

# ELKCRAFT

Capstone Project

**Adrien GIVRY**

**Charly JEAUC**

**Saad RAOUF**

**Hanseul SHIN**

PGP: 2017/2018



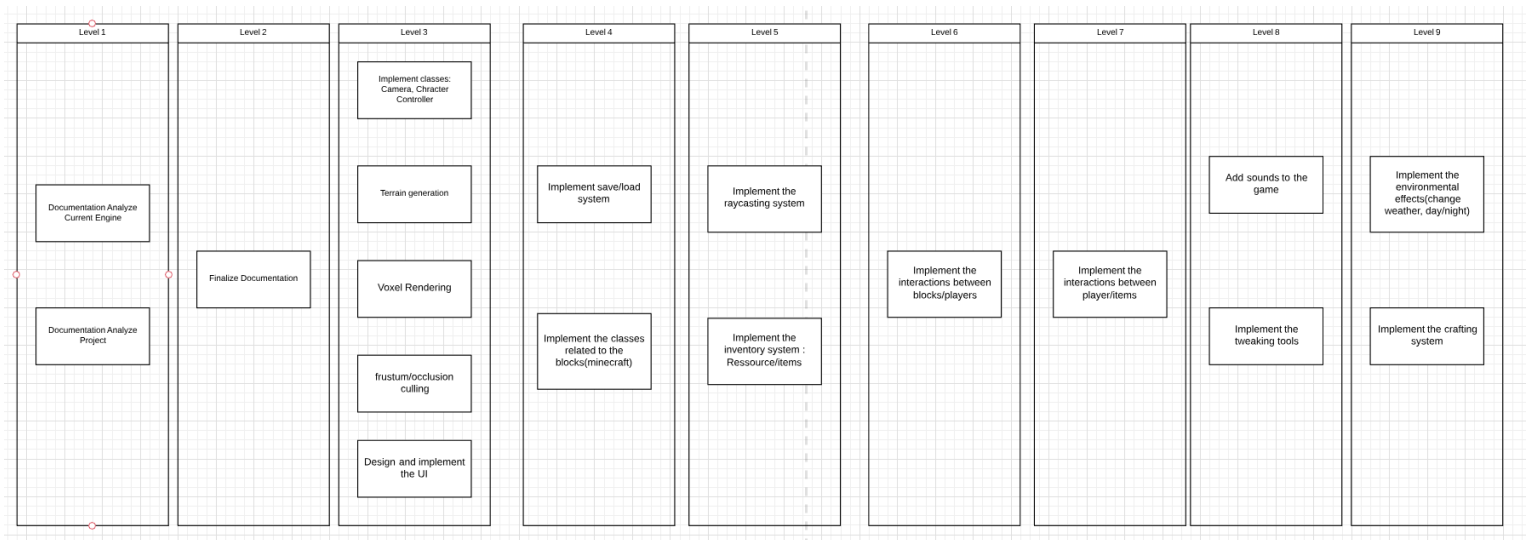
## SOMMAIRE

1. Tâches et responsabilités .....	4
2. État DU PROJET AU DEPART .....	5
2.1. ElkGameEngine.....	5
2.1.1. Diagramme de classes .....	5
2.1.2. Changements identifiés.....	5
2.2. ElkPhysics.....	6
2.2.1. Diagramme de classes .....	6
2.2.2. Changements identifiés.....	6
2.3. ElkRendering.....	7
2.3.1. Diagramme de classes .....	7
2.3.2. Changements identifiés.....	7
3. Création d'ElkCraft .....	8
3.1. Diagramme de classe proposé .....	8
3.2. Problématiques anticipées .....	8
4. État du projet à la GOLD.....	9
4.1. ElkEngine .....	9
4.1.1. Diagramme de packages .....	9
4.1.2. Modifications apportées .....	9
4.2. ElkGameEngine.....	10
4.2.1. Diagramme de classes .....	10
4.2.2. Modifications apportées .....	10
4.3. ElkRendering.....	11
4.3.1. Diagramme de classes .....	11
4.3.2. Modifications apportées .....	11
4.4. ElkPhysics.....	12
4.4.1. Diagramme de classes .....	12
4.4.2. Modifications apportées .....	12
4.5. ElkTools.....	13
4.5.1. Diagramme de classes .....	13
4.5.2. Modifications apportées .....	13

4.6.	ElkCraft .....	14
4.6.1.	Diagramme de classes .....	14
4.6.2.	Fonctionnalités implémentées .....	14
5.	Contenu final du projet .....	15
5.1.	Le « Player Controller » .....	15
5.2.	Visée et surbrillance de la sélection .....	16
5.3.	Interaction avec l'environnement .....	17
5.4.	Inventaire .....	18
5.5.	Santé et énergie .....	19
5.6.	Création d'objets .....	20
5.7.	Le cycle jour/nuit .....	21
5.8.	Système de sauvegarde .....	22
5.9.	Physique .....	23
5.10.	Système audio .....	24
5.11.	Animation .....	25
5.12.	Lumières .....	26
5.13.	Optimisation du rendu .....	27
6.	Contrôles .....	28
6.1.	Menu principal .....	28
6.2.	En jeu .....	28
6.3.	Mouvement .....	28
6.4.	Inventaire .....	28
6.5.	Interaction avec l'environnement .....	28
6.6.	Système de création d'objets .....	28
6.7.	Globaux .....	28
7.	Configuration .....	28

