



MASTER 1<sup>re</sup> ANNÉE, M1IF39

---

# Projet : Monde virtuel distant semi-autonome

---

UNIVERSITÉ CLAUDE BERNARD LYON 1, DÉPARTEMENT INFORMATIQUE

---

Révision : 0.2.17  
En date du : 21 avril 2015  
Date de génération : 5 mai 2015

---

---

*Auteur:*  
**Erwan Guillou**

---

*Année:*  
**2014-2015**



---

---

# *Table des matières*

---

---

<b>I</b>	<b>Description des données</b>	<b>1</b>
<b>1</b>	<b>Gestion des ressources</b>	<b>3</b>
1.1	Nommage des ressources . . . . .	4
1.2	Création des UUID de ressource . . . . .	4
<b>2</b>	<b>Structures de base</b>	<b>7</b>
2.1	Vecteurs . . . . .	8
2.2	Triangle . . . . .	8
<b>3</b>	<b>Assets 2D et 3D</b>	<b>11</b>
3.1	Les images . . . . .	12
3.2	Les matériaux . . . . .	12
3.3	Les matériaux au format Alias Wavefront . . . . .	12
3.4	Les lumières . . . . .	13
3.5	Les maillages . . . . .	13
3.6	les groupes de faces . . . . .	14
<b>4</b>	<b>Description du monde virtuel</b>	<b>17</b>
4.1	Le monde virtuel . . . . .	18
4.2	Les régions du monde . . . . .	18
4.3	Les objets du monde . . . . .	19
4.4	Les entités . . . . .	19
4.5	Les points d'apparition dans le monde . . . . .	20
4.6	Les éléments constitutifs d'entités . . . . .	20



# **Première partie**

## **Description des données**



# *Chapitre 1*

---

---

## *Gestion des ressources*

---

---

### **Sommaire**

1.1	Nommage des ressources . . . . .	4
1.2	Création des UUID de ressource . . . .	4

---

### 1.1 Nommage des ressources

Afin d'assurer l'unicité des noms de ressource, le nom, interne au gestionnaire, de chaque ressource est défini comme suit :

$$Resource[Type : Nom : id]$$

où

- *Type* est le nom du type de la ressource
- *Nom* est le nom effectif de la ressource (issu des données des fichiers lus)
- *id* est un entier définissant le nombre de fois où l'on a vu une ressource du même type ayant le même nom

**Nommage des régions** Les régions sont définies par leur indice *i* et *j* dans le monde. Le nom de la région est défini par

$$Monde[i, j]$$

où *Monde* est le nom du monde. La ressource région sera alors nommée :

$$Resource[WorldCell : Monde[i, j] : 0]$$

### 1.2 Création des UUID de ressource

Un UUID est décrit par une chaîne de 38 caractères :

$$AAAAA - BBBB - CCCC - DDEE - FF GG HH II JJ KK$$

où

- *AAAAA* décrit la valeur hexadécimale d'un entier non signé (uint)
- *BBBB* et *CCCC* décrivent la valeur hexadécimale d'entiers courts non signés (ushort)
- *DD*, *EE*, *FF*, *GG*, *HH*, *II*, *JJ* et *KK* décrivent les valeurs hexadécimales d'octets non signés (uchar)

```
#define A 54059 /* a prime */
#define B 76963 /* another prime */
#define C 86969 /* yet another prime */
#define D 132241

typedef union {
    uint vui;
    struct {
        ushort s1;
        ushort s2;
    } vus;
    struct {
        uchar c1;
```



```
        uchar c2;
        uchar c3;
        uchar c4;
    } vuc;
} Key;

uint genhash(const char* s, uint a, uint b)
{
    uint h = 31 /* also prime */;
    while (*s) {
        h = (h * a) ^ (s[0] * b);
        s++;
    }
    return h; // or return h % C;
}

QUuid createUUID ( QString reference )
{
    Key key1, key2, key3, key4;
    key1.vui = genhash(reference.toStdString().c_str(), A,B);
    key2.vui = genhash(reference.toStdString().c_str(), B,C);
    key3.vui = genhash(reference.toStdString().c_str(), C,C);
    key4.vui = genhash(reference.toStdString().c_str(), D,A);
    QUuid uuid (      key1.vui,
                      key2.vus.s1, key2.vus.s2,
                      key3.vuc.c1, key3.vuc.c2, key3.vuc.c3, key3.
                        vuc.c4,
                      key4.vuc.c1, key4.vuc.c2, key4.vuc.c3, key4.
                        vuc.c4);
    return mapNameToUuid [ reference ];
}
```



