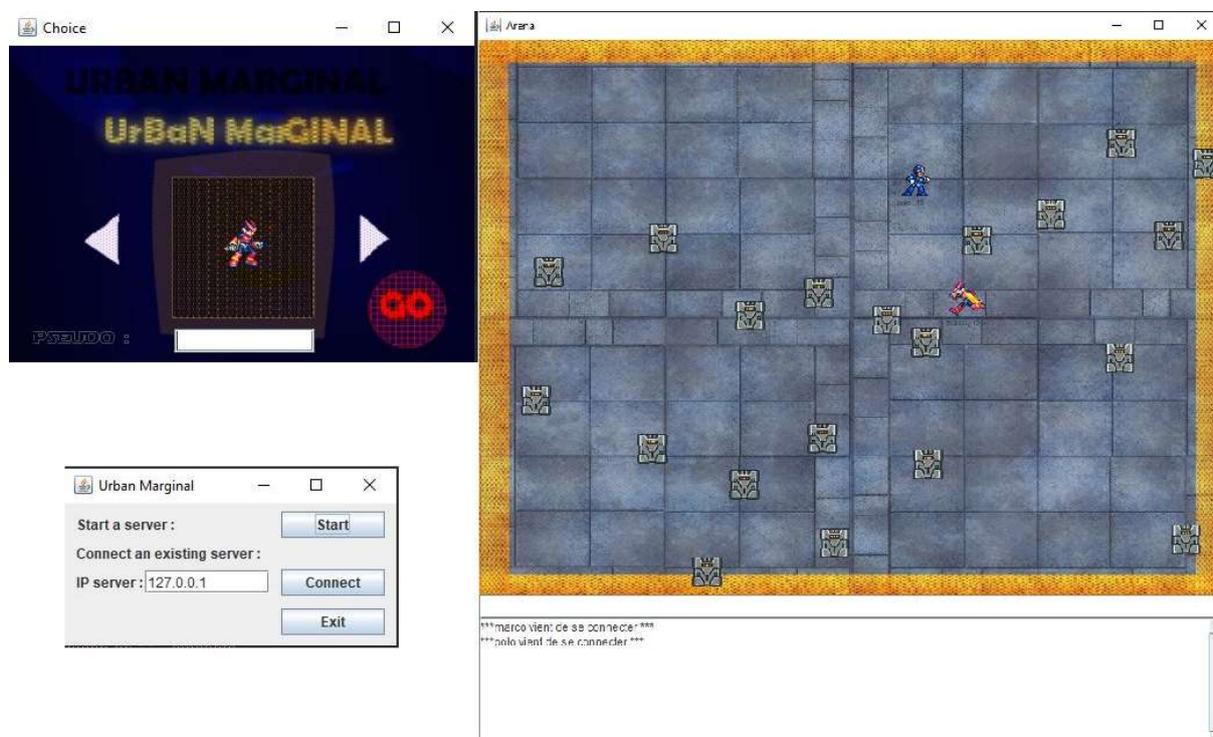


# Documentation du jeu Urban Marginal



# TABLE DES MATIÈRES

---

1. Présentation du jeu	3
2. Les outils utilisés	3
3. Architecture	4
3.1 Diagramme de classe	5
4. Conception	6
4.1 Découverte IDE Eclipse	6
4.2 Première frame	6
4.3 Construction partie Client/Serveur	7
4.4 Frame choix du joueur	7
4.5 Frame de l'arène	7
4.6 Classes principales	8
5. Tests unitaires	8-9
6. Déploiement	9
7. Pour conclure	9

# 1. Présentation du jeu

---

Urban Marginal est un jeu client/serveur multijoueur en 2D construit sur le modèle MVC (Modèle/View/Contrôleur). Le but est de battre l'adversaire en tirant des boules d'énergies. Un chat est disponible pour discuter. Il existe trois personnages différents.

Pour lancer le jeu, démarrez une nouvelle session Urban Marginal et cliquez sur Start (session serveur). Lancez une seconde session et cliquez sur Connect (session client).