# 1. TÂCHES ET RESPONSABILITÉS

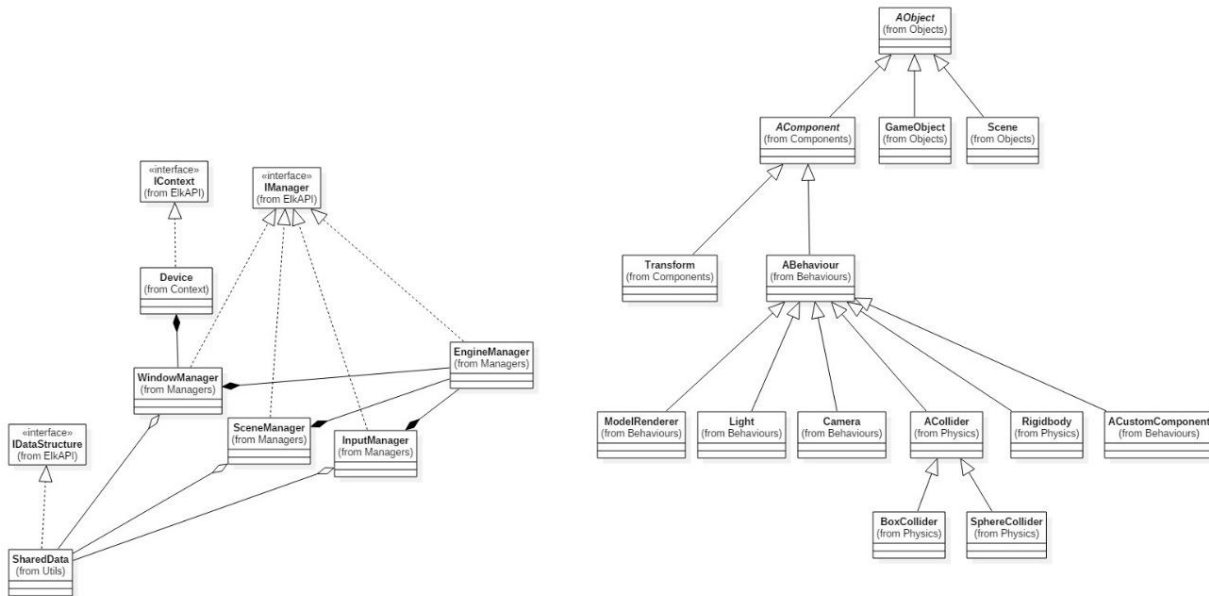
Name ▾	Previous ▾	Next ▾	Description ▾	Duration ▾	Resources ▾	Assigned to ▾
A	-	B	Documentation part 1 : Analyse the engine	1	2	
B	A	C	Documentation part 2 : Analyse the project	1	2	
C	B	D	Finalize the documentation	1	1	
D	C	J	Implement the classes related to player : camera, character controller ...	2	1	
E	C	-	Implement the algorithm related to the terrain generation	4	2	
F	C	-	Implement the algorithm related to the rendering (to optimize the framerate)	cont	3	
G	C	-	Design and implement the UI	10	1	
H	C	-	Implement the tweaking tools	4	1	
I	C	J	Implement the classes related to the blocks(minecraft)	4	1	
J	D,I,O		Implement the interactions between blocks/players/items	10	1	
K	O	Q	Add sounds to the game	2	1	
L	Bonus	Q	Implement the environmental effects(change weather, day/night)	Bonus	1	
M	O	Q	Implement save/load system	cont	2	
N	C	-	Implement the raycasting system	2	1	
O	G	J	Implement the inventory system : Ressource/items	10	2	
P	Bonus	Q	Implement the crafting system	Bonus	2	
Q	-	-	Finalize the game	cont	4	



## 2. ÉTAT DU PROJET AU DEPART

### 2.1. ElkGameEngine

#### 2.1.1. Diagramme de classes

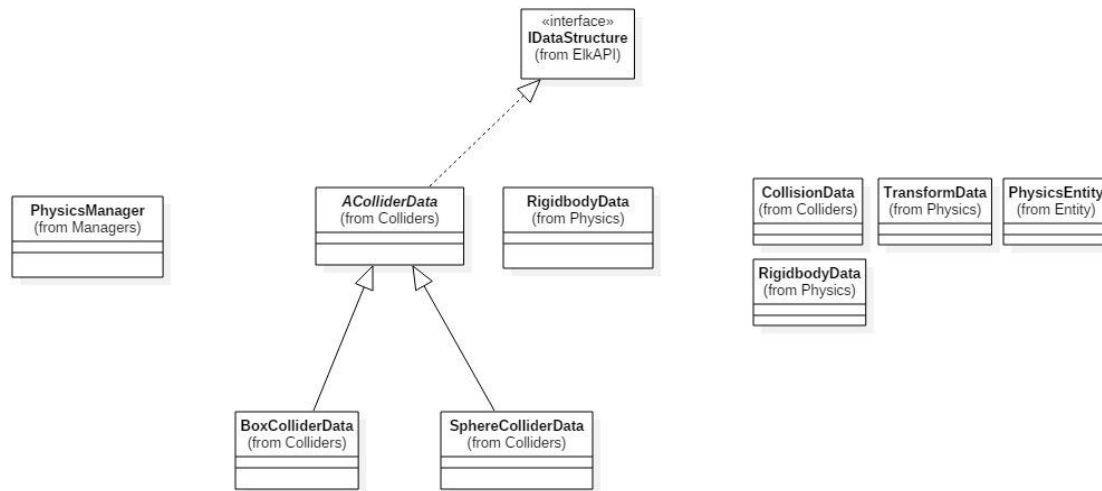


#### 2.1.2. Changements identifiés

- Supprimer les colliders et garder les box-colliders
- Ajouter les chunks, les blocs et le voxel
- Player Controller
- Système d'inventaire
- Système du Crafting
- Génération de terrain
- Sauvegarde et chargement
- UI

## 2.2. ElkPhysics

### 2.2.1. Diagramme de classes

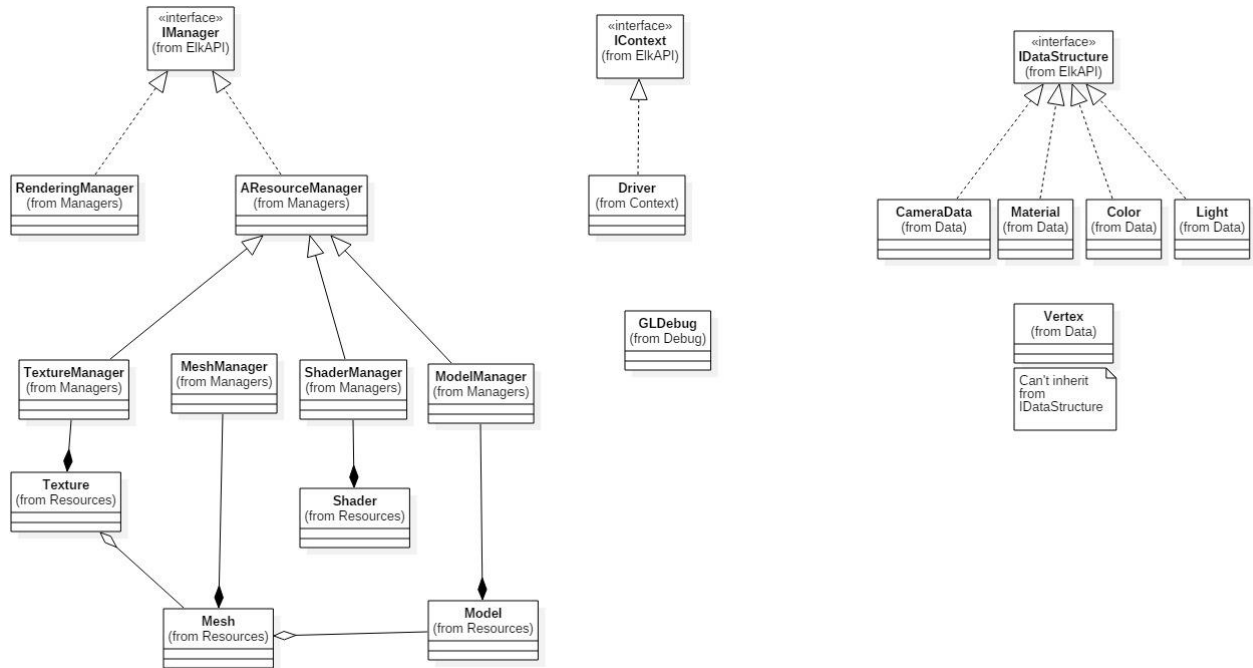


### 2.2.2. Changements identifiés

- Supprimer les colliders et garder les box-colliders
- Simplifier les collisions

