

# Premier Rapport de Soutenance

*Projet Stick 'n' Rage*

Groupe Slash

5 janvier 2009

# Table des matières

<b>1</b>	<b>Le Projet</b>	<b>3</b>
1.1	Précisions du Sujet . . . . .	3
1.1.1	Rappel des tâches : . . . . .	3
<b>2</b>	<b>Moteur Physique</b>	<b>4</b>
2.1	Avances . . . . .	4
2.1.1	Théorie de Newton . . . . .	4
2.2	Prévisions en vue de la Seconde Soutenance . . . . .	6
<b>3</b>	<b>Moteur Graphique</b>	<b>7</b>
3.1	Modelisation . . . . .	7
3.1.1	Réalisation du Logo . . . . .	7
3.1.2	Outils Graphiques . . . . .	7
3.1.3	Mesh . . . . .	8
3.1.4	Animations . . . . .	9
3.2	OpenGL / Delphi . . . . .	9
3.2.1	Introduction . . . . .	9
3.2.2	GLScene . . . . .	9
3.2.3	Level Design . . . . .	10
3.2.4	Personnages . . . . .	11
3.2.5	Fonctions de prévisualisation . . . . .	11
3.3	Prévisions pour la seconde soutenance . . . . .	12
<b>4</b>	<b>Retour sur le planning</b>	<b>13</b>
4.1	Site Web . . . . .	13
<b>5</b>	<b>Bibliographie</b>	<b>14</b>
5.1	Partie Graphique . . . . .	14
5.1.1	Blender . . . . .	14
5.1.2	OpenGL . . . . .	14
5.2	Partie Physique . . . . .	14
5.2.1	Theorie . . . . .	14

# Introduction

Dans notre cahier des charges, nous avons élaborer un planning en prévision de cette première Soutenance concernant le moteur Physique, le moteur Graphique ainsi que les Mesh, nous allons donc vous exposer à l'oral l'avancé de ces trois parties ainsi que nos previsions pour la prochaine soutenance.

-2cm



# Chapitre 1

## Le Projet

### 1.1 Précisions du Sujet

Ce projet sera donc, comme nous l'avons déjà dit dans notre cahier des charges, un jeu de combat 2D orienté vers des combats de type Tekken Like. Nous utiliserons pour ceci le personnage du Stickman qui nous évitera d'avoir un rendu moche et dangereux.

#### 1.1.1 Rappel des tâches :

Taches	Rippalka	Pull	Msb	Turgal
Moteur Graphique	+	+		
Moteur Physique			+	+
Son			+	+
Site Web	+		+	
Menu		+		+
IA				+
Mesh	+	+		
Editeur de map	+	+	+	+

## Chapitre 2

# Moteur Physique

### 2.1 Avances

Pour cette première soutenance, nous avons décidé de gérer les collisions avec un environnement quelconque avec de simples formes comme un carré. Nous pouvons donc dire "Mission accomplie" car nous avons réussi à animer un carré pouvant bouger et sauter (bien que l'interaction avec l'obstacle soit encore à améliorer). De plus, nous sommes en mesure d'afficher les coordonnées d'un objet soumis à une vitesse quelconque et un angle précis (bien que l'interaction avec l'obstacle soit encore à améliorer).

#### 2.1.1 Théorie de Newton

Grâce à la théorie de Newton, que nous avons appris en cours, et plus précisément grâce à la seconde loi, Nous avons réussi à modéliser des trajectoires d'objets ainsi que des collisions de ce même objet avec les éléments du décor.

La trajectoire a été réalisée en utilisant des vecteurs vitesses, des angles ainsi que des coordonnées x/y dans un repère à partir des formules suivantes :

Soit  $V$  le vecteur vitesse et  $V.X$ ,  $V.Y$  respectivement les composantes  $X$  et  $Y$  du vecteurs.

$$\vec{V} = \begin{cases} X = v_0 * \cos(\alpha) * t \\ Y = v_0 * \sin(\alpha) * t + \frac{t^2}{2} \end{cases}$$

