
Aplicación Interactiva para pacientes con Deficiencias Cognitivas Leves



Trabajo de Fin de Grado en Ingeniería Informática

David Salcedo Gutiérrez
Christian Gómez Redondo

Profesor director: Dr. Segundo Esteban San Román
Profesor subdirector: Dr. José Miguel Montañana Aliaga

Departamento Arquitectura de Computadores y Automática
Facultad de Informática
Universidad Complutense de Madrid

Junio 2016
Curso académico: 2015/2016

Documento maquetado con T_EX[!]S v.1.0+.

Este documento está preparado para ser imprimido a doble cara.

Aplicación Interactiva para pacientes con Deficiencias Cognitivas Leves

Trabajo de Fin de Grado en Ingeniería Informática

**David Salcedo Gutiérrez
Christian Gómez Redondo**

*Profesor director: Dr. Segundo Esteban San Román
Profesor subdirector: Dr. José Miguel Montañana Aliaga*

**Departamento Arquitectura de Computadores y Automática
Facultad de Informática
Universidad Complutense de Madrid**

**Junio 2016
*Curso académico: 2015/2016***

Copyright ©

Contenido: **David Salcedo Gutiérrez y Christian Gómez Redondo;**
Plantilla \LaTeX : **Marco Antonio y Pedro Pablo Gómez Martín**

*A todos nuestros familiares y amigos,
que siempre nos han apoyado y ayudado
durante la realización de este proyecto.*

Agradecimientos

En primer lugar, quisiéramos dar las gracias a nuestro subdirector de Trabajo de Fin de Grado, Jose Miguel Montañana Aliaga, por permitirnos trabajar en este proyecto y por estar siempre dispuesto a facilitarnos cualquier ayuda o material para el desarrollo del mismo.

Mencionar también a todas las personas que han evaluado nuestra aplicación y han dedicado una parte de su tiempo a valorarla y probarla, especialmente a nuestras familias y amigos.

Agradecer a David Hernando Hernández (ex-alumno de la Facultad de Informática y ex-miembro de este proyecto) su aportación en la organización de esta memoria.

Y sobre todo, quisiéramos dar las gracias a nuestro director de Trabajo de Fin de Grado, Segundo Esteban San Román por hacerse cargo de dirigir nuestro proyecto y guiarnos para conseguir nuestros objetivos.

Resumen

La Realidad Virtual es una tecnología que representa un escenario a través de medios electrónicos, permitiendo al usuario interactuar con el mismo. En esta memoria describimos el uso de esta tecnología para pacientes con Deficiencias Cognitivas Leves (DCL), con el fin de controlar sus deficiencias y mejorar en la medida de lo posible su calidad de vida.

La aplicación muestra un interfaz capaz de captura en 3D el movimiento de los usuarios. Esta funcionalidad permitirá al usuario interactuar con el sistema sin necesidad de disponer de ningún dispositivo físico de control.

Palabras clave: Realidad Virtual, DCL, Kinect.

Abstract

Virtual Reality entails presenting our senses with a computer generated virtual environment, which can be explored and interacted with by a person. This technology was already used in previous works for assessing patients with Mild Cognitive Impairment (MCI), and helping them to improve their lives.

In this project is developed a software application which with the hardware support of a Kinect can capture the user movements in 3D. This functionality allows to users with movement disabilities to interact with the system without the need of a physical interface. It will be of great value for elderly with a functional movement disorder.

Keywords: Virtual Reality, MCI, Kinect.

Índice

Autorización	VII
Agradecimientos	IX
Resumen	XI
I Memoria	1
1. Introducción al Proyecto	3
1.1. Introducción	3
1.1.1. Antecedentes	3
1.1.2. Objetivos y motivación	4
1.1.3. Plan de trabajo y organización de la memoria	5
1.2. Introduction	5
1.2.1. Previous Work	5
1.2.2. Goals and motivation	6
1.2.3. Workplan and report organization	6
2. Realidad virtual y su aplicación en el tratamiento de DCL	9
2.1. Introducción y contexto de la realidad virtual	9
2.2. Concepto de realidad virtual	11
2.3. Tecnologías necesarias para la ejecución de realidad virtual	11
2.4. Ventajas y desventajas generales de la realidad virtual	12
2.5. Importancia de la realidad virtual para personas con deficiencias cognitivas leves	12
2.6. Conclusiones	13
3. Arquitectura y tecnologías utilizadas en nuestra aplicación	15
3.1. Visual Studio 2015, C-Sharp y Windows Presentation Foundation	15
3.2. Dispositivo Kinect	16

3.2.1.	Definición y características	16
3.2.2.	Primer contacto con SDK v1.8	17
3.2.3.	Uso y final implementación del SDK Developer's Toolkit	18
3.3.	Base de Datos	18
3.3.1.	Azure	18
3.3.2.	SQL Server	19
3.4.	Aforge: Grabación de acciones de usuario	19
3.5.	Necesidad de realizar un control de versiones	19
4.	La Aplicación	21
4.1.	Contexto	21
4.2.	Funcionalidades	21
4.2.1.	Inicio y Registro	22
4.2.2.	Navegar por los menús	22
4.2.3.	Obtener ayuda	22
4.2.4.	Realización del puzzle	22
4.2.5.	Realización de compra	23
4.2.6.	Grabar sesión	23
4.3.	Diseño e implementación	25
4.3.1.	Patrón de diseño MVVM	25
4.3.2.	Diseño base de datos	25
4.3.3.	Usuario	27
4.4.	Vistas Principales	27
4.4.1.	Inicio y registro	28
4.4.2.	Elegir juego	28
4.4.3.	Puzzle	29
4.4.4.	Ventana juego puzzle	29
4.4.5.	Supermercado	30
4.4.6.	Ventana juego supermercado y lista de la compra	30
4.4.7.	Ventana base de datos de los usuarios	31
4.5.	Problemas detectados	31
4.5.1.	Biometría facial	32
4.5.2.	Dedos en la kinect	32
4.5.3.	Azure	32
5.	Evaluación con usuarios	33
5.1.	Plan de evaluación	33
5.1.1.	Propósito de la evaluación	34
5.1.2.	Requisitos para los participantes	35
5.1.3.	Diseño experimental	35
5.1.4.	Tareas a realizar	36

5.1.5.	Entorno y herramientas empleadas	36
5.1.6.	Obtención de feedback de los participantes	36
5.1.7.	Tareas del moderador	37
5.1.8.	Descripción de la metodología de análisis de datos	37
5.2.	Evaluación	37
5.2.1.	Informe de resultados y soluciones	37
5.2.2.	Conclusiones más relevantes	38
6.	Conclusiones y trabajo futuro	41
6.1.	Conclusiones	41
6.2.	Conclusions	42
6.3.	Líneas de trabajo futuras	43
6.3.1.	Actualización de Kinect	43
6.3.2.	Biometría	43
6.3.3.	Nuevos juegos e implementar nuevos gestos	43
6.3.4.	Mejorar la aplicación	44
6.3.5.	Azure	44
6.3.6.	Mejorar la gestión de las grabaciones de los usuarios	44
7.	Organización del trabajo y aportaciones.	45
7.1.	Organización del equipo	45
7.1.1.	Herramientas de comunicación	45
7.1.2.	Herramientas de organización	46
7.2.	Aportaciones al proyecto	46
7.2.1.	Christian Gómez Redondo	46
7.2.2.	David Salcedo Gutiérrez	48
	Bibliografía	51

Índice de figuras

2.1. Aplicación THOFU [28]	10
3.1. Kinect [12]	16
4.1. Gestos Push y Grip	22
4.2. Código para grabar sesión	24
4.3. Representación del patrón MVVM	25
4.4. Diagrama de la base de datos	26
4.5. Diagrama de clase de inicio y registro	26
4.6. Caso de uso del usuario	27
4.7. Ventanas de Inicio y registro	28
4.8. Ventana de selección de juego	28
4.9. Ventana de selección de dificultad y ventana de ayuda (puzzle)	29
4.10. Ventana del juego puzzle	29
4.11. Ventana de selección de dificultad (supermercado)	30
4.12. Ventanas de lista de la compra y del juego supermercado	31
4.13. Base de datos del juego puzzle y supermercado	31
7.1. Juego Torres de Hanoi	47
7.2. Paint	49

Índice de Tablas

3.1. Características de Kinect [4]	17
----------------------------------------------	----

Parte I

Memoria

Capítulo 1

Introducción al Proyecto

1.1. Introducción

La realidad virtual es una tecnología que combina la visualización de un escenario virtual con elementos reales interactivos en tiempo real. Aunque hace unos años esta tecnología no estaba muy extendida y era costosa, a día de hoy se aplica en multitud de campos, siendo el de los videojuegos y la medicina donde más se está desarrollando.

La realidad virtual ha originado grandes avances en la medicina, que se han traducido en mejoras terapéuticas, mayor certeza al realizar un diagnóstico, comodidades para el paciente, etc. [30]

A pesar de que estas técnicas no son nuevas, es a día de hoy, cuando esta tecnología es más accesible y menos costosa. Esto nos ha permitido el desarrollo de nuestro proyecto, destinado a una mejora en los diagnósticos de pacientes con deficiencias cognitivas leves, así como, una manera más cómoda para los mismos al realizar las pruebas, ya que pueden ejecutarlas desde la comodidad de su casa.

1.1.1. Antecedentes

El proyecto que vamos a describir en esta memoria, no está basado en ningún proyecto previamente desarrollado en la Universidad Complutense de Madrid (UCM), ya que, no hay ninguno desarrollado que se encargue de relacionar las deficiencias cognitivas leves (DCL) con realidad virtual y Kinect.

No obstante, sí hemos podido contar con ideas de otros proyectos que tratan las DCL sin realidad virtual y proyectos que usan la realidad virtual pero no tratan temas de DCL.

1.1.2. Objetivos y motivación

En la creación de este proyecto nos encontramos como principal objetivo realizar una aplicación que permita proporcionar a los médicos información para el diagnóstico de sus pacientes y a estos últimos realizar sus pruebas médicas de una forma innovadora y divertida, de manera que el usuario tenga una buena experiencia.

Para poder conseguir este objetivo hemos presentado al usuario nuestra aplicación con la cual puede interactuar, mediante gestos y movimientos, sobre un escenario que simula varios tipos de juegos, gracias al uso de la realidad virtual usando el dispositivo Kinect. Resumiendo, nuestro trabajo ha consistido en desempeñar las siguientes tareas:

- Buscar una librería que se asocie bien con Kinect que nos permita cumplir los requisitos que tenemos.
- Investigar sobre esa librería y familiarizarnos con ella.
- Analizar el uso de la realidad virtual en la medicina.
- Estudiar cómo se organiza un proyecto en C Sharp junto con el uso de la Kinect.
- Diseñar e implementar una aplicación haciendo uso de lo estudiado anteriormente.
- Realizar evaluaciones con usuarios para evaluar la experiencia de uso de la aplicación.