## 2.3. ElkRendering

### 2.3.1. Diagramme de classes

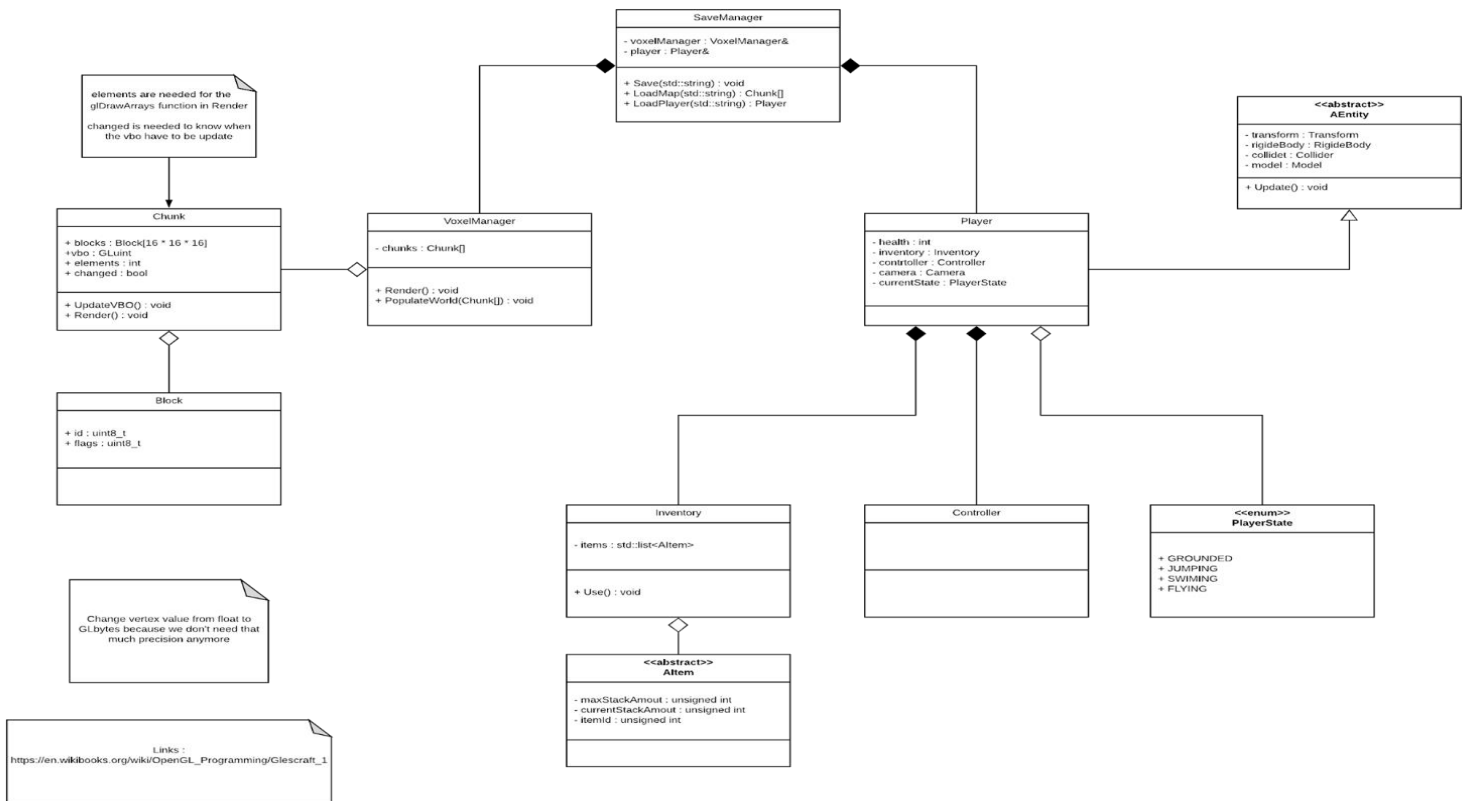


### 2.3.2. Changements identifiés

- Capacité à dessiner des voxels à partir d'une position
- Optimiser Rendering (frustum culling / occlusion culling)
- Render Éléments d'interface utilisateur

### 3. CRÉATION D'ELKCRAFT

#### 3.1. Diagramme de classe proposé



#### 3.2. Problématiques anticipées

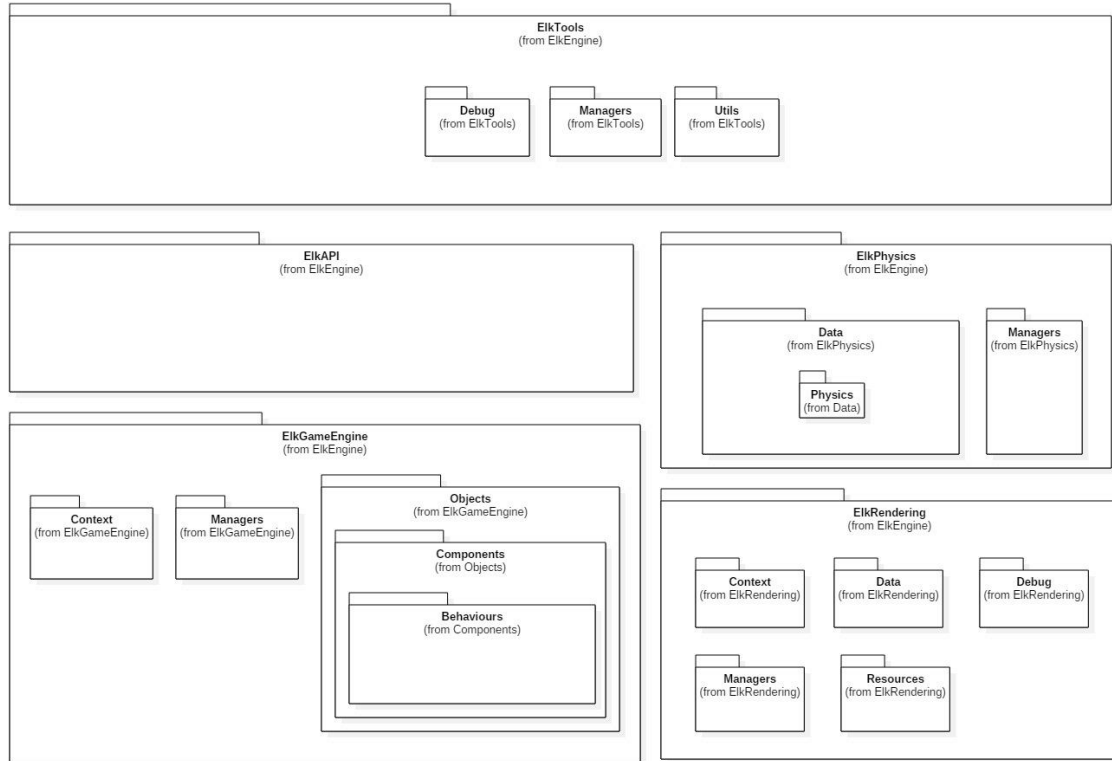
- Faire tomber des Items
- Interaction du bloc de gravier