Voici une partie de notre implantation des vecteurs vitesses nous permettant de placer chaque coordonnée du vecteur dans un tableau :

```

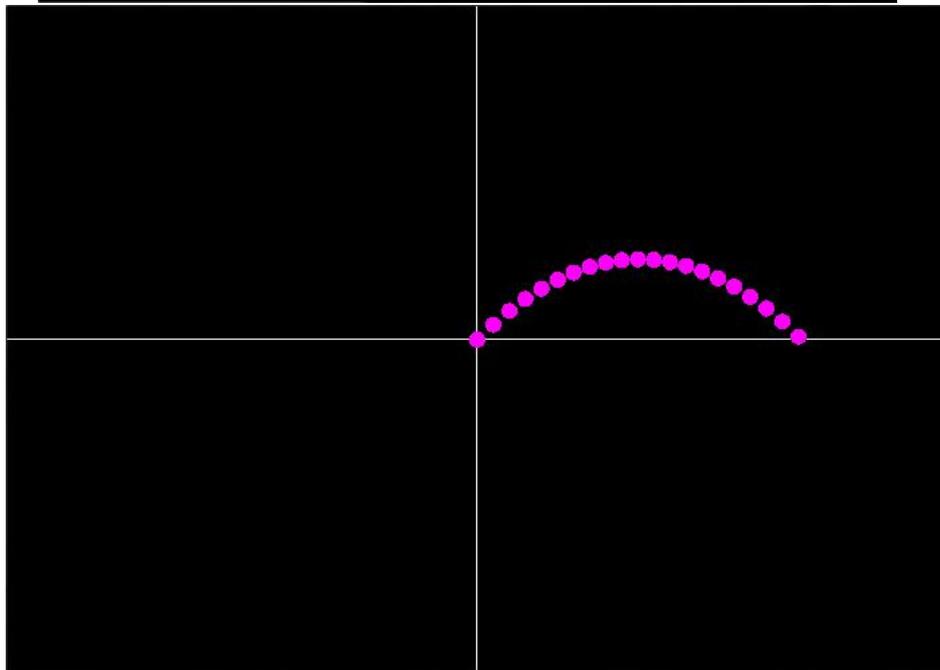
while P.Y >= sol do
  begin
    P.X := VX(coeff, v0)*t;
    P.Y := VY(coeff, v0)*t+g.Y*t*t/2;
    t:=t+dt;
    tab[li, col] := P.X;
    tab[li, col+1] := P.Y;
    li := li+1;
  end;

