TREBALL DE FI DE CARRERA

TÍTOL DEL TFC: Accessibilitat i videojocs

TITULACIÓ: Enginyeria Tècnica de Telecomunicació, especialitat Telemàtica

AUTORS: Adrià Solé Orrit
Albert Lozano Ciller

DIRECTOR: Toni Oller Arcas

DATA: 01 de juliol de 2011

Títol: Accessibilitat i videojocs

Autor: Adrià Solé Orrit, Albert Lozano Ciller

Director: Toni Oller Arcas

Data: 01 de juliol de 2011

Resum

En els últims anys la societat cada cop s'està envellint més gràcies als progressos de la medicina. L'esperança de vida ha augmentat bastant en la nostra zona i una part important de la nostra població té més de 60 anys.

Que l'esperança hagi augmentat també comporta algunes desavantatges, com per exemple problemes de salut. Molts cops és necessari l'ús de la rehabilitació, que ha millorat la seva efectivitat i cobertura als pacients gràcies a la integració de les noves tecnologies.

Actualment la crisis econòmica ha provocat retallades importants en la sanitat pública. S'han fet ajustaments de plantilla, i s'ha millorat l'eficiència per tal de reduir costos.

Una de les maneres de millorar l'eficiència és amb la tecnologia dels videojocs, poden millorar la cobertura als pacients i aportar la capacitat de fer més exercicis. Cada cop més s'està utilitzant per a la rehabilitació de pacients.

L'objectiu d'aquest treball es disposar d'una eina per la creació o pràctica d'exercicis de rehabilitació. Aquesta eina ha d'aprofitar les avantatges de les consoles de videojocs i reconèixer els moviments realitzats pel pacient per a poder fer un seguiment del progrés a distància.

Aquest treball també permetrà monitoritzar a un familiar o pacient que està a la seva casa, de manera que podrem reconèixer diferents patrons de comportament. Per exemple, si ha caigut, o ha obert el calaix de les medicines, etc..

El resultat d'aquests exercicis es publicaran en una xarxa social personalitzada, així els familiars o el personal sanitari podran fer un seguiment del pacient. També tindran la possibilitat de monitoritzar l'habitació del pacient de manera remota.

La aplicació resultant serà interactiva i de fàcil seguiment, que millorarà l'eficiència de la sanitat.

Title: Accessibilitat i videojocs

Author: Adrià Solé Orrit, Albert Lozano Ciller

Director: Toni Oller Arcas

Date: July, 1st 2011

Overview

Lately, the society is becoming older because of the medicine progress. The life hope has increased quite a lot in our area and a very important part of our population is more than sixty.

The increase of the life hope brings some disadvantages too, such as health problems. In several cases, we have to make use of the rehabilitation process that has improved its effectiveness towards the patients because of the new technologies integration.

Nowadays the economic crisis has caused important budget cuts in the public health system. In order to reduce economical costs, the efficiency has considerably improved and there is less staff.

A way of improving the effectiveness is with the videogames technology, they can not only improve the patient's range but also provide the ability of creating a largest amount of exercises. The technology is being used more often for patient's rehabilitation.

The main aim of this project is to dispose of a tool for the creation or practice of rehabilitation exercises. This tool should take advantage of the videogames and recognize all the patient movements so as to follow the progress in a long distance.

This project will allow to monitor either a family member or a patient which finds himself at home, so as to recognize many behavior patterns. For instance, when the patient has fallen or whether he has opened the medicine drawers…

The exercises results will be published in a personalized social web and both the family members and the health personnel will be able to look after the patient. They will also have the chance to monitor the patient room in a remote way.

The final application will be interactive and it should be an easy way of taking care of the patients. In other words, the health efficiency will be much better than before.

ÍNDEX

ÍNDEX D'IMATGES

ÍNDEX DE TAULES

INTRODUCCIÓ

La societat actual tendeix a envellir-se, molts cops gràcies a l'avanç de les tecnologies en el sistema sanitari. Però l'envelliment de la població no sempre és bo. Els recursos són limitats, i més ara que estem vivint una crisis econòmica que ha afectat al sistema sanitari també.

L'ajustament de plantilla és un dels resultats d'aquesta retallada. Això és un inconvenient, la gent viu més però l'edat comporta problemes de salut que han de ser tractats i supervisats.

Millorar l'eficiència és un dels objectius en l'actualitat. La tecnologia és una de les eines que ens estan ajudant a complir aquest objectiu. Ja s'està provant l'ús de consoles de videojocs per a la rehabilitació d'algunes dolències físiques o mentals, gràcies a una càmera que és capaç de reconèixer els teus gestos o fer més interactiva la teràpia del pacient.

L'objectiu d'aquest treball és aprofitar la tecnologia per a millorar l'eficiència del sistema sanitari. Per tant es vol crear una aplicació d'escriptori interactiva que motivi als pacients a fer la seva rehabilitació, els resultats d'aquests exercicis seran publicat al mur d'una xarxa social personalitzada. D'aquesta manera no caldrà fer desplaçaments fins a l'hospital, i el metge o familiar podrà fer un seguiment del pacient de manera remota.

Aquesta aplicació serà capaç de reconèixer gestos i persones, permet distingir una persona d'una altra. Ajudarà a millorar la qualitat de vida del pacient, ja que podrà fer la rehabilitació en la seva pròpia casa, i es podrà saber si ha obert el seu calaix de les medicines per exemple.

Com s'ha dit es poden reconèixer gestos, saber si s'ha obert un calaix, si una persona s'ha caigut o si ha agafat un objecte, etc. Per tant l'aplicació serviria per monitoritzar per exemple a un familiar, de manera que un altre familiar el podria vigilar.

El treball s'organitza de la següent manera. En els primers capítols s'exposaran els conceptes i un petit estat de l'art sobre el que es basa el treball, quines eines disponibles hi ha actualment en el context dels videojocs i exemples de projectes que fan servir aquestes eines. Seguidament s'exposaran els objectius inicials. A continuació s'explicarà l'arquitectura de manera general i els mòduls que la formen. En la implementació s'explica en detall cadascun d'aquests blocs, primer un resum de les llibreries utilitzades i després el desenvolupament de l'aplicació. Per acabar s'expliquen les conclusions d'aquest treball.

CAPÍTOL 1. CONCEPTES I OBJECTIUS

En aquest capítol es descriurà l'estat de l'art de les eines disponibles en el context dels videojocs que poden potenciar la rehabilitació. A continuació es farà un repàs dels dispositius que s'utilitzen actualment per a la rehabilitació. Per últim es detallaran els possibles casos d'ús que es podrien fer amb el Kinect com fer a passos un cafè, definir jocs, utilitzar-lo per a rehabilitació i fins i tot fer una monitorització dels moviments d'un usuari.

1.1 Conceptes i estat de l'art

Com ja s'ha comentat el que s'està posant de moda és fer ús de la tecnologia per a poder millorar l'eficiència i l'abast de la sanitat pública. És per això que cada cop més s'estan creant nous projectes pensats en la salut.

Un dels conceptes més interessants és: "Telerehabilitació a ritme de *bit*". Un dels exemples seria col·locar-se en el centre d'una catifa i tenir al davant una pantalla amb un trencaclosques que s'ha de completar amb els moviments dels peus. Un altre exemple, aquest cop per exercitar les extremitats superiors, seria aixecar els braços a diferents alçades fins a assolir la posició desitjada i mantenir-se uns segons immòbil per deixar la fitxa al lloc que toca. (Fig. 1.1).

Fig. 1.1 Exercicis rehabilitació amb Kinect

No es tracta de jocs d'entreteniment, sino d'exercicis que tenen com a objectiu la rehabilitació, però en els que el component lúdic motiva de tal manera al pacient que oblida el rigor de les sessions terapèutiques i inconscientment està fent accions que estimulen les seves capacitats motrius i intel·lectuals.

És un nou concepte de teràpia que tothom escolliria. Els ambients virtuals, la interactivitat o la recompensa són grans al·licients.

Play for Health (P4H), és una estratègia de telerehabilitació basada ens els denominats *serious game*, videojocs ideats de forma que a més

d'entreteniment compleixin altres funcions, com en aquest cas la rehabilitació física i l'entrenament cognitiu.

Àrea de Salud de la Fundació iBit en col·laboració amb IBSalut i Servei de Rehabilitació del Hospital Son Llàtzer [1] ha desenvolupat una plataforma que permet seleccionar un conjunt adient de jocs i modes d'interacció. En el domicili el pacient rep la sessió programada del dia, i comença a jugar seguint un ordre seqüencial. Mentrestant el programa va registrant les estadístiques dels videojocs i, un cop finalitzada la sessió les envia al terapeuta per a que aquest pugui fer el seguiment i, si fos necessari, reajustar el pla, augmentant la dificultat, per exemple.

El terapeuta controla el pacient des de casa. En el registre de dades veuen el temps, la velocitat, els erros i el tipus d'errors que ha fet el pacient, així com una gràfica sobre la seva evolució.

En aquesta plataforma es poden combinar dos elements: els videojocs i el mètode d'interacció. El primer és el repte, el que enganxa al pacient i l'anima a seguir, i el segon és la manera en la que es relaciona. El mètode d'interacció pot anar des d'una *webcam* amb guants, passant per una pista de ball, la Wiibalance, Wiimote o el Kinect. Depenent de si el que es vol treballar és l'equilibri, la coordinació, el control postural, la mobilitat fina o la de membres inferiors o superiors.

El projecte va començar a desenvolupar-se a mitjans de 2009, i s'està aplicant des de fa sis mesos al Hospital Son Llàtzer. A l'actualitat hi ha uns 40 pacients registrats al programa.

L'avantatja d'aquest projecte és la flexibilitat, que permet que, a banda de la salut, en el futur es pugui utilitzar en altres àmbits com el dels serveis socials, l'educació o l'esport. Els recursos bàsics ja estan desenvolupats a la plataforma, només s'hauria d'ampliar la funcionalitat.

Entre les diferents mètodes d'interacció hem escollit el Kinect, el qual es comentarà en el següent apartat, per a realitzar una proba de concepte.

1.2 Dispositiu Kinect

Aquest projecte pretén utilitzar la tecnologia interactiva per a realitzar jocs o exercicis que estimulin les capacitats físiques o mentals d'una persona. Per a fer aquesta teràpia es pot utilitzar el Kinect (Fig. 1.2) gràcies a la capacitat que té per interactuar amb una persona.

KINECT
for XBOX 360.

Fig. 1.2 Dispositiu Kinect

El dispositiu compta amb una càmera *RGB*, un sensor de profunditat, un micròfon i un processador personalitzat que executa un *software* patentat. El *software* proporciona captura de moviment de tot el cos en 3D, reconeixement facial i capacitats de reconeixement de veu. Més endavant s'explicarà amb més detall aquesta tecnologia.

El que diferencia el Kinect d'altres dispositius és que no es necessita l'ús d'un comandament per jugar, la càmera reconeix el cos a la perfecció i és capaç d'independitzar les extremitats i captar el seu moviment de forma independent. Tindre un control físic el fa més interactiu i crea una experiència amb més realisme.

Captar el moviment en 3D també és una característica que no tenen els altres dispositius, ja que pot arribar a captar la profunditat, un canvi substancial en la interactivitat del joc.

1.3 Altres plataformes

Hi ha altres elements que podrien fer accions semblants a la del Kinect. Totes elles però utilitzen un comandament per a interactuar amb el "joc" i no capten en 3D.

Tenim, per exemple, el Wii Balance Board (Fig. 1.3).

Fig. 1.3 Wii Balance Board

Aquesta taula és capaç de calcular la pressió exercida sobre ella. Conté 4 sensors per a poder calcular de manera precisa la pressió. Els desenvolupadors d'aquest dispositiu van crear aquesta taula pensant en fer exercici de manera divertida. També controla el pes, i fa estadístiques.

Un altre exemple seria el Wiimote (Fig. 1.4).

Fig. 1.4 Wiimote

Les característiques més destacables d'aquest dispositiu són la capacitat de detecció de moviment en l'espai i l'habilitat d'apuntar cap a objectes en la pantalla.

Per acabar, un altre dispositiu interessant és el PlayStation Move (Fig. 1.5)

Fig. 1.5 PlayStation Move

El comandament és semblant al Wiimote (sensors de moviment...) però també utilitza la càmera PlayStation Eye, que s'encarrega de detectar la posició del comandament a distància principal a partir de la grandària de la llum que detecta la càmera.

Un altre dispositiu és el Wii U [2]que sortirà a mitjans del 2012 (Fig. 1.6).

Fig. 1.6 Wii U

El comandament inclou una pantalla tàctil en la que es pot jugar també. A més té un sistema de realitat augmentada que permet una interacció més directa amb la pantalla tàctil i la de la televisió. La Wii U inclou giroscopi i una càmera frontal, es poden fer videoconferències.

1.4 Objectius del treball

L'objectiu d'aquest projecte és el d'oferir serveis que ajudin a completar la rehabilitació mitjançant exercicis interactius i que l'usuari des de casa els pugui realitzar, activitats com aixecar un braç fins a una certa alçada. Un altre objectiu és poder controlar a un familiar teu de manera remota, activitats quotidianes com fer un cafè correctament, o comprovar si s'ha obert el calaix

de les medicines, o saber si una persona ha caigut entren dins del marc d'aquesta monitorització.

Per aconseguir això hem hagut de proposar-nos els següents objectius:

- Desenvolupar una aplicació d'escriptori que interactuï amb l'usuari, per exemple visualitzar els moviments que aquest usuari realitza.
- Reconèixer moviments gràcies al Kinect, com caure al terra, agafar un objecte, obrir un calaix, etc..
- Informar al pacient si està realitzant l'exercici de manera correcta.
- Oferir una interfície web per a poder consultar els resultats.
- Planificar i organitzar tasques, com per exemple realitzar activitats domèstiques: fer un cafè, etc.
- Crear un entorn virtual per a que l'usuari realitzi els exercicis.
- Donar varies opcions de visualització remota dels resultats de l'exercici d'un usuari.
- Crear l'aplicació de manera distribuïda, en diferents blocs mitjançant APIs externes, etc.