## 4. ÉTAT DU PROJET À LA GOLD

### 4.1. ElkEngine

#### 4.1.1. Diagramme de packages

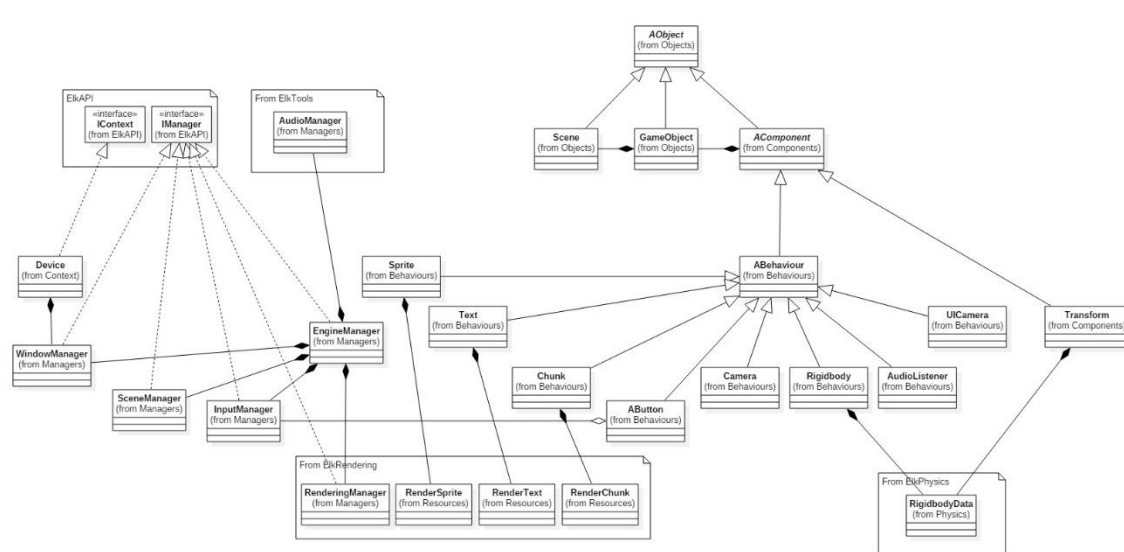


#### 4.1.2. Modifications apportées

- De nouveaux espaces de noms ont vu le jour

## 4.2. ElkGameEngine

### 4.2.1. Diagramme de classes

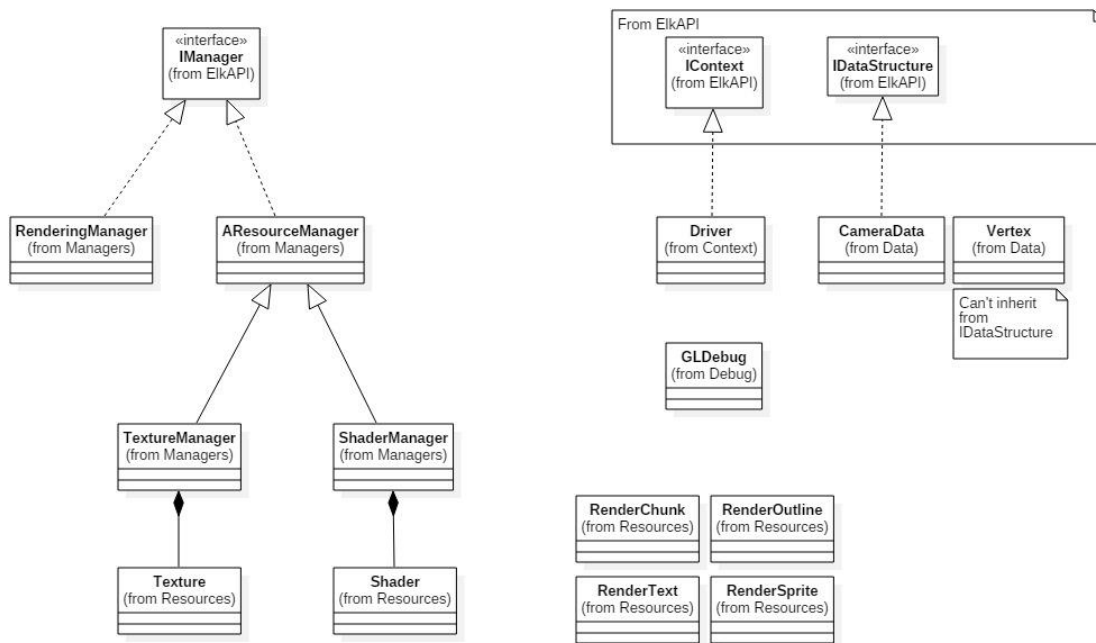


### 4.2.2. Modifications apportées

- La relation parent/enfant a été supprimée
- Nouvelle gestion des scènes
- Ajout de « Components » pour les « GameObjects »
- Suppression des « Components » inutiles (Lights, Colliders, ModelRenderer...)
- Refonte du « Parsing » de scène pour prendre en compte les nouveaux « Components »

## 4.3. ElkRendering

### 4.3.1. Diagramme de classes

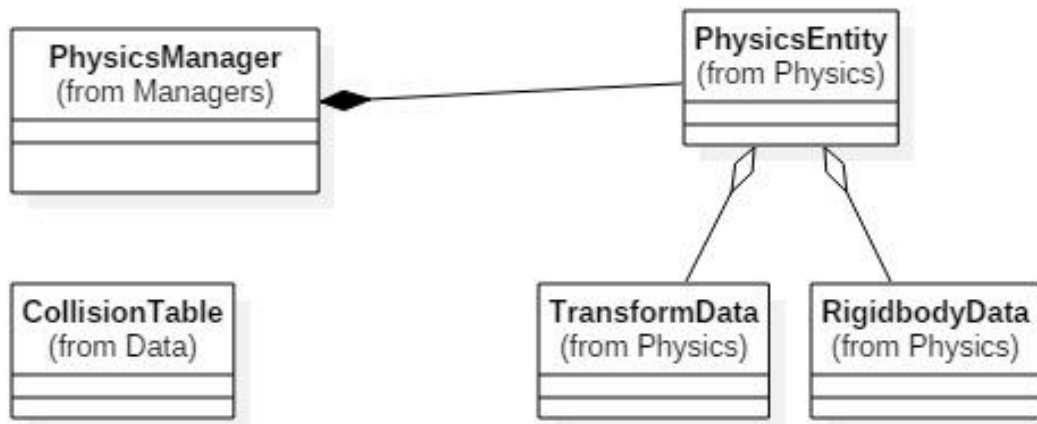


### 4.3.2. Modifications apportées

- Ajout des « RenderChunk », « RenderSprite » et « RenderText »
- Gestion d'une seconde caméra spécifique à l'interface utilisateur
- Textures adaptées à un jeu style « Minecraft »