```

```

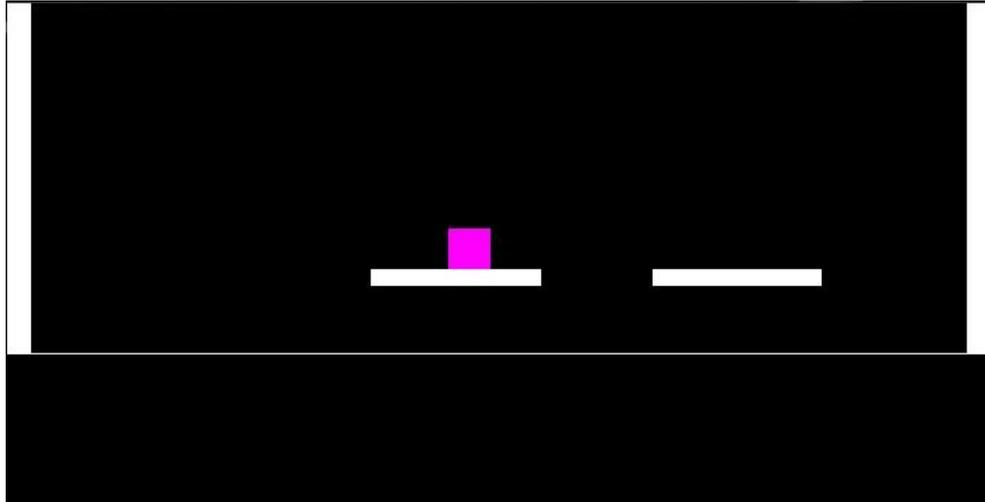
P.X := 0.000000000000E+0000 P.Y := 0.000000000000E+0000
P.X := 6.92964645562817E-0001 P.Y := 6.68954645562816E-0001
P.X := 1.38592929112563E+0000 P.Y := 1.28988929112563E+0000
P.X := 2.078897393668845E+0000 P.Y := 1.862889393668845E+0000
P.X := 2.77185858225127E+0000 P.Y := 2.38769858225127E+0000
P.X := 3.46482322781408E+0000 P.Y := 2.86457322781408E+0000
P.X := 4.15778787337690E+0000 P.Y := 3.29342787337690E+0000
P.X := 4.85075251893972E+0000 P.Y := 3.67426251893972E+0000
P.X := 5.54371716450253E+0000 P.Y := 4.00707716450253E+0000
P.X := 6.23668181006535E+0000 P.Y := 4.29187181006535E+0000
P.X := 6.92964645562817E+0000 P.Y := 4.52064645562816E+0000
P.X := 7.62261110119098E+0000 P.Y := 4.71740110119098E+0000
P.X := 8.31557574675380E+0000 P.Y := 4.85813574675380E+0000
P.X := 9.00854039231662E+0000 P.Y := 4.95085039231661E+0000
P.X := 9.70150503787944E+0000 P.Y := 4.99554503787943E+0000
P.X := 1.03944696834422E+0001 P.Y := 4.99221968344223E+0000
P.X := 1.10874343290050E+0001 P.Y := 4.94087432900506E+0000
P.X := 1.17803989745678E+0001 P.Y := 4.84150897456788E+0000
P.X := 1.247333636201307E+0001 P.Y := 4.69412362013069E+0000
P.X := 1.31663282656935E+0001 P.Y := 4.49871826569351E+0000
P.X := 1.38592929112563E+0001 P.Y := 4.25529291125633E+0000
P.X := 1.45522575568191E+0001 P.Y := 3.96384755681914E+0000
P.X := 1.52452222023819E+0001 P.Y := 3.62438220238196E+0000
P.X := 1.59381868479447E+0001 P.Y := 3.23689684794477E+0000
P.X := 1.66311514935076E+0001 P.Y := 2.80139149350759E+0000
P.X := 1.73241161390704E+0001 P.Y := 2.31786613907040E+0000
P.X := 1.80170807846332E+0001 P.Y := 1.78632078463322E+0000
P.X := 1.87100454301961E+0001 P.Y := 1.20675543019603E+0000
P.X := 1.94030100757589E+0001 P.Y := 5.79170075758847E-0001
P.X := 0.000000000000E+0000 P.Y := 0.000000000000E+0000

```



Nous avons ensuite ralis grce la trajectoire, des collisions avec des lments du dcor comme des obstacles ou l'environnement dans lequel

l'objet est enfermé. Il nous a suffi de récupérer dans un tableau les coordonnées successives de l'objet dans un intervalle de temps prédéfini et d'utiliser nos fonctions génériques pour pouvoir faire n'importe quelle application.



## 2.2 Prévisions en vue de la Seconde Soutenance

Pour la Seconde soutenance, les problèmes avec les obstacles seront résolus et nous serons même de nous orienter vers notre jeu. Ainsi, nous mettrons en commun les travaux concernant le moteur physique et le moteur graphique.

