

Treball de Fi de Grau

Enginyeria en Tecnologies Industrials

Simulació d'interaccions amb imatges mèdiques mitjançant dispositius hàptics

MEMÒRIA

Autor: Raimon Gaspar

Director: Toni Susin

Convocatòria: Febrer 2018



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resum

Aquest projecte parteix del treball realitzat per en Joan Fons Sànchez, “Inspecció interactiva i immersiva de models volumètrics. Aplicació diagnosi mèdica”.

En aquest projecte desenvoluparem noves eines de Realitat Virtual per a la interacció amb models volumètric mèdics. D'aquesta manera es vol millorar la diagnosi mèdica i a la vegada proporcionar noves eines de formació per als estudiants. En concret, s'ha utilitzat el dispositiu hàptic Phantom Omni que ens dona un efecte d'immersió simulant el tacte amb objectes virtuals.

El dispositiu hàptic utilitza uns motors i resistències per generar forces de retorn i simular la interacció de l'usuari amb l'entorn virtual. A més a més ens permet moure i orientar-nos de forma precisa en un espai 3D, tasca molt difícil amb un ratolí i teclat convencionals.

El projecte s'ha desenvolupat mitjançant el motor de videojocs multi plataforma Unity de Unity Technologies. Aquest motor disposa de una interfície i un llenguatge de programació senzills, a més de diferents llibreries i plugins que faciliten la integració del dispositiu hàptic.

Sumari

RESUM	0
SUMARI	3
1. GLOSSARI	5
2. INTRODUCCIÓ	6
2.1. Objectius del projecte.....	6
2.2. Abast del projecte	6
3. REALITAT VIRTUAL	8
3.1. Historia de la Realitat Virtual.....	9
3.2. Dispositius de Realitat Virtual	11
3.3. Phantom Omni	13
4. UNITY	15
4.1. Components bàsics.....	18
4.2. Plug-in Hàptic	20
4.2.1. Scripts bàsics.....	20
4.2.2. Modes de interacció	21
4.2.3. Propietats dels objectes	21
4.2.4. Efectes de l'entorn.....	22
5. VISUALITZACIÓ	24
5.1. Imatge mèdica.....	24
5.2. Renderitzat del volum	24
5.3. Funció de transferència.....	26
6. DESCRIPCIÓ DE LA APLICACIÓ	27
6.1. Interfície gràfica.....	27
6.2. Controls bàsics.....	29
6.3. Editor de la funció de transferència	29
6.4. Botons de moviment i rotació.....	30

6.5. Modes Normals	30
6.6. Modes Haptics	31
7. REPTES DEL PROJECTE	35
8. SEGÜENTS PASOS	36
8.1. Exportació de l'aplicació.....	36
8.2. Escala Hounsfield	36
8.3. Mallat del volum	37
9. CONCLUSIONS	40
BIBLIOGRAFIA	41
Referències bibliogràfiques	41
Bibliografia complementària	41

1. Glossari

Dispositiu Hàptic: Aparell de Realitat Virtual que simula el tacte amb objectes virtuals.

Phantom Omni: Dispositiu Hàptic desenvolupat per l'empresa Geomagic.

Imatge mèdica: Conjunt de tècniques de representació de parts interiors d'un cos.

Unity: Motor multi plataforma per al desenvolupament de Videojocs.

Game Object: Entitat bàsica de Unity a la que se li poden afegir components.

Asset: Qualsevol arxiu compatible amb unity. GameObject, imatges, vídeos, escenes, scripts, etc.

Editor de Unity: Entorn amb interfície d'usuari on desenvolupem les diferents escenes del "Joc".

C#: Llenguatge de programació.

Pluguin: Complement que afegeix una nova funcionalitat al programa.

Script: Arxiu que conté les ordres o instruccions que s'han d'executar.

Frame: Cada actualització que fa el motor de videojocs.

Vòxel: Unitat cúbica mínima d'un objecte tridimensional.

Crop Range: Capsa cúbica que marca els límits de la visualització. Només es visualitzaran els vòxels dins d'aquest cub.

2. Introducció

2.1. Objectius del projecte

En aquest projecte explorarem eines de Realitat Virtual a partir del projecte d'en Joan Fons que tracte sobre la visualització d'imatges mèdiques. La imatge mèdica es defineix com el conjunt de tècniques i processos de creació de representacions 2D o 3D de parts interiors d'un cos per a la posterior anàlisi clínica i intervenció mèdica. Joan Fons va crear un sistema de representació d'aquests models amb el motor de Unity i va integrar l'aparell de Realitat Virtual HTC Vive per a interaccionar amb els models.

El nostre objectiu es introduir en l'entorn Unity el dispositiu Hàptic Phantom Omni per donar un altre efecte d'immersió a l'usuari . Guanyant en precisió i millorar la forma d'interactuar amb el volum. Amb el propòsit d'experimentar els límits del nou dispositiu s'ha eliminat el dispositiu HTC Vive per augmentar l'eficiència i reduir els requeriments de Hardware.

En resum, es vol dissenyar un software senzill i intuïtiu per poder manipular les dades d'imatge mèdica des d'una altre perspectiva, el sentit del tacte. D'aquesta manera es pretén millorar la diagnosi mèdica i proporcionar noves eines en l'àmbit de la ensenyança.

2.2. Abast del projecte

La realització del projecte ha requerit d'una primera fase d'aprenentatge de les eines i programes a utilitzar. En primer lloc hem pres contacte amb el motor de videojocs Unity3D i el seu llenguatge de programació C#.

Un cop ens hem familiaritzat amb l'editor de Unity hem examinat el treball d'en Joan Fons. Per tal de treballar còmodament i per la limitació de potència dels ordinadors, s'ha eliminat totes els parts de codi referents al dispositiu HTC Vive. Això ha portat



molt de temps però ens ha ajudat en un altre etapa molt important que es la d'entendre com funciona el codi de visualització del volum. És imprescindible entendre l'algoritme de visualització per poder crear noves formes d'interacció amb el volum.

D'altra banda el projecte s'ha centrat en explorar les possibilitats del dispositiu Phantom Omni. Gràcies a un plugin gratuït per a connectar dispositius hàptics a Unity podem utilitzar el nostre aparell. El plugin inclou vaires escenes de d'exemple que ens faciliten molt l'aprenentatge.

Finalment hem instal·lat el plugin dins del projecte de visualització d'imatges mèdiques. D'acord amb l'objectiu del projecte s'ha dissenyat un sistema de interacció i manipulació del volum amb el nostre aparell de Realitat Virtual.

3. Realitat Virtual

La realitat virtual es pot definir com la representació d'objectes o escenes amb l'objectiu d'aparentar ser reals. Aquesta representació es dur a terme amb sistemes informàtics i poden estar basades en entorns real, imaginaris o modificacions de la realitat. Mitjançant diferents dispositius de immersió sensorial (ulleres, cascos, guants, etc) es capten els moviments de l'usuari i es reproduïxen en l'entorn virtual. D'aquesta manera s'aconsegueix que l'usuari tingui una millor percepció de realitat del que està veient i, en conseqüència, la desconexió de la realitat existent fora del entorn virtual.