## 4.4. ElkPhysics

### 4.4.1. Diagramme de classes

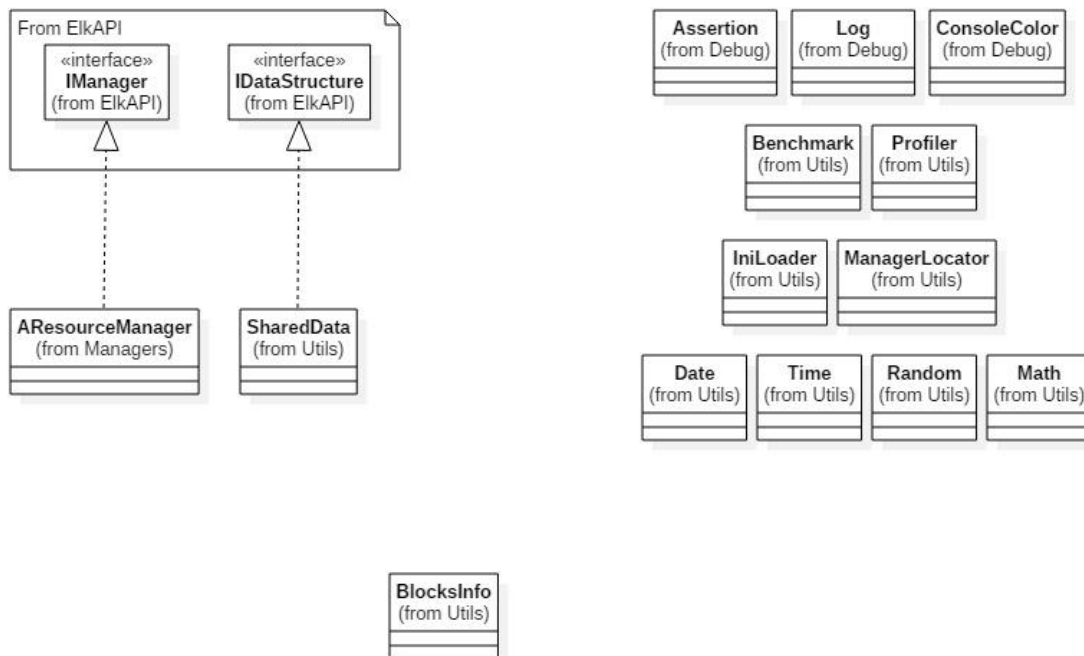


### 4.4.2. Modifications apportées

- Refonte intégrale du système de gestion de collision
- Ajout de la « CollisionTable » qui permet à la partie physique d'avoir connaissance du monde
- Suppression des colliders

## 4.5. ElkTools

### 4.5.1. Diagramme de classes

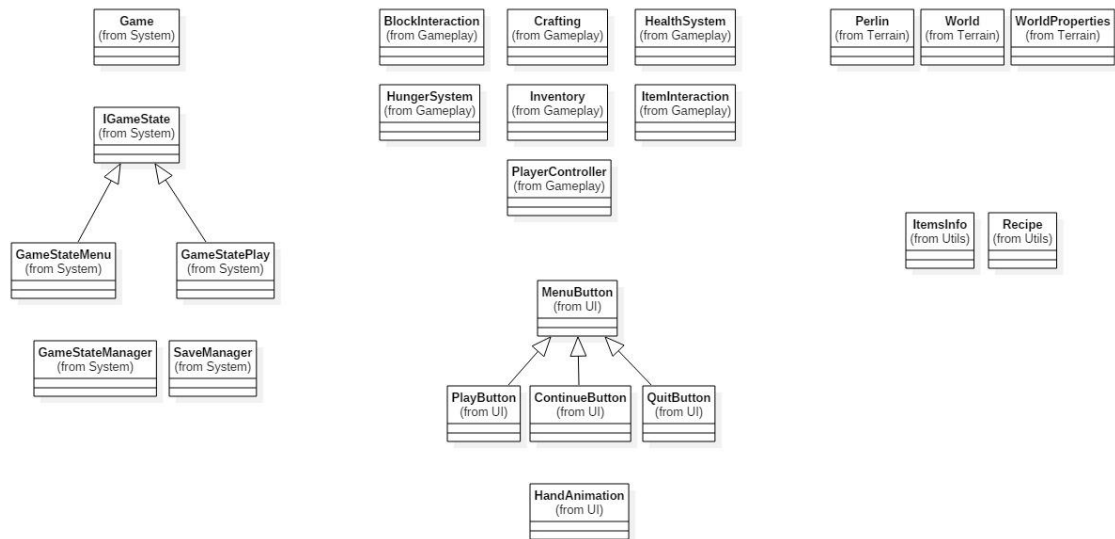


### 4.5.2. Modifications apportées

- Création du « ManagerLocator » (Service Locator design pattern)
- Ajout de fonctionnalités pour le profiler
- Générateur de nombres aléatoires
- Informations sur les blocs

## 4.6. ElkCraft

### 4.6.1. Diagramme de classes



### 4.6.2. Fonctionnalités implémentées

- Génération du monde
- Système de sauvegarde
- Menu principal
- Interaction avec les blocs
- Inventaire
- Création d'objets
- Mouvement du joueur
- Système de santé et d'énergie

## 5. CONTENU FINAL DU PROJET

### 5.1. Le « Player Controller »

La caméra est à la première personne. Le joueur peut se déplacer avec les touches du clavier et orienter sa vision avec la souris.



## 5.2. Visée et surbrillance de la sélection

Lorsque le joueur déplace la caméra un « raycast » est lancé depuis la position du joueur dans la direction de son regard. Celui-ci permet de déterminer le bloc que le joueur est en train de regarder. Un contour noir est affiché sur le bloc trouvé ce qui permet au joueur d'avoir un retour sur sa visée.





### 5.3. Interaction avec l'environnement

Le joueur peut détruire et poser des blocs. Lors de la destruction d'un bloc, une texture de brisure est affichée sur celui-ci. Elle permet d'indiquer au joueur le niveau de destruction actuel du bloc (De 0 à 10).

La durée de destruction d'un bloc dépend du type de bloc ainsi que de l'outil du joueur.

Le résultat de la destruction d'un bloc dépend également du type de bloc ainsi que de l'outil du joueur. Certains blocs (Comme les minerais de charbon) ne donnent aucun résultat s'ils sont détruits à la main. D'autres blocs, comme les feuilles, ont un taux de chance de donner un résultat.

