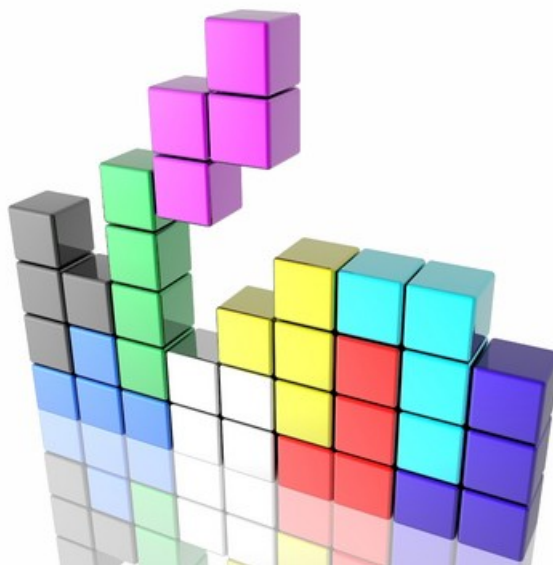


Solène QUATREVAUX  
Alexandre VERCRUYSSE  
Jonathan ARAMBURU  
Clément VAUTIER  
Renaud CREN

# PROJET 1A2A TETRIS 3D




# Introduction

Afin de valider notre DUT informatique, un projet nous a été attribué pour tester nos compétences en situation « réelle ». Celui que nous avons choisi est le Tetris 3D proposé par Monsieur Loudni.



réaliser le meilleur score.

D'abord, rappelons ce qu'est un Tetris. Il met le joueur au défi de réaliser des lignes complètes en déplaçant des pièces de formes différentes, les tetrominos, qui défilent depuis le haut jusqu'au bas de l'écran. Les lignes complétées disparaissent tout en rapportant des points et le joueur peut de nouveau remplir les cases libérées. Le jeu n'a pas de fin : le joueur perd la partie lorsqu'un tetrimino reste bloqué en haut. Il doit donc résister le plus longtemps à la chute continue des tetriminos, afin de

Ici, le défi est de réaliser ce même Tetris mais en incluant une profondeur tridimensionnelle. 


~~Ce dossier va donc nous présenter~~ le travail réalisé depuis l'attribution du projet jusqu'au rendu du 1<sup>er</sup> lot.  


## Table des matières


Introduction.....	2
I) Présentation du Projet.....	5
A- Demande du Commanditaire.....	5
B- Technologies Utilisées.....	6
Blender.....	6
JMonkey.....	6
TortoiseSVN.....	8
C- Charte de Projet.....	9
D- Backlog.....	10
E- Répartition des Tâches.....	11
II) Modélisation du Projet.....	12
A- Analyse.....	12
B- Choix des technologies.....	13
III) Réalisation.....	13
A- Conception.....	13
B- Recherches Effectuées et Problèmes Rencontrés.....	14
IV) État de l'Application et travail à venir.....	15
V) Conclusion.....	17

# I) Présentation du Projet

## A- Demande du Commanditaire


Suite à la réception du projet, nous avons contacté notre commanditaire afin de savoir quel allait être notre travail et quelles étaient ses ~~envies~~ concernant le ~~produit~~ que nous devons lui livrer. 



A l'issu de cette réunion ~~il est apparu que nous devons~~ réaliser une version amélioré du Tetris. ~~Cette amélioration porte sur l'ajout d'une dimension supplémentaire, la profondeur.~~ Cette profondeur rend le jeu plus complexe mais offre une plus grande liberté quant aux différents mouvements que le joueur peut appliquer aux pièces. 



Cette version en 3 dimensions implique l'utilisation d'un moteur ~~graphique~~. Par la volonté de notre commanditaire, nous nous sommes portés sur la bibliothèque ~~de fonctions~~ JMonkey afin de réaliser ce moteur 3D.

L'une des demandes secondaires de notre commanditaire était la portabilité sur Android. Suite à cette demande, le choix du langage de programmation fut évident. 

En effet, le langage le plus portable actuellement est Java. Notre choix se porta donc naturellement vers ce langage.


## B- Technologies Utilisées

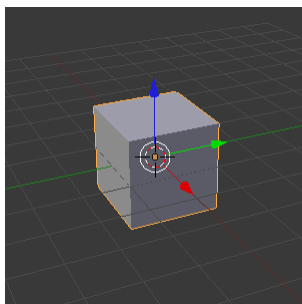


Afin de mener à bien ce projet, nous avons ~~du~~ apprendre à manipuler des objets en 3 dimensions à l'aide de 2 logiciels distincts : Blender et Jmonkey. De plus, afin de mieux répartir les tâches puis de rassembler les travaux de chacun, nous avons mis en place un serveur de gestion de version, TortoiseSVN.

