



Escuela técnica superior de ingeniería (ICADE)

Grado en ingeniería electromecánica

Desarrollo de simulador de armas y su integración en una cueva de realidad virtual-CAVE

Autor: Ramón Domínguez Ferreiro

Director: José San Martín López

20 de Julio de 2016



AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESIS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. **Ramón Domínguez Ferreiro** DECLARA ser el titular de los derechos de propiedad intelectual de la obra: **Desarrollo de simulador de armas y su integración en una cueva de realidad virtual-CAVE**, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.
- b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- e) Asignar por defecto a estos trabajos una licencia Creative Commons.
- f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

- a) Que la Universidad identifique claramente su nombre como autor de la misma
- b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- c) Solicitar la retirada de la obra del repositorio por causa justificada.
- d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

- a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.



- b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
- c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.
- d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a de de

ACEPTA

Fdo.....





Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título *Desarrollo de simulador de armas y su integración en una cueva de realidad virtual-CAVE* en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2015-16 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Ramón Domínguez Ferreiro

Fecha: 9/ 07/ 2016

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: José San Martín López

Fecha: 9/ 06/ 2016

Vº Bº del Coordinador de Proyectos

Fdo.: Álvaro Sánchez Miralles Fecha://





Desarrollo de simulador de armas y su integración en una cueva de realidad virtual-CAVE

Autor: Domínguez Ferreiro, Ramón.

Director: San Martín López, José.

Entidad Colaboradora: ICAI - Universidad Pontificia Comillas

Resumen del Proyecto

INTRODUCCIÓN:

Las últimas tendencias de mercado demuestran un interés tanto por parte de las marcas como de los usuarios por la tecnología de realidad virtual. La gran distribución de estas tecnologías está prevista para el actual año 2016 en el que grandes compañías como HTC y PlayStation lanzan al mercado sus gafas de realidad virtual [1].

Las gafas de realidad virtual son una tecnología en la que, mediante el uso de un casco; al que se le ha provisto de un sensor de movimiento, una pantalla de alta resolución a la altura de los ojos y auriculares, se aísla al usuario de la realidad que le rodea para centrarlo en un entorno virtual creado con el fin de generar una inmersión del usuario [2].

Este tipo de tecnología se pretende usar como una fórmula de ocio para el usuario en la que se ofrecen diversos tipos de experiencias interactivas gracias a la comunicación que tiene el usuario con el entorno simulado. Este tipo de experiencias pueden ser videojuegos, películas, muestras de productos, etc.

Frente a esta tendencia se extiende otra, más propia del ámbito privado o profesional que es el uso de elementos periféricos para crear la sensación de que el entorno que rodea al usuario es otro.

Esto se ve especialmente reflejado en las cuevas de realidad virtual, entornos de paredes en el que se proyecta la simulación que mediante unas gafas estereoscópicas se añade la sensación de profundidad a la imagen y se consigue la integración del usuario en dicha realidad [3].

PLANTEAMIENTO DEL PROBLEMA.

Las características que ofrecen estas simulaciones consiguen la integración del usuario a nivel óptico y auditivo, pero para poder ofrecer una experiencia realmente interactiva es necesario el desarrollo de periféricos que sirvan para mantener al usuario integrado dentro de la realidad que se le presenta.



En el caso de las experiencias de simulación militar, ya sea para la formación de cuerpos profesionales de la seguridad o para el disfrute de un usuario civil en su tiempo de ocio, es necesario el desarrollo de un periférico que simule el arma con el que se está trabajando.

El arma no solo ha de ser un modelo realista, sino que además debe contar con un sistema que transmita la información sobre su uso al simulador para que este reaccione de acuerdo a los movimientos que realice el usuario y un método con el que se debe recrear el tipo de retroceso que tiene un arma real para que se trate de una simulación lo más realista posible.

ESTADO DE LA TÉCNICA.

Actualmente en el mundo de los simuladores la tecnología en la que estos se centran para detectar en tiempo real la posición y orientación del arma es el uso de medios ópticos. El medio por el que esta detección es realizada es posible mediante dos configuraciones: colocar el receptor en el arma y los emisores que proporcionan la referencia en el algún punto del entorno conocido y medido o colocar los emisores necesarios en el arma y los receptores en un marco fijo. [4]

El campo profesional cuenta con sistemas más costosos, sofisticados y de mayor envergadura que los dispuestos para el uso civil con el fin de poder entrenar a escuadrones de forma simultánea. Al contar con un presupuesto mayor los sistemas incluyen numerosas pantallas a lo largo de circuitos establecidos que permiten realizar diversas configuraciones de simulación. [5]

Las réplicas que se utilizan cuentan con diversos sistemas de alimentación que pueden llegar a limitar la libertad de movimientos por parte del usuario condicionando el tipo de simulación que es posible realizar.

El acceso de civiles a este tipo de simuladores no suele ser posible debido a su alto coste y las opciones que disponen están limitadas a las que ofrecen compañías de ocio digital que son de menor coste, pero con el correspondiente peor acabado en lo referente a realismo. Productos que pueden ofrecer este tipo de experiencia son aquellos asociados a periféricos como el Playstation Move o el Wiimote. [6]

OBJETIVOS:

Puesto que los sistemas de realidad virtual dispuestos para formación de cuerpos de seguridad resultan demasiado costosos y aquellos dispuestos para el uso y disfrute de ocio se centran más en el realismo de la interfaz que de los periféricos, el objetivo del proyecto es diseñar un sistema de simulación semejante a los que se utilizan para la formación policial en algunos lugares de E.E.U.U. que tenga un acabado lo suficientemente realista, acercándose al nivel de calidad de los simuladores profesionales, pero un coste reducido para que sea accesible al público general.



La construcción de un modelo realista tiene una serie de requisitos que han de ser cumplidos en la consecución del mismo:

-Elemento Inalámbrico: El periférico que actúe como arma ha de ser completamente independiente de forma que no necesite ningún tipo de conexión, ya sea por motivos de alimentación u de transmisión de datos.

-Sistema de apuntado: Para que la simulación resulte realista es necesario la creación de un método de reconocimiento de la orientación del arma robusto con el fin de poder proporcionar al entorno creado los datos necesarios. Se busca también la innovación por lo que es de interés comprobar la viabilidad de tecnologías de localización que no se encuentren en uso en el mercado.

-Retroceso: Con motivo de otorgar el máximo realismo posible al simulador es condición que este tenga integrado un método que permita recrear la fuerza que el arma realiza contra el usuario que dispara y el desapuntado que se produce en la misma correspondiente al desplazamiento que se da por la energía del disparo.

-Entorno: Ha de crearse para este proyecto un entorno virtual que permita comprobar de las bondades que provee el sistema desarrollado y cuyas capacidades sean suficientes para generar una simulación realista.

Recaltar la característica del proyecto que ha de ser de bajo coste con el propósito de ver el grado de viabilidad que tendría la implementación de un producto de estas cualidades en el mercado.

METODOLOGÍA:

Para poder llevar a cabo este proyecto, debido a la gran cantidad de elementos distintos que toca por su naturaleza, es necesario separarlo en módulos centrados en cada una de las funcionalidades que ha de tener el simulador de forma que sea viable abarcarlo todo.

Las diferentes secciones de este proyecto son:

-Diseño físico del arma: Estudio de la réplica de airsoft del subfusil HK G36C y adaptación del mismo a los requisitos de espacio y diseño que haya que cumplir con el fin de poder integrar los elementos que constituyen la funcionalidad del simulador.

-Desarrollo de un entorno virtual: Uso de las herramientas necesarias provistas por el motor gráfico *Unreal Engine 4* para crear un entorno en el que se pueda realizar una simulación realista.

-Sistema de transmisión de datos: Integración de un sistema que permita la comunicación entre el periférico y el ordenador que se ocupe de realizar la simulación que no sacrifique la libertad de movimiento por parte del usuario gracias al uso de *Bluetooth*.



-Sistema de retroceso: Diseño de un sistema que permita recrear la sensación de desapuntado y fuerza liberada contra el usuario por el disparo a través del uso de motores DC vibradores colocados en puntos estratégicos del arma.

-Sistema de disparo: Creación de un diseño que permita reconocer la activación del gatillo del arma y la transmisión de esa información a un módulo de control mediante pulsadores.

-Sistema de apuntado: Innovación del sistema de reconocimiento de posición mediante la integración de un giróscopo y varios sensores de ultrasonidos con el fin de poder localizar de forma efectiva la orientación del arma del usuario.

RESULTADO:

Después de haber finalizado el desarrollo se ha obtenido un periférico funcional con aspecto realista para un entorno de simulación virtual que cumple el requisito de tener un bajo coste en comparación a otros elementos del mercado (113.45€).

Trabajar con el dispositivo resulta sencillo para un usuario que carezca de formación específica en el campo de la ingeniería, pues no requiere de conocimientos técnicos.

La funcionalidad del dispositivo consiste en una comunicación realizada mediante *Bluetooth* con el ordenador en el que se encuentre el *Unreal Engine* instalado para llevar a cabo la simulación. Conectado el dispositivo, una interfaz sencilla muestra al usuario el punto al que se encuentra apuntando gracias a los datos que provee el giróscopo y la posición que tiene en el espacio según la información que transmiten los ultrasonidos. Esta misma interfaz se ocupa de reflejar el disparo, en el caso de que sea realizado mediante el gatillo al que se le ha implementado un pulsador, o la recarga de balas en el arma, a través de la pulsación del pulsador implementado en el muelle de recarga. Cuando se realiza el disparo, la colocación de los motores en el mango del arma y el apoyo de la culata transmiten de forma efectiva la sensación de retroceso propia de un arma real. La implementación del muelle en el proceso de recarga añade un plus de fuerza en la sensación de retroceso consiguiendo un acabado más realista y un mayor grado de desapuntado.

COMPONENTES	COSTE TOTAL PRODUCTO (€)
Arduino UNO	23,68
Sensor de ultrasonidos HC_SR04	10,58
Pulsador	3,42
Módulo MPU-6500 (Giróscopo)	2,75
Ppn13lb11c (Motor DC de alta velocidad)	5,32
Pila 9V	9,56
Placa para soldadura	2,07
Cables conexión Macho-Hembra (Pack 40Uds)	3,99
Resistencia 2.2K Ω	0,08
Resistencia 330 Ω	0,04
Resistencia 3.3K Ω	0,04
Transistor NPN 2N222a	1,65
Conmutador de 2 posiciones.	2,7
Réplica Airsoft HK G36C	35,9
Módulo <i>Bluetooth</i> HC-05	3,9
Hilo de estaño para soldadura	7,77
TOTAL:	113,45



CONCLUSIONES:

Finalizado el proyecto se cuenta con una réplica de la HK G36C que contiene todas las funcionalidades necesarias para que el proyecto se considere exitoso.

Se cumple la condición de un producto inalámbrico, la transmisión de datos se realiza mediante un módulo *Bluetooth* instalado en el cargador del arma junto a la batería que se usa como fuente de alimentación.

Se ha conseguido comprobar que la viabilidad del uso de técnicas de ultrasonidos para la localización espacial es lo suficientemente buena como para seguir innovando a través de esa vía de investigación. Junto al uso del giróscopo constituyen un sistema de apuntado que ha superado las pruebas de repetición y uso que se le han dispuesto.

El retroceso conseguido gracias al uso de los motores DC permite al usuario tener una referencia, no solo visual u auditiva provista por el simulador, sino también táctil de la fuerza del disparo.

El entorno desarrollado contiene la programación suficiente para que responda según las ordenes que se envíen a través del Arduino UNO dispuesto en el arma, no solo en lo referente a conducta del usuario, también en el comportamiento de los elementos del mismo cuando se interactúa con ellos.

El coste final se encuentra alrededor de los 100€ que en el caso de realizarse una producción en masa se vería reducido por lo que resulta un producto de bajo coste en el mercado.

Si bien, durante el desarrollo realizado se ha comprobado que el acabado de precisión del giróscopo trabajando junto al sensor de ultrasonidos posee un margen de mejora y que sería necesario la implementación de algún módulo de control más preciso que integre de forma adecuada los datos que ambos proveen. Es posible el uso de otras tecnologías como los *LEAP sensors* o el uso de infrarrojos, pero eso implicaría renunciar a la libertad que ofrece el uso de ultrasonidos cuya novedad es que se evita el uso de emisores/receptores en el entorno dando libre albedrío al usuario en su movimiento.

Otro elemento del proyecto que podría ser mejorado es fuerza aplicada durante el retroceso del disparo, pues si bien los motores demuestran ser efectivos en su percepción son incapaces de llegar a transmitir la cantidad de energía necesaria al usuario.



REFERENCIAS:

[1] 3 grandes tendencias que impulsarán la realidad virtual entre las marcas en 2016 (2016).PuroMarketing.

<http://www.puromarketing.com/12/25995/grandes-tendencias-impulsaran-realidad-virtual-entre-marcas.html>

[2] Pastor.J. (2016). La guerra de la realidad virtual 2016 ya está aquí: comparativa a fondo de todas las opciones. Xataka.

<http://www.xataka.com/realidad-virtual-aumentada/la-guerra-de-la-realidad-virtual-2016-ya-esta-aqui-comparativa-a-fondo-de-todas-las-opciones>

[3] Espadero. JM (sf). QUIMERA. La cueva de realidad virtual de la Rey Juan Carlos.

<http://www.gmr.v.es/~jespa/quimera.html>

[4] Dondgdong Weng, Yue Liu, Yongtian Wang, Lun Wu “Study on an Indoor Tracking System Base on Primary and Assistant Infrared Markers”, Computer-Aided Design and Computer Graphics, 2007 10th IEEE International Conference on .

[5] VirTra Inc.

<http://www.virta.com/>

[6] Playstation Move tiene juegos para todos. Playstation.

<http://latam.playstation.com/ps3/playstation-move/games/>



FIREARM SIMULATOR DEVELOPMENT AND VIRTUAL REALITY INTEGRATION ON A CAVE DEVICE

Autor: Domínguez Ferreiro, Ramón.

Director: San Martín López, José.

Collaborating entity: ICAI-Universidad Pontificia Comillas.

Project Summary

INTRODUCTION

Market latest trends show an unusual interest on virtual reality technology, brands and users equally. A great distribution is thought for the present year 2016 since huge multinationals like HTC or Sony are releasing their new virtual reality headsets on the market [1].

Virtual reality headset is an edge technology using a helmet provided with a sensor move, a high resolution screen and headphones in order to focus the user on a virtual environment specifically developed to create a sensation of immersion [2].

This kind of technology is thought to be an entertainment providing different kinds of interactive experiences thanks to the communication the user has with the simulated environment. Between this experiences videogames, movies, product teasers... can be found.

On the other hand, professional institutions tend to use peripherals in order to simulate a different environment.

Virtual reality caves are the best example of this kind of tendency. Image projection over different walls accompanied by the use of stereoscopic glasses to create a depth sensation and allowing the user to interact with this fake reality [3].

APPROACH TO THE PROBLEM

These type of simulations allow the integration of the user on an optic and auditory level, but in order to offer an experience close to the reality, it is necessary to develop peripherals which can keep an immersion level according to the reality shown to the user.

Military simulation experiences need the development of a firearm like peripheral to work with. It does not matter if it is for professional security training or entertainment use.

Weapon model must have a system to allow the communication with the simulator, so all the information about weapon position or orientation can be used by the computer to react accord to the real world.



In addition, model also has to recreate the recoil provided by the real weapon so the use of the replica feels like using the real weapon.

STATE-OF-THE-ART

Nowadays, technology used on simulators is focus on detecting in real time the weapon orientation and position thorough optics. This detection may be done in two different ways: having a receiver on the weapon used and the signal emitters located on a fixed point with known references or having one or more emitters located on the weapon and make receivers fixed points [4].

Professionals use bigger, more sophisticated and way more expensive simulators than the ones provided for entertainment so they could train a great number of soldiers simultaneously. Having great budgets allows the deployment of several screens along corridors and rooms so different configurations and trainings are available. [5]

Replicas used by trained forces have different kinds of power supply. This may limit the freedom of movements to the user so simulations may be conditioned by that freedom.

Civil access to these simulators is not possible most of the time due to the cost of them, even if it is used by renting them. Digital Entertainment companies provide affordable options to the public, but the quality of the realism is way worse than the professional simulators. Playstation Move or Nintendo Wiimote are products that can provide a similar experience [6].

PROJECT OBJECTIVES

Since virtual reality systems disposed to the formation of security forces are not affordable and entertainment simulators are more focus on graphic realism than peripheral sensations, the goal of this project is to design a police training simulation system which has a finish similar to the professional simulator but with a price close to the entertainment simulator.

Building a realistic model has a series of conditions that must be accomplished:

- Wireless: Weapon peripheral must be totally independent so no connexion is needed. So the firearm must not depend of a wired power supply or make the data transfer thorough a cable.

- Pointing module: In order to make a realistic simulation, a way to locate the position and the orientation of the weapon must be made. This way the computer could use the data to correspond the movements on screen. This project search innovation so it is necessary to check the viability of the positioning technologies which are not in use by the market nowadays.



-Recoil: To make a maximum realistic firearm replica it is needed to install a system which reproduces the force made by the bullet when it is shot and the oscillations made by the weapon to miss the target.

-Environment: To create a realistic landscape where the characteristics of the weapon can be tested and its finish is good enough to emulate reality is condition of this project.

A low cost design must be emphasized so the viability of a market implementation can be tested.

METHODOLOGY

To carry this project on, it is necessary to split it on different modules according to the functionalities of the simulator due to its multidisciplinary nature. This way it is possible to cover all its parts by one person.

The different parts of this project are:

-Physical weapon design: Study of the airsoft automatic rifle HK G36C replica and modifications of it, so the requirements of space and design to incorporate the different parts of the simulator are fulfilled.

-Virtual environment development: Use of the tools provided by the game design engine Unreal Engine to develop an environment to recreate a portion of reality where a realistic simulation can be made.

-Data transfer system: The use of Bluetooth allows the communication between the peripheral and the computer without sacrificing freedom of movement.

-Recoil system: Recreate the force released against the user when this shoot a real weapon it is possible thanks to the correct positioning of DC motors on the handle and the butt of the firearm.

-Shooting system: Install a pair of switches on the trigger and the reload lever that allow to the control system know when the trigger has been pulled and when a reload of bullets has been made.

-Pointing system: Innovate the way of locating the weapon position and orientation thanks to the use of inertial and sonic technology. A gyroscope on the cannon of the firearm controls the variation of the orientation and two ultrasounds pointing at the front of the user and the floor keeps the system informed of the position of the weapon.



RESULTS

After the development of this project the result is a totally functional realistic appearance firearm to work on a virtual simulation environment.

The final budget of the materials is 113.45€ which is lower than other simulators on the market.

The device is designed to be user-friendly so a user without technical or engineering knowledge may work with it.

The functionality of the simulator consists of a communication done by the Bluetooth module which establish the data transfer between the control of the peripheral and the computer which has installed the Unreal Engine to run the simulation.

Once the device has been linked, a very simple interface shows the user where the gun is pointing at thanks to the data provided by the gyroscope and the position the user has on space according to the variation perceived by the ultrasounds.

The same interface displays the shot once it is made by pulling the trigger and pressing the switch or the reload made by pressing the switch located near the reload lever.

Once the shot has been made, the DC motors are activated for a lapse of time in order to create the sensation of recoil to the user similar to the provided by a real weapon.

Keeping the spring from the original shooting system of the replica allows to the user increase the force released during the shot bringing the recoil closer to its original sensation.

COMPONENTES	COSTE PRODUCTO (€)	TOTAL
Arduino UNO	23,68	
Ultrasound sensors HC_SR04	10,58	
Switches	3,42	
Module MPU-6500 (Gyroscope)	2,75	
Ppn13lb11c (DC Motor)	5,32	
Batery 9V	9,56	
Welding plate	2,07	
Cables (Pack 40Uds)	3,99	
Resistor 2.2KΩ	0,08	
Resistor 330Ω	0,04	
Resistor 3.3KΩ	0,04	
Transistor NPN 2N222a	1,65	
2 Positions commutator	2,7	
Airsoft replica HK G36C	35,9	
Bluetooth module HC-05	3,9	
Tin Thread	7,77	
TOTAL:	113,45	



CONCLUSIONS

The HK G36C replica obtained at the end of project has all the features necessities to consider this project as a success.

A wireless device has been developed since the data transmission is made thanks to a Bluetooth module placed close to the battery use as power supply inside the charger of the weapon.

Ultrasounds technics are a reliable way to locate on space the position of the user and it has been proved good enough to keep investigating on that direction. The tests made to the ultrasounds sensor have been overcome with the support information provided by the gyroscope.

DC motors are a recoil reference provided to the user thorough the weapon to know when the firearm have been shot without need of the visual and auditory information of the simulator.

Virtual environment has been programmed to respond according to the orders sent thorough the Arduino UNO disposed on the weapon corresponding to the moves made by the user. Also, the behaviour of the elements of the simulator is prepared to react when they have been shot.

Final budget is about 100€ and in case of mass production the cost will be even lower so in terms of market is a low cost product.

However, during the development of the firearm the work of the gyroscope and the ultrasound sensors had some issues when they work together so it is possible the design of a filter which integrates the data provided by both sensors in order to avoid the overlapping of orders in the simulator. It is possible the use of other technologies like LEAP sensors or infrared communication but that implies a sacrifice on freedom of movement offered by the use of ultrasounds which do not need emitters/receivers on the environment.

Another aspect which can be improved is the energy transmitted to the user by the DC motors. These devices cover their function of create a force to simulate the existence of a recoil but sometimes may be not enough to move the user.



REFERENCES:

[1] 3 grandes tendencias que impulsarán la realidad virtual entre las marcas en 2016 (2016).PuroMarketing.

<http://www.puromarketing.com/12/25995/grandes-tendencias-impulsaran-realidad-virtual-entre-marcas.html>

[2] Pastor.J. (2016). La guerra de la realidad virtual 2016 ya está aquí: comparativa a fondo de todas las opciones. Xataka.

<http://www.xataka.com/realidad-virtual-aumentada/la-guerra-de-la-realidad-virtual-2016-ya-esta-aqui-comparativa-a-fondo-de-todas-las-opciones>

[3] Espadero. JM (sf). QUIMERA. La cueva de realidad virtual de la Rey Juan Carlos.

<http://www.gmr.v.es/~jespa/quimera.html>

[4] Dondgdong Weng, Yue Liu, Yongtian Wang, Lun Wu “Study on an Indoor Tracking System Base on Primary and Assistant Infrared Markers”, Computer-Aided Design and Computer Graphics, 2007 10th IEEE International Conference on.

[5] VirTra Inc.

<http://www.virta.com/>

[6] Playstation Move tiene juegos para todos. Playstation.

<http://latam.playstation.com/ps3/playstation-move/games/>



ÍNDICE DE CONTENIDO:

1. MEMORIA DESCRIPTIVA:	26
a. Introducción.	28
b. Estado del arte.	28
c. Motivación.	35
d. Objetivos del proyecto.	36
e. Metodología.	36
2. ELECCIÓN DE TECNOLOGÍAS:	38
a. Selección del modelo del arma.	40
i. Fusil M4	41
ii. Fusil HK G36C	42
iii. Subfusil MP7	43
iv. Fusil AK-47	44
v. Escopeta Mossberg 500	45
vi. Pistola Colt 1911	46
b. Selección del sistema de apuntado.	48
i. Reconocimiento de referencias.	48
ii. Tecnología acústica.	49
iii. Tecnología de radiofrecuencias.	50
iv. Tecnología Mecánica.	50
c. Selección del sistema de retroceso.	54
i. Vibradores.	54
ii. Actuadores lineares.	54
iii. Pistones neumáticos.	55
iv. Muelles de retroceso.	56
d. Selección del microprocesador.	58
i. Familia Arduino.	59
ii. Raspberry PI	61
iii. PIC	62
e. Selección del sistema de disparo.	64
i. Pulsador.	64
ii. Fin de carrera.	64
iii. Sensor Capacitivo.	64
iv. Sensor de flexibilidad resistivo.	65
v. Fotorresistencia.	65
f. Selección del sistema de comunicación arma-simulador.	67
i. Bluetooth.	67
ii. WIFI.	68



iii. ZigBee.	68
3. ELECCIÓN DEL SOFTWARE.	70
a. Unreal Engine 4.	73
b. Unity 5.	76
4. MONTAJE DEL ARMA.	80
a. Desmontado del arma de airsoft.	82
b. Colocación del giróscopo.	83
c. Colocación de los interruptores.	84
d. Colocación de los motores DC.	85
e. Colocación de los sensores de ultrasonidos.	87
f. Colocación de los elementos en el cargador.	89
5. SISTEMA DE CIRCUITOS.	94
a. Bluetooth.	96
b. Giróscopo.	97
c. Sensores de ultrasonidos.	98
d. Pulsadores.	99
e. Motores DC.	100
6. SOFTWARE DE CONTROL DEL ARMA.	102
7. ENTORNO DE REALIDAD VIRTUAL.	106
a. Descripción del entorno.	108
b. Interfaz.	111
c. Personaje.	113
d. Bala.	118
e. Primera persona.	119
8. PRUEBAS Y ENSAYOS.	120
a. Pruebas de desapuntado.	121
b. Pruebas de oclusión.	124
c. Pruebas de repetibilidad.	126
d. Pruebas de autonomía y funcionamiento.	127
9. FUNCIONAMIENTO.	130



10.RESULTADOS Y CONCLUSIONES.	140
a. Resultados.	142
b. Conclusiones.	143
11.FUTUROS DESARROLLOS.	146
12.BIBLIOGRAFÍA.	152
ANEXO I: Configuración del Bluetooth.	
ANEXO II: Estudio Económico.	
ANEXO III: Calculo Teórico de la validez de algunos elementos usados en el proyecto.	
ANEXO IV: Código implementado en el arma.	