Le joueur peut également poser les blocs présents dans son inventaire.



## 5.4. Inventaire

Le joueur possède un inventaire composé de 10 emplacements. Chaque emplacement peut contenir un type d'objet unique. La quantité maximale que le joueur peut stocker dans un emplacement est de 64. Il est nécessaire de gérer correctement son inventaire pour ne pas perdre le résultat de la destruction d'un bloc/objet du monde.



## 5.5. Santé et énergie

Plus le temps passe, plus votre personnage va avoir faim.



Si son énergie (Représentée par des morceaux de viande) est vide, alors sa santé diminuera jusqu'à un demi cœur. La santé du joueur se régénère avec le temps, dépendamment de son niveau d'énergie.



Si le joueur possède 3 points d'énergie complet (6 demi morceaux de viande) ou moins, il ne pourra plus courir.

Afin de regagner de l'énergie, le joueur devra trouver de la nourriture. Présentement le joueur peut récupérer de l'énergie en mangeant des pommes qu'il pourra trouver dans les arbres.



## 5.6. Création d'objets

Le système de création d'objet fonctionne par le biais de recettes. Le menu de « crafting » permet au joueur de sélectionner une recette et de transformer les objets de son inventaire (Composants) en nouvel objet (Résultat).

Les composants nécessaires à la création d'un nouvel objet sont affichés à gauche. Le résultat de la recette est affiché à droite.

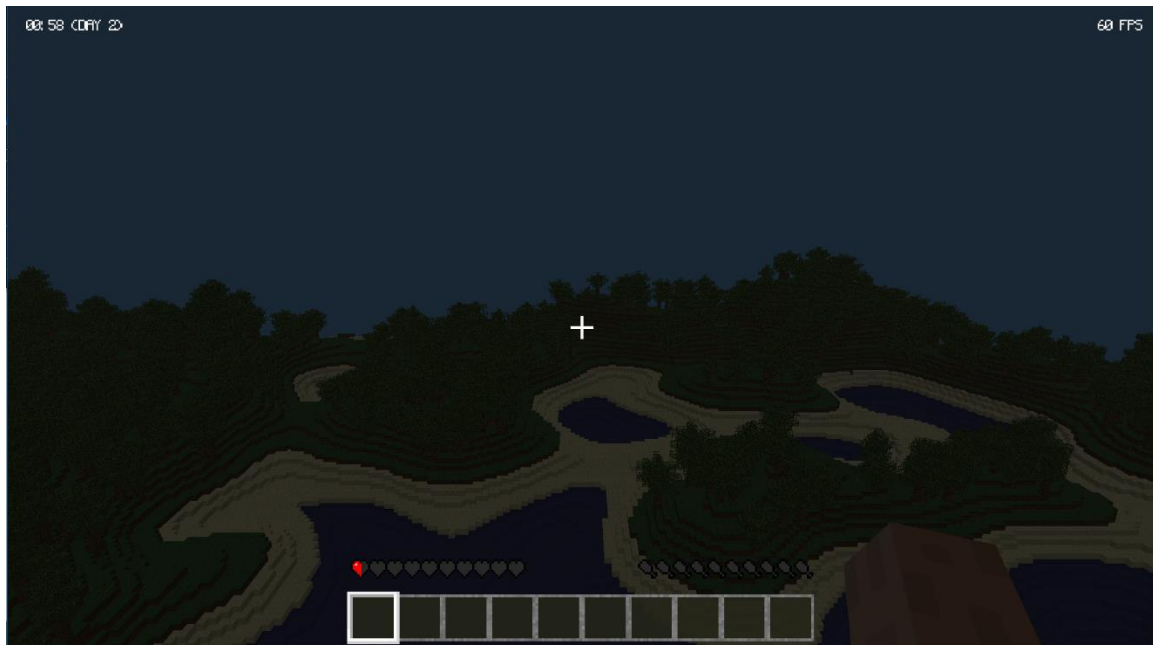
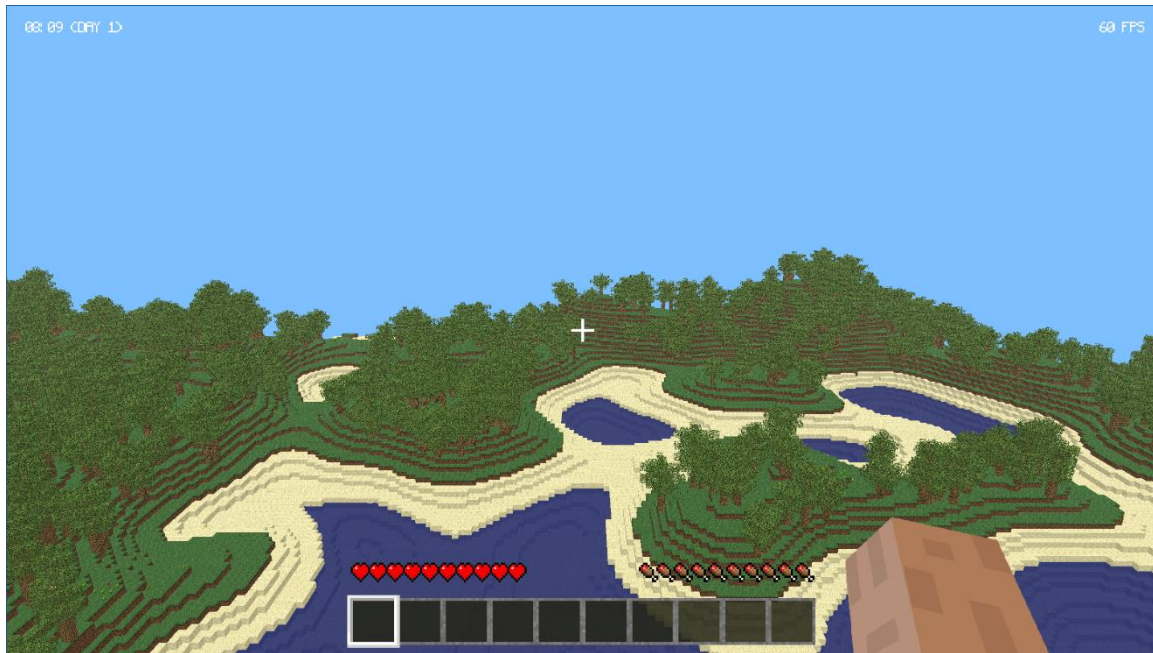




## 5.7. Le cycle jour/nuit

Le cycle jour/nuit applique des ombres sur les blocs en fonction de l'heure de la journée. Nous avons ajouté un affichage de l'heure au format 24h sur l'écran ainsi qu'un compteur de jour depuis le début de la partie. Le joueur peut facilement vérifier l'arrivée du jour et de la nuit. Le niveau de lumière appliqué au bloc ainsi qu'à la skybox est calculé grâce à la formule suivante :

$$\begin{aligned} \text{lightIntensity} = & 0.2 + 0.00002 * x * (x - 300) + (-3.7037037037036e - 8) * x * (x - 300) \\ & * (x - 400) + 3.7037037037036e - 11 * x * (x - 300) * (x - 400) * (x \\ & - 900) + (-1.1223344556677888e - 14) * x * (x - 300) * (x - 400) * (x - 900) \\ & * (x - 1000) \end{aligned}$$