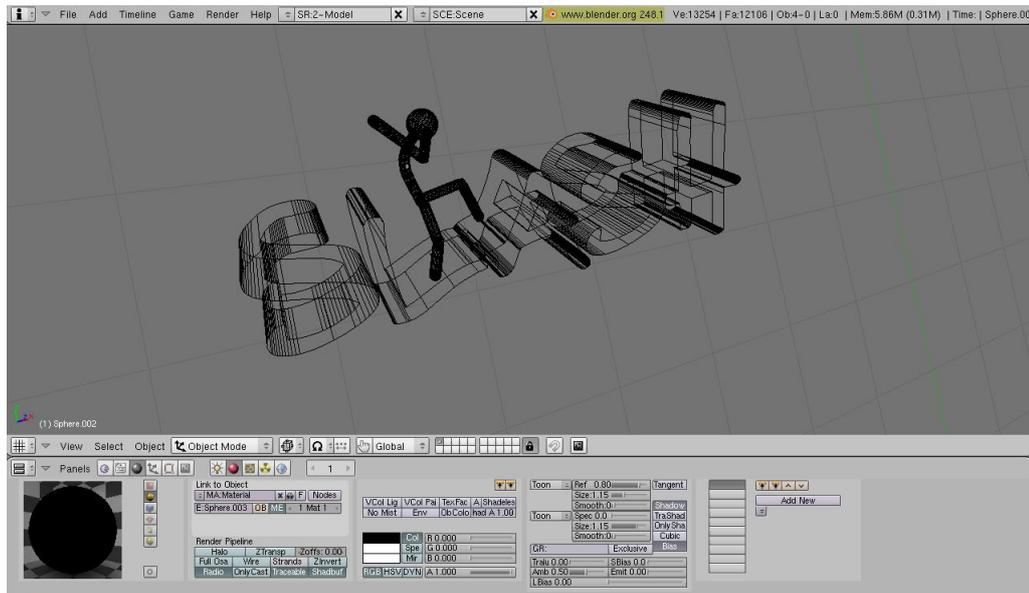
# Chapitre 3

## Moteur Graphique

### 3.1 Modelisation

#### 3.1.1 Réalisation du Logo

Nous nous sommes tout d'abord servi de Blender pour réaliser un Logo 3D comprenant notre Stickman et le nom 'SLASH'.



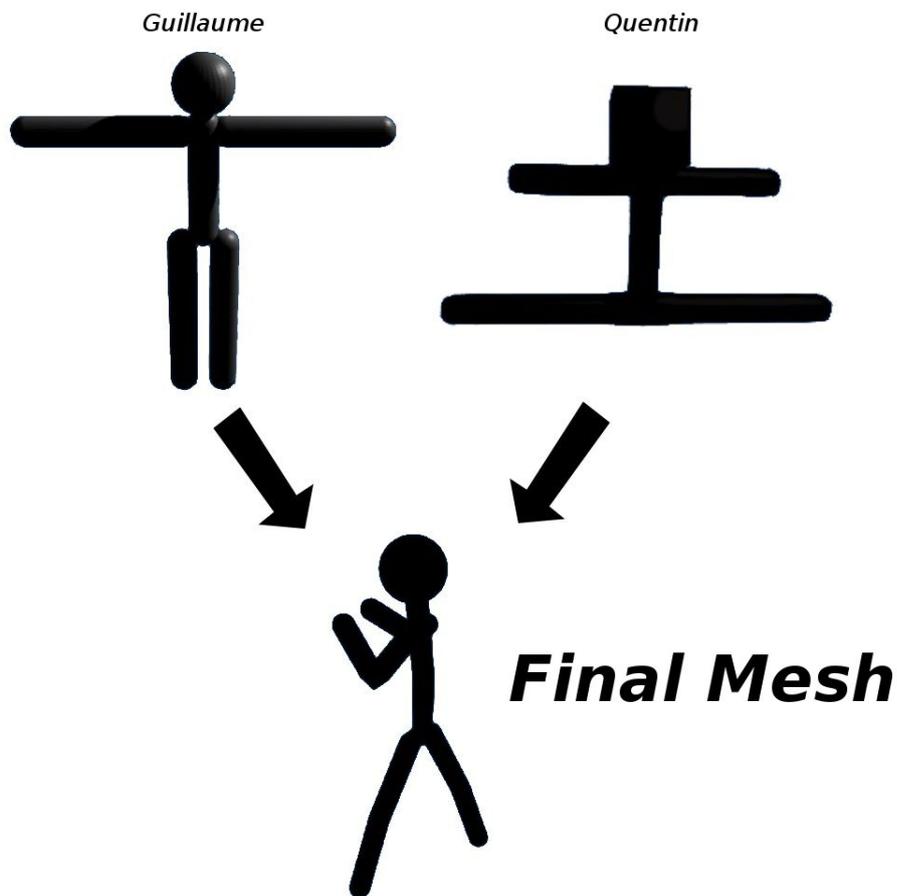
#### 3.1.2 Outils Graphiques

Après de nombreuses heures passées comprendre les utilisations d'OpenGL et de Blender, nous avons pris conscience de la puissance de ces deux outils, et par conséquent nous allons poursuivre notre projet avec ceux-ci. Malgré cela, ces deux outils ont vite laissé paraître leurs défauts, notamment au niveau des animations. En effet Blender ne gère l'export d'animation qu'avec quelques formats comme le MD2

( Format de Fichier de Quake II ), et ceux-ci s'importent très mal avec OpenGL. Or, notre jeu reposera entièrement sur des animations, c'est pourquoi nous sommes toujours la recherche d'une solution.