Figura 3.1: Usuari experimentant la Realitat Virtual

La Realitat Virtual es una de les revolucions tecnològiques més importants d'aquest segle. El seu potencial encara està per explorar, ja que, és una tecnologia relativament recent, al igual que els centres formatius dedicats a aquesta especialitat. Algunes de les múltiples aplicacions actuals es poden trobar en el món de l'educació, medicina, videojocs, turisme, arts plàstiques o simplement per visualitzar vídeos 360. De igual manera existeixen una multitud de softwares específics dedicats a aquestes especialitats com "3D Organan VR Anatomy", "Apollo 11 VR" o "VR cities". Un dels problemes actuals per fer arribar aquesta tecnologia al públic és l'elevat preu dels dispositius. Per sort, cada cop surten més alternatives econòmiques.



3.1. Historia de la Realitat Virtual

Com es sovint en les noves tecnologies les primeres aparicions les trobem en la ciència ficció. En el 1935 l'americà Stanley G. Weinbaum va escriure un llibre titulat "Las gafes de Pigmalió" on ens parla d'unes ulleres a través de les quals es pot experimentar el món amb olor, gust i tacte.



Figura 3.2: Sensorama.

Però el primer dispositiu real no va arribar fins el 1950 quan Morton Heilig va construir el prototip *Sensorma*, amb el qual es podien visualitzar imatges en 3 dimensions essent estimulat amb els sentits del olfacte, oïda i tacte. Al 1968 Ivan Sutherland crea el primer dispositiu de realitat virtual y augmentada, un casco conegut com HMD (Head-Mounted Display), però encara era molt elemental.

Diversos prototips es van anar creant durant els anys posteriors com el “videoplace”, “Super Cockpit” o el programa “Apen Movie Map” de MIT per a la simulació virtual de la ciutat de Aspen. En general es feien amb objectius militars, mèdics o per a simulacions.

Al 1991 es produeix un dels primers llançaments comercials, el Sega Genesis VR per la consola Megadrive. Posteriorment la NASA va dissenyar un sistema per conduir vehicles a Mart en temps reals. Al 2007 Google apareix el conegut “Google Street View” i al 2010 el revolucionari “Oculus Rift” comprat per Facebook al 2014. Al 2016 apareix el HTC Vive Steam VR que inclou un sensor de posició.



Figura 3.3: Sega Genesis

Actualment hi ha centenars de empreses desenvolupant aquest tipus de productes i generant milers de llocs de treball. Les més importants son Facebook, Google, Apple, Amazon, Microsoft, Sony i Smamsung.



3.2. Dispositius de Realitat Virtual

Alguns dels dispositius d'immersió que trobem actualment en desenvolupament són els següents:

Gafes amb pantalles. Aquest són els dispositius més populars i em podem trobar una gran varietat com Oculus Rift, Playstation VR o HTC Vive. Solen anar acompanyats de comandaments que controlen la posició de la ma i permeten interactuar amb el entorn virtual.



Figura 3.4: Oculus Rift.

Gafes per al mòbil. Aquesta es una solució més econòmica que aprofita el hardware i tecnologia que tenen els telèfons actuals. Per exemple, Gear VR i Google Cardboard.



Figura 3.5: Google Cardboard

Plataformes Omnidireccionals. L'usuari es pot moure lliurement per aquestes plataformes i a la vegada es moure en el món virtual.



Figura 3.6: Plataforma Omni

Hand tracking. Gràcies a dispositius com el LEAP Motion poden captar amb precisió els moviments de les nostres mans mitjançant LEDs. O dispositius com Gloveone que capten el moviment amb un guant.



Figura 3.7: LEAP Motion

Dispositius hàptics. Serveixen per simular el tacte amb els objectes. En aquest projecte s'ha utilitzat el dispositiu Phantom Omnium, el qual, es detallarà més endavant.



3.3. Phantom Omni

La percepció hàptica és la percepció sensorial dels esser vius que permet percebre estímuls mecànics. Es tracta de reconèixer i interactuar amb objectes i formes mitjançant el sentit del tacte.

La tecnologia hàptica es basa en incorporar el sentit del tacte en aplicacions computacionals mitjançant forces o sensacions tàctils. Això s'aconsegueix gràcies a dispositius tàctils amb diferents sensors i que permeten manipular objectes tridimensionals virtuals com ho faríem amb objectes reals..

Phantom Omni és un dispositiu hàptic portable desenvolupat per Sensable Technologies. L'aparell pot posicionar-se per un espai tridimensional de 15 centímetres d'amplada, 12 d'alçada i 7 de profunditat amb 6 graus de llibertat. Gràcies a uns sensors de posició i orientació controlem el dispositiu amb una resolució de 0,055 mil·límetres aproximadament.



Figura 3.8: Aparell Phantom Omni

El dispositiu utilitza motors que creen forces de retorn al usuari per simular el tacte i la interacció amb objectes virtuals. Aquestes forces tenen 3 graus de llibertat en el nostre model.

Les llibreries per utilitzar Phantom Omnium estan en el llenguatge C++, per tant, es necessita cert nivell de programació per utilitzar-les. D'altra banda podem utilitzar plugin per desenvolupar codi en altres entorns i amb altres llenguatges. En el nostre cas utilitzarem el plugin desenvolupat per la "Glasgow School of Art".

Actualment és comercialitza amb el nom de *Touch* per la empresa Geomagic. Les nombroses aplicacions del mecanisme les podem trobar en el àmbit de la medicina, cirurgia, rehabilitació, aprenentatge, modelatge 3D, fabricació de joguines, joies o desenvolupament de jocs.



4. Unity

Per a realitzar aquest treball s'ha utilitzat el motor de videojocs Unity de Unity Technologies. Un motor de videojocs es un entorn de treball dissenyat per a la creació d'aquests i que inclou el següents elements:

- Assets: Són els elements que seran introduïts al videojoc i inclouen models 3D, textures, materials, animacions, scripts o sons.
- Interfícies de programació (API): Sistema de rutines, protocols i eines desenvolupar l'aplicació.
- Procés de Render: És el procés en el qual es posen a la pantalla els elements que han de ser visibles com poden ser els assets.
- Objectes: Els Objectes del nostre joc amb diferents atributs.
- Il·luminació: Els motors de videojocs solen incloure els seus propis sistemes de il·luminació.
- Sistemes per a manejar el so i les pistes d'àudio.
- Sistema de física.



Figura 4.1: Logotip Unity

Unity es utilitza per a la creació de jocs 3D o 2D de alta qualitat per a mòbil, escriptori, consoles o Web. Unity és un dels motors més utilitzats en tot el món gràcies al seu suport per a múltiples plataformes, una interfície amigable per a usuaris no experts en programació i perquè té una versió gratuïta. El llenguatge de programació amb el que treballem sobre Unity és el C#.

Entre les plataformes objectius podem trobar Windows, Mac, Android, iOS, WebGL, Smart TV, PS4, Xbox, Oculus Rift o HTC Vive. A més a més existeixen una multitud de pluguins com el que explicarem més endavant per a la utilització del Phantom Omnium. Editor

L'editor de Unity és l'eina que ens permet editar l'escena del Joc. Les aplicacions que construïm estan formades per diverses escenes que es connecten entre si. Per exemple podem tenir una escena de "Menú Principal" que passa a una altra escena quan cliquem al boto de "Iniciar Joc".