## 5.8. Système de sauvegarde

Le joueur peut retrouver sa progression lorsqu'il relance le jeu grâce à un système de sauvegarde. Celui-ci sérialise au format binaire toutes les données du jeu (Inventaire, position, blocs du monde, etc...) et les utilise pour générer une partie lorsque le bouton « Load » du menu principal est appuyé. Si une nouvelle partie est créée, un nouveau fichier de sauvegarde est généré. Il est possible de stocker jusqu'à 256 sauvegardes différentes. Le fichier « save.info » permet d'indiquer au jeu quel sauvegarde celui-ci doit charger lorsque le bouton « Load » est appuyé.

Total block size :

11 Bytes	Header Block			
	Offset	Size (Bytes)	Type / Contents	Description
	0	3	uint8_t[3]	signature ("sav")
	3	1	uint8_t	number of Blocks
	4	1	uint8_t	size of Block header
	5	1	uint8_t	size of Block map gen
	6	1	uint8_t	size of Block player data
	7	4	uint32_t	size of Block chunk data

22 Bytes	Map Generation Block			
	Offset	Size (Bytes)	Type / Contents	Description
	0	4	float	noise redistribution
	4	4	float	noise persistence
	8	4	float	noise lacunarity
	12	4	float	noise scale
	16	2	uint16_t	map seed
	18	1	uint8_t	number of octave
	19	1	uint8_t	noise max height
	20	1	uint8_t	noise additional elevation
	21	1	uint8_t	water level

52 Bytes	Player Datas Block			
	Offset	Size (Bytes)	Type / Contents	Description
	0	12	float[3]	player position
	12	12	float[3]	player forward
	24	20	uint8_t[20]	type of block + amount
	44	1	uint8_t	player health
	45	1	uint8_t	player energy
	46	4	uint32_t	days counter
	50	2	uint16_t	day time

s * 4108 + 4 Bytes	Chunk Datas Block			
	Offset	Size (Bytes)	Type / Contents	Description
	0	4	uint32_t	number of chunks
	4	s * 4108	s * Chunk	Chunk position and content

Chunk

float [3]  
uint\_8 [4096]  
Total size : 4096 + 3 \* 4 = 4108 bytes

## 5.9. Physique

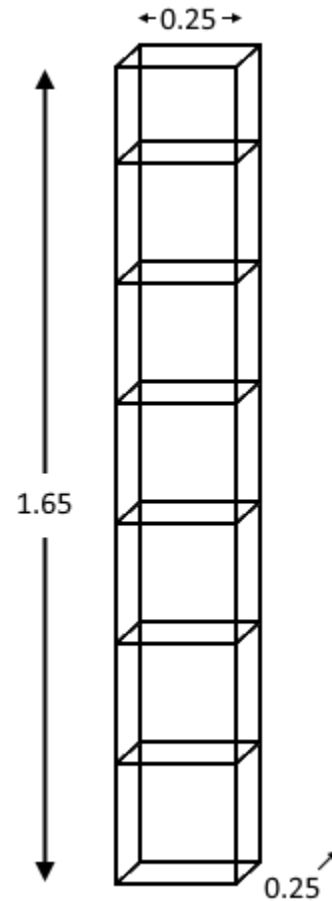
Le joueur peut se déplacer librement dans le monde tout en étant soumis aux lois de la physique.  
Le joueur est soumis à la gravité et ne peut plus avancer s'il rencontre un bloc.

Le joueur peut être représenté de manière physique par un parallélépipède

Le joueur ne peut sauter qu'une unité de bloc.

La vitesse du joueur est plus lente s'il entre dans un milieu liquide (Eau ou lave).

Le joueur subit des dégâts de chute proportionnel à la distance parcourue.



### 5.10. Système audio

Un ensemble de sons sont joués en fonction des actions du joueur et de l'environnement. Les sons intégrés au jeu sont les suivant :

- Marche (Dépend du bloc sur lequel le joueur marche)
- Placement de bloc (Dépend du bloc)
- Destruction de bloc (Dépend du bloc)
- Contact de l'eau
- Clic sur un bouton du menu
- Création d'objet réussie
- Création d'objet échouée

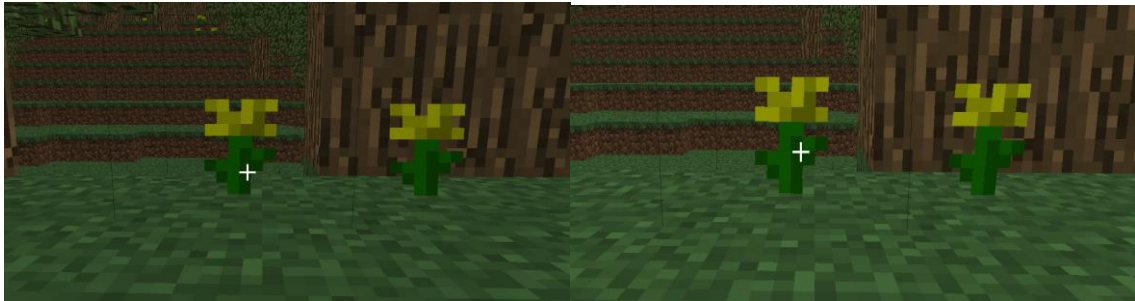
Les sons sont localisés dans l'espace. Plus le joueur se trouve proche de la source sonore, plus l'intensité du son joué sera importante.

Une musique de fond est également jouée au lancement d'une partie et à l'écran titre.



### 5.11. Animation

Plusieurs animations ont été intégrés au jeu afin de rendre l'expérience de jeu plus dynamique et vivante. Ces animations sont pour la plupart effectuées coté carte graphique (Vertex Shader) à l'exception de la main du joueur qui est déplacée par le processeur.



Animation des fleurs



Animation de la main du joueur ainsi que des feuilles



Animation de l'eau

## 5.12. Lumières

Chaque bloc possède un niveau d'intensité lumineuse variant entre 0 et 15.

