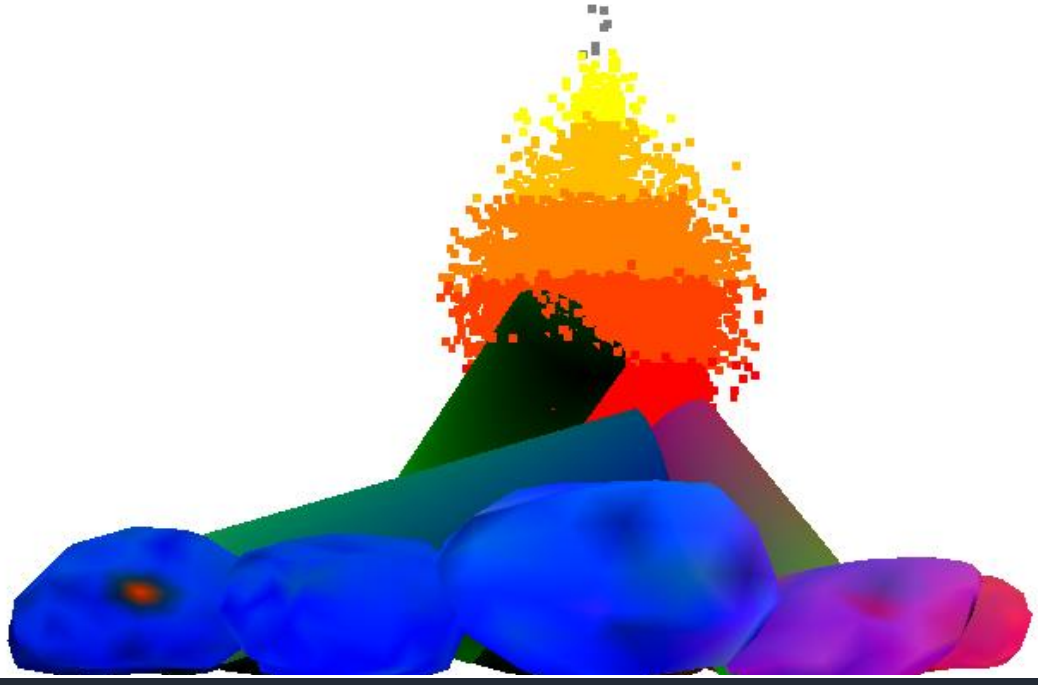
Systèmes de particules



IN55 – Synthèse d'images
Fabrice Lauri
UTBM | P17

Yawo Cyprien NKOUNOU
Florian SORON
Siyao HU

Sommaire

Introduction

Dans le cadre de l'Unité de Valeur IN55 (Synthèse d'image) enseignée à l'Université de Technologie Belfort-Montbéliard par M. Fabrice Lauri, nous avons eu l'occasion de travailler sur un projet de développement d'un système de particule. Nous avons choisi d'implémenter ce système de particule à travers l'exemple d'un feu de bois.

I. Présentation du projet

1. Présentation du groupe

Ce projet a été réalisé par NKOUNOU Yawo Cyprien, SORON Florian, et HU Siyao.

Ci-après un tableau récapitulant les rôles attribués à chaque membre du projet :

Nom du membre	Tâches réalisées
NKOUNOU Yawo Cyprien	Construction de l'environnement. Choix de design et GUI.
SORON Florian	Modélisation et rendu 3D avec Blender. Particule et cycle de vie.
HU Siyao	
Tous	UML - Rédaction du rapport - Prise en main de l'environnement Qt

2. Contexte

Étant tous étudiants en I2RV, le choix de l'UV IN55 a été une évidence plus qu'une obligation. Dans cette UV, nous avons notamment vu les concepts de base de la synthèse d'image, des calculs nécessaires pour faire les rendus. Nous avons pu mettre en œuvre ces connaissances dans le cadre des TPs.

Ce projet, en plus des TPs, nous a permis de mettre en pratiques nos connaissances et d'explorer divers outils et techniques de synthèse d'image. Pour le mener à bien, nous avons utilisé les technologies suivantes :

Blender est un logiciel libre et gratuit de modélisation, d'animation et de rendu en 3D.

Il dispose de fonctions avancées de modélisation, de sculpture 3D, de dépliage UV, de texturage, d'animation 3D, et de rendu. Il gère aussi le montage vidéo non linéaire, la composition, la création nodale de matériaux, la création d'applications 3D interactives ou de jeux vidéo grâce à son moteur de jeu intégré (le Blender Game Engine), ainsi que diverses simulations physiques telles que les particules, les corps rigides, les corps souples et les fluides.

Qt Creator est un environnement de développement intégré multiplateforme faisant partie du Framework Qt. Il est donc orienté pour la programmation en C++.