Els elements principals del Editor són els següents:

- Hierachy: Aquí podem trobar els diferents Objectes que hi ha a la escena. Es poden organitzar per jerarquies per a mantindre un ordre i localitzar-los més fàcilment. Un dels elements principals és l'objecte Camara encarregada de visualitzar l'escena.
- Scene view: Aquesta finestra ens permet veure els objectes que hi ha a l'escena. És un espai 3D per el qual podem navegar i seleccionar els diferents objectes.
- Game: Quan cliquem al botó de "Play" de l'editor es visualitzarà a la finestra Game l'escena, des de la perspectiva de la Camara.
- Project Window: Mostra els Assets del nostre projecte. Podem crear diferents carpetes per organitzar les nostres imatges, materials, scripts, clips d'àudio, etc.
- Inspector: Des d'aquesta finestra podem afegir i modificar diferents



components dels objectes com per exemple afegir un Script, una textura o un Rigidbody per afegir-li propietats físiques com la massa, pes, etc. Per que l'inspector ens mostri un objecte tenim que seleccionar-lo des del Hierachy o el Scene View.

- Consola: La finestra consola és una de les més importants de l'editor. Aquí és on es mostraran el errors que pugui haver en el nostre codi. Mitjançant al comanda `Debug.Log("missatge")` enviar un missatge des del nostre codi i es mostrarà a la consola.

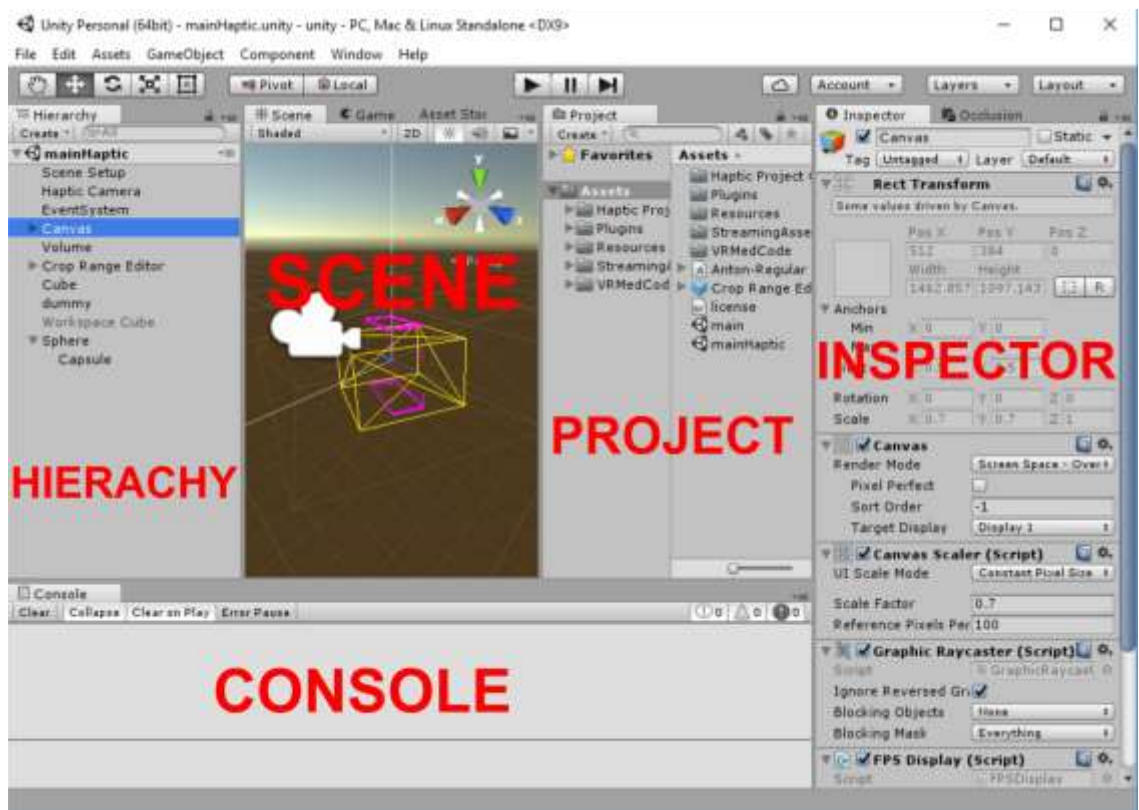


Figura 4.2: Interfície del Editor Unity.

4.1. Components bàsics

Unity ens proporciona una multitud de components per afegir als nostres GameObjects:

- Transform: Component per situar l'objecte a l'espai de l'escena. Tenim 3 coordenades per la posició, 3 per a la orientació i 3 per escalar la mida del objecte.
- MeshFilter: Des d'aquesta finestra seleccionem la malla o geometria dels nostres assets. També es pot crear la malla des d'un script.
- MeshRenderer: Mitjançant la malla del MeshFilter i el component Transform, renderitza l'objecte perquè es visualitzi. Podem afegir diferents textures.
- Light: Utilitzat per a la il·luminació de l'escena.
- Colliders: Serveix per delimitar el contorn del objecte i pugui col·lisionar amb altres colliders.
- Rigidbody: Permet afegir propietats físiques al objecte com la massa, gravetat, velocitat o fricció.
- Scripts: Mitjançant el llenguatge de programació C# escriurem el nostre codi per modificar l'escena segons les instruccions adients. En el nostre cas utilitzarem l'entorn de desenvolupament integrat (IDE) Microsoft Visual

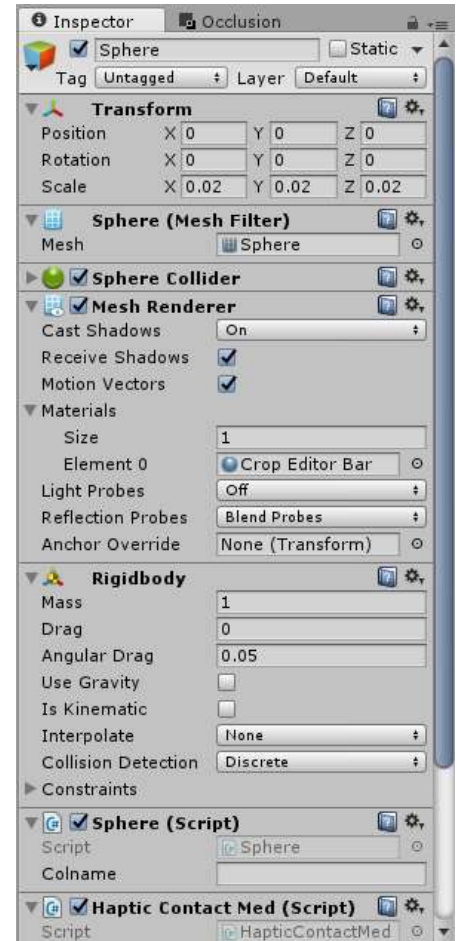


Figura 4.3: Components d'un GameObject.



Studio. Quan creem un script automàticament se'ns crea un codi amb una classe que conté dos funcions. Això és degut a que la classe es una subclasse de "MonoBehaviour" on podem trobar moltes altres funcions útils. A continuació les mes utilitzades en el projecte:

- Start: Instruccions que volem que s'executin quan creem l'objecte o quan s'inicia l'escena..
- Awake: Igual que Start però avança de l'inici de l'escena.
- Update: Instruccions que volem que s'executin cada frame.
- LateUpdate: Instruccions que volem que s'executin al final del frame.
- OnDisable: Instruccions que volem que s'executin quan desactivem el script.
- OnDestroy: Instruccions que volem que s'executin quan destruïm el GameObject.

```
1  [?] using UnityEngine;
2  [ ] using System.Collections;
3
4  [ ] public class Demo : MonoBehaviour {
5
6      // Use this for initialization
7  [ ] void Start () {
8
9      }
10
11     // Update is called once per frame
12 [ ] void Update () {
13
14     }
15 [ ] }
16
```

Figura 4.4: Script bàsic de Unity.