1.5 Casos d'ús

A continuació s'explicaran els diferents casos en el que aquest projecte pot ser utilitzat.

1.5.1. Cafè 3D

Aquest exercici està pensat per a persones que tenen problemes de memòria, com els que tenen alzheimer.

L'usuari "Benet" té un dispositiu que detecta si has agafat un objecte determinat a casa seva. Té una aplicació que li permet simular la preparació d'un cafè. Benet veurà a la pantalla de la seva televisió un seguit d'objectes en 3D ("pots"), com poden ser: la cafetera, el grans de cafè, l'aigua…

Acte seguit en Benet haurà d'agafar els pots en ordre per a fer un cafè (Fig. 1.7). En cas de que l'usuari s'equivoqui de pot, és a dir que no agafa la llet de la nevera en primer lloc o el sucre en segon lloc, etc., sonarà una alarma a la televisió i a més farem aparèixer un personatge dient-li que ha escollit la opció incorrecte. Si Benet s'equivoca masses cops seguits el personatge li donarà pistes.

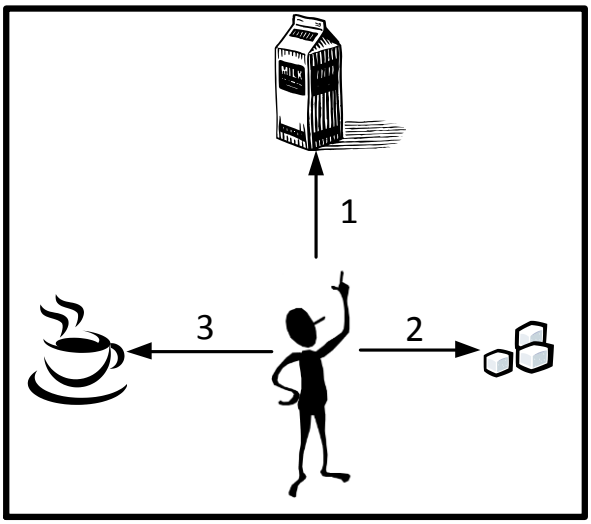


Fig. 1.7 Cafè 3D

1.5.2. Joc

En un escenari 3D definit hi ha un personatge que l'usuari Carles controlarà amb els seus moviments mitjançant el Kinect. En un ordinador a part (a quilòmetres de distància per exemple), es tindrà el mateix escenari i l'usuari Joan controlarà un altre personatge i visualitzarà l'avatar del Carles. Per tant es necessitarà transmetre la informació per un protocol de missatgeria.

Tenir un escenari virtual és més atractiu que un de real, a més permet crear objectes que pots no tenir en la vida real (dóna més llibertat).

Tal i com es veu a la Fig. 1.8 un exemple de joc seria el típic "Segueix-me", en el que en Carles fa uns moviments i en Joan que està a quilòmetres de distància ha de fer els mateixos moviments.

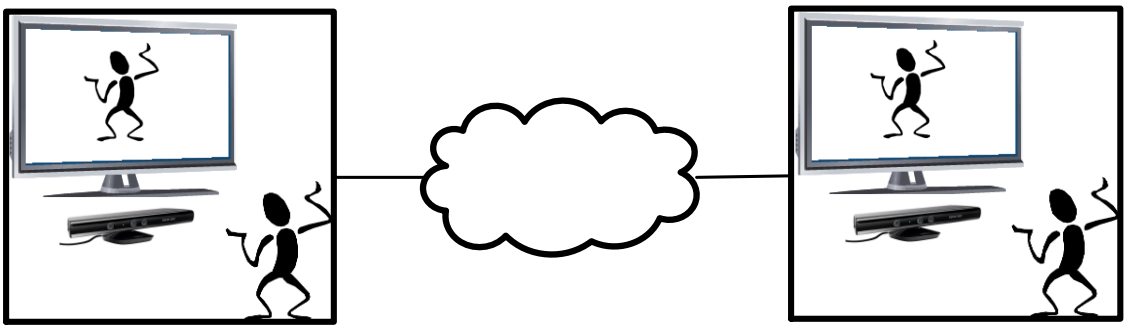


Fig. 1.8 Joc 3D

1.5.3. Telerehabilitació

En un escenari virtual on un avatar t'indica quin moviment s'ha de realitzar (fins a quin angle has d'apujar el braç per exemple) tu el segueixes, gràcies a la tecnologia del Kinect. D'aquesta manera pots fer la rehabilitació des de casa teva. Un altre cas possible seria que l'usuari tingues que ajupir-se i aixecar-se N cops com a complement per a una rehabilitació. Fins que aquest fet no es produís els N cops l'aplicació no donaria l'exercici com a correcte.

Si fas l'exercici de manera correcta o amb algun error, el resultat es podria publicar al mur d'Aaaida (xarxa social personalitzada) de manera que un familiar teu pugui fer un seguiment de la teva recuperació.

El familiar també podria fer un seguiment de si fa l'exercici de manera correcta si col·loquem una *webcam* estàndard que vegi l'habitació on el familiar realitza l'exercici de rehabilitació.

Una tercera opció és fer una gravació de l'exercici "virtual" realitzat per l'usuari.

1.5.4. Monitorització

Un altre cas seria tenir un Kinect a l'habitació de l'avi Benet. El Kinect adapta el seu món virtual al món real (l'habitació del Benet). L'usuari Benet fa la seva vida normal i corrent. Com la seva neboda Carme està preocupada per si en Benet cau o s'ha pres les medicines, s'ha col·locat un monitor (tal i com es veu a la figura Fig. 1.9 seria la part de la dreta), a la casa de la Carme, que visualitza el contingut del Kinect (l'habitació del seu avi), aquest contingut s'envia per mitjà d'un protocol de missatgeria a l'ordinador remot de la Carme.

Com el Kinect sap reconèixer gestos com obrir un calaix, i pot reconèixer si un usuari ha caigut, la Carme ja pot estar tranquil·la. A més de tenir la pantalla també es publiquen missatges al mur de la xarxa social on estan el Benet i la Carme. I la seva neboda pot consultar el mur del seu avi ja sigui a través de telèfon mòbil o consultant la pàgina web a través d'un ordinador.

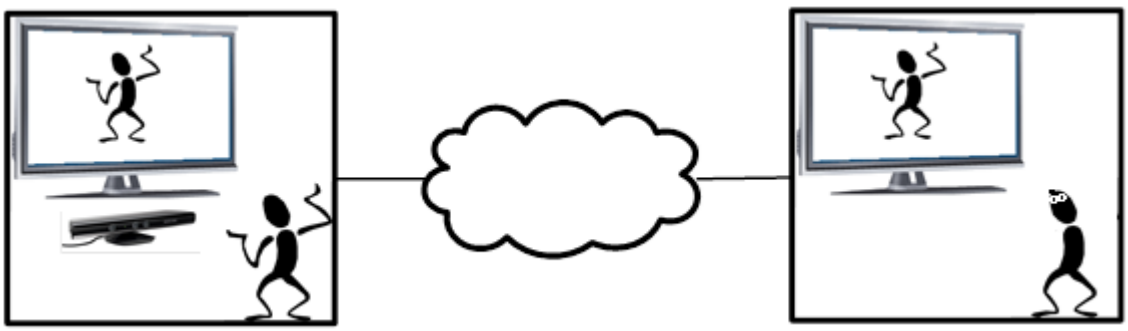


Fig. 1.9 Monitorització

CAPÍTOL 2. ARQUITECTURA

En el següent apartat s'explicarà a grans trets quines són les diferents seccions en la que es divideix el projecte.

2.1 Arquitectura general

El sistema consta de tres mòduls (veure Fig. 2.1), dos principals i un d'opcional. El primer mòdul és l'encarregat de l'adquisició, el tractament i la presentació de les dades. El segon mòdul conté tota la part de la xarxa social Aaaida i el tercer mòdul conté les eines necessàries per a fer una visualització remota tant dels moviments de l'usuari com de la xarxa social.

2.1.1. Mòdul 1: Adquisició, tractament i presentació de dades

Aquest mòdul conté tres components, el Kinect, un PC i una pantalla, aquest últim pot ser opcional com s'explicarà més endavant. El Kinect recollirà la informació referent a l'escena i als moviments de l'usuari i les enviarà al PC que té connectat en un flux de dades. Aquest PC processarà les dades rebudes del Kinect i les tractarà segons les especificacions de l'usuari. Per exemple, quan l'usuari faci un determinat gest, el PC generarà una alerta. Per últim la pantalla mostrarà els moviments de l'usuari mitjançant un avatar dibuixat en 3D que reproduirà els seus mateixos moviments. Com hem dit, aquest element pot ser opcional, ja que si no volem visualitzar els nostres moviments no ens fa falta. Per altra banda, la pantalla no necessàriament ha d'estar a la mateixa sala que el PC i el Kinect, sinó que pot estar a quilòmetres de distància i rebre les dades per Internet. Aquest cas ofereix a una altra persona la possibilitat de monitoritzar els moviments de l'usuari situat a la sala amb el Kinect.

2.1.2. Mòdul 2: Xarxa social Aaaida

El segon mòdul és la xarxa social Aaaida. Quan es produeix un determinat esdeveniment, un gest o un moviment, el PC del mòdul 1 envia una petició a aquesta xarxa social i es fa una publicació al mur de l'usuari amb informació sobre l'esdeveniment produït.

2.1.3. Mòdul 3: Monitorització

Opcionalment podria haver-hi un tercer mòdul, i els seus components podrien estar situats a una altra sala a quilòmetres de distància de la sala on hi és el Kinect. Bàsicament aquest mòdul ens serviria per, tal i com hem comentat abans, oferir a una altra persona la possibilitat de realitzar la monitorització de l'usuari del mòdul 1. Mitjançant una pantalla i rebent les dades que el PC del mòdul 1 li envia, podríem veure els moviments de l'usuari. També podria,

mitjançant un altre PC o un mòbil, consultar el mur de l'usuari i fer un seguiment de les publicacions que es fan.

Fig. 2.1 Arquitectura general

CAPÍTOL 3. IMPLEMENTACIÓ

En aquest capítol s'explicarà en profunditat els dos mòduls principals que hi ha al nostre sistema i l'entorn de desenvolupament que s'ha utilitzat.

A la primer part es descriuran les diferents tecnologies i llibreries utilitzades a cada mòdul. A continuació s'exposarà la part de desenvolupament, que està composada per la càrrega de l'escena, la sincronització del personatge, l'estructura de dades interna, la detecció de moviments, el mur d'Aaaida, i el diagrama funcional.

3.1 Implementació

A continuació es veuen a la Fig. 3.1 aquests dos mòduls. El mòdul 1 està format per dos elements: el Kinect i el PC. El mòdul 2 està format pel servei web d'Aaaida.

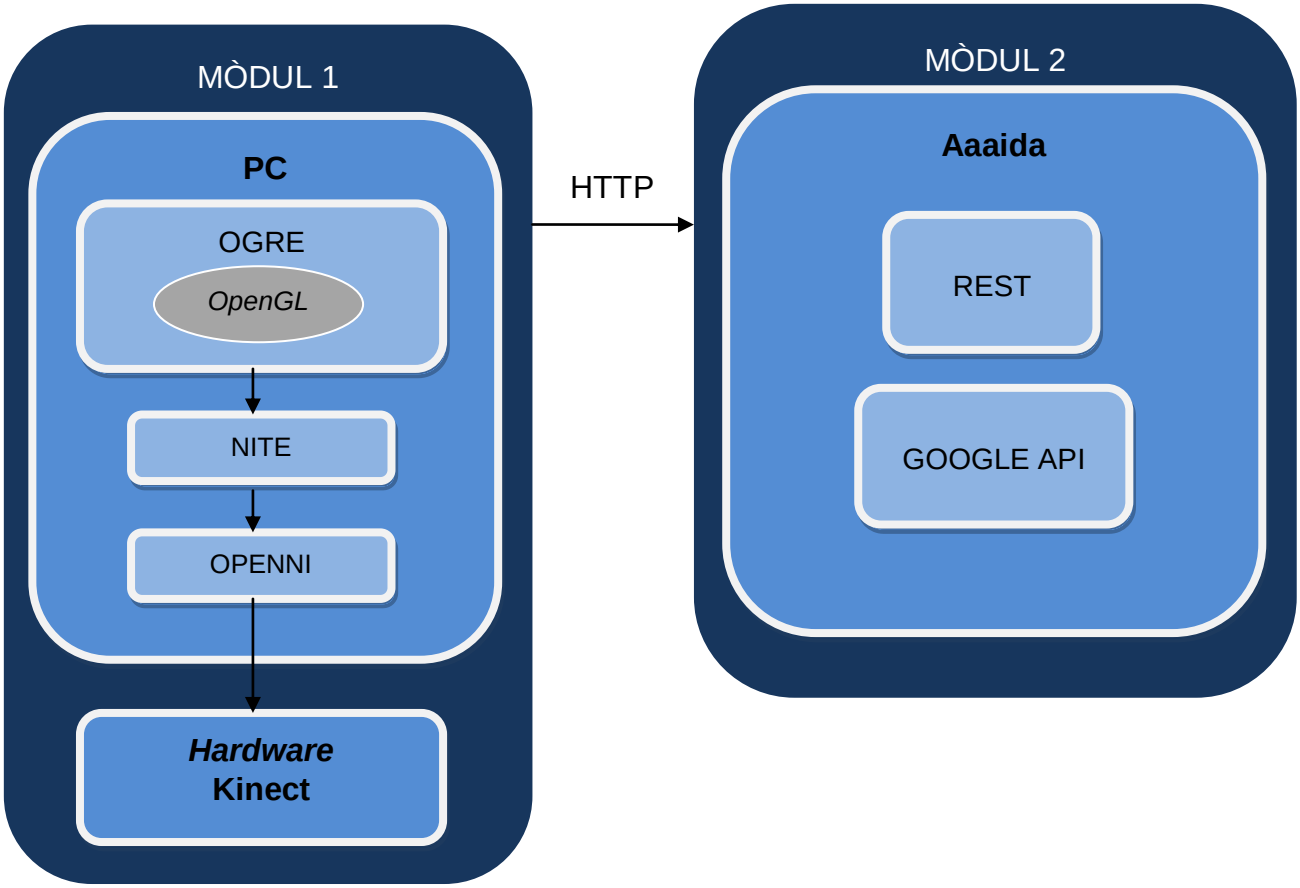


Fig. 3.1 Implementació modular

3.1.1. Mòdul 1: Adquisició, tractament i presentació de dades

Com hem dit aquest mòdul està format per dos elements, el Kinect i l'ordinador. L'ordinador a més està format per tres components: Ogre, NITE i OpenNI.