## 2. Les outils utilisés

---

Les Langages :

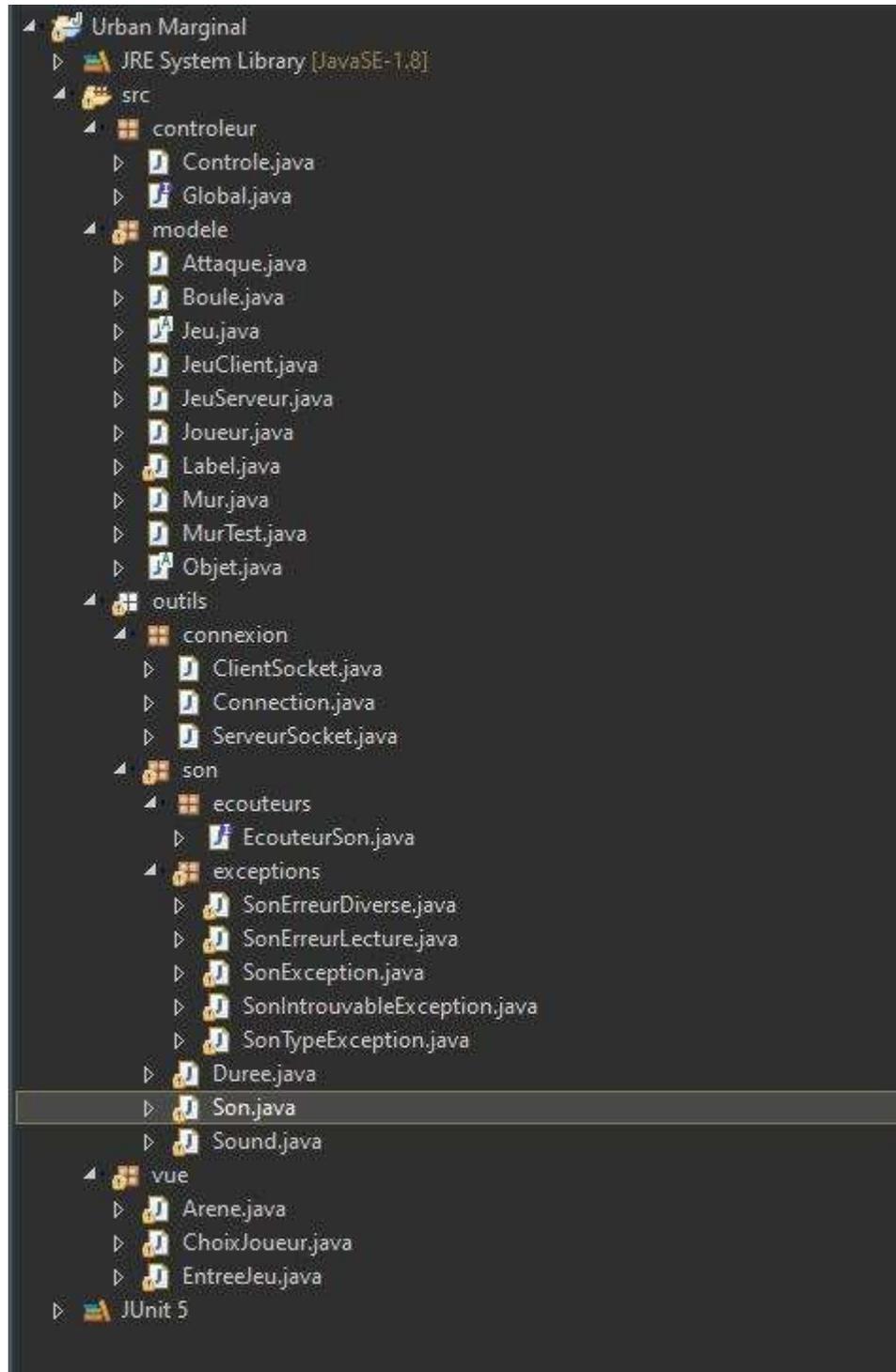
- JavaSE-1.8

Les logiciels :