Il intègre directement dans l'interface un débogueur, un outil de création d'interfaces graphiques, des outils pour la publication de code sur Git et Mercurial ainsi que la documentation Qt. L'éditeur de texte intégré permet l'autocomplétion ainsi que la coloration syntaxique. Qt Creator utilise sous Linux le compilateur gcc. Il peut utiliser MinGW ou le compilateur Visual Studio sous Windows.

OpenGL (***Open Graphics Library***) est un ensemble normalisé de fonctions de calcul d'images 2D ou 3D lancé par Silicon Graphics en 19921. Cette interface de programmation est disponible sur de nombreuses plateformes où elle est utilisée pour des applications qui vont du jeu vidéo jusqu'à la CAO en passant par la modélisation.

OpenGL permet à un programme de déclarer la géométrie d'objets sous forme de points, de vecteurs, de polygones, de bitmaps et de textures. OpenGL effectue ensuite des calculs de projection en vue de déterminer l'image à l'écran, en tenant compte de la distance, de l'orientation, des ombres, de la transparence et du cadrage.

3. Objectifs du projet

L'objectif de ce projet était d'utiliser les compétences acquises en synthèse d'images pour produire le rendu de systèmes de particules.

a. Systèmes de particules

Un système de particules est une technique graphique numérique utilisée par les logiciels graphiques (2D ou 3D) ou d'effets vidéo. Elle permet de simuler de nombreux phénomènes naturels tels que feu, explosion, fumée, eau, nuage, poussière, neige, feux d'artifices, et animés à l'aide de forces qui agissent sur celles-ci telles que la gravité, le vent, l'inertie, etc.

Les composants essentiels d'un système de particules sont :

- Le système global
- L'émetteur
- Le modificateur

i. Le système global

Il s'agit ici de notre environnement.

ii. L'émetteur

Son rôle est comme son nom l'indique, d'émettre des particules. C'est lui qui les génère. Et c'est donc lui qui détermine leurs paramètres à ce moment-là. Un système de particule ne peut gérer qu'un seul émetteur à la fois. En revanche, un même émetteur peut être utilisé pour deux systèmes différents.

Les émetteurs de particules permettent de contrôler les éléments suivants :

- La propagation des particules (départ, direction)
- Le nombre de particules émises par seconde
- La couleur des particules
- La taille des particules
- La durée de vie des particules

iii. Le modificateur

Les modificateurs permettent d'agir sur les particules émises par un émetteur. En effet, sans modification celles-ci seraient bien mornes : pas de changement de couleur, de direction, etc... On peut utiliser plusieurs modificateurs sur un seul système, mais on ne peut pas utiliser un seul modificateur sur plusieurs systèmes.

b. Notre implémentation

Pour implémenter le système de particules, nous avons choisi un feu de bois. Pour ce faire, nous avons modélisé un environnement avec des pierres et du bois. Cet environnement va accueillir le feu.

II. Conception

1. Périmètre du projet

La première phase de la conception a consisté à délimiter un périmètre de développement clair

a. Objectif initial

Nous avons décidé d'implémenter notre système de particules sous forme d'un feu de bois, à l'exemple de celui-ci :

Figure 1 : Objectif initial - le feu de bois

À cette étape, nous avons scindé notre projet en deux parties.

La première partie porterait sur l'environnement; à savoir les pierres et les buches qui constitueraient la base sur laquelle nous allons appliquer le feu.

La deuxième partie porterait donc sur le cœur de notre projet : le système de particules (le feu).

b. Choix des outils

Pour la première partie (environnement), nous avons fait le choix de réaliser une modélisation 3D. Pour ce faire, nous avons utilisé le logiciel de modélisation et d'animation Blender.

Notre choix s'est porté sur Blender, car il est facile d'utilisation, et certains membres de notre équipe disposaient d'une expérience préalable avec ce logiciel.

La seconde partie de notre projet était l'implémentation du système de particule, et nous avons utilisé Qt Creator avec le template fourni par M. Lauri.

Le choix de Qt Creator était évident puisque c'est avec ce logiciel que nous avons été initiés au rendu.

2. Conception logicielle

Pendant la phase de conception de notre projet, nous avons étudié le scénario d'utilisation de notre simulateur et nous avons détaillé le fonctionnement qu'aurait ce simulateur.

a. Diagramme de contexte

Voici le scénario d'utilisation de notre système, dans un diagramme de contexte.

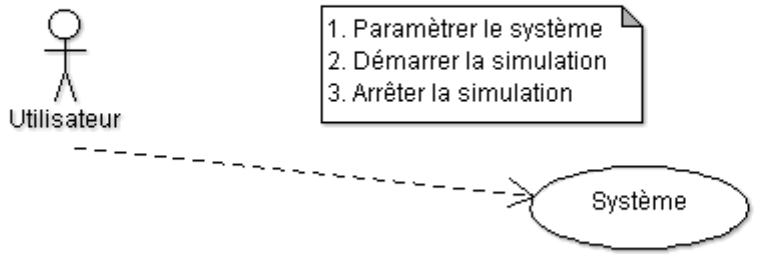


Figure 2 : Diagramme de contexte

❖ Commentaires

L'utilisateur est une personne qui désire voir le rendu de la simulation d'un feu dans de bois.