ÍNDICE DE FIGURAS

<i>Figura 1: Grafico sobre el rechazo ante el parecido humano y el “Uncanny Valley”.</i>	30
<i>Figura 2: Ejemplo de Cueva de realidad virtual CAVE.</i>	30
<i>Figura 3: Ejemplo de HMD acompañado de periféricos, Playstation VR.</i>	31
<i>Figura 4: Especialista quirúrgico realizando un procedimiento teleoperado.</i>	32
<i>Figura 5: Simulador Virtra 300 en mitad de una demostración.</i>	34
<i>Figura 6: Imagen de un fusil M4.</i>	41
<i>Figura 7: Imagen de una carabina HK G36C.</i>	42
<i>Figura 8: Imagen de un subfusil MP7.</i>	43
<i>Figura 9: Imagen de un fusil AK-47.</i>	44
<i>Figura 10: Imagen de una Mossberg 590.</i>	45
<i>Figura 11: Imagen de una M1911.</i>	46
<i>Figura 12: Imagen del Giróscopo GY-521.</i>	52
<i>Figura 13: Imagen del Módulo Ultrasónico HC-SR04.</i>	53
<i>Figura 14: Imagen de un motor extraído DC extraído de un mando DUALSHOCK.</i>	54
<i>Figura 15: Imagen de un actuador lineal.</i>	55
<i>Figura 16: Imagen de un pistón neumático.</i>	55
<i>Figura 17: Imagen de diversos muelles comercializados para las armas de airsoft.</i>	56
<i>Figura 18: Imagen de un pulsador.</i>	64
<i>Figura 19: Imagen de un fin de carrera.</i>	64
<i>Figura 20: Imagen de un sensor capacitivo.</i>	65
<i>Figura 21: Imagen de un sensor flexible resistivo.</i>	65
<i>Figura 22: Imagen de una fotorresistencia.</i>	65
<i>Figura 23: Imagen promocional del videojuego Detroit Become Human.</i>	73
<i>Figura 24: Imagen en 4k realizada por el artista Rense de Boer en Unreal Engine 4.</i>	74
<i>Figura 25: De Izquierda a derecha y de arriba abajo, imágenes promocionales de Kingdom Hearts 3, Paragon, Vampyr, Tekken 7 y Gears of War 4.</i>	75
<i>Figura 26: Imagen de una estructura a derribar de Angry Birds.</i>	77
<i>Figura 27: De izquierda a derecha y de arriba abajo, imágenes in-game de Slender: The Arrival, Firewatch, Fallout Shelter, Broforce, Ori and The Blind Forest Y Roller Coaster Tycoon World.</i>	78
<i>Figura 28: Esquema general de la disposición de los distintos módulos en el arma una vez abierta.</i>	82
<i>Figura 29: Detalle del hueco disponible para instalar el giroscopo.</i>	83



<i>Figura 30: Detalle del giroscopo una vez ha sido pegado y conectado a los cables de conexionado.</i>	83
<i>Figura 31: Detalle de la colocación del pulsador detrás del gatillo.</i>	84
<i>Figura 32: Detalle del pulsador de recarga en el que se puede apreciar la distancia con respecto a la palanca de recarga del muelle.</i>	85
<i>Figura 33: Detalle de la colocación de uno de los motores en el mango del arma.</i>	85
<i>Figura 34: Detalle de la colocación del segundo de los motores, pila de alimentación del módulo de retroceso y circuito asociado al mismo.</i>	86
<i>Figura 35: Imagen de la localización de los sensores en el arma.</i>	87
<i>Figura 36: Imagen de un usuario con el arma real con localización del sensor vertical marcado en rojo y distancia desde el mismo hasta el centro de gravedad del cuerpo.</i>	88
<i>Figura 37: Imagen de los sensores una vez colocados y fijados a la boca del arma.</i>	88
<i>Figura 38: Detalle del sensor frontal mostrando el hueco para los cables de los pines.</i>	89
<i>Figura 39: Apreciación del espacio disponible dentro del cargador.</i>	90
<i>Figura 40: Comprobación del espacio que ocupan los componentes en el cargador.</i>	90
<i>Figura 41: Detalle del cargador una vez se han realizado todas las conexiones antes de cerrarlo.</i>	92
<i>Figura 42: Detalle del Bluetooth con sus pines.</i>	97
<i>Figura 43: Imagen de los pines que posee el giróscopo.</i>	97
<i>Figura 44: Imagen de los pines propios del sensor de ultrasonido.</i>	98
<i>Figura 45: Conexión de los elementos que van directos al Arduino.</i>	99
<i>Figura 46: Esquema de la conexión de los pulsadores.</i>	100
<i>Figura 47: Esquema del circuito asociado a los motores.</i>	101
<i>Figura 48: Plano lateral del simulador.</i>	108
<i>Figura 49: Detalle de un árbol con partículas implementadas.</i>	109
<i>Figura 50: Plano cenital donde se aprecia la no proyección de sombras hacia los laterales de los elementos.</i>	110
<i>Figura 51: Valores definidos en el actor SkySphereBlueprint.</i>	110
<i>Figura 52: Sección del Blueprint del First Person HUD ocupada de dibujar la cruceta.</i>	112
<i>Figura 53: Sección del Blueprint que se ocupa de mostrar las balas del cargador.</i>	112
<i>Figura 54: Sección del Blueprint responsable de abrir el puerto de comunicación serie.</i>	113
<i>Figura 55: Sección del Blueprint responsable de cerrar los puertos al finalizar la comunicación.</i>	114
<i>Figura 56: Fragmento de sección del Blueprint donde se está separando y guardando el valor correspondiente a la variación de movimiento en el eje X.</i>	114



<i>Figura 57: Sección del Blueprint asociada al movimiento de la cámara.</i>	115
<i>Figura 58: Sección del Blueprint responsable del avance del personaje.</i>	116
<i>Figura 59: Sección del Blueprint responsable del salto.</i>	116
<i>Figura 60: Parte del Blueprint de la sección de disparo que actúa como detector de flanco.</i>	117
<i>Figura 61: Sección del Blueprint que localiza al actor y obtiene un punto en función del modelo.</i>	117
<i>Figura 62: Sección del Blueprint encargada de generar la aparición de la bala.</i>	118
<i>Figura 63: Imagen del volumen de colisión (en rojo) asignado al proyectil.</i>	118
<i>Figura 64: Blueprint asociado al First Person Projectile.</i>	119
<i>Figura 65: Configuración realizada sobre el modo de juego.</i>	119
<i>Figura 66: Área en la que se encuentran contenidos todos los tiros de la prueba de desapuntado.</i>	122
<i>Figura 67: Trayectoria descrita por el arma debido al desapuntado producido por el muelle.</i>	124
<i>Figura 68: Situación inicial de las planchas para la oclusión.</i>	125
<i>Figura 69: Situación de las planchas en situación de conflicto.</i>	125
<i>Figura 70: Arma inmovilizada contra la mesa para asegurar la ausencia de vibraciones.</i>	126
<i>Figura 71: Area localizada donde se han producido los 40 disparos en la simulación.</i>	127
<i>Figura 72: Menú de configuración de Bluetooth en Windows 10.</i>	132
<i>Figura 73: Menú donde se muestra la configuración de los puertos del arma Bluetooth.</i>	133
<i>Figura 74: Launcher de Epic Games Company.</i>	134
<i>Figura 75: Pantalla del inicio del proyecto en Unreal Engine.</i>	134
<i>Figura 76: Gestor de contenido en la carpeta FirstPersonBP.</i>	135
<i>Figura 77: Gestor de contenido en la carpeta de Blueprints.</i>	135
<i>Figura 78: Ventana emergente del First Person Character.</i>	135
<i>Figura 79: Bloque blueprint Open Serial.</i>	136
<i>Figura 80: Previsualizador del nivel donde se aprecia el botón reproducir.</i>	137
<i>Figura 81: Imagen del problema de reconocimiento de distancia del ultrasonidos.</i>	148
<i>Figura 82: Conexión de Bluetooth para su configuración.</i>	162
<i>Figura 83: Parte trasera de un módulo Bluetooth con botón.</i>	163



ÍNDICE DE TABLAS

<i>Tabla 1: Características del Fusil M4.</i>	41
<i>Tabla 2: Características de la HK G36C.</i>	42
<i>Tabla 3: Características del Subfusil MP7.</i>	43
<i>Tabla 4: Características del AK-47.</i>	44
<i>Tabla 5: Características de la Mossberg 590.</i>	45
<i>Tabla 6: Características de la M1911.</i>	46
<i>Tabla 7: Características del HC-SR04.</i>	53
<i>Tabla 8: Comparativa de los modelos de Arduino que se encuentran actualmente en el mercado.</i>	60
<i>Tabla 9: Comparativa de las especificaciones de los distintos modelos de Raspberry Pi en el mercado.</i>	61
<i>Tabla 10: Medida de los materiales del proyecto.</i>	170
<i>Tabla 11: Medida del equipo y de las herramientas.</i>	170
<i>Tabla 12: Medida de Software.</i>	171
<i>Tabla 13: Medida de la mano de obra.</i>	171
<i>Tabla 14: Precios unitarios de los componentes.</i>	172
<i>Tabla 15: Precio unitario de cada hora de trabajo en cada una de las actividades.</i>	173
<i>Tabla 16: Suma parcial del coste de los componentes.</i>	174
<i>Tabla 17: Suma parcial de los costes asociados a la mano de obra.</i>	175
<i>Tabla 18: Coste total asociado al proyecto.</i>	176



ÍNDICE DE ECUACIONES

<i>Ecuación 1: Formula de la intensidad en la base del transistor.</i>	101
<i>Ecuación 2: Relación entre la intensidad del colector y la base en hipótesis de funcionamiento en activa del transistor bipola.r</i>	101
<i>Ecuación 3: Ecuación del coste total de un producto.</i>	177
<i>Ecuación 4: Ecuación del beneficio.</i>	177
<i>Ecuación 5: Inclusión de la ecuación 3 en la ecuación 4.</i>	178
<i>Ecuación 6: Ecuación del beneficio incluyendo la venta a particulares y organizaciones.</i>	178
<i>Ecuación 7: Conversión de los ángulos de giro del arma a radianes.</i>	182
<i>Ecuación 8: Obtención de la velocidad angular de arma.</i>	182
<i>Ecuación 9: Obtención del tiempo que tarda el brazo en realizar todo el recorrido.</i>	182
<i>Ecuación 10: Obtención de la cantidad de ángulos que se miden en la fracción de tiempo que se tarda en realizar todo el recorrido del brazo.</i>	182
<i>Ecuación 11: Formula de la velocidad en función de la aceleración.</i>	183
<i>Ecuación 12: Formula de la distancia en función de la aceleración y velocidad inicial.</i>	183
<i>Ecuación 13: Formula de la fuerza obtenida como masa por aceleración.</i>	183



CAPITULO:1

MEMORIA DESCRIPTIVA





Introducción:

El ocio digital ha aumentado su diversidad de forma abrumadora a lo largo de los últimos años y cada vez llega de forma más diferente pues hay ejemplos como: las gafas de realidad virtual; *Oculus Rift* o *PlayStation VR*, los periféricos con sensores de movimiento en mayor o menor medida; el *Wii Mote* de Nintendo o el sistema *Sisaxis* de *PlayStation* y las cámaras con reconocimiento de posición de movimiento; el *EyeToy* de *PlayStation* o el *Kinetic* de *Xbox*.

Esta variedad de sistemas no solo es aplicable al campo del ocio, sino que también se pueden incorporar a la formación de personas en distintos campos, como puede ser la medicina o el entrenamiento militar. Esto se debe a que es posible simular una gran variedad de situaciones con antelación y un gran alto grado de precisión; ya sea un asalto armado a una casa o una operación especialmente complicada, si se dispone de las tecnologías de simulación adecuadas.

Actualmente es posibles simular este tipo de situaciones con relativamente poco equipo: una intervención quirúrgica puede realizarse tan solo con unas gafas de realidad virtual y un dispositivo que permita la realimentación háptica mientras que una operación militar se realiza mediante pantallas rodeando al usuario, unas gafas que permitan la inmersión y una réplica de arma que pueda comportarse como una real.

Las implicaciones de dicha situación conllevan que con el desarrollo adecuado en mente es posible diseñar un prototipo de bajo coste con las tecnologías actuales para uso doméstico.

Estado del arte

REALIDAD VIRTUAL

Si bien no existe una definición formal de lo que es la realidad virtual, podríamos acercarnos a la misma definiéndola como una Quimera según el Diccionario de la Real Academia Española “*Aquello que se propone a la imaginación como posible o verdadero, no siéndolo.*” [1], una ilusión que realmente no se encuentra ahí realizada por métodos informáticos para que aquellos que se ven envuelta en la misma la sientan como algo próximo.

Siendo un poco menos literarios se puede hablar de que consiste en la inmersión de un sujeto en un entorno controlado y diseñado por ordenadores tal que todas las sensaciones del mismo sean percibidas como algo cierto y tangible o dicho de otra forma “Real”.



Este tipo de situaciones ha sido ampliamente representada de forma idealizada en los medios de ocio como son el cine “The Matrix Trilogy” [2] , la Televisión “Eureka” [3], la literatura “Sword Art Online” [4] o los videojuegos “.HACK//” [5].

Si bien el ocio muestra de forma bastante fiel el tipo de integración que se pretende conseguir, la situación actual dista bastante de ser mundos de donde entrar y salir tan fácilmente.

Actualmente la realidad virtual se encuentra en un estado de progreso desde diferentes frentes en busca de los resultados más óptimos posibles, estas tendencias se pueden separar principalmente en dos: una primera; CAVE más orientada a crear un entorno alrededor del sujeto donde poder simular nuestra realidad y la otra llamada HMD (*Head Mounted Display*) [6]; que se centra en sumergir al individuo en diversos aparatos electrónicos que engañen directamente a su cuerpo para creer que está en una situación diferente.

En la línea de la primera corriente se encuentran principalmente elementos de inmersión total no invasivo como pretenden ser las cuevas de realidad virtual [7]: habitaciones equipadas con proyectores sobre sus paredes que rodean a la persona de forma que sienta que está en otro entorno, normalmente suelen llevar gafas de visión 3D que permiten añadir la profundidad que le falta a una simple proyección y que permiten a su vez realizar algún tipo de seguimiento sobre la posición de la persona dentro de la sala para realimentar de esta forma la simulación y a su vez para poder obtener la posición de los ojos y así alterar la imagen de forma que no se pierda la sensación de tridimensionalidad que se construye por la bilocación de imágenes separadas según la distancia entre ambos ojos para que el cerebro perciba una profundidad inexistente. Este tipo de habitaciones suelen estar gestionadas por un único ordenador cuya potencia podrá variar según la simulación necesaria y el entorno simulado puede ser tanto de imágenes reales grabadas por video como de imágenes 3D desarrolladas por animadores profesionales, aunque ahora mismo para este tipo de simulaciones se prefiere usar grabaciones debido a la dificultad de superar el *Uncanny Valey* (fig. 1). El *Uncanny Valey* (O Valle Inquietante en español) es la sensación de disconformidad que encuentra nuestro cerebro cuando algo pretende ser totalmente realista y humano sin llegar a conseguirlo llegando a provocar rechazo y la pérdida total de inmersión en la imagen visionada [8].

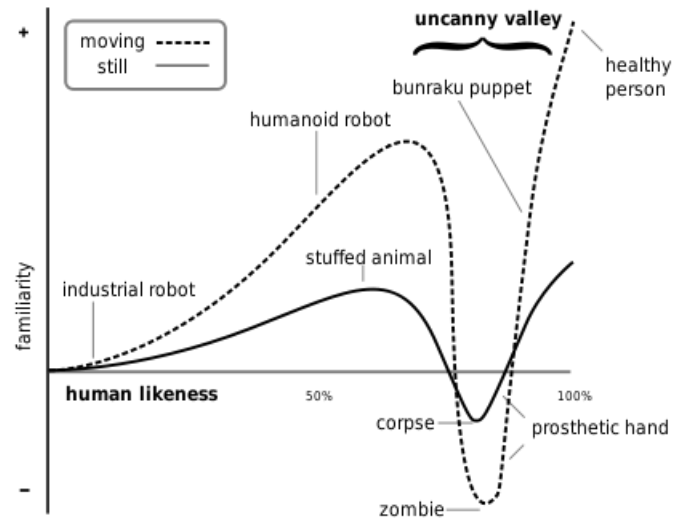


Figura 1: Grafico sobre el rechazo ante el parecido humano y el “Uncanny Valley”

Dentro de esta corriente no existen solo simuladores para una inmersión total (fig. 2), sino también para una simulación parcial que permita la realización de una tarea como si se estuviese en la situación de llevarla a cabo. Esto suele ser pantallas a lo largo de pasillos para entrenamiento de campo en operaciones militares o algún dispositivo con realimentación acompañado de alguna pantalla para la realización de operaciones quirúrgicas complicadas de forma que sea posible la práctica y previsualización total de un procedimiento antes de llevarlo a cabo sobre algún paciente.



Figura 2: Ejemplo de Cueva de realidad virtual CAVE

La HMD; que resulta más invasiva al usuario porque se suele rodear de periféricos, concentra su especial atención en los sentidos de la vista y el oído a los cuales pretende engañar mediante el uso de las anteriormente mencionadas “gafas de realidad virtual”.



Estas gafas, que bien podrían recibir el nombre de casco, aíslan totalmente al individuo de todo estímulo visual y auditivo externo para centrarlo específicamente en aquellas señales que le lleguen a través del aparato. Se facilita así la tarea de crear la imagen tridimensional que permita la inmersión, pues la imagen ya ha recibido el tratamiento estereoscópico necesario y el propio casco con su sensor permite el seguimiento de la posición de la cabeza para adaptar así la imagen a la colocación de la misma.

Las gafas suelen ir acompañadas, si se quiere una sensación total de inmersión interactiva, de periféricos que reaccionen de acuerdo a la realidad que se está percibiendo a través de las mismas. Este tipo de periféricos suelen ser los denominados *virtual gloves* [9]; guantes de realidad virtual, que están incorporados con sensores que permiten saber el agarre que está realizando la mano y la posición de la misma para que esta pueda actuar con la realidad que se le está mostrando a través de las gafas. Incluso con el uso de *virtual gloves*, este tipo de simulación suele acompañarse de gadgets con realimentación háptica en según qué procedimientos para que la persona que las usa pueda tener la sensación de estar cogiendo cosas con peso o que cuando actúa con alguno de los elementos de la falsa realidad, éste le envía algún tipo de estímulo para que realmente sienta que hay algo ahí.



Figura 3: Ejemplo de HMD acompañado de periféricos, Playstation VR

Este tipo de simulación de realidad tiene el inconveniente de que no se puede utilizar, tal y como está en la actualidad, para realizar escenarios que requieran de movimiento dentro del mismo por parte del usuario, sino que ha de depender de algún tipo de elemento externo que permita indicarle al sistema el desplazamiento, como puede ser por ejemplo un *joystick*. Se puede comprobar esto en alguna de las simulaciones más pulidas en el uso de unas gafas de realidad virtual, como son las ofrecidas por la compañía Sony en sus *Playstation VR*, pues son simulaciones que requieren a la persona que



simplemente esté sentada y utilice algún tipo de periférico como el *PlaystationMove* para interactuar con ellas (fig. 3) [10].

DISPOSITIVOS HÁPTICOS

Uno de los requisitos para que se puedan producir avances en el campo de la realidad virtual es el estudio de todo aquello relacionado con el sentido del tacto como pueden ser las sensaciones térmicas o de presión para su implementación en esos mundos ficticios. Aquellos dispositivos denominados hápticos son los que sirven para evocar sensaciones en el cuerpo humano del contacto con diversos objetos como si de verdad se estuviese en contacto con ellos [11].

El principal problema dentro de la tecnología que implica un engaño al sentido del tacto es la tasa de refresco de la misma. Por poner un sentido frente a otro: la vista solo necesita de una tasa superior a 24 fotogramas por segundo para percibir fluidez en el movimiento y no percibir un hueco en negro en su visión [12]. Por otro lado, los corpúsculos de Pacini; responsables de percibir las variaciones de presión, tienen una velocidad de hasta 1500 Hz siendo su frecuencia cómoda de trabajo alrededor de 300Hz [13].

El estudio de elementos hápticos tiene su centro de aplicación en la medicina. El fin del mismo es el poder simular el tipo de sensaciones que un cirujano tiene cuando está en contacto con una arteria, musculo o hueso y la temperatura dentro del cuerpo humano para así poder practicar procedimientos complicados antes de llevarlos a cabo. Es posible gracias a esta tecnología realizar la cirugía mediante robots en el caso de que requiera un alto grado de precisión (fig. 4) o simplemente no haya un doctor adecuado en las proximidades y sea necesario operar desde la distancia.



Figura 4: Especialista quirúrgico realizando un procedimiento teleoperado.

Sin embargo, los elementos hápticos tienen otro tipo de aplicaciones que ayudan al cerebro humano a poder entrar en el estado de tensión necesario para poder llevar a cabo ciertas tareas. Puede ser el simple caso de la vibración de un mando de videoconsola para mantener alerta al jugador en una situación difícil o el medidor de resistencia incorporado a los mandos de un avión que se vuelve más rígido según la resistencia que está ofreciendo



el aire para que así el piloto sepa que fuerza está consiguiendo con el ángulo de ataque de su vuelo.

Este tipo de sensaciones de tensión y pérdida del control es la que se pretende conseguir en los simuladores de arma, pues el disparar tiene un arma tiene más consecuencias más allá de la bala que sale disparada. Cuando un arma es accionada ocurren dos cosas: primero; el arma ejerce una fuerza contra la persona que la ha disparado, la cual es de tener en cuenta, ya que en caso de que el agarre no sea firme puede provocar que el usuario la suelte y en las de mayor calibre llega a causar dislocamientos en las articulaciones del brazo, y lo segundo es que el retroceso desestabiliza el arma o por expresarlo de otra manera “la desapunta”. Estas cosas han de ser tenidas en cuenta por los tiradores profesionales, por lo que es requerido que en sus entrenamientos con réplicas cuando disparen creen exactamente el mismo tipo de sensación que el arma original.

Para conseguir ese efecto, se utilizan en la actualidad, cartuchos de balas que en vez de pólvora se encuentran rellenos de aire a presión o en su defecto de algún tipo de gas licuado, como puede ser el CO₂, que cuando es liberado, el cambio de presión con respecto a la atmosfera hace que se produzca una expansión brusca del fluido causando así una reacción semejante a la del retroceso de un arma.

De esta forma no solo se consigue simular el retroceso del arma, además, cuando se gastan los disparos se obliga al usuario a recargar de la misma forma que un arma real [14].

SIMULADORES DE OCIO

El mercado del entretenimiento electrónico es sinónimo de videojuegos, las tres empresas dominantes del mercado son *Sony*, *Microsoft* y *Nintendo* de las cuales solo las dos compañías japonesas tienen periféricos para sus consolas que se aproximan a la simulación de armamento.

El *PlayStation Move* y el *Wii Mote*, con forma de un prisma rectangular que se adapta a la mano del usuario, constan en su interior de giróscopos y acelerómetros que permiten reconocer el tipo de movimiento que está realizando acompañados de un sensor óptico que permite reconocer a qué punto se está apuntando con el mando [15].

Estos mandos van acompañados de estructuras que simulan la forma de un arma para una mayor comodidad en los juegos FPS, pero puesto que están pensados para movimientos en la mano ninguno está equipado con un sistema de retroceso o desapuntado que le de realismo cuando se trata de disparar con ellos. Sin embargo, gracias a que es uno de los mercados más boyantes de la última década, la calidad de las imágenes llega a un grado de realismo cada vez mayor, con resultados sorprendentes en el diseño de entornos, como en el videojuego *Crysis 3* [16].



SIMULADORES PROFESIONALES

A día de hoy, la empresa dominante en el sector de la simulación realista en el campo tanto bélico como de protección profesional para España es Virtra [14]. Esta empresa americana copa a fecha actual la mayor parte del sector con sus diferentes modelos de simulador, siendo sus principales compradores empresas de formación de guardias de seguridad o cuerpos oficiales del estado.

Los tres modelos que se encuentran en circulación implican diferentes grados de inmersión dependiendo del número de pantallas utilizadas siendo 6 y proporcionando 300º grados de visión con el Virtra V300 (fig. 5), 3 (180º grados de visión, Virtra V180) o una única pantalla (100º grados de visión, Virtra V100), esta última es a la que se pretende aproximar este proyecto [14].

El entorno virtual que utilizan busca parecerse a la realidad lo más posible por lo que no existe diferencia alguna en el tipo de resultado que se busca obtener aquí, más allá de no tener diseñadores y animadores 3D profesionales que modelen todo el entorno.



Figura 5: Simulador Virtra 300 en mitad de una demostración.

Este tipo de simuladores al que se pretende emular en este proyecto tienen costes terriblemente elevados (entre 150K y 300K USD donde lo más habitual es 200K), lo que los aleja del público general ya no solo en su compra, sino también para sesiones de uso individuales alquilándolos (1,5K USD es el precio de una sesión estándar).



La ventaja que ofrece este tipo de simuladores profesionales es la interconexión de los dispositivos permitiendo así el enfrentamiento entre equipos que no tienen por qué estar en la misma área de trabajo para su entrenamiento. Mediante indicadores en sus trajes y el propio funcionamiento del arma, un soldado puede saber si ha sido disparado o no, su tasa de acierto en disparos y su propio rendimiento frente a otros compañeros gracias a la información provista por el propio equipamiento.

Otras empresas del sector son: MiloRange; que trabaja con pantallas distribuidas a lo largo de un edificio en las zonas clave, pero que no simulan un entorno virtual completo, o con proyectores en una pared fija semejantes al Virtra V100, MEGGITT y TITraining que trabajan de la misma manera.

Motivación:

Se pretende acercar de una forma económica y lo más cercana posible a lo que realmente es un simulador profesional para el entrenamiento militar al público general, de forma que pueda ser incorporado a su ocio, para aquellos que lo deseen, desde la comodidad de su casa y usando la menor cantidad de infraestructura posible.

De la misma forma se pretende que el desarrollo sea lo suficientemente cercano al de un simulador profesional como para que sea viable su venta a los cuerpos del estado u empresas de seguridad privada.

Esto permite crear una oportunidad de negocio donde el mercado no se encuentra saturado a nivel de bajo coste y la inclusión de un nuevo elemento no supone un desafío de inversión ni de competidores.

Además, a nivel personal, el proyectista se encuentra interesado por el desarrollo de un dispositivo funcional cuyo uso lúdico pueda ser relativamente común.



Objetivos del Proyecto:

A lo largo de este proyecto se espera obtener al final del mismo:

- Un dispositivo de Hardware que funcione de forma inalámbrica para que con el software adecuado se disponga de un periférico a un computador que sirva para simular un sistema de apuntado y selección de la misma forma que un ratón de teclado.

- Un sistema de reacción háptica lo más real posible que consiga simular el efecto del retroceso de un arma de cualquier calibre implementado sobre una réplica de arma.

- El desarrollo de un entorno virtual similar a lo comúnmente conocido como FPS (First Person Shooter) que no solo sirva para la simulación real de combate, sino que también sea útil como elemento de ocio para el público.

- Demostrar que es viable la creación de un producto funcional de estas características de coste reducido con materiales de bajo precio.

Metodología De Trabajo:

Para completar los objetivos indicados se separará las diferentes partes que implican a este proyecto en diferentes módulos con el fin de unirlos al final y poder así trabajar con diversas unidades independientes que no tengan que depender las unas de las otras para su correcto funcionamiento.

Los diferentes módulos en los que se pretende separar dicho sistema son:

- El diseño físico del arma: Para el proyecto se usará una réplica real de un arma para mayor sensación de realismo, será necesario comparar diferentes modelos de arma que permitan la inclusión en su interior de los componentes electrónicos necesarios, así como el sistema de retroceso.

- El desarrollo del entorno virtual: Para el desarrollo del entorno de simulación se trabajará con diversos motores gráficos que ahora mismo son de uso libre mientras no se realice un uso comercial de ellos, como puede ser el *Unreal Engine 4* o el *Unity*. Será necesario también el poder desarrollar los elementos que vaya a haber dentro del entorno virtual.

- Sistema de Transmisión de Datos: Se pretende usar tecnología inalámbrica para la comunicación del periférico con la computadora por lo que será necesario el diseño de un programa que permita la transmisión de información procesada sobre apuntado y disparo, se usará una red *WiFi* o *Bluetooth* dependiendo de lo que acaben requiriendo las especificaciones.



-Sistema de Retroceso: La acción de disparar un arma conlleva un retroceso que es necesario de su simulación, será necesario desarrollar los componentes hápticos mediante la colocación de vibradores y percutores en distintos puntos del modelo que reaccionen ante el sistema de disparo o la inclusión de un actuador lineal que reproduzca un “golpe” para recrear el retroceso. Este sistema tiene la responsabilidad no solo de simular el retroceso producido por el disparo, también es el responsable de provocar el desapuntado del arma del mismo modo que ocurriría en un arma real.