- Eclipse version Photon Release (4.8.0) + WindowsBuilder

## 3. Architecture

---



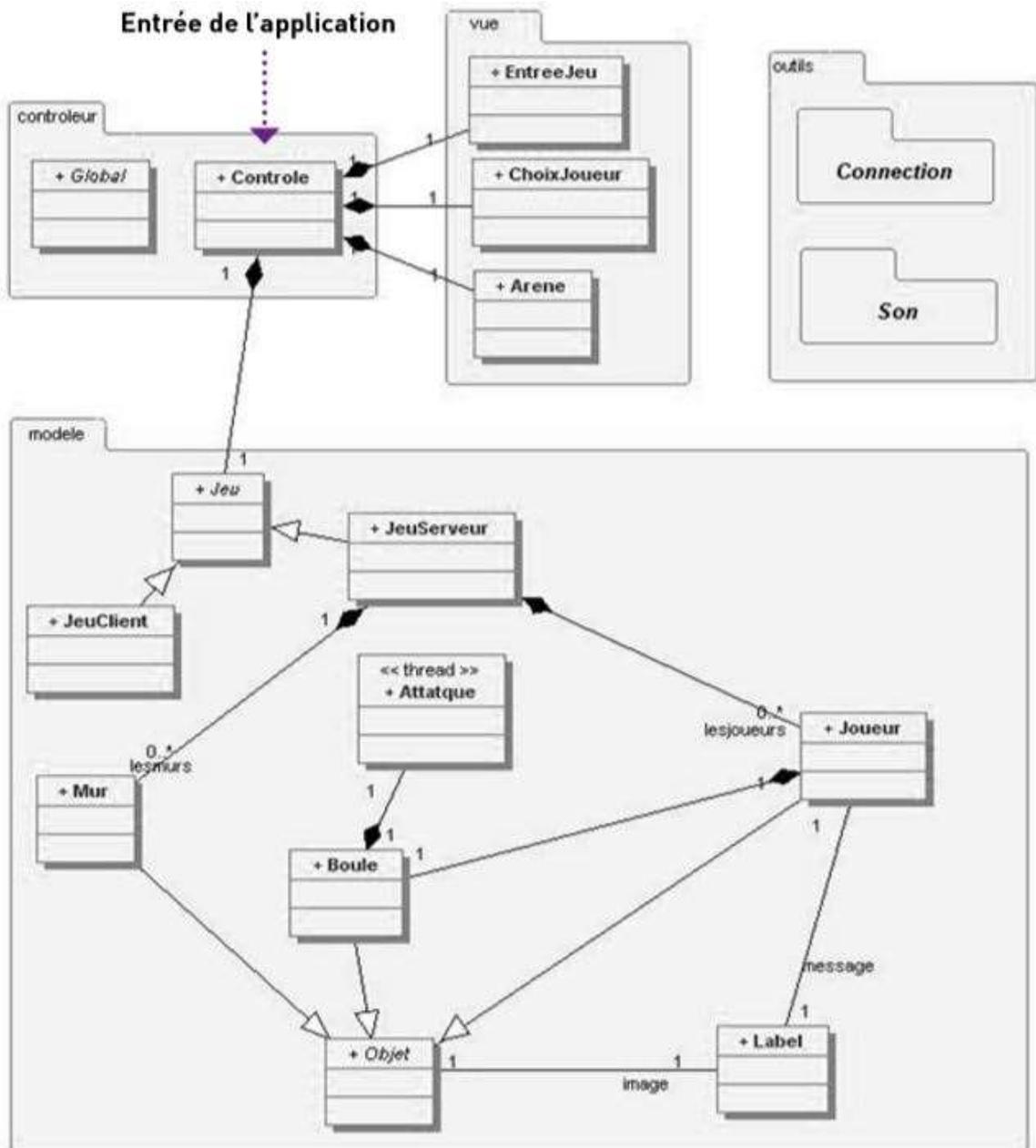
### 3.1. Diagramme de classe

**Package controleur** : Il contient l'interface Global qui sera implémentée par quasiment toutes les classes de l'application. Cette interface apporte toutes les constantes globales. Il contient aussi la classe Controle. L'application démarre sur cette classe qui s'occupe d'instancier les 3 fenêtres et la classe Jeu du package modele. La classe Controle va s'occuper de faire le lien entre la vue et le modele.