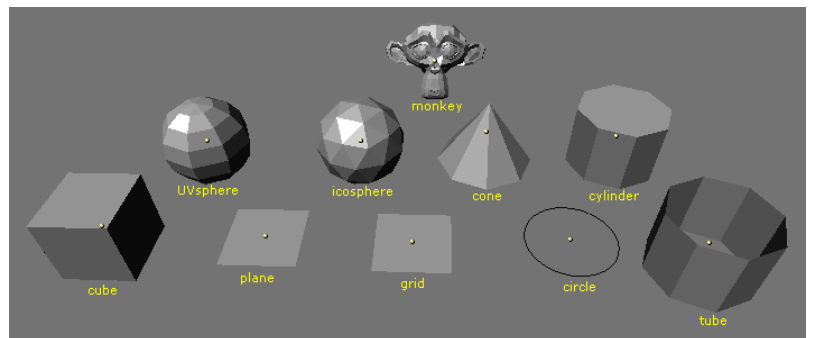
### Blender



Blender est un logiciel de modélisation en 3 dimensions. Il permet, en démarrant d'un simple cube, de concevoir une infinité de ~~forme~~ . C'est en utilisant ce logiciel que nous avons conçu les différentes pièces utilisées dans notre jeu.




Le cube servant de base à toutes formes Blender



Un aperçu des formes que l'on peut réaliser

### JMonkey



JMonkey est une bibliothèque ~~de fonctions utilisée par le langage de programmation~~  Java. ~~Elle permet de~~ générer, ~~de~~ déplacer et d'animer des objets en 3 dimensions. C'est grâce à cette bibliothèque que nous pouvons **gérer** les différentes pièces présentes dans l'espace de jeu de notre Tetris.



Lors de l'utilisation de JMonkey, on crée une application, qui va être ~~exécuter~~ au moment de son appel. Le code de cette application est décomposée en 2 parties principales : l'initialisation et la mise à jour.

L'initialisation s'effectue par la méthode `simpleInitApp()`. C'est dans cette fonction que l'on crée les différents objets que l'on veut afficher.

```
public void simpleInitApp() {  
    // create a blue box at coordinates (1,-1,1)  
    Box box1 = new Box( Vector3f.ZERO, 1f,2f,.5f);  
    blue = new Geometry("Box", box1);  
    Material mat1 = new Material(assetManager, "Common/MatDefs/Misc/Unshaded.j3md");  
    mat1.setColor("Color", ColorRGBA.Blue);  
    blue.setMaterial(mat1);  
    blue.move(0,0,0);  
}
```

Ici, on crée un objet 'blue', et on en fait un cube, en lui donnant des dimensions. On lui donne également une couleur avec `setColor(...)` et on le positionne dans l'espace, grâce à `move(...)`.

C'est en créant des objets dans cette méthode que l'on va pouvoir les afficher lors de l'ouverture de l'application.

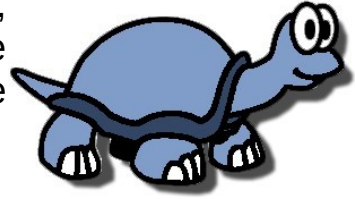
La mise à jour **consiste en le déplacement** des différents objets lors du fonctionnement de l'application. On utilisera pour cela la méthode `simpleUpdate()`.

```
public void simpleUpdate(float tpf) {  
  
    if(isRunning&&(red.getLocalTranslation().getY()>blue.getLocalTranslation().getY()+1)){  
        blue.setLocalTranslation(new Vector3f( 1f,1f,1f));  
    }  
}
```

Cette méthode est hérité de la bibliothèque JMonkey et elle constitue une boucle qui est répétée tant que l'application est active. C'est donc dans cette méthode que l'on va insérer les différents actions à effectuer répétitivement.

## TortoiseSVN

TortoiseSVN est un système de gestion de versions, c'est-à-dire que l'on peut y stocker des fichiers, que l'on peut mettre à jour tout en gardant sur le même serveur un accès ~~au version antérieure~~ du-dit fichier.



Ainsi, si une mise à jour du fichier créé des problèmes, on peut très facilement revenir à la version précédente ~~qui ne cause pas de problème~~.



Cela permet également de faire du partage de fichiers entre les différents membres du groupe de projets.