L'utilisateur charge le projet, en choisissant le fichier .pro du répertoire du projet.

Ensuite, l'utilisateur peut installer les librairies nécessaires si besoin.

Il peut enfin lancer la simulation, et voir le rendu.

L'utilisateur a enfin la capacité d'arrêter la simulation en fermant la fenêtre de la simulation.

b. Diagramme de séquence

Voici le diagramme de la simulation.

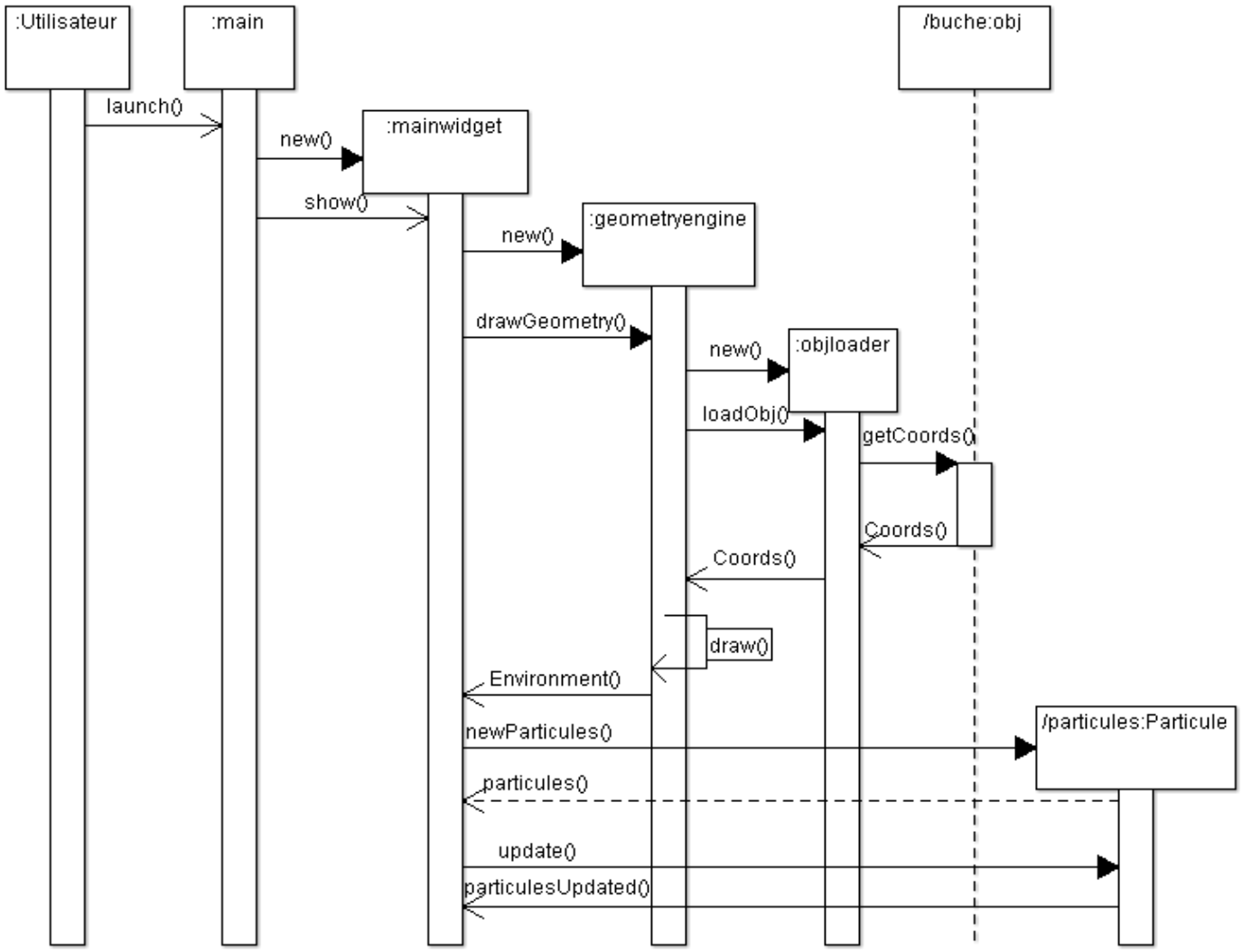


Figure 3 : Diagramme de séquence

- Commentaires

Au lancement de la simulation, le *main* instancie un *mainwidget*. Ce *mainwidget* crée la fenêtre de rendu et instancie un *geometryengine*, à qui il fait appeler la fonction *drawGeometry().*

La fonction *drawGeometry()* instancie un *objloader* dont la fonction *loadObj()* permet de récupérer les coordonnées nécessaires pour dessiner l'environnement (buches et pierres).

mainwidget instancie un certain nombre de particules et les fait évoluer en les mettant à jour de façon périodique.

