

RV01 : Réalité virtuelle

# INSTANT SCULPT

Rapport final

Valentin Cecchin

Aurélie Cammerman



# Table des matières

<b>Table des matières .....</b>	<b>2</b>
<b>I. Introduction.....</b>	<b>3</b>
1. But du projet .....	3
2. Nom du projet .....	3
3. Objectif pédagogique .....	3
4. Présentation des acteurs et de l'équipe encadrante .....	3
<b>II. Description du projet.....</b>	<b>4</b>
1. Description du projet.....	4
2. Matériel utilisé.....	4
3. L'environnement du projet .....	5
4. Principe de fonctionnement .....	8
<b>III. Bilan du projet.....</b>	<b>9</b>
1. Niveaux de réalisation .....	9
2. Différentes phases du projet.....	10
3. Contraintes et difficultés rencontrées.....	14
4. Pistes d'amélioration.....	14
<b>IV. Conclusion .....</b>	<b>15</b>

# I. Introduction

## 1. But du projet

Le projet doit permettre à l'utilisateur de sculpter en réalité virtuelle un objet qu'il aurait. Pour cela, il pourra ajouter ou enlever de la matière dans l'espace virtuel. Il s'agit donc de créer et modifier des géométries polygonales en temps réel avec une action directe de l'utilisateur sur l'objet.

## 2. Nom du projet

Notre projet s'appellera Instant Sculpt, puisque l'utilisateur peut sculpter en temps réel et en 3D.

## 3. Objectif pédagogique

L'objectif pédagogique de RV01 est d'initier les étudiants à la réalité virtuelle en leur donnant l'opportunité de créer une application contenant des interactions 3D pour la navigation et la manipulation d'objets virtuels.

L'UV nous apprend tout d'abord à utiliser le moteur graphique 3D Unity puis nous devons développer une application en veillant à soigner les interactions avec l'utilisateur. Il faut aussi réfléchir à d'autres aspects permettant de rendre le monde plus immersif : gestion de la physique, modélisation 3D, acoustique...

## 4. Présentation des acteurs et de l'équipe encadrante

### **Ce projet sera réalisé par deux étudiants d'ingénierie mécanique :**

- Aurélie Cammerman IM05 (IDI)
- Valentin Cecchin IM05 (filiale libre)

Pour Aurélie, cette UV est aussi l'occasion de créer une application avec une démarche centrée utilisateur tel qu'étudié en filière IDI. Nous nous concentrerons sur la nature des interactions et sur le design graphique afin de garantir une expérience agréable pour l'utilisateur.

Ce projet de RV01 consiste en une interaction complexe entre utilisateur et géométrie. Il s'intègre dans la continuité de la filière libre de Valentin basée sur les manipulations 3D graphiques et mécaniques pour la simulation, ici géométrique.

**L'équipe encadrante est constituée de :**

- Indira Thouvenin, enseignante chercheuse à l'UTC et responsable de l'UV RV01
- Azzeddine Benabbou, responsable du TD de l'UV RV01

## II. Description du projet

### 1. Description du projet

Le projet doit permettre de sculpter un objet en réalité virtuelle. L'utilisateur doit pouvoir enlever ou ajouter de la matière à une forme de base en temps réel. La géométrie créée sera polygonale, ce qui signifie qu'elle pourra aisément être réutilisée à d'autres fins, par exemple comme modèle 3D graphique pour un jeu ou pour l'impression 3D.

Le projet se concentrera exclusivement sur l'aspect de modélisation en temps réel dans un contexte de réalité virtuelle

Afin de réaliser cette interaction, nous avons décidé d'adapter l'algorithme des Marching Cubes comme base de notre projet. Nous avons eu l'occasion d'étudier cet algorithme dans l'UV TN20. Ce projet n'utilisera donc pas des techniques de subdivisions adaptatives telles qu'utilisées dans les logiciels d'infographie 3D pour le sculptage.

### 2. Matériel utilisé

Le projet sera développé sous Unity, un moteur 3D permettant le développement d'application de réalité virtuelle gratuitement.