-Sistema de Disparo: Es requisito diseñar un sistema que permita reconocer la activación del gatillo del arma, para eso se comprobara mediante ensayos que sistema de reconocimiento resulta ser más robusto y resistente a rebotes, malas pulsaciones y permita un tiempo adecuado de reconocimiento ante el retroceso.

-Sistema de Apuntado: Es necesario para la simulación que sobre la pantalla proyectada se sepa a donde se está apuntando para ello se investigaran las tecnologías de detección de movimiento y de localización de posición para comprobar cuál de ellas nos da un mayor resultado y precisión de forma que tengamos las menores limitaciones posibles.

Los módulos de trabajo pueden intercambiar su orden de trabajo en el tiempo según la disponibilidad de los materiales necesarios para su fabricación y los tiempos de envío de los mismos.

Recursos empleados:

Los materiales que se pretende utilizar para este proyecto han de ser todos productos de bajo coste para poder considerar que se han cumplido los objetivos, así que salvo alguna especificación que requiera cierto grado de precisión o rendimiento elevado se tendra a sacrificar a cambio de productos más baratos mientras la funcionalidad no se vea afectada.

Para los programas informáticos que se vayan a utilizar, especialmente aquellos de modelado para el entorno virtual, se escogerá siempre aquellos que tengan licencia académica gratuita, sean de código libre o que permitan su uso no comercial.

Puesto que el objetivo final es la mayor sensación de integración posible, eso se podría realizar con la implantación del sistema virtual y el arma en una cueva de realidad, por lo que si en última instancia resulta ser posible se realizara en la dispuesta por la Universidad Rey Juan Carlos.



CAPITULO:2

ELECCIÓN

DE TECNOLOGÍAS





Selección del modelo del Arma

El objetivo del proyecto es un simulador realista de armas de fuego en una cueva de realidad virtual, motivo por el cual el aspecto del arma que se vaya a utilizar en el proyecto ha de ser lo más parecido al de un arma que se encuentre en uso o al menos haya sido relevante en la historia militar como para considerarse algo icónico y que haya influido en el diseño de las armas actuales para que el disparar con ella no se aleje de la sensación de disparar con un arma moderna.

Para poder contar con un arma realista se descarta automáticamente el contar con un arma real que posteriormente se vaya a vaciar por los problemas legales que conlleva obtener un arma en España y las dificultades que presenta el modificar una estructura metálica en caso de ser necesario para implementar las partes electrónicas que requiera el sistema.

Otra opción descartada desde el primer momento es la posibilidad de realizar la carcasa en una simulación 3D para imprimirla posteriormente en una impresora 3D. Esta opción se descarta principalmente por dos motivos. El primer motivo es el tiempo y técnica necesaria para realizar un diseño lo suficientemente realista de un arma real en un programa de modelado, lo cual no resulta óptimo para la realización del proyecto. Además, no es posible hallar planos de derechos de autor libre lo suficientemente detallados de los que partir a la hora de realizar un diseño en 3D de un arma real. El segundo motivo es el coste y acabado que tendría la carcasa de ser realizada mediante este método. Un diseño en 3D de una pistola funcional ha tenido un coste de 8000\$ [17], lo cual dista mucho de un objetivo *low cost* y si bien se bajase el coste muy difícilmente la calidad de los materiales sería capaz de soportar la percusión constante para simular el retroceso.

Las opciones de armas que se barajan en este proyecto son aquellas que se puedan encontrar a través de réplicas para uso recreativo de airsoft. El airsoft es un deporte basado en la simulación militar utilizando réplicas de armas de fuego llamadas ADL (Arma lúdico-deportivas) que disparan bolas de plástico biodegradable de 6mm de diámetro cuya velocidad y alcance son muy similares a las de las armas reales.

Las armas de airsoft cuentan con una variedad de dispositivos para simular el disparo y el retroceso muy variada como la inyección de aire o el muelle percutor que se tendrán en cuenta a la hora de seleccionar una réplica por su posibilidad de implementación en el arma [18].



Fusil M4

Los fusiles M4 (fig. 6) pertenecen a la familia de los fusiles de asalto, pero que se han versionado en carabina. Fabricados por la empresa Colt son famosos principalmente por ser los fusiles típicos de las fuerzas de asalto especiales de Estados Unidos como pueden ser los SWAT. Es un arma reciente pues lleva solo 20 años en servicio, pero aún se sigue utilizando en la actualidad.

Este tipo de fusil esta principalmente pensado para el combate urbano por lo que revela su potencial en entornos cerrados en las manos de unidades móviles y aéreas [19].



Figura 6: Imagen de un fusil M4

PESO	3.68kg (Descargada) 3.77 (Con cargador)
MUNICION	5.56mm
LONGITUD	757mm (Con culata retraída) 838mm (Con culata extendida)
ALCANCE	500m
CARGADOR	45
PRECIO	49.95€
DISTRIBUIDOR	Tienda Ranger
FABRICANTE	Colt

Tabla 1: Características del Fusil M4.



FUSIL HK G36C

Versión compacta del fusil diseñado por Heckler & Koch a principios de los 90 como sustituto de los HK G3. Este fusil es reglamentario del Bundeswehr alemán y las fuerzas armadas españolas. Una de las principales características es la versatilidad de la misma existiendo una gran variedad de modelos disponibles del mismo arma según sean las necesidades del cuerpo armado que la requiera [20] (fig. 7).



Figura 7: Imagen de una carabina HK G36C

PESO	2.82Kg (Descargada)
MUNICION	5.56mm
LONGITUD	500mm (Con culata retraída) 720mm (Con culata extendida)
ALCANCE	200m a 800m
CARGADOR	30 cartuchos
PRECIO	35.€
DISTRIBUIDOR	Tienda Ranger
FABRICANTE	Heckler & Koch

Tabla 2: Características de la HK G36C.



SUBFUSIL MP7

El MP7 (fig. 8) surge como una petición de la OTAN en 1989 de crear un arma que tuviese la misma portabilidad que una pistola, pero que a la vez tuviese la potencia suficiente para penetrar la armadura corporal que pudiesen llevar los soldados enemigos. Diseñada por Heckler & Koch cuenta con una carcasa realizada con polímeros y se sigue usando en la actualidad como arma de defensa personal de algunos ejércitos [21].



Figura 8: Imagen de un subfusil MP7

PESO	1.8Kg (Descargada) 2kg (Cargada con 40 cartuchos)
MUNICION	4.6mm
LONGITUD	340mm (Con culata retraída) 541mm (Con culata extendida)
ALCANCE	200m
CARGADOR	20 o 40 cartuchos
PRECIO	84€
DISTRIBUIDOR	Airsoft51
FABRICANTE	Heckler & Koch

Tabla 3: Características del Subfusil MP7.



FUSIL AK47

Posiblemente una de las armas más conocidas, el fusil AK47 (fig. 9) se popularizó durante la guerra de Vietnam y día de hoy es el arma más manufacturada. Tiene su origen en la Unión Soviética y fue su fusil oficial hasta su disolución. A día de hoy ya no se fabrica de forma oficial, pero si se hace de manera ilegal y es el arma icónica de los movimientos de guerrilla armados [22] [23].



Figura 9: Imagen de un fusil AK-47

PESO	4.3Kg (Descargada)
MUNICION	7.62mm
LONGITUD	870mm
ALCANCE	443m en semiautomático y 302m en automático
CARGADOR	10,20 o 30 cartuchos
PRECIO	149€
DISTRIBUIDOR	AirsoftItaca
FABRICANTE	Izmash

Tabla 4: Características del AK-47



MOSSBERG 500

La M590 pertenece a la serie de escopetas Mossberg 500 (fig. 10) desarrolladas por la empresa americana Mossberg & Sons caracterizadas por tener oculto su martillo. Esta arma fue desarrollada durante el periodo de la guerra del golfo para ser utilizada por las fuerzas armadas de los Estados Unidos.

Esta arma cuenta en su currículum con haber superado la prueba Mil-Spec 3443-E, una de las más duras a las que se puede someter un arma a plena carga. En la actualidad sigue siendo la única arma de estas especificaciones que la haya soportado [24].



Figura 10: Imagen de una Mossberg 590.

PESO	3.3Kg (Descargada)
MUNICION	18.53mm
LONGITUD	508mm
ALCANCE	40m
CARGADOR	5 u 8 cartuchos
PRECIO	35€
DISTRIBUIDOR	Airsoftlamancha
FABRICANTE	O.F. Mossberg & Sons

Tabla 5: Características de la Mossberg 590.



PISTOLA COLT 1911

Desarrollada en el 1911 la M1911 (fig. 11) es una pistola semiautomática y fue el arma auxiliar del ejército estadounidense hasta el año 1985 que fuese sustituida por la Beretta M9 aunque los SWAT la siguen utilizando como arma personal. Debido a su popularidad aún se sigue produciendo hoy en día y existen varios modelos de la misma arma para uso de civiles.

Se trata del arma que constituyó las bases del principio de retroceso corto, rápidamente copiado por sus coetáneas y que se sigue aplicando en la actualidad. Este proceso apenas se ha rediseñado a lo largo de su vida más allá de una revisión realizada a raíz de su uso en la primera guerra mundial [25].



Figura 11: Imagen de una M1911.

PESO	1.13Kg (Descargada)
MUNICION	11.43mm
LONGITUD	219mm
ALCANCE	50m
CARGADOR	7 cartuchos
PRECIO	29.90€
DISTRIBUIDOR	Tienda Ranger
FABRICANTE	Colt's Manufacturing Company

Tabla 6: Características de la M1911.



ELECCION DE LA TECNOLOGÍA

Una vez presentados los diferentes modelos con sus armas es el momento de ofrecer un criterio que permita realizar la decisión.

Lo primero a lo que se atenderá es al tamaño del arma con la que se debe trabajar. El arma ha de ser capaz de albergar en su interior todas las piezas que constituyan el sistema de retroceso y el microcontrolador del arma. El espacio reducido que ofrece la pistola a pesar de ser un arma tan reconocida por el público general obliga a descartarla. También se descarta la MP7 por tener unas dimensiones semejantes a las de la pistola. Esto se debe a que la mayoría de los componentes electrónicos tienen un tamaño obligatorio asignado que no se puede modificar al encontrarse en sus placas el circuito impreso.

Se atenderá también a la geometría de la propia arma para asegurar que, aunque de dimensiones en su interior pueda haber los cm necesarios, la distribución de las partes fijas permite incorporar las piezas realizando el menor número de modificaciones necesarias. Esto descarta el uso de la M590, pues todo el espacio disponible se encuentra en la parte de la culata ya que el resto de la configuración del arma es demasiado estrecha para incluir cualquier otra pieza.

El factor económico es una sección importante del proyecto por lo que se debe descartar cualquier tipo de réplica que aumente de forma excesiva el coste final del simulador. Los 149€ de la Ak-47 sobrepasan el objetivo de bajo coste pensado para el simulador que debería de rondar los 100€, es por este motivo que se debe descartar el uso de esta réplica.

Se reducen las posibilidades a dos opciones: La HK G36 y la M4. Las características de ambas armas son muy similares, el precio de la M4 es ligeramente mayor a la G36, pero también dispone de más espacio en su interior por lo que la colocación de los componentes electrónicos en su interior es más sencilla.

Atendiendo al modo que estas armas realizan el disparo de balas de airsoft, la M4 utiliza un accionamiento eléctrico y la G36 un muelle de retroceso.

Se elegirá la G36 ya que el muelle constituye un medio más sencillo de trabajar y que puede ser integrado en el diseño del simulador. En el caso de un accionamiento eléctrico los requisitos de este para funcionar pueden ser complicados de cumplir mediante las corrientes que ofrece el microcontrolador que se vaya usar.



Selección del sistema de apuntado:

En este apartado nos centraremos en reconocer que tipo de tecnologías existen en el mercado a un precio relativamente económico, pues no deja de ser una de las finalidades de este proyecto, para poder crear una forma de localización del apuntado que está realizando la réplica de arma que habremos seleccionado. Esto es crear un método en el que se sepa a donde está dirigida la boca del arma creándonos algún tipo de orientación virtual que nos sirva de apoyo. Debe ser una tecnología que transmita una cantidad de información procesable a tiempo real y por lo tanto ha de estar basada en algún sistema de medición que nos dé una medida precisa en poco tiempo.

Reconocimiento de referencias

La tecnología óptica resulta una de las opciones más robustas y precisas gracias a que la identificación de una señal de este tipo no suele presentar problemas dentro de un entorno controlado sin obstáculos pues la detección depende de la resolución del elemento receptor.

Una de las formas más comunes de realizar la identificación de la posición de un objeto a día de hoy es el reconocimiento mediante una referencia previamente preparada y una cámara. Expresándolo de otra manera, se trata de un sistema de reconocimiento de balizas que deben ser reconocibles mediante algún tipo de receptor óptico (Las cámaras son la opción habitual, pero bien es cierto que en aplicaciones más sencillas y de menor precisión se pueden utilizar otros elementos como pueden ser los fotodiodos) que es capaz de detectar la distancia relativa entre ambos. Las balizas de referencia que se utilizan pueden ser activas o pasivas dependiendo de si estas son emisoras de algún tipo de luz o si se limitan a reflejarlo.

Este tipo de sistema de detección solo atiende a dos configuraciones posibles dependiendo de cuál de los elementos del sistema de referencia se mantenga inmóvil [26].

- ***Receptor móvil y baliza fija (“Inside-out systems”):***

En este caso es el sensor el que se coloca en el elemento móvil de nuestro sistema. De esta forma cuando el sensor se desplaza alrededor de la baliza es capaz de detectar el movimiento relativo que se está realizando y se puede calcular la distancia entre ambos.

En el proyecto implicaría colocar la cámara en la punta del arma e identificar la zona de apuntado mediante algún tipo de emisor.



- **Receptor fijo, baliza móvil (“Outside-in systems”):**

Para esta situación es la baliza la que se coloca en el arma de nuestro proyecto para que una cámara fija cerca de la pantalla detecte su posición.

El número de balizas que se deben usar en este sistema de posicionamiento depende del número de dimensiones en las que queremos tener controlado nuestro objeto.

Si quisiésemos localizar a nuestro objeto en un plano (2D) serían necesarias dos fuentes de forma que la distancia entre ambas puede darnos una idea de la distancia a la referencia y la pendiente del segmento que las une nos proporcionaría la inclinación del objeto.

Para realizar un posicionamiento en tres dimensiones necesitamos al menos 3 balizas para que entre ellas se pueda formar un plano y que con dicho plano se pueda obtener la distancia y la inclinación que tienen con respecto a la cámara que se encuentra fija.

Tecnología acústica

Uno de los métodos más antiguos de localización de elementos en el espacio es el sonar, invento que plasma perfectamente el tipo de uso que queremos para nuestro proyecto [27].

La ecolocalización se basa en la emisión de un pulso de ultrasonidos y detectar el tiempo que tarda en volver después de rebotar en las superficies cercanas para saber la distancia que nos separa de ellas [28].

Otra forma de trabajar con el sonido es utilizar una señal lo suficientemente rápida para que el tiempo que tarda en recorrer la misma distancia que el sonido sea despreciable (radiofrecuencia, eléctrica, luminosa...) y así poder avisar al receptor de que se va a enviar una señal sonora para calcular el tiempo que tarda en llegar al mismo desde que se envía el aviso.

Es una tecnología fiable en el sentido de que la velocidad del sonido en medio aéreo es suficiente (340m/s) para que en entornos pequeños (una cueva de realidad virtual tiene un área alrededor de 10m²) [29] con cuerpos grandes que se muevan a una velocidad reducida (el brazo humano de un deportista entrenado llega a moverse a 10m/s) la precisión sea más que suficiente [30].

El principal condicionante de esta forma de localización tridimensional es que es dependiente de las condiciones ambientales que, comparando las condiciones extremas de temperatura, humedad, presión... pueden llegar a dar resultados de funcionamiento considerablemente diferentes entre ellos. Además, la forma de la sala, si esta es más oclusiva, y los elementos del entorno que se pueden interponer en el camino de la onda



afectan de forma excesiva a la señal de retorno de la onda haciendo que se obtengan resultados no fiables.

Tecnología de radiofrecuencias (WiFi, Bluetooth, RFID, Ultra-WideBandwidth)

Los emisores y receptores de ondas electromagnéticas también pueden ser utilizados para la localización de elementos en el espacio, un ejemplo conocido es el GPS [31].

El principal problema de este tipo de tecnología es que son ondas de gran amplitud que son capaces de cubrir distancias inmensas en comparación a la que se trabajará en este proyecto (para un satélite dar una precisión de 3 metros en lo que refiere a un coche en una carretera de varios kilómetros es un hito, pero en nuestro caso estamos trabajando con variaciones de centímetros).

Este tipo de ondas son capaces de atravesar la mayoría de las superficies por lo que retenerlas dentro del espacio donde se trabaja e impedir la interferencia de señales ajenas es demasiado complicado y caro para el caso en cuestión.

Tecnología Mecánica

Basada en el acoplamiento de sensores inerciales, existen en el mercado a un precio muy económico comúnmente llamados Giróscopos y Acelerómetros, esta tecnología no nos da directamente los ángulos si no que mediante el cálculo gracias a la velocidad angular obtenida gracias al giróscopo y a la aceleración lineal que nos proporciona el acelerómetro.

Este aparato se conoce como un elemento de 6DOF ("*6 Degrees Of Freedom*") ya que cada uno de los elementos, acelerómetro y giróscopo, poseen cada uno 3 grados de libertad, uno por eje [32].

El acelerómetro es capaz de medir la aceleración provocada por la gravedad, así que colocado perpendicular o paralelo a ella somos capaces de obtener siempre un punto de referencia desde el que partimos. El problema que presenta el acelerómetro es que es capaz de medir toda aceleración en todo momento por lo que al acelerar el giróscopo sin girarlo este lo detectara como un cambio de rotación.

El giróscopo basa sus medidas en el conocimiento de la posición inicial en ángulos de la que parte, a la que añade la velocidad que detecta que se está moviendo para obtener la que será su posición final. El giróscopo a diferencia del acelerómetro da las medidas con mucha precisión, pero como las obtiene mediante el cálculo estimado, lo normal es que aparezca un pequeño error. Este pequeño error va sumándose hasta el punto en el que el valor real y el obtenido no tienen nada que ver, a este suceso se le conoce como *Drift*.



ELECCIÓN DE LA TECNOLOGÍA

Se han mencionado los tipos de tecnología que resultan viables a este proyecto, pero la implementación de todos los sistemas sería redundante y debido a la falta de precisión lo más normal es que algunos de los sistemas se contradijesen entre ellos, por lo tanto, es de proceder realizar una elección entre los mismos.

La tecnología mediante radiofrecuencias ha de quedar descartada solo por la naturaleza de nuestro proyecto, se trata de un simulador de arma que trabaja a muy poca distancia que al pretender ser de bajo coste para que cualquiera lo pueda disponer donde le plazca no se disponen de los métodos de aislamientos de radiación ajena que existe en el ambiente por lo que aunque llegados el caso la precisión que ofrecen pudiese llegar a servirnos es mejor descartarlas debido al ambiente saturado de redes que existe ahora mismo.

Después se descarta el uso de tecnología ópticas dentro de este proyecto debido a la limitación del rango de movimiento que están poseen. Esto se debe a que la lente de las cámaras posee un rango de visión dentro del cual la detección es efectiva. La posición que esta cámara debe adoptar en un proyecto de estas características es con el usuario apuntando directamente a ella. Esto provoca que en el caso de que un usuario se coloque inmediatamente próxima a ella, pero a uno de sus laterales ella no lo detectará. Para el caso de simular una cobertura en el entorno virtual esto es una limitación del tipo de situaciones que se pueden montar por lo que se desecha el uso de cámaras u otra tecnología semejante.

Eso nos deja con la tecnología acústica y la tecnología inercial, el problema que tienen estas técnicas principalmente es que por separado no son capaces de darnos toda la información que llegamos a necesitar, pero unidas podemos usar la localización mediante ultrasonidos para localizar el arma en cualquier punto de la sala donde nos encontremos y los sensores inerciales del giróscopo nos ofrecen información detallada sobre la posición del arma por lo que juntando ambos conocimientos es viable triangular a donde se está apuntando y desde donde.

Los elementos que han sido seleccionados para realizar las funciones de localización han sido:



-Giróscopo GY-521

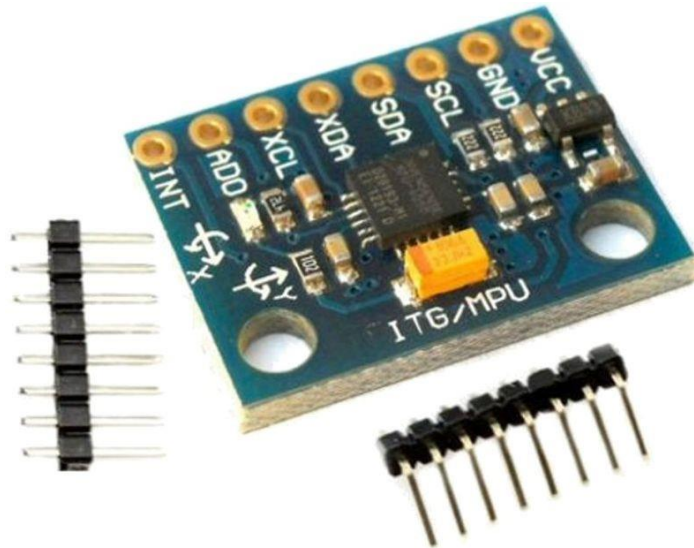


Figura 12: Imagen del Giróscopo GY-521.

Este giróscopo (fig. 12) se encuentra fabricado por la marca HAOYU electronics y tiene un precio muy reducido en el mercado, en el caso del proyecto se ha adquirido por 2.60€ siendo sus especificaciones técnicas [33]:

- Modelo: GY-521
- Color: Azul
- Material: PCB + plástico + cobre
- Chip: MPU-6050
- Fuente de alimentación: 3 ~ 5V
- Modo de comunicación: protocolo de comunicación estándar CII
- Chip integrado convertidor AD de 16 bits, salida de datos de 16 bits
- Los giroscopios van: +/- 250 500 1.000 2.000 grados/seg
- Gama Aceleración: 2 g +/-, +/- 4g, +/- 8 g, +/- 16g
- PCB chapado oro de la inmersión, proceso de soldadura de la máquina para asegurar la calidad
- Lanzamiento Pin: 2.54mm
- 1 x módulo
- 2 x pernos

Dimensiones: 0,83 en x 0,63 en x 0,12 (2,1 x 1,6 cm x 0,3 cm)
Peso: 0,18 oz (5 g)



-Módulo Ultrasónico HC-SR04



Figura 13: Imagen del Módulo Ultrasónico HC-SR04

El módulo (fig. 13) consta de dos partes diferenciadas, un altavoz que emite el pulso ultrasónico que luego recibirá el micrófono, midiendo el tiempo que tarda en ser recibido dicho pulso de vuelta es posible medir rangos de distancia de 2 a 400cm con precisión y sabiendo que el entorno de la cueva virtual es cuadrado y de 320 cm nuestro módulo ofrece un grado de precisión más que suficiente. Es un componente electrónico relativamente caro, 5.29€ en este caso, pero bien es cierto que es un producto habitual en los proyectos con un Arduino, por lo que aporta muchas facilidades al ya existir varias bibliotecas que trabajan con él y lo explotan al máximo, siendo más conveniente que fabricar un sistema de recepción mediante altavoz-micrófono [34].

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

Tabla 7: Características del HC-SR04

En el proyecto será necesarios dos componentes como este, uno que mida la distancia al suelo y otro a la superficie proyectada o pantalla.



Selección del sistema de retroceso

El sistema de retroceso pretende ser uno de los puntos innovadores en este simulador, pues los existentes en el mercado para el ámbito casero no gozan de la realimentación provocada por el disparo que suelen originar las balas. Por ello es necesario encontrar un método que sea accesible económicamente y a su vez no interfiera con el diseño del arma que planeamos usar ni aumente excesivamente su peso convirtiéndolo en algo poco realista.

Vibradores (Motores DC)

Estos pequeños motorcillos (fig. 14) pueden alimentarse con 2.5V y son relativamente sencillos y suelen encontrarse en diversos tipos de juguetes como son los mandos de las videoconsolas [35].



Figura 14: Imagen de un motor extraído DC extraído de un mando DUALSHOCK.

Basan su funcionamiento en una rueda excéntrica que al alimentarse gira a altas velocidades generando ese traqueteo.

El principal inconveniente de este tipo de aparatos es que son muy pequeños y la fuerza que transmiten es muy pequeña por lo que en realidad es necesaria la instalación de varios de ellos para dar una sensación real de fuerza y que en cualquier caso no sería controlada en una dirección como el retroceso si no una vibración como su nombre indica.

Actuadores lineares.

Los actuadores lineares (fig. 15) son en realidad instrumentos que realizan un movimiento de percusión constante en la dirección que hayan sido colocados [36].

Este tipo de actuador es capaz de liberar una gran cantidad de newtons dando realmente la sensación de retroceso del arma ya que además podemos concentrarlo en puntos concretos como la culata del arma que es donde realmente debe notarlo el usuario en el agarre.



Los problemas que presentan es que dicha fuerza no se puede controlar en un mínimo por lo que el retroceso que nos ofrecen es bastante mayor del que necesitamos en la mayoría de los casos, luego resulta inviable dar el voltaje que necesitan pues se trata de un mínimo de 12V y es complicado encontrar una fuente que nos ofrezca dicha tensión en el espacio reducido que poseemos ya que el propio actuador por sus dimensiones es probable que no quepa dentro del arma.



Figura 15: Imagen de un actuador lineal.

Pistones Neumáticos

Basados en la acción de la diferencia de presión de fluidos y la fuerza que estos liberan al descomprimirse son de las tecnologías más antiguas que nos podemos encontrar y su nivel de precisión para lo que lo pretendemos utilizar es más que suficiente.

Existen dos tipos de actuadores neumáticos [37], de efecto simple (fig. 16) y de acción doble. En el primer tipo se pretende usar una fuerza hidráulica para empujar y el pistón y una diferente para hacerlo retroceder, en el segundo caso es una fuerza hidráulica la que se ocupa tanto de empujarlo como de retraerlo por lo que es necesaria la implementación de un solenoide que nos permita controlar la dirección del empuje. En su interior poseen un resorte que es capaz de cambiar su constante elástica dependiendo del paso de corriente que está recibiendo.



Figura 16: Imagen de un pistón neumático.

Estos productos pueden llegar a darnos 1Mpa lo que es una presión considerable, sin embargo, nos encontramos con que presentan problemas debido a sus dimensiones y a su peso además de que no suelen bajar de los 30€ lo cual es un inconveniente, pues encarece bastante nuestro producto.



Muelles de retroceso

Las réplicas de armas suelen venir equipadas con un sistema de retroceso [18] ya que al usarse para *airsoft* al disparar las balas, aunque sean de plástico se origina algún tipo de retroceso. Este retroceso suele generarse por el tipo de sistema de disparo, en nuestro caso nos fijaremos en el que se basa en el accionamiento de un muelle (fig. 17).