III. Développement

Notre développement a porté sur deux parties: La modélisation de l'environnement et le système de particules

1. Modélisation de l'environnement

Pour modéliser notre environnement, nous avons utilisé Blender.

Nous avons tout d'abord créé un plan, auquel nous avons appliqué une texture. Nous avons ensuite créé des cylindres auxquels nous avons appliqué les textures appropriées sur les extrémités et sur les côtés. Nous avons enfin importé des pierres.

Notre environnement était ainsi constitué.

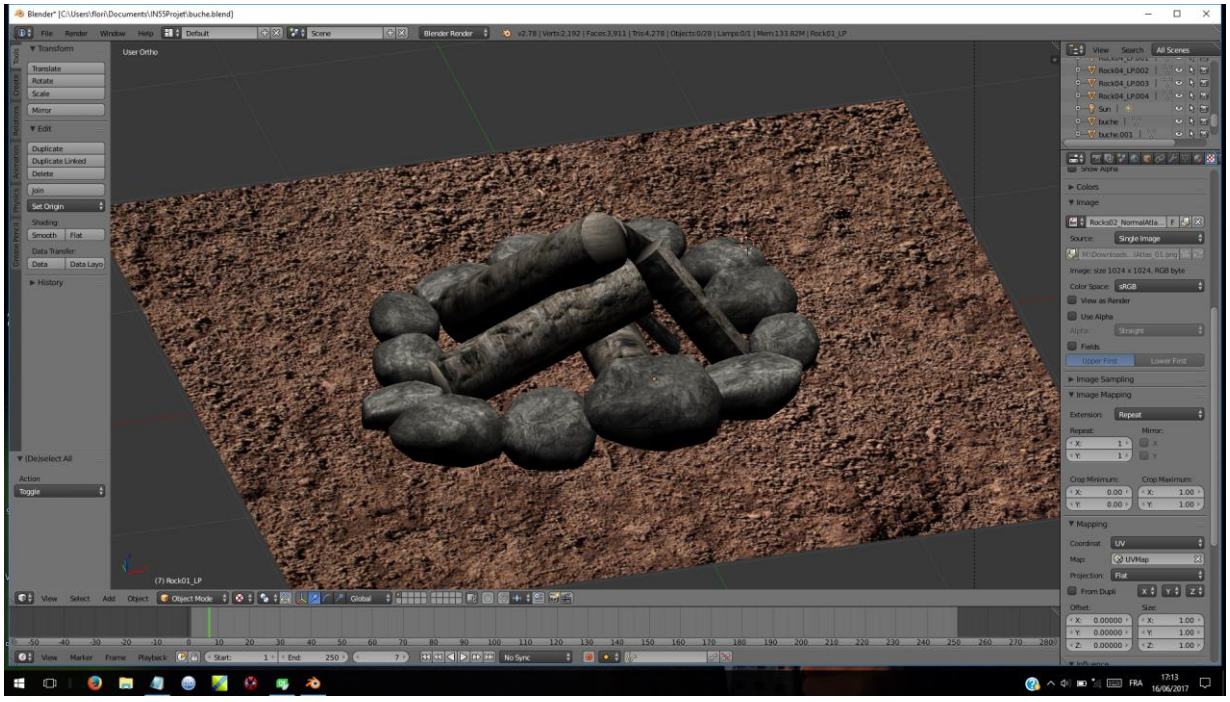


Figure 4 : Modélisation de l'environnement dans Blender

Figure 5 : L'environnement 3D

Après avoir modélisé l'environnement, nous avons exporté les éléments nécessaires pour obtenir cet environnement dans Qt Creator.

Nous avons ainsi récupéré un fichier *.obj*, contenant les coordonnées de l'environnement (vertices, textures, normales), et un fichier *.mtl* contenant les paramètres des textures, tels que l'illumination.

2. Le système de particules

Cette partie du projet a été réalisée dans Qt Creator.

Pour simuler le rendu, nous avons suivi le diagramme de séquence présenté plus haut.

Voici la structure ou arborescence de notre projet.