- Components IU: Components com canvas, panells, text o Imatge per a la interfície gràfica del nostre joc.

4.2. Plug-in Hàptic

Per tal d'integrar el dispositiu hàptic dins de l'entorn de Unity, s'ha utilitzat el pluguin gratuït "*Unity 5 Haptic Pluguin for Geomagic OpenHaptics 3.3*" de la *Glasgow School of Art*. És pot descarregar gratuïtament des de la pròpia botiga d'assets de Unity. Bàsicament el que ens permet és programar amb C#, ja que, el llenguatge original del dispositiu és C++. [1]

La instal·lació es molt senzilla, un cop descarregat i importat el plugin només hem de seguir les passes del PDF que trobem a la carpeta que se'ns ha generat al Editor. Tanmateix tenim algunes escenes d'exemple que ens ajudaran a comprendre el funcionament del scripts.

4.2.1. Scripts bàsics

Per a la correcte utilització del plugin hem de crear un script amb una estructura determinada. Per a la inicialització del programa:

1. Inicialitzem l'aparell.
2. Fixem l'espai de treball del dispositiu hàptic i el mode d'interacció.
3. Fixem les geometries del objectes que volem tocar.
4. Carreguem les dades al dispositiu.

Per a cada frame:

1. Actualitzem l'espai hàptic
2. Obtenim la informació adient de l'escena i la posició de l'aparell.
3. Actualitzem l'escena i els dispositiu, si es necessari.



4.2.2. Modes de interacció

El pluguin permet funcionar l'aparell en diferents modes:

- Mode 0. Simple Contact Mode: Contacte amb els objectes virtuals.
- Mode 1. Object Manipulation Mode: Contacte i manipulació dels objectes virtuals.
- Mode 2. Custom Force Effects Mode: Simulació de forces tangencials i vibracions.
- Mode 3. Puncture Mode: Simulació de injeccions i punxades.

Mode	name	Contact	Manip.	Env. Force Effects	Mechanical Force Effects	Puncture Injection
0	Simple Contact	Yes	No	Yes	No	No
1	Object Manipulation	Yes	Yes	Yes	No	No
2	Custom Force Effect	Yes	No	Yes	Yes	No
3	Puncture	Yes	No	Yes	No	Yes

Figura 5.5: Taula amb els modes del pluguin

4.2.3. Propietats dels objectes

Quan creem un objecte en el nostre entorn virtual hem d'assignar-li algunes propietats.

Propietats bàsiques:

- Stiffness: Rigidesa de la superfície.
- Damping: Elasticitat de la superfície.

- Static Friction: Simulació de fricció estàtica.
- Dynamic Friction: Simulació de fricció dinàmica.
- Pop Through: Controla la força necessària per a travessar una superfície.

Propietats avançades:

- Mass: Massa del objecte.
- Fixed Object: Booleà que indica si l'objecte és fix o no.

Propietats per a objectes amb efectes de força:

- Tangencial Stiffness: Simulació de la duresa d'una superfície quan contacte amb un disc rotatiu.
- Amortiment tangencial: Controla la elasticitat de la superfície.

Propietats per a la punció:

- Punctured Static Friction: Simulació de fricció estàtica quan punxem un objecte.
- Punctured Dynamic Friction: Simulació de fricció dinàmica quan punxem un objecte.

4.2.4. Efectes de l'entorn.

D'altre banda també podem afegir efectes al entorn o espai virtual sense tenir en compte els objectes.

- Constant Force Effects: Simulació forces constants al dispositiu en funció de les variables que vulguem.
- Friction Force Effects: Fricció constant a l'entorn.
- Spring Forces Effects: Simulem un molla en un punt de l'espai i el dispositiu



es atret cap a ell.

- Viscosity Force Effect: Simulació de viscositat.
- Motor Vibration: Simulació d'una vibració provocada per un eina amb motor.
- Contact Vibration Force: Similar a Motor Vibration però només quan estem en contacte amb una superfície.
- Tangencial Force: Simulació d'una força tangencial provada per un disc rotatiu en contacte amb una superfície.

5. Visualització

Per a realitzar aquest projecte hem partit del treball d'en Joan Fons, *Inspecció interactiva i immersiva de models volumètrics. Aplicació diagnòstic mèdica*. En el treball es va integrar un algoritme de visualització de models volumètrics dins del motor de Unity. [2]

5.1. Imatge mèdica

La imatge mèdica es un camp molt connectat a la enginyeria i la física. Al llarg de la història s'han desenvolupat multitud de tècniques per aconseguir imatges del interior del cos humà com les ecografies o ressonàncies. La forma de representar-les solen ser en suports 2D (radiografies) però en els últims anys s'han desenvolupat softwares per a la representació en 3D.

El següent pas el trobem en la Realitat Virtual que permet a l'usuari un efecte d'immersió en l'entorn virtual. Combinant els models 3D i els aparells de RV s'aconsegueix una millor interacció amb el volum i, per tant, una millor comprensió del que s'està veient. Aquesta nova eina pretén aportar noves perspectives tant en el camp de la medicina com en el de l'ensenyança.

5.2. Renderitzat del volum

La representació tridimensional es fa mitjançant models de volum. Generalment tenim un conjunt de dades que delimitant un espai, amb unes propietats o mesures en cada punt. Cada punt del espai que conté informació s'anomena vòxel que es pot entendre com l'extensió 3D del píxel. En la medicina les mesures es prenen, per exemple, amb ressonàncies, tomografies o ecografies. La visualització del volum s'ha integrat Unity3D mitjançant un model de vòxels en el qual associem als possibles valors dels vòxels a un color.



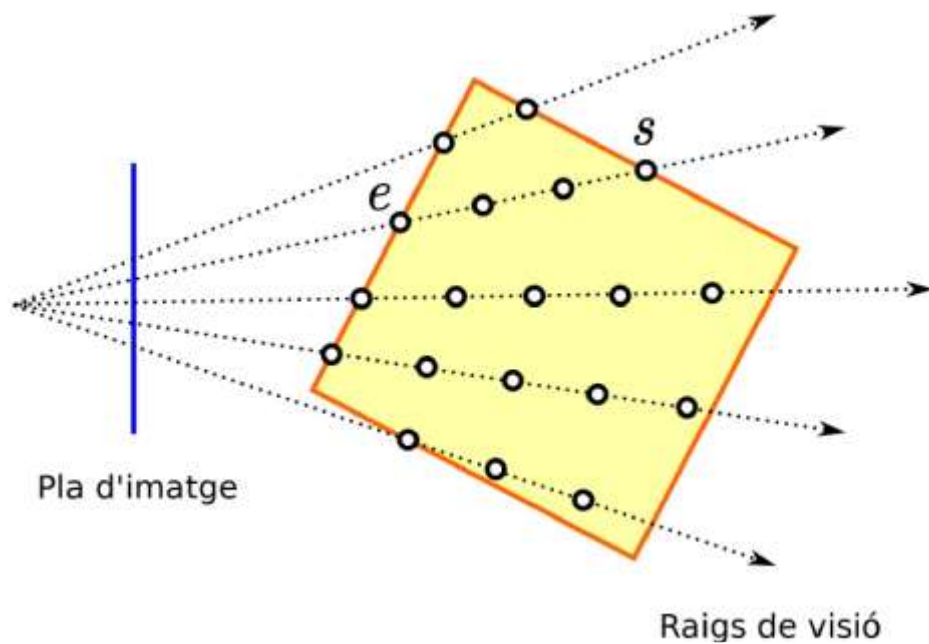


Figura 5.1: Representació d'un algorisme de raycasting. Es traça un raig de visió des de cada píxel i es prenen mostres del model a punts discrets.