Los muelles que tienen este tipo de réplicas ofrecen una potencia de disparo considerable, pues el disparo tiene que semejar el rango real de un arma del calibre correspondiente.

El problema que presenta este mecanismo es la rigidez con la que está implantado en él diseño interno de la propia arma lo cual implica que es necesario modificar la arquitectura interior de la réplica para conseguir que la recarga y el disparo se vean registrados dentro del sistema electrónico.

Añadir también que el muelle requiere de una recarga manual si es que queremos volver a tener retroceso en el disparo posterior.



Figura 17: Imagen de diversos muelles comercializados para las armas de airsoft.



ELECCIÓN DE LA TECNOLOGÍA

Al igual que en el sistema de disparo es necesario utilizar la implementación de dos sistemas si lo que se pretende obtener es un resultado más realista y fiable.

Se desecha la idea de la utilización de un actuador lineal. En primer lugar, su excesivo coste disuade sobre su posible uso ya que, puesto que se pretende diseñar un simulador de bajo coste, el utilizar una tecnología que en el peor de los casos puede valer tanto como el resto de los componentes de cara a obtener un buen producto no es adecuado con la idea de bajo coste.

El hecho de que tengan una demanda tan alta de energía también disuade del uso de este tipo de accionamientos, pues implica la necesidad de colocar un mayor número de fuentes de alimentación para poder suministrar la energía de funcionamiento necesaria lo cual complica el diseño interno del arma debido al reducido espacio con el que se cuenta dentro de la misma.

No se usarán tampoco los pistones neumáticos ya que la implementación de los mismos cuenta con muchas dificultades por las condiciones de funcionamiento que necesitan para poder inyectar el gas como corresponde. Además, sus dimensiones hacen que sea necesario un espacio que no se encuentra disponible dentro del arma que se utilice.

Esto implica que se utilizaran las otras dos tecnologías que se han mencionado con anterioridad, los motores vibradores y el muelle de la propia arma.

Estos sistemas no funcionarían a la vez si no que se pretende configurar dos modos de armas. Uno que simule el modo de ráfaga rápida más propio de un subfusil gracias a los vibradores ya que ante tal cantidad de disparos (750 por minuto en lo que implica a una carabina estándar) [20] ninguna de las otras opciones ofrece una repetición de la fuerza lo suficientemente rápida como para simular de forma real la cadencia y otro que esté basado en el uso del muelle que al dar un golpe más brusco puede simular el cambio a un arma de mayor calibre, pero con menor cadencia de disparo.



Selección del microprocesador

El elemento central alrededor del que van a girar las demás partes del sistema del arma es aquel que sea capaz de realizar las funciones lógicas relacionadas con los procedimientos de apuntado, disparo, comunicación y retroceso. Esto requiere de una potencia de computación que poseen los microprocesadores comerciales que se han popularizado en los últimos años gracias a las modas del DIY (“Do It Yourself”, Hágalo usted mismo).

Afortunadamente este tipo de productos son de muy poco peso y reducido tamaño que permite implementarlos dentro de casi cualquier arma. Además, al contar con un tamaño tan pequeño, su demanda de energía es pareja y así es viable alimentarlos con fuentes de alimentación químicas portátiles como pilas y semejantes solucionado el problema de tener que usar fuentes reguladoras de tensión o conectarlo a la red manteniendo la condición de inalámbrico en nuestro proyecto.

Una gran ventaja que tienen este tipo de dispositivos es el hecho de que mantienen la programación que le es implementada incluso si se les corta la alimentación de la que dependen. Esto es necesario para la consecución de este proyecto, pues al encontrarse toda la parte de hardware contenida dentro del arma no es viable desmontarla y cargar la programación que vayamos utilizar dentro del controlador cada vez que se pretenda utilizar la misma. Además de esta forma se garantiza la autonomía del proyecto sabiendo que para que este funcione solo es necesario tener una fuente de alimentación en el proyecto para que este funcione.

El tipo de procesador que se vaya a utilizar está condicionado por la posterior conexión del software que utilice el micro con el del motor gráfico de videojuego que se vaya a utilizar, pues una mala elección puede conllevar la necesidad de utilizar software intermedios, como puede ser por ejemplo Processing, que se ocupen de recibir los valores por parte del motor gráfico y del microprocesador, traducirlos y enviarlos al contrario. Si se trabaja a nivel profesional esto requiere de técnicos con mayor formación que puedan gestionar al menos dos de los tres softwares o incluso la necesidad de implementar trabajadores extra solo para esa etapa del desarrollo. En el caso de este proyecto requiere de aprender el lenguaje necesario para poder realizar dicha comunicación lo cual complica su consecución alargando plazos y dificultando su realización.



FAMILIA ARDUINO

Los Arduino son una familia de microprocesadores que goza de la hegemonía en el mercado de las DIY gracias a su accesibilidad y bajo coste. Dentro de esta misma familia se encuentran componentes con distintas características que permiten la elección dentro de una amplia gama de productos para hallar aquel se ajuste más a las necesidades del proyecto.

La compañía trabaja el hardware libre por lo que los planos y esquemáticos necesarios sobre la placa se encuentran disponibles para todo el mundo que deseen utilizarlos abriendo un gran número de puertas para aquellos que lo necesiten. Las placas integran un microcontrolador de tipo Atmel AVR de forma habitual, así como los puertos digitales y analógicos necesarios para la interacción de la placa que además pueden ampliarse gracias a las placas de expansión disponibles en el mercado como por ejemplo los módulos de conexión ZigBee.

El software está basado en el entorno de Processing y el lenguaje de programación tanto como el bootloader vienen a ser una versión de C++ basado en Wiring. La configuración de los mismos se realiza por comunicación en serie gracias a un convertidor de niveles RS-232 a TTL serial [38].

A continuación, se podrá ver una tabla (Tabla 8) con todos los modelos actuales de Arduino en el mercado. Se descartarán aquellos cuyas dimensiones sean excesivas para el arma que se utilice en el proyecto, aquellas cuya especificación de tensión sea demasiado alta debido a las dificultades para alimentarlas correctamente y aquellas cuya especificación no llegue a los 5V, pues es la demanda estándar de los productos que se han seleccionado en los puntos anteriores.

También ha de tenerse en cuenta que estos posean una conexión USB por la cual se posible realizar la programación del mismo ya que si no es así será necesaria la aparición de productos que permitan llevarla a cabo. Ampliando de esta manera la cantidad de hardware que necesitaremos para poder llevar a cabo esta tarea lo cual conlleva un aumento del coste del procedimiento y mayores dificultades para poder llevarlo a cabo.

Se tendrá en cuenta también el ámbito de lo económico y disponibilidad para realizar la elección del microcontrolador.



Name	Processor	Operating/Input Voltage	CPU Speed	Analog In/Out	Digital IO/PWM	EEPROM [kB]	SRAM [kB]	Flash [kB]	USB	UART
101	Intel® Curie	3.3 V / 7-12V	32MHz	6/0	14/4	-	24	196	Regular	-
Due	ATSAM3X8E	3.3 V / 7-12 V	84 MHz	12/2	54/12	-	96	512	2 Micro	4
Gemma	ATtiny85	3.3 V / 4-16 V	8 MHz	1/0	3/2	0.5	0.5	8	Micro	0
LilyPad	ATmega168V ATmega328P	2.7-5.5 V / 2.7-5.5 V	8MHz	6/0	14/6	0.512	1	16	-	-
LilyPad SimpleSnap	ATmega328P	2.7-5.5 V / 2.7-5.5 V	8 MHz	4/0	9/4	1	2	32	-	-
LilyPad USB	ATmega32U4	3.3 V / 3.8-5 V	8 MHz	4/0	9/4	1	2.5	32	Micro	-
Mega 2560	ATmega2560	5 V / 7-12 V	16 MHz	16/0	54/15	4	8	256	Regular	4
Mega ADK	ATmega2560	5 V / 7-12 V	16 MHz	16/0	54/15	4	8	256	Regular	4
Micro	ATmega32U4	5 V / 7-12 V	16 MHz	12/0	20/7	1	2.5	32	Micro	1
MKR1000	SAMD21 Cortex-M0+	3.3 V / 5V	48MHz	7/1	8/4	-	32	256	Micro	1
Nano	ATmega168 ATmega328P	5 V / 7-9 V	16 MHz	8/0	14/6	0.512 1	1 2	16 32	Mini	1
Pro	ATmega168 ATmega328P	3.3 V / 3.35-12 V 5 V / 5-12 V	8 MHz 16 MHz	6/0	14/6	0.512 1	1 2	16 32	-	1
Pro Mini	ATmega328P	3.3 V / 3.35-12 V 5 V / 5-12 V	8 MHz 16 MHz	6/0	14/6	1	1	32	-	1
Uno	ATmega328P	5 V / 7-12 V	16 MHz	6/0	14/6	1	2	32	Regular	1
Yún	ATmega32U4 AR9331 Linux	5 V	16 MHz 400MHz	12/0	20/7	1	2.5 16MB	32 64MB	Micro	1
Zero	ATSAMD21C18	3.3 V / 7-12 V	48 MHz	6/1	14/10	-	32	256	2 Micro	2

Tabla 8: comparativa de los modelos de Arduino que se encuentran actualmente en el mercado.



RASPBERRY PI

La *Raspberry Pi* es un conocido ordenador reducido de bajo coste diseñado por la fundación inglesa de nombre homónimo con finalidades didácticas en las escuelas. Cuenta con propiedad registrada, pero su uso es libre de forma que es posible para un particular el hacer uso de esta tecnología.

Al tratarse de un miniordenador es necesario utilizar un sistema operativo para gestionarlo y aunque la empresa distribuidora cuenta con una versión *Open Source* del que ellos mismos utilizan; *Raspbian*, una adaptación del *Debian*, es posible utilizar otros SO del mercado, como es por ejemplo el Windows 10 [39].

A continuación, se observa una tabla (Tabla 9) con las principales características de los diferentes modelos de *Raspberry* obtenida de la página oficial con el fin de poder compararlas con nuestras necesidades.

				
Raspberry Pi:	Model A+	Model B	Model B+	2, Model B
Price:	\$19.99	\$39.99	\$29.99	\$39.99
Availability:	Add to Cart	Add to Cart	Add to Cart	Add to Cart
Quick summary:	Cheapest, smallest single board computer.	The original Raspberry Pi.	More USB and GPIO than the B. Ideal choice for schools	Newest, most advanced Raspberry Pi.
Chip:	Broadcom BCM2835			Broadcom BCM2836
Processor:	ARMv6 single core			ARMv7 quad core
Processor Speed:	700 MHz			900 MHz
Voltage and Power Draw:	600mA @ 5V			650mA @ 5V



GPU:	Dual Core VideoCore IV Multimedia Co-Processor			
Size:	65x56mm	85x56mm		
Memory:	256 MB SDRAM @ 400 MHz	512 MB SDRAM @ 400 MHz		1 GB SDRAM @ 400 MHz
Storage:	Micro SD Card (not included)	SD Card (not included)	Micro SD Card (included)	Micro SD Card (not included)
GPIO:	40	26	40	
USB 2.0:	1	2	4	
Ethernet:	None	10/100mb Ethernet RJ45 Jack		
Audio:	Multi-Channel HD Audio over HDMI, Analog Stereo from 3.5mm Headphone Jack			

Tabla 9: Comparativa de las especificaciones de los distintos modelos de Raspberry Pi en el Mercado.

Se observa que el rango de precios en el que se mueven nuestras tarjetas es lo suficientemente accesible sabiendo que este se trata de un proyecto de bajo coste y que el elemento más caro del mismo es este elemento junto al modelo de arma que se vaya a utilizar.

La potencia de los mismos en lo que a especificaciones técnicas es más que correcta ya que se trata de frecuencias de trabajo más que suficientes para el proyecto y la memoria con la que se va a trabajar apenas llegará al MB.

PIC (Peripheral Interface Controller)

Los PIC son una opción muy simple de microcontrolador construida por la empresa Microchip Technology. Son opciones que funcionan a 32, 16 u 8 bits dependiendo de la familia del mismo que se ven ejecutadas cada 4 tiempos del oscilador (Equivalente a 1 tiempo de ejecución).

Este tipo de controlador funciona en un rango de entre 4 y 10MHz y gracias a que en su interior se le vio implementada una EEPROM (memoria de tipo no volátil) es viable el mantener un programa en su interior sin necesidad de conectarla a un ordenador de forma permanente y así darle autonomía a la misma.

El consumo del mismo es muy reducido ya que proporcionan corrientes de entre 20mA y 25mA por lo que la duración en régimen de funcionamiento se garantiza de larga duración dentro de las opciones trabajadas.

Es necesario la creación de una placa base donde implementar el mismo e instanciar los componentes en una protoboard. Esto también se requiere para la programación del mismo pues es necesario establecer comunicación con el computador por medio de adaptadores PickIT.[40]



ELECCION DE LA TECNOLOGÍA

Lo primero que se descartará serán los PIC, si bien es cierto que son la opción más barata dentro de las que se han ofrecido (Rondan los 10€ de precio como mucho) tienen una funcionalidad muy reducida debido al estricto modo de trabajo que tienen ya que el juego de instrucciones que poseen es lo que realmente determina todo lo que se puede hacer con ellas y al tener que trabajar con un software tan específico como es un motor gráfico no se cuenta con los medios suficientes para trabajar de forma cómoda con ellos.

Además, el hecho de utilizar un PIC conlleva de montar la placa asociado a las comunicaciones del mismo con el resto de dispositivos, lo que aumenta el tiempo requerido de trabajo. Atendiendo también a la comodidad de programación, también es necesario soldar y preparar las comunicaciones del PIC con el ordenador para poder programarlo. Esto añade un grado de carga de trabajo innecesario.

Después se descartan los ordenadores reducidos Raspberry. El motivo es que al ser estos unos micrordenadores, el consumo energético que tienen y la demanda de intensidad es demasiado alta para poder considerarla un régimen de trabajo viable si la intención es que el tiempo de funcionamiento sea lo suficientemente alto para poder operar varias horas con el simulador sin tener que abrir el mismo para cambiar las baterías.

Mencionados los motivos por los cuales se descartan las otras tecnologías es necesario elegir uno de los modelos expuestos de Arduino.

La opción seleccionada será el modelo Arduino Uno. Los motivos que llevan a elegir este modelo en concreto tienen que ver con los criterios previamente expuestos: ha de tener una entrada USB que permita programar cómodamente el microcontrolador y un precio y una disponibilidad asequible siendo el Uno el que cumple mejor estas características (De los 4 que tienen una conexión USB regular, el UNO es 10€ más barato que el resto como mínimo).

Añadir también que la popularidad creciente de este producto hace que existan ahora mismo una gran variedad de librerías y periféricos orientados al mismo, facilitando así el trabajo con los demás elementos del proyecto, como pueden ser el módulo de comunicación *Bluetooth* o el sistema de ultrasonidos, ya que ambos cuentan con librerías de trabajo orientadas propias.



Selección del sistema de disparo:

Aquello a lo que se refiere el sistema de disparo es a aquella parte del diseño que cumpla la funcionalidad de indicar cuando se está apretando el gatillo de la réplica. El registro del disparo resulta la parte más fundamental de todo el diseño ya que es inútil un arma que después de todo, no dispara. Este módulo debe ocuparse de actualizar el valor de la entrada de Arduino para que este pueda registrar que se ha producido el disparo.

El esquema de este circuito resulta ser especialmente sencillo al simplemente necesitar una señal 0 (circuito abierto) cuando no se está pulsando el gatillo y una señal 1 (circuito cerrado) cuando se ha apretado el gatillo y se supone que se está ejecutando el disparo.

PULSADOR

Los pulsadores (fig. 18) son la opción más barata y sencilla para abrir y cerrar un circuito eléctrico con la posibilidad de obtener uno normalmente abierto o normalmente cerrado dependiendo de cómo se pretenda que funcione la lógica.



Figura 18: Imagen de un pulsador

FIN DE CARRERA

Este tipo de sensores (fig. 19) se identifican en cierta forma con los pulsadores ya que se limitan a indicar cuando se ha recorrido su totalidad la carrera un componente mecánico. Constan de dos posiciones de forma que con un mismo fin de carrera es posible obtener una configuración normalmente abierta y una configuración normalmente cerrada.



Figura 19: Imagen de un fin de carrera.

Son bastante más precisos si los comparamos con los pulsadores en lo que a rebotes se refiere.

SENSOR CAPACITIVO

También conocidos como sensores de proximidad, los sensores capacitivos (fig.20) basan su funcionamiento en la variación del campo eléctrico. Estos sensores son capaces de percibir cuando un objeto metálico o no metálico está realizando una interferencia con su campo magnético y de que intensidad, de esta forma es capaz de percibir si un objeto se encuentra cerca o lejos de él [41].

Este podría usarse para saber cuándo el gatillo se ha acercado a la posición de disparo.



Figuras 20 y 21: Imagen de un sensor capacitivo (20) y de un sensor flexible resistivo (21).

SENSOR DE FLEXIBILIDAD RESISTIVO

Los sensores de flexibilidad resistivos (fig. 21) basan su funcionamiento en la variación de la resistencia que posee según la presión aplicada en su superficie. Están pensados con una cara adhesiva para poder facilitar su colocación, pero gozan del inconveniente de ser poco exactos cuando se está tratando con muestras parecidas.

FOTORESISTENCIA

Una LDR, *light-dependent resistor* (Resistencia dependiente de la luz. Traducción literal del inglés) (fig. 22), es un tipo de resistencia cuyo valor varía en función de la luz que les está incidiendo [42].

Esta resistencia podría detectar el disparo si se coloca de tal forma que cuando el gatillo sea apretado la luz que le llegue se vea afectada de forma considerable y así se pueda considerar que un disparo ha sido efectuado.

Destacar que esta resistencia no da valores digitales sino analógicos por lo que el pin con el que se ha de trabajar debe ser analógico.

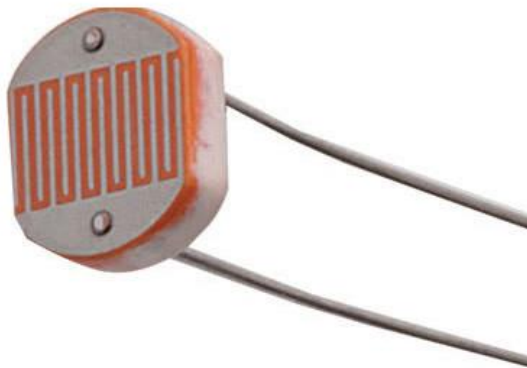


Figura 22: Imagen de una fotoresistencia.



ELECCIÓN DE LA TECNOLOGÍA

Basándose en el diseño interno del arma una vez esta está abierta es viable instalar simplemente un dispositivo de pulsación con algún tipo de filtro que permita obviar los rebotes y un detector de flanco dentro de nuestro programa para asegurarnos que la señal recibida es correcta.

Para esto podemos decidirnos entre el pulsador normal y el de fin de carrera, puesto que el rango económico en el que se mueven es el mismo, lo más lógico será decantarse por aquel que ofrezca una mayor facilidad de implementación dentro del diseño del arma, siendo esto más sencillo con el pulsador.

Descartamos el fotorresistor porque al trabajar con un elemento que va a ser móvil, no sabemos el tipo de sombras que se le pueden llegar a proyectar por lo que resulta complicado ser capaces de definir un rango de trabajo para el gatillo donde se ofrezca un resultado viable.

Los demás sensores expuestos también nos pueden dar un buen resultado, pero son considerablemente más caros que el pulsador por lo que para respetar el objetivo de bajo coste, los desecharemos sabiendo que el pulsador es más que suficiente.



Selección del sistema de comunicación arma-simulador:

Se ha de establecer un sistema que permita el envío de información desde el microprocesador del arma, que se encuentra realizando el apuntado virtual y procesando la información de los disparos, al ordenador donde se está llevando a cabo la proyección de la simulación.

Para ello, la primera de las condiciones que se ha de cumplir es que esta comunicación se realice de forma inalámbrica ya que la existencia de un cable conectado al arma conlleva una reducción del rango de movimiento, así como la posibilidad de enredos e incomodidades durante el trabajo con el arma por lo que queda estipulada la necesidad imperiosa de que el periférico no ha de estar físicamente conectado a nada.

La segunda es una tasa de transmisión estable y lo suficientemente rápida para poder actualizar la posición de la mira según a donde esta esté apuntando en un tiempo para el cual el ojo humano no note el denominado *lag* [12]. Esto impide el uso de tecnología de infrarrojos ya que el envío de datos depende de estar localizados en una posición estable donde se permita la comunicación ininterrumpida y en este caso el arma al encontrarse en movimiento durante la mayor parte de su periodo de funcionamiento no es posible establecer un flujo constante.

BLUETOOTH

El *Bluetooth* es una especificación industrial pensada para el uso de redes inalámbricas de uso personal para posibilitar el envío de datos y voz en un enlace por radiofrecuencia en la banda de los 2.4GHz

Esta tecnología es de uso habitual en las telecomunicaciones y la informática personal, incluyendo los *smartphones*, impresoras, ordenadores personales de sobremesa o portátiles por introducir unos cuantos ejemplos habituales en el día a día.

Saber que es una tecnología de comunicación tan fácil de encontrar en los ordenadores de hoy en día es un aliciente porque nos permite aumentar el margen de público al que puede alcanzar nuestro simulador sabiendo que no es necesario que la gente añada periféricos a su ordenador para poder trabajar con el aumentando el coste del mismo.

Los transceptores de *Bluetooth* tienen la ventaja de que requieren de muy poco consumo eléctrico para un rango efectivo más que suficiente (un dispositivo *Bluetooth* de clase 2 ofrece un alcance de 5 a 10 metros con una potencia máxima permitida de 2.5mW) para el espacio con el que se trabaja dentro de este proyecto [43].



WIFI

El wifi o Wi-Fi es la forma más popular en la actualidad para conectar de forma inalámbrica los distintos dispositivos del hogar, ya sean ordenadores y teléfonos móviles o televisores y calefacción. Esta conexión se realiza mediante un punto de acceso de red inalámbrica.

En ese sentido presenta la misma ventaja que el *Bluetooth* sabiendo que la mayoría de los dispositivos serán compatibles con el mismo y por lo tanto no es necesario trabajar con algún tipo de adaptador externo para los ordenadores en los que se quiera trabajar con el simulador que se está diseñando.

Las redes wifi están basadas en el estándar IEEE 802.11, tienen un alcance que ronda los 20 metros en interiores y usa frecuencias a 2.4 y 5 GHz cubriendo sin problemas los requisitos con los que se trabaja en este proyecto.

Tiene la ventaja de poder trabajar en una banda de trabajo limpia que es la de 5 GHz ya que ha sido recientemente habilitada y no existe ningún otro tipo de tecnología que se le superponga en ese sentido y aunque el alcance a esta frecuencia es menor, es útil para este caso ya que la separación entre elementos no llega a la decena de metros [44].

ZIGBEE

El ZigBee se comunica en las bandas reservadas para uso ISM (Industrial, Scientific and Medical) cuyo uso no está controlado en el rango de 2.4 GHz y tiene la posibilidad de comunicarse en un rango de 10 a hasta 100 metros en interiores. Está basada en el estándar IEEE 802.15.4 sobre redes inalámbricas de área personal.

Existe la posibilidad de comprar módulos de comunicación ZigBee para Arduino y los ordenadores en los que se vaya a utilizar el simulador contando con el inconveniente de que es algo cuya instalación es necesaria para ambos dispositivos [45].



ELECCION DE LA TECNOLOGÍA

Para la funcionalidad que se pretende tener en el proyecto, cualquiera de las tecnologías que se han mencionado arriba son más que suficientes para cubrir todas las especificaciones. Por lo tanto, el factor decisivo de las mismas será el componente económico que es uno de los puntos a tener en cuenta en nuestro diseño.

- El módulo ESP8266 para conexión Wi-Fi de Arduino tiene un precio de 5.23€. [46]
- El XbeeShield de Arduino para la conexión Zigbee tiene un precio de 18€. [47]
- El HC-05 de Arduino para establecer conexión mediante *Bluetooth* tiene un precio de 4.9€. [48]

Siendo necesarios dos elementos de conexión para realizar una red ZigBee y siendo estos a la vez los que más dinero cuestan están automáticamente descartados del proyecto.

La diferencia económica entre el HC-05 de comunicación *Bluetooth* y el ESP8266 de comunicación Wi-Fi es exigua por lo tanto tendremos que fijarnos algo más en las características de las mismas.

El Wi-Fi presenta un mayor rango de actuación en comparación al del *Bluetooth* ya que este tiene un alcance estándar de 10 metros mientras que el Wi-Fi tiene un alcance de 20 metros. En cualquier caso, la distancia en la que el proyecto se pretende desplazar es de 1-3 metros por lo que es una característica que no nos sirve a la hora de discernir entre ambos.

El ESP8266 tiene un input voltaje de 3 a 3.6V con un consumo estándar de corriente de 80mA lo cual conlleva a un valor típico de 0.288W.

El HC-05 tiene un input voltaje de 1.8, 3.3 o 5V con un consumo típico de corriente de 40mA lo que implica que el consumo de potencia habitual rondará los 0.132W.

Si se presta atención al consumo energético y por lo tanto se pretende alargar la vida del producto a un menor coste lo que se hará entonces es utilizar el módulo de comunicación *Bluetooth*, esta idea se ve reforzada también por el hecho de que es el más barato de ambos productos.



CAPÍTULO:3

ELECCIÓN DEL SOFTWARE





Uno de los elementos claves es el entorno virtual en el que se sumergirá el usuario con el fin de integrarle en un mundo distinto con el que practicar puntería con el arma que posee entre sus brazos. El diseño de este elemento ha de realizarse mediante algún tipo de software que permita crear entornos en 3D que se *rendericen* a tiempo real con el fin de que respondan ante las modificaciones en el entorno que provoque el uso del arma en la simulación [49].

Actualmente en el mercado existe un tipo de producto denominado motor gráfico que es el software con el que se realizan las producciones audiovisuales interactivas. Este tipo de software provee de un entorno en el que localizar elementos modelados en 3D de otros programas como Maya o Blender y trabajar con ellos y sus animaciones a través de código de programación. A los modelados 3D se les provee de unas características propias de objetos reales como la detección de colisiones con otros elementos del entorno, sonidos, música, conexión entre elementos y una inteligencia artificial.

Lo que caracteriza a un buen motor gráfico es la facilidad para plasmar de forma gráfica las ideas con las que trabaje un desarrollador y su acabado posterior.

Esta tecnología se presenta con el fin de comprender sus bondades más allá de un videojuego. Un motor gráfico permite desarrollar un entorno virtual de forma totalmente libre gracias a las herramientas que provee. Gracias a los medios provistos por el motor no solo es posible desarrollar un videojuego, sino también un entorno no totalmente realista debido a la existencia del *Uncanny Valley* presentado en la introducción de este proyecto, pero si lo suficientemente bueno como para que al usuario le permita sentirse inmerso y percibir un grado de realidad lo suficientemente alto. A día de hoy los videojuegos son capaces de presentar texturas y modelos hiperrealistas que rozan la naturalidad propia de los elementos reales, un ejemplo de esto sería el videojuego *Detroit Become Human* (fig. 23), pendiente de publicación, pero del que se han liberado los *trailers* y modelos de la protagonista pudiéndose comprobar el acabado que presenta.