# Chapitre 2

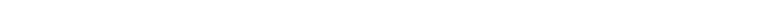


## Structures de base



### Sommaire

2.1	Vecteurs . . . . .	8
2.2	Triangle . . . . .	8



### 2.1 Vecteurs

**Vector2d** : défini par 2 float x et y. Encodé sur 2\*4 octets.

**Vector3d** : défini par 3 float x, y et z. Encodé sur 3\*4 octets.

**Vector4d** : défini par 4 float x, y, z et w. Encodé sur 4\*4 octets.

**Vector2ui** : défini par 2 unsigned int w et h. Encodé sur 2\*4 octets.

**ColorRGB** : défini par 3 float r, g et b. Encodé sur 3\*4 octets.

### 2.2 Triangle

```
typedef struct {  
    bool m_hasNormals;  
    bool m_hasTexcoords;  
    int m_vertexIndices [ 3 ];  
    int m_normalIndices [ 3 ];  
    int m_texcoordIndices [ 3 ];  
} Triangle;
```

**Encodage binaire** Voir table 2.1

TABLE 2.1 Encodage binaire d'un triangle

taille	type	contenu
1	bool	attribut hasNormals
1	bool	attribut hasTexcoords
3*4	int	attribut vertexIndices
3*4	int	attribut normalIndices
3*4	int	attribut texcoordIndices



# *Chapitre 3*

---

---

## *Assets 2D et 3D*

---

---

### **Sommaire**

3.1	Les images . . . . .	12
3.2	Les matériaux . . . . .	12
3.3	Les matériaux au format Alias Wavefront	12
3.4	Les lumières . . . . .	13
3.5	Les maillages . . . . .	13
3.6	les groupes de faces . . . . .	14

---

## 3.1 Les images

**Nom :** Image

**Type :** Ressource partagée

**Mode :** Interface et Implémentation

- Interface : [Image] → [80b90ea1-4539-0a94-ddb9-0ccb094cb446]
- Implémentation : [Image] → [0685f590-f83a-0e1f-d272-2a46b8321d24]

**Attributs :**

- *\_RealName* (QString), nom effectif de la ressource
- *Width* (unsigned int), largeur de l'image
- *Height* (unsigned int), hauteur de l'image
- *Depth* (unsigned int), nombre de bits par pixel
- *nChannels* (unsigned int), nombre de composante par pixel
- *Data* (unsigned char\*), données brutes de l'image

**Encodage binaire** Voir table 3.1

## 3.2 Les matériaux

**Nom :** Material

**Type :** Ressource partagée

**Mode :** Interface

- Interface : [Material] → [559b1809-a799-ae25-d1bd-03cb51a8c902]

**Attributs :** Selon implémentation

**Encodage binaire** Selon implémentation

## 3.3 Les matériaux au format Alias Wavefront

**Nom :** WavefrontMaterial [Material]

**Type :** Ressource partagée

**Mode :** Implémentation

- Implémentation : [WavefrontMaterial] → [3c50697c-543a-0e7f-6ab6-23ae942b73cc]



**Attributs :** TODO

**Encodage binaire** TODO

## 3.4 Les lumières

**Nom :** Light

**Type :** Ressource partagée

**Mode :** Interface et Implémentation

- Interface : [Light] → [1179fdb0-2986-e729-d4a0-12420ab6436f]
- Implementation : [Light] → [3cf07b05-6309-a9c5-fb18-c6991b107714]

**Attributs :**

- *\_RealName* (QString), nom effectif de la ressource
- *Type* (type énuméré), type de la source lumineuse (Ponctuelle, Directionnelle, Surfique)
- *Intensity* (float), intensité de la source lumineuse
- *Color* (Vector3d), couleur (dans l'espace RGB) de la source lumineuse

**Encodage binaire** Voir table 3.2

## 3.5 Les maillages

**Nom :** Mesh

**Type :** Ressource partagée

**Mode :** Interface et Implémentation

- Interface : [Mesh] → [1afd9661-9e85-bd2a-d96b-52ce8561271e]
- Implementation : [Mesh] → [4394cb82-98f0-f0ef-1e3a-e4956402ace7]