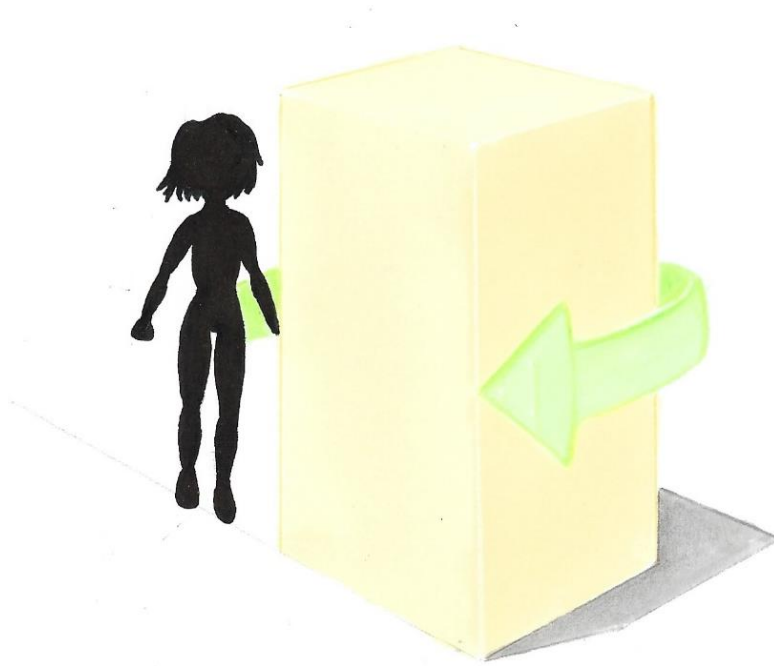
Pour l'environnement virtuel, nous avons choisi d'utiliser le casque de réalité virtuelle HTC Vive et ses deux manettes.

Par ailleurs, nous utiliserons un casque sonore pour appliquer des effets sonores lorsque l'utilisateur ajoute et enlève de matière afin d'accroître la sensation d'immersion.

### 3. L'environnement du projet

L'application s'ouvre sur un menu, où l'utilisateur pourra choisir des options telles que la taille du bloc qu'il souhaite sculpter ou encore le niveau de raffinement qu'il souhaite.

Puis lorsque l'utilisateur souhaite commencer, le bloc apparaît devant lui. L'utilisateur est libre de tourner autour du bloc afin de sculpter toutes les facettes. Lorsque l'utilisateur se déplace dans l'environnement réel, il se déplace aussi dans l'environnement virtuel, tel que montré sur la Figure 1.



*Figure 1 : L'utilisateur peut se déplacer dans l'environnement réel afin de sculpter les différentes facettes du bloc*

Ensuite, l'utilisateur peut commencer à sculpter le bloc en ajoutant ou enlevant de la matière, voir Figure 2. Pour sculpter, l'utilisateur devra appuyer sur la gâchette de la manette. Il pourra choisir d'ajouter de la matière ou d'en enlever en appuyant sur une touche de la manette qui permettra d'effectuer le changement. Un signal sonore et la modification directe de la géométrie devant lui pourront lui confirmer que l'action de changement a bien été effectuée. L'interaction de sculptage sera donc directe

Lorsque l'utilisateur sera en train de sculpter, nous devons lui indiquer où se trouve sa main dans l'espace virtuel. Pour cela nous avons pensé à deux solutions :

- une main virtuelle pourra lui indiquer la position où il se trouve
- une orbe située à bout de main pourra lui indiquer la position et la taille de la partie qu'il s'apprête à sculpter

