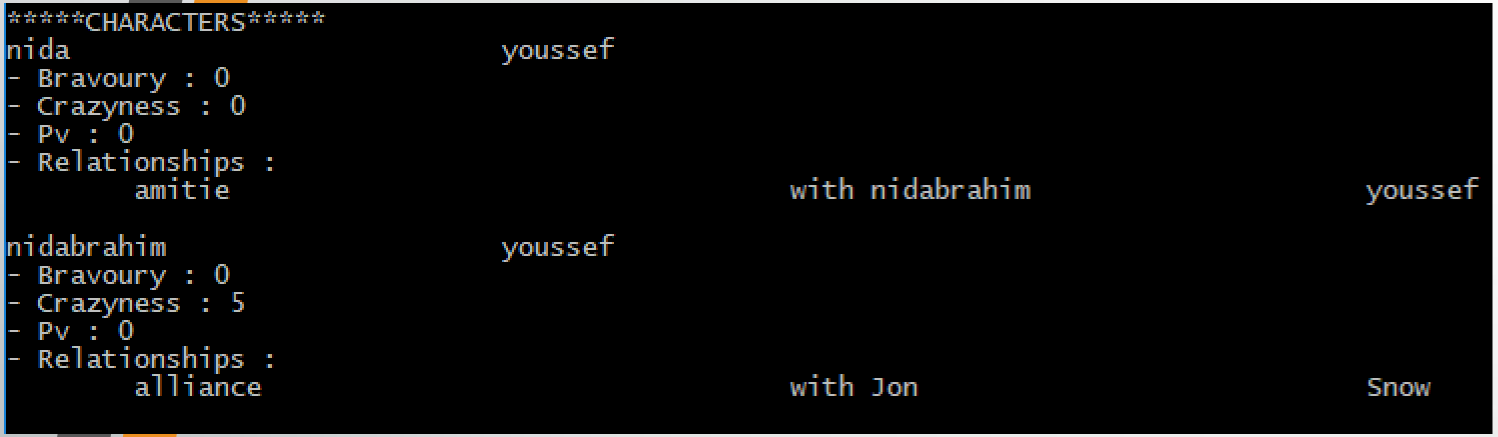
```
******* Menu *******
 1 - List all Houses
 2 - List all Characters
 3 - List all Territories
 4 - List all Fights
 5 - Make 2 Houses fight each other
 6 - Add a House
 7 - Add a Character
 8 - Quit
 Choice ? :
```

Listes



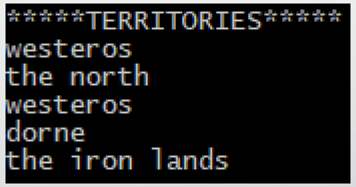
```
*****HOUSES*****
Baratheon
- Number of unities : 60
- Housers :
        Stark                          Sansa
        Stark                          Arya

Stark
- Number of unities : 40
- Housers :
        Targaryen                      Daenerys
        Baratheon                      Tommen
        Baratheon                      Joffrey
```

1.



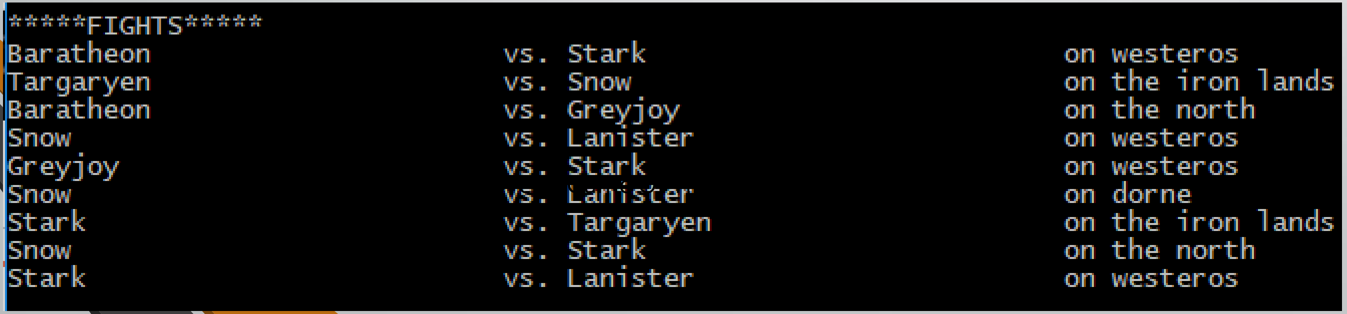
```
*****CHARACTERS*****
nida                           youssef
- Bravoury : 0
- Crazyness : 0
- Pv : 0
- Relationships :
      amitie                                    with nidabrahim                    youssef

nidabrahim                     youssef
- Bravoury : 0
- Crazyness : 5
- Pv : 0
- Relationships :
      alliance                                  with Jon                           Snow
```

2.



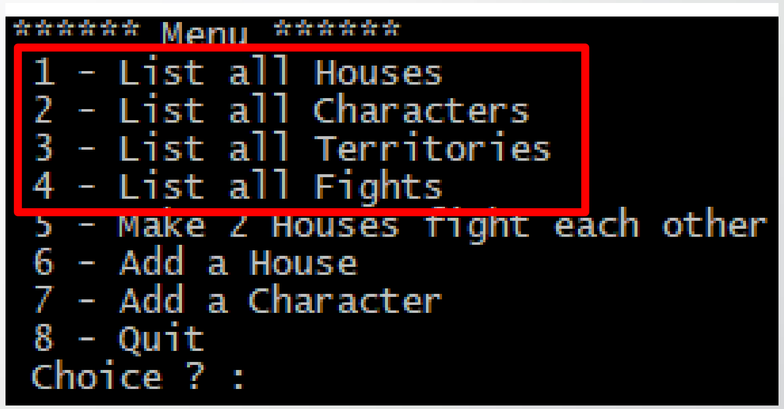
```
*****TERRITORIES*****
westeros
the north
westeros
dorne
the iron lands
```

3.



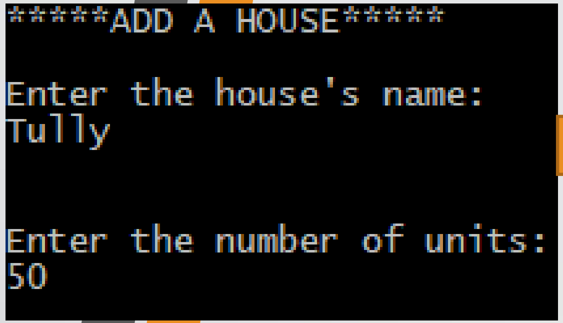
```
*****FIGHTS*****
Baratheon                      vs. Stark                          on westeros
Targaryen                      vs. Snow                           on the iron lands
Baratheon                      vs. Greyjoy                        on the north
Snow                           vs. Lanister                       on westeros
Greyjoy                        vs. Stark                          on westeros
Snow                           vs. Lanister                       on dorne
Stark                          vs. Targaryen                      on the iron lands
Snow                           vs. Stark                          on the north
Stark                          vs. Lanister                       on westeros
```

4.

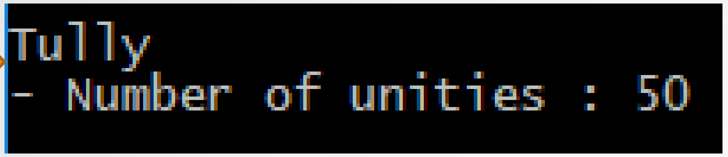


```
****** Menu ******
 1 - List all Houses
 2 - List all Characters
 3 - List all Territories
 4 - List all Fights
 5 - Make 2 Houses fight each other
 6 - Add a House
 7 - Add a Character
 8 - Quit
 Choice ? :
```

Ajouts

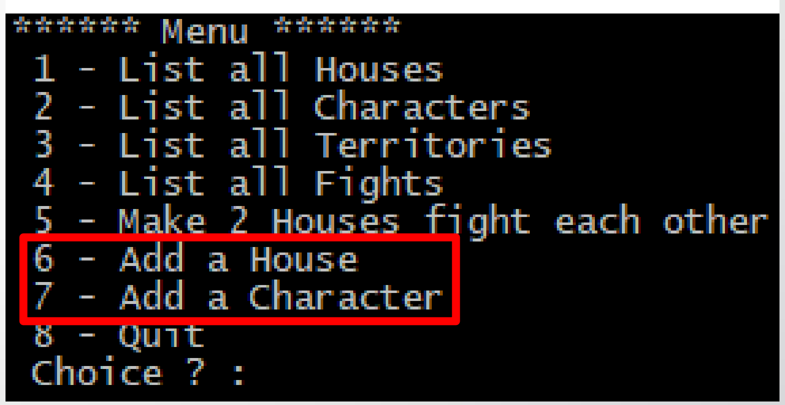


```
*****ADD A HOUSE*****

Enter the house's name:
Tully


Enter the number of units:
50
```

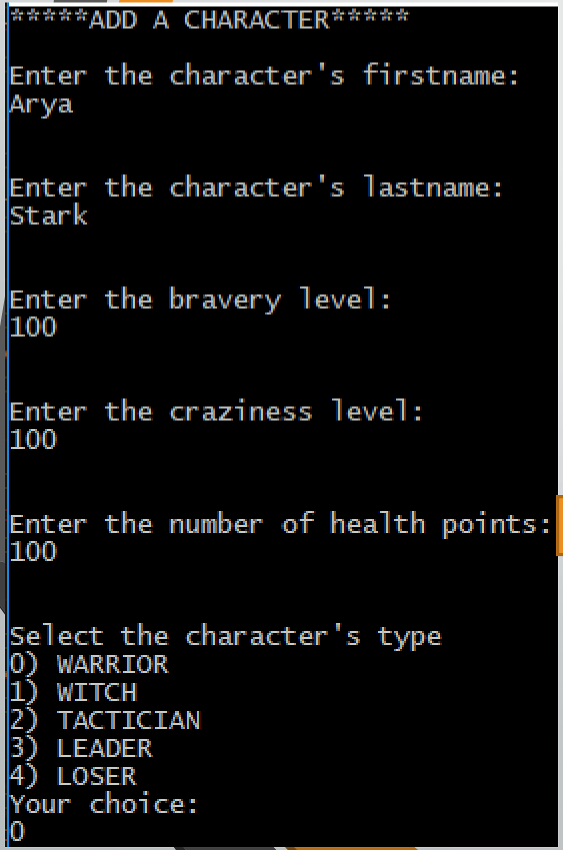


```
Tully
- Number of unities : 50
```

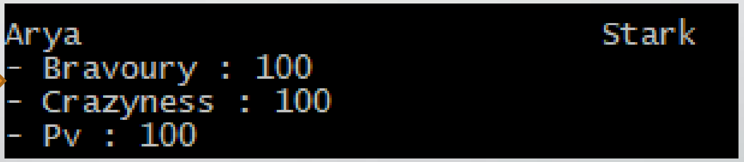
Ajout d'une maison



```
****** Menu ******
 1 - List all Houses
 2 - List all Characters
 3 - List all Territories
 4 - List all Fights
 5 - Make 2 Houses fight each other
 6 - Add a House
 7 - Add a Character
 8 - Quit
 Choice ? :
```