Puesto que los videojuegos son capaces de trabajar con distintos tipos de periféricos (joysticks, mandos, conjuntos de ratón y teclado, etc) y que estos no guardan siempre la misma configuración de botones dependiendo del fabricante, los motores gráficos presentan un modelo de configuración de los controles del videojuego totalmente abierto a los que se pueden añadir más métodos que los existentes en el propio motor. La libertad de configuración, permite al desarrollador crear su propia forma de trabajar con un periférico mientras sea posible la comunicación de este con el motor gráfico mediante algún tipo de *plugin*.



Figura 23: Imagen promocional del videojuego *Detroit Become Human*.

Toda esta libertad indica que mediante el uso de un motor gráfico es posible realizar el desarrollo del entorno realista que se busca en el simulador de este proyecto por lo que se elegirá uno de los motores más comunes que tengan una opción de desarrollo gratuita.

UNREAL ENGINE 4

Unreal Engine 4 es el sucesor del motor gráfico creado por la compañía *Epic Games* y que es el sucesor del *Unreal Development Kit*. Se considera este motor gráfico para el proyecto porque a pesar de no ser realmente libre, la condición de pago a la compañía por su uso es de un 5% de los beneficios trimestrales, siempre y cuando estos superen los 3000\$. Puesto que el planteamiento de comercializar el producto vendrá después de la comprobación de la fiabilidad que ofrece el prototipo del arma, al no obtener beneficio durante su desarrollo es factible utilizar este motor [50].

Unreal Engine 4 es el motor más popular del mercado en el momento de publicación de este proyecto, más de 100 juegos se han realizado o se van a realizar con este mismo motor o variaciones del mismo gracias a las posibilidades de modificación y reconfiguración que este ofrece. La categoría de estos juegos pasa desde la etiqueta *indie* propia de productos creados por desarrolladoras pequeñas y presupuestos reducidos a la de juegos catalogados como triple A por el presupuesto propio de una película de cine, pues 100 millones es lo que se sospecha que le ha costado a *Epic Games* la realización del *Gears Of War 4* [51]. Además, este motor no se está utilizando solo como motor de videojuego normal, sino que también se está implementando para demos técnicas de funcionamiento en *Oculus Rift*. Esto demuestra que es un motor que es amigable con tecnología de realidad virtual [52].



Unreal tiene un motor con un módulo que permite trabajar en 2D como en 3D lo que implica que el tipo de desarrollo que se va a realizar no se encuentra limitado por la movilidad que ha de tener el producto.

El acabado y trabajado de la iluminación dinámica de *Unreal* son conocidos por ser los mejores del mercado y el acabado gráfico que presentan da una sensación de hiperrealismo gracias a la posibilidad de implementación de diversas técnicas como es la fotogrametría en juegos (fig. 24).



Figura 24: Imagen en 4k realizada por el artista Rense de Boer en Unreal Engine 4.

Una de las características más destacables en el *Unreal Engine 4* es su lenguaje de programación. *Unreal* está preparado para realizar el scripting de un proyecto mediante programación en C++. Sin embargo, cuenta con propio lenguaje de programación visual creado específicamente para este programa denominado *Blueprint*. *Blueprint* es un lenguaje de programación secuencial centrado en eventos que se controla mediante la interconexión de bloques con funciones y variables asignadas. Es posible realizar la creación de nuevos bloques para *Blueprint* si se cuenta con los conocimientos de C++ adecuados para poder desarrollarlos.

La gran ventaja que ofrece este nuevo método de programación es que se presenta más intuitivo y visual que el clásico scripting. Esto se debe a que las distintas funciones se encuentran realmente conectadas mediante cables por lo que es posible “ver” cuál es la secuencia lógica de acciones que realizará un objeto o cómo se comportará una variable.

La posibilidad de trabajar con un lenguaje de programación amigable con una persona que no está acostumbrada a trabajar con él, está aprendiendo o nunca ha recibido una formación en programación es uno de los motivos por los que muchos pequeños desarrolladores están usando *Unreal* como motor gráfico al carecer de los medios para



contratar un equipo totalmente especializado con las funciones que necesitan sus proyectos.

Epic Games pone a disposición de sus usuarios un *Marketplace* donde la compañía y usuarios ofrecen a la venta los productos que han creado para el motor gráfico. Esto incluye *Blueprints*, modelos 3D, texturas, materiales, etc.

Disponer de un mercado integrado de los *assets* que se pueden necesitar en un proyecto permite cubrir con materiales de terceros aquellos elementos que no sea viable realizar para la culminación de un proyecto por falta de técnicas o tiempo.

En la actualidad una muestra de los juegos (fig. 25) [52] que utilizan *Unreal Engine* y su acabado son:

- Kingdom Hearts 3.
- Paragon.
- Vampyr.
- Tekken 7.
- Gears Of War 4

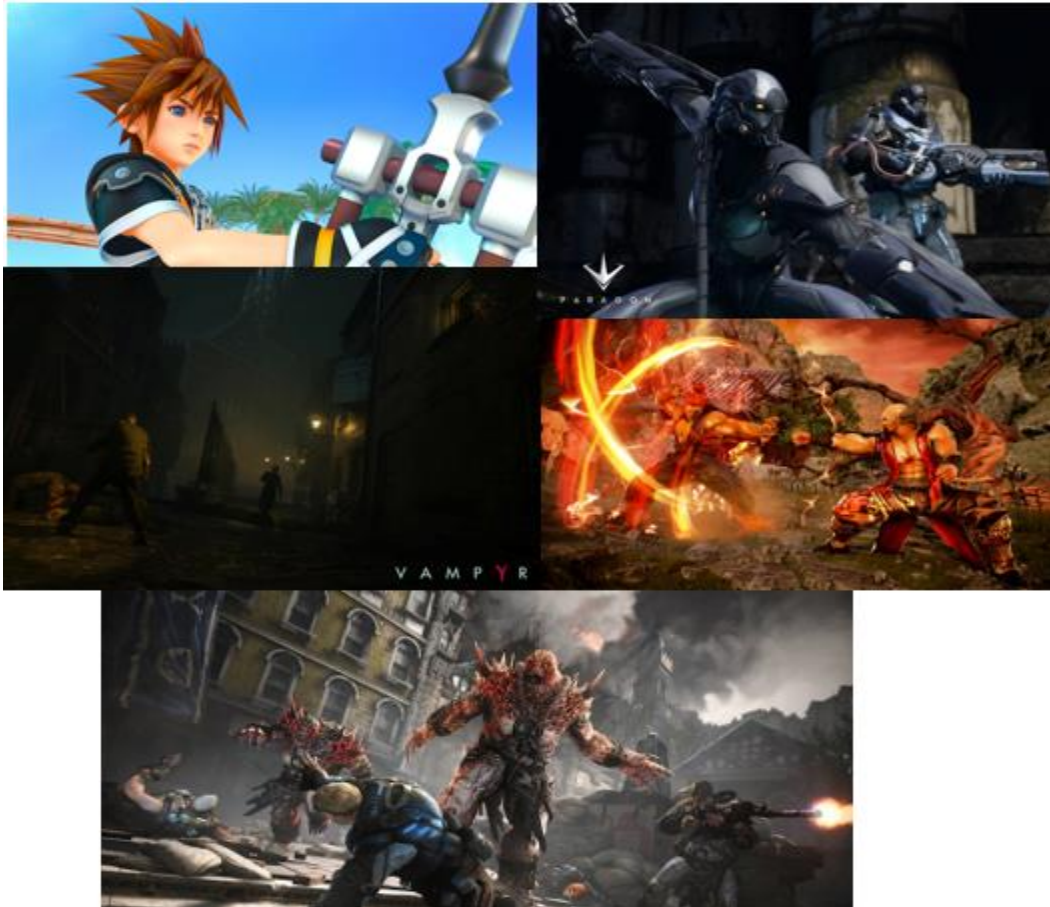


Figura 25: De izquierda a derecha y de arriba abajo, imágenes promocionales de Kingdom Hearts 3, Paragon, Vampyr, Tekken 7 y Gears of War 4.



UNITY 5

Unity Technologies, desarrolladores del motor Unity 5, comenzaron en el mundo de los negocios como una empresa de desarrollo de videojuegos en el año 2004 que para la publicación de su primer videojuego crearon una plataforma de desarrollos con una herramientas muy potentes e intuitivas. Al ver que las ventas del videojuego que habían publicado, Gooball, no había cumplido las cuotas de ventas necesarias y era lo que se denominaba un “fracaso empresarial”; la empresa decidió poner a la venta las herramientas que ellos mismos habían diseñado para realizar el videojuego [53].

Unity cuenta con una variedad de motores dentro de una misma versión cuyas funciones dependen de la plataforma con la que el diseñador desee trabajar y el tipo de necesidades que tenga el equipo de diseño. La versión *Pro* estándar del programa tiene un coste de 1500€ de base, pero en caso de necesitar módulos de comunicación entre ordenadores para el caso de equipos de diseño, la versión específica del programa que permite trabajar con una plataforma específica u otro tipo de funcionalidad desarrollada que no esté incluida en la versión *Pro* es necesario realizar el desembolso de más licencias de 1500€ [54].

Para no obligar a los desarrolladores pequeños a realizar este tipo de desembolsos por cada estación de trabajo de la empresa, Unity Technologies también ofrece una versión *indie* del motor gráfico totalmente gratuita con una cantidad reducida de las funciones que ofrece la versión *Pro*, pero que es suficiente para el desarrollo de juegos de pequeños estudios. Consideración por parte de la empresa de no ser necesario pagar más allá del coste que haya tenido el motor gráfico, aunque se realice una comercialización del producto desarrollado, incluso si este se ha realizado con la versión gratuita.

Gracias a este tipo de características, Unity ha sido durante la última década el motor favorito de los desarrolladores *Indie* gracias a las comodidades de pago que ofrecía según las necesidades de la empresa. Más de 50 juegos se han realizado en los últimos 3 años con este motor entre los que destacan conocidos juegos por el público general como Angry Birds o Plague Inc [55].

El problema de Unity ha sido que, hasta la última actualización, el motor no era lo suficientemente potente para crear juegos de la llamada “Next-Gen”; juegos con una gran cantidad de carga gráfica simultánea en pantalla y que a la vez se ven sometidos a una simulación de físicas y partículas. Esto provoca que en la lista de juegos que están programados en este motor apenas se encuentren licencias de nivel “triple A” que son el mejor indicativo del nivel gráfico que se puede alcanzar con un motor específico. De hecho, es complicado encontrar un juego realizado en Unity que se pueda proponer como estandarte de calidad del motor en gráficos de tipo realista [56].



Unity si bien no destaca por sus gráficos espectaculares o realistas si cuenta con un gran motor de simulación de físicas que permite ser capaz de crear colisiones de objetos en pantalla de forma que se vean fluidas y naturales. Esto queda patente en tipo de productos que se han mencionado con anterioridad, pues si los juegos *indies* no destacan por unos gráficos espectaculares o hiperrealistas si lo hacen por una jugabilidad que tiende hacia las mecánicas innovadoras [57]. Un juego que demuestra este tipo de tendencia es el anteriormente mencionado Angry Birds, un juego que, mediante el trazado de parábolas, se juega con la trayectoria y velocidad de un cuerpo con el fin de derribar una estructura (fig. 26) [58].



Figura 26: Imagen de una estructura a derribar de Angry Birds.

Unity se encuentra programado en C, C++ y C#, sin embargo, es este último el lenguaje de programación en exclusiva que utiliza la versión 5 del programa. C# tiene la ventaja de ser mucho más sencillo de utilizar que C++ ya que cuenta con una extensa variedad de librerías que ofrece una gran cantidad de funciones directamente programadas e implementadas con apenas unas pocas líneas de código.

Entre las librerías que ofrece Unity se encuentran además las necesarias para implementar el videojuego en un casco de realidad virtual como puede ser el HTC Vive o el Oculus Rift. La posibilidad de realizar un juego para un casco de realidad virtual indica que el motor es válido para la creación de experiencias de inmersión realista para el usuario.

La ventaja que tiene C# se basa en una curva de aprendizaje muy poco pronunciada con la que un pequeño grado de familiarización con el lenguaje, en el caso de ya conocer otros previos, consigue resultados en tiempos de producción muy reducidos.



Al igual que *Unreal*, *Unity Technologies* pone a disposición del usuario una tienda de *assets* o recursos de producción para que los proyectos de los desarrolladores tengan la posibilidad de adquirir productos para su proyecto que sean inviables de crear para un estudio pequeño o por plazos ajustados asociados al desarrollo. La gran mayoría de productos de este mercado se encuentran desarrollados por profesionales o la propia compañía que manufactura el motor gráfico.

En la actualidad una muestra de los juegos (fig. 27) [55] que utilizan *Unity* y su acabado son:

- Slenderman: The Arrival.
- Firewatch.
- Fallout Shelter.
- Broforce.
- Ori and the blind forest.
- Roller Coster Tycoon World.

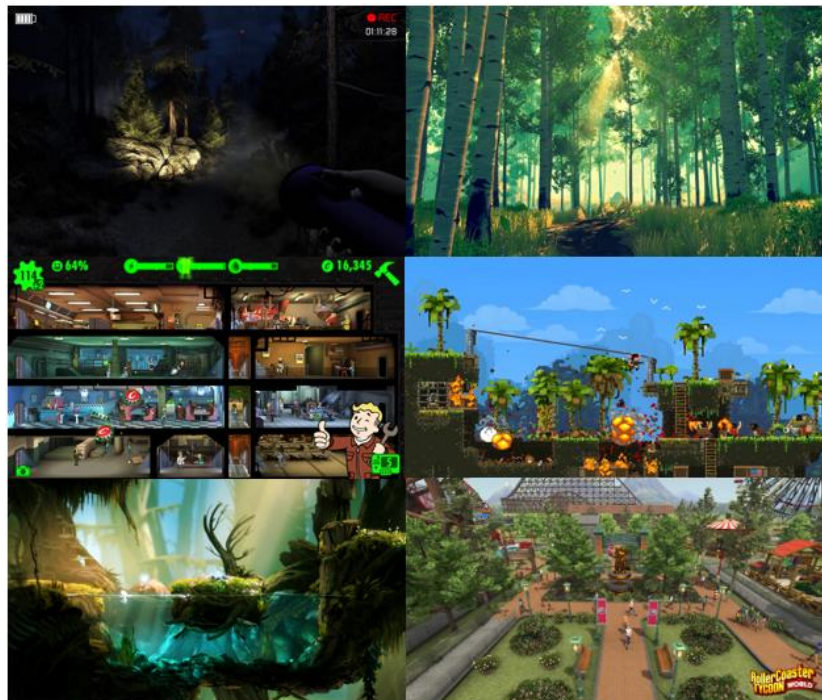


Figura 27: De izquierda a derecha y de arriba abajo, imágenes in-game de *Slender: The Arrival*, *Firewatch*, *Fallout Shelter*, *Broforce*, *Ori and The Blind Forest* Y *Roller Coaster Tycoon World*.



ELECCIÓN DE LA TECNOLOGÍA

Presentadas dos alternativas viables de motor gráfico es necesario decidir una para llevar a cabo el proyecto.

Unity cuenta a su favor con una *assetstore* mayor que la que ofrece *Unreal* y cuyos fabricantes suelen ser profesionales consagrados del sector. Por otro lado, todos estos productos son de pago y *Unreal* cuenta con una comunidad de usuarios mayor que pone a disposición del desarrollador novel un mayor número de recursos gratuitos o de pago sometido a un uso comercial. Esto también se ve reflejado en las comunidades online de usuarios siendo los foros de consulta de *Unreal* más activos que los de *Unity* y cuya actividad se ve derivada incluso a otro tipo de páginas web como *Reddit*.

Teniendo en cuenta el lenguaje de programación con el que se ha de trabajar al seleccionar uno de los productos, la formación que se ha proporcionado durante la carrera ha sido para familiarizarse con el lenguaje C++. Además, comparando ambos lenguajes de programación, *Blueprint* y C#, la posibilidad de trabajar con un entorno de desarrollo visual que facilite la comprensión del programa influye a la hora de seleccionar un lenguaje en el que sea necesario formarse para la consecución del proyecto.

Atendiendo al grado de realismo y calidad gráfica que ofrecen los dos motores, *Unreal Engine* se postula como la mejor de las opciones al ser capaz de tolerar con mayor facilidad la carga gráfica que conlleva una simulación realista. *Unity* tiene a su favor una mejor simulación de colisiones y físicas aplicadas a objetos, pero la sensación de inmersión principal se consigue por el realismo del entorno que rodea al usuario.

Puesto que ninguno de los motores tiene coste alguno para un desarrollador pequeño que no pretende obtener beneficio de una comercialización, ambos resultan igual de óptimos en el sentido económico. Aunque, el hecho de que la versión gratuita que se pone a disposición de *Unity* tiene funciones y características reducidas con respecto a la versión Pro es un punto en contra de este motor.

Por los motivos expuestos anteriormente se considera que la mejor opción para llevar a cabo este proyecto es el uso del motor *Unreal Engine* provisto por *Epic Games Company*. Para la obtención del mismo es necesario registrarse en la página web de la compañía con el perfil de estudiante/desarrollador con el fin de poder descargarse el instalador del programa.



CAPÍTULO:4

MONTAJE DEL ARMA





Desmontado del arma de airsoft

Para poder trabajar con la réplica dispuesta es necesario abrirla a la mitad por los medios dispuestos por el diseño y así ser capaces de introducir en su interior los componentes electrónicos que son la base de todo el sistema modular del proyecto de simulación.

El modelo de Heckler & Koch G36 Compacto obtenido dispone en su carcasa de 6 tornillos sujetando el riel de la mira que se deberán extraer para así posteriormente poder liberar los 15 restantes que mantienen las dos piezas longitudinales de la carcasa unidas entre sí.

Una vez abierta valorando los espacios disponibles es viable la introducción de todos los aparatos electrónicos para el proyecto en la parte de la recámara y el cañón sin necesidad de reducir a elementos como: el contrapeso de hierro; que da la sensación de realismo al sujetar el arma debido a que la réplica se encuentra fabricada en plástico y no en metal, el mecanismo de disparo; que al incluir el propio del arma no se ve necesario diseñar uno nuevo y contribuye al diseño y el sistema de recarga por muelle; que servirá como medio de generación de retroceso.

Sin embargo, será necesario el vaciado de todos los “props” de plástico incluidos en el cargador que pretendían ser balas para que el arma vista desde fuera diese apariencia de estar cargada, pues se requiere de ese espacio para la colocación de diversos módulos, así como la perforación de puntos concretos de la carcasa para poder incluir los pulsadores necesarios para los módulos de disparo, recarga y alimentación (fig. 28).



Figura 28: Esquema general de la disposición de los distintos módulos en el arma una vez abierta.

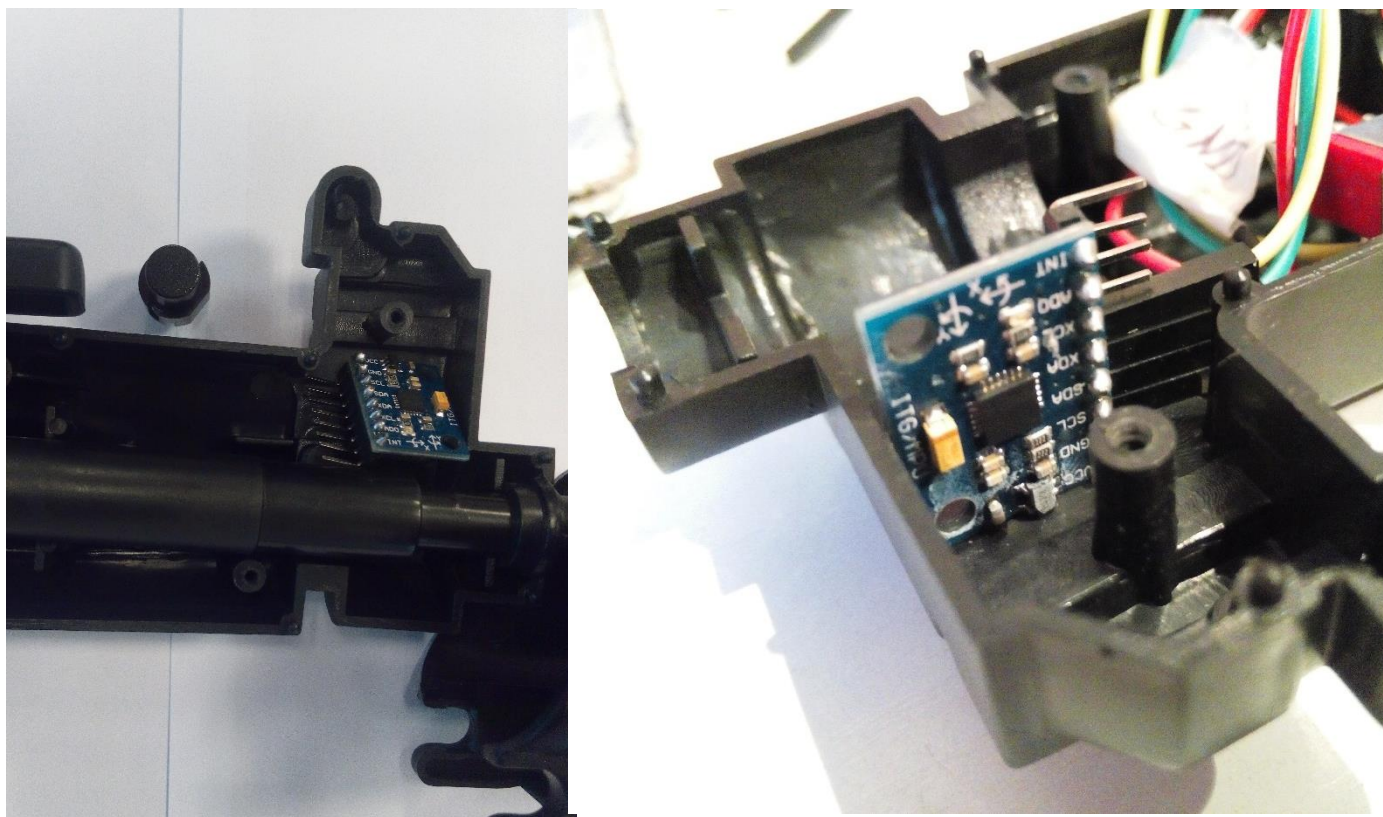


Colocación del giróscopo

El punto adecuado para la colocación del giróscopo será en la punta del arma (fig. 29). Esto se debe a que para poder simular correctamente el retroceso se pretende que cuando se dispare con el muelle cargado, las vibraciones producidas por el golpe repentino del elástico creen un ruido en la medida semejante al desapuntado propio de la fuerza de retroceso en las armas de fuego.

Puesto que va a estar soportando vibraciones de corte muy fuerte y agresivo, lo propio será sujetarlo a la carcasa del arma para que este tipo de perturbaciones se vean desplazadas también hacia la coraza exterior. Además, puesto que el funcionamiento del giróscopo se basará en las variaciones de medida en la posición que este perciba, sujetarlo no deja de ser la opción más lógica puesto que si no estuviese fijo podría desplazarse dentro del arma con los movimientos de la misma generando un tipo de mediciones que no son las que se pretende obtener.

Para poder inmovilizar el giróscopo dentro del arma se adherirá a esta mediante el uso de pegamentos de contacto de alta resistencia e instantáneo, esto dificultará el proceso de cambio del componente en caso de ser necesario en algún momento, pero es requisito si se pretende usar este método de localización de posición (fig. 30).



Figuras 29 y 30: Detalle del hueco disponible para instalar el giroscopo y detalle del giroscopo una vez ha sido pegado y conectado a los cables de conexionado.



Colocación de los interruptores

En el sistema existen dos pulsadores, uno correspondiente al gatillo que enviará la señal de disparo al Arduino y otro que indicará que se acaba de recargar el arma cuando esta se quede sin balas con el fin de poder volver a recargarlo.

Para la colocación de estos pulsadores se disponen dos puntos del arma donde resulta más lógica su colocación para el usuario que está sujetando el arma. En el caso del pulsador de disparo, este se encontrará justo detrás del gatillo del arma para que así cuando este se accione también se active el sector electrónico del proyecto (fig. 31). En el caso del de recarga se colocará en la parte superior del arma justo debajo del riel de la mira y un poco adelantado con respecto a la palanca de recarga así con la sujeción normal del arma el pulsador estará al alcance del dedo pulgar del usuario siendo cómoda su pulsación. En el caso de estar usando el modo de disparo manual, al recargar agarrando la palanca queda justo delante del dedo índice del usuario siendo así sencillo el realizar la recarga tanto del muelle como del entramado digital (fig. 32).

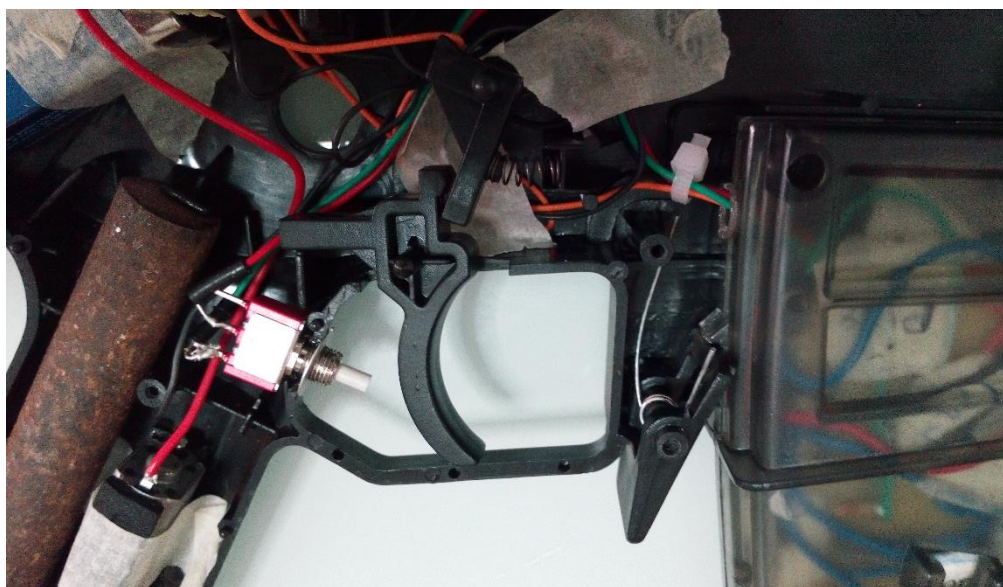


Figura 31: Detalle de la colocación del pulsador detrás del gatillo

Para la colocación de estos pulsadores es necesario modificar la carcasa del arma con el fin de crear un hueco por el que se puedan introducir. Para ello se utilizará un soldador de estaño caliente con el que fundir el plástico del armazón usando la punta que tiene el diámetro suficiente para el paso de las protusiones del pulsador. Estos se fijarán también mediante adhesivo instantáneo para asegurar la pulsación y que no oscilen o se desplacen al ejercer presión sobre ellos.