### 3.1.3 Mesh

Dans un premier temps, nous nous sommes concentrés sur la réalisation de notre personnage principal, savoir un Stickman. Nous nous sommes documentés séparément, puis avons synthétisé nos recherches pour obtenir le FINAL MESH, ci-dessous :



Comme on peut le voir, Quentin était parti sur un mesh base de cubes, Guillaume base de cylindres et de sphères. Les deux étant composés d'énormément de points (celui de Quentin dépassait les 215 000 Vertex!!!) et n'ayant pas un rendu acceptable, nous avons dû uniformiser et optimiser tout cela. Le rendu final contient près de 6000 Vertex et est plus vrai que nature.

### 3.1.4 Animations

Grâce à ce Mesh, nous avons réalisé plusieurs animations basiques sous Blender, à savoir : coup de poing, coup de pied, animation au repos et marche. Nous les avons réalisées en vue de les importer dans OpenGL par la suite. Le format d'export est toujours déterminé car malgré plusieurs tentatives avec Blender aux formats : .3ds .md2 .md3 .md5 .obj, les animations ne sont toujours pas exportées. Nous avons néanmoins fait une tentative en rechargeant le mesh pour chaque Frame (le mesh ayant précédemment été exporté en plusieurs fichiers .3ds pour chaque Frame), mais ceci ne nous paraît pas être une solution optimale car cela équivaldrait à charger 25 fois le mesh en une seconde (25FPS). Puis nous avons pour l'instant provisoirement lié les touches du clavier pour déplacer simultanément les deux personnages selon les axes X/Y.

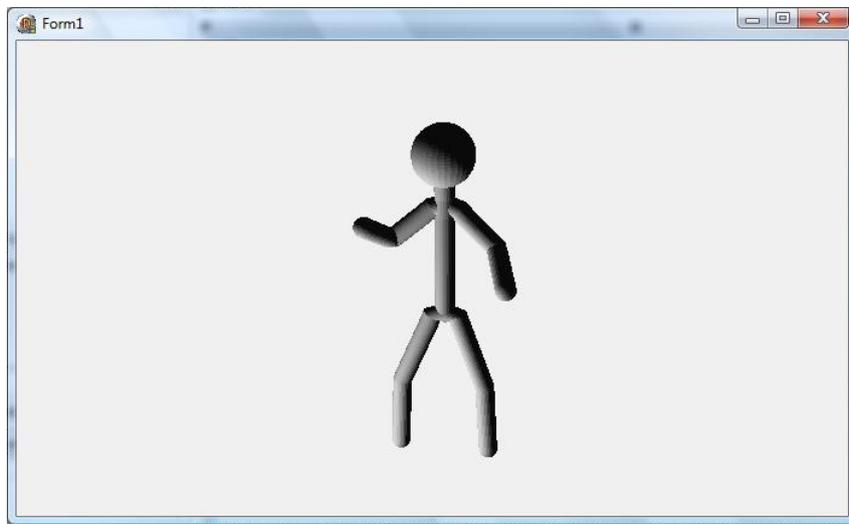
## 3.2 OpenGL / Delphi

### 3.2.1 Introduction

Après avoir difficilement affiché nos premières primitives sous OpenGL, nous nous sommes rendu compte du niveau de difficulté pour la réalisation d'un rendu graphique de jeu vidéo. C'est pourquoi nous avons cherché des bibliothèques pour nous aider et c'est ainsi que nous sommes tombé sur GLScene et Glut.

### 3.2.2 GLScene

Nous avons choisi d'utiliser principalement la bibliothèque 3D GLScene au dépens de Glut sur laquelle nous ne nous sommes pas encore vraiment intéressé. La documentation pour cette bibliothèque est pratiquement inexistante, mais néanmoins GLScene nous a permis de réaliser rapidement un rendu satisfaisant grâce à son fonctionnement intuitif. En effet, GLScene permet la création de scènes complètes : Camera, Source de lumière, Objets/Acteurs. Nous l'avons donc pour l'instant utilisé dans la majorité des tâches effectuées.