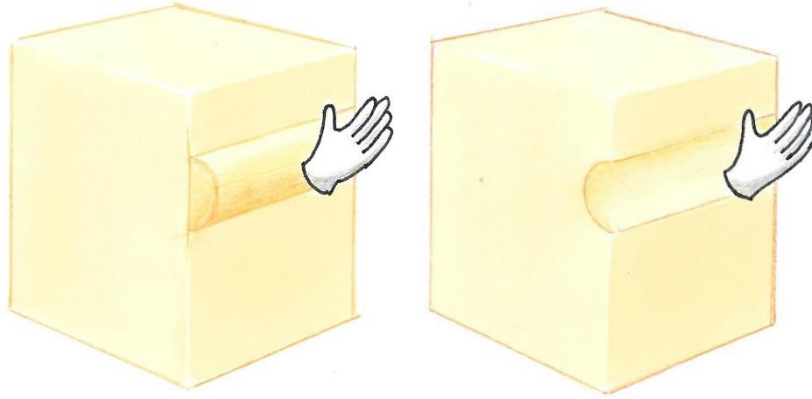


Figure 2 : Illustration de l'enlèvement ou ajout de matière sur le bloc

Puisqu'il est possible d'utiliser deux manettes avec le casque HTC Vive, nous ferons en sorte que l'utilisateur puisse interagir avec ses deux mains.

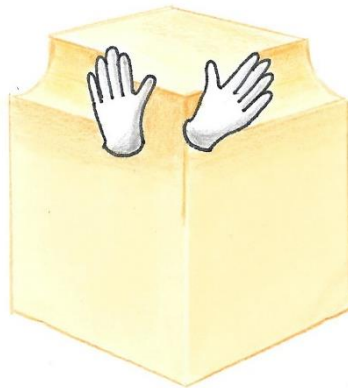


Figure 3 : Illustration de l'enlèvement de matière sur le bloc avec les deux mains

Afin d'améliorer notre projet, de le rendre plus immersif et plus complet, nous pourrions ajouter des effets sonores lorsque l'utilisateur est en train de sculpter. Nous pourrions aussi ajouter des options telles que la forme de la brosse ou sa taille, ainsi qu'indiquer l'option choisie dans un coin de l'écran.

Concernant les interactions pour changer le style de la brosse, nous avons pensé faire apparaître un menu lorsque l'utilisateur appuie sur un bouton de la manette. Il devra alors lever ou baisser sa main pour choisir l'option qui lui convient. Voir Figure 1.

Pour le moment, nous avons pensé ajouter 4 options dans le menu :

- ajout / enlèvement de matière sur le bloc
- ajout / enlèvement de matière dans le vide

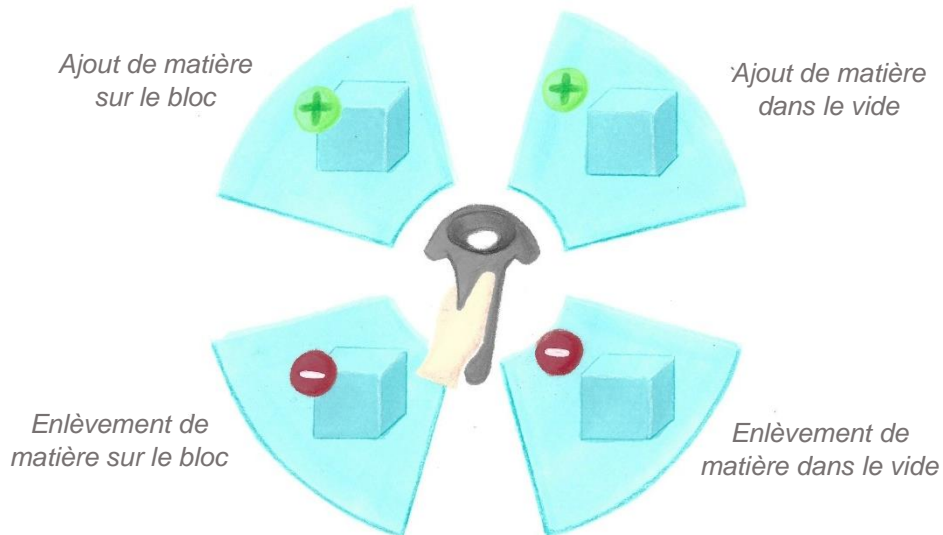


Figure 4 : Illustration du menu permettant de choisir l'option de la brosse

Par ailleurs, pour changer la taille de la brosse, nous avons pensé comme interaction que lorsque l'utilisateur appuierait sur un bouton de la manette, un mouvement de sa main décalée par rapport à la verticale ferait grossir la brosse et à contrario, en ramenant sa main à la verticale, la taille de la brosse diminuerait. Voir Figure 5.