3.1.1.1. Kinect

El Kinect [3] és un perifèric creat per l'empresa Microsoft que es pot connectar tant a la Xbox 360 com a l'ordinador. Està format per: una càmera, un micròfon i un motor.

La càmera permet generar una imatge tridimensional del que té al davant. A més pot reconèixer parts del cos humà. Per a fer-ho utilitza una llum infraroja. Quan els rajos infrarojos impacten a un objecte sòlid, aquests reboten i gràcies al temps que triguen els rajos a tornar al dispositiu podem saber la distància a la qual es troba l'objecte (Fig. 3.2). No solament es pot saber la distància, sinó que es pot fer una reconstrucció de l'objecte. Gràcies a l'ús de senyals infraroges té una major precisió en la detecció de profunditat i no depèn tant de la llum ambiental.

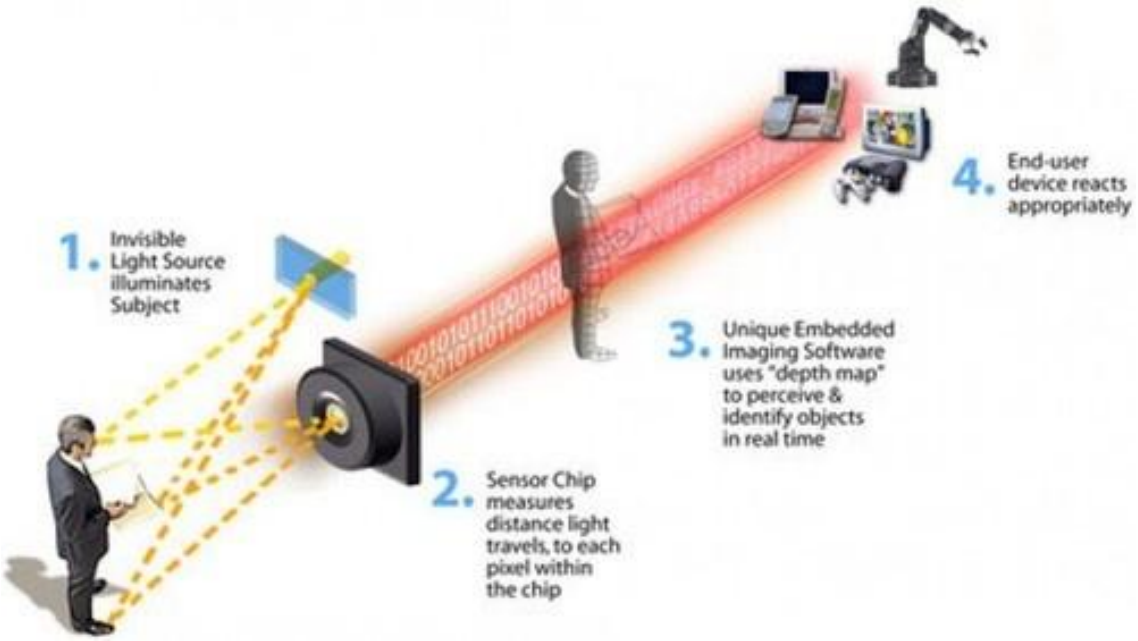


Fig. 3.2 Funcionament del detector volumètric

Els components que formen el Kinect són senzills. Compta amb una parell de sensors de profunditat lleugerament separats. La clau de tenir dos sensors, és proporcionar a Kinect una visió 3D del seu entorn. Tal i com les persones tenen dos ulls que els permeten distingir les distàncies i les profunditats dels objectes que ens envolten. Entre aquests dos sensors es troba una càmera "comú" RGB.

El que verdaderament és valuós del Kinect sens dubte no és el hardware, sino la forma en que el hardware treballa. Ja que Kinect es tracta de jugar sense

controls, és molt important poder seguir el moviment de la persona que es té davant del televisor. Per això Kinect utilitza els seus sensors de profunditat, projecta cents de punts infrarojos per tota la habitació. Si gravem amb una càmera infraroja una habitació on s'estigui utilitzant el Kinect veurem una cosa semblant a aquesta imatge(Fig. 3.3). Com hem dit abans els sensors detecten la profunditat d'un objecte mesurant el temps que tarda en rebotar la "llum" infraroja.

Fig. 3.3 Captura llum infraroja

Essencialment tots els píxels que Kinect rep com a soroll Infraroig són convertits en una escala de colors. Els cossos depenent de la distància es capturen com a vermells, verds, blaus fins a arribar a tons grisos, que representen a objectes molt llunyans (Fig. 3.4) . Kinect fa aquesta captura a 30 quadres per segon, és a dir que repeteix el procés 30 cops per segon.

Fig. 3.4 Captura amb detecció de cossos segons la distància

El *hardware* també conté micròfons que es troben als laterals del dispositiu per evitar confondre el soroll provinent d'altres fonts (com el so del mateix televisor) amb la veu del jugador (Kinect accepta instruccions per veu). Un *software* anomenat "*Beam Forming*" treballa en conjunt amb la càmera per a endevinar la teva posició i crear una esfera de so. La posició en la que han d'estar els micròfons és la raó per la qual el Kinect és tan llarg.

Kinect també disposa d'un motor situat a la base d'aquest. El motor és capaç de moure la unitat a dalt i a baix fins a 30 graus. El dispositiu conté aquest sistema d'inclinació que s'ajusta d'acord a la distància que està el jugador del televisor. D'aquesta manera s'estalvia calibrar el dispositiu cada cop que es canvia d'habitació, mous mobles o t'allunyes més del normal. El motor del Kinect és pràcticament silenciós, de manera que no interfereix amb els micròfons.

A diferència de Wiimote [4] i Playstation Move [5], Kinect és un dispositiu que captura a l'usuari sencer i al seu entorn, però principalment que més d'un usuari pot utilitzar a la vegada.

El dispositiu està dissenyat per a que puguin interactuar fins a sis persones diferents, capturant i processant la representació individual i auditiva de manera individualitzada.

Ja que el Kinect reconeix la posició dels objectes en l'habitació, necessita saber què són, és a dir necessita distingir les persones del demés objectes. Per a això, Kinect compta amb un *software* integrat amb una base de dades de cents de cossos de persones en diferents posicions i angles.

Un cop el Kinect identifica qui o quins són els jugadors, forma amb ells "esquelets" digitals per simplificar la imatge i poder entendre els moviments del jugador. Un *software* intern distingeix entre l'esquelet digital de cada jugador, fins i tot quan els jugadors intercanvien posicions.

El *software* del Kinect a més de distingir la teva veu d'entre altres sons, també ho fa de les veus d'altres persones. De manera que quan se li dóna una instrucció a Kinect, sap qui li ha ordenat.

Kinect compta també amb un model acústic d'acord a cada país. D'altre manera seria molt difícil adaptar el reconeixement de veu entre els diferents "accents" del món. Fins i tot reconeix entre diferents accents dins d'un mateix país i idioma.

3.1.1.2. PC

Tal com s'ha vist a l'apartat d'arquitectura, necessitem un PC que rebi el flux de dades generat pel Kinect i el processi. Aquest component és el més important ja que és on s'analitzaran les dades de l'escena i es faran les accions pertinents. A grans trets, hi ha tres components de *software*. El primer serà OpenNI, el segon NITE i el tercer Ogre.

3.1.1.2.1. OPENNI

OpenNI és un *framework* multi-plataforma que defineix unes API de codi obert desenvolupades en el llenguatge C++ i que es poden utilitzar per desenvolupar aplicacions que utilitzen Interacció Natural.

El terme Interacció Natural s'utilitza quan la interacció entre un dispositiu i la persona es basa en els sentits humans. És a dir, en aquest cas la pròpia persona seria el perifèric, substituint als ratolins o als teclats. A l'actualitat ja existeixen diversos exemples de la utilització d'Interacció Natural com és el reconeixement de veu, on els dispositius reben instruccions a través de la veu de la persona.

Per portar a terme aquestes operacions es poden distingir dos tipus d'elements. Per una banda hi ha els sensors de visió i d'àudio, que són els dispositius que "veuen" i "escolten" les figures y el seu voltant. Per altra banda, els components de *middleware* de visió i de percepció d'àudio. Aquests últims són els components de *software* que analitzen i comprenen les dades d'àudio i visuals que es recullen de l'escena mitjançant els sensors anteriors.

Així, l'objectiu principal d'OpenNI es formar un conjunt *d'APIs* estàndard que permeti la comunicació entre els dos mòduls anteriors. Un dels avantatges més importants d'aquestes API és que trenca la dependència entre el sensor i el *middleware*. Això vol dir que les aplicacions desenvolupades amb OpenNI, poden ser exportades i funcionar en altres mòduls de *middleware* diferents. També permet desenvolupar components de *middleware* per processar les dades amb independència del hardware que les hagi produït sempre i quan sigui compatible amb OpenNI.

L'API d'OpenNI permet als desenvolupadors d'aplicacions d'interacció natural realitzar un seguiment de les escenes de la vida real (3D) mitjançant la utilització de dades que es calculen a partir de la entrada d'un sensor (per exemple, la representació d'un cos complet, o la posició de la mà).

Les capacitats més importants que dóna són: reconeixement de veu, gesticulació de mans i seguiment del moviment del cos. Dintre d'aquestes destaquen la capacitat *skeleton* (fa un seguiment de la posició de les articulacions de l'esquelet i el seu moviment).

Al següent gràfic (Fig. 3.5) es pot veure el concepte d'OpenNI descompost en capes.

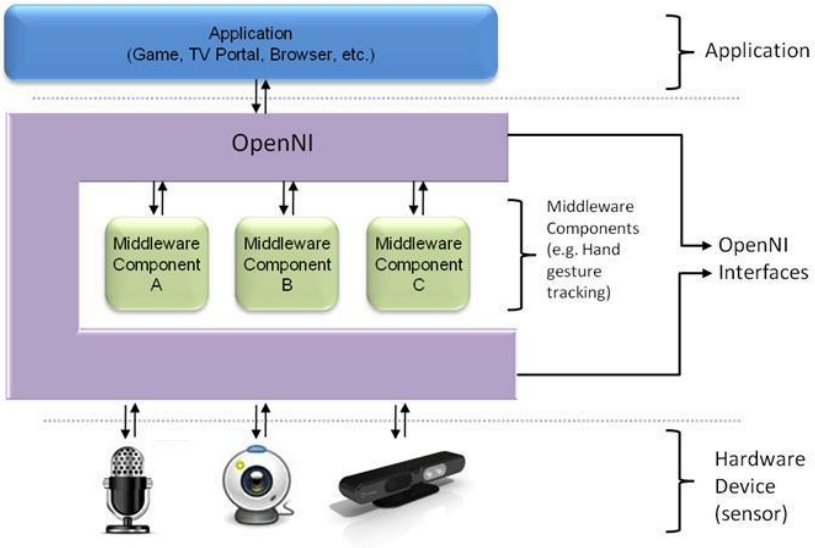


Fig. 3.5 Capes del sistema

Part superior: representa el *software* que implementa les aplicacions d'interacció natural. En aquest projecte serien les llibreries Ogre.

Part intermitja: representa OpenNI, que proporciona les interfícies de comunicació entre els sensors i els components *middleware*, i que analitza les dades del sensor. En aquest projecte el component *middleware* que interpreta les dades recollides pel sensor és un mòdul anomenat NITE.

Part inferior: es compon dels dispositius *hardware* que capturen les dades d'àudio i visuals de l'escena. En el nostre cas seria el Kinect.

3.1.1.2.2. NITE

Com s'ha comentat abans, es necessita un element que comprengui les dades recollides pel sensor. Aquest element serà el mòdul NITE, desenvolupat en el llenguatge C++ i serà qui analitzarà les dades del sensor mitjançant OpenNI com a interfície amb el propi sensor.

El mòdul NITE és una solució que permet a un dispositiu percebre el món en tres dimensions i traduir-les en una imatge sincronitzada amb profunditat, de la mateixa manera que fem els humans. És a dir, és l'element que comprèn la interacció de l'usuari dins de l'entorn.

El *middleware* NITE assegura un processament eficient de les dades visuals, la qual cosa el fa ideal per a plataformes amb baixa capacitat de processament i memòria. Conté la infraestructura algorítmica per a la identificació d'un usuari, detecció de característiques i reconeixement de gestos, així com el marc de control que gestiona l'etiquetatge d'usuaris en l'escena i l'adquisició i alliberament de control entre els usuaris. Això ens permet que, mentre un usuari està interaccionant amb el sistema, els gestos d'un altra persona dins de

l'escena serien ignorats, permetent a l'usuari una interacció sense interrupcions.

NITE processa els gestos de l'usuari, i permet a l'usuari realitzar una operació o una altra en funció del gest de l'usuari com pot ser dibuixar un cercle amb la mà o emular un *click* amb la mà, movent-la endavant i tornant a la seva posició.

Una altra característica important de NITE és el seu enfocament per als denominats "Games for All". Aquests jocs es caracteritzen per ser actius, físics i socials. Un dels exemples més clars d'aquest tipus de jocs són la majoria dels jocs de la Wii on es juga amb més persones al menjador. En aquest sentit, NITE proporciona un conjunt de controls de joc que permeten el desenvolupament de jocs de cos complet sobre plataformes senzilles. Això permet que el desenvolupador es centri més en aspectes com la creativitat gràfica i l'experiència de l'usuari, sense haver de patir pel desenvolupament més dur del software de processament d'imatge i de profunditat.