Per a renderitzar el model de vòxels s'utilitza un traçat de raigs de visió:

- i. En primer lloc es recorren les dades del model volumètric i es calcula el valor acumulat. Com els punts que marquen els raigs sovint no coincideixen amb els vòxels on hi ha dades, es fa una interpolació amb els punts més pròxims.
- ii. Es calcula el gradient en cada punt i s'emmagatzemen els valors. Aquestes dades s'utilitzen posteriorment per al càlcul d'il·luminació.
- iii. Mitjançant una funció de transferència que s'ha definit prèviament, s'assigna un color a cada valor de propietat.
- iv. Per a un resultat més realista, s'utilitzen model de il·luminació. En el nostre cas el mètode del gradient.

- v. Per últim, es combinen els colors de cada punt del raig de visió per obtenir el valor final.

5.3. Funció de transferència

Com ja hem dit anteriorment, el model volumètric conté un valor per cada punt però no es suficient per a la visualització. Per això hem de definir una funció de transferència. Aquesta funció ens associa un color per a cada valor. El color està en format RGBA, que a diferència del format estàndard RGB, podem assignar un quart valor “alpha” que ens modificar la opacitat.

La funció de transferència pot tenir rangs diferents per a cada interval de valors. Per tant podrem calorejar el núvol de punts del model volumètric i distingir els diferents elements que el componen segons ens convingui.

Per exemple, podem assignar el color blanc als valors de propietat més alts, es a dir, els ossos. Això es degut a que els ossos tenen una densitat molt elevada i aquesta es proporcional al valor de propietat. Els valors més baixos els assignem de color carn ja que tenen valors de densitat menors.



6. Descripció de la aplicació

6.1. Interfície gràfica

El disseny de l'aplicació inclou la creació d'una interfície amb menús per comunicar-nos amb el programa i poder seleccionar els diferents modes o funcions. Malgrat que l'objectiu principal es centre en interactuar amb el volum mitjançant el dispositiu Phantom Omni, hem volgut incloure una alternativa per navegar i comunicar-nos a través del ratolí i el teclat.

Per tant, tindrem les opcions "Normal" i "Hàptic" que podrem canviar en qualsevol moment prement la tecla "SPACE" del nostre teclat. Automàticament canviarem a l'escena adient i es modificaran els controls i menús que convinguin.

En el projecte original de Unity hi havia inclosos alguns menús però com estava pensat per funcionar amb HTC Vive s'han hagut d'eliminar i dissenyar de nou. D'altre banda també podíem trobar formes de modificar el volum des de l'editor de Unity però també s'han eliminat. Això es degut a que volem un software exportable, es a dir, que no necessitem instal·lar Unity per a poder utilitzar-lo.