## C- Charte de Projet

### CONTEXTE

- M. Loudni veut un Tetris, mais en 3D.
- Celui-ci doit être réalisé en Java, avec la librairie JMonkey.
- La portabilité vers Android doit être possible.

### ENJEUX



- Maîtriser Jmonkey et Blender afin de pouvoir mettre au point un moteur 3D (destiné à l'animation de pièces ~~de~~ jeu) adapté à un Tetris en 3D.
- Mettre en place un moteur ~~de~~ jeu capable de gérer le bon déroulement d'une partie de Tetris.



### OBJECTIFS

- Réaliser une application logicielle destinée a jouer ~~a~~ Tetris mais en utilisant l'axe des Z afin de donner une profondeur au ~~gameplay~~.
- Doter cette application d'un système permettant de changer l'angle de vue afin de visualiser chaque face de l'espace de jeu

### JALONS DU PROJET

- 28/01/2015 : Attribution du projet
- 09/02/2015 : Rencontre avec le commanditaire  
Début du projet
- 10/06/2015 : Rendu du 1<sup>er</sup> Lot
- "/"/2015 : Rendu du Lot 2
- "/"/2016 : Rendu du Lot 3
- "/"/2016 : Rendu du Lot 4



Numéro	Description	Priorité	Module	Lot
2	Faire tomber une piece dans l'espace de jeu	1	Moteur du jeu	Lot 1
4	En tant qu'utilisateur, je peux naviguer dans les menus du jeu	2	Moteur du jeu	Lot 1
18	Réalisation des pièces sous Blender			Lot 1
1	En tant qu'utilisateur, je peux démarrer une nouvelle partie	1	Moteur du jeu	Lot 2
3	Effacer une ligne de l'espace de jeu lorsqu'elle est pleine	1	Moteur du jeu	Lot 2
5	En tant qu'utilisateur, je peux faire tourner la caméra autour du plateau	2	Moteur du jeu/Moteur Graphique	Lot 2
6	En tant qu'utilisateur, je peux déplacer une pièce en rotation dans l'espace de jeu	2	Moteur du jeu/Moteur Graphique	Lot 2
7	En tant qu'utilisateur, je peux déplacer une pièce en translation dans l'espace de jeu	2	Moteur du jeu/Moteur Graphique	Lot 2
10	En tant qu'utilisateur, je peux mettre le jeu en pause	3	Moteur du jeu	Lot 2
17	Réalisation du plateau sous Blender			Lot 2
8	En tant qu'utilisateur, je peux sauvegarder une partie	3	Moteur du jeu	Lot 3
9	En tant qu'utilisateur, je peux charger une partie sauvegardée	3	Moteur du jeu	Lot 3
11	Augmenter la vitesse de descente des pièces au fur et à mesure des niveaux	3	Moteur du jeu	Lot 3
12	En tant qu'utilisateur, je peux connaître la prochaine pièce qui va tomber	3	Interface de jeu	Lot 3
13	En tant qu'utilisateur, je peux visualiser le niveau auquel je me trouve	4	Interface de jeu	Lot 3
14	En tant qu'utilisateur, je peux visualiser le temps qui s'est écoulé depuis le début de la partie	4	Interface de jeu	Lot 3
15	En tant qu'utilisateur, je peux connaître mon score actuel	4	Interface de jeu	Lot 3
16	En tant qu'utilisateur, je peux connaître le high score	4	Interface de jeu	Lot 3


D- Backlog


## E- Répartition des Tâches

Afin ~~de mener à réaliser~~ dans les temps l'application que nous devons fournir à notre commanditaire, nous avons ~~subdiviser~~ le travail à faire en 3 grandes parties : la conception des pièces, la conception du moteur du jeu et la conception du moteur 3D.

La conception des pièces a pour but de créer avec Blender les différentes pièces de jeu qui seront plus tard implantées dans le Tetris. Cette tâche ne nécessitant que peu de conception elle a été ~~confié a~~ une seule personne.



La conception du moteur du jeu a pour but de mettre au point la partie du programme qui a pour ~~but~~ de gérer ~~la relation avec le joueur et la répétition de la boucle de jeu~~. 2 personnes furent affectées à cette tâche afin de  rechercher quelles seraient les possibilités de connection entre le moteur 3D et le moteur de gestion du jeu.

Le moteur 3D est la partie de l'application qui sera utilisée par le moteur du jeu afin de manipuler les différentes pièces dans l'espace de jeu. C'est essentiellement sur cette partie que le travail a été concentré sur ~~cette~~ première itération. 2 personnes furent affectées à cette tâche. 