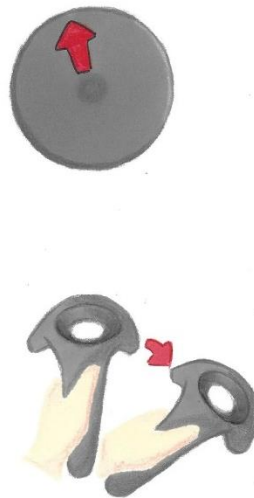


Figure 5 : Illustration du changement de la taille de la brosse, en inclinant la main la taille de la brosse grossit

Le projet regroupera donc une interaction principale en réalité virtuelle qui sera toute modification apportée à l'objet sculpté. D'autres interactions pourront être implémentées, comme le choix de la taille ou du type (ajout, retrait, inflation...) de la zone de sculptage.

#### 4. Principe de fonctionnement

Afin de réaliser ce projet nous nous appuyerons sur l'algorithme des Marching cubes afin d'ajouter ou de retirer de la matière.

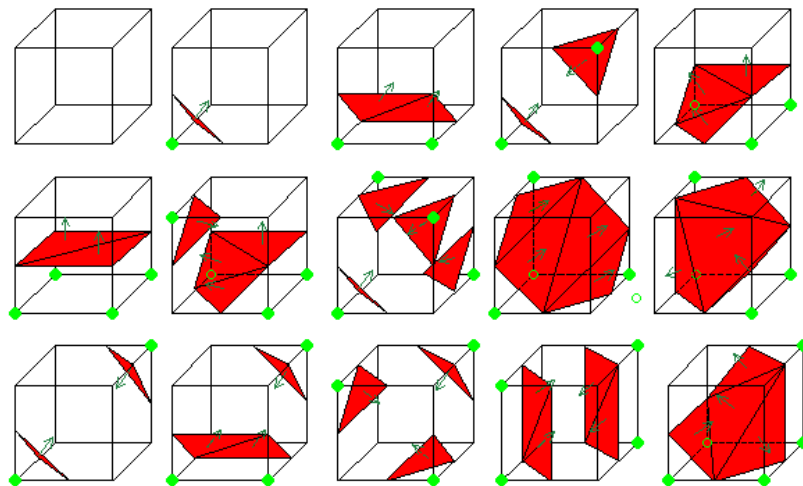


Figure 6 : Cas de génération des triangles pour des données entrées en voxels par l'algorithme des marching cubes

Les marching cubes sont un algorithme de polygonisation à partir de champ scalaire. L'algorithme consiste en une recherche de motifs 3D dans une table de voxel (pixel 3D). Chaque motif est associé à une surface triangulée. L'ensemble de toutes les surfaces générées forme le polygone final. Cette technique est très utilisée en imagerie médicale pour reconstituer des modèles 3D à partir de mesures discrètes dans l'espace. Ici, nous l'utiliserons pour créer une surface polygonale à partir de donnée correspondant à la position des mains de l'utilisateur.

A la différence d'un algorithme de subdivision dynamique, le sculptage réalisable avec les marching cubes sera discret et non pas continu. L'aspect obtenu ne sera pas lisse, les surfaces auront un effet de vague dépendant directement de la résolution de grille choisie. Une amélioration finale du projet consistera alors à lisser l'objet final afin d'obtenir le meilleur rendu possible.