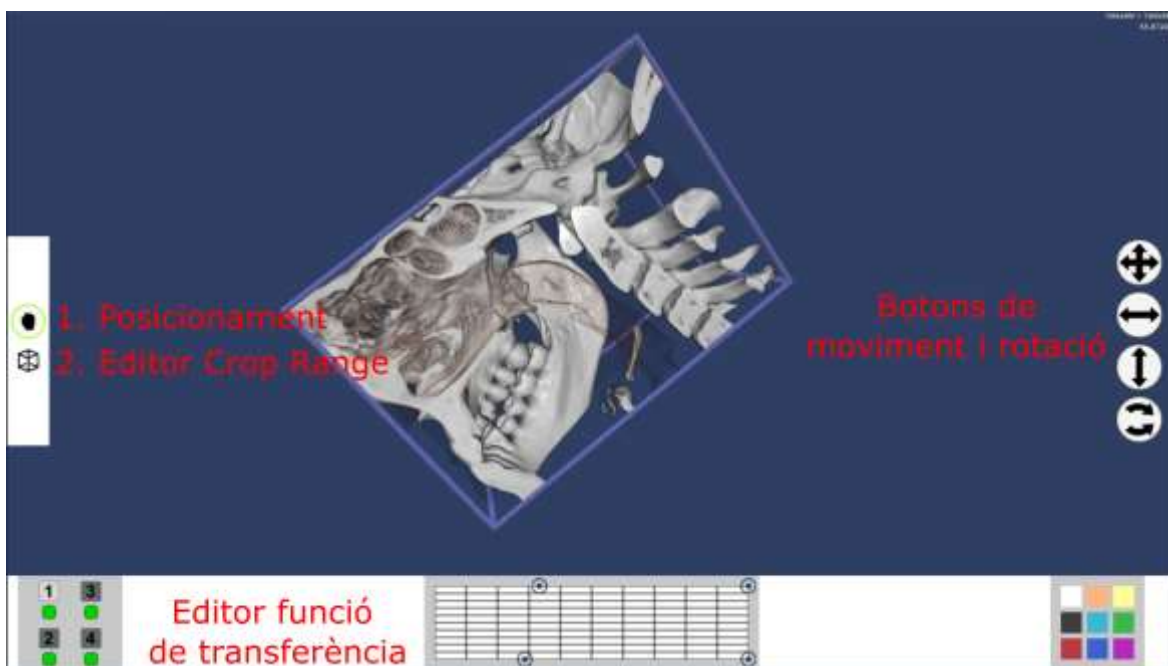


Figura 6.1: Interfície d'usuari, Mode Normal

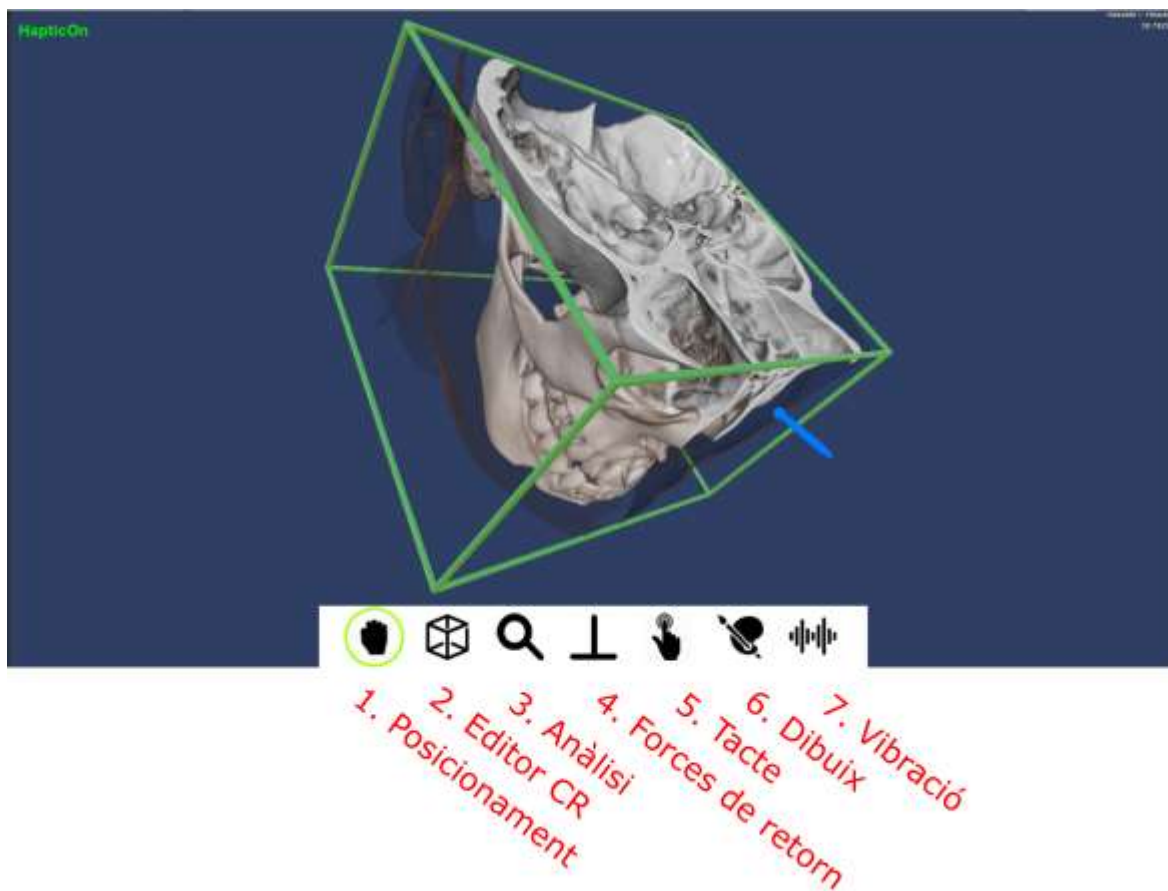


Figura 6.2: Interfície d'usuari , Mode Hàptic



6.2. Controls bàsics

El controls s'han dissenyat perquè l'usuari tingui una experiència amb corba d'aprenentatge senzilla i intuïtiva.

Ratolí: Com a qualsevol aplicació es seleccionaran diferents botons de la pantalla amb el botó esquerra del ratolí. A més clicant el botó dret i desplaçant el ratolí podrem rotar el volum virtual en qualsevol moment. També podrem fer rodar el "scroll" del ratolí per fer zoom i veure el volum amb més detall.

Teclat: Prement el botons numèrics del teclat podrem seleccionar els diferents modes i funcions que s'explicaran més endavant. Tanmateix podem utilitzar el botó "CTRL" per a modificar el la caixa que limita el renderitzat del volum (Crop Range), que s'explicarà més endavant..

Phantom Omni: Gràcies a aquesta eina podem desplaçar-nos en tres direccions per l'espai virtual. Els dos botons del dispositiu serviran per fer dues funcions diferents. El botó primari tindrà una funció dependent del mode en el que estiguem. El volum està envoltat per una capsula que podem tocar amb el hàptic. Per tal de entrar en el seu interior o al inrevés, utilitzarem el botó secundari per amagar la capsula.

6.3. Editor de la funció de transferència

L'editor de la funció de transferència ens permet crear, eliminar o modificar els diferents rangs de la funció de transferència. Aquesta funció de l'aplicació només està disponible en el mode "Normal".

En la finestra principal podrem veure diferents botons amb números. Cada numero correspon a un rang de la funció de transferència. A més tenim el botó "+" per a poder crear nous rangs.

Si cliquem a un dels botons numèrics se'ns obrirà una altre finestra amb diferents

opcions:

- **Mostrar/amagar:** Botó per amagar o mostrar el rang seleccionat. Estarà de color vermell per amagar i verd per mostrar.
- **Gràfic:** Prement els vèrtex del trapezi modificarem el rang. El eix vertical correspon al valor d'opacitat del color i esta normalitzat. El eix horitzontal, també normalitzat, ens permet seleccionar l'interval de valors que volem que es visualitzin. Els vòxels amb un valor de propietat normalitzat dins dels rangs es visualitzaran a la pantalla.
- **Color primari:** Permet seleccionar el color primari del rang de transferència

6.4. Botons de moviment i rotació

A la part dreta de la pantalla, en el mode "Normal", podem trobar 4 botons. El primer ens permet moure en dues direccions en el pla de la pantalla. Els 3 següents rotaran el volum en els eixos X,Y y Z respectivament. Les rotacions son referents als eixos del volum, si rotem amb el ratolí els eixos són paral·lels al pla de la pantalla.

6.5. Modes Normals

Com hem comentat anteriorment, l'aplicació pot ser utilitzada sense necessitat d'un dispositiu hàptic. L'experiència serà més limitada.

1. **Posicionament del Volum:** Prement el boto esquerra del ratolí seleccionem un punt del mode. A continuació desplaçarem el ratolí per i com a conseqüència el volum es mourà en direcció vertical o horitzontal.
2. **Editor Crop Range:** Anomenem Crop Range a la capça que envolta el volum. El sistema només visualitza els vòxels que estan al interior d'aquest tub. Si seleccionem amb el botó dret una d'aquestes cares podem



desplaçar-les i modificar els límits de la caixa.

6.6. Modes Hàptics

Gràcies al dispositiu Phantom Omni podem moure'ns per l'espai tridimensional de manera molt còmode i precisa. D'aquesta manera l'exploració e interacció del volum es molt més còmode i senzilla.

1. Posicionament del volum:

Amb la nostra eina de realitat virtual podem tocar la caixa que envolta el volum. Un cop estiguem en contacte amb la caixa podem prémer el botó primari del dispositiu i automàticament tindrem agafat el volum. Notarem una sensació de gravetat que simula el pes del volum. Movent el dispositiu podem posicionar el volum on vulguem, dins del espai hàptic.

2. Editor Crop Range:

Editar el Crop Range amb el ratolí pot ser un problema, ja que, algunes cares no es poden seleccionar perquè estan darrera d'altres. Amb el Phantom Omni, a part de ser molt més precís, podem seleccionar qualsevol cara. Quan enganxem una cara és produirà una força normal al pla. Per tant, harem de fer força per estirar o comprimir el cub.

3. Anàlisi:

Desplaçant el hàptic i seleccionant qualsevol punt, dins del espai del volum, podem obtindre el valor de propietat normalitzat del vòxel més proper. A més se'ns crearà un nou rang de funció de transferència que pintarà de color verd el vòxels que tinguin un valor de propietat similar. A la figura 6.3 podem observar com seleccionant el vòxels al interior d'una dent, es coloreja tota la dentadura.