Por otro lado, también nos motivaba el hecho de adentrarnos y trabajar con una tecnología no muy común como es la realidad virtual y de la cual no hemos visto nada en la carrera.

Nuestra idea era realizar un proyecto que fuese usado por usuarios reales. Personalmente, esta era nuestra principal motivación, desarrollar una aplicación que, de una forma entretenida y divertida, ayudase a los pacientes a la hora de realizar sus pruebas de evaluación.

1.1.3. Plan de trabajo y organización de la memoria

Nuestra planificación constó en documentarnos sobre la realidad virtual, para luego investigar y conocer más usos sobre ella (Capítulo 2).

Realizamos después un estudio completo sobre que entorno de desarrollo vamos a escoger y sobre las librerías de Kinect que más se ajustaban a nuestras necesidades (Capítulo 3).

Una vez que nos documentamos bien e hicimos algunas aplicaciones con Kinect, nos reunimos con nuestro director y subdirector, para ver cuál sería la aplicación final que desarrollaremos según los objetivos marcados (Capítulo 4).

Con una versión final de la aplicación, realizamos evaluaciones con usuarios para poder detectar posibles errores (Capítulo 5). A continuación obtenemos las conclusiones y enunciamos posibles mejoras de futuro (capitulo 6).

Terminamos indicando en que contribuyó cada integrante del grupo y la organización final que seguimos (Capitulo 7).

1.2. Introduction

Virtual reality is a technology that combine the visualization of a virtual environment with real elements in real time. Although some years ago this technology wasn't very common and it was too expensive, today this technology is used in many fields, but the field's most developed are videogame and medicine.

Virtual reality has originated many advances in medicine, which they have translated into therapeutic improvements, greater certainty of diagnosis, improvement patient's comfort, etc.

Although these techniques aren't new, it is nowadays when this technology is less expensive and more accessible. This give us the chance to develop ours project, ours project is destined to improve the diagnosis of the patients with Mild Cognitive Impairment, as well as, more comfortable way to perform tests to the user, because they can do it while they are in their home.

1.2.1. Previous Work

The project that we are going to comment in this memory, it isn't based in any other developed project in the Universidad Complutense de Madrid

(UCM), as there isn't any developed project that related the Mild Cognitive Impairment (MCI), with the virtual reality and kinect.

However, we take ideas of project that treat the WDC without virtual reality and projects that use virtual reality without treat MCI.

1.2.2. Goals and motivation

In the Creation of this project, we have as principal objective develop an application that allowed the medic get information about the patient's diagnosis and these last one perform the medic's test in an innovative and fun way, obtaining the user a good experience.

To get this objective we have presented the user our application with which can interact using gestures and movement onto scenarios that simulate different types of games, this is possible with the help of virtual reality and the device Kinect. Summarizing, Our project consists in the performance of the following tasks:

- Search for a library that goes well with the Kinect that allows us to satisfy the requirements.
- Investigate the library and learn how to use it.
- Analyze the use of virtual reality in medicine.
- Study how to organize an application using C Sharp and Kinect.
- Design and implements an application using C Sharp and Kinect.
- Perform evaluations with users to evaluate the experience using the application.

On the other hand,our motivation was the fact of going deeper and work with an uncommon technology like virtual reality ,which we haven't seen nothing in the career

Our idea was to perform a project that it could be use by real user, that was our principal motivation. Personally, this was our principal motivation,to develop an application that it could be entertainment and funny assisting the patients with the evaluation test.

1.2.3. Workplan and report organization

Our planning consisted in document ourselves about virtual reality, then uses research and learn more about it (Chapter 2).

Then we perform a deep study of the developer environment that we are going to choose and the libraries of Kinect that best fit our needs (Chapter 3)

Once we document ourselves and develop some applications with Kinect, we met with our director and subdirector, to analyze which application could be developed according to the objectives(Chapter 4)

With a stable version of the application , we perform evaluations with users in order to detect possible mistakes (Chapter 5). Later we obtain conclusion and we expose possible future work improvement (Chapter 6).

Finally, we will expose the input of all the members of the project and the finally taken organization (Chapter 7).

Capítulo 2

Realidad virtual y su aplicación en el tratamiento de DCL

La realidad virtual es una tecnología relativamente nueva y desconocida que avanza a pasos agigantados ante las necesidades tecnológicas de la sociedad actual. Es por esto que, al comienzo del proyecto, necesitábamos conocer el marco en el que se utiliza la realidad virtual y en qué consiste exactamente para poder construir una aplicación sujeta a nuestros objetivos.

Por esto mismo, era fundamental para nosotros conocer cuáles son los contextos en los que se puede utilizar la realidad virtual como herramienta de mejora de la sociedad, ya sea en educación, entretenimiento, medicina, etc. Nuestro objetivo realizar una aplicación que ayudase a los pacientes con DCL, por lo que vimos interesante centrarnos en descubrir qué se ha hecho sobre esta enfermedad y conocer cuáles son las principales necesidades.

En resumen, en este capítulo podremos ver con detalle en qué consiste la realidad virtual, en qué contextos se usa y la aplicación que tiene en el tratamiento de las DCL.

2.1. Introducción y contexto de la realidad virtual

La investigación y creación de software donde se hace uso de la realidad virtual (en adelante RV) está creciendo en los últimos tiempos. Sus múltiples aplicaciones y su atractivo aspecto visual está provocando que cada vez más los desarrolladores del campo de la informática y de las nuevas tecnologías se interesen por ella y la incluyan en sus proyectos.

En la actualidad se está produciendo un aumento muy significativo del número de dispositivos móviles así como de las prestaciones de los mismos, lo que está favoreciendo el surgimiento de aplicaciones cada vez más sofisticadas.

A día de hoy hay muchas aplicaciones interesantes sobre RV que, aún no llegando a sacar el máximo rendimiento de dicha tecnología, nos ofrecen la posibilidad de pensar que la RV estará muy presente en nuestra vida diaria en un futuro cercano. Por ejemplo, nos llama la atención la aplicación THOFU (Tecnologías para el Hotel del Futuro), que permite a los usuarios planificar las rutas a través de un dispositivo móvil. Consiste en un servicio de presentación de indicaciones para guiar a un usuario desde su posición hasta determinados lugares de interés y proporcionar señales de evacuación en caso de emergencia. [27]



Figura 2.1: Aplicación THOFU [28]

La RV está teniendo un mayor crecimiento en los videojuegos. Cada año las videoconsolas ofrecen mejores prestaciones para que el usuario tenga una mejor percepción e inmersión en el juego, provocando una forma de jugar más completa y divertida.

Esta tecnología también es muy útil a la hora de usarse para la enseñanza. Por ejemplo, si quieres aprender a tocar un instrumento y no cuentas con el material necesario o con un profesor que te enseñe, la RV te puede proporcionar un escenario donde tengas la oportunidad de poder tocar un instrumento y también enseñarte como hacerlo de manera correcta. [23]

Por otro lado, la RV también es un método potencialmente prometedor para el tratamiento de algunas fobias. Por ejemplo, algunos psiquiatras ya la están usando para ayudar a sus pacientes a sobrellevar su miedo a volar o a lidiar con su claustrofobia, ya que esta tecnología proporciona un ambiente controlado en donde los pacientes pueden enfrentarse a sus miedos e incluso poner en práctica algunas soluciones, todo en un ambiente privado y seguro que sea cómodo para las personas. [28]

Como hemos podido ver la realidad virtual puede tener muchos usos, y esto junto al desarrollo de las dispositivos tecnológicos y la facilidad de accesibilidad para el público, provoca que tenga tal crecimiento e índice de desarrollo.

2.2. Concepto de realidad virtual

La tecnología de la Realidad Virtual es un entorno en tres dimensiones sintetizado por computadora en el que varios participantes acoplados de forma adecuada pueden atraer y manipular elementos físicos simulados en el entorno y, de alguna manera, relacionarse con las representaciones de otras personas pasadas, presentes o ficticias o con criaturas inventadas. [23]

Las propiedades principales de los sistemas de realidad virtual son:

- **Interacción:** Permite al usuario actuar sobre el escenario, permitiendo que el sistema responda a sus estímulos; creando interdependencia entre ellos.
- **Inmersión:** Es la capacidad de bloquear toda distracción y enfocarse solo en la información u operación que se está simulando.
- **Simulación:** Esta propiedad se encarga de la manipulación de los sentidos del usuario. Por ejemplo aplicar la sensación de profundidad que afectaría a la visión o si hablamos del oído los sonidos tienen efectos estereofónicos (direccionalidad). [22]

2.3. Tecnologías necesarias para la ejecución de realidad virtual

Para poder disfrutar de aplicaciones de RV, se necesita un dispositivo que soporte esta tecnología. Este dispositivo va a tener como mínimo las siguientes capacidades:

- Capacidad de recoger las interacciones y movimientos del usuario.
- Pantalla donde se va a proyectar el escenario virtual.
- Un procesador potente (1.5GHz) y una gran cantidad de memoria RAM (2GB) para procesar las interacciones del usuario.

2.4. Ventajas y desventajas generales de la realidad virtual

La adaptación de la realidad virtual a nuestro estilo de vida, ha dado lugar a la incorporación de la misma a ámbitos que anteriormente no usaban prácticamente tecnología alguna, como puede ser la enseñanza, el arte o la historia.

Las ventajas del uso de la realidad virtual en la sociedad se pueden resumir en las siguientes [29]:

- Es segura, ya que el usuario controla el ambiente virtual.
- Útil para personas discapacitadas y para superar fobias.
- Permite poner a prueba hipótesis.
- Incrementa la habilidad de los usuarios a la hora de manipular diversos objetos.
- Ayuda en la simulación de situaciones peligrosas sin poner en riesgo la vida humana.
- Importante a la hora de facilitar el aprendizaje en estudiantes.

En cuanto a desventajas, podemos destacar las siguientes:

- Costes elevados tanto en hardware como en software.
- Mantenimiento especializado en la mayoría de los dispositivos utilizados.
- Sin una adecuada capacitación, el usuario puede tener problemas a la hora de dominar el programa
- Complejidad al desarrollar los programas.
- El tiempo de respuesta a veces puede demorarse varios segundos.
- No garantiza que las personas realicen con éxito las mismas pruebas en situaciones reales.

2.5. Importancia de la realidad virtual para personas con deficiencias cognitivas leves

Actualmente en los países occidentales, debido al envejecimiento de la población, han aumentado el número de personas que padecen deficiencias

cognitivas. Debido a eso, se ha iniciado recientemente la exploración de una nueva vía de soluciones de asistencia cognitiva, basado en los videojuegos.

Esta consiste en la explotación de los videojuegos como una plataforma de software que permite el apoyo como nuevas herramientas de asistencia, menos costosas y más accesibles, que podrían utilizarse, por ejemplo, como una herramienta de reeducación que ayuda a retardar el deterioro de las personas que sufren de Alzheimer.