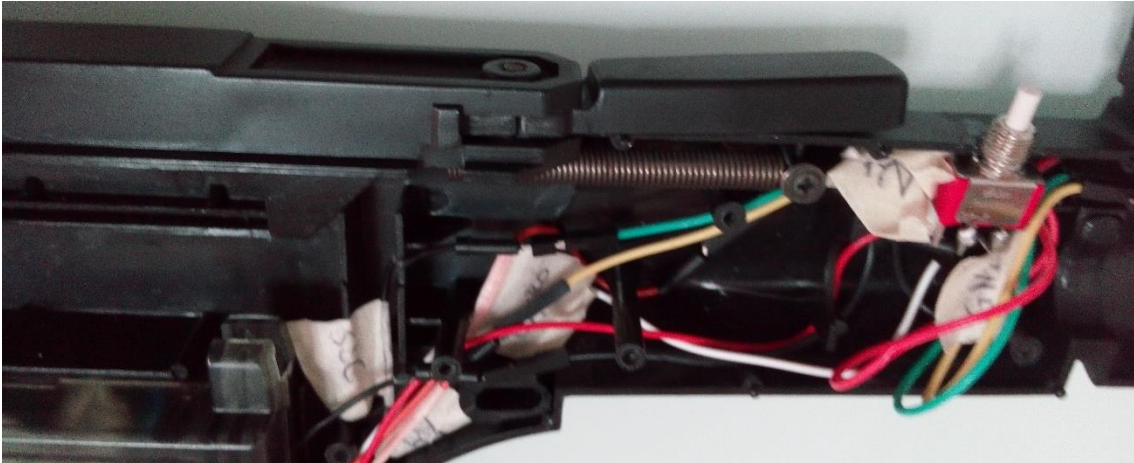


Figura 32: Detalle del pulsador de recarga en el que se puede apreciar la distancia con respecto a la palanca de recarga del muelle.

Colocación de los motores DC

Los motores de DC presentan características especiales con respecto a los demás elementos del proyecto siendo la principal, el hecho de que es necesaria una fuente de alimentación solo para el consumo esperado por estos elementos. Los motores no solo necesitan una fuente de alimentación, también un circuito de entrada asociado al Output del Arduino que regule la activación de los mismos. Toda la culata será reservada para los motores y el material asociado.



Figura 33:Detalle de la colocación de uno de los motores en el mango del arma



Los motores se fijarán en dos puntos clave a la hora de percibir el retroceso, esto se debe a que lo importante es que el usuario sienta que existe un retroceso no que todo el arma produzca vibraciones descontroladas con el fin de provocar una reacción en el usuario.

El primero de los puntos es el mango del gatillo que es en el que se va a colocar la mano dominante del usuario (fig. 33). Al tratarse de un punto clave de un arma es normal que el agarre que se realice sobre una localización como esa sea elevado, especialmente al disparar, pues aumenta el agarre del usuario al tener que hacer fuerza sobre el gatillo.

El segundo de los puntos será la parte superior del arma donde se acopla el apoyo del hombro al arma (fig. 34). Este punto se elige debido que de la misma manera que el usuario tiende a hacer fuerza agarrando demasiado fuerte el mango del arma ,también tiende a hacer demasiada presión contra el hombro en el apoyo, por lo que se transmite directamente la vibración del motor de DC contra el hueso de la clavícula.

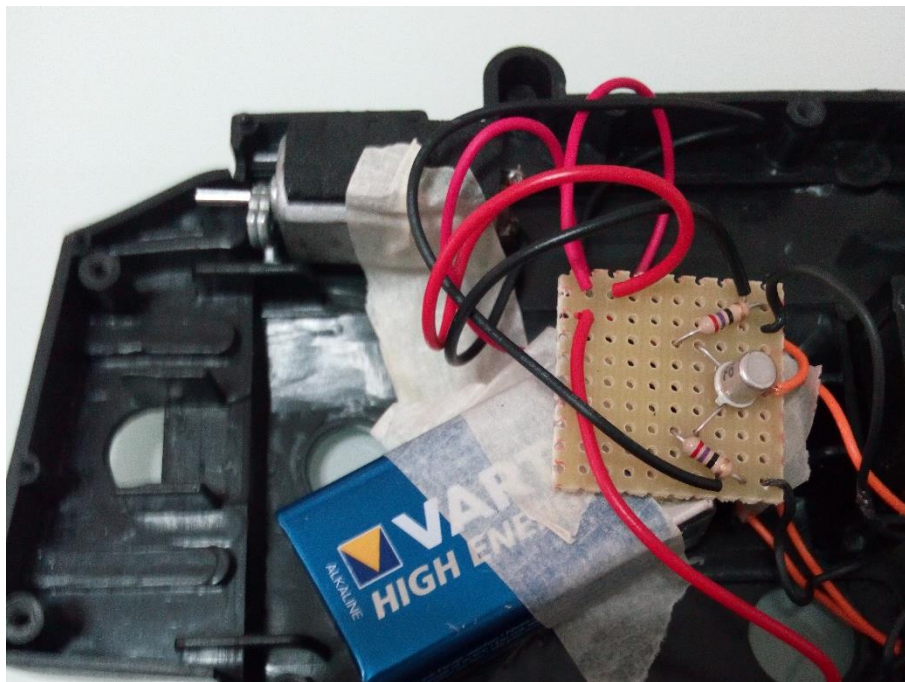


Figura 34: Detalle de la colocación del segundo de los motores, pila de alimentación del módulo de retroceso y circuito asociado al mismo.

La fijación que se hace de los motores contra el armazón no ha de ser excesiva pues entonces la vibración se vería amortiguada contra toda la estructura que al ser hueca no presenta resistencias internas debido al cambio de medio impidiendo así que toda esta fuerza vaya dedicada en su mayoría contra el usuario. Por ese motivo se decide usar un material mas endeble que permita un mayor grado de movimiento en la vibración como es el turró para poder funcionar de acuerdo a las expectativas planteadas.



Se aprovechara también este tirro para fijar la batería contra uno de los huecos libres que queda en la carcasa puesto que debe encontrarse fija, pero a diferencia del giróscopo si será habitual el tener que cambiarla pasado cierto periodo de uso.

La placa de circuito a la que se soldaran los componentes correspondientes: dos resistencias y un transistor NPN, se mantendrá libre sin estar fijada a ningún elemento, pues la presión ejercida entre las dos placas y la rigidez de los cables permiten mantenerla con un grado de inmovilidad mas que suficiente para que no oscile dentro de la carcasa.

Colocación de los sensores de ultrasonidos

Los sensores de ultrasonidos son el elemento que permite al usuario moverse en el entorno del que se ve rodeado, por lo tanto es necesario que el punto del arma donde se encuentren no conlleve una oclusión por parte del diseño del mismo arma o por parte del cuerpo del usuario al entrometerse en el rango de rebote de la onda sonora.

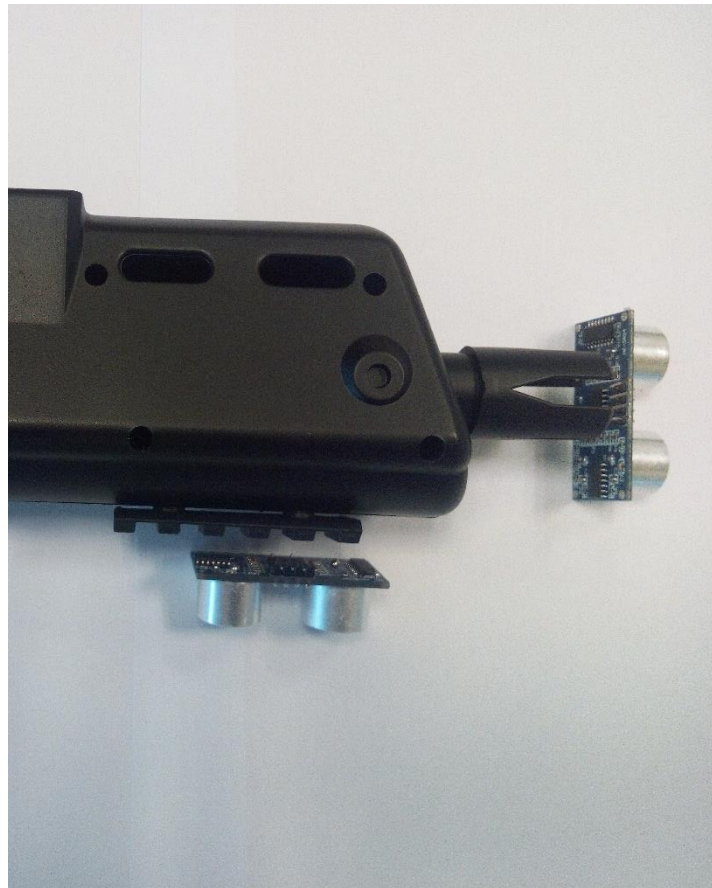


Figura 35: Imagen de la localización de los sensores en el arma



El punto idóneo donde colocar el sensor de ultrasonidos para el movimiento frontal o asociado al caminar es en la boca del arma (fig. 35), pues es necesario que este se encuentre siempre en algún punto donde sea fácil controlar si se esta realizando una variación contra cualquiera de las superficies que rodean al usuario. Es en esta localización donde va a tener al menos una pantalla enfrente de él proporcionado un punto fijo y estable.

En el caso del sensor de distancia vertical o de salto, se colocará en el frontal del arma donde se encuentra el riel en el que normalmente se colocaría la mira de apuntado láser. Esto se debe a que es necesario que el sensor no se encuentre ningún tipo de obstáculo móvil que pueda dar algún error de medida y en este tipo de obstáculos móviles se encuentran categorizadas las piernas del propio usuario. El agarre del arma tal y como se realiza cuando la culata del arma se encuentra colocada, añade la distancia suficiente desde la punta del arma hasta el cuerpo del usuario proveyendo de un margen de 0.5m en el cual es improbable que se vayan a encontrar las piernas del usuario (fig. 36).

Figura 36: Imagen de un usuario con el arma real con localización del sensor vertical marcado en rojo



y distancia desde el mismo hasta el centro de gravedad del cuerpo

Para sujetar estos sensores se recurrirá a dos planchas de algún material ligero para crear una superficie en la boca como el riel donde se puedan sujetar sin problemas los sensores mediante algún tipo de material adhesivo que los mantenga fijos en su posición (fig. 37).



Figura 37: Imagen de los sensores una vez colocados y fijados a la boca del arma.



Primero se envolverá el cartón contra la boca del arma con el tirro y posteriormente se utilizará el mismo tirro para unir el sensor contra la superficie que se acaba de crear dejando al limite de la misma los pines del sensor para que estos no se vean cubiertos por el adhesivo. Para el caso del sensor vertical se aprovechara la existencia de una mayor superficie lisa desde la que partir y se unirá la plancha de cartón con el arma gracias a pegamento adhesivo de contacto instantáneo en vez de usar cinta para así obtener una mayor seguridad en la fijación del elemento. El resto de los pasos sera igual a los realizados con el sensor frontal (fig. 38).

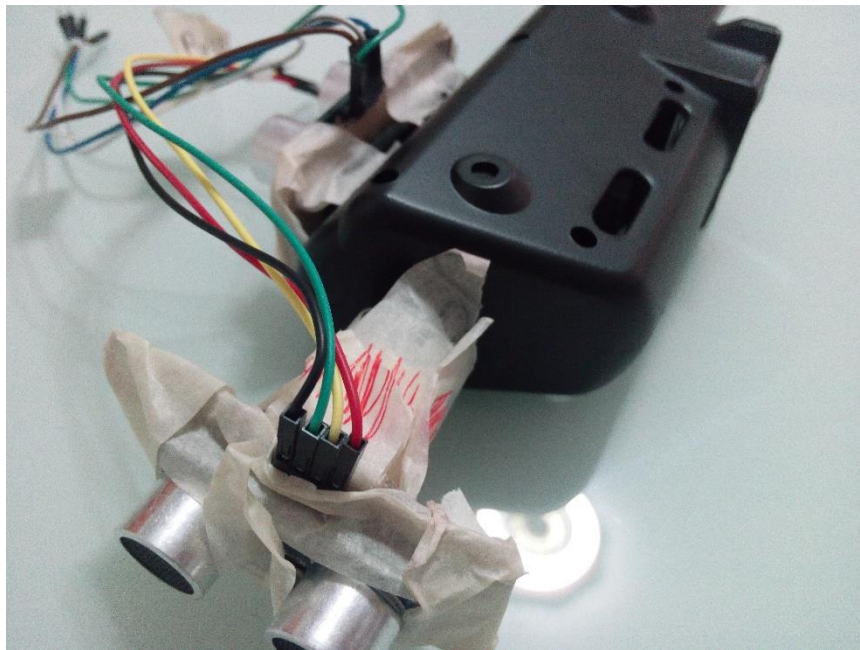


Figura 38: Detalle del sensor frontal mostrando el hueco para los cables de los pines

Colocación de los elementos en el cargador

La colocación de los elementos restantes se realizará en el cargador del arma por ser el único espacio restante en el arma del que se pueden sacar cables por el exterior de la misma para conectar de forma cómoda los sensores de ultrasonidos al Arduino (fig. 39).

Dentro del mismo ira dispuesto el Arduino dado que cabe perfectamente dentro del cargador y gracias al grado de curvatura que este posee, el microprocesador tampoco se desplazará una vez colocado en el interior del cargador. También se incluirá el sensor de Bluetooth, la fuente de alimentación y por uno de los laterales también se incluirá el interruptor de encendido del arma (fig. 40).

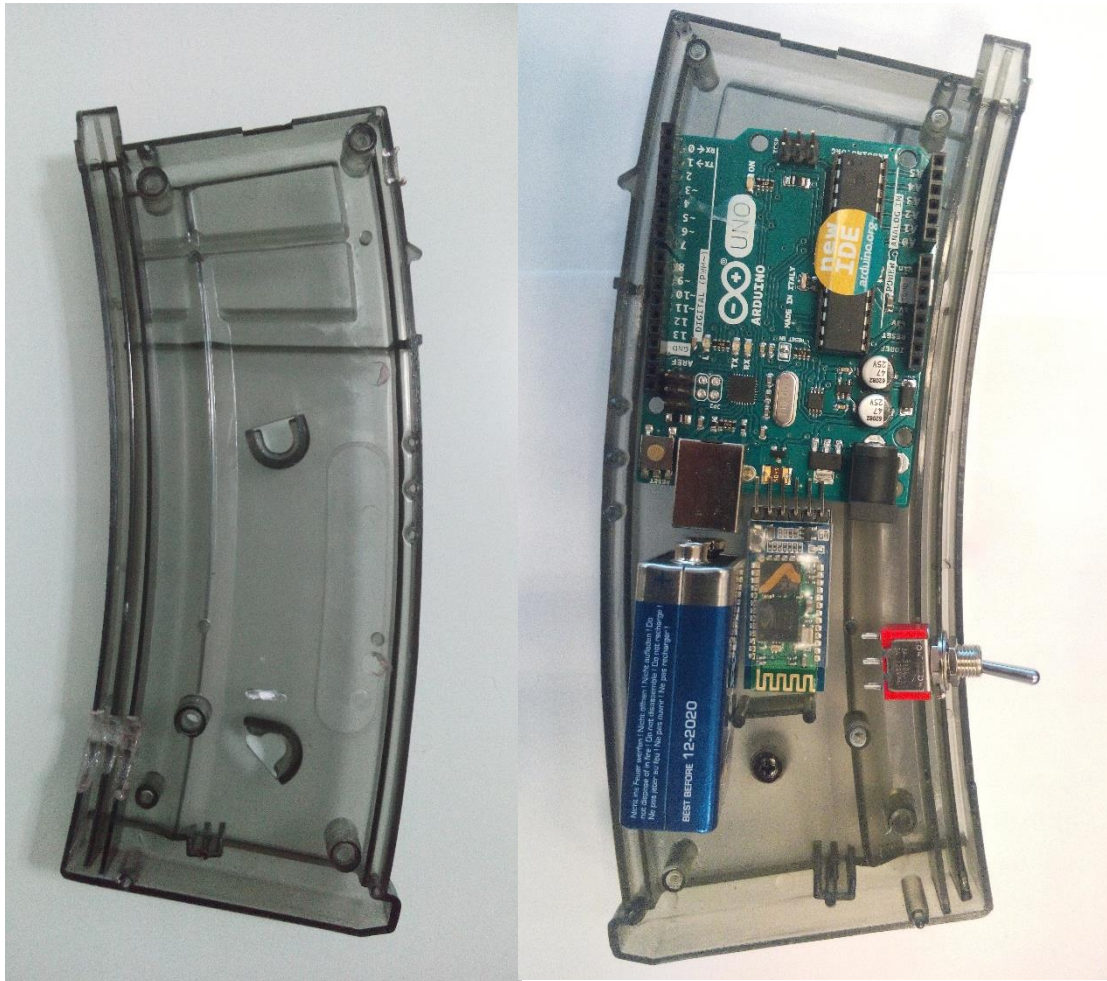


Figura 39 y 40: En la figura 39 se aprecia el espacio disponible dentro del cargador y en la figura 40 se comprueba que dicho espacio permite albergar los componentes.

Es viable colocar el *Bluetooth* en el interior del cargador ya que las ondas del mismo no se ven bloqueadas por el plástico, así se obtiene un lugar donde tenerlo fijo gracias a las condiciones del propio espacio. Además, se decide este punto como lugar de colocación del *Bluetooth* porque este posee un LED en su circuito impreso que indica el tipo comunicación que se está realizando y gracias a él obtenemos un indicador más visual para cualquier tipo de usuario sobre el estado de apagado o encendido del arma, puesto que la carcasa del cargador es transparente y es más sencillo de observar que voltear el arma para comprobar la posición del interruptor en ON u OFF.

La alimentación se obtendrá gracias a un adaptador para baterías y una clavija de alimentación de 5.5x2.1mm de los cuales uno de los cables se cortará para poder soldarlo a las patas correspondientes del interruptor.

Para una mayor seguridad de las conexiones y controlar el espacio dentro y fuera del cargador, los cables que se agrupen hacia una misma dirección del arma o se vayan a conectar a un mismo aparato electrónico se sujetaran entre ellos mediante bridas.

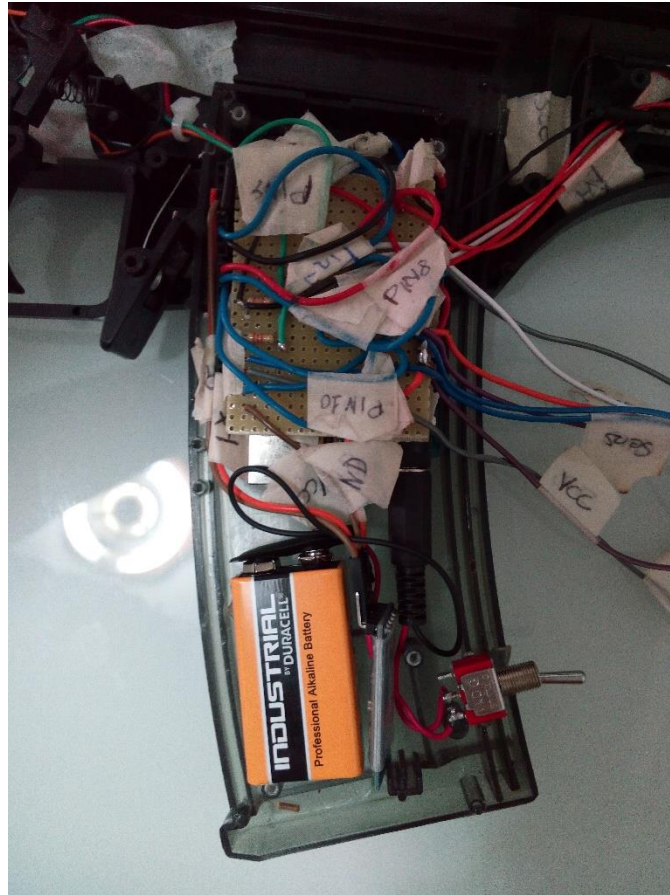
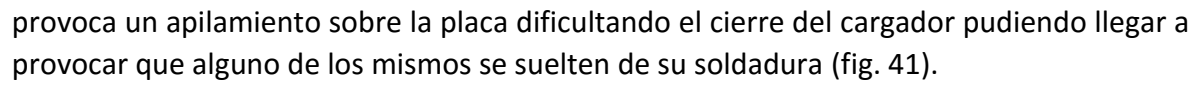


Para la conexión de todos los cables a los pines correspondientes del Arduino se usará una placa de circuito a la que se soldarán todos los cables que vengan desde los demás componentes electrónicos y desde ella se sacarán cables que vayan directamente al Arduino. Este tipo de placas son lo suficientemente estrechas como para no aumentar el espacio necesario dentro del cargador ya que se cortará una de unas dimensiones de 4 cm menor que el Arduino para poder colocarse encima de él. A esta placa de circuitos se le colocará una vez esté terminada una base de cinta aislante por debajo de ella, cubriendo las soldaduras, para asegurar que no hace conexión de forma accidental con alguno de los pines macho que sirven para conectar los módulos externos de Arduino provocando algún tipo de error. En esta placa también se soldarán las resistencias que requieren los circuitos de los pulsadores para evitar la necesidad de colocar más placas de circuitos en el interior del arma.

A esta placa van soldados de la siguiente forma:

- 7 Cables unidos por soldadura que corresponden al Vcc del Giróscopo, los dos sensores de ultrasonidos, los dos pulsadores, el módulo *Bluetooth* y el que corresponde a la alimentación desde el Arduino.
- 5 Cables unidos por soldadura que corresponden al GND del Giróscopo, los dos sensores de ultrasonidos, las dos resistencias del circuito asociado a los pulsadores, el módulo *Bluetooth* y el que corresponde a la tierra desde el Arduino.
- 4 Cables unidos dos a dos que corresponden a la emisión y transmisión del giróscopo con el Arduino.
- 4 Cables unidos dos a dos que corresponden a la emisión y transmisión del *Bluetooth* con el Arduino.
- 4 Cables unidos dos a dos que corresponden a la emisión y recepción del pulso sónico del sensor de ultrasonidos frontal.
- 4 Cables unidos dos a dos que corresponden a la emisión y recepción del pulso sónico del sensor de ultrasonidos vertical.
- 2 Cables que corresponden a la pata de vuelta del pulsador y el input del Arduino con el pulsador que se unen a la resistencia del disparador.
- 2 Cables que corresponden a la pata de vuelta del pulsador y el input del Arduino con el pulsador que se unen a la resistencia de la recarga.
- 2 Cables que corresponden al Output del Arduino con el circuito de activación de los motores.

Estas conexiones han de hacerse con cuidado y tener muy controlado el largo del cable que deberá de ser de la longitud justa para alcanzar desde la soldadura de la placa hasta el Arduino ya que al tener que doblarlos sobre sí mismos, la acumulación de estos







CAPÍTULO:5

SISTEMA DE CIRCUITOS





El sistema de control del proyecto está basado en el uso de componentes electrónicos que tienen como base el Arduino UNO para la coordinación de las diferentes señales. El objetivo de este apartado es la comprobación de cómo se realizan dichas conexiones y porque se hacen de esa manera, así como las principales características técnicas de estos circuitos.

Conexiones directas al Arduino

La mayoría de los elementos se encuentran directamente conectados al Arduino sin necesidad de la intermediación de otros elementos electrónicos ya que su funcionamiento está pensado para que trabajen con los pines de un microprocesador de características semejantes (fig. 45).

El *Bluetooth* está pensado para ser conectado directamente al propio Arduino es por eso que no es necesario nada más que alimentarlo correctamente. Para el proyecto una vez terminado simplemente será necesario la utilización de 4 pines (fig. 42) cuyas funciones y conexiones son:

- **+5V/VCC:** En la pieza dispuesta es el pin que corresponde a la alimentación del módulo de *Bluetooth*. Como fue mencionado en el apartado 6 de la sección anterior, no va conectado directamente a un pin, sino que va a una unión general desde la que se alimentan todos los módulos.
- **GND:** Corresponde a la tierra del módulo *Bluetooth*. Al igual que el pin de +5V no está conectado a ningún pin en concreto si no que se realiza una soldadura de todos los cables que vayan a tierra para conectarlos a la del Arduino.
- **Tx y Rx:** Son los pines que se ocupan de la transmisión de datos del *Bluetooth* por lo que uno se configura como salida y el otro como entrada. Puesto que no existe ningún tipo de exigencias sobre el pin del Arduino que deban ocupar, ya que simplemente ha de poder configurarse como digital y puerto serie, en este caso irán conectados a los pines 3 y 2 respectivamente.

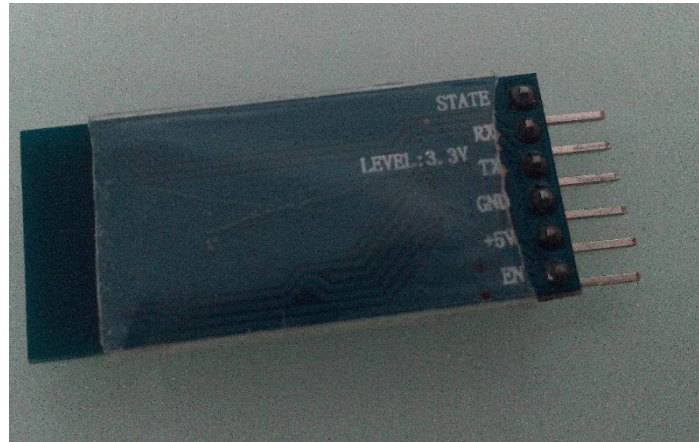


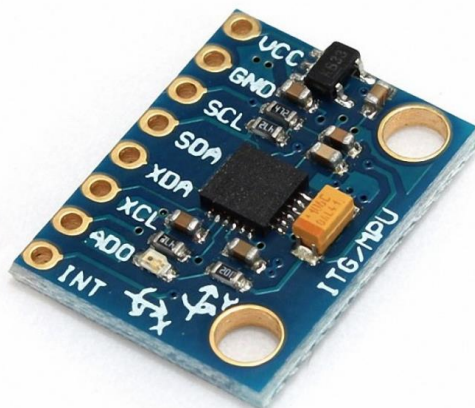
Figura 42: Detalle del Bluetooth con sus pines.

- **GIRÓSCOPO.**

El giróscopo comparte con el módulo *Bluetooth* la característica de que no es necesario nada más que el Arduino para alimentarlo en su funcionamiento, sin embargo, debido a que los pines con los que trabaja han de ser analógicos y la propia librería con la que este aparato funciona, los pines de conexión son más específicos (fig. 43).

- ❖ **VCC:** Comparte las características del *Bluetooth*.
- ❖ **GND:** Comparte las características del *Bluetooth*.
- ❖ **SCL:** Se encuentra conectado como requisito al pin A5.
- ❖ **SDA:** Se encuentra conectado como requisito al pin A4. La conexión específica de estos pines se debe a que la comunicación del giróscopo se hace mediante una conexión I2C y la forma correcta de recibir en orden la información por los puertos es la mencionada en el caso de utilizar un Arduino UNO.

Figura 43: Imagen de los pines que posee el giróscopo.





- **SENSORES DE ULTRASONIDOS**

Los sensores de ultrasonidos presentan las mismas características que el módulo de *Bluetooth* por lo que toda conexión depende directamente del Arduino. Estos aparatos solo disponen de 4 pines (fig. 44), pero será necesarios utilizarlos todos. El único requisito de los pines específicos de este aparato es el que sean pines digitales por lo que no existe ningún tipo de impedimento a la hora de ser configurados de forma distinta a la aquí mostrada.