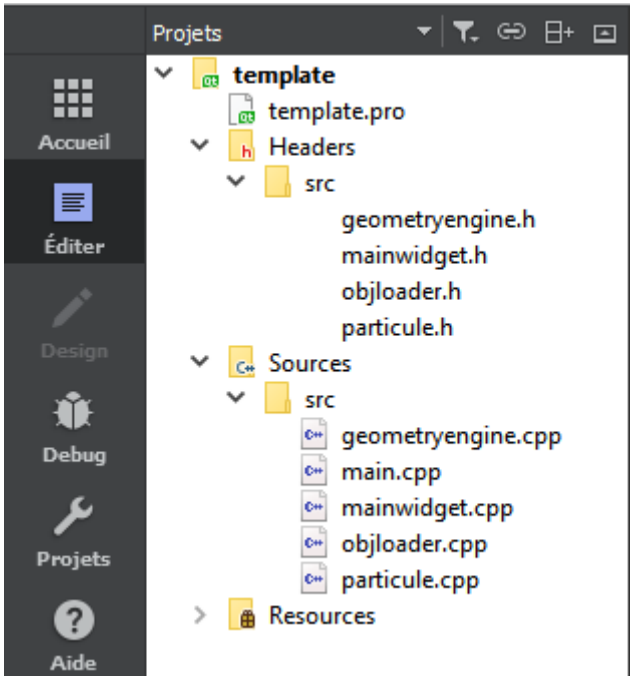


Figure 6 : Structure du projet

a. Rendu de l'environnement

Ce rendu utilise le fichier buche.obj exporté de Blender.

mainwidget est le point de départ de la simulation. *mainwidget* instancie *geometryengine* qui à son tour instancie *objloader* qui récupère des coordonnés de *buche.obj*.

Ces coordonnées seront utilisées par *geometryengine* pour produire le rendu de l'environnement

Nous avons alors le résultat suivant :

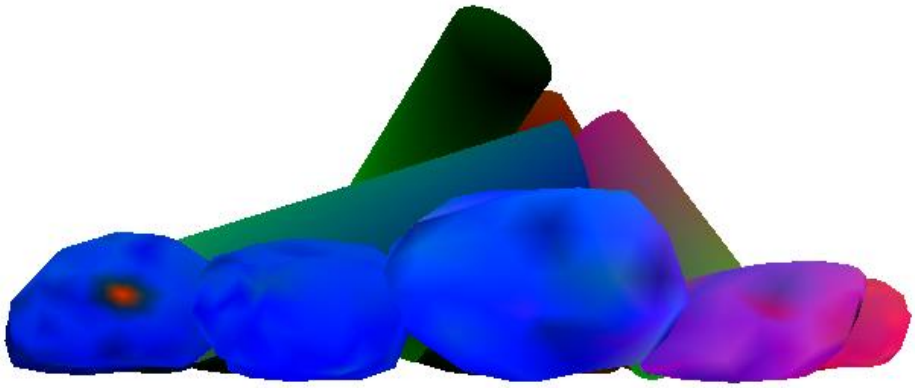


Figure 7 : Rendu de l'environnement (vue de profil)

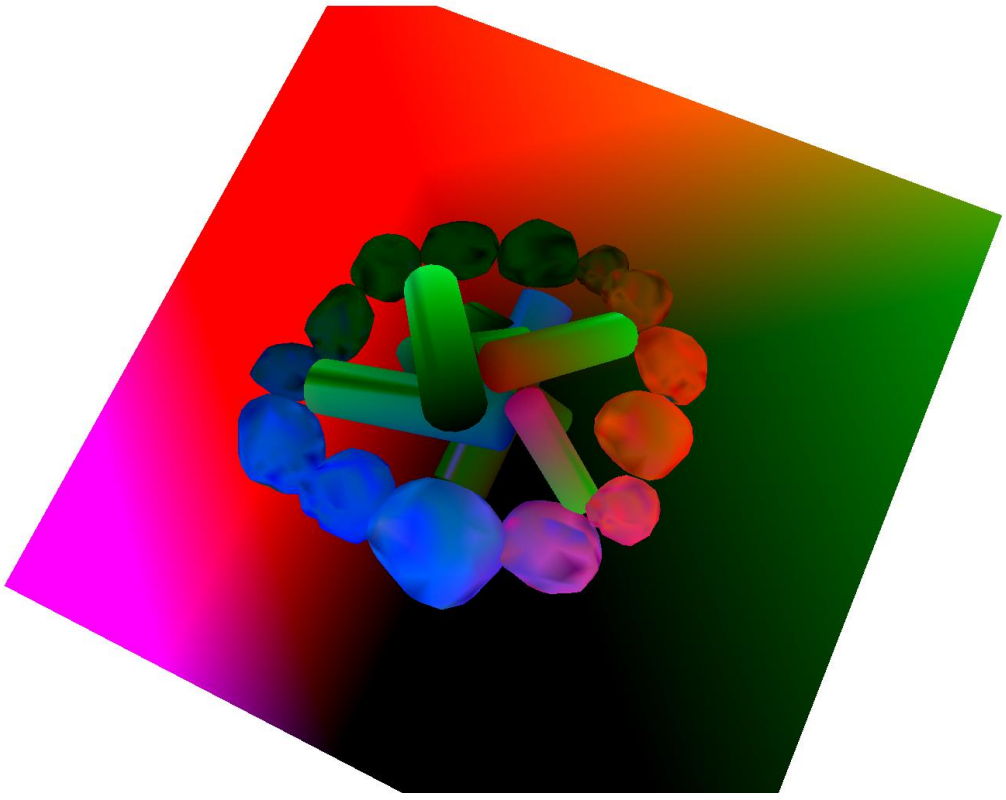


Figure 8 : Rendu de l'environnement (vue de haut)