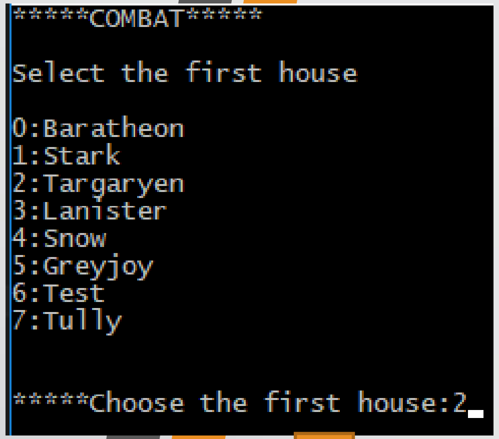
```
*****ADD A CHARACTER*****

Enter the character's firstname:
Arya


Enter the character's lastname:
Stark


Enter the bravery level:
100


Enter the craziness level:
100


Enter the number of health points:
100


Select the character's type
0) WARRIOR
1) WITCH
2) TACTICIAN
3) LEADER
4) LOSER
Your choice:
0
```



```
Arya                            Stark
- Bravoury : 100
- Crazyness : 100
- Pv : 100
```

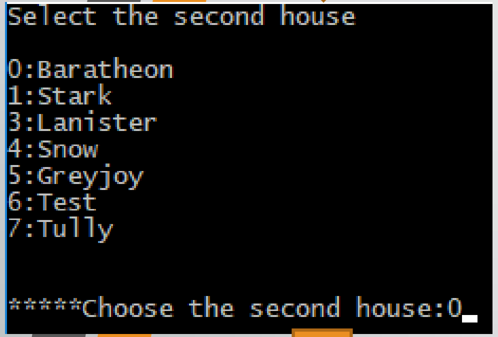
Ajout d'un personnage

Combat



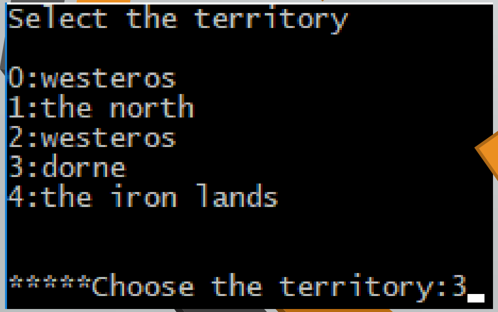
```
*****COMBAT*****

Select the first house

0:Baratheon
1:Stark
2:Targaryen
3:Lanister
4:Snow
5:Greyjoy
6:Test
7:Tully


*****Choose the first house:2_
```

Maison 1



```
Select the second house

0:Baratheon
1:Stark
3:Lanister
4:Snow
5:Greyjoy
6:Test
7:Tully


*****Choose the second house:0_
```

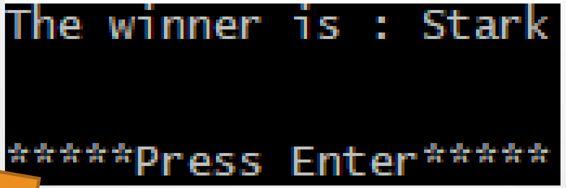
Maison 2



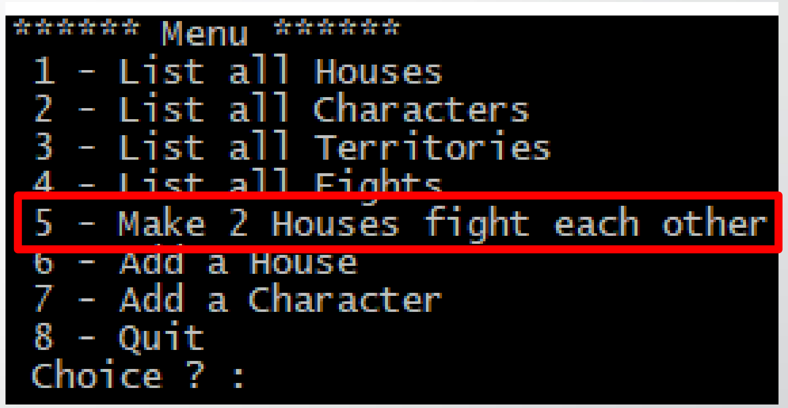
```
Select the territory

0:westeros
1:the north
2:westeros
3:dorne
4:the iron lands


*****Choose the territory:3_
```

Territoire

Vainqueur



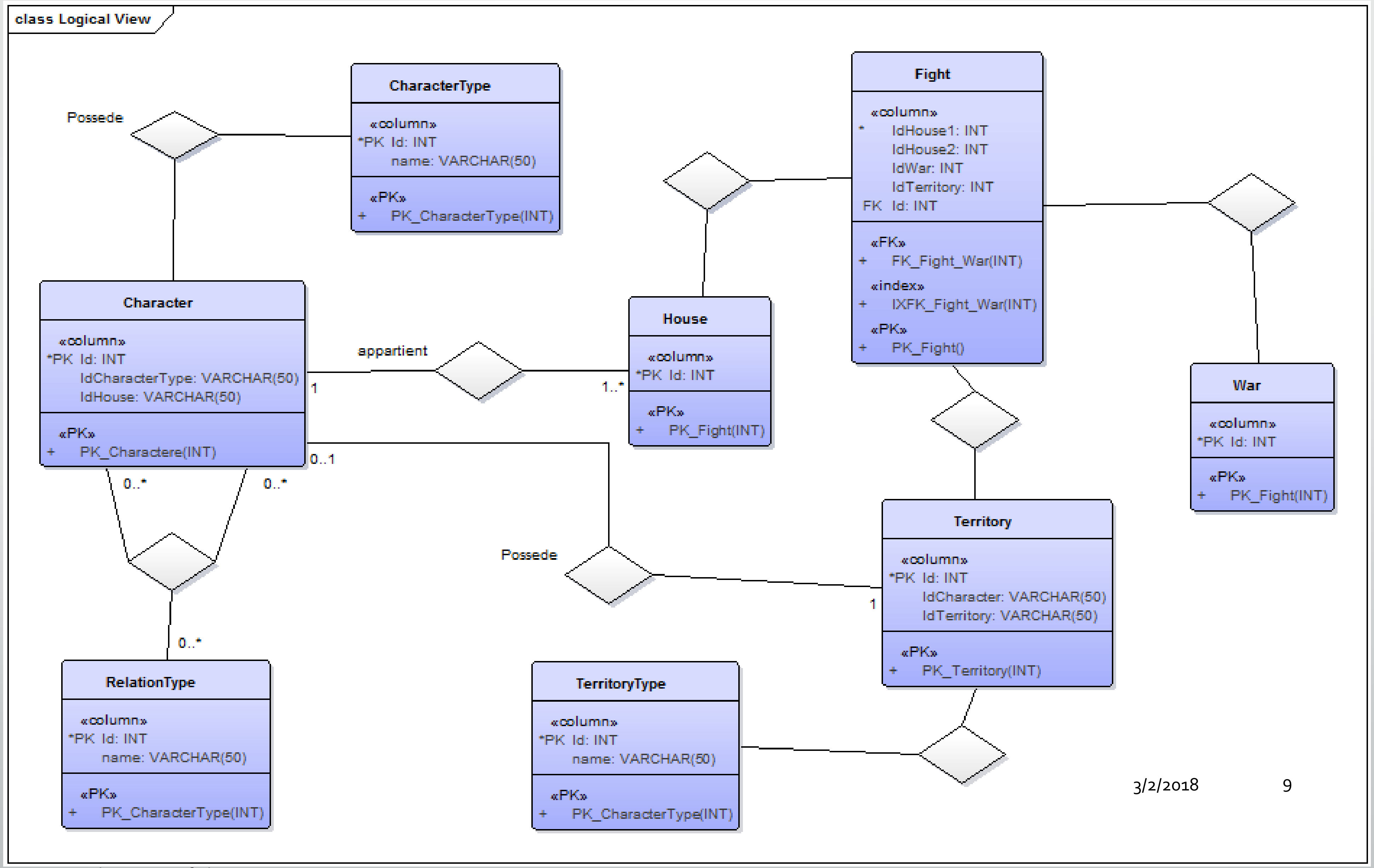
```
The winner is : Stark


*****Press Enter*****
```

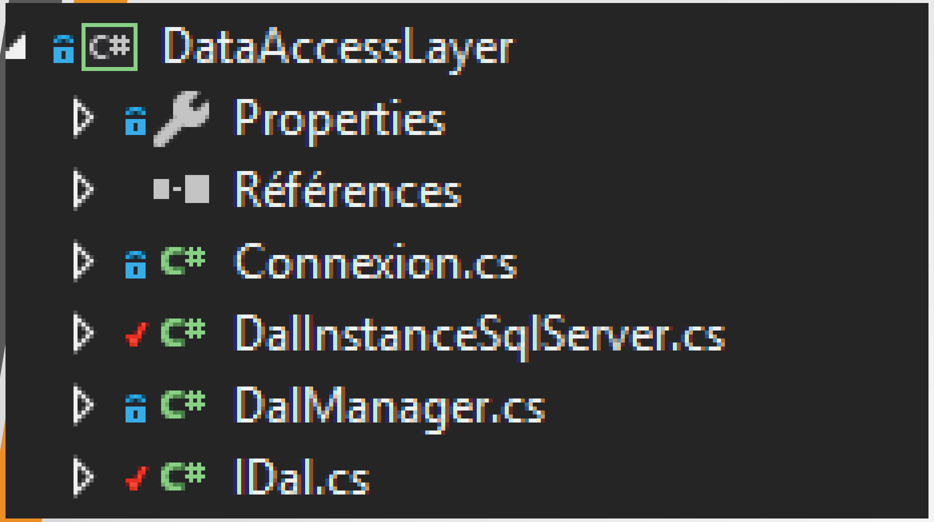


```
****** Menu ******
 1 - List all Houses
 2 - List all Characters
 3 - List all Territories
 4 - List all Fights
 5 - Make 2 Houses fight each other
 6 - Add a House
 7 - Add a Character
 8 - Quit
 Choice ? :
```