**Package vue** : Il contient les trois fenêtres de l'application.

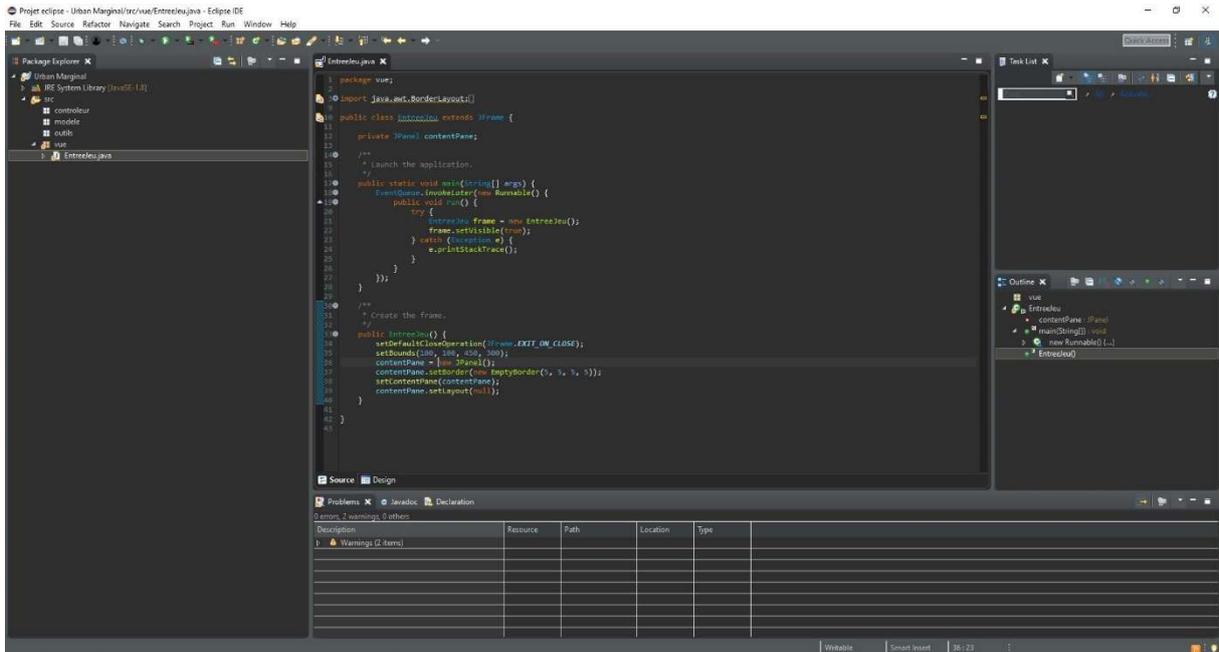
**Package modele** : Il contient les classes métiers de l'application. On y trouve la classe Jeu avec 2 classes filles (pour distinguer le jeu côté serveur ou côté client), la classe Objet pour représenter les objets qui seront manipulés (Joueur, Mur, Boule), la classe Label pour mémoriser et numéroter un JLabel (un JLabel est l'objet graphique qui servira pour l'affichage et qui sera transmis entre le serveur et le client via le réseau) et la classe Attaque (qui permet de gérer le tir d'une boule sans bloquer le reste du jeu).

**Package outils** : Il contient deux packages qui seront réutilisables pour d'autres applications (le package connexion pour gérer l'aspect client/serveur et le package son pour jouer du son !).



## 4. Conception

### 4.1. Découverte IDE Eclipse



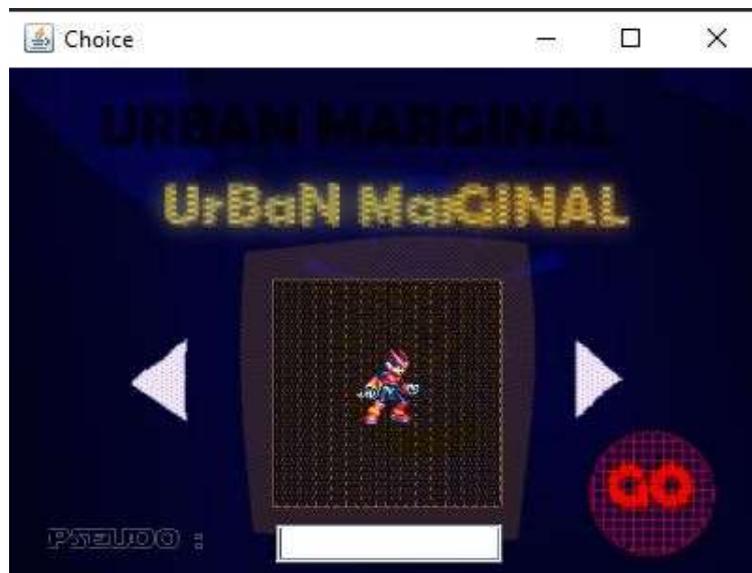
### 4.2. Première frame



### 4.3. Construction partie Client/Serveur

Construction des classes Controle, ServeurSocket, Connection, ClientSocket. Puis création des classes Jeu, JeuClient, JeuServeur