Varios estudios académicos y juegos, como por ejemplo en Nintendo con Brain Age y Big Brain Academy [21] han surgido. Pero estas iniciativas sufren múltiples limitaciones importantes, ya que en realidad no se adaptan a las necesidades de percepción e interacción de las personas que sufren de DCL. Para hacer frente a estas limitaciones, se está desarrollando el uso de la Realidad Virtual.

La realidad virtual es muy útil para probar la forma de actuar de las personas con DCL en entornos físicos y sociales mediante la simulación de entornos virtuales. Esta característica puede ayudar a entender cómo estos pacientes interactúan con sus entornos, ya sea físicamente con los objetos o socialmente con otras personas de su entorno. En caso de que provoque una mejora en estas personas, tal modificación puede llevarse a cabo fuera del escenario virtual.

2.6. Conclusiones

La realidad virtual es una tecnología en continuo crecimiento, y muy útil en la vida de las personas, incluso en tratamientos de personas con enfermedades. La realidad virtual aplicada en este ámbito ha provocado mejorías y rehabilitaciones completas para estas personas de una manera totalmente novedosa.

Además la realidad virtual ofrece gran variedad de opciones para interactuar con el usuario de forma muy intuitiva y atractiva, mediante presentaciones, juegos, videos, documentación, etc. Por este motivo en muchos tratamientos se han implantado aplicaciones de realidad virtual.

La realidad virtual es, por tanto, una de las tecnologías que más en auge y que, junto con la realidad aumentada, más va a ser utilizada en la vida cotidiana de la sociedad en los próximos años.

Capítulo 3

Arquitectura y tecnologías utilizadas en nuestra aplicación

Este capítulo detalla cuáles son las tecnologías implementadas para el desarrollo de nuestra aplicación, exponiendo además porque son las que más nos conviene para el desarrollo de este proyecto.

3.1. Visual Studio 2015, C-Sharp y Windows Presentation Foundation

Aunque ya hablaremos detalladamente más adelante sobre la Kinect, es este dispositivo sobre el cual gira todo nuestro proyecto, por lo que ha sido un pilar muy importante a la hora de elegir el entorno de programación.

Como entorno de desarrollo elegimos Visual Studio, ya que, admite una gran variedad de lenguajes de programación como C-Sharp, Visual Basic, C++, C, Python, JavaScript, etc. Además se pueden añadir una gran variedad de plugins para admitir más tipos de lenguajes o aumentar la funcionalidad.

Una vez que nos decantamos por Visual Studio teníamos decidir que lenguaje de programación se adaptaba mejor a nuestras necesidades. Aunque sabíamos que Kinect tiene compatibilidad con C, C++, Visual Basic, y algunas versiones de Java y Python, llegamos a la conclusión de que C-Sharp era el lenguaje que menos problemas podía dar en cuanto compatibilidad, porque, al ser ambos productos de Microsoft la compatibilidad entre Kinect y C-Sharp tenía que ser mayor que con otros lenguajes ajenos a Microsoft.

C-Sharp es un lenguaje de programación orientado objetos diseñado por microsoft para ejecutarse en .NET Framework que deriva del C y C++. Usamos la versión de NET framework 4.5, el framework que estaba vigente cuando se publicó el sdk v1.8.

Por otro lado, y a cuanto ventanas gráficas se refiere, decidimos prescindir de Window Form (interfaz clásica de las ventanas para C-Sharp) ya que se invertía mucho tiempo en configurar las propiedades de las ventanas. Es por eso que decidimos utilizar Windows Presentation Foundation (WPF).

WPF es una tecnología creada por Microsoft que permite la creación de interfaces gráfica combinando la tecnología característica de aplicaciones de Windows y de aplicaciones web. Esta herramienta nos permite ahorrar mucho tiempo debido a la sencillez que proporciona a la hora de configurar las ventanas.

3.2. Dispositivo Kinect

Kinect es el dispositivo que nuestro subdirector Jose Miguel nos ha proporcionado, para llevar a cabo con éxito la realización de nuestro proyecto. A continuación os exponemos que es este dispositivo y que hemos necesitado para que funcione correctamente.

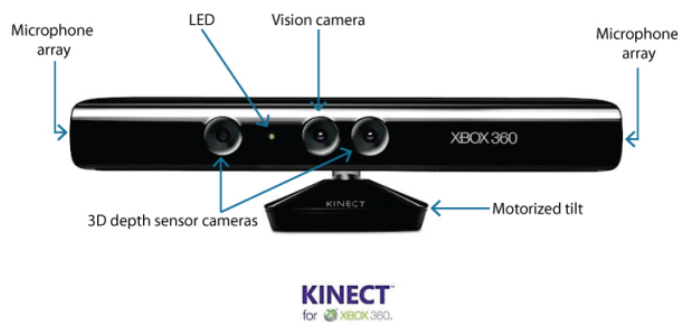


Figura 3.1: Kinect [12]

3.2.1. Definición y características

Kinect v1 es un sensor de movimiento que nació como complemento de Microsoft para la consola de juegos Xbox 360, pero más tarde Microsoft publicó un adaptador y varios sdk que permite usar la kinect en Windows. El dispositivo proporciona una interfaz natural de usuario (NUI) que permite

a los usuarios interactuar de forma intuitiva y sin ningún dispositivo intermedio, tal como un controlador.

El sistema Kinect identifica usuarios individuales a través de reconocimiento facial y reconocimiento de voz. Una cámara de profundidad, lo que "ve en 3-D", crea una imagen icono de un jugador y un sensor de movimiento detecta sus movimientos. El software de reconocimiento de voz permite al sistema entender los comandos hablados y el reconocimiento de gestos corporales permite el seguimiento de los movimientos de los usuarios.

Todos estas funcionalidades nos permite llevar a cabo todas las objetivos propuestos para el desarrollo de nuestra aplicación.

Característica	Sensor Kinect de Xbox 360	Sensor Durango
Campo de Visión	57.5° horizontal x 43.5° vertical	70° horizontal x 60° vertical
Profundidad	0.8 m -> 4.0 m	0.8 m -> 4.0 m
Secuencia de Color	640 x 480 x 24 bpp 4:3 RGB 30cps 640 x 480 x 16 bpp 4:3 YUV 15 cps	1920 x 1080 x 16 bpp 16:9 YUY2 30 cps
Profundidad de Corriente	320 x 240 16 bpp, 13-bit	512 x 424 x 16 bpp, 13-bit
Infrarrojo	No	Rango dinámico de 512 x 424, 11-bit
Registro	Color <-> profundidad	Color <-> profundidad e infrarrojo
Captura de Audio	4-micrófonos proveen 48 Hz de audio	4-micrófonos proveen 48 KHz de audio
Data Path	USB 2.0	USB 3.0
Latencia	~90 ms con procesamiento	~60 ms con procesamiento
Motor de Inclinación	Solamente vertical	No cuenta con motor de inclinación

Tabla 3.1: Características de Kinect [4]

3.2.2. Primer contacto con SDK v1.8

Un SDK (Software Development Kit), nos permite desarrollar aplicaciones y ejecutar un emulador. En nuestro caso en un principio utilizamos el SKD v1.8.

La v1.8 es el último SDK disponible para Kinect v1, este SDK se puede usar para el manejo de todos los componentes del sensor Kinect y también permite detectar el esqueleto de una persona con la librería Skeleton. Esto funciona muy bien para determinadas aplicaciones que solo sirvan para detectar gestos o para usar Kinect como cámara, pero a la hora de hacer una aplicación en la que el usuario se tenga que moverse por menús o pulsando botones se vuelve pesado y difícil.

3.2.3. Uso y final implementación del SDK Developer's Toolkit

Aunque las primeras pruebas con Kinect y la primera versión se hizo usando solo SDK v1.8, los problemas que nos surgieron comentados en el anterior apartado, nos obligó a buscar otras soluciones. Es por eso que elegimos SDK Developer's Toolkit.

En este SDK se introdujeron muchos controles nuevos para interactuar con Kinect como el KinectRegion que crea una región donde se mueve una mano, que el usuario usa como ratón o el KinectButton que introduce la mecánica de push o el KinectScrollViewer que permite hacer grip y un manejo más sencillo del sensor de kinect.

La integración de este sdk permite que la aplicación vaya más fluida y facilita mucho la creación de menús y la interacción entre el usuario y la aplicación.

3.3. Base de Datos

Uno de nuestros objetivos a la hora de realizar el proyecto, era registrar el tiempo de respuesta, aciertos y errores de los usuarios que utilizasen la aplicación. Para ello utilizamos dos grandes herramientas que nos permiten crear y albergar una base de datos: Azure y SQL Server.

3.3.1. Azure

Azure es la cloud de microsoft con una gran cantidad de servicios puede alojar bases de datos, servidores, máquinas virtuales, etc.

Cuando empezamos a desarrollar la aplicación pensamos que era buena idea alojar la base de datos en la nube y barajamos varias posibilidades como Pydio, AppHarvor o Amazon Web Services, pero al final nos decantamos por usar Azure por ser mucho más cómodo de utilizar, debido a que está completamente integrado en Visual Studio y se puede gestionar desde él.

También influyó mucho la suscripción gratuita que regala Microsoft por ser de la UCM, para decidir la nube donde alojar la base de datos. Gracias a ello nos permitía alojarla sin ningún coste.

La base de datos está configurada para que esté encriptada, tenga un timeout de 60 segundos y que la aplicación pueda tanto leer como escribir en ella. El diseño de la base de datos alojada en Azure se realizó en SQL Server.

3.3.2. SQL Server

SQL Server, desarrollado por Microsoft, es un sistema que sirve para el manejo de bases de datos relacionales.

Se utilizo SQL Server por su compatibilidad con C-Sharp y porqué Visual Studio tiene integrado el SQL Server explorer, que permite configurar las bases de datos de SQL Server sin tener que usar programas adicionales.

Con SQL Server se hizo una base de datos local que se establece al instalar la aplicación en un equipo. Esta base de datos sirve como backup de la base de datos de Azure por si falla la conexión con Azure, de este modo no se perderá la sesión que tiene el usuario en ese momento.

Posteriormente la base de datos de SQL Server se sincroniza con Azure. Esta funcionalidad todavía no está implementada debido a que es necesario adquirir la versión de pago.

3.4. Aforge: Grabación de acciones de usuario

En el diseño inicial del proyecto teníamos en mente que necesitábamos grabar de alguna forma interactuaba el usuario con la aplicación, para poder reproducirlo posteriormente y ver como se había desenvuelto el usuario.

La grabación también era necesaria para poder llevar un seguimiento y el progreso que iba llevando el usuario

Se hicieron varias pruebas con librerías, como Expression. Encoder de Microsoft la cual tuvimos que desear porque la versión gratuita no permite la grabación, además llevaba sin soporte desde 2010 y ya no se podía comprar la versión de pago.