b. Rendu du système de particules

Dans *particule*, on définit les coordonnées des vertices et des couleurs pour une particule carrée rouge.

Dans *mainwidget* on définit un nombre maximum de particules et on génère ce nombre de particules. C'est l'**émission des particules**.

```
        // On dessine toutes les particules
        for(int i = 0; i<NBR_MAX_Particule;i++)
        {
        part[i] -> drawParticle(&program);
        }

}
```

Figure 9 : Emission des particules

Une fois que les particules sont générées, on les modifie en suivant un algorithme particulier.

A chaque instant *t* qui passe, on modifie la position et la couleur d'un certain nombre de particules jusqu'à les avoir toutes modifiées.

La fonction *update()* de *particule* modifie les coordonnées des vertices, et les coordonnées des couleurs en essayant de simuler le comportement d'un feu. C'est la modification.

Ainsi nous avons :

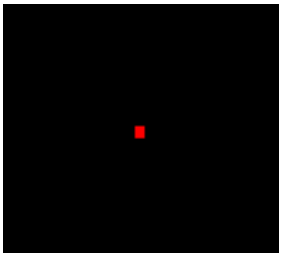


Figure 10 : Rendu à t=0

À t=0, toutes les particules sont créées en un même endroit.

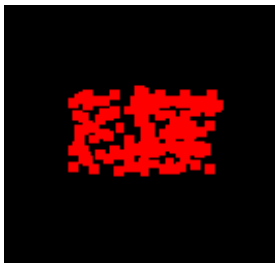


Figure 11 : Rendu à t=1

À t=1, exactement 1% des particules sont mises à jour et dispersées.

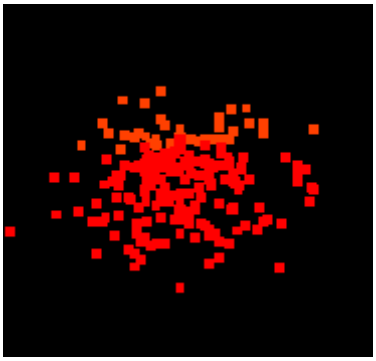


Figure 12 : Rendu à t=3

À t=3, un plus grand nombre de particules est modifié. La dispersion est plus grande.

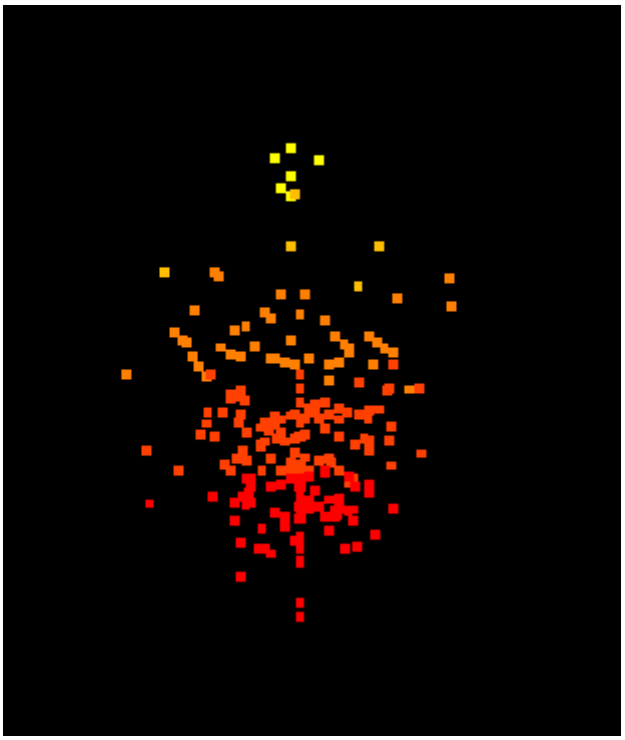


Figure 13 : Rendu à t=25

À t=25, le quart des particules a été modifié. Un grand nombre de ces particules a été modifié plusieurs fois, et on constate une élévation.

Les particules au-delà d'un certain seuil sont mises à jour avec une couleur différente.