També suporta múltiples usuaris, i està dissenyat per una àmplia gamma de jocs, com boxa, carreres, plataforma, etc. De moment el màxim que pot suportar és 2 usuaris a la vegada, ja sigui en mode competitiu o col·laborador. El *framework* de NITE també proporciona eines per personalitzar el control sobre determinats gestos i definir nous gestos i funcionalitats.

3.1.1.2.3. OGRE

Per últim tenim el component que recull les dades generades per NITE i les utilitza per presentació. Aquest element s'utilitzarà per mostrar a l'usuari per la pantalla un món virtual en 3D, que contindrà objectes dibuixats anteriorment per l'usuari i inserits a l'escena. També contindrà un avatar en 3D dibuixat per l'usuari, i que reproduirà els mateixos moviments que faci l'usuari.

Ogre 3D [6] és un motor de renderitzat 3D orientat a escenes, escrit en el llenguatge de programació C++. Les seves llibreries eviten la dificultat de la utilització de capes inferiors de llibreries gràfiques com OpenGL i Direct3D. En el nostre cas, utilitzarà les llibreries d'OpenGL per mostrar el món virtual per pantalla.

OpenGL és una especificació estàndard que defineix una *API* multillenguatge i multiplataforma per escriure aplicacions que produeixin gràfics 2D i 3D. La interfície consisteix funcions que es poden utilitzar per dibuixar escenes tridimensionals complexes a partir de primitives geomètriques simples, tals com a punts, línies i triangles. El problema que té és que és una *API* basada en procediments de molt baix nivell i requereix un coneixement molt ampli. És per això que Ogre facilita aquesta tasca i ho fa de manera transparent a l'usuari.

Les aplicacions que utilitzen Ogre són portables entre plataformes i sistemes. Amb aquesta aplicació es pot modelar l'escena com es vulgui situant objectes en qualsevol posició i després poder visualitzar l'escena per pantalla. El món virtual que l'usuari veu per pantalla estarà dibuixat amb el programa Blender

3D. Amb aquest programa es pot fer modelatge, creació i animació de gràfics tridimensionals, inclús afegir textures per a que sembli fet de fusta o d'algun altre material. Per a que Ogre pugui mostrar els objectes modelats amb Blender prèviament haurem d'exportar l'objecte amb un *script*. *L'script* el que farà serà bolcar l'objecte en un fitxer mesh. Un objecte Mesh representa a un model discret, un conjunt de geometria que normalment és petit comparat amb l'escala del món. Els objectes Mesh representen objectes mòbils. L'únic que s'haurà de fer serà introduir el nom d'aquest fitxer a la configuració d'Ogre i ja es podrà afegir el nou objecte.

3.2 Mòdul 2: Xarxa social Aaaida

En el treball es fa ús d'un servei Web extern que aporta la funcionalitat de poder introduir i obtenir informació des de Google Health.

El servei Web s'encarrega d'intercanviar dades entre diferents aplicacions, independentment del llenguatge de programació en las que hagin estat construïdes o sobre quines plataformes es trobin. Aquest apartat forma part de la integració amb el projecte Aaaida, per això fa ús del seu servei Web (veure Fig. 3.6).

Com ja s'ha dit abans en el treball, per cada esdeveniment que succeeixi en l'aplicació es publicarà en el portal web Aaaida un avís. Per a publicar els resultats es fa servir REST. A continuació s'explicarà amb detall Aaaida i REST.

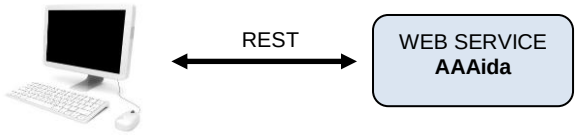


Fig. 3.6 Web Service

3.2.1. Aaaida

Aaaida [7] és una xarxa social personal, particular, fins i tot íntima, on se sap l'estat de les persones o coses que realment preocupen: l'estat dels vincles. Així doncs, Aaaida permet crear vincles, conèixer el seu estat mitjançant mesures, i compartir-los amb altres membres amb els que coincideixen amb aquesta preocupació. Per exemple una persona podria compartir amb les seves germanes l'estat de salut dels seus pares; amb la seva dona, la febre i la medicació que li ha donat al seu fill petit aquesta setmana; amb els seus amics, els quilòmetres que han recorregut aquest dissabte; fins i tot amb el metge, la seva tensió arterial, que no la porta molt controlada.

És important remarcar que les dades d'Aaaida són privades, propis de cada usuari, i el seu ús, finalitat, i fins i tot veracitat, depenen única i exclusivament de cada usuari.

Aaaida té en la seva versió web (Fig. 3.7) la forma inicial, natural, d'introduir dades de les mesures i interactuar. Tanmateix, es pot utilitzar també qualsevol mitjà que permeti fer arribar la informació a la plataforma, com per exemple SMS, aplicacions mòbils, i, fins i tot sensors.

La implementació actual està en versió alfa tancada i s'està recopilant opinions i impressions del concepte per anar-lo millorant de cara a la versió beta que apareixerà pròximament.

En el nostre cas es publica al mur un missatge quan es provoquen esdeveniments. Un exemple seria quan l'avi Benet obre el calaix de les medicines.

Fig. 3.7 Portal Web Aaaida

3.2.2. REST

REST o *Representational State Transfer* no és exactament un protocol, sino una arquitectura que pot ser utilitzada per altres protocols que utilitzin un vocabulari establert, com HTTP, que fa ús de mètodes estandarditzats. Una arquitectura defineix les característiques o propietats que ha de tenir un protocol basat en ell. Un servei Web REST, és un servei Web implementat en HTTP seguint els principis REST.

Les característiques més importants d'aquesta arquitectura són:

- Els elements de dades: recursos (com dades d'objectes), identificadors de recursos (adreces d'internet, *URLs*), i representacions de recursos (com documents HTML) són accedides a través d'una interfície estàndard com HTTP.

- Components: servidors origen, *proxies*... es comuniquen transferint representacions de recursos a través de la interfície, sense operar directament amb els mateixos recursos. Això es fa generalment amb operacions ben definides, com “Get” i “Put”.
- Connectors: clients, servidors i *caches* presenten una interfície abstracta de comunicació, amagant detalls d’implementació dels mecanismes de comunicació.
- Interacció sense estats previs: totes les peticions fetes als connectors han de contenir tota la informació necessària per entendre les peticions sense dependre de peticions prèvies (pàgina web sense *cookies*).

3.2.3. Emmagatzematge de dades

Com ja s’ha dit abans en aquest treball, es fa ús del servei Web del projecte Aaaida. Actualment aquest servei emmagatzema la informació en una aplicació *Google*, *Google Health*, en aquesta es pot organitzar, monitoritzar i comprovar els progressos de la salut, tot i que l’ús d’aquesta eina recau en la facilitat d’obtenir un lloc on emmagatzemar dades de manera més o menys senzilla.

Per a poder treure o introduir informació dels usuaris, el servei Web converteix aquesta informació en un document etiquetat com XML.

Actualment s’està treballant en una nova versió d’Aaaida que utilitzarà una base de dades interna, de manera que el procés d’emmagatzemar dades es realitzarà de manera més ràpida.

3.3 Desenvolupament

En aquest apartat es detallaran les diferents funcionalitats que s’han desenvolupat per facilitar la interacció de l’usuari amb el món virtual. Aquestes funcionalitats van des de la càrrega de l’escena, passant per la sincronització del personatge, les estructures de dades i fins a la detecció de moviments. Finalment el resultat d’aquestes accions repercuteixen al mur d’Aaaida, ja que gràcies a aquestes funcionalitats es creen esdeveniments que són publicats al mur d’aquesta xarxa social. L’última part d’aquest apartat conté un diagrama de funcions de la lògica del projecte.

3.3.1. Càrrega de l’escena

La interacció entre l’usuari i el món virtual és molt important, i quantes més possibilitats d’interacció amb els objectes hi hagi més atractiu i interactiu serà per l’usuari. És per aquest motiu que s’ha desenvolupat una estructura de dades que ofereix la possibilitat i facilita l’acció d’inserir objectes 3D al món virtual. Aquests objectes, prèviament hauran de ser modelats i exportats per l’usuari amb el programa Blender. Per a que l’usuari no tingui que fer un aprenentatge en el modelatge d’objectes 3D, s’ha modelat alguns objectes i els s’han afegit a l’aplicació donant la possibilitat a l’usuari d’incloure-les a l’escena.

Aquests objectes són una caixa, un armari, un calaix, un got, una gerra, una taula i una cadira. A la Fig. 3.8 es pot veure l'escenari amb tots els objectes possibles.

Fig. 3.8 Escenari complet

Per fer possible la inserció de manera transparent a l'usuari, l'aplicació carregarà la informació de l'escenari des d'un fitxer extern anomenat "Escena.txt" (Fig. 3.9). Aquest fitxer és configurable per part de l'usuari i ens ofereix un alt grau d'escalabilitat a l'aplicació perquè es poden afegir tants objectes com es vulguin a l'escena. Per cada línia, es defineix l'objecte que es vol afegir a l'escena i una sèrie de propietats que s'explicaran en detall més endavant. La configuració d'un objecte es farà amb els paràmetres següents:

```
caixa,caixa,true,1,4,1,0,0,0,0
taula,taula,false,0,-7,4,0,0,90,0
cadira,cadira,false,0,-10,1,2,0,0,0
```

Fig. 3.9 Fitxer de càrrega "Escena.txt"

La primera de totes és el nom que li volem donar al objecte. L'aplicació no permet que dos objectes tinguin el mateix nom. Si es produeix el cas sortirà un error informant-nos de que existeixen dos objectes amb el mateix nom. La

segona propietat és el tipus, que serà un dels esmentats anteriorment (caixa, got ,taula, etc). Les següents propietats s'explicaran amb detall a l'apartat d'estructura de dades. Per fer un resum, la següent propietat, indica si l'objecte es pot agafar o no. El següent paràmetre és el valor de la finestra, que també es detallarà més endavant. El següents tres paràmetres indicaran la posició de l'objecte en l'escena sent el primer la posició en l'eix de les X, el segon en l'eix de les Y i el darrer en l'eix de les Z. Els darreres tres paràmetres serveixen per rotar l'objecte original en el eix de les X,Y i Z respectivament.

3.3.2. Sincronització personatge

Per al desenvolupament de la nostra aplicació s'ha reaprofitat un exemple proporcionat a la mateixa web d'OpenNI. L'exemple consta d'un avatar en el centre de la pantalla, modelat amb el programa Blender. Aquest avatar consta d'un esquelet dotat d'articulacions per tot el cos, fins i tot pels dits, per a que després es pugui simular el moviment de les articulacions a l'aplicació. Per a que es pugui fer la unió de l'esquelet de l'usuari real a les articulacions de l'avatar i que aquest reprodueixi els moviments és necessari la identificació prèvia de l'esquelet. Per a aquesta tasca, l'usuari s'ha de posar en una postura determinada amb els braços en horitzontal i les mans cap a dalt (Fig. 3.10). Quan l'aplicació ha acabat de calibrar l'esquelet de l'usuari és quan relaciona les articulacions de l'esquelet amb les de l'avatar. Aquesta lògica de detecció ja ve implementada a l'exemple. A més, també s'aprofitarà el model 3D del propi avatar.

Fig. 3.10 Postura calibratge

3.3.3. Estructura de dades

Per fer possible la inserció d'objectes a l'escena de manera transparent a l'usuari s'ha dotat als objectes d'una sèrie de propietats que es definiran per l'usuari segons les seves necessitats. En aquest apartat s'explicarà en detall les propietats que l'usuari ha de configurar al fitxer "Escena.txt" per inserir un objecte a l'escena.

Tots els objectes són afegits a un vector d'objectes (*hash*), el qual contindrà tots els objectes de l'escena. Aquesta estructura ofereix la possibilitat d'escalar l'aplicació perquè es poden afegir tants objectes com es vulgui. Aquest vector es recorre per fer vàries comprovacions de control com per exemple, si algun objecte està agafat, no es podrà agafar cap altre objecte fins que s'hagi deixat.

Com es veu a la Fig. 3.11 la *hashtable* conté tots els objectes de l'escena amb les propietats de cadascun. Tots els objectes estan implementats amb una sèrie de mètodes per poder llegir i escriure els valors de les propietats. Des de la lògica d'implementació es darà ús d'aquests mètodes per realitzar les diferents accions necessàries. També hi hauran una sèrie de *listeners*. Aquests *listeners* són unes funcions que estaran "escoltant" constantment les dades proporcionades per OpenNI. Aquestes funcions permetran estar constantment a l'espera d'una acció determinada definida per l'usuari. Quan aquesta acció es produeixi, es produirà un esdeveniment. Aquest esdeveniment es tractarà a la lògica d'una forma o una altra depenent del tipus que sigui i es realitzaran les accions pertinents. El resultat d'aquest esdeveniment es veurà reflexat de forma visual a la pantalla.