## II) Modélisation du Projet

### A- Analyse

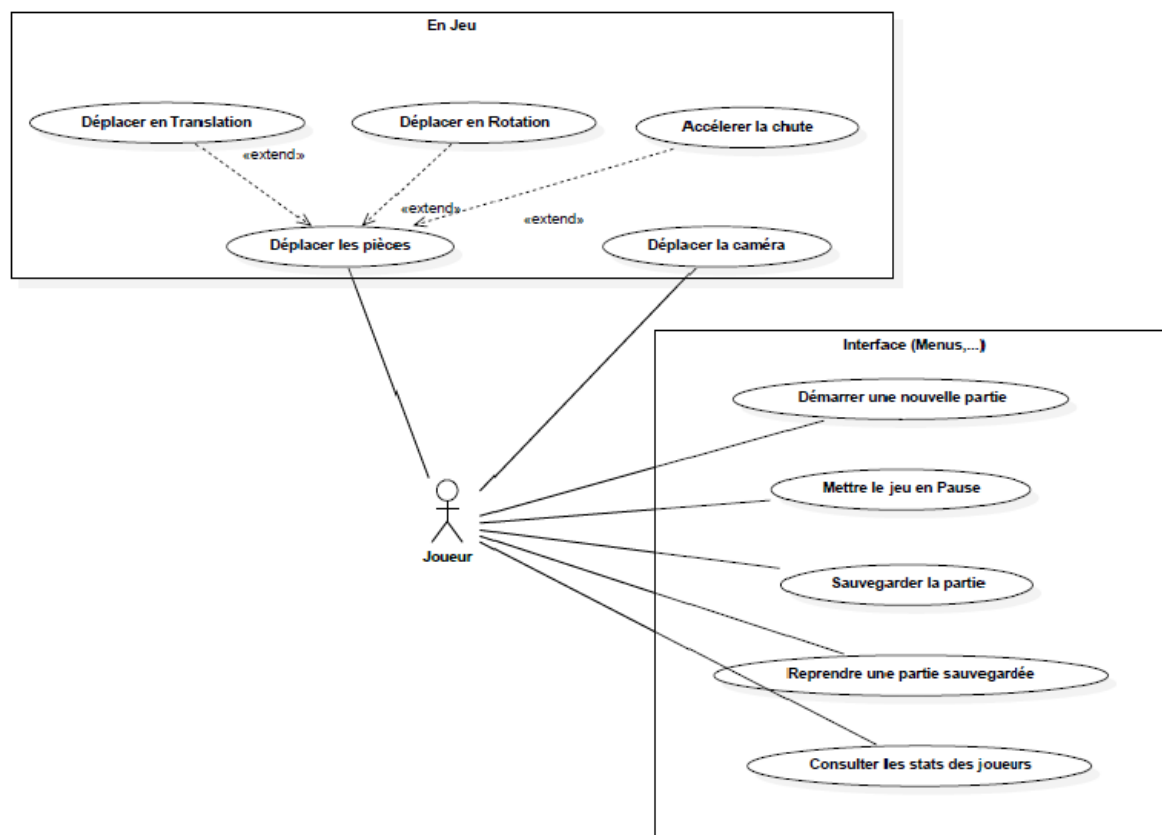
Afin de mieux visualiser comment le joueur interagira avec le programme, nous avons créé un diagramme de cas d'utilisation, regroupant les principales actions que l'utilisateur pourra effectuer.

Ces actions ont été réparties en 2 blocs distincts : le bloc Interface et le bloc Jeu.

Le bloc Interface regroupe les actions que l'on peut effectuer au travers des différentes interface du jeu, du menu principal, permettant de démarrer une nouvelle partie ou de consulter les stats des joueurs, au menu en jeu permettant de sauvegarder sa partie par exemple.

Le bloc Jeu regroupe les actions gérées par le moteur du jeu et qui influent sur le déroulement de celui-ci. On y trouvera les déplacements des pièces et ceux des caméras.

Ce découpage permet de définir deux catégories de fonctions qui seront implémentées dans deux endroits distincts dans le code.