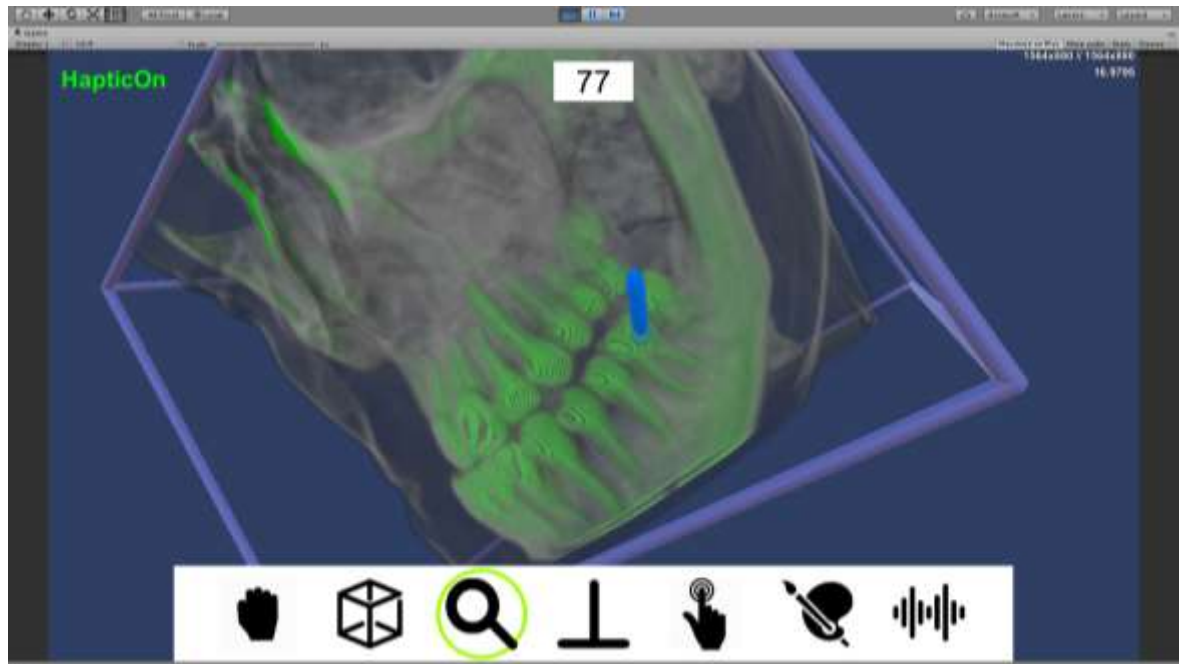


Figura 6.3: Exemple d'utilització del mode Anàlisi.

4. Forces de retorn:

Per a el funcionament d'aquest mode s'ha creat una algoritme per calcular un gradient en la posició del dispositiu hàptic en temps real. En funció de la direcció del gradient i del seu mòdul introduïrem una força al nostre dispositiu. D'aquesta manera es vol simular que "xoquem" amb el volum.

Per tal de que el dispositiu no es col·lapsi hem hagut de baixar la velocitat d'actualització d'aquestes forces. A més a més, per tal d'estabilitzar les forces fem una mitjà del gradient amb els veïns del vòxel sobre el que estem.

Fórmula utilitzada simplificada:

$$\text{Força a aplicar} = \text{Constant correctora} \cdot \text{Vector Gradient}$$

$$\text{Gradient} = (V.Mx - V.m_x, V.My - V.m_y, V.Mz - V.m_z)$$

$V.M$ = Valor de propietat màxim

$V.m$ = Valor de propietat mínim



5. Tacte amb plans normals: Per a que el dispositiu hàptic detecti un objecte, aquest ha de tenir un component “collider”. El volum està compost per milers de vòxels i no podem crear un GameObject amb un collider per a cadascun. Per tal de solucionar aquest problema s’ha creat un algoritme que pren mostres dels punts veïns de la posició a la que estem:

- i. Es defineix prèviament una funció de transferència i elegim el rang de valors que volem tocar. Dividim l’espai del workspace en un **espai discret**.
- ii. Seleccionem punts de l’espai discret que envolten la posició del dispositiu hàptic. Assignem a cada punt el valor de propietat del vòxel més proper.
- iii. Eliminem els punts amb un valor de propietat fora dels marges del rang de la funció de transferència.
- iv. En cada un dels punts vàlids creem un objecte pla amb un “collider”.
- v. Orientem els plans normals segons l’orientació de l’aparell hàptic.
- vi. Repetim des del punt 2 cada cop que canviem de posició.

Si apropem lentament el dispositiu al volum tindrem la sensació de que l’estem tocant. És important que l’eix del dispositiu el mantinguem normal a la superfície que volem tocar, ja que els plans amb “colliders” són normals a aquest eix.

Aquest mateix algoritme s’ha provat amb GameObjects cúbics o esfèrics. Degut a la possibilitat de que el nostre dispositiu que atrapat al interior d’un d’aquets GameObjects hem optat per utilitzar plans. A la figura 6.4 podem observar els plans normals de color negra.

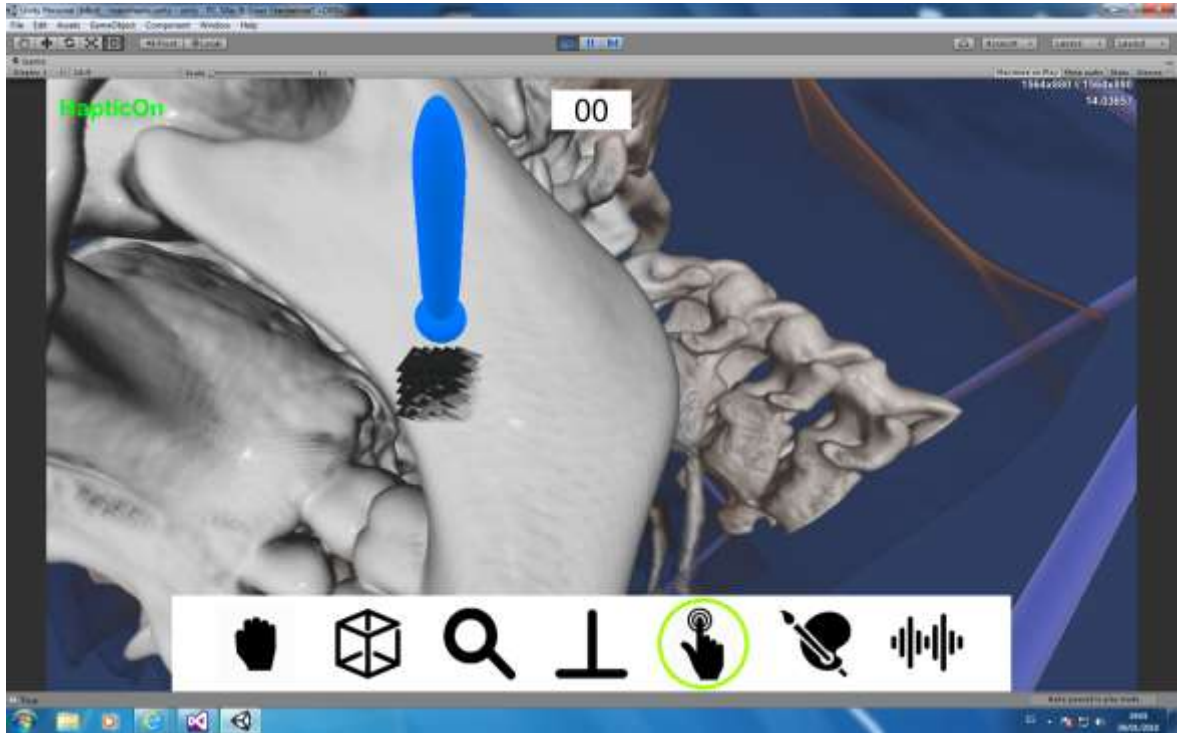


Figura 6.4: Exemple d'utilització del mode Tacte.

6. Dibuix:

Aprofitant que podem moure'ns per l'espai tridimensional amb gran facilitat s'ha dissenyat un mode de funcionament amb el qual podem dibuixar sobre el volum. D'aquesta manera podem marcar o senyalar les parts del volum que creem convenients. Amb el botó secundari podem esborrar les marques.

7. Vibració:

En aquest cas utilitzarem el mode vibració del pluguín. Prendrem mostres del valor de propietat en els punts per on es desplaçem amb el dispositiu Phantom Omnim. L'aparell entrarà en vibració amb una intensitat proporcional a aquest valor de propietat. D'aquesta manera podrem explorar el volum des d'una altra perspectiva sensorial diferent del sentit de la vista.