### 4.4. Frame choix du joueur



### 4.5. Frame de l'arène



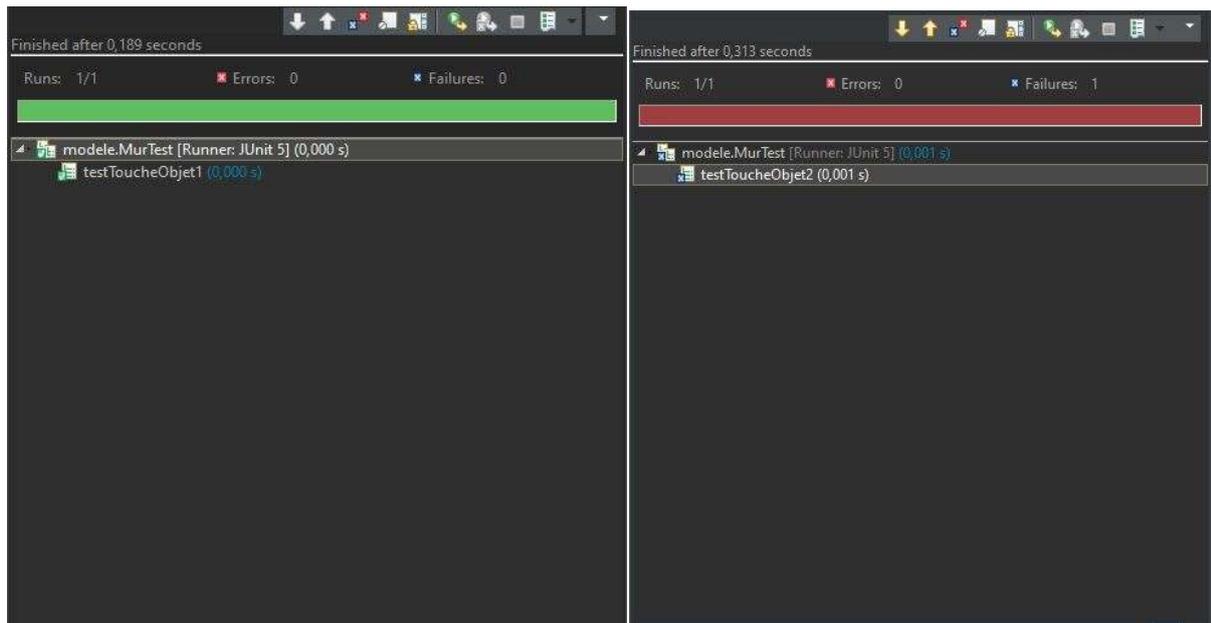
## 4.6. Classes principales

Création des classes principales Label, Objet.  
Création de la classe Mur.  
Création de la classe Joueur.  
Création de la classe Boule.  
Création de la classe Attaque.  
Création de la classe Son

## 5. Tests unitaires

---

Les tests ont été réalisés avec JUnit 5. Les tests effectués avaient pour but de savoir si la collision des différents objets était fonctionnelle.



```

1  /**
4  package modele;
5
6  import static org.junit.Assert.*;
9
10 /**
11  * @author Fernandez Tony
12  *
13  */
14 public class MurTest {
15
16     private Mur mur1 = new Mur() ;
17     private Mur mur2 = new Mur() ;
18
19     /**
20     * Test method for {@link modele.Objet#toucheObjet(modele.Objet)}.
21     */
22     @Test
23     public void testToucheObjet1() {
24         mur1.posX = 10 ;
25         mur1.posY = 10 ;
26         mur2.posX = 20 ;
27         mur2.posY = 20 ;
28         assertTrue(mur1.toucheObjet(mur2));
29     }
30
31     /**
32     * Test method for {@link modele.Objet#toucheObjet(modele.Objet)}.
33     */
34     @Test
35     public void testToucheObjet2() {
36         mur1.posX = 10 ;
37         mur1.posY = 10 ;
38         mur2.posX = 60 ;
39         mur2.posY = 60 ;
40         assertTrue(mur1.toucheObjet(mur2));
41     }
42 }
43 }
44

```

## 6. Déploiement

---

Le projet a été compilé en .jar et testé sur 2 postes différents.

## 7. Conclusion

---

Ce TP m'a permis d'apprendre énormément sur la programmation objet, le système de classe et le langage java.

Pour tester le jeu ou télécharger le projet merci de suivre ce lien :

- [Jeu : Téléchargement](#)