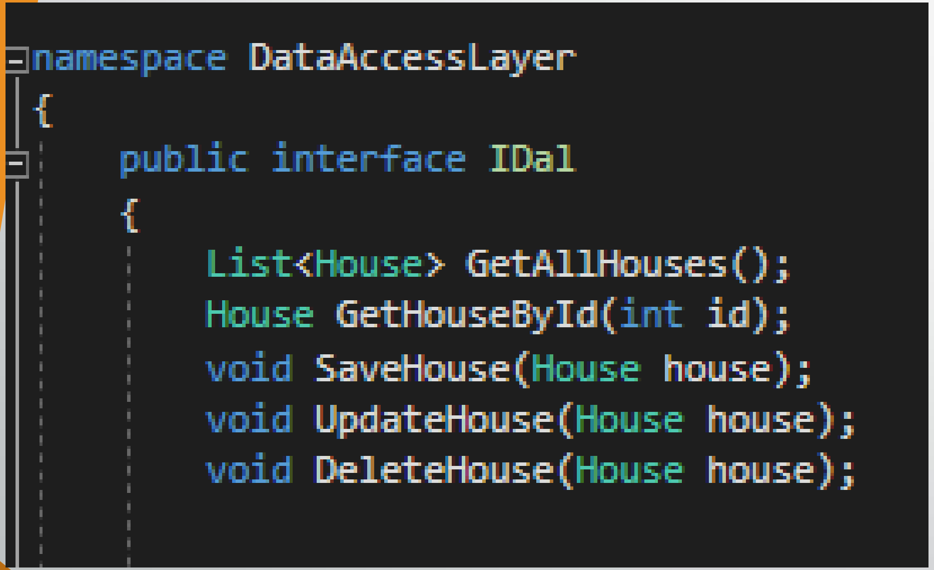
Data Access Layer

Schéma Relationnel

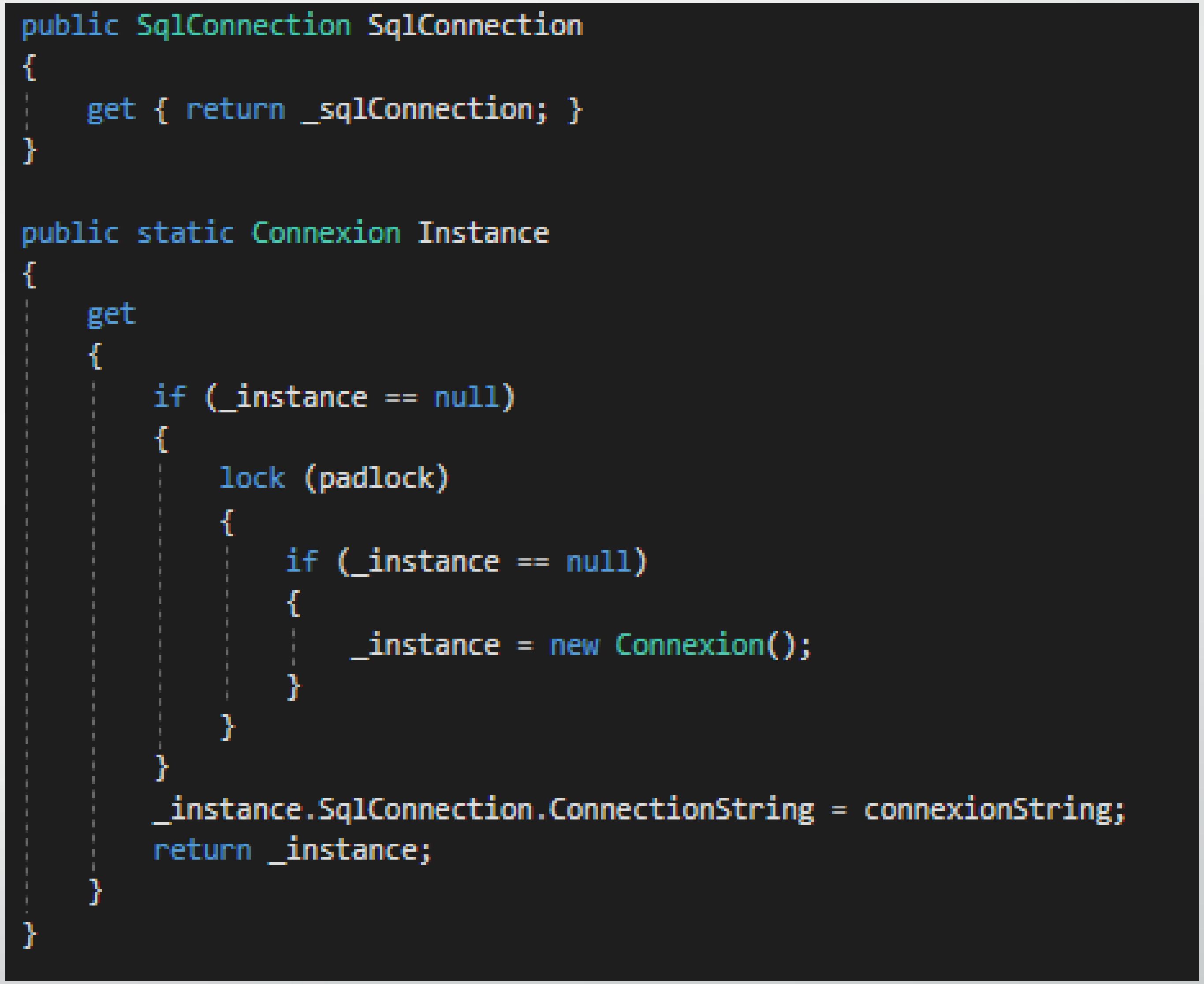




```
DataAccessLayer
    Properties
    Références
    Connexion.cs
    DalInstanceSqlServer.cs
    DalManager.cs
    IDal.cs
```

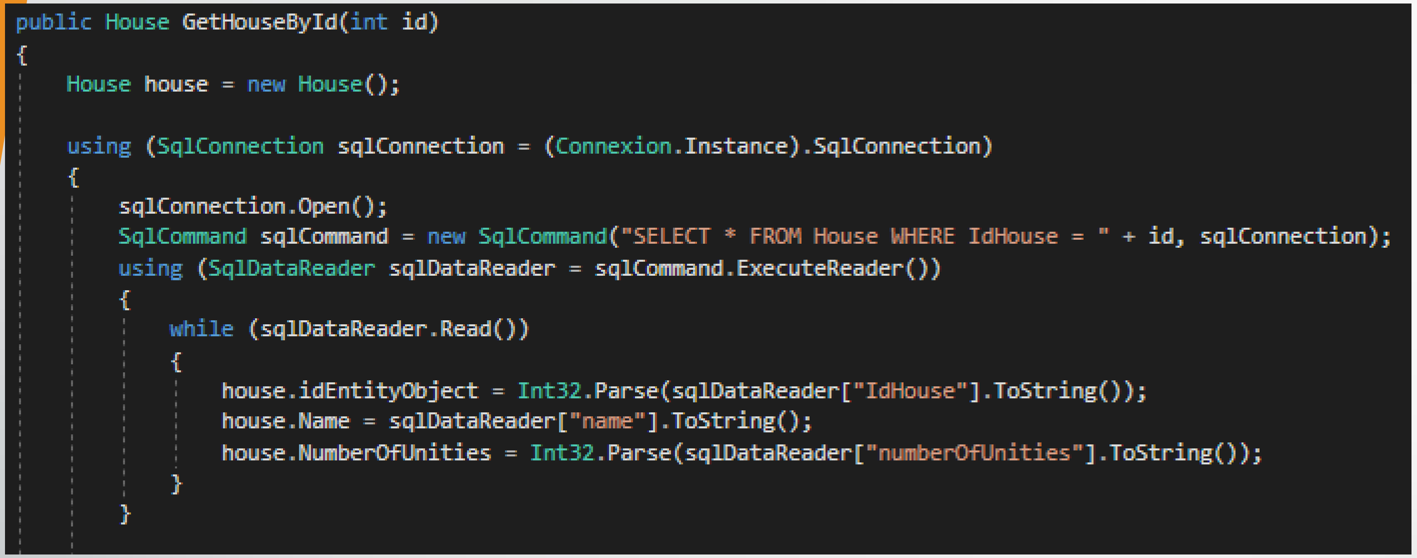


```
namespace DataAccessLayer
{
    public interface IDal
    {
        List<House> GetAllHouses();
        House GetHouseById(int id);
        void SaveHouse(House house);
        void UpdateHouse(House house);
        void DeleteHouse(House house);
```

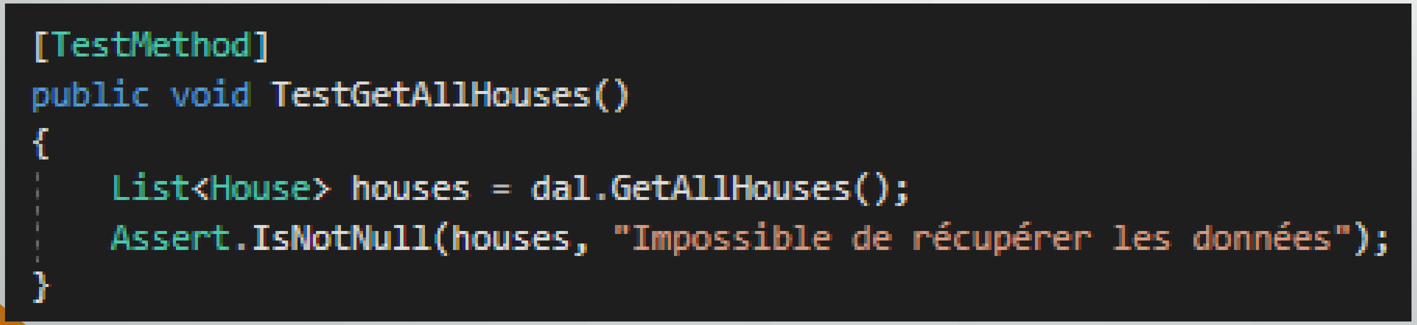


```
public SqlConnection SqlConnection
{
    get { return _sqlConnection; }
}

public static Connexion Instance
{
    get
    {
        if (_instance == null)
        {
            lock (padlock)
            {
                if (_instance == null)
                {
                    _instance = new Connexion();
                }
            }
        }
        _instance.SqlConnection.ConnectionString = connexionString;
        return _instance;
    }
}
```

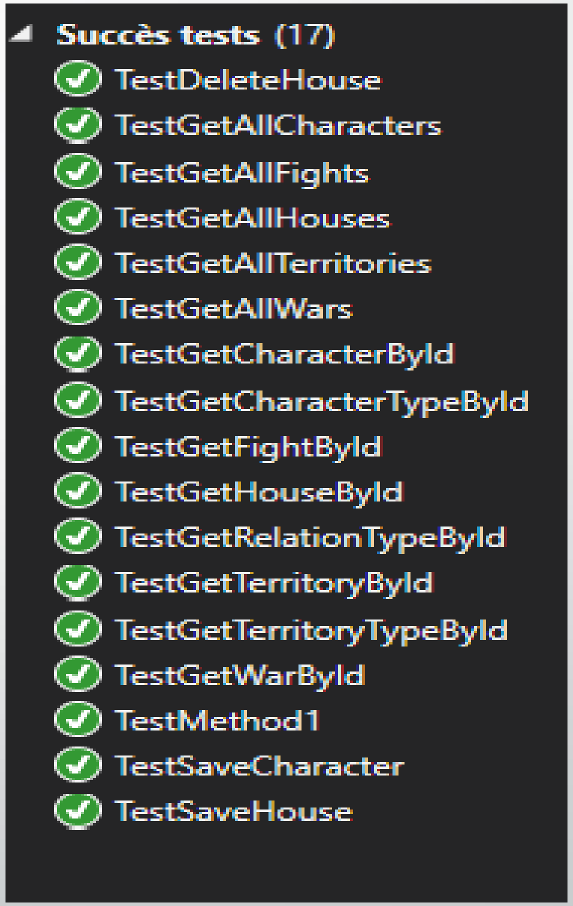
Tests



```
public House GetHouseById(int id)
{
    House house = new House();

    using (SqlConnection sqlConnection = (Connexion.Instance).SqlConnection)
    {
        sqlConnection.Open();
        SqlCommand sqlCommand = new SqlCommand("SELECT * FROM House WHERE IdHouse = " + id, sqlConnection);
        using (SqlDataReader sqlDataReader = sqlCommand.ExecuteReader())
        {
            while (sqlDataReader.Read())
            {
                house.idEntityObject = Int32.Parse(sqlDataReader["IdHouse"].ToString());
                house.Name = sqlDataReader["name"].ToString();
                house.NumberOfUnities = Int32.Parse(sqlDataReader["numberOfUnities"].ToString());
            }
        }
```