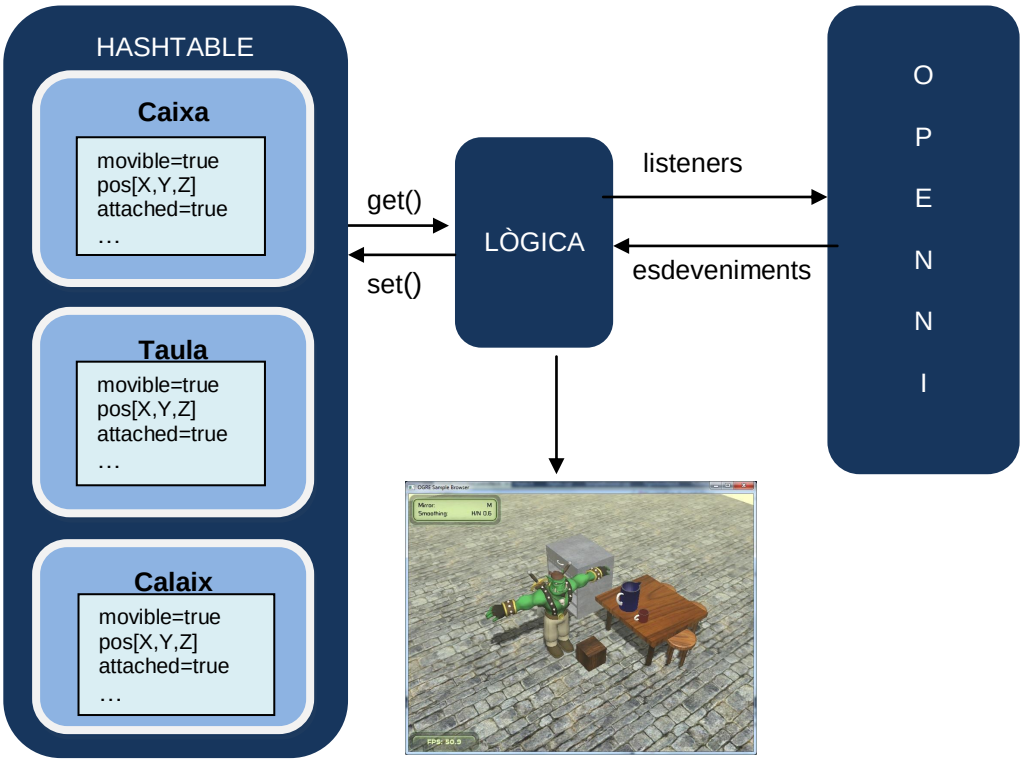


Fig. 3.11 Estructura Interna

La primera és la propietat “movible”. Aquesta propietat donarà l’oportunitat a l’usuari de decidir si l’objecte que està inserint es pot agafar o no. Si es posa aquesta opció a vertadera, l’usuari podrà agafar i deixar aquest objecte al món virtual. Si aquesta opció es posa a fals, l’objecte modelat serà inserit a l’escena però l’usuari no el podrà agafar, és a dir, només formarà part del decorat. Aquesta opció és interessant ja que per exemple, una armari no té massa sentit que es pugui agafar.

La següent propietat és “finestra”. Quan s’agafa un objecte, el que fa l’aplicació és unir el centre de massa de la mà (centre del palmell) i el centre de massa de l’objecte i així quan es mou la mà, l’objecte es mou amb ella. Aquest centre de massa es pot posar on es vulgui quan es fa el model en Blender. En el cas d’un calaix tindria sentit posar el centre de massa al tirador, així, quan la mà de l’usuari s’acostés al tirador, s’uniria el calaix amb la mà. Però a l’hora d’agafar l’objecte, és molt difícil fer coincidir les coordenades exactes X,Y i Z de la mà de l’usuari amb les coordenades del centre de massa de l’objecte. Per aquest motiu s’estableix una finestra (Fig. 3.12), que facilita agafar l’objecte. Aquesta propietat és un número enter, el qual s’utilitzarà per ampliar el rang de posicions que fan que l’usuari agafi l’objecte. Aquest número és el valor que s’amplia pel punt de centre de massa en les dues direccions per les coordenades X,Y i Z. Així, per agafar l’objecte serà necessari passar la mà per aquesta finestra al voltant del centre de massa i resultarà més fàcil agafar-lo.

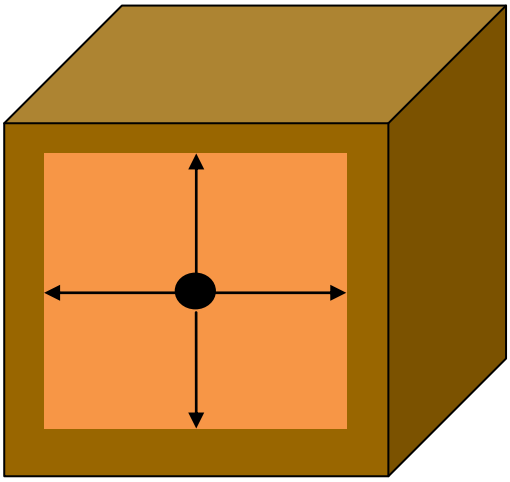


Fig. 3.12 Finestra al voltant del centre de massa

La propietat “posicionHash” indica la posició de l’objecte a l’estructura de dades interna.

La propietat “posicion” guardarà les coordenades X,Y i Z on l’usuari situï l’objecte a l’inici de l’aplicació.

La propietat “nombre” servirà per identificar l’objecte en el moment d’agafar-lo i deixar-lo.

La propietat “attached” servirà per saber en un determinat moment si l’objecte està agafat o no. En el cas de que l’usuari ja tingui un objecte agafat, no podrà agafar un altre.

Per últim, la propietat “objetivo” servirà per saber si l’usuari té la intenció d’agafar un objecte. Quan passi la mà per la finestra d’un determinat objecte, aquest es convertirà en objectiu i ens servirà per agafar l’objecte desitjat.

3.3.4. Detecció de moviments

Aquí s’explica quines accions es poden realitzar en l’aplicació. Agafar i deixar objectes, obrir i tancar el calaix de les medicines i detectar si un usuari ha caigut. Per últim també pot controlar si s’ha fet un exercici de rehabilitació. Aquestes accions són les que com s’ha dit abans provoquen esdeveniments que seran publicat al mur d’Aaaida.

3.3.4.1. Agafar i deixar objectes

A tots els objectes que s’afegeixin a l’escena i que es poden agafar, se’ls hi aplicarà les mateixes directrius. Els procediments que s’explicaran ara pel cas de la caixa, s’apliquen a tots els objectes movibles. L’única excepció és el cas del calaix, que s’explicarà més endavant.

Com ja s’ha comentat abans si l’usuari acosta la mà i el centre de massa entra en la finestra de la caixa que es pot agafar, aquesta s’enganxarà a la mà. Com podem veure a la Fig. 3.13 quan el punt vermell de la mà es trobi dintre de la finestra taronja de l’objecte, el punt negre de l’objecte s’unirà al vermell de la mà i quan l’usuari mogui la mà, l’objecte es mourà amb ella com es mostra a la Fig. 3.14.

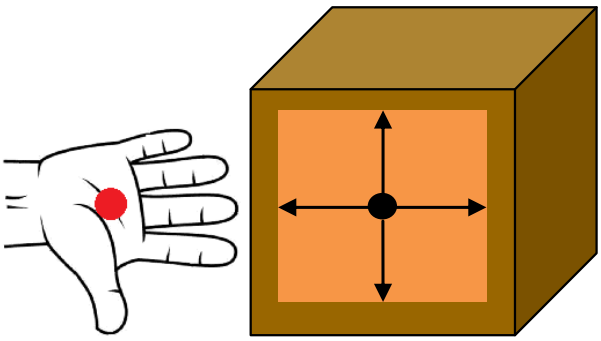


Fig. 3.13 Agafar objecte de manera teòrica

Fig. 3.14 Agafar objecte en la nostra aplicació

En el cas que es vulgui deixar la caixa, es pot fer de dos maneres diferents. La primera és deixar-la en el mateix lloc que estava. Per a això l'únic que s'haurà de fer és passar la mà pel mateix lloc on s'ha agafat i la caixa tornarà a la seva posició original. La segona de les maneres és deixar-la a qualsevol lloc del terra. L'únic s'ha de fer és acostar la mà al terra i quan s'està suficientment a prop la caixa es desprendrà de la mà i quedarà al terra amb la les mateixes coordenades X i Z de la mà, excepte la Y que serà 0 perquè quedarà al terra com es veu a la Fig. 3.15. Depenent dels objectes que s'han agafat tindrà sentit o no deixar-lo al terra, per exemple un got no tindria massa sentit deixar-lo al terra així que es deixaria al mateix lloc on s'ha agafat. Encara que si l'usuari vol deixar-lo al terra també ho podria fer.

Per a que l'usuari pugui veure quina acció ha realitzat, quan es produeixi qualsevol acció com agafar o deixar un objecte mostrarem un missatge per pantalla. El missatge es mostrarà a la part superior i s'informarà a l'usuari de l'acció realitzada. Al cap de 2 segons el missatge desapareixerà.

Fig. 3.15 Deixar objecte en la nostra aplicació

En aquest punt es va presentar un problema. Quan es deixava la caixa al terra, la mà seguia estant dins de la finestra feta per agafar l'objecte. Com que l'aplicació va actualitzant la imatge i el cos constantment, la caixa tornava a la mà un altre cop i la tornava a deixar i així successivament. Per solucionar aquest problema es va definir la condició de que quan deixàvem la caixa, s'hauria d'allunyar la mà de la mateixa finestra per tornar a agafar-la. Així quan es deixa la caixa la mà no ha sortit fora de la finestra i no s'activa l'acció d'agafar-la fins que l'usuari no allunyi la mà i torni a apropar-la en cas de que la vulgui tornar a agafar.

3.3.4.2. Obrir i tancar calaix de les medicines

Per al cas d'obrir el calaix d'un armari, hem hagut de tractar-ho com un cas especial perquè evidentment el calaix està adherit a l'armari i no t'ho pots emportar amb la mà on tu vulguis. En aquest cas, quan s'acosta la mà al tirador del calaix, es tancarà la mà per simular que s'agafa i solament es mourà en l'eix de les X (Fig. 3.16).

Fig. 3.16 Obrir calaix

La dificultat d'obrir el calaix del món virtual sense obrir-lo en el món real, és que no solament es mourà la mà en l'eix de les X per obrir el calaix sinó que inevitablement també es mourà en els eixos Y i Z. Per solucionar aquest problema es defineix una altra finestra al voltant del tirador. Mentre la mà estigui en aquesta nova finestra, es podrà obrir i tancar el calaix encara que es mogui la mà cap a dalt o cap a baix inconscientment. En el moment que es surti d'aquesta finestra, voldrà dir que la mà s'ha desviat excessivament i serà un signe de que l'usuari vol deixar l'objecte. Quan la mà es trobi fora d'aquesta finestra, es deixarà de controlar el calaix. Quan es deixi el calaix, aquest quedarà en la posició que s'hagi deixat, és a dir, si estava obert quedarà obert. En aquest cas, no es pot unir el centre de massa de la mà i el del calaix perquè en el cas que l'usuari mogués la mà cap a dalt l'objecte també ho faria i el calaix solament s'ha de moure en l'eix de les X.

Per actualitzar la posició del calaix mentre s'obra i es tanca el que es fa és mantenir sempre la distància entre la mà i el calaix mentre es té agafat, així quan la mà es desplaci dues unitats cap a l'esquerra, actualitzarem la posició del calaix i també es mourà dues posicions a l'esquerra. Aquest procés s'anirà repetint fins que el calaix s'hagi obert del tot o s'hagi tancat del tot. Per exemple, si ja s'ha obert el calaix del tot i l'usuari segueix estirant, la posició del calaix no s'actualitzarà i hi haurà un moment en que la mà surti fora de la finestra i l'objecte deixi d'estar agafat.

3.3.4.3. Usuari ha caigut

És important per a complir un dels objectius d'aquest treball detectar si un usuari ha caigut. De manera que si estem monitoritzant a l'avi Benet ens avisi l'aplicació quan s'ha produït l'esdeveniment de que l'avi ha caigut.

L'aplicació s'ha desenvolupat de manera que, si el cap i l'estómac són molt a prop del nivell del terra llavors es considera que l'usuari ha caigut.

En el moment que l'usuari cau es mostra un missatge per pantalla (Fig. 3.17) i a més es publica també al mur d'Aaaida.

Fig. 3.17 Caiguda de l'usuari

3.3.4.4. Exercici de rehabilitació

L'aplicació també pot anar dedicada a controlar que l'usuari faci uns determinats exercicis predefinits anteriorment. Aquesta opció pot ser útil per fer rehabilitació i que el responsable pugui dissenyar els exercicis adients i fer un control posterior al mur d'Aaaida.

En aquest cas hem implementat un exercici bastant senzill. L'usuari ha d'aixecar el braç esquerra 5 cops per a realitzar l'exercici correctament. Quan ho hagi fet, sortirà un missatge per pantalla (Fig. 3.18) informant de que s'ha fet l'exercici correctament.

Fig. 3.18 Exercici rehabilitació

3.3.5. Xarxa social Aaaida

Quan es produeixi un esdeveniment en el que es mostri un missatge per pantalla, també ho es publica al mur d'Aaaida [8]. A més, també es publica quan s'arrenca l'aplicació i quan es tanca, veure (Fig. 3.19).