## ~~B- Choix des technologies~~

~~Le Langage retenu pour le développement de l'application est le JAVA ; C'est ce langage qui a été choisi pour 2 raisons :~~

~~Notre commanditaire nous a demandé une application dans ce langage~~

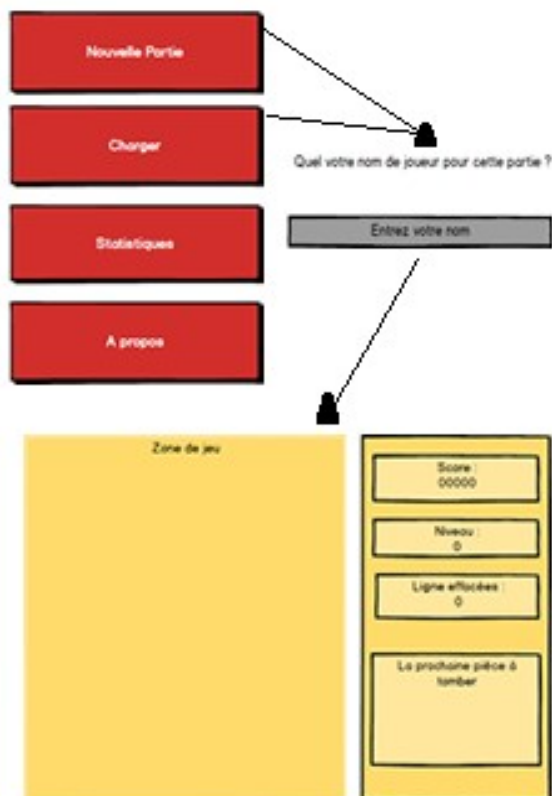
~~Le souhait émis par notre commanditaire que l'application puisse être portée sur Android~~

~~Nous utiliserons en ajout à ce langage la bibliothèque JMonkey afin de gérer les modèles 3D.~~

## III) Réalisation

### ~~A- Conception~~

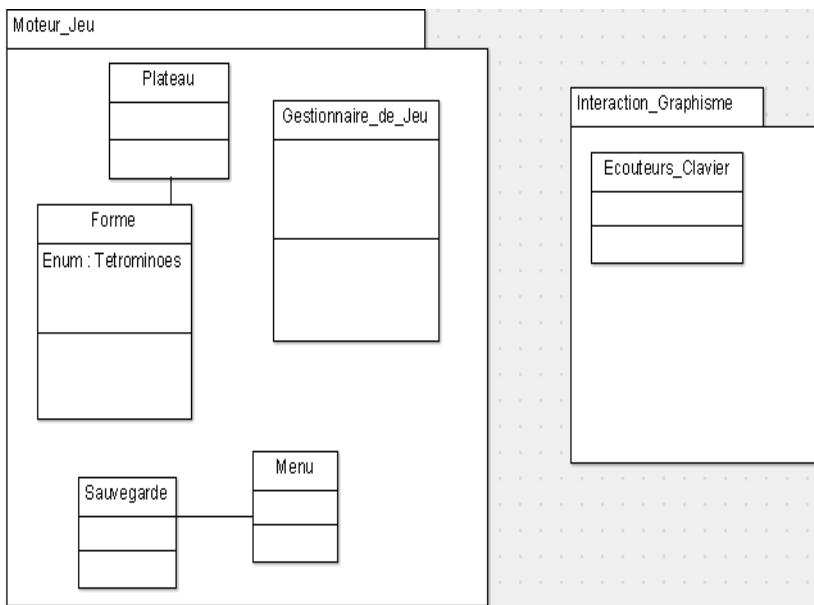
Voici ci-dessous une maquette de l'interface de notre application :



En rouge : les différents boutons  
En gris : zone de texte à entrer  
En jaune : Zone d'information ou jeu

La maquette est encore très simple car nous n'avons pas d'idée précise pour l'interface.

Par conséquent, on y trouve les bases qui seront inchangés.



Tout comme la maquette, le diagramme de classe reste plutôt vague, en effet, nous avons préféré nous concentrer sur l'apprentissage de l'utilisation des outils et la réalisation du 1er lot qui ne nécessitait pas un diagramme de classe