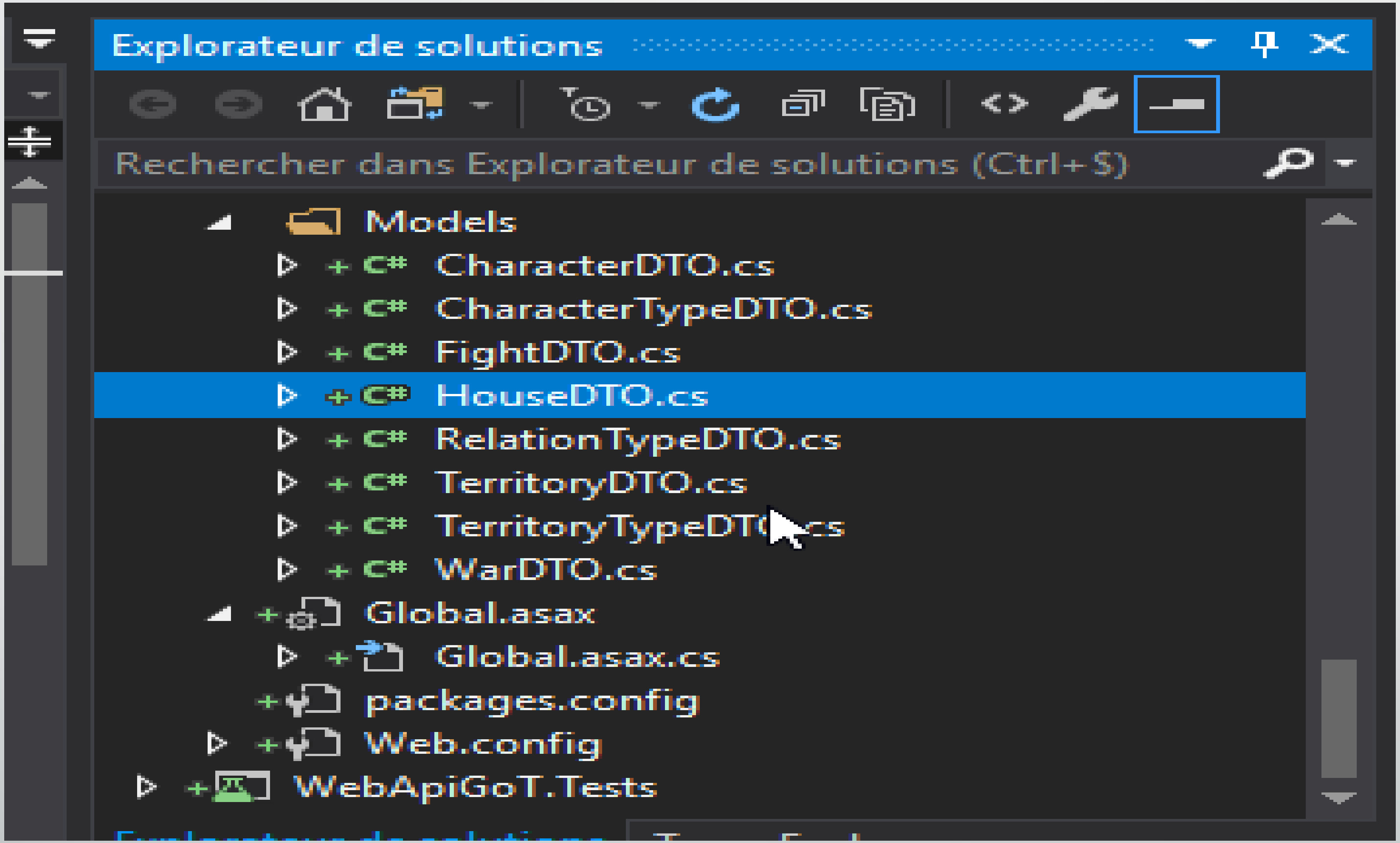


```
[TestMethod]
public void TestGetAllHouses()
{
    List<House> houses = dal.GetAllHouses();
    Assert.IsNotNull(houses, "Impossible de récupérer les données");
}
```



```
◢ Succès tests (17)
    ✓ TestDeleteHouse
    ✓ TestGetAllCharacters
    ✓ TestGetAllFights
    ✓ TestGetAllHouses
    ✓ TestGetAllTerritories
    ✓ TestGetAllWars
    ✓ TestGetCharacterById
    ✓ TestGetCharacterTypeById
    ✓ TestGetFightById
    ✓ TestGetHouseById
    ✓ TestGetRelationTypeById
    ✓ TestGetTerritoryById
    ✓ TestGetTerritoryTypeById
    ✓ TestGetWarById
    ✓ TestMethod1
    ✓ TestSaveCharacter
    ✓ TestSaveHouse
```

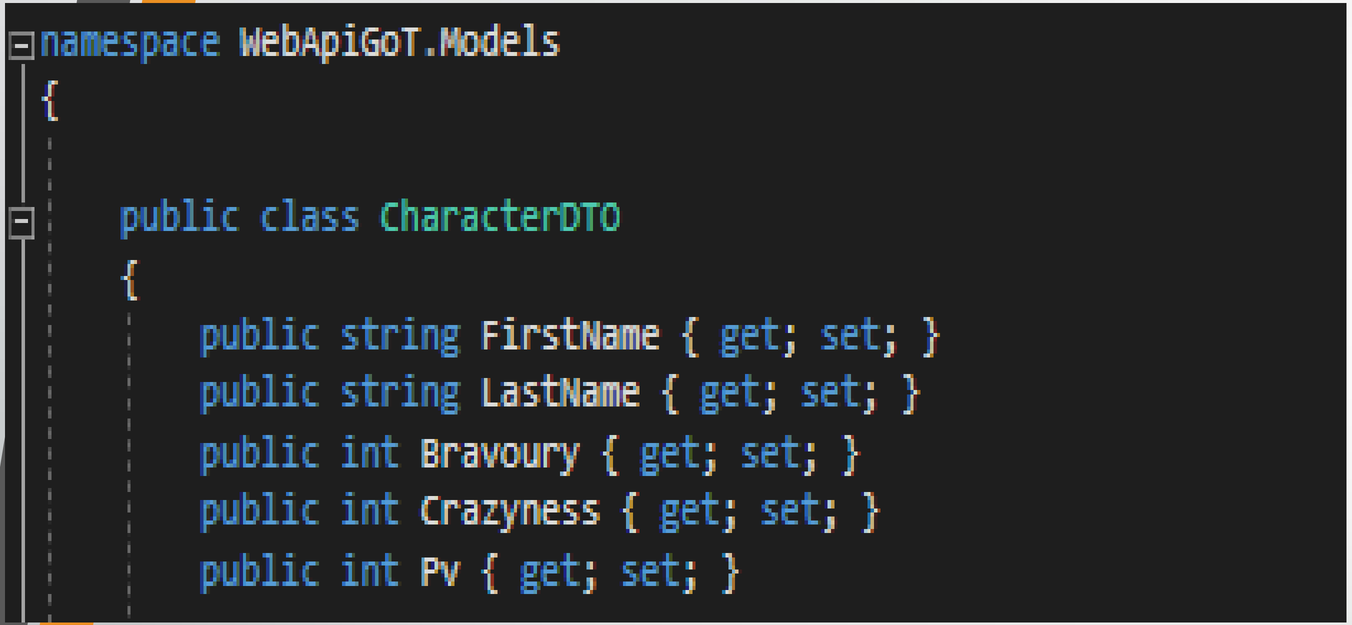
Architecture du projet



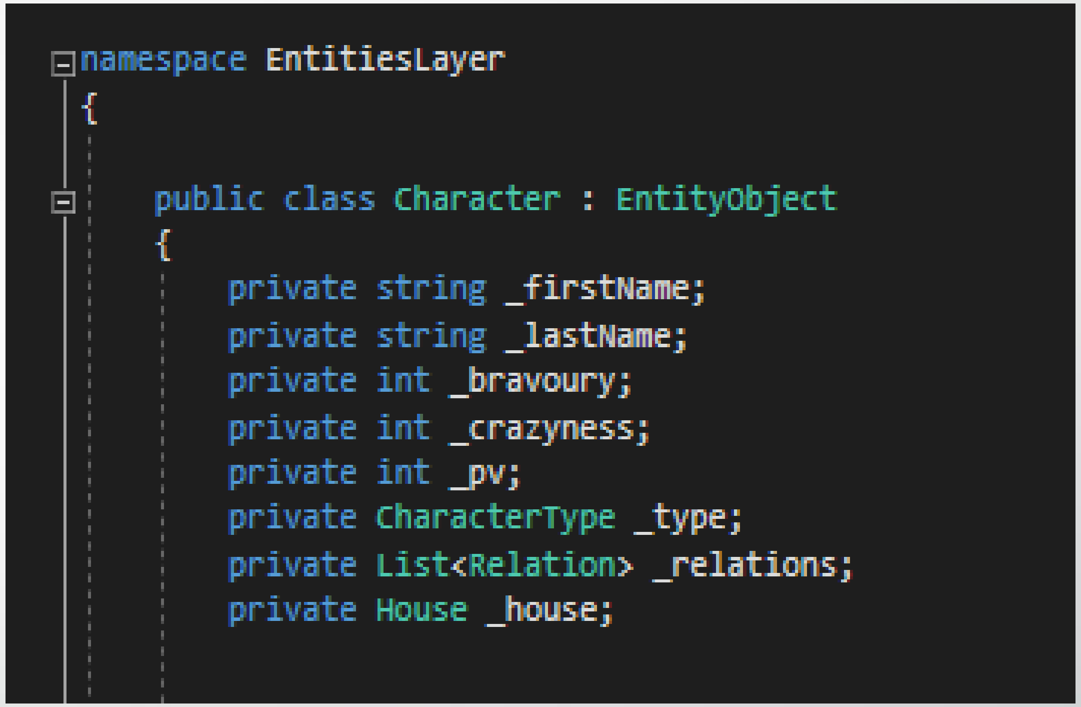
Explorateur de solutions

Rechercher dans Explorateur de solutions (Ctrl+$)