## III. Bilan du projet

### 1. Niveaux de réalisation

#### Niveau 1 :

- Génération de surfaces polygonales ✓
- Interaction fluide de sculptage dans une zone restreinte ✓
- Possibilité d'interagir en retirant de la matière ✓

#### Niveau 2 :

- Changement de propriétés des brosses de sculptage (taille, ajout de surface dans le vide ou à côté de polygones existants) ✓
- Choix de résolution de la zone d'édition au démarrage ✓
- Création d'un menu principal et de menus pour changer la taille de la brosse ✓

#### Niveau 3 :

- Surfaces courbes plutôt que polygonales ✓
- Fonctionnalités de sauvegarde et chargement de surface créée ✗
- Ajout de sons, retours haptiques, retours graphiques, retours sonores ✓

Nous avons réalisé presque tout ce que nous nous étions fixés dans le cahier des charges initialement. Quelques modifications ont cependant été choisies. Pour l'ajout de matière à coté de polygone ou dans le vide, nous n'avons pas séparé les options puisque les deux options ne faisaient aucune différence lorsqu'on édite la géométrie à portée de main. A la place, nous laissons la possibilité à l'utilisateur d'utiliser une brosse d'édition soit cubique soit lisse, ce qui ajoute de la variété d'édition. Pour ce qui est de la sauvegarde de la géométrie, nous n'avons pas eu le temps de la mettre en place.

## 2. Différentes phases du projet

### Présentation de la génération / du retrait de géométrie

Tout d'abord, nous nous sommes concentrés sur l'ajout et le retrait de matière sous forme polygonale (cube), nous avons obtenu le résultat suivant

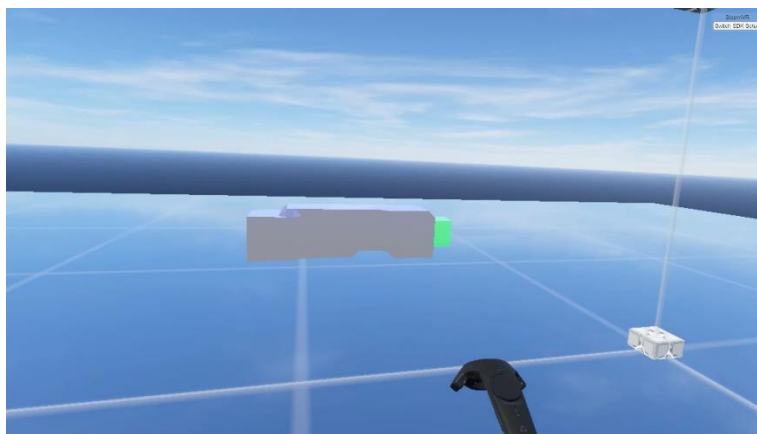


Figure 7: Ajout de matière géométrie non lisse (niveau 1)

Ensuite, nous avons modifié l'algorithme afin d'avoir une géométrie plus lisse quasi sphérique. Nous avons obtenu le résultat suivant :

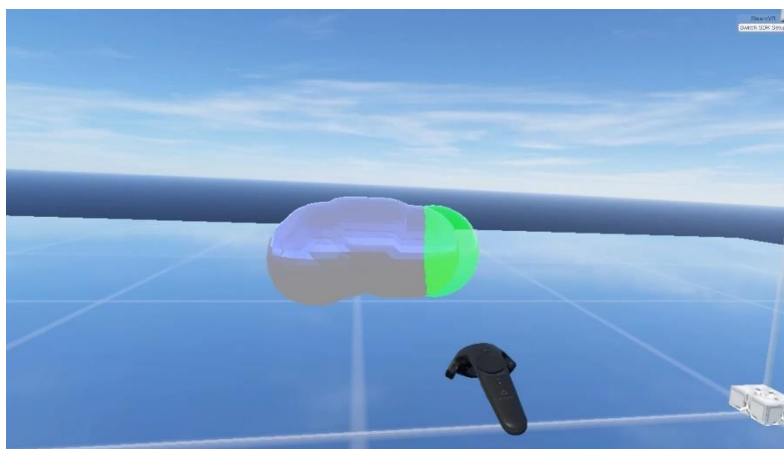


Figure 8: Ajout de matière géométrie lisse (niveau 3)

Nous avons aussi choisi de pouvoir enlever de la matière :