- **VCC:** Comparte las características del *Bluetooth*.
- **GND:** Comparte las características del *Bluetooth*.
- **ECHO:** Se trata de un pin que ha de ser configurado como input y corresponde al puerto por el que se manda la señal del tiempo que se ha tardado en recibir el eco de la onda sonora una vez se ha enviado desde el altavoz de ultrasonidos. En el caso del sensor horizontal se conectará al pin 8 y en el del vertical al pin 10.
- **TRIG:** Se trata de un pin que se configurará como output y corresponde a la recepción por parte del sensor de la orden de emitir el pulso ultrasónico a través del altavoz. En el caso del sensor horizontal se conectará al pin 9 y en el del vertical al pin 11.



Figura 44: Imagen de los pines propios del sensor de ultrasonidos.

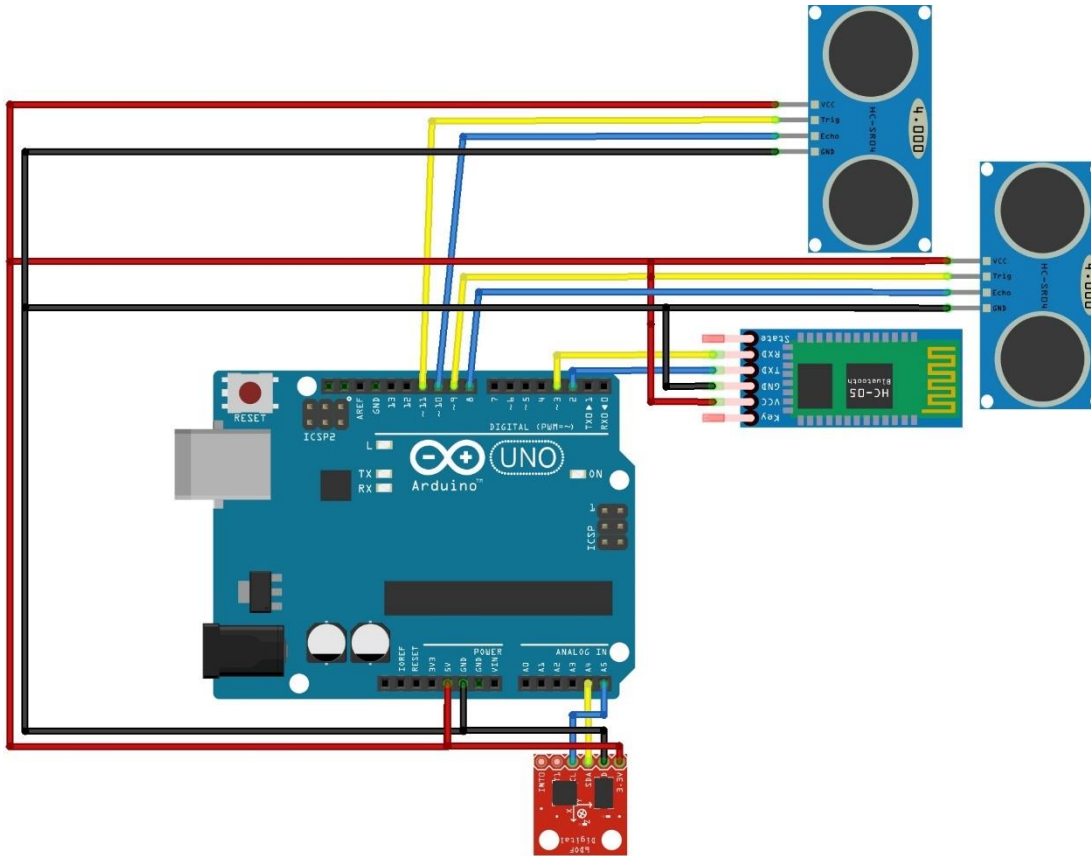


Figura 45: Conexión de los elementos que van directos al Arduino.

• PULSADORES

Ambos pulsadores se han de conectar de forma idéntica sin embargo requieren de algo de circuitería para asegurar que el pin no recibe una señal errónea que no se corresponde con la posición del pulsador.

Para evitar el error en la señal se dispone de una resistencia entre el pin y la tierra que haga de elemento conductor y a la vez garantice la existencia de una diferencia de 5V entre dos puntos del circuito.

Esta resistencia puede ser del valor que para la construcción del circuito sea más cómodo. La condición que presenta la selección de esta resistencia es que la corriente no sea demasiado baja o demasiado alta para impedir que el Arduino falle por una sobrecarga de intensidad o no perciba la diferencia de tensión por culpa del bajo amperaje. En este caso se opta por usar dos resistencias disponibles de 2.2K Ω .

Los pines a los que se conectan los pulsadores dependen de la configuración del Arduino que quiera el constructor del modelo. Para el proyecto se ha usado el pin 4 en el disparador y el pin 5 en la recarga (fig. 46).

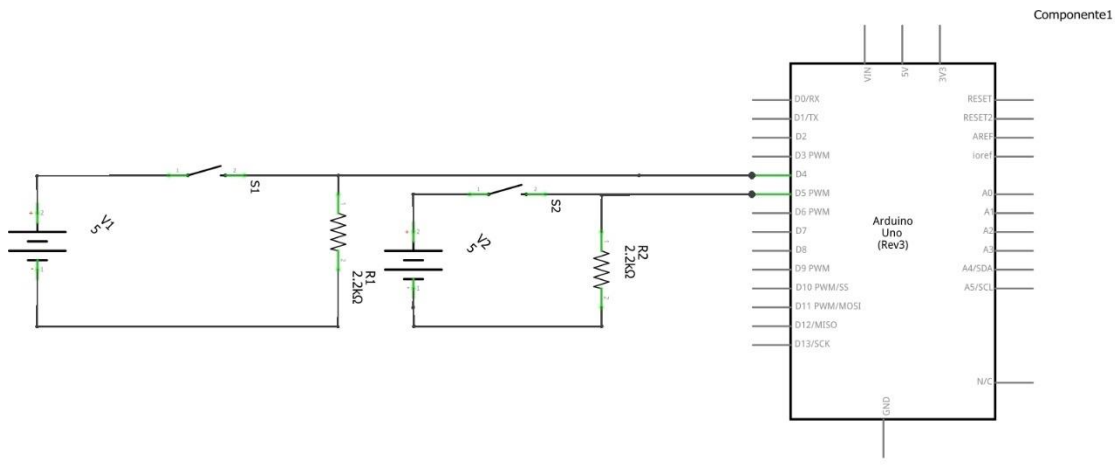


Figura 46: Esquema de la conexión de los pulsadores. Se ha usado una batería como fuente de alimentación en vez de 5V y GND para mayor claridad en el circuito.

• MOTORES DC

Los motores de corriente continua tienen una especificación de 75mA con una mínima de 20 mA por lo que será necesario la colocación de unas resistencias que conectadas a la batería independiente asociada a esta sección del circuito regulen la intensidad que circulará por ellos.

Se usará un transistor NPN 2N2222a con el fin de regular el paso de corriente de acuerdo a la señal digital enviada a través del Arduino. Se descarta la utilización de un relé debido a que la velocidad de respuesta del transistor es más rápida al ser electrónica y no física. En este proyecto es condición que el tiempo entre el disparo y la activación del retroceso sea lo más próximo a cero posible. El requisito se debe a la disociación que se produciría entre el disparo en la pantalla y el retroceso en la mano lo cual rompería la sensación de inmersión.

Dependiendo del estado del Output del Arduino, el transistor se encontrará en un estado u en otro. En caso de que la señal sea un 0 el transistor se encontrará en estado de OFF ya que la $V_i < V_{BE}$ (según la datasheet la V_{BE} es 0.6V) y en el caso de que este conduzca estará en un estado activo

La intensidad por el colector se calculará como una mezcla de las ecuaciones 1 y 2 obteniendo la intensidad de 75mA indicada en el datasheet de los motores.



$$i_B = \frac{V_i - V_E - V_{BE}}{R_i}$$

Ecuación 1: Formula de la intensidad en la base del transistor

La beta del transistor es de 100 por lo que la resistencia necesaria será de 3.6KΩ. Esta resistencia tampoco ha de ser extremadamente precisa puesto que el requisito es que el motor dé un par próximo al nominal para asegurar la sensación de retroceso en el usuario. En el caso del proyecto se usaron dos, una de 3.3KΩ y otra de 330Ω.

$$i_C = \beta * i_B$$

Ecuación 2: Relación entre la intensidad del colector y la base en hipótesis de funcionamiento en activa del transistor bipolar.

Los motores se colocarán en serie puesto que su rango de voltaje en funcionamiento es de 3V a 10V y con la pila de 9V y la caída de tensión a la entrada del colector caen 3.15V a cada una y ya se ha comprobado que la I_c es la correcta.

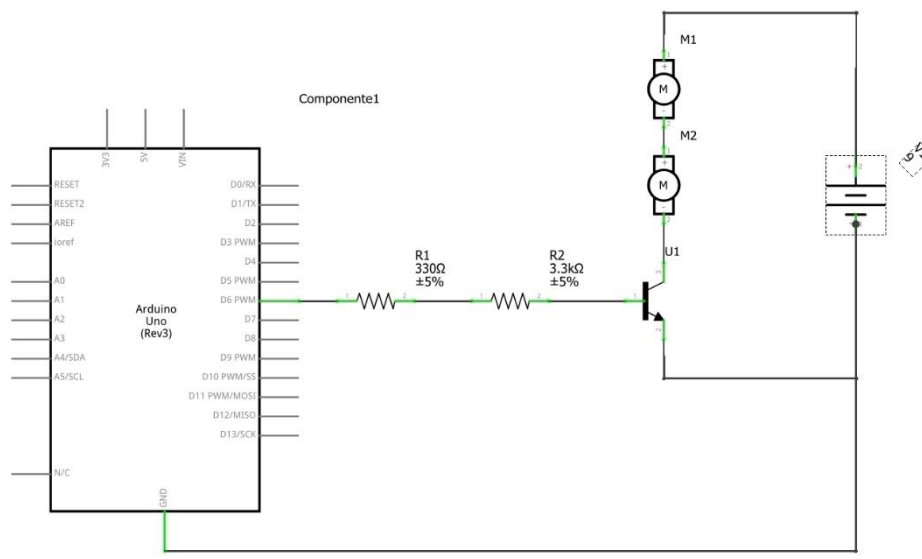


Figura 47: Esquema del circuito asociado a los motores

El pin al que se conectará este subsistema depende del diseño que se le quiera dar a la circuitería ya que el único requisito es una salida digital por parte del Arduino (aunque también funcionaría con una analógica en caso de que se quiera graduar la intensidad en la base para un mayor ajuste del par). En este caso se ha optado por usar el pin 6 (fig. 47).



CAPÍTULO:6

SOFTWARE DE CONTROL DEL ARMA





El control de este proyecto se realiza en dos partes, una que es el gobierno del arma en el Arduino y otra que es el procesamiento de la información provista desde el arma en el ordenador que se realiza en el motor gráfico. En este apartado nos centraremos en la explicación del código que está implementado en el Arduino comentando las partes más importantes.

En el programa debe incluir las siguientes dos librerías:

- `<SoftwareSerial.h>`: Esta librería está diseñada para permitir la comunicación en serie por parte del Arduino más allá de los pines 0 y 1 que son los reservados para la programación mediante USB gracias a una UART instalada en el ATmega. En el caso del proyecto esta librería es la que permite la transmisión de datos con el módulo *Bluetooth* y sin ella no se podrían reconfigurar los puertos necesarios para ello [59].
- `<Wire.h>`: Esta librería permite al Arduino comunicarse con dispositivos I2C / TWI mediante el uso de pines reservados (A4 y A5 en el Arduino Uno). Es necesaria porque el giroscopo se trata de un dispositivo I2C y sin las funciones provistas por esta librería no sería posible establecer comunicación [60].

Una vez realizada la definición de todas las constantes del programa y la declaración de todas sus variables se inicia el bloque *setup* que solo se ejecutará una vez y sirve para definir los puertos que se van a utilizar, así como establecer las características de la comunicación con el giroscopo y el Arduino.

Cuando el bloque *setup* termina inicia la función *loop* que es el programa propiamente dicho. Cuando este bloque sea ejecutado del todo volverá al principio y continuará haciendo lo mismo hasta que se apague el Arduino.

La obtención de los valores de posición del arma se realiza mediante el código descrito en las líneas 91-108 del anexo IV. La función *Wire.beginTransmission* se ocupa de comenzar una conexión de datos con la dirección del esclavo I2C proporcionada que se verá reiniciada con la función *Wire.endTransmission* al estar asignada a un booleano de tipo *false*. La función *Wire.write* se ocupa de indicar el registro desde el que se pide que se escriban los valores y la *Wire.requestFrom* de señalar cuantos registros se deben mandar contando desde el indicado en *Wire.write* [61].

Los registros necesarios desde la IMU son los que ocupan desde la posición 0x3B a la 0x40, pues contienen la información del acelerómetro correspondiente a la posición del giroscopo y los que van desde la posición 0x43 a la 0x48, pues contienen la información de la velocidad angular que posee en ese instante el giroscopo. La información relativa a cada uno de los ángulos y velocidades se guarda en dos registros [62].



$$\theta_i = \theta_{i-1} + \omega * \Delta t$$

Ecuación 3: Obtención de un ángulo mediante la información provista por el giróscopo.

La obtención del valor real de la posición ha de realizarse mediante la ecuación 3 más el valor obtenido es sensible a la variación provocada por el ruido de medidas, especialmente las provistas por el giróscopo. Por este motivo se debe implantar algún tipo de filtro que permite evitar un *drift* o desplazamiento en la medida desplazando el valor para el cual se considera que el arma se encuentra centrada.

Para el correcto funcionamiento del Arduino se deshecha la posibilidad de la inclusión de un filtro de Kálmán. Esto se debe a que al trabajar con una alta cantidad de diferenciales que se deben aplicar en los cálculos de los 3 ejes el resultado del filtro es demasiado lento para la velocidad a la que trabaja el Arduino.

$$\theta_i = f * (\theta_{i-1} + \omega * \Delta t) + (1 - f) * \theta_{acelerometro}$$

Ecuación 4: Ecuación del filtro complementario

Se implementará entonces un filtro de tipo complementario cuya fórmula corresponde a la de la ecuación 4 donde la f es el grado de confianza que se le decide dar a la medida del ángulo actual con respecto al anterior en comparación al que provee en ese instante el acelerómetro. Esta fórmula se encuentra escrita en forma de código en las líneas 126,127 y 128 del anexo IV correspondiendo a cada uno de los ejes espaciales [63].

La obtención de la medida de los ultrasonidos se realiza entre las líneas 140-178 del anexo IV. Para poder obtener la medida es condición asegurarse de que el pin del *trigger* se encuentra a 0. Después de esperar unos milisegundos por la perturbación que se haya podido causar se manda el pulso escribiendo HIGH en el pin. Se añade un pequeño *delay* entre la función *pulseIn*, que da el valor en tiempo de lo que ha tardado la onda en regresar, para asegurar que el valor que recibe no es el de la onda al propagarse por estar el micrófono adyacente al altavoz de ultrasonidos.

Al final del programa se deben escribir en el *Bluetooth* todas las variables que sean necesarias para el trabajo del *Unreal Engine* para que sean enviadas por comunicación serie.



CAPÍTULO:7

ELECCIÓN DEL SOFTWARE





En esta sección de la memoria se procederá a explicar los elementos que conforman el entorno de simulación y como estos se encuentran programados en el motor gráfico seleccionado.

DESCRIPCIÓN DEL ENTORNO

El entorno pretende simular un jardín en el que se encuentran localizados los elementos propios de una localización semejante en la vida real. Para ello se ha hecho uso del pack de recursos “*Foliage Starter Kit*” desarrollado por el usuario del foro de *Unreal Engine* “fighter5347” para obtener los modelados 3D que forman parte del entorno. De este pack se han utilizado los arboles provistos por el modelo, así como los arbustos de baja estatura colocándose un total de 15 árboles y 16 arbustos [64].

Para el desarrollo de la hierba se ha utilizado la herramienta *Foliage* provista por el motor gráfico del juego permitiendo una distribución aleatoria de los modelos de flores y hierbas provistos por el pack. Esto ha permitido mediante un pincel de área 5m de diámetro con variaciones de densidad de 1 a 4 puntos la colocación de un total de 16000 *assets* de hierba de forma aleatoria en el entorno para un mayor grado de naturalidad. También se le han incluido texturas de hierba al bloque sobre el que se encuentra sostenido el nivel con el fin de que este no desentone con los *assets* que este tiene colocado (fig. 48).



Figura 48: Plano lateral del simulador.



El kit de desarrollo también incluye partículas de hojas en el aire. Para un mayor grado de realismo este tipo de partículas se le han añadido a los árboles implementados en el proyecto con el fin de que cuando el usuario se acerque a uno de ellos observe como se le caen las hojas (fig. 49).



Figura 49: Detalle de un árbol con partículas implementadas.

Para el resto de la configuración del lugar se han provisto bloques creados con la herramienta provista por el propio motor gráfico a los que se les ha dotado de diferentes texturas y materiales para no desentonar con el entorno creado. Para el caso de las paredes se ha utilizado un material reflectante cerámico para aumentar la luminosidad del entorno y evitar el mayor número de sombras posibles. En el caso de los bloques que se han determinado estáticos se ha usado una textura de construcción mediante bloques de piedra semejantes a los que se pueden encontrar en los muros de las casas viejas con jardín.

La iluminación se ha colocado totalmente cenital superior con el fin de evitar la proyección de sombras de gran tamaño que ralenticen la simulación y a la vez no eliminarlas para evitar la sensación de incomodidad en el cerebro al percibir que ninguno de los elementos proyecta algún tipo de sombra (fig. 50).



Figura 50: Plano cenital donde se aprecia la no proyección de sombras hacia los laterales de los elementos.

El entorno del cielo se ha realizado mediante el uso de una de las herramientas provista por el motor gráfico, esta herramienta permite establecer el color deseado para el cielo, así como las nubes y la velocidad a las que estas se mueven. Se han establecido unos parámetros que permiten percibir el movimiento de las nubes añadiendo dinamismo a la escena sin necesidad de crear elementos móviles reales (fig. 51).

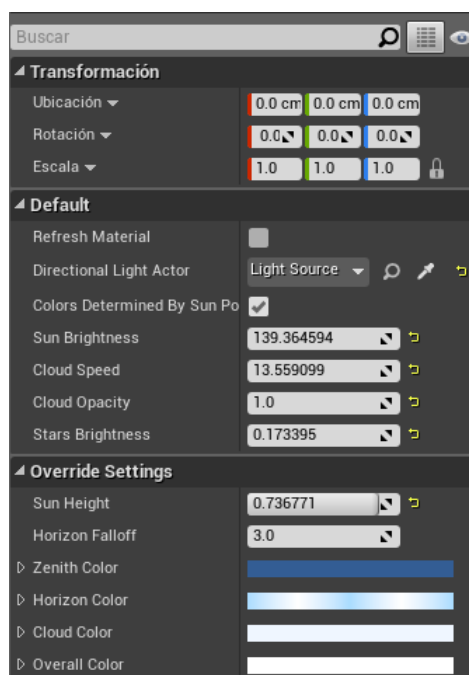


Figura 51: Valores definidos en el actor SkySphereBlueprint.

Se han colocado un total de 6 cubos como elementos dinámicos no fijos al entorno con una distribución en grupos de dos separados 120° entre ellos teniendo como centro el punto de partida del usuario establecido en el centro del jardín.



Estos cubos se han realizado gracias a la herramienta provista por el motor gráfico y se les ha asignado un material totalmente blanco que se perciba como alienígena dentro del entorno para que el usuario entienda a primera vista que ese es el elemento dinámico al que se le puede disparar y que responderá ante los estímulos externos que reciba.

El modelo del arma que posee el jugador en el juego y el diseño del personaje se han obtenido del starter kit que provee el *Unreal Engine* una vez es instalado.

INTERFAZ

Interfaz es el nombre que reciben los elementos fijos en pantalla que constituyen las fuentes de información necesarias por el usuario para moverse y trabajar con la simulación. En el caso de este proyecto puesto que lo que se pretende es conseguir la inmersión del usuario la interfaz es un elemento muy reducido, pues la acumulación de elementos en pantalla aleja al usuario de la sensación de realidad.

En la interfaz se localizan simplemente dos elementos programados para dar la información al usuario más relevante.

El primero de los elementos es una cruceta de color rojo que se encuentra siempre en el centro de la cámara que sirve para indicar al usuario a donde está apuntando el arma en ese momento.

El segundo de los elementos es un número en la esquina superior izquierda de la pantalla que se corresponde con el número de balas que tiene el arma en su cargador.

Estos elementos se configuran en único *Blueprint* denominado *FirstPersonHUD*. Este *Blueprint* se inicia con el propio juego y se activa con el bloque *RecibirTrazoHUD* que se activa cuando existe algún trazo programado en el juego. La primera sección del programa realiza la función de colocar la cruceta en el centro de la pantalla, para eso utiliza el bloque *TrazarTexturas* que permite colocar elementos fijos en la cámara. La información la recibe desde el evento que inicia el programa y una función que determina la variable de color (fig. 52).

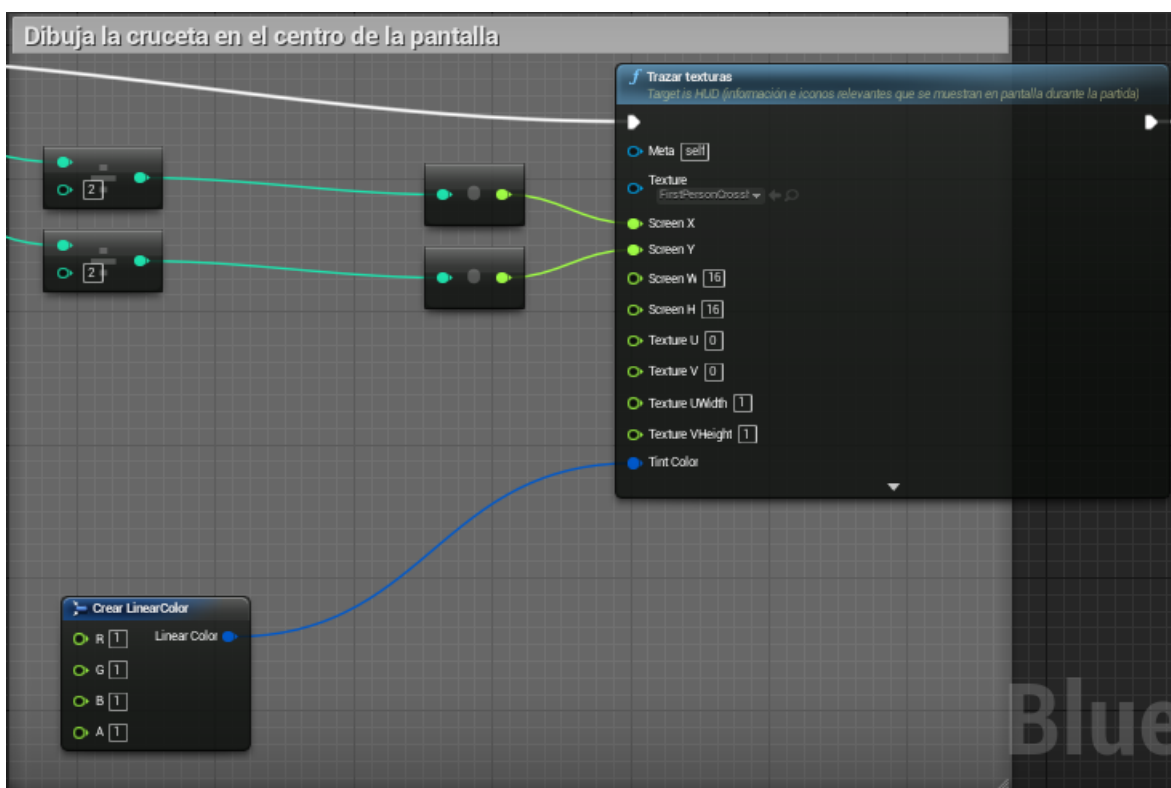


Figura 52: Sección del Blueprint del First Person HUD ocupada de dibujar la cruceta.

En la segunda sección del programa se lee el serial del que se están recibiendo los valores del arma y una vez separado el valor del cargador este se incorpora al bloque *TrazarTexto* en pantalla que se ocupa de localizarlo en su lugar correspondiente. No se realiza una modificación del color puesto que el negro tiene una connotación militar y concuerda con la fuente *Firefight* descargada para esta función (fig. 53)[65].

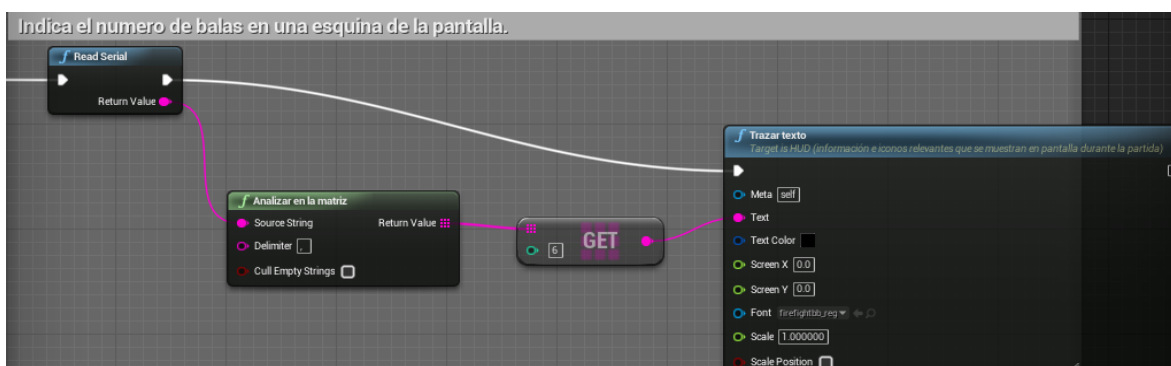


Figura 53: Sección del Blueprint que se ocupa de mostrar las balas del cargador.



PERSONAJE

Los controles y movimientos del personaje se encuentran en un Blueprint asociado al modelo con el que se identificara el usuario bajo el nombre *FirstPersonCharacter*.

Dentro del archivo de programación las diferentes funciones se agrupan en diferentes secciones.

La primera sección se inicia con el simulador gracias al bloque *BeginPlay* y su función es abrir al motor gráfico el puerto COM al que se encuentra sincronizado el ordenador para que este pueda empezar a recibir información (fig. 54).

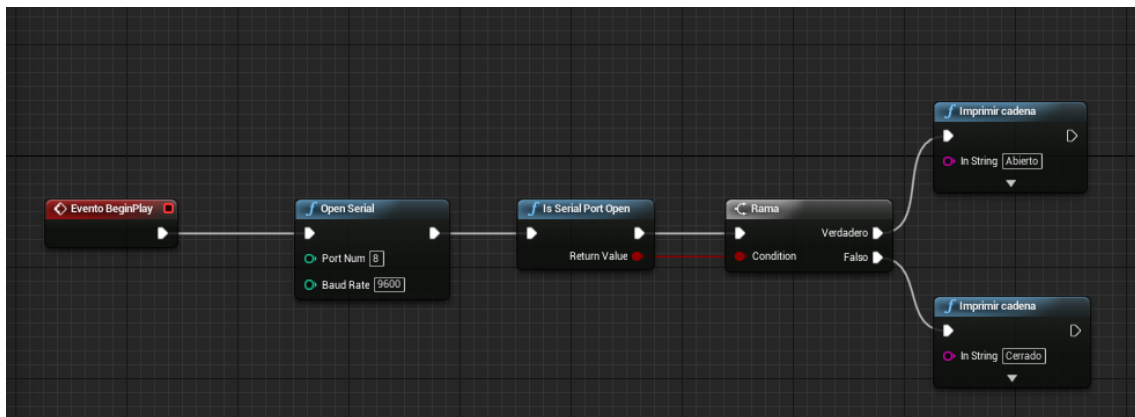


Figura 54: Sección del Blueprint responsable de abrir el puerto de comunicación serie.