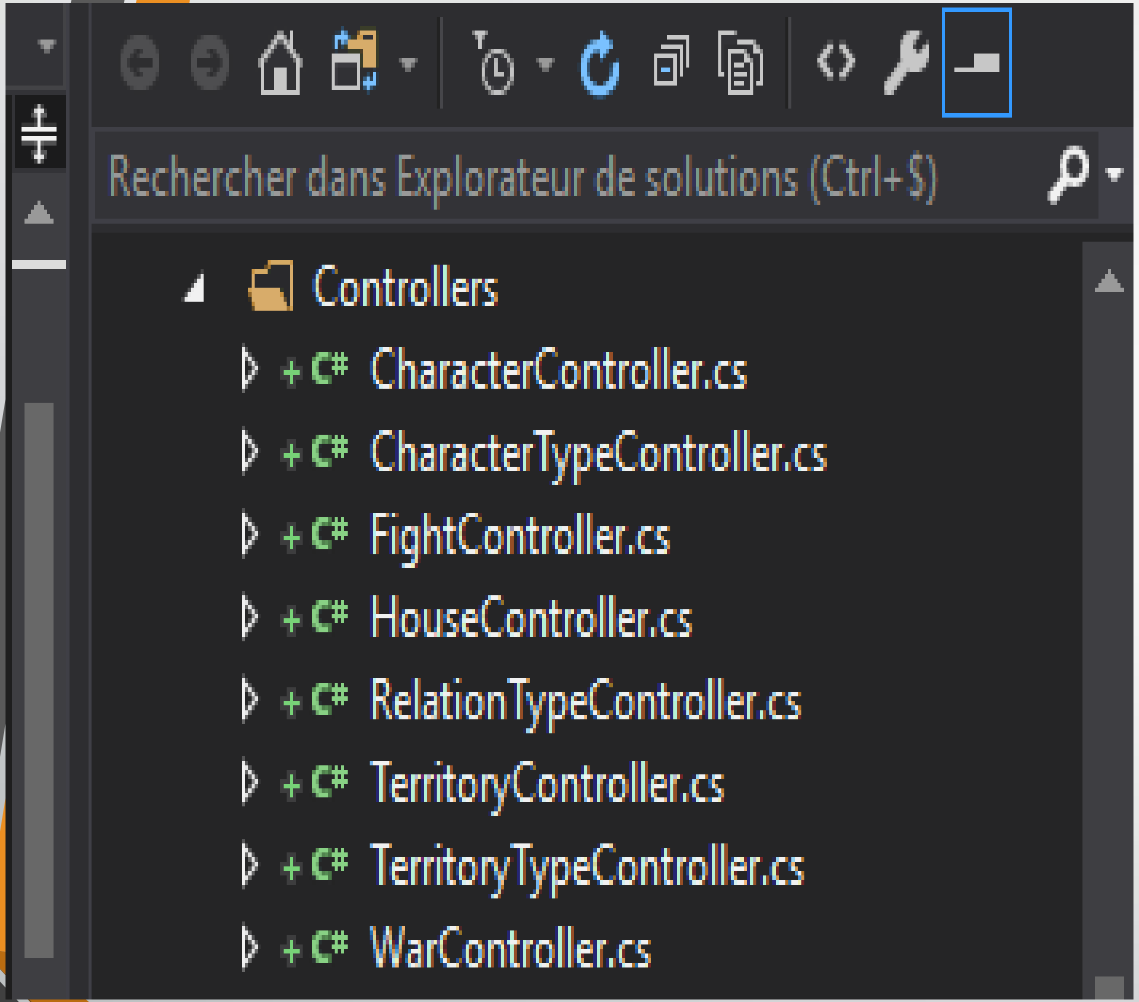
```
◢ Models
  ▷ + C# CharacterDTO.cs
  ▷ + C# CharacterTypeDTO.cs
  ▷ + C# FightDTO.cs
  ▷ + C# HouseDTO.cs
  ▷ + C# RelationTypeDTO.cs
  ▷ + C# TerritoryDTO.cs
  ▷ + C# TerritoryTypeDTO.cs
  ▷ + C# WarDTO.cs
◢ + Global.asax
  ▷ + Global.asax.cs
+ packages.config
▷ + Web.config
▷ + WebApiGoT.Tests
```



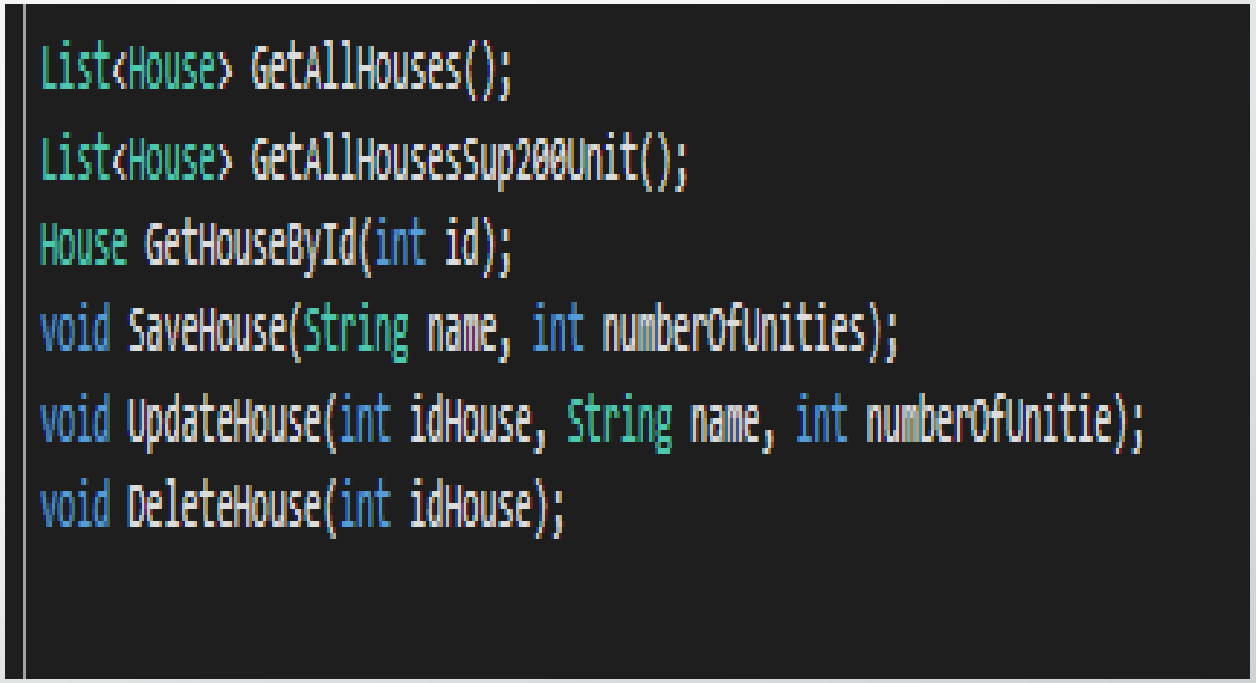
```
namespace WebApiGoT.Models
{

    public class CharacterDTO
    {
        public string FirstName { get; set; }
        public string LastName { get; set; }
        public int Bravoury { get; set; }
        public int Crazyness { get; set; }
        public int Pv { get; set; }
```



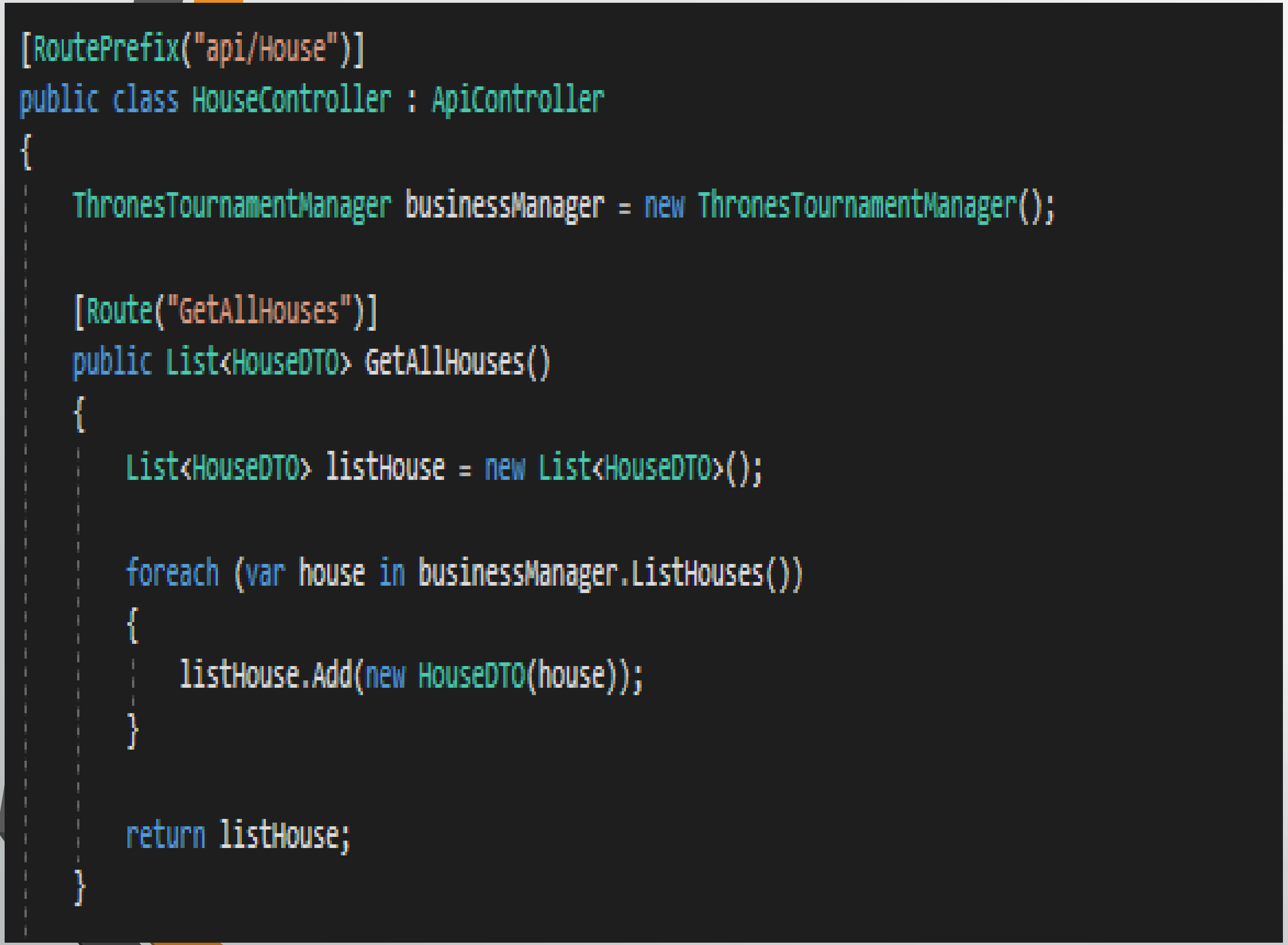
```
namespace EntitiesLayer
{

    public class Character : EntityObject
    {
        private string _firstName;
        private string _lastName;
        private int _bravoury;
        private int _crazyness;
        private int _pv;
        private CharacterType _type;
        private List<Relation> _relations;
        private House _house;
```