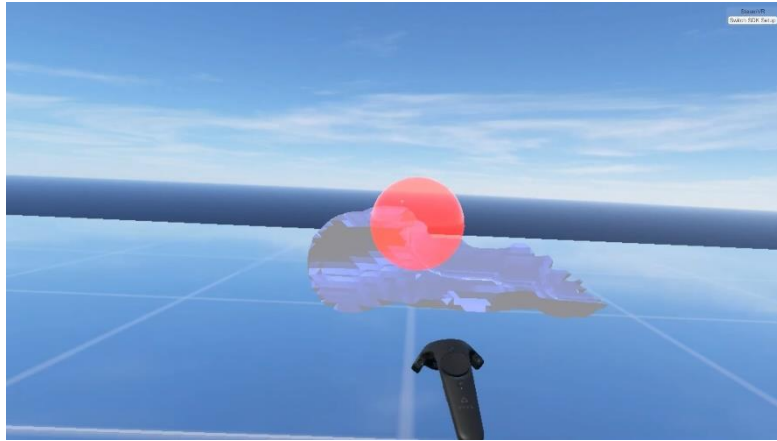


Figure 9: Enlèvement de matière géométrie lisse (niveau 3)

Nous nous sommes assurés que le programme soit fonctionnel avec les deux manettes simultanément.

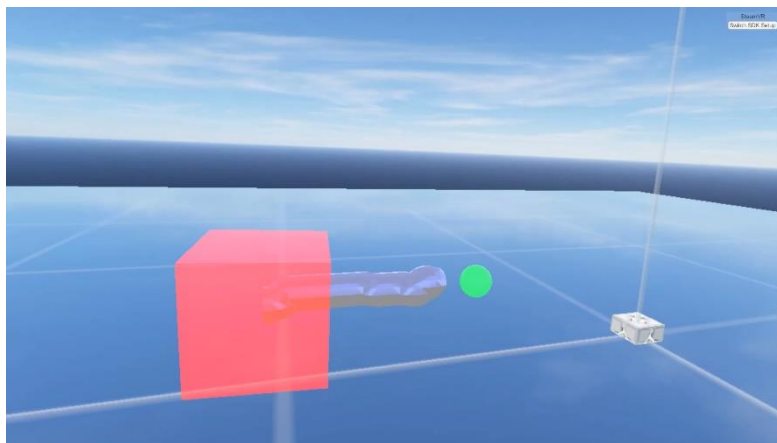


Figure 10: Enlèvement de matière avec la main gauche, ajout avec la main droite(niveau 3)

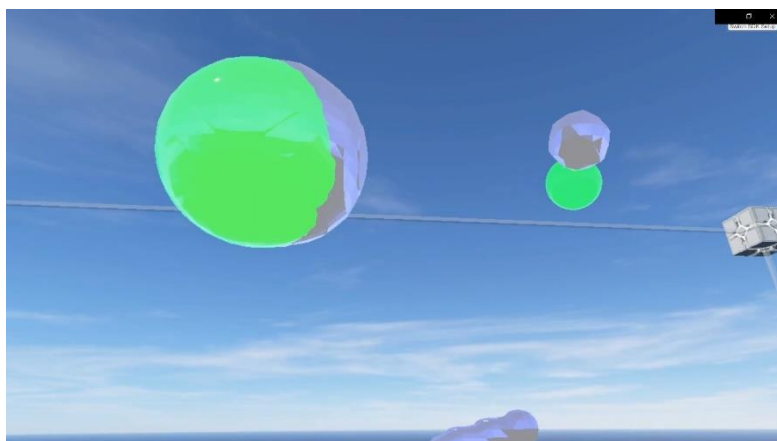


Figure 11: Ajout de matière avec les 2 mains

Pour chacune des options (ajout et enlèvement de matière), des vibrations permettent d'améliorer l'immersion.

## Présentation des différents menus et interfaces utilisateur

Pour le menu principal, nous avons choisi de faire un menu en 3 dimensions, l'utilisateur est au milieu de la pièce et lorsqu'il se tourne, il voit un écran différent un peu comme dans une pièce. On met en face de l'utilisateur un écran présentant le projet, à droite, un écran pour régler les options avant de commencer et à gauche un écran pour les sauvegardes (les sauvegardes n'ont pas été développées dans le projet par manque de temps). L'écran actif se met en surbrillance afin que l'utilisateur comprenne où il peut interagir.