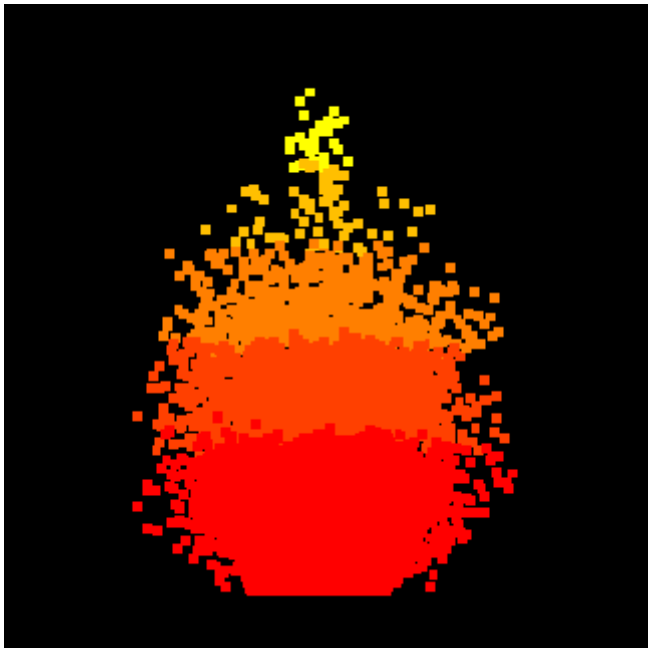


Figure 14 : Rendu à t=100

À t=100, toutes les particules ont été modifiées au moins une fois, et on a un rendu qui simule du feu, avec une variation de couleur des particules par palier, allant du rouge au jaune.

c. Rendu final

En combinant le rendu de l'environnement et le rendu du système de particules, on a le résultat escompté.

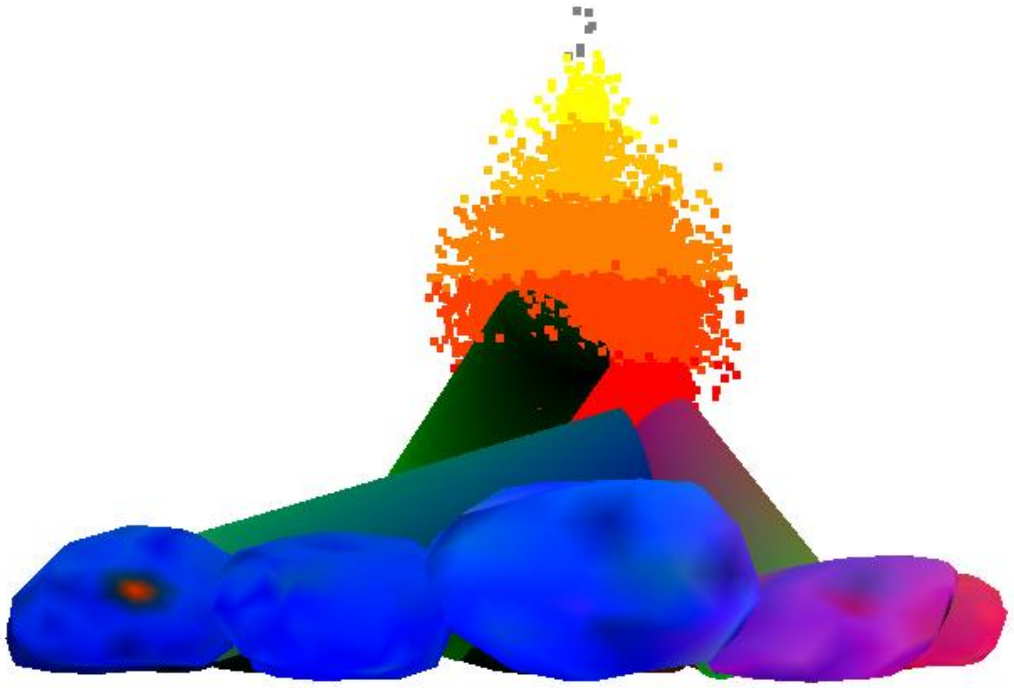


Figure 15 : Rendu final (Vue 1)