Rechercher dans Explorateur de solutions (Ctrl+$)

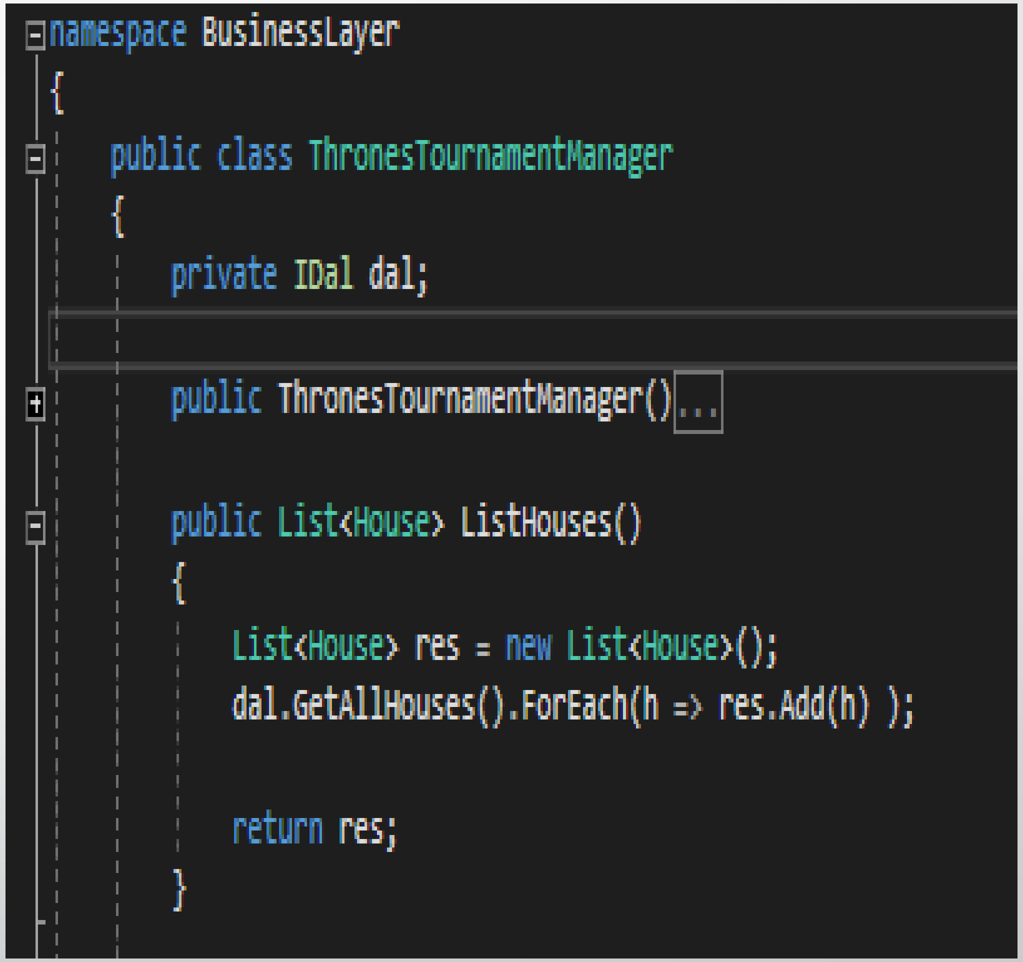
- Controllers
  - CharacterController.cs
  - CharacterTypeController.cs
  - FightController.cs
  - HouseController.cs
  - RelationTypeController.cs
  - TerritoryController.cs
  - TerritoryTypeController.cs
  - WarController.cs



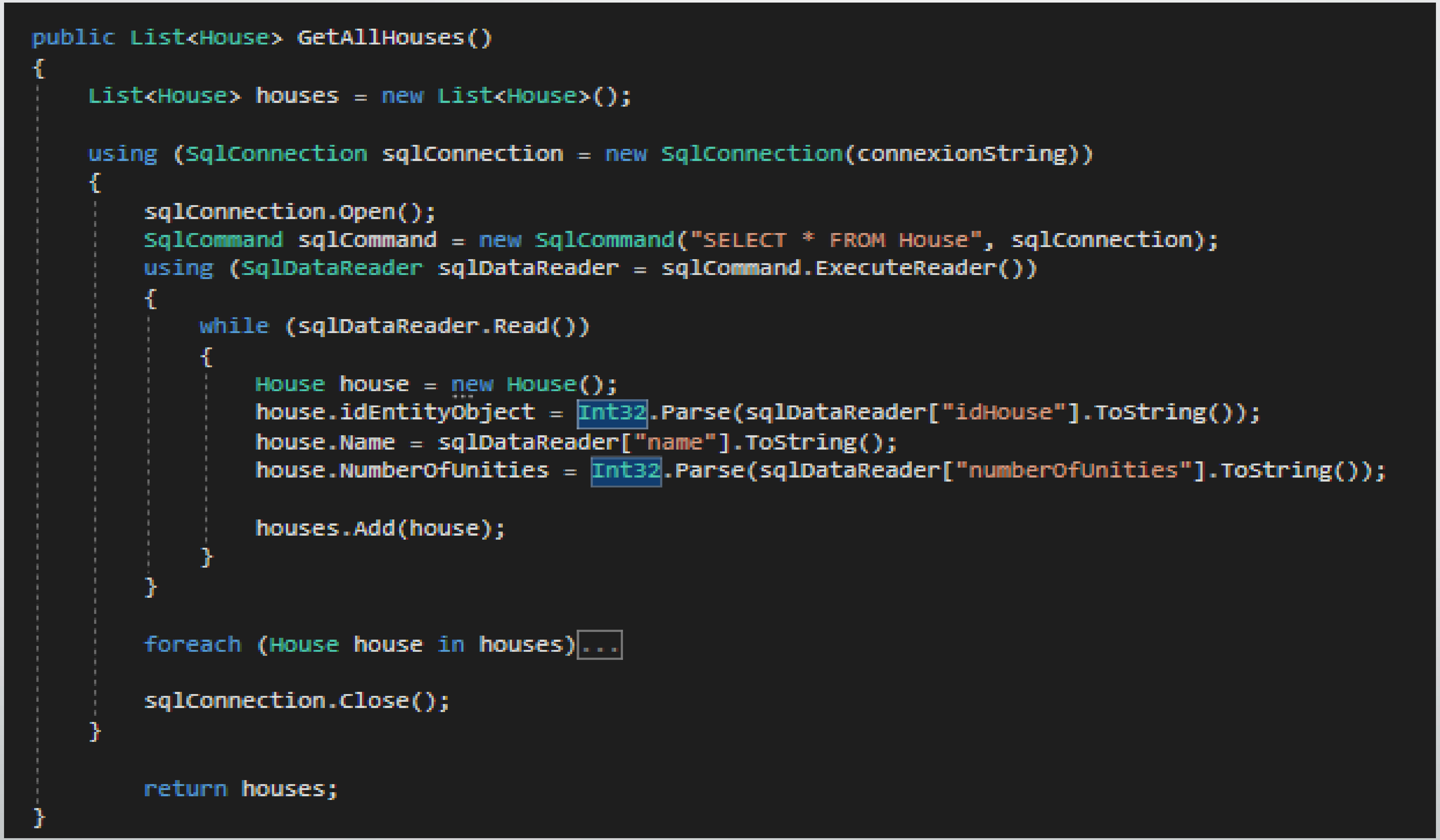
```
List<House> GetAllHouses();
List<House> GetAllHousesSup200Unit();
House GetHouseById(int id);
void SaveHouse(String name, int numberOfUnities);
void UpdateHouse(int idHouse, String name, int numberOfUnitie);
void DeleteHouse(int idHouse);
```



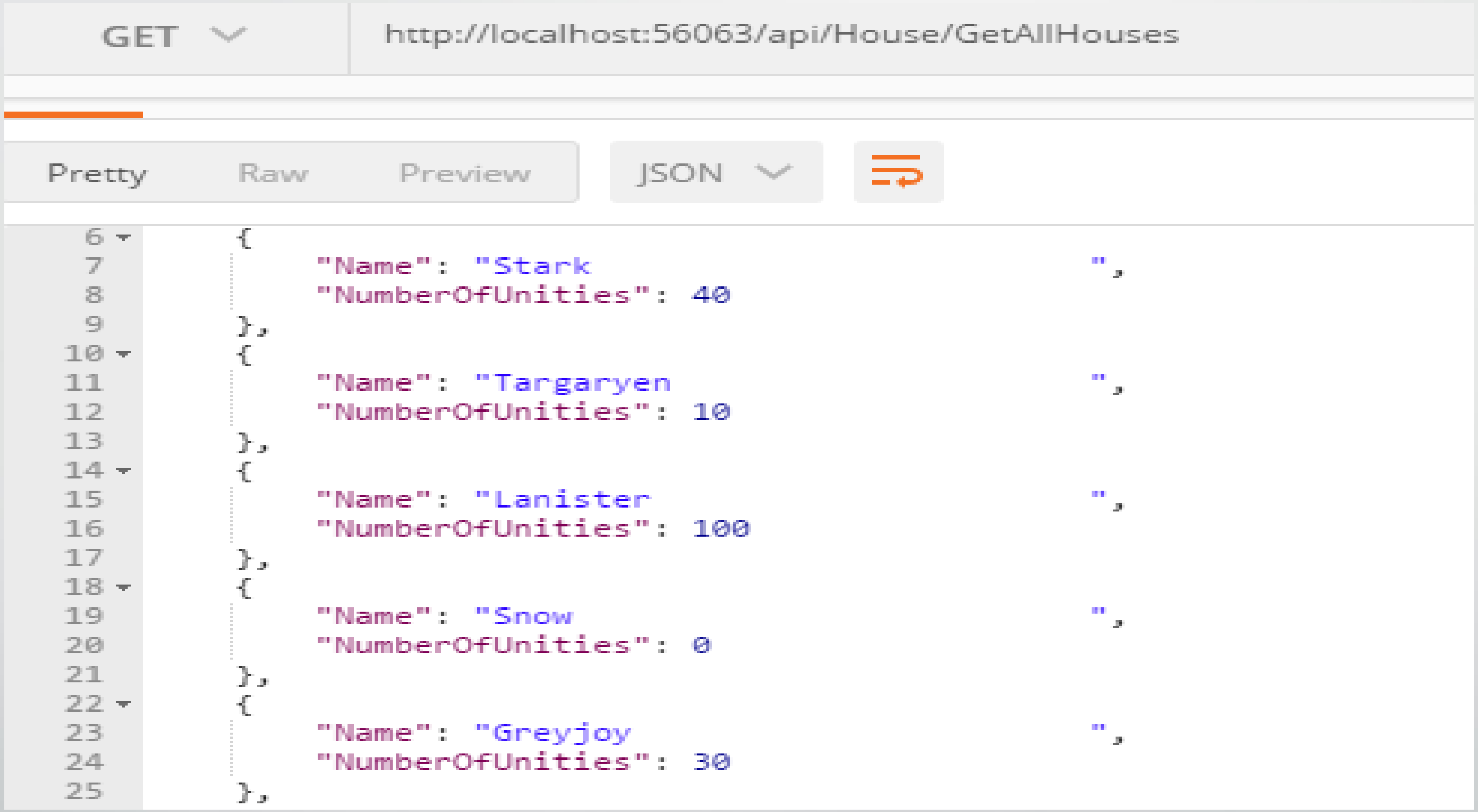
```
[RoutePrefix("api/House")]
public class HouseController : ApiController
{
    ThronesTournamentManager businessManager = new ThronesTournamentManager();

    [Route("GetAllHouses")]
    public List<HouseDTO> GetAllHouses()
    {
        List<HouseDTO> listHouse = new List<HouseDTO>();

        foreach (var house in businessManager.ListHouses())
        {
            listHouse.Add(new HouseDTO(house));
        }

        return listHouse;
    }
```



```
namespace BusinessLayer
{
    public class ThronesTournamentManager
    {
        private IDal dal;

        public ThronesTournamentManager()...

        public List<House> ListHouses()
        {
            List<House> res = new List<House>();
            dal.GetAllHouses().ForEach(h => res.Add(h) );

            return res;
        }
```