**Attributs :**

- *\_RealName* (QString), nom effectif de la ressource
- *#Vertices* (uint), nombre de sommets
- *Vertices* (Vector3d\*), description des sommets
- *#TexVertices* (uint), nombre de coordonnées de texture
- *TexVertices* (Vector2d\*), description des coordonnées de texture
- *#Normals* (uint), nombre de normales
- *Normals* (Vector3d\*), description des normales

- *#Faces* (uint), nombre de triangles sans matériaux
- *Faces* (Vector3d\*), description des triangles sans matériaux
- *MaterialGroups[]* (MaterialGroup\*), description des groupes de faces

**Encodage binaire** Voir table 3.3

### 3.6 les groupes de faces

**Nom :** MaterialGroup

**Type :** Ressource

**Mode :** Implémentation

**Attributs :**

- *Material* (QUuid), UUID du matériaux
- *Mesh* (QUuid), UUID du maillage parent
- *#Faces* (uint), nombre de triangles
- *Faces* (Triangle\*), description des triangles

**Encodage binaire** Voir table 3.4

TABLE 3.1 Encodage binaire d'une image

taille	type	contenu
38	UUID	Uuid du type de ressource
4	uint	Longueur du nom du type de la ressource
$n_1$	ASCII	nom du type de la ressource
38	UUID	Uuid de la ressource
4	uint	Longueur du nom de la ressource
$n_2$	ASCII	nom de la ressource
4	uint	attribut Width
4	uint	attribut Height
4	uint	attribut Depth
4	uint	attribut nChannels
$n_3$	uint	attribut Data

TABLE 3.2 Encodage binaire d'une lumière

taille	type	contenu
38	UUID	Uuid du type de ressource
4	uint	Longueur du nom du type de la ressource
$n_1$	ASCII	nom du type de la ressource
38	UUID	Uuid de la ressource
4	uint	Longueur du nom de la ressource
$n_2$	ASCII	nom de la ressource
4	uint	attribut Type
4	float	attribut Intensity
3*4	Vector3d	attribut Color

TABLE 3.3 Encodage binaire d'un maillage

taille	type	contenu
38	UUID	Uuid du type de ressource
4	uint	Longueur du nom du type de la ressource
$n_1$	ASCII	nom du type de la ressource
38	UUID	Uuid de la ressource
4	uint	Longueur du nom de la ressource
$n_2$	ASCII	nom de la ressource
4	uint	attribut #Vertices
$n_1*12$	Vector3d	attribut Vertices
4	uint	attribut #TexVertices
$n_2*8$	Vector2d	attribut TexVertices
4	uint	attribut #Normals
$n_3*12$	Vector3d	attribut Normals
4	uint	attribut #Faces
$n_4*38$	Triangle	attribut Faces
4	uint	nombre d'éléments de l'attribut MaterialGroups[]
$n_5*n$	MaterialGroup	contenu de l'attribut MaterialGroups[]

TABLE 3.4 Encodage binaire d'un groupe de faces

taille	type	contenu
4	uint	Longueur du nom du type de la ressource
$n_1$	ASCII	nom du type de la ressource
38	UUID	attribut Material
38	UUID	attribut Mesh
4	uint	attribut #Faces
n	Triangle*	attribut Faces

# *Chapitre 4*

---

---

## *Description du monde virtuel*

---

---

### **Sommaire**

4.1	Le monde virtuel . . . . .	18
4.2	Les régions du monde . . . . .	18
4.3	Les objets du monde . . . . .	19
4.4	Les entités . . . . .	19
4.5	Les points d'apparition dans le monde .	20
4.6	Les éléments constitutifs d'entités . . .	20

---

## Description du monde virtuel

---

Le monde virtuel est divisé, de façon régulière, en région. Chaque région contient un certain nombre d'objets. Chaque objet décrit une instance d'une entité positionnée dans la cellule. Chaque entité est décrite par un ensemble d'éléments constitutifs. Chaque élément constitutif, positionné relativement à l'entité, fait référence à des assets 2D et/ou 3d.