## B- Recherches Effectuées et Problèmes Rencontrés

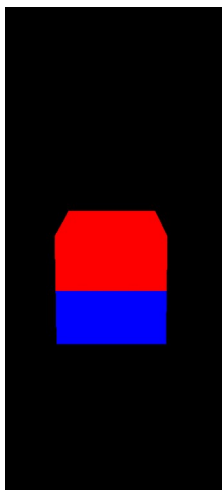
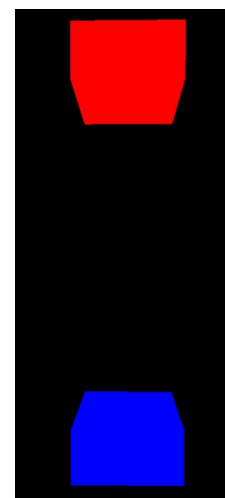
L'utilisation de nouveaux logiciels demandés, nous a pris des heures d'apprentissage pour nous habituer, nous n'avons pour le moment qu'étudier qu'une partie des fonctionnalités disponibles, assez pour avancer notre projet. Mais on s'attend à rencontrer des problèmes dues à la prise en main de ces logiciels ( SVN, Jmonkey, Blender ). A ce moment, nous approfondirons l'utilisation de ces logiciels. Notre problème majeur fut de se lancer, nous ne savions pas qu'est ce qui devait être fait en premier, la partie motrice ou la partie graphique de l'application. Après avoir pu commencer, nous nous sommes confrontés à un premier problème de prise en main de Jmonkey. Nous ignorions que Jmonkey travaillait en 2 temps, une partie initialisation puis une partie de mise à jour, au même propriété qu'une boucle qui se répète tant que l'application est active. Pour pouvoir progresser efficacement dans notre projet, nous avons cherché des codes java d'applications Tetris, qu'il a fallu étudier en profondeur pour comprendre comment notre projet s'exécutera et en garder les points nécessaires que nous pourrons adapter en 3D.

## IV) État de l'Application et travail à venir

Dans le cadre de la première itération du projet, nous devons fournir à notre commanditaire une application permettant de visualiser la chute d'un cube simple dans l'espace de jeu. Ce cube doit s'immobiliser lorsqu'il rencontre un second cube, servant ici à simuler le plateau de jeu de l'application finale.

Nous sommes à présent en mesure de délivrer l'application demandée à notre commanditaire. Cette partie a pour but de décrire les fonctionnalités implémentées dans le rendu.

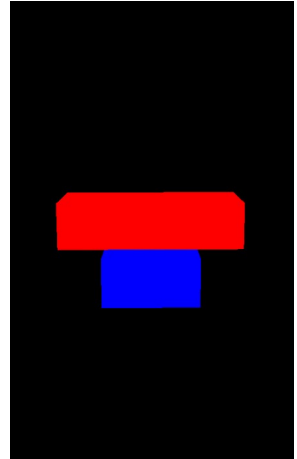
Conformément à la demande, nous sommes capable de faire descendre le cube dans l'espace de jeu. Dans la capture ci-contre, la forme bleue est fixe et sert de point d'arrêt pour le cube rouge, qui lui est en mouvement



Nous sommes également en mesure d'immobiliser la pièce rouge lorsqu'elle arrive en contact avec la pièce bleue.

Nous avons cependant réussi à implémenter 2 fonctions supplémentaires qui nous seront très utiles lorsque nous serons à un stade plus avancé du projet. Pour la première, il s'agit d'une fonction «pause» permettant d'interrompre la descente du cube rouge à n'importe quel moment, et de la reprendre par la suite, le tout par une simple pression sur la touche «espace».

La seconde fonction ajoutée permet d'effectuer une rotation de 90° sur la pièce rouge.



Par la suite, tout cela nous permettra d'avoir des bases afin de mettre au point des fonctions plus avancées applicables au jeu de Tetris en 3D. Le principal point de difficulté à venir est l'implémentation dans l'espace de jeu de pièces conçues sous Blender. Cet import ainsi que la manipulation de ces pièces de la même manière que le cube de l'application du Lot 1 constituera les difficultés majeures du prochain lot.

## V) Conclusion

Dans le cadre de la production du Lot 1 de notre projet 1A2A, nous devions rendre une visualisation en 3 dimensions de la chute d'une forme parallélépipédique, ainsi que l'immobilisation de cette dernière sur une surface représentant le plateau de jeu du Tetris.

Malgré les quelques problèmes rencontrés décrits précédemment, nous avons réussi à mener à bien notre objectif, et de fournir un premier aperçu de notre travail à notre commanditaire.

Cette première itération nous a permis de nous familiariser avec la méthode agile de gestion de projet. En effet, même si dans notre parcours d'étude nous avons déjà eu à réaliser différents projets, celui-ci est le premier qui nous confronte à cette nouvelle façon de travailler.

Nous avons également pu profiter de cette période pour nous approprier l'utilisation des différentes technologies décrites précédemment, de telle sorte que, dans l'avenir, nous puissions être à l'aise avec la manipulation de ces outils et que les différentes tâches soit plus faciles à réaliser.