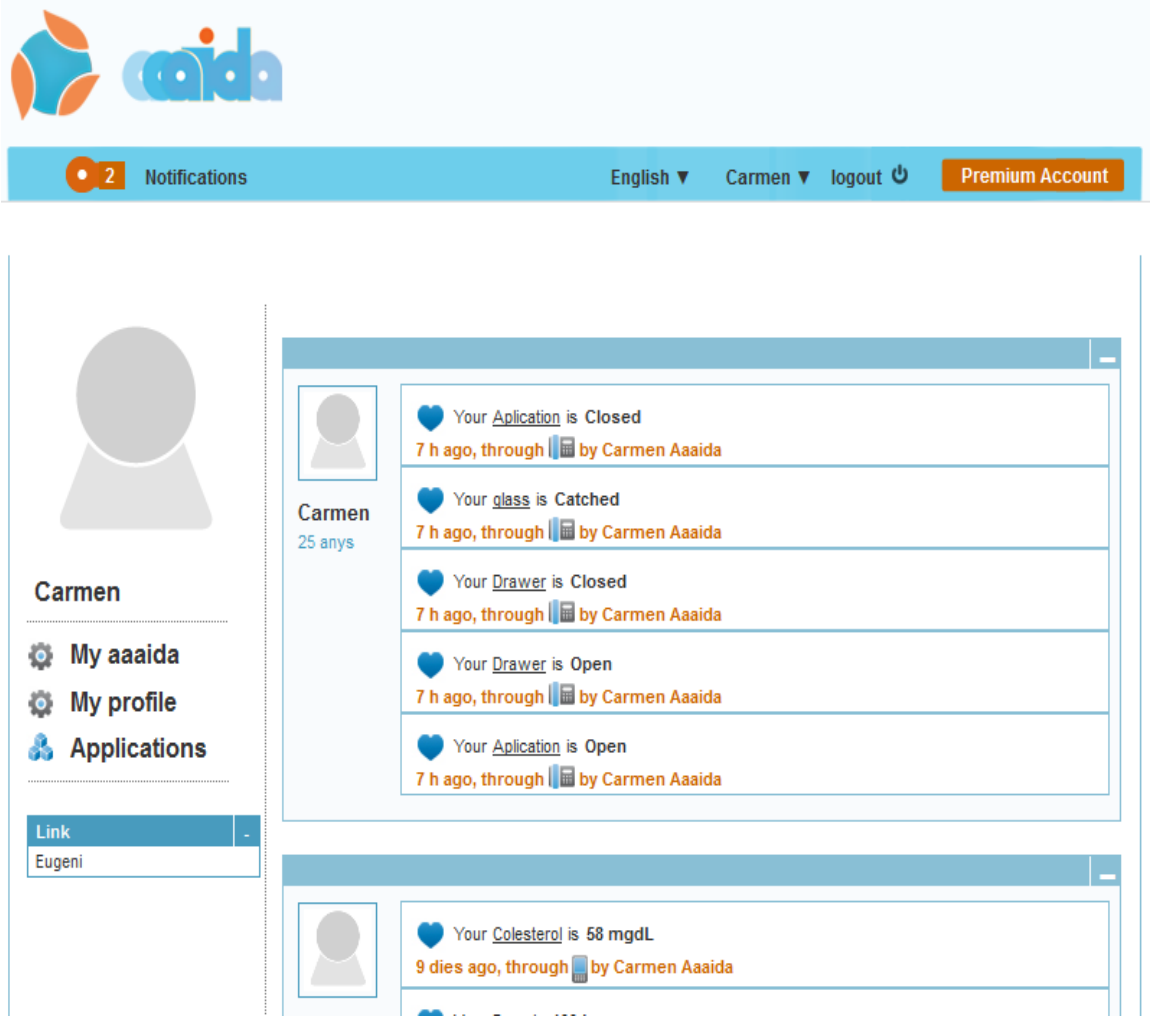


Fig. 3.19 Mur D'Aaaida

Això es farà mitjançant una sèrie de peticions REST cap al servidor remot d'Aaaida. Tenir un servidor remot té forces avantatges, ja que es pot gestionar els missatges des d'una màquina externa i aquesta ja s'encarrega de fer la publicació al mur de la xarxa social. El primer missatge que s'envia es fa quan s'arranca l'aplicació. Aquest missatge serà una petició GET per autenticar-se amb la següent URL:

http://147.83.113.169/MedmonManager/rest-services/medmonRest/getPatientProfileGoogleHealth/{user}/{password}

Els camps *d'user* i *password* s'hauran d'omplir amb el nom d'usuari i la contrasenya d'un usuari de la xarxa d'Aaaida.

Com a resposta d'aquest missatge es rebrà una resposta amb el codi 200 OK informant de que l'autentificació s'ha fet correctament. El contingut de la resposta serà un XML on es trobarà el *tag* <idProfile>, contingut del qual s'haurà de guardar. Aquesta identificació és única per a cada usuari d'Aaaida i és necessària per enviar els missatges per fer les publicacions. Quan es vulgui fer una publicació, s'enviarà un altre petició GET amb la següent URL:

http://147.83.113.169/MedmonManager/rest-services/medmonRest/addLabTestResultGoogleHealth/{userName}/{IDprofile}/{password}/{title}/{content}/{labTestName}/{labTestDate}/{labTestUnit}/{labTestValue}

Els camps *username* i *password* s'ompliran de la mateixa manera que al primer missatge amb els mateixos valors. El camp *IDProfile* s'omplirà amb el contingut del tag idProfile que s'ha rebut dins del XML del missatge de resposta quan s'ha fet l'autenticació. Els camps més importants a omplir són el *title* i el *content*. Tal i com està configurat el servidor remot, les publicacions que es fan al mur d'Aaida només poden ser de l'estil "El teu *title* es *content*". La xarxa Aaaida dóna la possibilitat d'escollir entre els idiomes de català, castellà i anglès. Aleshores els possibles missatges serien "El teu *title* es *content*" en català, "Tu *title* es *content*" en castellà i "Your *title* is *content*" en anglès. Per exemple, un dels esdeveniments que es podria produir és la obertura del calaix. En aquest cas s'ompliria el camp *title* amb el valor "calaix" i el camp content amb "obert". Així el missatge que sortiria al mur seria "El teu calaix és obert". El següent camp a omplir és el *labTestUnit*, que en aquest cas sempre es posarà Kinect per diferenciar les publicacions d'aquesta aplicació de les altres aplicacions que fan servir Aaaida. El camp *labTestDate* s'ompliarà amb la data actual. Això servirà a l'aplicació d'Aaaida per indicar el moment en que s'ha fet la publicació i la quantitat de temps que ha passat des de que s'ha fet aquesta publicació. Per últim, els camps de *labTestUnit* i *labTestValue* són per indicar la unitat de mesura (Kg,l,Km,etc) i el valor d'aquesta mesura. En aquest cas, aquest camps no tenen sentit ja no es quantifica res, així el que es posarà el valor 0.

En aquest cas, tal i com està configurat el servidor remot, la resposta vindrà amb el codi 204 que indicarà que no hi ha contingut. Això significarà que la publicació ha estat correcte. A la següent figura (Fig. 3.20) es pot veure aquest intercanvi de missatges. Els dos primers pertanyen a l'autenticació i la resposta amb *l'idProfile* i els dos següents a una publicació d'un esdeveniment.

Source	Destination	Protocol	Info
84.88.32.58	147.83.113.169	HTTP	GET /MedmonManager/rest-services/me
147.83.113.169	84.88.32.58	HTTP/XML	HTTP/1.1 200 OK
84.88.32.58	147.83.113.169	HTTP	GET /MedmonManager/rest-services/me
147.83.113.169	84.88.32.58	HTTP	HTTP/1.1 204 Sin Contenido

Fig. 3.20 Exemple de peticició a Aaaida

La formació d'aquests missatges HTTP i el seu enviament es faran des de l'aplicació amb la llibreria libcurl. Libcurl és un conjunt de llibreries que permeten una fàcil implementació de clients HTTP, FTP, etc. Quan es produeixi un esdeveniment que es vulgui publicar al mur d'Aaaida, es construirà *l'URL* amb els paràmetres necessaris i s'enviarà la petició.

Per a major facilitat de l'usuari tots els missatges que s'envien cap al mur d'Aaaida, es tenen en una classe a part. Així quan l'usuari vulgui canviar les

dades que es publiquen al mur d'Aaida, només haurà d'anar aquesta classe i canviar el missatge. Això evita que l'usuari hagi de llegir i buscar entre tot el codi per canviar solament una paraula. Els missatges estan formats de la següent manera:

Application:Open

Application seria el camp *title* i *Open* seria el camp *content*. Quan es produeixi un esdeveniment, es cridarà a la funció que retorni el missatge sencer desitjat, llavors l'aplicació separarà aquests dos elements i construirà la URL corresponent.

3.3.6. Diagrama de funcions

A la Fig. 3.21 es pot veure a mode de resum un diagrama on s'explica la lògica de l'aplicació.

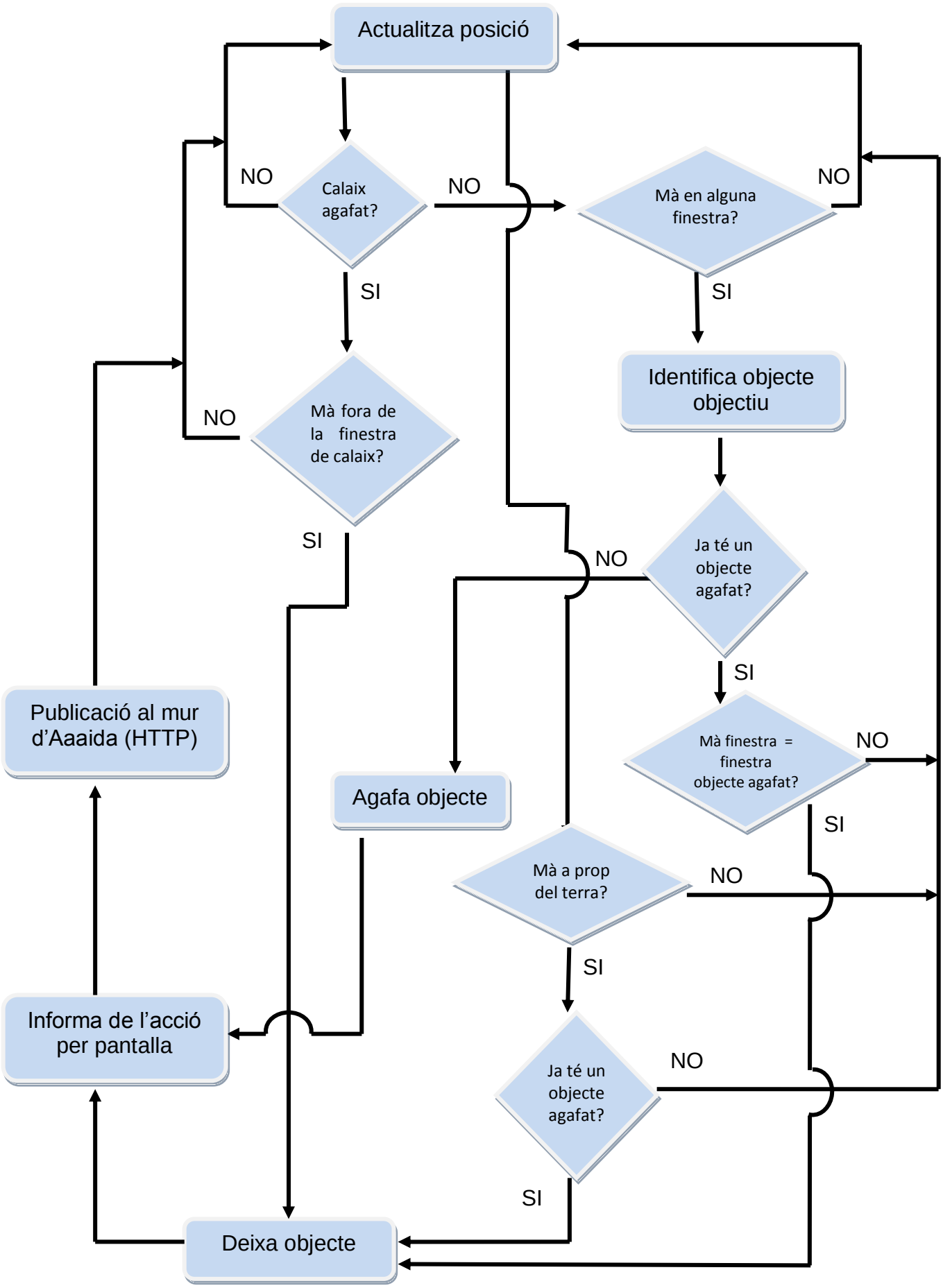


Fig. 3.21 Diagrama funcional

L'estat bàsic del qual es parteix és la constant actualització de la posició de l'usuari. La primera comprovació que es fa és si té el calaix agafat. Si no el té agafat i l'usuari passa la mà per alguna finestra, l'aplicació distingirà i identificarà quin objecte es vol agafar.

Si l'usuari no té cap objecte agafat, l'agafarà directament i l'aplicació mostrarà un missatge per pantalla informant-nos de l'acció realitzada. Seguidament s'enviarà una petició HTTP al servidor d'Aaaida i es publicarà una entrada amb informació de l'acció realitzada. Si ja té qualsevol objecte agafat, l'usuari no podrà agafar un altre i haurà de deixar el que té agafat per agafar un altre.

Quan es tingui el calaix agafat s'haurà de comprovar constantment si la mà es troba fora de la finestra del calaix. Mentre estigui dintre es podrà obrir i tancar el calaix. En el moment que hi sigui fora, llavors es deixarà el calaix i no es podrà continuar obrint i tancant i seguidament es repetiran els passos d'informar per pantalla i publicació a Aaaida.

Per deixar un objecte que es té agafat, excepte el calaix, tenim dues possibilitats. Si es vol deixar-lo al lloc on estava s'haurà de passar la mà pel mateix lloc (mateixa finestra) on l'hem agafat. Com s'ha comentat abans, si ja té un objecte agafat i passa la mà per una altra finestra no podrà agafar aquest objecte sense haver deixat abans el que té agafat. Aleshores s'haurà d'identificar la finestra per on s'ha passat la mà, i si es té un objecte agafat i la finestra és la d'aquest objecte, es deixarà l'objecte a la seva posició original. La segona possibilitat és que l'usuari el vulgui deixar al terra. En aquest cas el que ha de fer és apropar la mà. En cas de que ho faci, si no té cap objecte agafat, no es realitzarà cap acció. En el cas de què sí que tingui algun objecte agafat, l'objecte es desprendrà de la mà i quedarà al terra. Després d'això, sortirà el missatge corresponent per pantalla i es farà la publicació al mur d'Aaaida.

3.4 Entorn de desenvolupament

En aquest apartat s'explicarà amb quines eines s'ha treballat per a poder realitzar aquest projecte.

3.4.1. Visual Studio

Per a la programació en C++ s'ha utilitzat el programa Visual Studio en la seva versió Express 2010, que és gratuïta. Ha estat molt útil sobretot per l'opció del Debug que incorpora, que permet arrencar el programa instrucció per instrucció.

3.4.2. Subversion

La llicència la té Apache [9]. És un software que proporciona un control de versions, mantenint copies de diferents versions dins d'un repositori a punt per la seva recuperació en cas de necessitat.

És possible també accedir a aquestes versions a través de la xarxa de mode que pot ser recuperat des de diferents ordinadors i persones en el cas d'un projecte compartit, com en aquest cas, facilitant per tant el treball en equip. Per al seu ús existeixen molts *plugins* que ajuden a la seva utilització, en aquest cas no s'ha pogut utilitzar el plugin que ens recomana la pàgina web d'apache (AnkhSVN) ja que no funciona per a la versió Visual Studio Express 2010. Per tant s'ha utilitzat el programa TortoiseSVN.