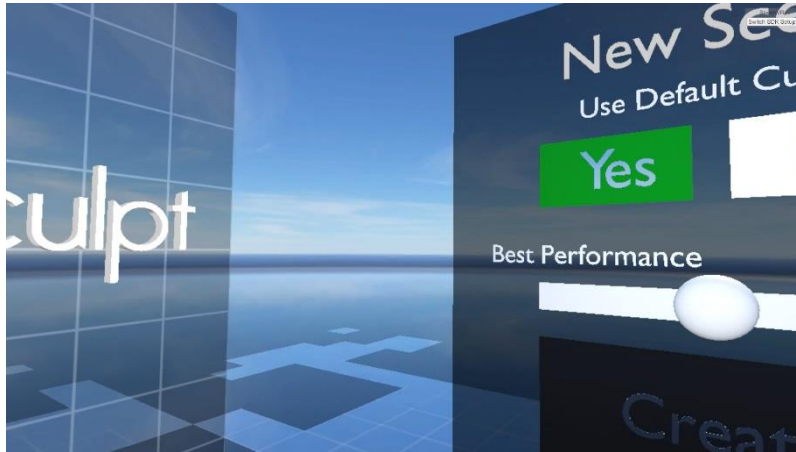


Figure 12: Menu principal, lorsque l'utilisateur tourne vers la droite

L'écran de droite présente les options que l'on peut régler avant de commencer le sculptage. On peut notamment ajouter un cube de départ afin de sculpter dessus, changer la résolution de la scène (meilleure performance ou meilleure qualité). Enfin, on appuie sur « Create » pour commencer. Les interactions avec les boutons du menu se font avec des changements de couleur afin de donner un retour à l'utilisateur.



Figure 13: Menu principal, écran de droite pour les réglages

Une fois entré dans la scène de sculptage, nous avons choisi de faire une interface permettant de changer la taille de la sphère de sculptage



Figure 14: Menu de changement de la taille de la brosse

Pour choisir la taille de la brosse de sculptage, on appuie sur le bouton latéral de la manette, un effet sonore nous indique que l'on est rentré dans le menu. Une sphère apparaît à l'écran, on peut grossir la sphère en éloignant la manette du point initial et au contraire diminuer la taille en se rapprochant du point initial. La sphère qui apparaît correspond à la nouvelle taille de brosse que l'on aura pour sculpter. En relâchant le bouton latéral, un effet sonore nous indique que l'on est sorti du menu. On peut alors sculpter avec la nouvelle taille de brosse choisie.

Ensuite, nous avons fait une interface permettant de changer le type de brosse par l'utilisateur. Les 4 types des brosses que nous avons choisi sont :

- Brosse carrée, ajout de matière
- Brosse sphère, ajout de matière
- Brosse carrée, enlèvement de matière
- Brosse sphère, enlèvement de matière

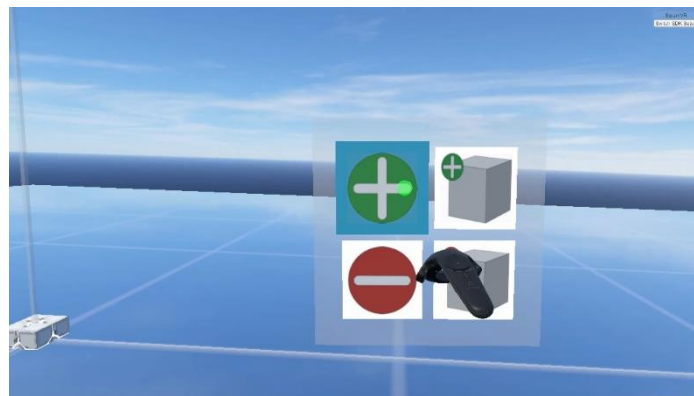


Figure 15: Menu de changement du type de brosse

Lorsque l'on appuie sur le touchpad de la manette, un menu apparaît, et un son nous indique que nous sommes entrés dans le menu. Il suffit alors de choisir l'option que l'on souhaite en déplaçant la manette. L'icône sélectionnée se met en surbrillance. Il suffit alors de relâcher le touchpad pour sélectionner l'option.