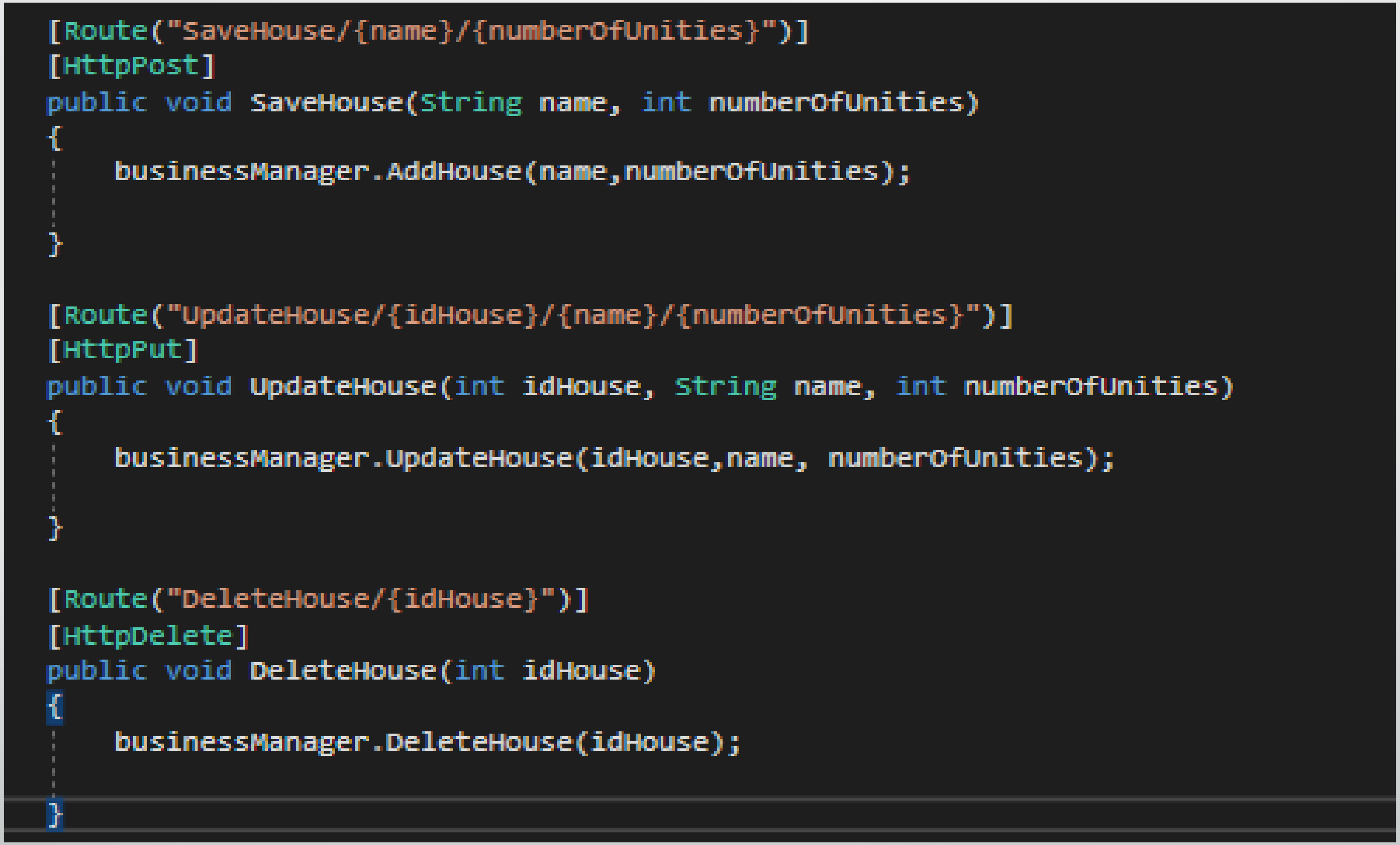


```
public List<House> GetAllHouses()
{
    List<House> houses = new List<House>();

    using (SqlConnection sqlConnection = new SqlConnection(connexionString))
    {
        sqlConnection.Open();
        SqlCommand sqlCommand = new SqlCommand("SELECT * FROM House", sqlConnection);
        using (SqlDataReader sqlDataReader = sqlCommand.ExecuteReader())
        {
            while (sqlDataReader.Read())
            {
                House house = new House();
                house.idEntityObject = Int32.Parse(sqlDataReader["idHouse"].ToString());
                house.Name = sqlDataReader["name"].ToString();
                house.NumberOfUnities = Int32.Parse(sqlDataReader["numberOfUnities"].ToString());

                houses.Add(house);
            }
        }

        foreach (House house in houses)...

        sqlConnection.Close();
    }

        return houses;
}
```

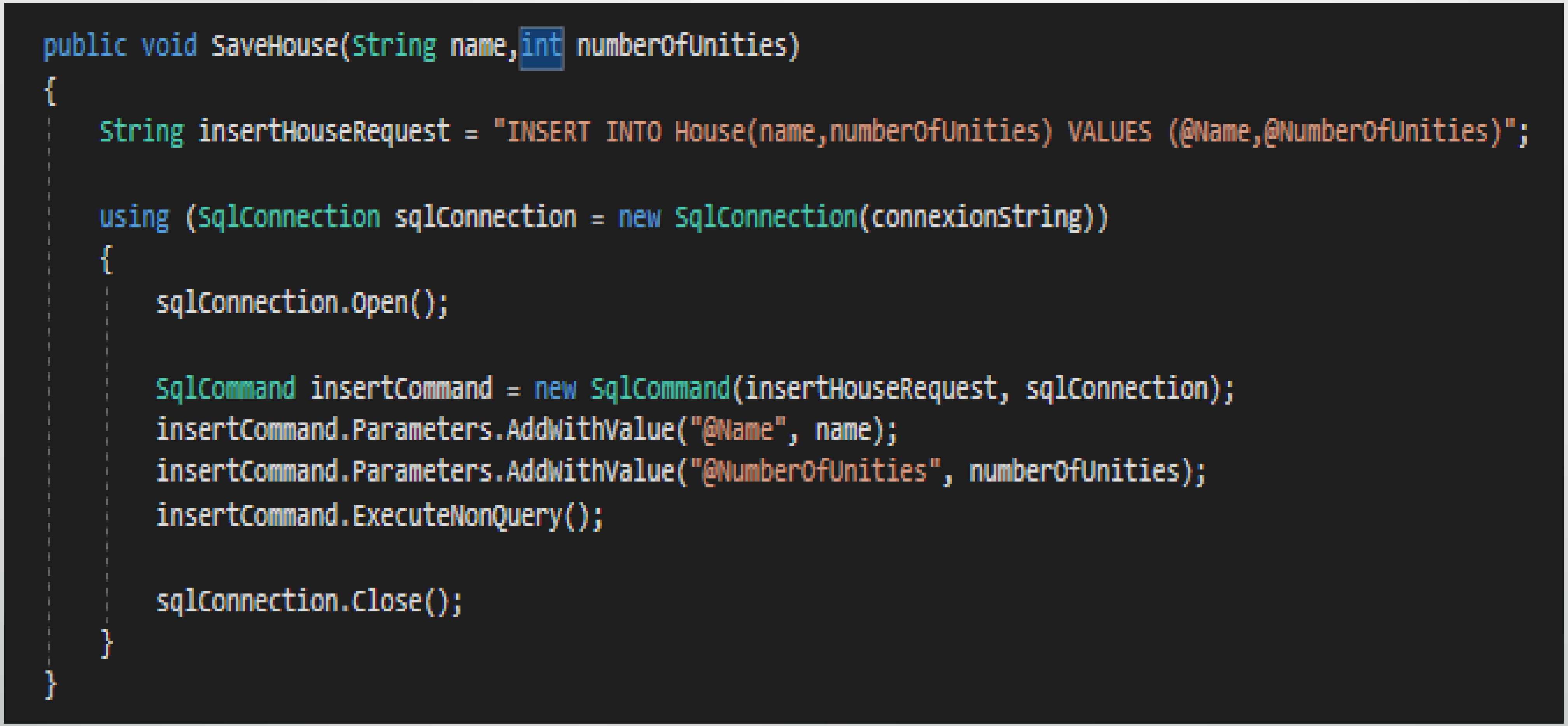


http://localhost:56063/api/House/GetAllHouses

```
    {
        "Name": "Stark                                        ",
        "NumberOfUnities": 40
    },
    {
        "Name": "Targaryen                                    ",
        "NumberOfUnities": 10
    },
    {
        "Name": "Lanister                                     ",
        "NumberOfUnities": 100
    },
    {
        "Name": "Snow                                         ",
        "NumberOfUnities": 0
    },
    {
        "Name": "Greyjoy                                      ",
        "NumberOfUnities": 30
    },
```