3.4.3. Ordinador

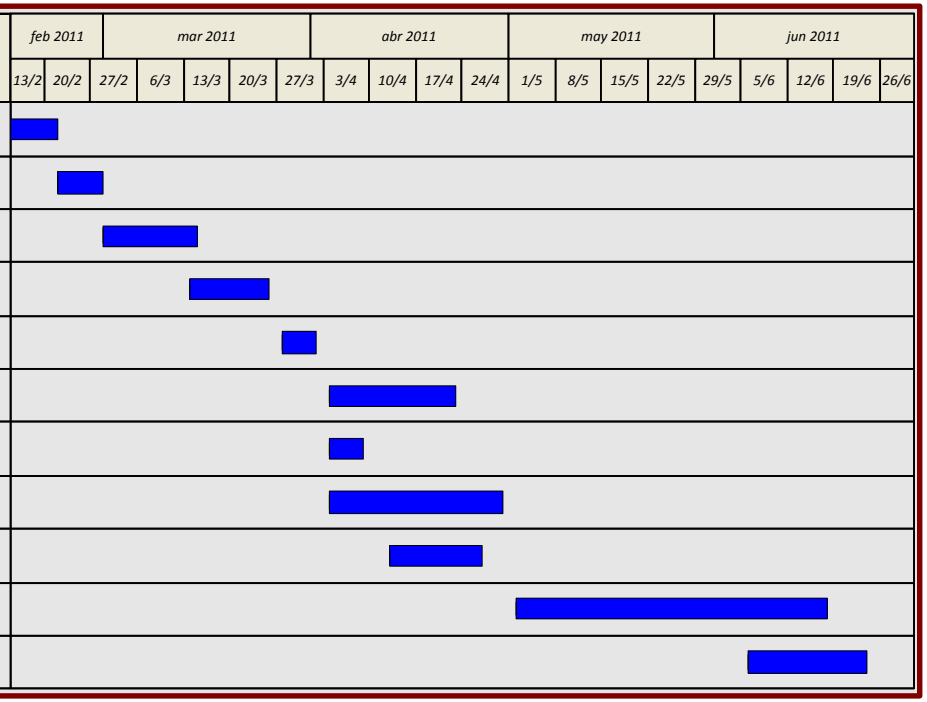
L'ordinador que s'ha utilitzat per aquest projecte té quatre processadors (Quad-Core) de 2,66 GHz cadascun, 4GB de memòria RAM i 1024MB de targeta gràfica. El sistema operatiu que s'ha utilitzat és el Windows 7. Hem pogut experimentar que és suficient per a utilitzar el Kinect.

S'han fet proves amb altres ordinadors de menors capacitats, i s'ha vist que l'aplicació també funcionava en aquells casos. L'únic és que la part 3D es carregava més lentament. Això implica que no es necessita un ordinador de grans capacitats per a fer funcionar l'aplicació.

CAPÍTOL 4. PLANIFICACIÓ

Taula 1 Planificació del projecte

Id.	Nombre de tarea	Comienzo	Fin	Duración
1	Flex y XMPP	15/02/2011	21/02/2011	5d
2	Apache Wicket	22/02/2011	28/02/2011	5d
3	OpenGL/WebGL i HTML5	01/03/2011	14/03/2011	10d
4	Blender	14/03/2011	25/03/2011	10d
5	CL NUI 4	28/03/2011	01/04/2011	5d
6	OPENNI	04/04/2011	22/04/2011	15d
7	NITE	04/04/2011	08/04/2011	5d
8	Disseny	04/04/2011	29/04/2011	20d
9	OGRE3D	13/04/2011	26/04/2011	10d
10	Programació i entendre exemple	02/05/2011	17/06/2011	35d
11	Memòria	06/06/2011	23/06/2011	14d



Totes aquestes tasques del projecte (Taula 1) es poden dividir en 4 blocs principals:

- Aprenentatge de noves tecnologies: Flex, XMPP, Blender, OpenGL, WebGL, HTML5.
- Kinect, aquesta part va ser quan es va estar provant divers pel Kinect, al final ens vam decidir per l'OpenNI (i NITE) en comptes del CL NUI 4 ja que tenia molta més documentació i exemples.
- Disseny: ha estat una de les parts més importants. Primer vam haver de mirar primer tot el codi de l'aplicació Sinbad, exemple de OpenNI amb el que ens hem basat per fer el treball. Seguidament vam pensar quines parts del codi hauríem d'afegir les funcionalitats necessàries per a complir l'objectiu del treball.
- Documentació: preparació de la memòria.

Tot i que la part de Blender hem posat 10 dies l'hem estat utilitzant durant tota la fase de Programació i entendre exemple per a crear el nostre escenari, així com també en la part de Orgre3D, en les conversions de Blender a fitxers amb extensió .mesh (Ogre3D). Els 10 dies de Blender els he considerat com l'aprenentatge inicial.

La *API* de CLNUI 4 cal dir que era massa simple, servia per veure el mapa de profunditat que et donava el Kinect, i poc més.

Com podem veure la part més densa ha estat l'enteniment de l'exemple i la programació d'aquest per fer la nostra aplicació. Seguidament ve la part de disseny que està lligada a la posterior programació. I finalment les parts d'instal·lació i enteniment de com funciona OpenNI, juntament amb la redacció de la memòria.

CAPÍTOL 5. CONCLUSIONS

En aquest apartat s'exposen les conclusions que hem arribat al final del projecte. Primer s'analitzaran els objectius aconseguits i l'impacte mediambiental que té el projecte. Després es farà una valoració personal del mateix i per últim es proposarà una sèrie de millores que es podrien fer en un futur.

5.1 Objectius aconseguits

Durant el desenvolupament de l'aplicació s'ha intentat aconseguir tots els objectius que es van proposar. La majoria d'ells s'han assolit, inclús s'han ampliat com es detallarà a continuació, però d'altres s'han deixat per ampliacions futures.

- S'ha aconseguit que l'usuari pugui visualitzar els seus propis moviments. Els seus moviments són reproduïts per un avatar integrat en un escenari 3D. Així es compleixen dos dels principals objectius que són la visualització de les teves accions en un món virtual.
- És possible reconèixer una sèrie de moviments predefinits com que l'usuari caigui a terra, obrir el calaix, etc. i posteriorment fer les accions pertinents per a cada esdeveniment.
- S'informa a l'usuari dels esdeveniments produïts mitjançant uns missatges que surten per pantalla. S'ha ampliat aquest concepte i també es publiquen els esdeveniments produïts a la web de la xarxa social Aaaida. Amb això s'ha aconseguit que un possible familiar faci un seguiment via web de les accions realitzades per l'usuari.
- S'aconsegueix que l'usuari interactuï amb un món virtual modelat. S'ha ampliat aquest objectiu i oferim la possibilitat d'afegir objectes modelats en 3D a gust de l'usuari.

Com s'ha comentat abans, alguns objectius no s'han assolit. La visualització de l'usuari des d'un equip remot és una possibilitat que es va plantejar però ens va preferir centrar-nos en coses més dirigides al propi desenvolupament de l'aplicació com reconèixer que l'usuari s'ha caigut, etc. Ens va semblar que no tenia sentit fer aquesta tasca si el propi funcionament de l'aplicació no estava ben implementat.

Un altre dels objectius per aconseguir ha sigut la separació de blocs com OpenNI (generació de dades) i Ogre (accions pertinents). És a dir, realitzar un conjunt d'APIs per separar a nivell de codi les nostres implementacions i la generació interna de dades per part d'OpenNI perquè ara mateix es troba tot dintre d'un mateix codi.

Com a conclusió diríem que s'han aconseguit la majoria d'objectius a excepció d'un parell que es podrien deixar per implementacions futures. A més, durant el desenvolupament, s'ha decidit ampliar algun objectiu i afegir noves funcionalitats molt útils com la d'afegir objectes a qualsevol lloc de l'escena.

5.2 Impacte mediambiental

Se sap que les nostres accions, motivades per la conseqüència de diverses finalitats, hi ha cops que provoquen efectes col·laterals sobre el mitjà natural o social. Mentre els efectes de l'objectiu perseguit solen són positius, molts cops els efectes secundaris són negatius.

L'aplicació permet fer rehabilitació a distància, des de la casa del pacient, i monitoritzar una persona a distància. Els centres de rehabilitació estan limitats, és a dir no es té un sempre al costat de casa teva, i els seus recursos són limitats per a poder atendre a totes les persones a la vegada.

El fet de que es pugui fer rehabilitació o monitorització des de casa implica una reducció de costos i augment de l'eficiència de l'ús dels recursos de la sanitat pública. El pacient també pot realitzar els exercicis quan ell disposi de temps a casa seva i no només quan pugui anar cap al centre de rehabilitació, per tant es millora també l'entrenament del pacient.

Això millora la qualitat de vida. A més no cal anar fins a l'hospital a fer els exercicis, ni tampoc cal tenir un cuidador a casa cada dia. Això provoca que no hagis d'agafar un automòbil per anar al centre, per tant es redueix els gasos perjudicials per la salut. També suposa una reducció del soroll, de la sinistralitat, menys congestió i com era d'esperar una menor contribució al canvi climàtic al reduir-se també les emissions de CO_2.

5.3 Conclusions personals

Abans de començar aquest projecte teníem clar que volíem fer un projecte amb ingredients innovadors i que no tingués coses que no s'haguessin fet abans. Quan vam començar el projecte, el dispositiu Kinect era bastant nou i no hi havien gaire projectes relacionats. Per altra banda, els drivers d'OpenNI són una mica primitius i constantment estan sortint noves versions per arreglar els bugs. Aquest fet ens va obligar a llegir molta documentació i a entendre el codi pas per pas perquè no hi havia exemples senzills ni tutorials. També vam trobar l'inconvenient de que els divers encara no suportaven l'utilització dels micròfons per fer reconeixement de veu, fet que creiem que amb el nou SDK ja es podrà implementar. Tot això ens va obligar a fer un alt grau d'autoaprenentatge i fer moltes proves.

Un altre fet que ens va dificultar el desenvolupament va ser el llenguatge que utilitzen tots els mòduls. Tots estan escrits en el llenguatge C++ i encara que a la carrera vam fer alguns exemples, no teníem tanta facilitat com podríem tenir

en Java. Pensem que aquest fet ens ha sigut molt útil perquè la majoria de les llibreries i aplicacions de baix nivell estan escrites en C# o C++ i ens ha donat la possibilitat d'aconseguir experiència per a un futur.

A banda de la dificultat inicial, ens hem donat que aquest dispositiu té unes possibilitats increïbles en molts àmbits i no tenim cap mena de dubte que poc a poc anirà cobrant protagonisme davant dels seus competidors i inclús creiem que intentaran fer altres dispositius semblants.

5.4 Futures millores

En aquest apartat s'explicarà quines noves aportacions es poden fer al nostre projecte per a futures implementacions.

En l'aplicació es té Ogre3D, OpenNI i NITE en el mateix codi. Això provoca que sigui una mica més engavanyador el codi. Es proposa que en un futur es separin aquests dos elements i s'utilitzin *API's* externes, i un protocol de missatgeria com XMPP per a passar-se les dades.

Microsoft ha tret ja el seu propi *SDK [10]* de manera gratuïta per a que la gent pugui fer les seves pròpies aplicacions. La única trava és que només està disponible per a Windows 7. Tot i així ja s'ha pogut veure que té molta documentació i diversos exemples. Una de les noves aportacions serà provar d'utilitzar aquesta nova API per fer aquesta aplicació, ja que aquesta incorpora la possibilitat d'utilitzar tots els micròfons per a reconeixement de veu, etc.

Una incorporació possible al treball és anar afegint nous objectes a l'escenari, només cal modelar-los en Blender. Després es poden afegir a la llista que s'ha creat per carregar l'escenari inicial, tot tenint cura de posar correctament els paràmetres (si és movible, la finestra adient...).

Un altre cas que s'ha pensat seria afegir o treure els objectes per mitjà d'una pàgina web. L'aplicació ja conté els mètodes d'afegir i eliminar objectes, per tant només caldria desenvolupar una interfície web que interactuï amb la nostra aplicació.

A més, el projecte permetria la monitorització remota d'un pacient, només caldria enviar les dades del Kinect d'aquest usuari per mitjà d'un protocol de missatgeria com XMPP cap a la persona q vol monitoritzar-lo.

Tal com s'havia comentat a l'apartat 1.5.3 quan es parlava de telerehabilitació, la implementació de possibles exercicis físics és bastant senzilla. El desenvolupador pot definir una sèrie de moviments amb els braços o cames, i fins que l'usuari no les hagi realitzat, l'exercici no estarà acabat. Aquests exercicis poden ser una opció per fer rehabilitació des de casa.