Entonces decidimos usar Aforge.Net, una libreria de codigo abierto para C-Sharp que sirve para desarrollar aplicaciones en los campos de visión artificial, procesamiento de imágenes, redes neuronales, etc. Concretamente se uso Aforge.Video que permite grabar la pantalla en varios formatos, controlar el flujo de la grabación, seleccionar el rango que se quiere grabar, etc.

3.5. Necesidad de realizar un control de versiones

Nunca habíamos usado un programa para controlar las versiones y al principio pensamos que no lo íbamos a necesitar, pero a medida que el proyecto iba creciendo nos dimos cuenta que debíamos mantener un control de

la versiones que íbamos realizando.

Ya que si el proyecto tomaba un rumbo equivocado podíamos recuperar rápidamente la versión anterior y poder llevar control de los cambios que íbamos haciendo.

Nos facilitó mucho el trabajo un plugin llamado Team Explorer para Visual Studio, el cual permitía usar Git (software de control de versiones) desde Visual Studio, actualmente con Team Explorer en Visual Studio 2015 se puede elegir que programa de control de versiones quieres utilizar. [10] [26]

Capítulo 4

La Aplicación

En este capítulo se explicará detalladamente el contexto, la funcionalidad, la implementación y las vistas principales de nuestra aplicación.

4.1. Contexto

El objetivo de la aplicación es desarrollar una herramienta que sea útil tanto para los médicos, ayudándoles a hacer el seguimiento de sus pacientes, como a los mismos pacientes reforzando su memoria.

Para realizar el seguimiento se han implementado juegos serios que sirven para que el usuario se entretenga jugando a los juegos mientras se recopila información y se evalúa algún aspecto médico o deficiencia del usuario.

Las sesiones de juego del usuario se guardarán en una base de datos a la que posteriormente el médico tendrá acceso pudiendo evaluar la sesión del usuario viendo los errores cometidos.

Los juegos se han implementado usando Kinect por ello su principal característica es que para jugar el usuario tendrá que mover el cuerpo, produciendo así que sea más atractivo para el usuario.

4.2. Funcionalidades

En este apartado explicaremos las funcionalidades principales que tiene la aplicación, explicando con detalle las acciones que pueden realizar los

usuarios. Para ejecutarlas el usuario puede efectuar dos gestos dependiendo de la acción que quiera realizar. Los gestos son:

- **Push:** Es el gesto de empujar con la mano.
- **Grip:** Es el gesto de cerrar la mano.



Figura 4.1: Gestos Push y Grip

4.2.1. Inicio y Registro

El usuario inicia sesión metiendo su identificador de usuario y su contraseña en los campos correspondientes. Los usuarios que no estén en la base de datos no podrán iniciar sesión. Para poder entrar en la aplicación necesitarán registrarse introduciendo, su usuario, nombre, apellidos y contraseñas y quedarán registrados en la base de datos, para posteriormente inicie sesión el usuario.

4.2.2. Navegar por los menús

Para navegar por los menús el usuario tiene que mover la mano que está detectando Kinect, hasta la opción que quiera elegir y hacer la acción de *push*. Según va realizando la acción la mano se irá poniendo morada, hasta que se termine de pulsar el botón, que se pondrá de color azul.

4.2.3. Obtener ayuda

Una vez en la ventana de ayuda al usuario se le mostrara un vídeo de como jugar al juego seleccionado pudiendo pausar, reiniciar o reanudar el vídeo de ayuda.

4.2.4. Realización del puzzle

Una vez el usuario esté listo para empezar pulsará el botón Play, las piezas se recolocarán aleatoriamente, el cronómetro y la grabación de la sesión

se ejecutan y continuarán hasta que el usuario consiga completar el puzzle o se dé por vencido.

En el puzzle el usuario puede seleccionar dos piezas con el movimiento *Push* e intercambiarlas hasta que todas las piezas estén en la posición correcta. Una vez terminado el puzzle aparecerá una ventana que le notificara al usuario que ha terminado.

Al completar el juego la aplicación guardará en la base de datos la grabación de los movimientos del usuario, el tiempo transcurrido y la fecha y hora para llevar un seguimiento de los usuarios. La figura 4.3.5 muestra un ejemplo de cómo se hace el puzzle.

4.2.5. Realización de compra

Después de seleccionar aceptar en la lista de la compra, al usuario le aparecerá una ventana en la que deberá hacer el gesto de *Grip*, para moverse entre las estanterías y encontrar el producto deseado.

Una vez encontrado el usuario tendrá que hacer el movimiento de *Push* para seleccionar el producto (aparecerá un recuadro amarillo sobre el producto seleccionado) y seleccionar mediante el movimiento de *Push* el hueco de la cesta donde quiere poner el objeto.

Una vez completada la lista de la compra, es decir, introducido todos los productos de la lista en la cesta de la compra (los huecos libre en la parte de abajo de la ventana) el juego se acabará y la aplicación insertará en la base de datos el tiempo que ha tardado el usuario, una grabación de como ha completado el usuario el juego, la fecha y hora para llevar un seguimiento de los usuarios.

4.2.6. Grabar sesión

Al comenzar cada juego se iniciará una grabación de la partida del usuario, quedando registrado en vídeo la sesión.

Terminado el juego la grabación se guardará en la base de datos junto con el tamaño del vídeo. Mientras se va realizando la grabación el vídeo se crea de forma local en la carpeta de la aplicación, cuando se termina de guardar en la base de datos se borra. Si el puzzle o la compra no se llega a terminar el vídeo se borra.

```

    /// <summary>
    /// Create the event of the record and start the record
    /// </summary>
    private void StartRecord() //Object stateInfo
    {
        // create screen capture video source
        this._streamVideo = new ScreenCaptureStream(this._screenArea);

        // set NewFrame event handler
        this._streamVideo.NewFrame += new NewFrameEventHandler(this.video_NewFr

        // start the video source
        this._streamVideo.Start();

        // _stopWatch
        this._stopWatch.Start();
    } // void StartRecord()

    /// <summary>
    /// Event record
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="eventArgs"></param>
    private void video_NewFrame(object sender, NewFrameEventArgs eventArgs)
    {
        if (this._isRecording && !this.pause)
        {
            this._frameCount++;
            this._writer.WriteVideoFrame(eventArgs.Frame);
        }
        else if (!this.pause)
        {
            RecorderScreen.nameVideoAnt = RecorderScreen.nameVideo;
            _stopWatch.Reset();
            //Thread.Sleep(500);
            _streamVideo.SignalToStop();
            //Thread.Sleep(500);
            _writer.Close();
        }
    } // void video_NewFrame(object, NewFrameEventArgs)

```

Figura 4.2: Código para grabar sesión

4.3. Diseño e implementación

4.3.1. Patrón de diseño MVVM

Para el diseño de la aplicación hemos utilizado el patrón Model-View-ViewModel (MVVM), es un patrón que permite sacar un mayor rendimiento a WPF.

MVVM permite Obtener una Completa Separación entre las 3 partes (View, el Viewmodel y el Model). Consta de 3 partes:

- **Model:** El modelo establece la lógica del negocio y los Datos. Provee eventos y notificaciones mediante *INotifyPropertyChanged* y *INotifyCollectionChanged*.
- **View:** Corresponde con la vista o interfaz de la aplicación.
- **Viewmodel:** Implementa propiedades y comandos sustituyendo al controlador del MVC. Media entre la vista, los datos y la lógica encapsulándolos.
- **Binding:** Se incluye en la vista y sirve para unir los controles de la View con los datos de ViewModel.

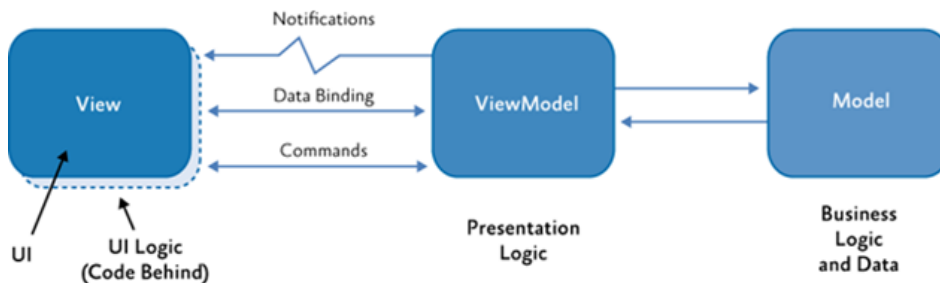


Figura 4.3: Representación del patrón MVVM

4.3.2. Diseño base de datos

El diseño de la base de datos es un diseño relacional (véase figura 4.3) con tres tablas principales: usuario, puzzle y supermarket.

De las tabla de puzzle se heredan tres tablas clasificadas por dificultad y de la tabla de supermarket heredan otras tres tablas que van en función del número de productos de la lista de la compra. La relación entre las tablas

supermarket y usuario es de varios a 1 que es la misma relación que tiene la tabla puzzle y usuario.

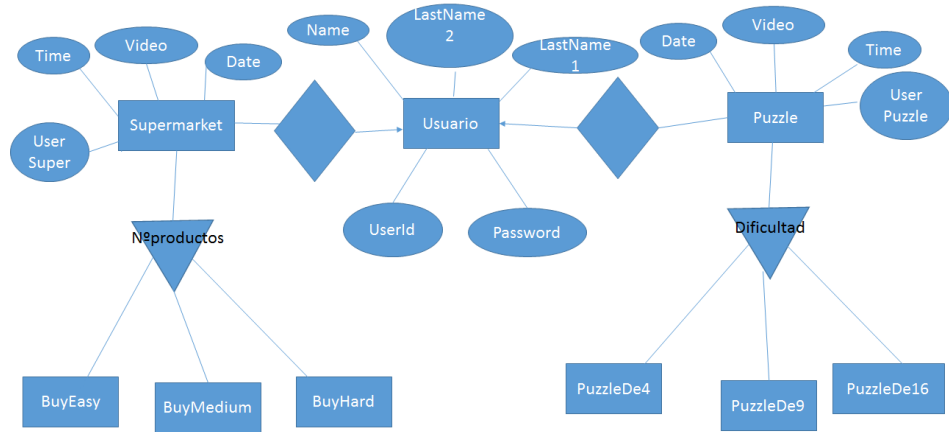


Figura 4.4: Diagrama de la base de datos

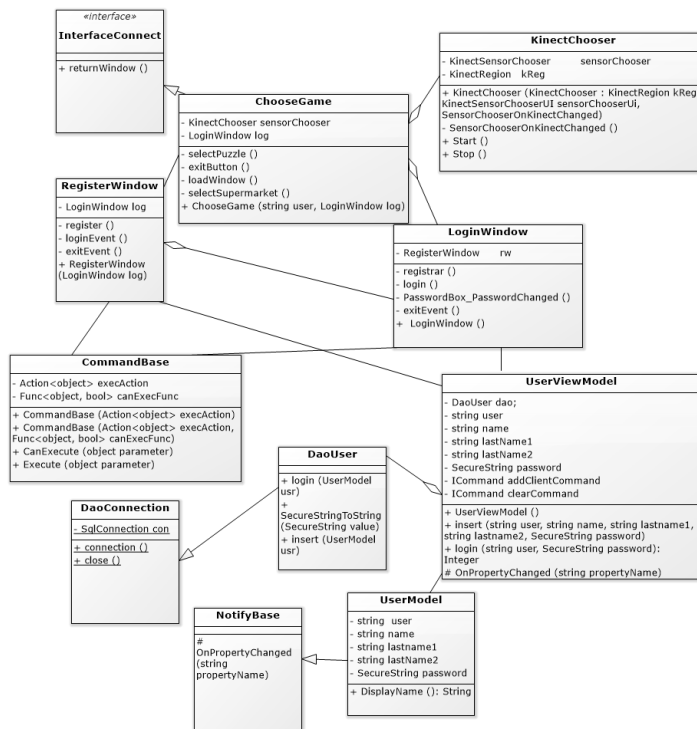


Figura 4.5: Diagrama de clase de inicio y registro

Un ejemplo donde se utiliza MVVM es el proyecto inicio (véase figura 4.4).