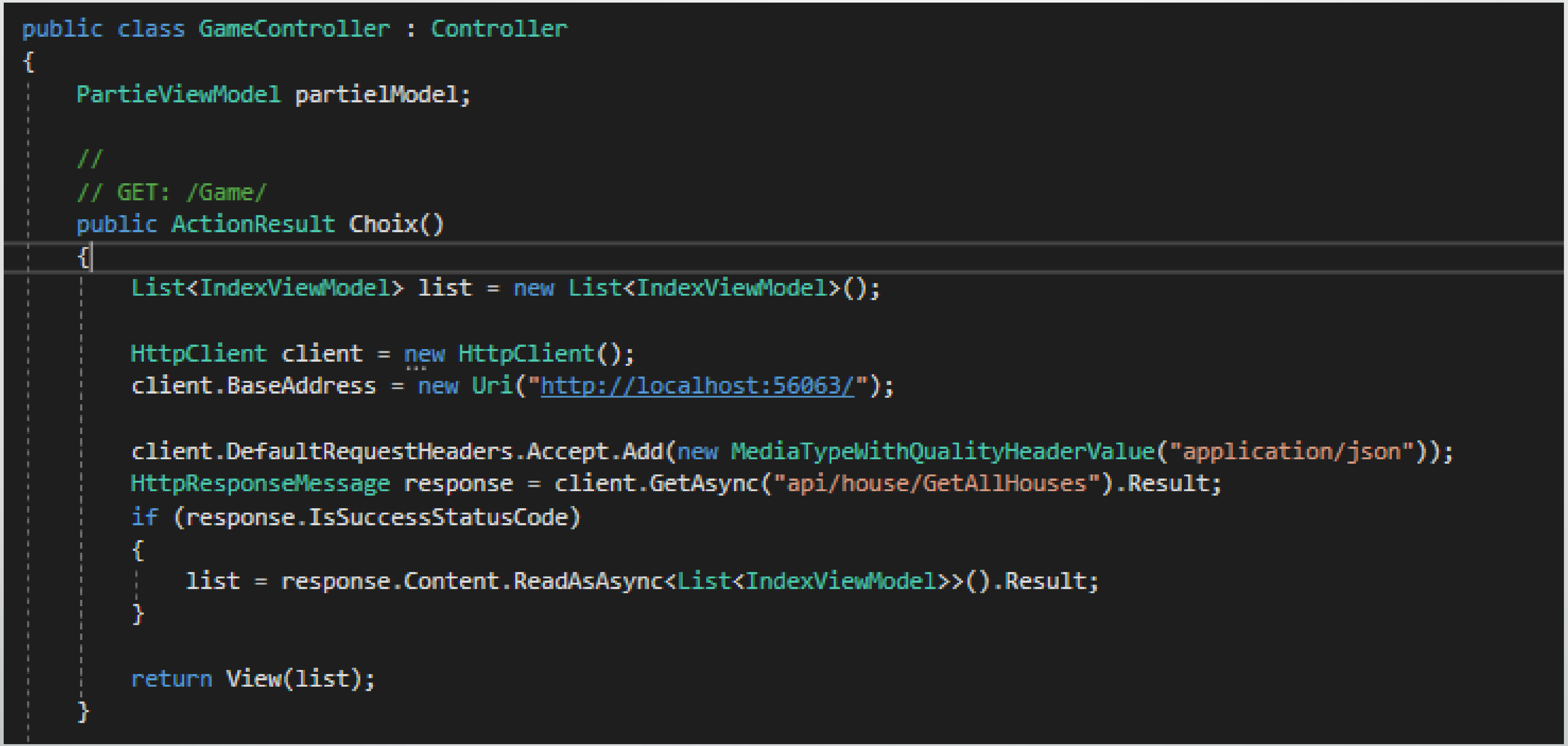


```
[Route("SaveHouse/{name}/{numberOfUnities}")]
[HttpPost]
public void SaveHouse(String name, int numberOfUnities)
{
    businessManager.AddHouse(name,numberOfUnities);

}

[Route("UpdateHouse/{idHouse}/{name}/{numberOfUnities}")]
[HttpPut]
public void UpdateHouse(int idHouse, String name, int numberOfUnities)
{
    businessManager.UpdateHouse(idHouse,name, numberOfUnities);

}

[Route("DeleteHouse/{idHouse}")]
[HttpDelete]
public void DeleteHouse(int idHouse)
{
    businessManager.DeleteHouse(idHouse);

}
```



```
public void SaveHouse(String name,int numberOfUnities)
{
    String insertHouseRequest = "INSERT INTO House(name,numberOfUnities) VALUES (@Name,@NumberOfUnities)";

    using (SqlConnection sqlConnection = new SqlConnection(connexionString))
    {
        sqlConnection.Open();

        SqlCommand insertCommand = new SqlCommand(insertHouseRequest, sqlConnection);
        insertCommand.Parameters.AddWithValue("@Name", name);
        insertCommand.Parameters.AddWithValue("@NumberOfUnities", numberOfUnities);
        insertCommand.ExecuteNonQuery();

        sqlConnection.Close();
    }
}
```

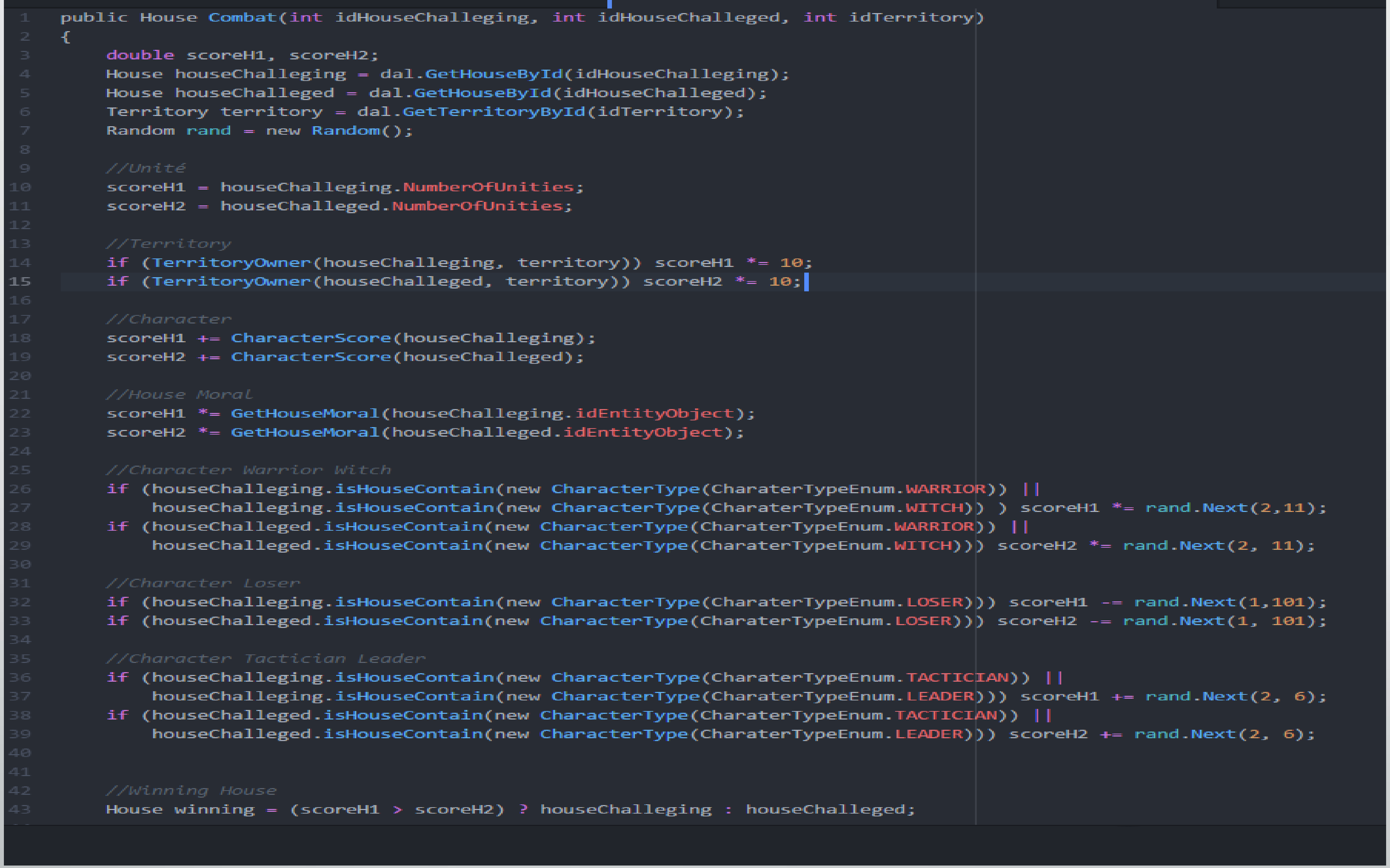
Couche Présentation



```
public class GameController : Controller
{
    PartieViewModel partielModel;

    //
    // GET: /Game/
    public ActionResult Choix()
    {
        List<IndexViewModel> list = new List<IndexViewModel>();

        HttpClient client = new HttpClient();
        client.BaseAddress = new Uri("http://localhost:56063/");

        client.DefaultRequestHeaders.Accept.Add(new MediaTypeWithQualityHeaderValue("application/json"));
        HttpResponseMessage response = client.GetAsync("api/house/GetAllHouses").Result;
        if (response.IsSuccessStatusCode)
        {
            list = response.Content.ReadAsAsync<List<IndexViewModel>>().Result;
        }

        return View(list);
    }
```

Règles du jeu

Déroulement d’un combat



```
public House Combat(int idHouseChalleging, int idHouseChalleged, int idTerritory)
{
    double scoreH1, scoreH2;
    House houseChalleging = dal.GetHouseById(idHouseChalleging);
    House houseChalleged = dal.GetHouseById(idHouseChalleged);
    Territory territory = dal.GetTerritoryById(idTerritory);
    Random rand = new Random();

    //Unité
    scoreH1 = houseChalleging.NumberOfUnities;
    scoreH2 = houseChalleged.NumberOfUnities;

    //Territory
    if (TerritoryOwner(houseChalleging, territory)) scoreH1 *= 10;
    if (TerritoryOwner(houseChalleged, territory)) scoreH2 *= 10;

    //Character
    scoreH1 += CharacterScore(houseChalleging);
    scoreH2 += CharacterScore(houseChalleged);

    //House Moral
    scoreH1 *= GetHouseMoral(houseChalleging.idEntityObject);
    scoreH2 *= GetHouseMoral(houseChalleged.idEntityObject);

    //Character Warrior Witch
    if (houseChalleging.isHouseContain(new CharacterType(CharaterTypeEnum.WARRIOR)) ||
        houseChalleging.isHouseContain(new CharacterType(CharaterTypeEnum.WITCH)) ) scoreH1 *= rand.Next(2,11);
    if (houseChalleged.isHouseContain(new CharacterType(CharaterTypeEnum.WARRIOR)) ||
        houseChalleged.isHouseContain(new CharacterType(CharaterTypeEnum.WITCH))) scoreH2 *= rand.Next(2, 11);

    //Character Loser
    if (houseChalleging.isHouseContain(new CharacterType(CharaterTypeEnum.LOSER))) scoreH1 -= rand.Next(1,101);
    if (houseChalleged.isHouseContain(new CharacterType(CharaterTypeEnum.LOSER))) scoreH2 -= rand.Next(1, 101);

    //Character Tactician Leader
    if (houseChalleging.isHouseContain(new CharacterType(CharaterTypeEnum.TACTICIAN)) ||
        houseChalleging.isHouseContain(new CharacterType(CharaterTypeEnum.LEADER))) scoreH1 += rand.Next(2, 6);
    if (houseChalleged.isHouseContain(new CharacterType(CharaterTypeEnum.TACTICIAN)) ||
        houseChalleged.isHouseContain(new CharacterType(CharaterTypeEnum.LEADER))) scoreH2 += rand.Next(2, 6);


    //Winning House
    House winning = (scoreH1 > scoreH2) ? houseChalleging : houseChalleged;
```

Gestion du projet

Affectation des tâches

tâches	Concernés
Conception	Youssef et Anass
Appli console	Ravel
Dev des api	Said
Dev site web	Hamza
Tests	Youssef

Travail en équipe sur github : https://github.com/nidabrahim/GameOfThrones

Démonstration