### 3.2.3 Level Design

Pour cette première soutenance, nous nous sommes principalement intéressé au Level Design, c'est-à-dire à l'élaboration du terrain de jeu. Nous sommes tout d'abord sur l'idée d'un décor basic (montagne, hangar) mais nous avons pris conscience de la monotonie de la scène.



C'est pourquoi nous avons créé un plan de scène un peu plus complexe, que nous avons réalisé en trois parties :

- GLSkyBox

La SkyBox est en faite un cube englobant la camera de la scène de telle manière qu'elle soit en permanence fixée au centre de ce cube. En chargeant une texture sur chaque face de cette SkyBox, elle donne un effet d'horizon à la scène.

- Environnement

Pour éviter d'obtenir une scène isolée, nous avons immergé notre arène de jeu au milieu d'une ville qui est en harmonie avec la SkyBox. En effet, le mélange de la Skybox et de la ville 3D renforce l'effet d'immersion. Cette ville est en faite un mesh trouvé sur internet que nous avons redimensionné puis chargé dans OpenGL.

- Arène

Toujours dans l'optique d'immersion du joueur dans le jeu, l'arène est située au sommet d'un immeuble, lui-même situé au milieu de la ville. Pour cela, nous avons simplement réalisé l'intégralité de l'immeuble constituant l'arène à l'aide de cubes. De plus, l'arène comprend un DummyCube (Cube invisible sur la scène) à son centre sur lequel est centré la camera principale. Ceci permet une vue en permanence de l'arène.

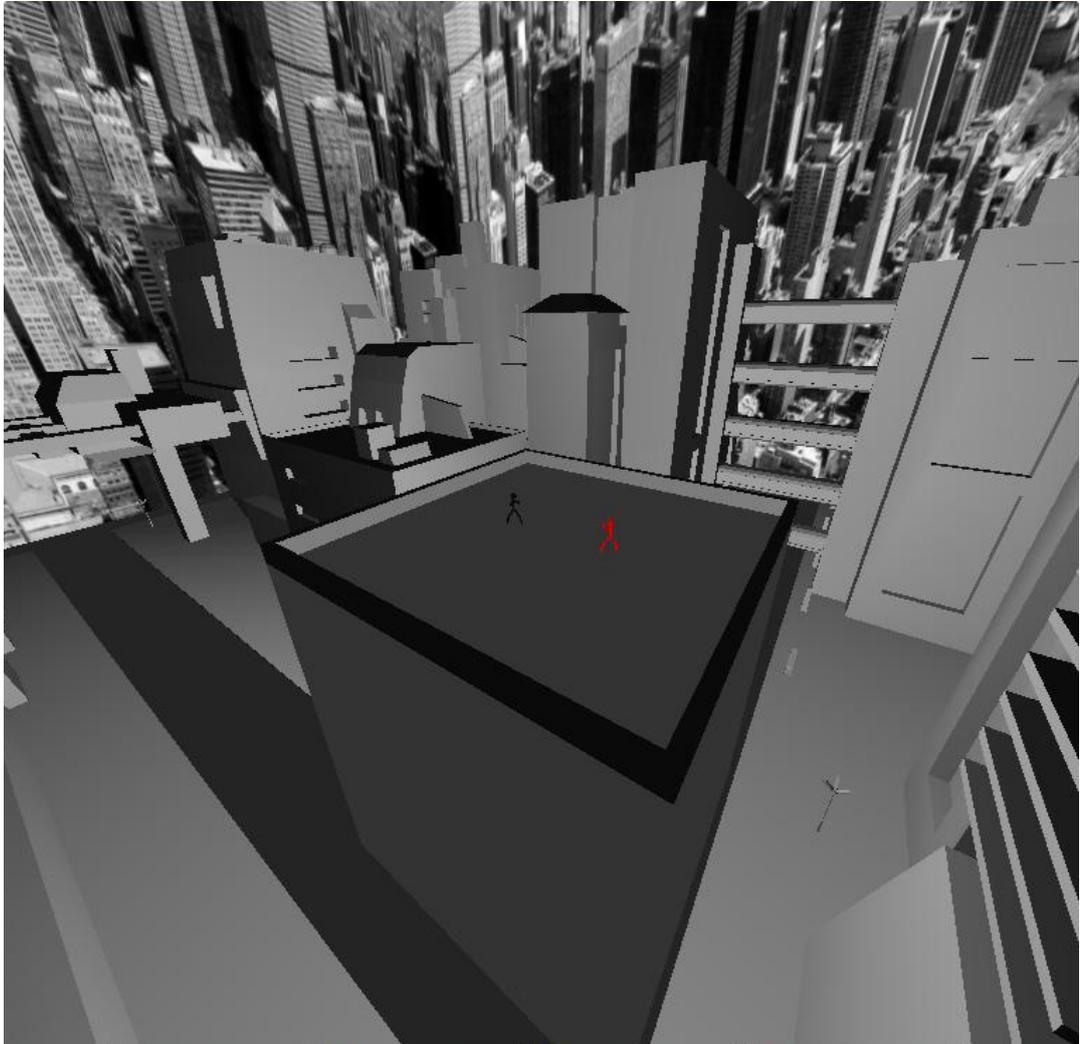
Par soucis de notre univers de Stickman, nous avons opté pour un environnement en noir et blanc. En effet, cela permet de mettre en avant les personnages qui sont eux en couleur.