Las vistas de *RegisterWindow* y *LoginWindow* comunican con *UserViewModel* que es el *ViewModel* cuando pulsamos en inicio en el caso de *LoginWindow* y Registrar en el caso de *RegisterWindow*.

Una vez la ejecución está en el *Viewmodel* se crea un comando (*CommandBase*) que crea un nuevo usuario de tipo *UserModel*.

Si la ejecución viene de *LoginWindow* se consultará con la base de datos si el usuario está registrado, después de la consulta *NotifyBase* le notificará los cambios a *LoginWindow*. Si por el contrario la ejecución viene de *RegisterWindow*, se registrará el usuario en la base de datos y posteriormente *NotifyBase* le notificará los cambios a *RegisterWindow*.

4.3.3. Usuario

La aplicación tiene un único tipo usuario que puede hacer todas las acciones descritas (véase figura 4.5). El usuario puede iniciar sesión y registrarse en la aplicación, manejar Kinect, jugar a los dos juegos descritos anteriormente y podrá desplazarse por los menús cuando la Kinect.

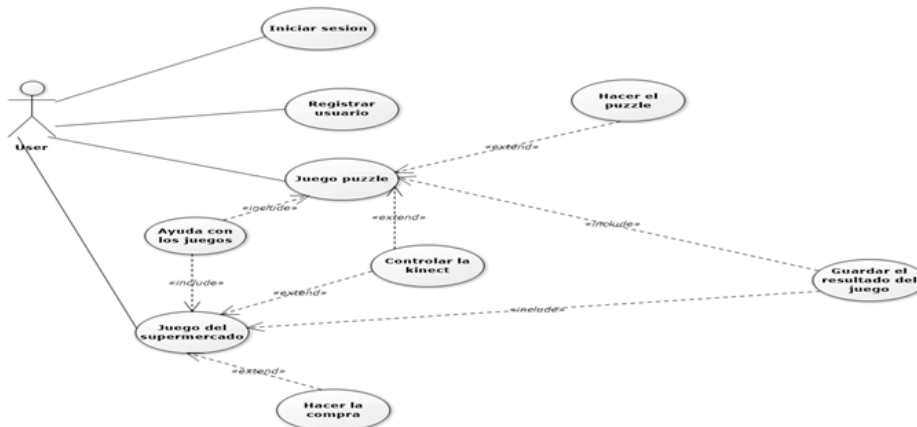


Figura 4.6: Caso de uso del usuario

4.4. Vistas Principales

A continuación se mostrarán y explicarán las vistas más importantes del proyecto

4.4.1. Inicio y registro

En la ventana inicio (véase figura 6), el usuario introduce su identificador y su contraseña (Campo especial de WPF para escribir contraseña de forma segura). Después de elegir la opción inicio, su solicitud se procesa y la aplicación conecta con la base de datos (local o Azure).

Por otro lado en la ventana de registro, el usuario puede rellenar todos los campos (usuario, contraseña, nombre y apellidos) y dar al botón de registrarse, y quedará registrado en la base de datos (Azure o local).

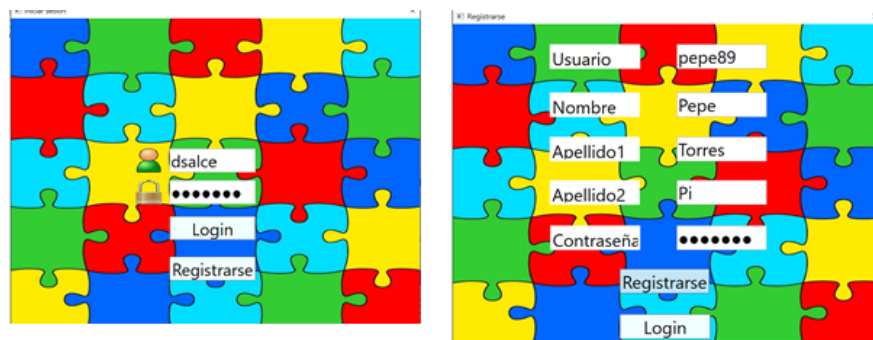


Figura 4.7: Ventanas de Inicio y registro

4.4.2. Elegir juego

Muestra al usuario la posibilidad de elegir el juego (puzzle o supermercado) o volver a la ventana de inicio dando al icono cerrar.



Figura 4.8: Ventana de selección de juego

4.4.3. Puzzle

La vista de selección de puzzle consta de tres dificultades(fácil, medio y difícil). El usuario puede también seleccionar el botón de ayuda y le mostrará la ventana que contiene un video explicativo de como jugar con el puzzle.

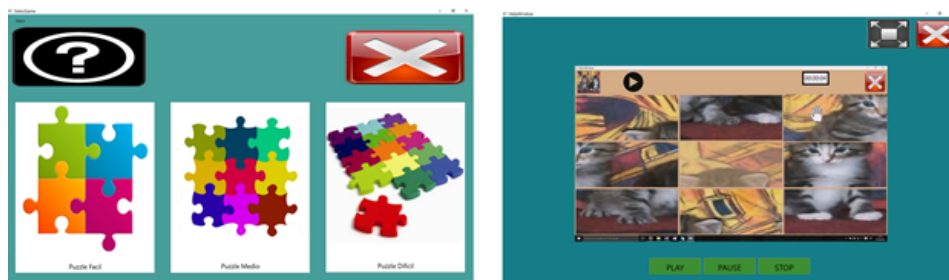


Figura 4.9: Ventana de selección de dificultad y ventana de ayuda (puzzle)

4.4.4. Ventana juego puzzle

La ventana de puzzle tiene un *usercontrol* con un cronómetro para saber el tiempo que está tardando el usuario, un botón de play que inicia el puzzle y un botón que cuando cliques muestra la imagen del puzzle que sirve de ayuda para completar el puzzle.

Para interactuar con las piezas del puzzle el usuario tiene que hacer el movimiento de *push*. Si el usuario pincha en el icono cerrar volverá al menú de selección de dificultad de puzzle.

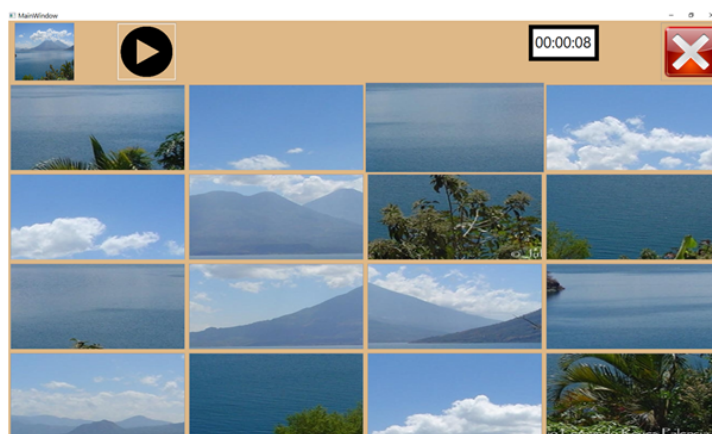


Figura 4.10: Ventana del juego puzzle

4.4.5. Supermercado

La vista de selección de lista de la compra contiene un juego de memoria que consiste en enseñar un número de productos en función de la dificultad (fácil, medio y difícil) escogida por el usuario.

También contiene una ventana de ayuda como en el proyecto de puzzle que sirve para enseñar al usuario como se juega con un vídeo explicativo.



Figura 4.11: Ventana de selección de dificultad (supermercado)

4.4.6. Ventana juego supermercado y lista de la compra

Ventana en la que se muestra el número de productos que debe memorizar el usuario. Cuanto más difícil se seleccione el modo de juego más productos se añadirán a la lista de la compra que se muestra una vez seleccionado el modo de juego, pudiendo estar todo el tiempo que necesite el usuario para memorizarla.

Si el usuario pincha en aceptar pasará a la venta de supermercado y si da en cancelar volverá al menú anterior.

Una vez el usuario ha aceptado, aparecerá la ventana de supermercado representada que contiene un *usercontrol* con un cronómetro para saber cuánto tiempo está tardando el usuario.

Las estanterías están sobre un *KinectScrollViewer* y un *Stackpanel* que permiten hacer *grip* sobre ellas para moverse entre estanterías.

Igual que las estanterías la cesta de la compra contiene un *KinectScroll-*

Viewer y un *Stackpanel* que permiten hacer *grip* sobre la cesta de productos desplazándose el usuario por la cesta, pudiendo buscar cualquier producto de la cesta.

Los productos de la cesta de la compra se seleccionan mediante el gesto de *push*. Si el usuario selecciona el icono cerrar, volverá a la ventana del menú de selección de dificultad.

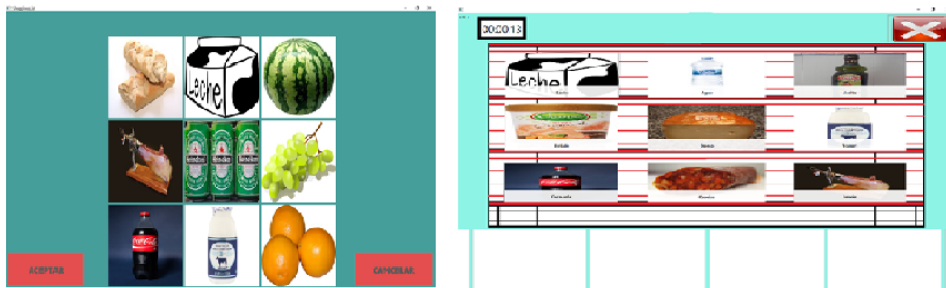


Figura 4.12: Ventanas de lista de la compra y del juego supermercado

4.4.7. Ventana base de datos de los usuarios

Cuando el usuario finaliza alguno de los dos juegos, queda registrado en una base de datos el tiempo que ha tardado en terminar el juego y el vídeo con la grabación de los movimientos ejecutados durante el proceso.

dbo.PuzzleDe9 [Data]					dbo.BuyEasy [Data]				
date	userPuz	time	video	date	userSuper	time	video		
04/06/2016 23:07:24	angi	00:00:14	0x0000001C667...	04/06/2016 23:08:32	angi	00:00:12	0x0000001C667...		
04/06/2016 23:05:14	ch	00:00:24	0x0000001C667...	04/06/2016 23:06:06	ch	00:00:19	0x0000001C667...		
28/03/2016 20:21:23	dsalce	00:00:09	0x0000001C667...	28/03/2016 20:22:34	dsalce	00:00:10	0x0000001C667...		
04/06/2016 23:03:31	dsalce	00:00:16	0x0000001C667...	28/03/2016 20:23:33	dsalce	00:00:05	0x0000001C667...		
				04/06/2016 23:10:45	hsg00	00:00:12	0x0000001C667...		