BIBLIOGRAFIA I REFERÈNCIES

BIBLIOGRAFIA

Apache. *Apache Wicket Examples*. Consultat el 23 / Febrer / 2011, a Apache Wicket Examples: http://wicket.apache.org/learn/examples/

Foundation, Blender. *Blender Tutorials*. Consultat el 20 / Març / 2011, a Blender Tutorials: http://www.blender.org/education-help/tutorials/

Ogre Community. *Ogre - Open Source 3D Graphics Engine*. Consultat el 2011 / Abril / 13, a Ogre - Open Source 3D Graphics Engine: http://www.ogre3d.org/tikiwiki/Tutorials

OpenGL.org. *OpenGL API Code & Tutorials*. Consultat el 4 / Març / 2011, a OpenGL API Code & Tutorials: http://www.opengl.org/code/

OpenNI Organization. (Novembre / 2010). *OpenNI Documentation*. Consultat el 4 / Abril / 2011, a OpenNI Documentation: http://www.openni.org/documentation

OpenNI Organization. (Novembre / 2010). *OpenNI resources*. Consultat el 2011 / Abril / 4, a OpenNI resources: http://www.openni.org/downloadfiles

PrimeSense. *PrimeSense, NITE Middleware*. Consultat el 4 / Abril / 2011, a PrimeSense, NITE Middleware: http://www.primesense.com/?p=515

Team, P. *Desarrollo en WebGL*. Consultat el 3 / Març / 2011, a Desarrollo en WebGL: http://sites.google.com/site/desarrolloenwebgl/

REFERÈNCIES

[1] Projecte de telerehabilitació de l'hospital Son Llàtzer
http://www.elmundo.es/elmundo/2011/05/10/baleares/1305014145.html

[2] Explicació del hardware Wii U
http://www.ddsmedia.net/blog/2011/06/nintendo-presenta-su-wii-u/

[3] Explicació del hardware Kinect
http://esencialbit.com/analisis/como-funciona-kinect/

[4] Explicació Wiimote
http://es.wikipedia.org/wiki/Wiimote

[5] Explicació PlayStation Move
http://es.wikipedia.org/wiki/PlayStation_Move

[6] Explicació Ogre
http://es.wikipedia.org/wiki/OGRE_3D

[7] Explicació de la xarxa social Aaaida
http://www.alteraid.com/2011/04/trontoll-explica-aaaida.html

[8] Pàgina web d'Aaaida
http://147.83.113.169/medmon/home/BodySignIn

[9] Web oficial del software Apache Subversion
http://subversion.tigris.org/

[10] Pàgina principal del nou SDK de Microsoft
http://research.microsoft.com/en-us/um/redmond/projects/kinectsdk/

ACRÒNIMS

API (*Application Programming Interface*) – Conjunt de funcions i procediments per a ser utilitzats per una altra aplicació com a cap d'abstracció.

FTP (*File Transfer Protocol*) – Protocol de xarxa per a la transferència d'arxius entre sistemes connectats a una xarxa TCP basat en l'arquitectura client-servidor.

HTML (*HyperText Markup Language*) – Llenguatge basat en etiquetes predominant per a l'elaboració de pàgines Web. És utilitzat per a descriure l'estructura i el contingut en forma de text, així com per a complementar el text amb objectes tals com imatges.

HTTP (*Hypertext Transfer Protocol*) – Protocol sense estat de la capa d'aplicació utilitzat en cada transacció de la World Wide Web.

OpenNI (*Open Natural Interaction)* – És un *framework* multi-plataforma que defineix unes API de codi obert desenvolupades en el llenguatge C++ i que es poden utilitzar per desenvolupar aplicacions que utilitzen Interacció Natural

REST (*Representational State Transfer*) – Tècnica d'arquitectura software per a sistemes hipermedia distribuïts com la xarxa Internet.

URL (*Uniform Resource Locator*) – Seqüència de caràcters, d'acord a un format modèlic i estàndard, que s'utilitza per a nombrar recursos a Internet per la seva localització o identificació, com per exemple documents textuals, imatges, vídeos, presentacions digitals, etc.

XML (*eXtensible Markup Language*) – Metallenguatge d'ús general que serveix per a definir altres llenguatges de programació o formats d'intercanvi d'informació segons diverses necessitats.

XMPP (*Extensible Messaging and Presence Protocol*) – És un protocol obert i extensible basat en XML, originalment ideat per a missatgeria instantània.

ANNEXES

A. Ampliació Kinect

A.1 OPENNI

A.1.1. Mòduls

L'API d'OpenNI permet utilitzar dispositius físics i components de middleware, els quals s'anomenen mòduls. Aquests mòduls es separen en dos tipus, els que generen les dades i els que les processen. Evidentment no es pot utilitzar qualsevol mòdul amb OpenNI, sinó que hauran d'implementar certes interfícies Els mòduls que actualment suporta OpenNI per la generació de dades són:

- Sensor 3D
- Càmera RGB
- Càmera d'infrarojos
- Dispositiu d'àudio (micròfon)

D'altra banda els mòduls suportats pel processament de dades són:

- Anàlisi del cos complet: component de software que processa la informació sensorial i genera les dades relatives al cos (típicament una estructura de dades que descriu punts d'articulació, orientació, centre de massa, etc).

- Anàlisi de "hand point": component de software que processa la informació sensorial i genera la localització d'un hand point. (Ex. Un hand point pot ser el centre de la mà, el genoll,etc.).

- Detecció de gestos: component de software que identifica gestos predefinits i alerta a l'aplicació. (Ex. Una mà fent el gest de saludar).

- Analitzador d'escena: component de software que analitza la imatge de l'escena per produir informació sobre la identificació individual de les figures de l'escena o la distinció entre el pla principal on es troben les figures i el pla de fons.

A.1.2. Nodes de producció

Normalment, les dades que recull el dispositiu és un mapa de profunditat, on cada píxel es representat per la seva distància al sensor. Després, el dispositiu de *middleware* recollirà aquestes dades i les processarà per que puguin ser

utilitzades per l'aplicació. Els nodes de producció són uns components que tenen un rol productiu en el procés de creació de dades requerides per les aplicacions d'Interacció Natural. Cada node encapsula la funcionalitat que es refereix a la generació de dades d'un tipus específic. Aquest nodes, són els elements fonamentals que les interfícies d'OpenNI ofereixen a les aplicacions.

Però, l'API dels nodes de producció solament defineixen el llenguatge. La lògica de la generació de dades han de ser implementades pels mòduls que utilitzaran OpenNI.

En principi, cada node es una unitat independent que produeix un tipus específic de dades, i pot proporcionar-les a un objecte, a un altre node de producció a l'aplicació mateixa. Normalment, els nodes de producció utilitzen a altres nodes que representen tipus de dades de més baix nivell, analitzen aquestes dades de baix nivell i les converteixen en dades de més alt nivell.

Ex: L'aplicació vol fer un seguiment del moviment d'una figura humana en una escena 3D. Això requereix un node de producció que proporcioni les dades del cos a l'aplicació, anomenat generador d'usuari. Aquest necessita les dades de més baix nivell d'un altre node, el generador de profunditat. Aquest, és implementat pel sensor, el qual agafa les dades proporcionades pel sensor de profunditat (flux de dades a *N* imatges per segon) i extreu un mapa de profunditat, que és recollit pel generador d'usuari.

Els nodes de producció són de diferents tipus i s'agrupen en dos categories, els nodes de producció relacionats amb el sensor, i els relacionats amb el middleware. Actualment, els tipus de nodes que suporta OpenNI per a la primera categoria són:

- Dispositiu: node que representa un dispositiu físic (càmera RGB, sensor de profunditat). La seva principal funció es permetre la configuració del dispositiu.

- Generador de profunditat: node que genera un mapa de profunditat.

- Generador d'imatge: node que genera els mapes d'imatge en color.

- Generador d'infrarojos: node que genera els mapes d'imatge d'infrarojos.

- Generador d'audio: node que el flux d'àudio.

En tots els casos, el dispositiu o el sensor, ha d'implementar el seu node corresponent si desitja ser certificat com a compatible amb OpenNI.

En el cas dels nodes relacionats amb el *middleware*, els suportats són:

- Generador d'alerta de gestos: genera els retorns de crida (*callbacks*) a l'aplicació quan un determinat gest és identificat.

- Analitzador d'escena: analitza l'escena identificant el pla principal del de fons, els diferents usuaris de l'escena, etc. La sortida principal és un mapa de profunditat etiquetat, on cada píxel té una etiqueta on s'especifica si forma part d'una figura o del pla de fons. Normalment s'utilitza OpenGL per a la visualització d'aquesta escena.

- Generador de "*hand point*": suporta la detecció i el seguiment de la mà. Genera els retorns de crida que produeixen alertes quan la mà és detectada o quan està sent seguit el seu moviment i canvia la seva posició.

- Generador d'usuari: genera la representació parcial o total del cos en la escena 3D.

A part d'aquests nodes, també són suportats els nodes per gravació, per reproduir les dades gravades i per comprimir o descomprimir les dades gravades.

A.1.3. Cadenes de producció

OpenNI permet el registre de diversos components de *middleware* i dispositius a la vegada. En l'exemple d'abans, quan volíem fer un seguiment del moviment d'una figura humana, un node agafava les dades d'un altre i la seqüència de nodes era generador d'usuari => generador de profunditat. Aquesta cadena de nodes és el que s'anomena cadena de producció.

Els diferents proveïdors poden tenir diferents implementacions sobre el mateix tipus de nodes. És per això, que OpenNI permet especificar quina cadena de producció es vol utilitzar segons les necessitats o les preferències.

Normalment, les aplicacions solament estan interessades en el node que genera les dades pràctiques d'alt nivell (per exemple, generador de *hand point*). OpenNI permet utilitzar aquest últim node sense preocupar-se de la cadena de producció que hi ha per sota d'aquest node, encara que existeix la possibilitat de configurar aquesta cadena a gust del desenvolupador.

A.1.4 Capacitats

El mecanisme de capacitats ajuda a la flexibilitat del registre de múltiples components de *middleware* i dispositius a OpenNI. Cada proveïdor pot tenir diferents capacitats i opcions de configuració per als seus nodes de producció, és per això que la API d'OpenNI defineix algunes extensions. Aquestes extensions opcionals s'anomenen capacitats, i proporcionen una funcionalitat addicional, delegant la responsabilitat al proveïdor la decisió d'implementar o no les extensions que vulgui segons les seves necessitats. Un node de producció pot ésser preguntat per si suporta una capacitat específica, i si la suporta, aquestes funcions poden ser cridades pel node.

OpenNI, a part de de venir de sèrie amb algunes capacitats, permet afegir de noves. Cada node pot declarar els capacitats que suporta. Posteriorment, quan es demana l'enumeració de les cadenes de producció, la aplicació pot especificar quina capacitat vol que els nodes suportin, i solament es retornaran els mòduls que suportin aquesta capacitat. Algunes de les capacitats que actualment se suporten són:

- Sincronització de *frames*: permet que dos sensors generant dades de *frame* (per exemple, profunditat i imatge) es sincronitzin per a que els seus *frames* arribin al mateix temps.

- Mirall: permet la creació del reflex de les dades produïdes pel generador. És útil quan el sensor es troba en front de l'usuari, així quan la imatge capturada és reflexada, la mà dreta de l'usuari apareixerà com l'esquerra de la figura reflexada.

- Detecció de posicionament: permet al generador d'usuari reconèixer quan l'usuari es col·loca en una determinada posició.

- *Skeleton*: permet al generador d'usuari generar les dades corresponents a l'esquelet de l'usuari. Aquestes dades inclouen la localització dels punts d'articulació, la capacitat de seguir les posicions de l'esquelet i les capacitats de calibratge de l'usuari.

- Posició d'usuari: permet al generador de profunditat optimitzar el mapa de profunditat que és generat per a una àrea específica de l'escena.

- *Error State*: permet a un node informar de que està en estat d'error i per tant no realitzarà les seves funcions.

A.2 OGRE 3D

Ogre3D [6] (acrònim de l'anglès Object-Oriented Graphics Rendering Engine) és un motor de renderitzat 3D orientat a escenes, escrit en el llenguatge de programació C++ i utilitzant Programació Orientada a Objectes.
Les seves biblioteques eviten la dificultat de la utilització de capes inferiors de llibreries gràfiques com OpenGL i Direct3D, i a més, proveeixen una interfície basada en objectes del món i altres classes d'alt nivell.
Actualment Ogre funciona en Windows, Linux i Mac OSX usant *plugins* per utilitzar la *API* de renderització (actualment Direct3D o OpenGL). Les aplicacions que utilitzen Ogre són portables entre plataformes i sistemes. Els blocs més importants de Ogre són:

- Control de l'Escena: serveix per tractar amb els continguts de l'escena (com està estructurada, com es veu des de les càmeres, etc.)

- Control de Recursos: tots els renderitzats necessiten recursos (la seva geometria, textures, fonts de lletra, i altres.)

- Renderització: finalment els continguts es visualitzen en la pantalla. Els objectes específics de l'API del sistema de renderització com buffers i estats de renderització són activats durant procés.

Per a la nostra aplicació els objectes Mesh tenen una gran importància. Un objecte Mesh representa a un model discret, un conjunt de geometria que normalment és petit comparat amb l'escala del món. Els objectes Mesh representen objectes mòbils. Amb l'aplicació Blender, modelarem l'objecte desitjat, dotant-ho de moviment si ens cal (ex. Esquelet humà). Posteriorment s'exportarà a un objecte Mesh de Ogre i així podrem animar els nostres models 3D ja siguin persones o objectes.

A.3 OpenGL

OpenGL (Open Graphics Library) és una especificació estàndard que defineix una API multillenguatge i multiplataforma per escriure aplicacions que produeixin gràfics 2D i 3D. La interfície consisteix en més de 250 funcions diferents que es poden utilitzar per dibuixar escenes tridimensionals complexes a partir de primitives geomètriques simples, tals com a punts, línies i triangles. OpenGL té dos propòsits essencials:

- Ocultar la complexitat de la interfície amb les diferents targetes gràfiques, presentant al programador una API única i uniforme.

- Ocultar les diferents capacitats de les diverses plataformes hardware, requerint que totes les implementacions suportin la funcionalitat completa d'OpenGL.

OpenGL és una API basada en procediments de baix nivell que requereix que el programador dicti els passos exactes necessaris per renderitzar una escena. El disseny de baix nivell d'OpenGL requereix que els programadors coneguin en profunditat la pipeline gràfica (accepta primitives com punts o línies i les converteix en píxels) a canvi de donar-los llibertat per implementar nous algorismes gràfics.

A.4 Blender3D

Blender és un programa informàtic multiplataforma, dedicat especialment al modelatge, animació i creació de gràfics tridimensionals. És gratuït i de codi obert. Algunes de les característiques que ens ofereix són:

- Característiques interactives per a jocs com a detecció de col·lisions, recreacions dinàmiques i lògica.
-Motor de jocs 3D integrat.
- Capacitat per a una gran varietat de primitives geomètriques, incloent

corbes, malles poligonals, buits, NURBS, metaballs.

En el nostre cas, ho utilitzarem per modelar el món virtual amb el qual l'usuari interactuarà.