### 3.2.4 Personnages

Nous avons, grâce à GLScene, importé nos Mesh que nous avons placé au milieu de l'arène. Nous avons ensuite ré-appliqué les textures sur nos personnages, car celles-ci ont disparu lors de l'export au format .3ds à partir de Blender.

### 3.2.5 Fonctions de prévisualisation

Pour l'instant nous avons créé quelques fonctions pour avoir une meilleure perception de nos rendus. Nous avons par exemple créé une fonction permettant grâce à la souris de se déplacer autour de la scène, ou encore une fonction ZOOM -/+.



### 3.3 Prévisions pour la seconde soutenance

Pour la deuxième soutenance, nous avons prévu d'avancer le système d'animation afin d'obtenir des animations fluides, puis par conséquent de créer une fonction permettant d'adapter la distance de la camera par rapport à la scène en fonction de la position des personnages. Ensuite nous allons commencer l'interface de jeu comprenant timer, barres de vie, etc...

## Chapitre 4

# Retour sur le planning

### 4.1 Site Web

Nous allons développer cette partie en plusieurs points comprenant le site web ainsi que le menu.

Le site internet, créé par Guillaume et Thomas, sera terminé en vue de la seconde soutenance, il comportera, comme cela a été dit précédemment dans le cahier des charges de début de projet : une présentation du groupe et du projet, une page de téléchargement ainsi qu'un module de news.

Le menu sera traité par Quentin et sera déjà bien entamé lors de la seconde soutenance où nous pourrons vous montrer l'avancée du travail.

# Chapitre 5

## Bibliographie

### 5.1 Partie Graphique

#### 5.1.1 Blender

- [www.SiteDuZero.com](http://www.SiteDuZero.com)
- [www.tuto-blender.com](http://www.tuto-blender.com)
- [www.daz3d.com](http://www.daz3d.com) (Ville 3D)
- [www.dafont.com](http://www.dafont.com)

#### 5.1.2 OpenGL

- [www.Developpez.com](http://www.Developpez.com)
- [www.codes-sources.com](http://www.codes-sources.com)
- [www.delhipage.free.fr/opengl.html](http://www.delhipage.free.fr/opengl.html)
- [www.glscene.org](http://www.glscene.org)
- [www.eraquila.iquebec.com/site/delphi/opengl/gl1.html](http://www.eraquila.iquebec.com/site/delphi/opengl/gl1.html)
- Google est notre ami

### 5.2 Partie Physique

#### 5.2.1 Theorie

- [www.Developpez.com](http://www.Developpez.com)
- [www.codes-sources.com](http://www.codes-sources.com)
- [www.programmersheaven.com](http://www.programmersheaven.com)
- Google est notre amis nous aussi.

# Conclusion

Pour conclure, nous dirons que nous ne sommes pas en retard sur le planning que nous avons fait. Passé le stade de la découverte des différents outils ! Notre groupe est prêt à partir sur de nos nouvelles bases pour mener à bien ce projet : Stick'n'Rage.