Figura 4.13: Base de datos del juego puzzle y supermercado

4.5. Problemas detectados

En este apartado expondremos los principales problemas que nos han surgido en el desarrollo del proyecto.

4.5.1. Biometría facial

Se intentó que el registro y inicio de personas fuera mediante la biometría facial, pero nos encontramos bastantes problemas para intentar llevar esta idea a cabo.

Kinect no tiene una gran resolución, lo que dificulta que las imágenes de la cara del usuario fueran lo suficientemente nítidas para que se pudiera reconocer, y registrar una persona requería tomar varias imágenes de la cara para que la reconociera y guardarlas en la base de datos y aun así era muy probable que no la reconociera.

Encontramos trabajos que hablaban sobre el tema pero no era seguro que tuviéramos éxito en desarrollarlo y nos iba llevar demasiado tiempo y esto podía provocar que no pudiésemos desarrollar otras partes del proyecto.

4.5.2. Dedos en la kinect

Kinect no incorpora detección de dedos, por lo tanto, no detecta el movimiento de coger un objeto. Realiza algo similar si se usa *grip*, pero solo se puede usar en *KinectScrollViewer* que solo sirve para desplazarte por un scroll.

Para solucionar este problema se intentó usar Opencv o Emgu que es como se llama la librería en C-Sharp con un mal resultado ya que no funcionó correctamente.

Este problema lo solucionamos usando el gesto de *push* para seleccionar una pieza y después seleccionar la pieza con la que se va intercambiar.

4.5.3. Azure

En algunas redes no se permitía conectar con Azure como en la UCM o en Eduroam, la IP estaba en el rango de direcciones especificado en el firewall de Azure y el puerto era el común por el que se conecta en SQL Server 1433 TCP.

Para solucionar este problema se decidió que la aplicación también pudiera trabajar en local y que posteriormente las bases de datos tanto local como la de Azure se deberían sincronizar para mantener una información coherente entre ellas.

Capítulo 5

Evaluación con usuarios

Con la aplicación casi terminada y una vez corregidos todos los errores que hemos detectado a lo largo del desarrollo de la aplicación, vamos a realizar las evaluaciones de la aplicación con usuarios, para poder detectar otros posibles problemas y ver si la aplicación cumple las expectativas puestas en ella.

A continuación, presentamos el plan de evaluación, detallando cómo será la evaluación con usuarios y las tareas que deben realizar. También definiremos nuestro trabajo como moderadores y observadores de la sesión de evaluación.

Con las evaluaciones obtenidas de las pruebas realizadas a los usuarios, analizaremos los resultados y pensaremos la mejor forma de aplicar el feedback recibido por los usuarios.

5.1. Plan de evaluación

En este apartado vamos a definir el plan de proyecto con todos los aspectos que queremos evaluar de la aplicación.

¿Qué estamos evaluando?

- **Proyecto:** Aplicación Interactiva para pacientes con Deficiencias Cognitivas Leves.
- **Premisa:** Personas mayores que necesiten ejercitar la memoria o que

quieren tener un seguimiento de la degradación que van sufriendo en la memoria.

- **Objetivo de la aplicación:** conseguir realizar los puzzles o la compra en el supermercado en el menor tiempo posible.

5.1.1. Propósito de la evaluación

Queremos comprobar, por un lado, si el diseño es claro para el usuario, es decir, saber navegar por la aplicación, si le resulta costoso usarla, el sistema implementado usando Kinect es adecuado, etc. Y por otro lado si cumple su principal cometido que es evaluar la memoria del usuario.

5.1.1.1. Objetivos generales

Se realizará una prueba para evaluar los principales aspectos de la aplicación, tanto a nivel de interfaz como a nivel de contenido. A continuación indicamos los tres objetivos mas importantes a la hora de realizar la evaluación:

- Comprobar la eficacia de la interfaz, los usuarios deben saber navegar por la aplicación sin perderse.
- Evaluar si los usuarios son capaces de terminar los puzzles y las listas de la compra.
- Evaluar la dificultad que tiene el usuario para adaptarse a Kinect y su particularidad a la hora de interactuar con las ventanas.

5.1.1.2. Preguntas de investigación

En este apartado, mostramos las diferentes preguntas que realizamos con el fin de conocer mejor la relación de los usuarios con nuestra aplicación.

- ¿Qué te parece el uso de kinect para jugar a los juegos?.
- ¿Te parece bien como están colocados los botones de la aplicación o crees que estarían mejor colocados en otro lugar para acceder mejor con Kinect?.
- ¿Hay componentes que no sabes para qué sirven? Si es así indíquenoslo.
- ¿Es capaz de finalizar los puzzles sin ayuda?
- ¿Es capaz de finalizar la lista de la compra sin ayuda?
- ¿Es capaz de hacerse con el control de Kinect o le resulta muy complicado su control?

- ¿Es capaz de encontrar lo que desea de forma fácil?
- ¿Es útil la ventana de ayuda o habría que introducir una mejor explicación?
- ¿La formación recibida antes de empezar las tareas es suficiente?
- ¿Hay alguna funcionalidad que te gustaría que estuviera incluida en la aplicación y que no está ¿Cual?

Requisitos para los participantes

5.1.2. Requisitos para los participantes

Habrà varios tipos de participantes:

- personas menores de 30.
- personas entre la edad de 30 a 60 años.
- personas mayores de 60

Para evaluar nuestra aplicación necesitamos personas con los rangos de edad mencionados previamente, aunque nuestra aplicación va dirigida principalmente a ancianos con problemas cognitivos leves, pensamos que obtener feedback de gente más joven nos puede ayudar también a mejorar la aplicación.

5.1.3. Diseño experimental

Se realizará hasta un total de 5 evaluaciones más o menos por cada tipo de usuario, con una duración aproximada de 45 minutos, aunque puede ser más tiempo si el usuario no entiende las explicaciones. El plan a realizar con el usuario es:

1. Explicarle en qué consiste la aplicación.
2. Explicar al usuario los juegos.
3. Hacer pruebas para que el usuario coja un poco de soltura con Kinect.
4. Dar la lista de tareas al usuario y pedirle que las realice. El usuario debe intentar realizarla sin ayuda. Mientras tanto, el observador o moderador irá tomando nota de cómo el usuario realiza las tareas y nos irá describiendo lo que va haciendo.
5. El usuario comenta con el moderador los aspectos de la aplicación que le han gustado y los que no, como le ha resultado la experiencia y se le harán las preguntas de investigación.
6. Recopilación de datos y conclusión de los datos obtenidos.

5.1.4. Tareas a realizar

En este apartado vamos a definir las tareas que tiene que realizar el usuario para realizar el plan de evaluación.

1. Registrarse o iniciar sesión en el caso de que el usuario ya esté registrado.
2. Ir al menú de selección de puzzle.
3. Seleccionar la ventana de ayuda.
4. Visualizar el vídeo.
5. Salir de la ventana de ayuda.
6. Seleccionar el puzzle de dificultad media o difícil.
7. Terminar el puzzle.
8. Ir al menú de selección de lista de la compra.
9. Seleccionar la lista de dificultad fácil.
10. Terminar la compra.
11. Salir de la aplicación.

5.1.5. Entorno y herramientas empleadas

Las evaluaciones se llevarán a cabo en un lugar tranquilo para evitar las posibles distracciones del usuario.

Será necesario un equipo con la configuración que necesita la aplicación para funcionar (librerías y SDKS) , una Kinect, una televisión grande o un proyector, una sala provista de mesa, sillas, la lista de las tareas que tiene que realizar el usuario, papel y bolígrafo para tomar anotaciones.

5.1.6. Obtención de feedback de los participantes

Contemplamos varias fuentes de datos:

1. Se realizará una encuesta inicial para conocer el contexto del usuario antes de realizar la evaluación (nivel de conocimiento tecnológico, si resuelven rompecabezas habitualmente...).
2. En la medida de lo posible se intentará grabar la interacción del usuario con la aplicación.

3. Haremos un repaso de las tareas al terminar y después los usuarios recibirán un cuestionario en el que evaluarán las tareas realizadas del 1-5, siendo 1 el nivel mas fácil y 5 el más difícil.

5.1.7. Tareas del moderador

El moderador impartirá una pequeña formación de cómo usar Kinect y de qué trata la aplicación. También será el encargado de supervisar las pruebas Kinect que se harán previamente al reparto de tareas para que el usuario sepa como funciona.

El moderador se hará cargo de pedir al usuario que realice las tareas especificadas, y observará y anotará todo aquello que considere relevante durante la realización de la evaluación y ayudará al usuario si éste lo necesita.

Al final el moderador comentará las tareas con el usuario y le entregará el cuestionario de evaluación de tareas

5.1.8. Descripción de la metodología de análisis de datos

Una vez terminadas todas las evaluaciones, revisaremos las anotaciones tomadas por el moderador, los vídeos grabados, las anotaciones, los cuestionarios y las conversaciones realizadas con los usuarios.

Con todos los datos obtenidos realizaremos un informe con los errores encontrados, bugs, problemas detectados en la interfaz y cuales son las tareas que han resultado más tediosas o difíciles a los usuarios.

Por último propondremos soluciones para los errores encontrados anteriormente y revisaremos las tareas que no hayan obtenido buena nota en los cuestionarios.

5.2. Evaluación

La evaluación con los usuarios tuvieron lugar desde el 1 al 15 de mayo. Hemos intentado que el número de personas que evalúan la aplicación, fuera equitativo para tener datos suficientes de cada tipo de usuario.

5.2.1. Informe de resultados y soluciones

En este apartado vamos a exponer los fallos y errores detectados tanto por el moderador como los expuestos por el usuario y vamos a detallar cómo los hemos solucionado.

Problema 1: En algunos botones hemos detectado que el usuario no es capaz de pulsarlos con la Kinect. **Solución:** Ver si los botones pueden hacerse más grandes o cambiarse de sitio para que usuario pueda pulsarlos mejor.