La couleur de chaque bloc est multipliée par un coefficient (Entre 0 et 1) représentant le niveau d'obscurité du bloc. Ce coefficient peut se calculer grâce à la formule suivante :

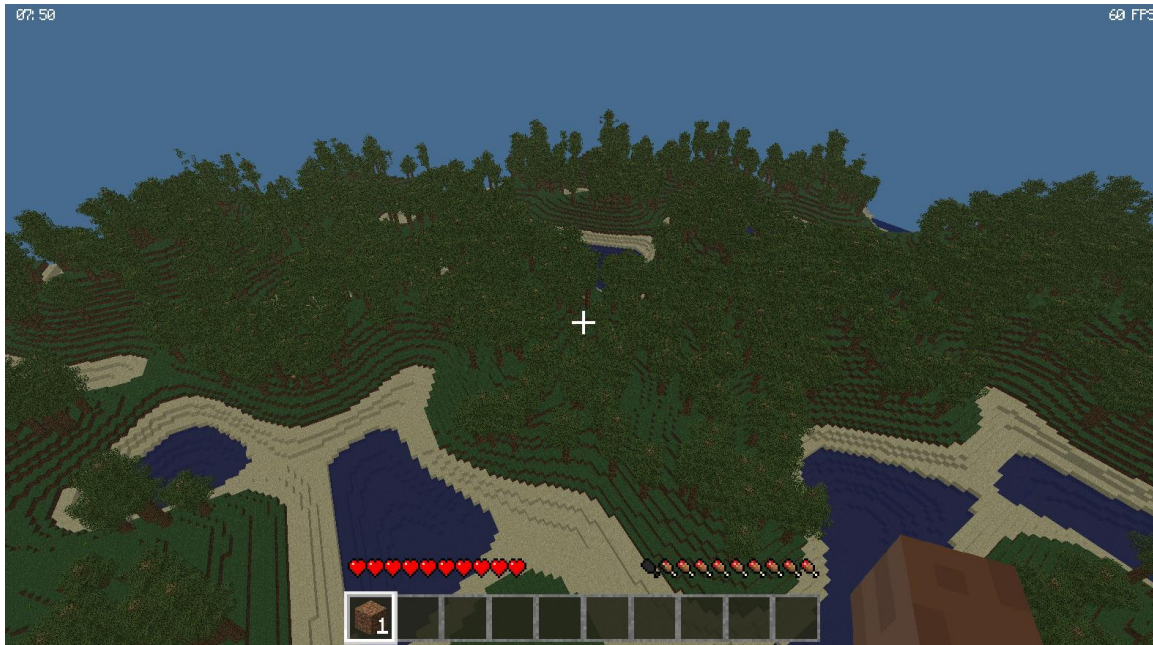
$$obscurité = 0,8^{15-niveauDeLumière}$$

Chaque bloc propage son intensité lumineuse sur ses voisins en réduisant l'intensité lumineuse de 1 par bloc parcouru. Un bloc non éclairé a par défaut une intensité lumineuse de 0 ce qui le rend totalement noir.

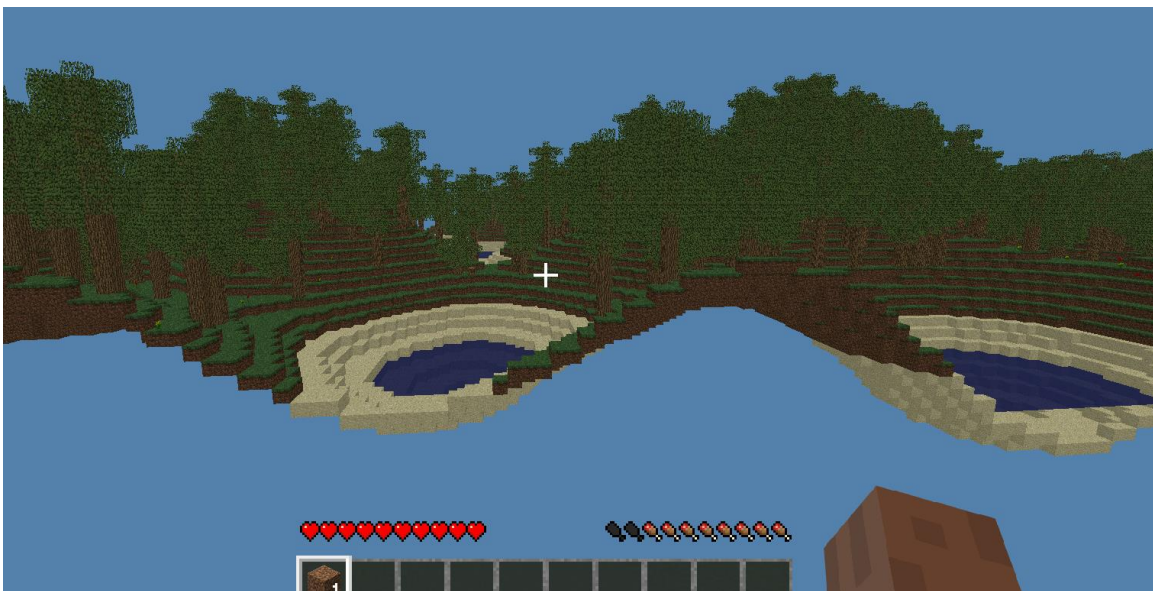


### 5.13. Optimisation du rendu

Afin de ne pas afficher l'intégralité des blocs du monde (Qui peut être constitué de plusieurs millions de blocs), il est nécessaire de sélectionner les blocs à afficher en utilisant des algorithmes dits de « culling ». On applique ici le « Frustum Culling » qui permet d'afficher uniquement les chunks (Region de 16 blocs par 16 blocs par 16 blocs) dans le champ de vision de la caméra, ainsi que le « Surface Culling » pour n'afficher que la surface du monde.



Application du « Frustum Culling »



Application du « Surface Culling »

## 6. CONTRÔLES

6.1.	Menu principal [CLIC GAUCHE]	Activer un bouton
6.2.	En jeu [ESCAPE] [LEFT_ALT]	Retour au menu principal Libère le curseur de la souris
6.3.	Mouvement [W] [S] [A] [D] [SPACE] [LEFT_CTRL] ou [W][W] [M]	Avancer tout droit Reculer Mouvement latéral gauche Mouvement latéral droit Sauter / Nager vers la surface Sprinter Changement de mode (Vol/Marche)
6.4.	Inventaire [Q] [0-9] ou [MOLETTE]	Détruire l'objet Sélectionner emplacement
6.5.	Interaction avec l'environnement [CLIC GAUCHE] [CLIC DROIT]	Casser un bloc Poser un bloc / Utiliser un objet
6.6.	Système de création d'objets [E] [A] [D] [F] [ESCAPE]	Ouvrir le menu de création d'objets Recette précédente Recette suivante Créer l'objet Fermer le menu
6.7.	Globaux [F1] [F2] [F3] [F4] [F5] [F6] [F7] [F8]	Afficher le taux de rafraichissement dans la console Benchmark Profiler sur plusieurs frames Profiler frame courante Recompiler les shaders Mode de rendu par défaut Mode de rendu « wireframe » Mode de rendu « normals »

## 7. CONFIGURATION

Avant de lancer le jeu il est possible de configurer celui-ci en se rendant dans le dossier Config/. Chaque fichier contient un ensemble de clé/valeurs qui seront interprétées par le jeu au démarrage. Amusez-vous à changer ces valeurs !