7. Reptes del projecte

El primer repte, com es evident, ha sigut entendre els scripts i algorismes de visualització i tractament de les dades del volum. Aquesta part es indispensable per al procés d'adaptació a la nova eina de Realitat Virtual, Phantom Omni. De igual manera hem hagut d'entendre el funcionament del motor Unity i el pluguin hàptic.

Durant el disseny de l'aplicació i els diferents algorismes ens hem trobat amb algunes dificultats que no comptàvem en un inici. Com ja s'ha comentat anteriorment el projecte que hem heretat estava pensat per ser utilitzar amb unes ulleres de realitat virtual i uns comandaments amb control de posició. Per tant s'han agut de dissenyar noves formes d'interacció amb el ratolí del ordinador.

Durant la creació de modes d'interacció amb el dispositiu veiem que la posició del volum i la seva mida son inestables. Entenem que això es conseqüència de que la posició del dispositiu va variant constantment i el sistema entra en un bucle i mai s'estabilitza. La solució trobada a aquest problema es utilitzar la funció `lateUpdate()` en comptes de `Update()`. D'aquesta manera ens assegurem de que els càlculs dins d'aquesta funció es realitzin al final del frame, posteriorment als càlculs de visualització i del hàptic.

Una altre dificultat trobada han sigut els canvis el color del volum. El volum no pot canviar la seva escala ni inserir-se dins d'un altre `GameObject`, ja que, entra en conflicte amb els càlculs de visualització i del gradient, modificant la textura. Quan agafem el volum amb l'aparell de Realitat Virtual, el pluguin ens posar el volum dins del `GameObject` que representa el dispositiu. Això provoca un canvi en l'escala i ,en a conseqüència, en la textura i color. Per evitar que això succeeixi s'ha creat un `GameObject` cúbic que es posiciona i s'orienta en funció del model i el `Crop Range`. Per tant, quan agafem el volum en realitat estem agafant aquest cub i posteriorment posicionem el volum.

8. Següents passos

Degut al temps limitat per realitzar el treball no s'ha pogut perfeccionar l'aplicació tant com voldríem. En aquest apartat proposarem algunes vies de millora.

8.1. Exportació de l'aplicació

Un dels inconvenients que té el projecte actual és que has de tenir instal·lat el Unity al ordinador. Durant la seva realització hem procurat que l'aplicació sigui el més independent possible de l'editor de Unity i també de les eines de Realitat Virtual. Idealment el projecte hauria de exportar-se en un arxiu executable. A partir d'aquest arxiu hauríem de poder obrir l'aplicació i poder carregar diferents volums i funcions de transferència.

Amb la mateixa línia de fer l'aplicació més flexible el mateix programa podria ser utilitzat amb diferents eines de Realitat Virtual o amb ninguna d'elles, segons el hardware del que disposi l'usuari o la seva finalitat.

8.2. Escala Hounsfield

La escala Hounsfield és una escala quantitativa que serveix per descriure els diferents nivells de radiodensitat dels teixits humans. Les unitats utilitzades són "Unitats Hounsfield" o "HU". El rang de valors útils d'aquesta escala abraça des de -1024 HU fins a 3071 HU.

Si la funció de transferència i valors els valors de propietat del nostre model volumètric són compatibles amb aquesta escala, podem seleccionar fàcilment els diferents teixits i pintar-lo de colors diferents. D'aquesta manera els rangs de la funció de transferència es crearien automàticament el que facilita molt la posterior feina d'exploració del volum.



Material	Hounsfield Unit
Air	-1000
Lung	-500 to -200
Fat	-200 to -50
Water	0
Blood	25
Muscle	25 to 40
Bone	200 to 1000

Figura 7.1: Taula Hounsfield

8.3. Mallat del volum

En aquest treball em explorat algunes possibilitat per a poder “tocar” el volum amb dispositius hàptics. El problema principal és que hi ha una gran quantitat de vòxels i no podem crear un mallat amb tants vòxels i no podem crear un mallat amb tants punts. Per a que el dispositiu hàptic pugui col·lisionar amb el volum es imprescindible un mallat. Nosaltres hem trobat solucions de baix rendiment però un mallat obriria moltes possibilitat d'interacció com per exemple la deformació. Per tant, proposem dues opcions:

1. Un mallat complet de baixa qualitat que utilitza només una part dels vòxels de tot el volum.
2. Un mallat amb una tècnica similar a la utilitzada en aquest treball. Mallant les zones properes a la posició del hàptic. Això permet utilitzar més vòxels, ja que la zona a mallar és menor.

9. Planificació i pressupost

9.1. Fases del projecte

El temps dedicat al projecte es pot desglossar en les tasques principals que s'han dut a terme durant els cinc mesos de duració. A continuació es mostra una taula es mostra les hores dedicades a cada tasca.

Fases	Hores
Definició i planificació del projecte	20
Aprenentatge Unity	10
Aprenentatge algoritme visualització	50
Aprenentatge Phantom Omni	20
Programació	120
Memòria	60
Reunions	5
Total	285

Taula 1: Planificació temporal del projecte



9.2. Pressupost

A continuació s'avaluarà el cost econòmic del projecte. Els dos apartats principals són el cost de recursos humans i el cost material.

Per a l'estudi del cost de recursos humans del projecte assignarem un preu per hora i utilitzarem el temps dedicat a cadascuna de les fases per calcular el preu total.

Preu per hora: 20 €/hora.

Hores dedicades: 285 hores.

Concepte	Cost estimat
Recursos humans	5700 €
Ordinador	600 €
Phantom Omni	2.200 €
Microsoft Office	450 €
Total	8950 €

Taula 2: Taula de costs del projecte

10. Conclusions

Aquest projecte ha sigut un gran repte ja que hem hagut d'aprendre a utilitzar el motor Unity 3D, entendre els algorismes de visualització del treball heretat d'en Joan Fonts i conèixer el funcionament del dispositiu Phantom Omni. Un cop finalitzat, podem afirmar que el motor de videojocs Unity es una eina molt útil i ideal per a crear experiències de Realitat Virtual i per a la seva investigació. És un programa molt potent del que podem trobar una gran quantitat de documentació a Internet. A més el pluguin hàptic ens ha facilitat molt la integració del aparell dins de Unity.

L'objectiu principal del projecte ha sigut investigar noves eines de Realitat Virtual relacionades amb el món de la salut. En concret hem elegit el dispositiu Phantom Omni i creiem que té un enorme potencial. Amb un ordinador convencional es complica moure'ns a través del espai i el volum, però amb aquest dispositiu podem fer-ho i a més podem sentir-lo. El resultat obtingut es molt satisfactori i en un futur pot ser una eina molt útil per els metges i estudiants de medicina.

Durant els mesos invertits en la realització del projecte s'ha treballat molt dur però amb moltes ganes. Ha servit per obtenir una gran quantitat de coneixements d'informàtica i per descobrir nous programes de molta utilitat. La Realitat Virtual és una revolució tecnològica i treballar al Centre de Realitat Virtual es una gran oportunitat per descobrir aquest món.



Bibliografia

Referències bibliogràfiques

- [1] *M. Poyade, M.Kargas, V.Portela, Digital design Studio (DDS). Haptic Plug-In for Unity. 2014. Glasgow School of Art, Glasgow, United Kingdom.*
- [2] *JOAN FONS SANCHEZ. Inspecció interactiva i immersiva de models volumètrics. Aplicacio diagnosi mèdica. Teball de Fi de Grau-UPC. 2017.*

Bibliografia complementària

Web Documentació Unity - <https://docs.unity3d.com/Manual/index.html>