Problema 2: Los botones de los bordes les costaba mucho a los usuarios acertar en ellos haciendo que se frustrasen y cometieran errores. **Solución:** Cambiamos de sitio los botones que al efectuar movimientos no interrumpen el correcto funcionamiento de la aplicación, es decir, solo se mueven los botones que no afectan a la interacción del usuario con otros elementos.

Problema 3: Hemos detectado que los usuarios no aprenden lo suficiente del vídeo de ayuda para jugar a los juegos. **Solución:** Modificar el vídeo y buscar otra forma otra forma complementaria para enseñar a los usuario como jugar.

Problema 4: Los tiempos de carga para algunas ventanas son demasiado largos y el usuario piensa que la aplicación se ha quedado colgada. **Solución:** Bajar el tiempo de carga de las ventanas si se puede. En caso de que no se pueda, añadimos ventanas con el mensaje de cargando.

Problema 5: Los usuario nos han comentado que el cronómetro se ve muy pequeño y que estaría bien si pudiésemos hacerlo más grande. **Solución:** Se agranda el UserControl del cronómetro.

Problema 6: Cuando el usuario reiniciaba el puzzle, el juego no era capaz de detectar si el puzzle estaba terminado. **Solución:** Una vez que el usuario elige empezar se bloquea el botón y si quiere reiniciar, tendrá que salir al menú de selección de dificultad y volver a seleccionar el puzzle.

Problema 7: Algunas veces la base de datos no guarda el vídeo del usuario por un error. **Solución:** Se corrige el error dejando más tiempo al thread que guarda el vídeo.

5.2.2. Conclusiones más relevantes

Casi todas las personas que han probado la aplicación han quedado satisfechas señalando que este tipo de juegos con Kinect tiene mucho potencial y les parece una gran idea usar juegos serios para ayudar a las personas y al médico para poder llevar un seguimiento.

Nos han comentado mucho usuarios, si se podría hacer aplicaciones con Kinect que sirviesen de rehabilitación o para ejercitarse, ya que, cuando están jugando no les limita el dolor que tienen por que no están tan pendientes de

él, pudiendo esforzarse más en los ejercicios.

También nos ha sorprendido mucho la rapidez con la que se adaptan a Kinect y cogen la soltura para realizar los movimientos sin equivocarse.

Capítulo 6

Conclusiones y trabajo futuro

6.1. Conclusiones

El proyecto nos ha ayudado a comprender el proceso de desarrollo que debe de llevar un proyecto grande y nos hemos visto obligados a realizar una investigación de las tecnologías mirando las restricciones de las mismas y pensando cual nos iban a permitir cumplir con los objetivos de la aplicación.

A parte de la investigación que hemos realizado, también hemos aprendido a usar nuevas tecnologías aplicando lo aprendido en la carrera y en las investigaciones para desarrollar en proyecto.

Hemos podido comprobar durante todo el proyecto que los usos que se le puede dar a la realidad virtual y Kinect son innumerables. Las disciplinas donde se puede aplicar son muchas, ya sea en un ambiente de entretenimiento (juegos lúdicos o juegos educativos), o en un ambiente clínico (aplicaciones para tratar las deficiencias cognitivas leves, simular un escenario médico, en rehabilitación para que sea más entretenido para los pacientes ejercitarse, etc).

En nuestro caso hemos llegado a la conclusión de que el uso de la realidad virtual y Kinect para detectar deficiencias cognitivas leves tiene muchas ventajas. Permite a los médicos realizar pruebas al paciente de una manera mas entretenida para ellos, ya que, al estar interactuando con la aplicación están entretenidos provocandoles una una mayor atención en las pruebas. Además aparte de ayudar a ejercitar la memoria también les ayuda a ejercitar algunos músculos.

En cuanto los objetivos principales marcados en un principio creemos que los hemos cumplido, si bien es cierto hay algunos objetivos secundarios que no hemos sido capaces de cumplirlos debido a las restricciones de Kinect en algunos casos y en otros por el tiempo y esfuerzo que había que invertir para llevarlos a cabo, estos últimos podían producir que no entregáramos el proyecto en el tiempo establecido.

En relación a la planificación nos hemos dado cuenta de que la planificación no ha sido realista, hemos ido retrasados respecto a la planificación inicial. Con esta premisa nos hemos dado cuenta de que planificar un proyecto es una tarea compleja y ardua. Debemos pararnos a pensar con detenimiento cuanto tiempo real nos va llevar realizar los objetivos.

Finalmente concluir que realizar este proyecto ha sido una experiencia muy enriquecedora y gratificante, tanto a nivel académico como personal. Hemos aprendido mucho durante los meses que hemos trabajado en él, conociendo nuevas tecnologías, desarrollando con lenguajes de programación diferentes a los aprendidos en nuestra carrera y superando los retos que nos iban apareciendo en el desarrollo de la aplicación.

6.2. Conclusions

During this project, we learn the different stages of a project development, from conducting a research of existing technologies, to concluding which is the best option to meet our objectives.

Additionally, we have also learned the value of the knowledge, acquired during the years we were university students, when we had to use new technologies.

The uses of the virtual reality and the Kinect are potentially innumerable. For instance, they can have uses from entertainment or educational games to clinical applications such for treating mild cognitive impairments or simulating medical case studies.

In our project, we can conclude that the interaction of patients with the virtual reality and Kinect has many advantages when using it to detect mild cognitive impairment. It is because an entertained patients put more attention during the test. Moreover, apart from helping them exercise memory, patients exercise some muscles.

The main objectives had been successfully met, although some secondary

objectives had not been completed due to Kinect restrictions and the deadline of the project.

During the project, some tasks have been delayed from the original planning, because their complexity. We learn that planning a project is a difficult task because it is not easy to know details of work never done before.

Our final conclusion consists of the whole project has been a very academically enrich experience, which we have enjoyed. We have learned how to use new technologies, program with new languages, and overcome challenges during the development of the project.

6.3. Líneas de trabajo futuras

En este apartado vamos a comentar ideas nuevas que se podrían incluir en un futuro y los cambios que se podrían hacer para hacer una aplicación más completa y atractiva para el usuario.

6.3.1. Actualización de Kinect

Sustituir Kinect v1 por la v2 y actualizar las librerías. La resolución con la que captura Kinect v2 sería muy superior (1920x1080 30 fps 16:9) y nos permitirá mejorar la detección de los movimientos del usuario y implementar nuevos movimientos. Esto nos posibilita implementar juegos más complejos con una interacción más fluida entre usuario y aplicación.

6.3.2. Biometría

Con una mayor resolución Kinect v2, nos permitirá hacer un inicio de sesión y registro por biometría haciendo más cómoda y simple la acción de iniciar sesión a los usuarios.

6.3.3. Nuevos juegos e implementar nuevos gestos

Extender la funcionalidad de la aplicación implementando nuevos juegos, cubriendo más aspectos de las deficiencias cognitivas leves.

Se añadirían juegos, por ejemplo, un juego de cartas que consiste en buscar la pareja de la carta que has levantado previamente, este juego permitirá medir la memoria a corto plazo de la persona.

También se incluirían las nuevas funcionalidad que nos ofrecería Kinect v2 en aplicación para desarrollar los juegos nuevos, como implementar nuevos gestos para hacer más intuitivos los juegos.

6.3.4. Mejorar la aplicación

Mejorar la interfaz gráfica de la aplicación para hacerla más atractiva al usuario e intentar simplificar la aplicación en algunos aspectos, con el fin de reducir la confusión que sufre el usuario en algunas ventanas.

6.3.5. Azure

Con el objetivo de mejorar la velocidad de la base de datos de Azure, podríamos contratar un plan, el cual mejoraría la rapidez de nuestra aplicación a la hora de hacer operaciones con la base de datos, capacitándola para que reciba mas carga de trabajo.

6.3.6. Mejorar la gestión de las grabaciones de los usuarios

Mejoraremos la gestión de la sesión grabada de los usuarios introduciendo modificaciones en la base de datos, en vez de guardar el vídeo en bytes que satura la base de datos y hace que sea más lenta, guardaremos el enlace al vídeo.

Para almacenar las grabaciones crearemos un repositorio local para que la aplicación pueda guardar las grabaciones de los usuarios en local y otro en la nube para tener una copia de las grabaciones y que el médico pueda acceder a ellas cómodamente.

Capítulo 7

Organización del trabajo y aportaciones.

7.1. Organización del equipo

Siempre hemos intentado trabajar en paralelo, para evitar solapar nuestros desarrollos y para tener los dos un amplio conocimiento de cómo funciona la aplicación y de cómo estaba implementada. Para ello, mientras uno trabajaba en la aplicación, el otro investigaba sobre soluciones de problemas que teníamos o sobre posibles mejoras que podríamos incorporar.

7.1.1. Herramientas de comunicación

Para el desarrollo de la aplicación y mantener versiones de código estables del mismo en un lugar seguro, usábamos un plugin llamado Team Explorer para Visual Studio, el cual permitía usar Git (software de control de versiones) desde Visual Studio.

Para la comunicación y compartición de archivos entre nosotros y con nuestro director de proyecto hemos utilizado varias herramientas:

- **Gmail**, se ha usado principalmente para comunicarnos con nuestro director.
- **Google Drive**, para compartir todo tipo de archivos necesarios para el desarrollo del proyecto: imágenes, documentación, enlaces, etc.
- **Whatsapp**, para comunicarnos rápidamente entre nosotros y para soluciones rápidas.

- **Skype**, para comunicación puntual sobre el desarrollo sobre todo de la memoria.

7.1.2. Herramientas de organización

Hemos necesitado usar distintas herramientas para mantener una buena organización. A continuación indicamos las herramientas que hemos utilizado:

- **GanttProject**, programa gratuito que te permite hacer diagramas de gantt, gestionar proyectos y exportar el diagrama como imagen.
- **Software Ideas Modeler**, programa gratuito si eres estudiante, que permite hacer cualquier tipo de diagrama (clases, casos de uso, actividad, componentes, etc) de forma fácil e intuitiva y exportarlo en formato pdf o como imagen.

7.2. Aportaciones al proyecto

Ambos miembros del grupo hemos participado alrededor de las mismas actividades sin establecer una división clara, ya que, nos parecía mejor que ambos tuviéramos conocimiento de todo lo desarrollado en la aplicación. De este modo mejoramos nuestros conocimientos sobre esta tecnología y sus entornos de trabajo.

7.2.1. Christian Gómez Redondo

Desde el principio he participado y realizado diferentes aspectos del proyecto. Para llevar a cabo la realización final del proyecto tuvimos que trabajar juntos y llevamos a cabo un reparto de tareas. A continuación, paso a comentar y enumerar las tareas realizadas:

- Lectura y búsqueda en artículos, libros y páginas web sobre el uso de la realidad virtual, para conocer bien la tecnología que íbamos a implementar de la cual no ves nada en la carrera.
- Realización de tutoriales en Visual Studio y C-Sharp para obtener más conocimientos sobre el entorno de desarrollo y sobre el lenguaje de programación que vamos a usar, para saber cómo funciona y organiza. [5][6]
- Documentarse sobre Kinect y su uso en Windows y sobre que librerías nos permitiría alcanzar nuestros objetivos. [13-19]

- Realización de prototipos en Visual Studio, de pequeñas aplicaciones para entender bien las prestaciones de Kinect y el funcionamiento de Visual junto con C-sharp. Por ejemplo me encargué de desarrollar el prototipo del juego de las Torres de Hanoi (véase figura 7.2).