### 4.1 Le monde virtuel

**Nom :** World

**Type :** Ressource partagée

**Mode :** Interface et Implémentation

- Interface : [World] → [e60f58ac-4656-86f1-188a-e788121cfed7]
- Implementation : [World] → [8139645d-dc0d-a852-d77d-89ffe3ae9079]

**Attributs :**

- *\_RealName* (QString), nom effectif de la ressource
- *Extents* (Vector3d), dimensions du monde
- *Subdivision* (Vector2ui), découpage du monde en régions sur les axes X et Z
- *SpawnPoints[]* (SpawnPoint\*), liste des points d'appartition
- *Semantics[]* (QString), liste des informations sémantiques du monde
- *Cells[]* (QUuid), liste des UUIDs des régions du monde

**Encodage binaire** Voir table 4.1

### 4.2 Les régions du monde

**Nom :** WorldCell

**Type :** Ressource partagée

**Mode :** Interface et Implémentation

- Interface : [WorldCell] → [cf3451c8-8ab6-8282-26cf-acf314e75c8f]
- Implementation : [WorldCell] → [9855558d-bbf9-13a9-4946-8969a5bd7cc5]

**Attributs :**

- *\_RealName* (QString), nom effectif de la ressource
- *World* (QUuid), UUID du monde
- *Indices* (Vector2ui), indices de la région dans le monde
- *Extents* (Vector3d), dimensions de la région
- *Position* (Vector3d), position de la région (de son centre)

— *GameObjects[]* (QUuid), liste des objets de la région

**Encodage binaire** Voir table 4.2

## 4.3 Les objets du monde

**Nom :** GameObject

**Type :** Ressource partagée

**Mode :** Interface et Implémentation

- Interface : [GameObject] → [4d990a95-eae9-8802-4195-595c9db74560]
- Implementation : [GameObject] → [e1bf9bb2-3648-1d98-96da-a6f5fc95a5fb]

**Attributs :**

- *\_RealName* (QString), nom effectif de la ressource
- *Cell* (QUuid), UUID de la région
- *Position* (Vector3d), position de l'objet
- *Rotation* (Vector4d), rotation de l'objet
- *Scale* (Vector3d), mise à l'échelle de l'objet
- *Semantics[]* (QString), liste des informations sémantiques de l'objet
- *Entity* (QUuid), UUID de l'entité à instancier

**Encodage binaire** Voir table 4.3

## 4.4 Les entités

**Nom :** GameEntity

**Type :** Ressource partagée

**Mode :** Interface et Implémentation

- Interface : [GameEntity] → [ab89e749-399d-f6b2-972d-9737e7b3a207]
- Implementation : [GameEntity] → [c2475906-4124-98ad-4077-daca4645492a]

**Attributs :**

- *\_RealName* (QString), nom effectif de la ressource
- *Semantics[]* (QString), liste des informations sémantiques de l'entité
- *Elements[]* (QUuid), UUID des éléments constitutifs

**Encodage binaire** Voir table 4.4

## 4.5 Les points d'apparition dans le monde

**Nom :** SpawnPoint

**Type :** Ressource

**Mode :** Implémentation

**Attributs :**

- *\_RealName* (QString), nom effectif de la ressource
- *World* (QUuid), UUID du monde
- *Location* (Vector3d), position dans le monde
- *Size* (float), taille de la zone (circulaire)
- *Semantics[]* (QString), liste des informations sémantiques

**Encodage binaire** Voir table 4.5

## 4.6 Les éléments constitutifs d'entités

**Nom :** GameEntityElement

**Type :** Ressource

**Mode :** Implémentation

**Attributs :**

- *Entity* (QUuid), UUID de l'entité
- *Position* (Vector3d), position de l'élément
- *Rotation* (Vector4d), rotation de l'élément
- *Scale* (Vector3d), mise à l'échelle de l'élément
- *Semantics[]* (QString), liste des informations sémantiques de l'élément
- *Resources[]* (QUuid), UUID des assets à utiliser

**Encodage binaire** Voir table 4.6



TABLE 4.1 Encodage binaire d'un monde

taille	type	contenu
--------	------	---------

TABLE 4.2 Encodage binaire d'une région du monde

taille	type	contenu
--------	------	---------

TABLE 4.3 Encodage binaire d'un objet du monde

taille	type	contenu
--------	------	---------

TABLE 4.4 Encodage binaire d'une entité du monde

taille	type	contenu
--------	------	---------

TABLE 4.5 Encodage binaire d'un point d'apparition

taille	type	contenu
--------	------	---------

TABLE 4.6 Encodage binaire d'un élément constitutif

taille	type	contenu
--------	------	---------