### 3. Contraintes et difficultés rencontrées

Voici les difficultés que nous avons envisagées:

- Interagir avec l'utilisateur, entre les deux manettes et la géométrie créée et entre l'utilisateur et les menus
- Optimiser l'algorithme de polygonisation afin que l'interaction soit fluide
- Traiter l'interpolation du sculptage entre deux positions calculées par le HTC Vive
- Éviter de surcharger le processeur et la RAM du pc utilisé

La plus grosse difficulté rencontrée a été de faire fonctionner l'algorithme des Marching Cubes en temps réel pour la réalité virtuelle et de manière à prendre en compte des modifications de l'utilisateur. La solution trouvée consiste à ne déclencher la génération de géométrie par les Marching Cubes que localement, là où l'utilisateur essaye de créer ou effacer une surface. Plutôt que de modifier à chaque interaction la géométrie, elle est supprimée localement puis recréée.

L'espace de modification étant discret, il a fallu interpoler les positions des manettes entre chaque frame sur la grille de voxel. Ceci a été résolu en adoptant l'algorithme de tracé de ligne de Bresenham.

### 4. Pistes d'amélioration

#### - **Etude d'ergonomie**

Il serait intéressant de faire une étude d'ergonomie sur l'application que nous avons développée. Ceci permettrait de vérifier si les interactions sont intuitives ou au contraire trop obscures. Cette étude fournirait un retour pour déterminer s'il faut modifier ou revoir les menus.

#### - **Amélioration des performances**

La géométrie générée par l'algorithme contient un nombre excessif de polygones. Il est plutôt commun en éditant d'atteindre les millions de triangles. Ceci peut représenter une charge excessive dans le pipeline de rendu graphique. Nous pouvons constater des ralentissements lorsque la carte graphique doit calculer l'affichage de tous ces triangles. Plusieurs approches peuvent être envisagées afin d'améliorer les performances.

- **Algorithme Decimate**

L'utilisation d'un algorithme de décimation permettrait de réduire le nombre de triangle visible. Cette méthode consiste à modifier la géométrie, en fusionnant des triangles étant sur le même plan et étant cotes à cotes. Ceci permettrait en modifiant les surfaces générées de réduire le nombre de triangle total. Cependant, cette méthode ajoute une étape de calcul non négligeable qu'il faudrait exécuter à chaque modification de surface.

- Algorithme Trans-Voxel

Actuellement, l'application développée se base sur une grille de voxel, c'est à dire une grille uniforme. Une manière de gagner en performance serait d'utiliser plusieurs grilles, de résolution variable, qui seraient détaillées uniquement aux zones éditées par l'utilisateur. Ainsi, toutes les zones non éditées ne coûteraient pas en performance. Cependant, les Marching Cubes seuls sont très difficile à adapter à un ensemble de grilles de résolutions différentes. L'algorithme trans-voxel permet de créer des transitions géométriques pour les Marching Cubes à l'interface de deux grilles différentes.

- Algorithme Level of Detail

Un algorithme de level of detail permettrait d'améliorer les performances au niveau du rendu graphique. Il s'agirait de simplifier l'affichage de la géométrie créée qui se situe loin de l'utilisateur.

- **De nouvelles interactions**

De nouvelles interactions complexes pourraient être envisagées pour ajouter des fonctionnalités à l'application développée. Par exemple, une possibilité d'utiliser un miroir serait intéressante. Il s'agirait de choisir un plan de symétrie dans la zone d'édition qui offrirait de nouvelles possibilités de sculpture.

## IV. Conclusion

Ce projet de réalité virtuelle a été l'occasion d'apprendre à manipuler les technologies de réalité virtuelle, mais également à réutiliser des connaissances théoriques provenant de notre parcours.

L'aboutissement de ce semestre est un solide prototype d'application qui pourrait être réutilisé par exemple pour un environnement interactif pour un jeu vidéo en réalité virtuelle, ou encore pour proposer une façon de modéliser différente des solutions existantes.