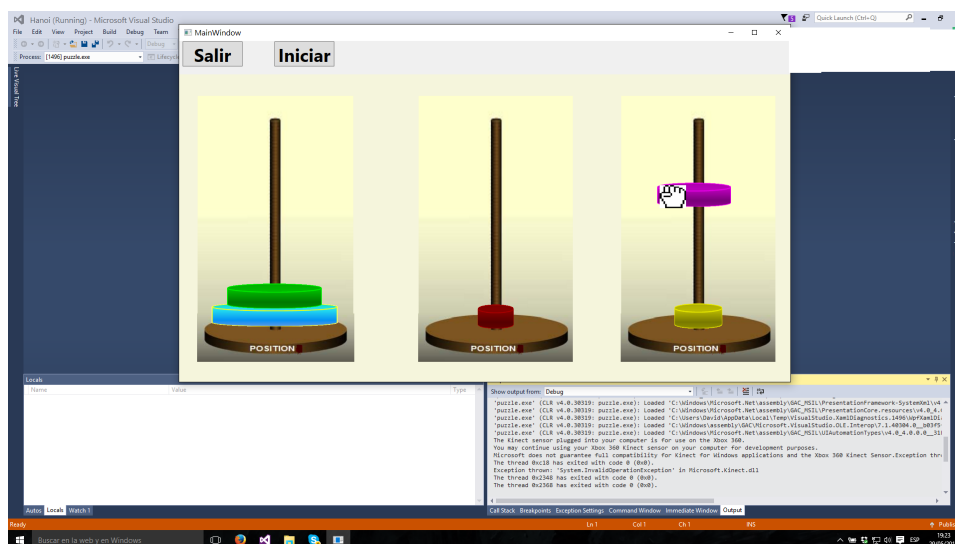


Figura 7.1: Juego Torres de Hanoi

- Participación en las reflexiones que dieron origen a la idea final junto a mi compañero David y nuestro profesor José Miguel.
- Realización de prototipos en papel, de cómo queríamos que fuese nuestra aplicación tanto en diseño como en funcionalidad.
- Desarrollo de caso de uso investigando en apuntes de otros años, para determinar los pasos que debe realizar el usuario cuando este usando nuestra aplicación. [25]
- Desarrollo del diagrama de actividad siguiendo los apuntes de la asignatura Ingeniería de Software, para detallar el proceso que sigue el usuario hasta completar cada acción del caso de uso. [11]
- Colaboración tanto en el diseño, como en la programación de los dos juegos que finalmente íbamos a desarrollar.
- Investigación de librerías y adaptación de Aforge a nuestro proyecto, para poder grabar la sesión con los usuarios. En un principio probé Expression Encoder, pero no daba los resultados esperados y decidimos usar Aforge, concretamente AforgeVideo. [1]

- Configuración de la base de datos, previamente elegida por David, alojada en Azure y del servidor de Azure. Entre otros factores configuré las cadenas de conexión, el firewall, usuarios y activar el cifrado de datos.
- Asistencia a las reuniones con nuestro profesor para aclarar ideas y dudas, así como para mostrarles nuestros avances en la aplicación.
- Contribución en la creación del plan de evaluación de la aplicación y de los cuestionarios que los usuarios debían realizar en ella.
- Evaluaciones con usuarios y la correspondiente interpretación de los resultados.
- Posterior realización de cambios acordados tras las evaluaciones.
- Creación de esta memoria y escritura en todos los capítulos de la misma. Repaso final y lectura de todo el documento, incluyendo todos los aspectos que aún no estaban incluidos en ella.

7.2.2. David Salcedo Gutiérrez

He participado en todas las etapas del proyecto, en el desarrollo de la aplicación, la programación del código, el estudio de las tecnologías usadas en el proyecto y en el desarrollo de la memoria.

La toma de decisiones en la dirección que hemos ido orientando el proyecto, ha sido realizada tanto por Christian como por mi. A continuación enumerare y comentare las tareas o aportaciones que he realizado en el proyecto:

- Aprendizaje del lenguaje de programación C-Sharp. Desarrollé una aplicación durante prácticas en empresa en C-sharp para poder aprender correctamente el lenguaje antes de empezar el proyecto. [5][6]
- Participé en la planificación y reparto de tareas del proyecto junto Christian. También realizamos un diagrama de Gantt para dejar reflejada la planificación del proyecto. [11]
- Diseño de los diagramas consultando en los apuntes de otros años, proyectos e Internet, para realizar los diagramas. Los diagramas que he realizado son: Diagramas de Clases, Diagramas de Secuencia, Diagrama de Componentes y Diagramas de Despliegue.
- Aprendizaje e investigación del uso de la Kinect, he realizado tutoriales de como implementar gestos en la kinect y desarrollé un piano y la versión previa del puzzle implementado en el proyecto. [13-19]

- Intentando que Kinect detectase los dedos para poder coger objetos, conocí la librería Kinect for Windows Developers Toolkit. Es la principal librería que hemos usado para el manejo de Kinect para aprender a usarla desarrollé un paint como se ve en la imagen que viene a continuación(véase Figura 7.3).



Figura 7.2: Paint

- Participé en la actualización y adaptación del puzzle desarrollado previamente con todo lo aprendido en el desarrollo del paint usando los controles de la nueva librería algunos de estos controles son: *KinectRegion*, *KinectButton*, *KinectScroll*...
- Investigación de las cloud para poder alojar la base de datos de nuestra aplicación. Entre las cloud que estuve investigando para utilizar están pydio(Sirve para crear una cloud particular es gratuita de software libre), AppHarbor, AWS(Amazon Web Service) y Azure que es la que decidimos usar por motivo de compatibilidad. [2] [3] [24]
- Participé en la implementación de los factores de autenticación de la aplicación (Login y Registro) junto con el patrón de diseño MVVM y el diseño del diagrama de la base de datos que está alojada en Azure. [20]
- Participación en desarrollo del juego Supermercado con la introducción, entre otras cosas, del movimiento de Grip y Push.

- Despliegue de la aplicación para poder instalar el proyecto en otros equipos. He investigado cómo publicar un proyecto en Visual Studio y cómo hacer que, con la instalación, descargue las librerías que necesita para funcionar. Se intentó integrar las librerías de la kinect para que se instalarán junto al proyecto pero no fue posible.
- Participé en las pruebas de evaluación de usuario, detectando errores y ejerciendo como moderador en algunas de las pruebas.
- Escritura en todos los apartados de la memoria y revisión de contenidos de la misma.

Bibliografía

1. Aforge. Librerías y framework de AForge : <http://www.aforgenet.com/framework/samples/>
2. Azure. Portal de Azure: <https://portal.azure.com/>
3. Azure.Documentación y ejemplos de Azure: <https://azure.microsoft.com/en-us/documentation/articles/>
4. caractKinect. <http://playerslink.la/se-filtran-aspectos-tecnicos-del-kinect-de-xbox-720/>
5. C-Sharp. Tutoriales y ejemplos de C-Sharp: <https://msdn.microsoft.com/es-es/library/kx37x362.aspx>
6. C-Sharp. Aprender manejo de hilos en C-Sharp: <http://www.albahari.com/threading/>
7. Fobias. <http://www.efesalud.com/blog-salud-prevencion/las-fobias-mas-comunes/>
8. Foro. Foro de stackoverflow para resolver dudas: <https://stackoverflow.com/>
9. Foro. Foro de microsoft para resolver dudas: <https://social.msdn.microsoft.com/Forums/es-ES/home/>
10. Git. Control de versiones: <https://git-scm.com/>
11. GanttProject. Diagrama de Gantt: <https://www.ganttproject.biz/>
12. KinectImagen. <http://kinectgva.blogspot.com.es/>
13. Kinect.Tutoriales: <https://blogs.msdn.microsoft.com/esmsdn/2011/07/07/reto-sdk-de-kinect-desarrolla-con-kinect/>
14. Kinect. Implementación de la kinect en varios ejemplos en C-Sharp: <https://msdn.microsoft.com/en-us/library/hh855377.aspx>
15. Kinect. Hacking the Kinect Created by ladyada: <https://blog.adafruit.com/2014/12/31/goodbye-original-kinect-we-had-fun-hacking-you-and-so-did-everyone-else-kinectwindows-kinectwindows/>

16. Kinect. Juegos educativos con la kinect: <http://www.kinecteducation.com/>
17. Kinect. Beginning Kinect Programming with the Microsoft Kinect SDK Created by Jarrett Webb and James Ashley: <https://www.apress.com/9781430241041>
18. Kinect. Kinect for Windows SDK Programming Guide Created by Abhijit Jana: <https://www.packtpub.com/game-development/kinect-windows-sdk-programming-guide>
19. Kinect. The Kinect for Windows sensor and SDK and SDK Created by Dr Leandro Dihl: http://www.inf.pucrs.br/~smusse/CG/PDFs2014_1/Kinect.pdf *Kinect.Tu*
http://www.codeproject.com/Tips/701338/Kinect-status-and-setup-the-Kinect-for-interaction
20. MVVM. Explicación detallada del modelo MVVM con ejemplos: <http://www.codeproject.com/Articles/View-View-Model-MVVM-Explained>
21. Nintendo. <https://www.nintendo.es/Juegos/Nintendo-DS/Big-Brain-Academy-270143.html>
22. Propiedades RV. <http://www.cs.upc.edu/~virtual/SGI/guions/ArquitecturaRV.pdf>
23. Realidad Virtual. William R. Nugent, Virtual Reality: Advanced Imaging Special Effects Let You Roam in Cyberspace, Journal of the American Society for Information Science, September 1991.
24. SQL Server. Explicación de cómo sincronizar dos bases de datos Sql Server: <https://blogs.msdn.microsoft.com/zkap/2012/04/12/synchronize-sql-server-and-sql-azure-using-sync-framework/>
25. Software Ideas Modeler. Diagramas de clases, Casos de uso: <https://www.softwareideas.net/>
26. Team explorer. Para usar git desde el visual studio: <https://www.microsoft.com/en-us/download/details.aspx?id=47727>
27. THOFU. <http://www.cedint.upm.es/es/proyecto/thofu-0>
28. TeachU. <http://devpost.com/software/teach-u-vr-87nlkz>
29. VentajasDesventajas. https://prezi.com/jewfetlnyu_e/ventajas-y-desventajas-de-la-realidad-virtual/VirtualRealityMedicine. <http://www.techrepublic.com/article/10-ways-virtual-reality-is-revolutionizing-medicine-and-healthcare/>