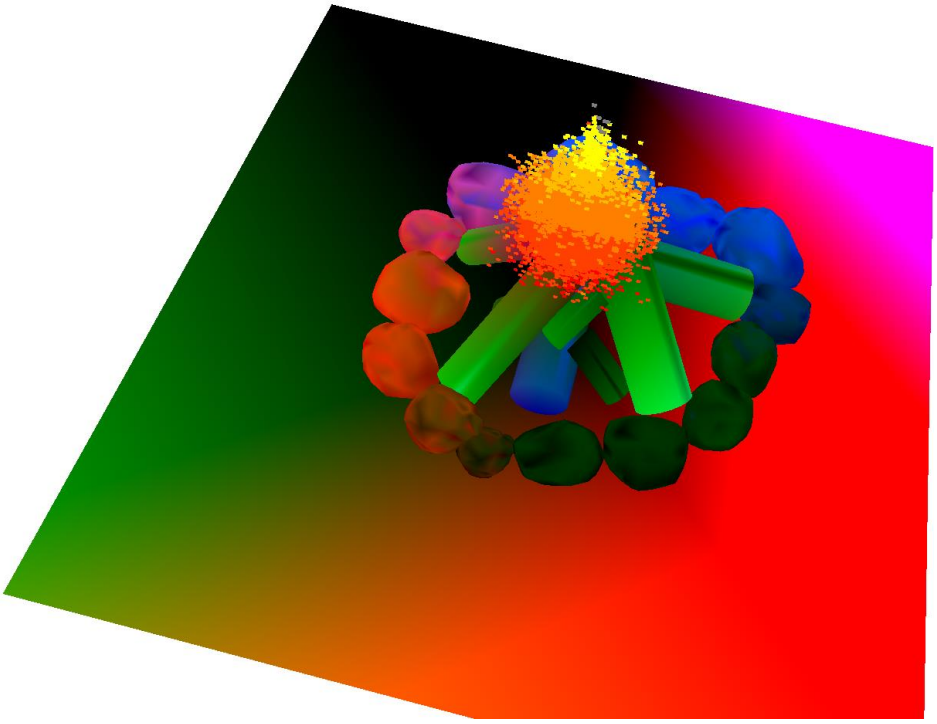


Figure 16 : Rendu final (Vue 2)

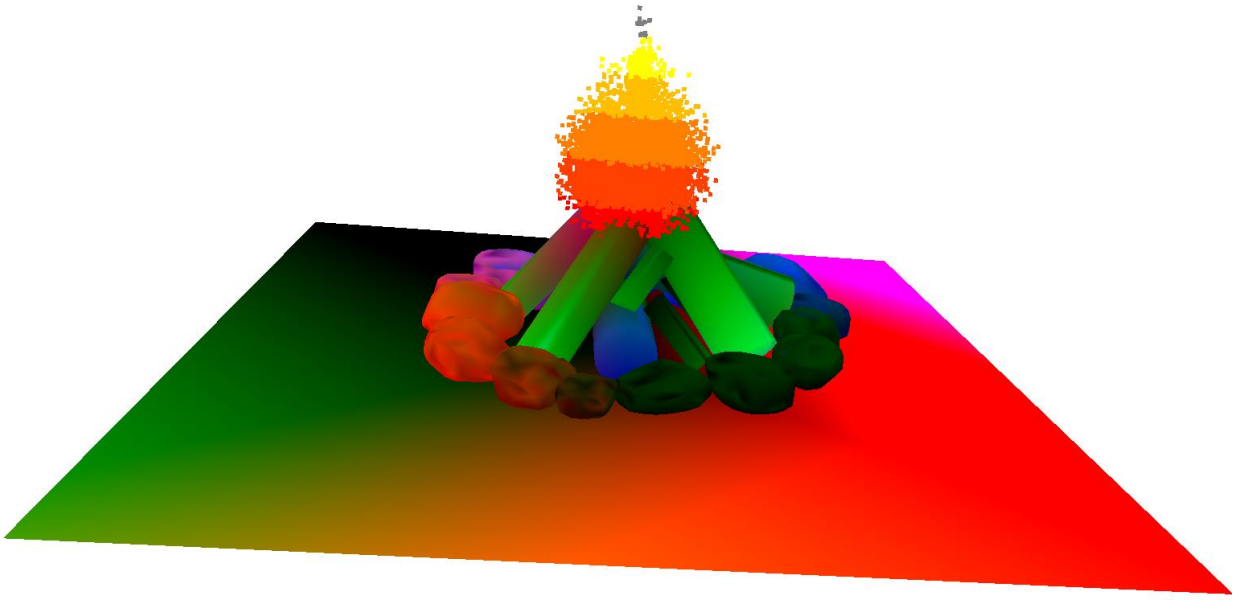


Figure 17 : Figure 16 : Rendu final (Vue 3)

IV. Difficultés rencontrées

Pendant la réalisation de ce projet, nous avons été confrontés à un certain nombre de difficultés, notamment pour importer la modélisation de l'environnement de Blender dans Qt Creator. Nous avons finalement réussi mais nous n'avons pas pu importer les textures et les appliquer.

Conclusion

Ce projet nous a permis de mettre en pratique nos connaissances acquises au cours de l'UV IN55, mais aussi d'explorer d'autres horizons notamment avec la modélisation 3D dans Blender. Nous avons aussi été confrontés à un certain nombre de difficultés, mais nous avons pu les surmonter pour la plupart et nous avons réussi à générer le rendu du système de particules sous forme de feu de bois.