Las funciones relacionadas con la apertura y comunicación con los puertos COM viene dada por un *plugin* ofrecido en los foros de *Unreal Engine* por el usuario “grizly32” denominado UE4duino. El inconveniente que presenta este *plugin* es que no es *buildable*. Esto implica que no es posible realizar un archivo ejecutable al final del proyecto que cuando se lance muestre directamente el simulador funcional, sino que es necesario tener instalado el *Unreal Engine* con este *plugin* y lanzar el simulador desde la reproducción del motor [66].

La segunda sección también se encuentra relacionada con el *UE4duino* y es la orden de cerrar el puerto serie cuando se termine de ejecutar la simulación. Esto es necesario debido a que de quedar abierto el puerto cuando esta finalice, al intentar volver a ejecutar el programa no se podría establecer conexión porque consideraría que el puerto ya se encuentra en uso (fig. 65).

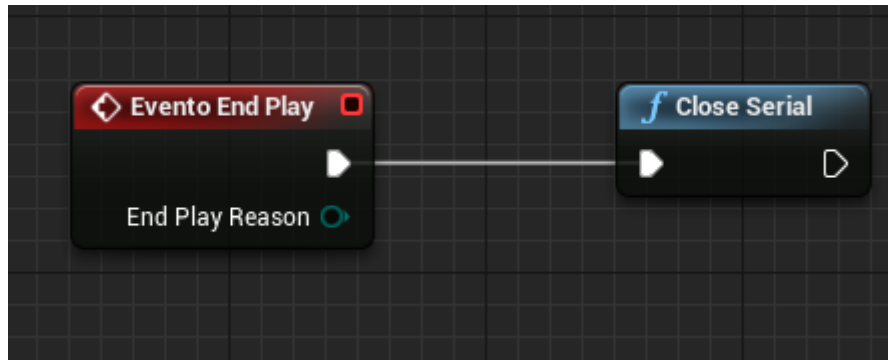


Figura 55: Sección del Blueprint responsable de cerrar los puertos al finalizar la comunicación.

La tercera sección es donde se gestiona la comunicación con el Arduino. Para ello se utiliza la función *ReadSerial*. Esta función lee toda una cadena de mensaje recibida por comunicación serie hasta que detecta un salto de línea. Acto seguido, archiva todo el mensaje leído como una cadena de caracteres, así que para poder separar los valores de variables es necesario saber qué posición ocupan dentro de la cadena y establecer algún elemento separador en el medio de esta que se pueda analizar mediante la función *AnalizarEnLaMatriz*. Esta función separará las cadenas de caracteres en cadenas más pequeñas consistentes en las variables aisladas.

Una vez aisladas los valores de los elementos de la cadena de caracteres se asignarán al tipo correspondiente de variable al que pertenecen para poder trabajar con ellas en otras secciones del simulador (fig. 56).

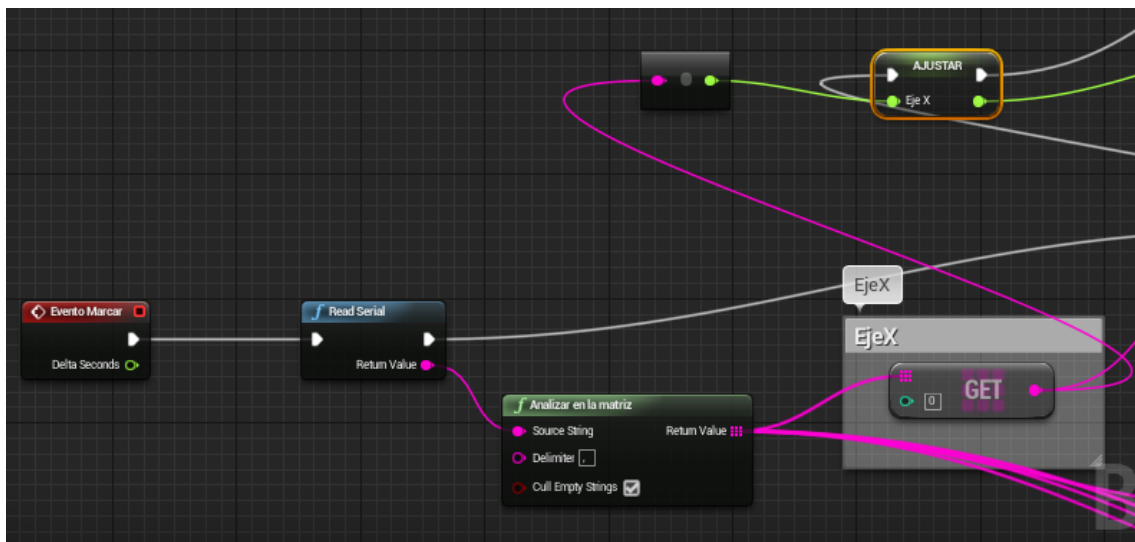


Figura 56: Fragmento de sección del Blueprint donde se está separando y guardando el valor correspondiente a la variación de movimiento en el eje X.



En la siguiente sección de programa se realiza el control de movimiento de la cámara y con él, la orientación que posee el arma. Para esto se asignan movimientos a cada una de las variaciones del eje de la carabina según como se encuentra colocado el giróscopo dentro de ella.

Para mover el arma en el eje vertical se asigna la variación del eje X a la función *Añadir entrada de ángulo del controlador* y para la rotación del usuario sobre sí mismo en sentido horario u anti-horario se asignan las variaciones del eje Y y Z al bloque *Añadir guiñada del controlador*.

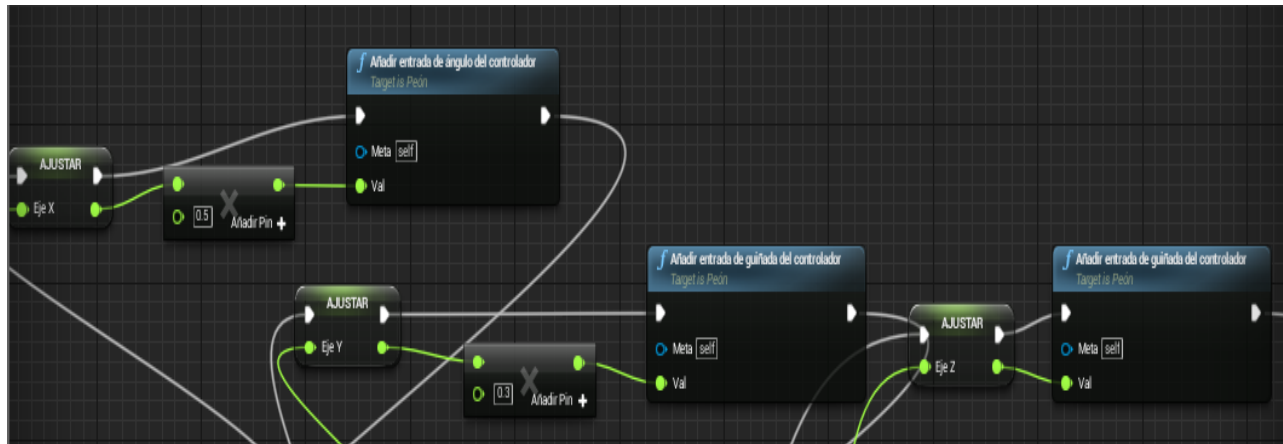


Figura 57: Sección del Blueprint asociada al movimiento de la cámara.

Estos valores no se introducen directamente a la entrada del bloque si no que han de pasar antes por un multiplicador que permite ajustar la sensibilidad de variación de cada variable antes de ser procesada en el motor. Esto permite graduar la sensibilidad que desea el usuario para su arma y que se ajuste a su movimiento de forma más fluida (fig. 57).

El movimiento del personaje en el simulador se realiza mediante una función que solo se activa cuando el motor detecta que se va a producir un avance por parte del usuario denominada *InputAxis MoveForward*. Puesto que la variable de movimiento funciona a través de la modificación de su valor cuando se produce una variación en el mismo esta función solo se activará cuando el usuario decida avanzar. Esta función va conectada al bloque *AñadirEntradaDeMovimiento* que es el bloque que realiza el avance de forma efectiva, al que han de ir conectados la variable que designa cuanto se ha de avanzar y una función que nos permite obtener la dirección en la que se encuentra orientado el personaje denominada *ObtenerElActorReenviaElVector* (fig. 58).

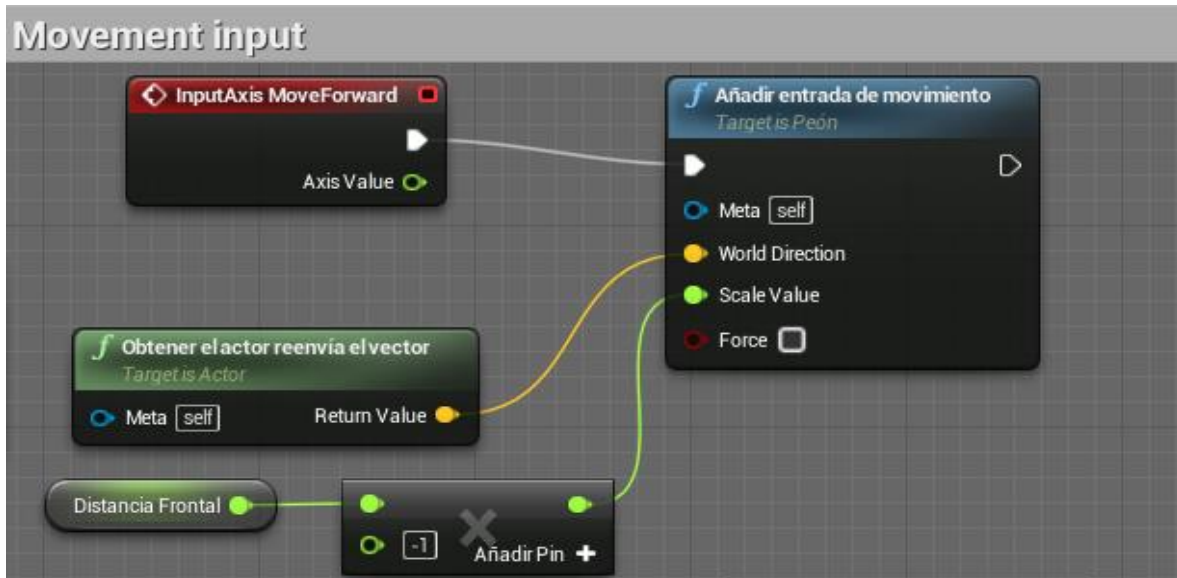


Figura 58: Sección del Blueprint responsable del avance del personaje.

Para configurar el salto se han de utilizar los bloques específicos que tiene el motor gráfico del juego para el salto de personajes. Estos han de ir asociados a la variable booleana definida en el Arduino para que mediante un *branch* se pueda controlar cuando se realiza el salto y cuando se ha de frenar (fig. 59).

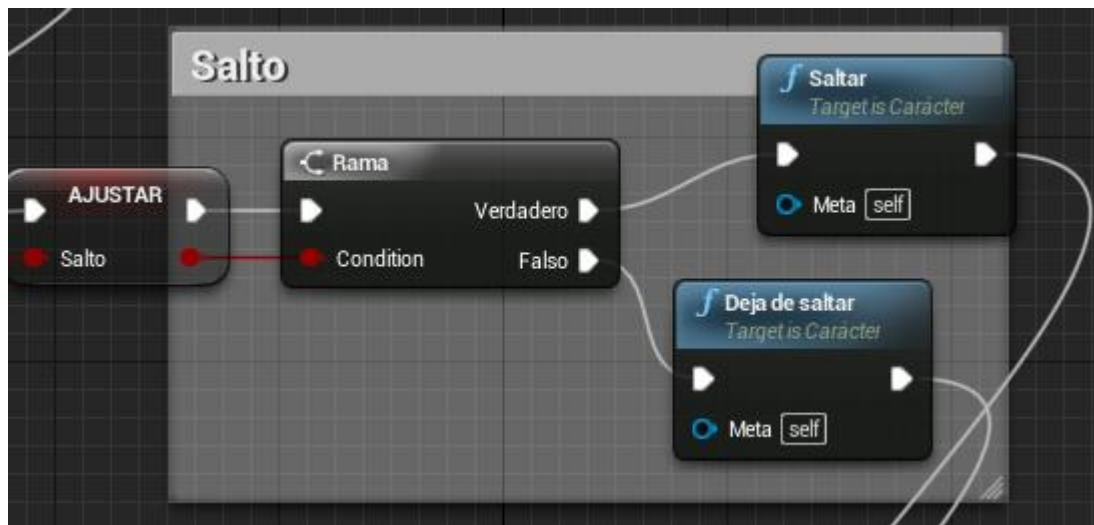


Figura 59: Sección del Blueprint responsable del salto.

La última sección del Blueprint del personaje está separada en dos partes y es la correspondiente a cuando se efectúa un disparo en el arma.

La primera parte es el equivalente a un detector de flanco que detecta cuando se ha realizado un disparo. Para ello se guarda la variable y en la siguiente ejecución del programa se revisa si esta ha cambiado con el fin de que el disparo se produzca una sola vez. Este



detector de flanco es necesario añadirlo al programa, a pesar de que ya existe uno dentro del Arduino, debido a que a las velocidades que trabaja el ordenador puede darse el caso de que ejecute el programa una segunda vez antes de que reciba la segunda entrada de datos por *Bluetooth* y entonces se reciba dos veces seguidas la señal de disparo (fig. 60).

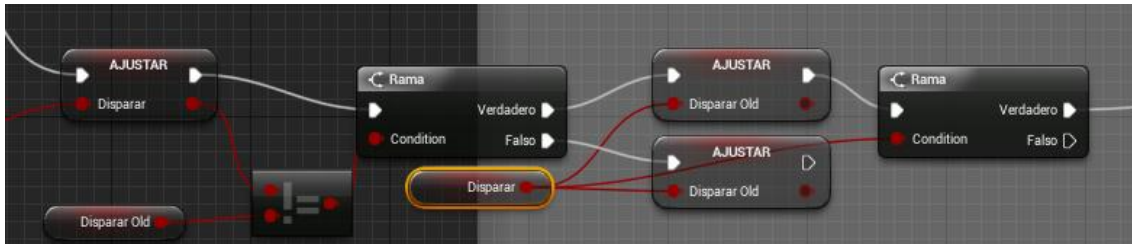


Figura 60: Parte del Blueprint de la sección de disparo que actúa como detector de flanco.

La segunda parte son las funciones necesarias por el programa para detectar donde se encuentra el actor y el arma. Esto permite añadir un Offset al punto de aparición de disparo para que la bala de forma que esta no surja directamente desde el cuerpo del modelo, sino que lo haga en donde se encuentre la punta del cañón del arma (fig. 61).

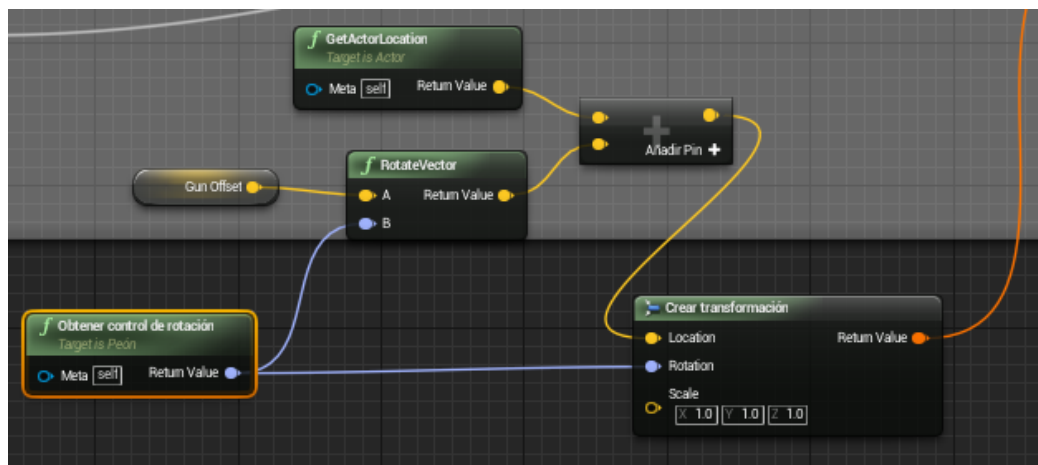


Figura 61: Sección del Blueprint que localiza al actor y obtiene un punto en función del modelo.

Una vez se obtiene toda esta información se introduce en la función específica *SpawnActor First Person Projectile* que realiza la creación del modelo de bala justo en la punta del arma. Cuando esto se produce se realiza una llamada a la función *ReproducirSonidoEnLaUbicación* que provista de la información de donde se encuentra ubicado el actor, reproduce el sonido de disparo que se le ha provisto al arma (fig. 62).

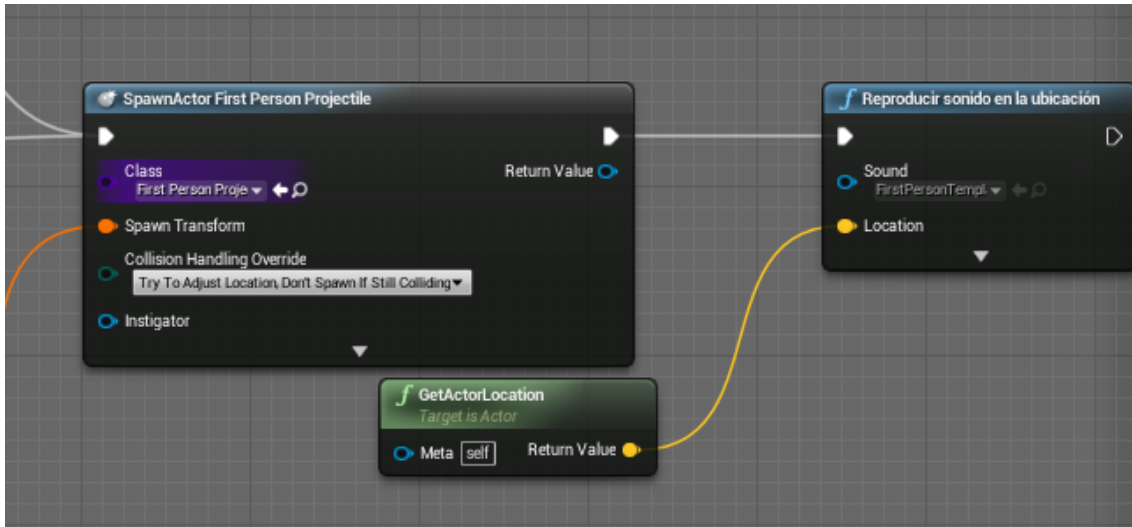


Figura 62: Sección del Blueprint encargada de generar la aparición de la bala.

BALA

La bala utiliza un modelo de pelota de goma creado mediante los BSP que provee el *Unreal* a la que se le aplica una textura que da la sensación de ser una pelota de goma.

Este actor tiene asociado un *Blueprint* para que cuando se genere se le dote de una velocidad y una fuerza para que actúe contra los elementos dinámicos del sistema. Para ello lo primero es asignarle un área de colisión tendrá un volumen ligeramente superior al de la pelota. Esto se realiza anidando un *colision component* de tipo *sphere* como se ve en la figura 63.

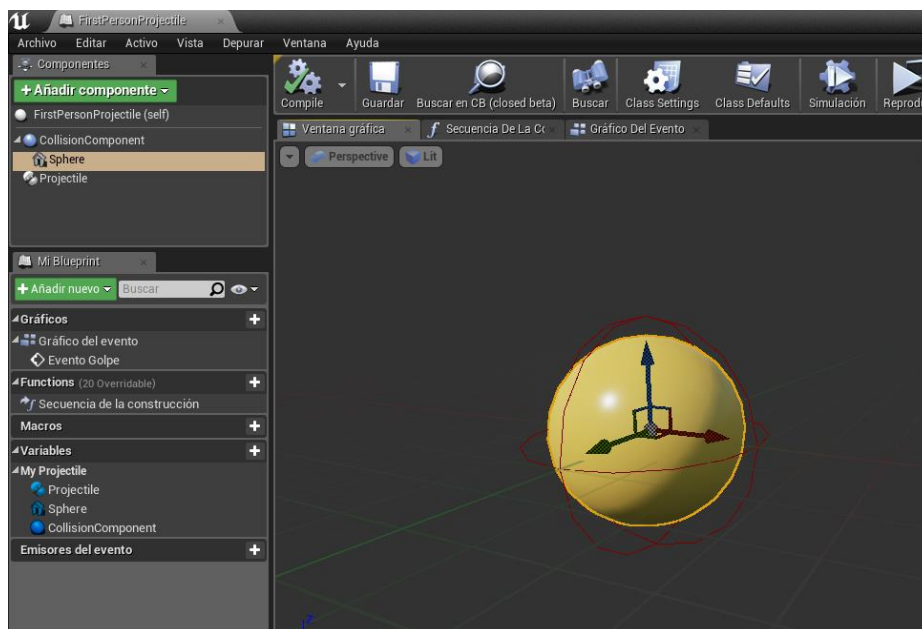


Figura 63: Imagen del volumen de colisión (en rojo) asignado al proyectil.



La configuración del *Blueprint* se realiza de forma que este se active cuando el modelo detecte un contacto con algún otro actor, comenzando por el que va a tener con el arma del usuario que va a ser lo que le provea de una velocidad inicial.

El bloque *AñadirImpulsoEnLaUbicación* se ocupa de proyectar la bala con la dirección y velocidad que se le ha indicado. Este bloque da lugar al *AplicarDaño*, esta función se ocupa de dañar el modelo de forma que pasados unos segundos después de haberse creado el actor este se destruya a posteriori por habérsele causado el daño suficiente dando lugar al bloque *DestroyActor* (fig. 64).

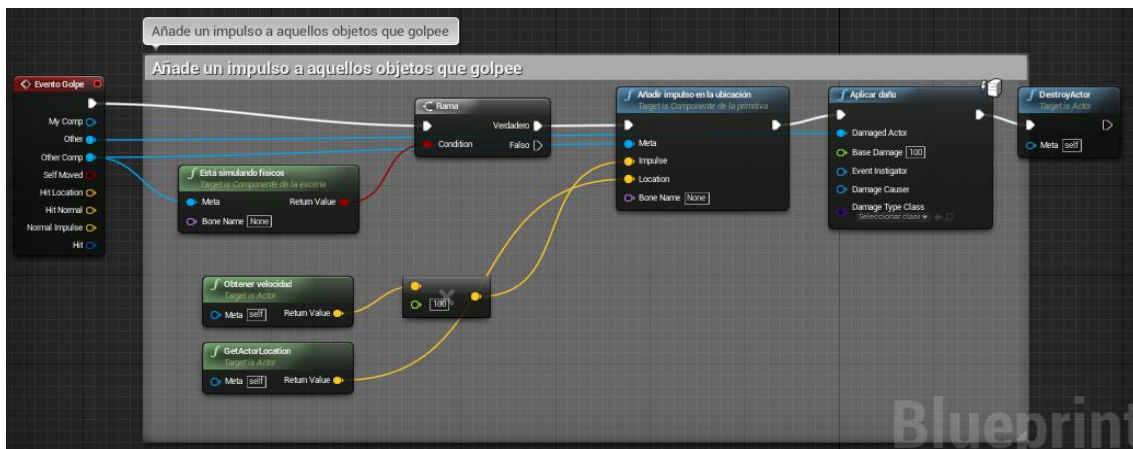


Figura 64: Blueprint asociado al First Person Projectile.

PRIMERA PERSONA

Para que el programa realice la simulación con una vista en primera persona, es necesario asignarle al actor que realiza la función de personaje una cámara que es la que servirá como visor del juego. Una vez hecho esto es necesario crear un *Blueprint* de modelo de juego que provee el propio motor gráfico y realizar la configuración que se ve en la figura 65.

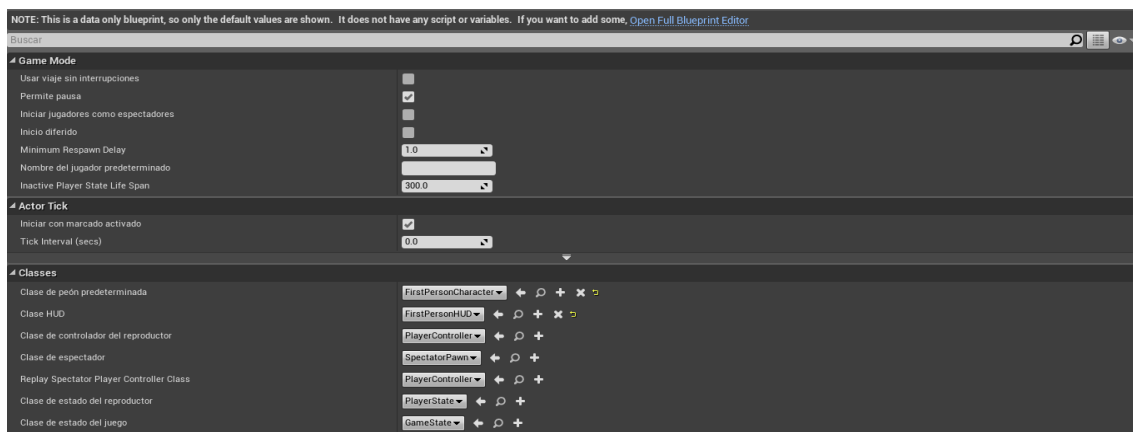


Figura 65: Configuración realizada sobre el modo de juego.



CAPÍTULO:8

PRUEBAS





Pruebas de desapuntado

Uno de los requisitos a la hora de plantear este proyecto era el desapuntado natural del arma debido a las fuerzas ocasionadas por el retroceso de la misma. Por este motivo se han de desarrollar unos ensayos que permitan comprobar que después de disparar es necesario seguir moviendo el arma para corregir la desviación provocada por la misma.

En este caso se trata de poder demostrar que el arma al disparar pierde precisión por lo que lo que se busca en estos ensayos al contrario de lo que pueda parecer lógico es que la distancia entre disparos si no se trata de corregir la precisión sea lo más acusada y visible posible.

Puesto que el arma tiene dos sistemas de desapuntado será necesario la realización de dos pruebas distintas comprobando el valor de cada uno de los métodos por separado para saber si de este modo es posible validarlos como un sistema aceptable de retroceso, más allá de la fuerza soportada por el arma.

- **Prueba de desapuntado natural**

Se ha realizado una grabación del arma sostenida por un usuario estándar en el que se le ha pedido que no mueva el arma de su posición de partida una vez se inicie la simulación y que no realice más esfuerzo que el sostener la misma y recargarla mediante el botón provisto en la parte superior del arma. Se realizarán 20 disparos para comprobar el rango de movimiento que tiene el arma por sí misma en esa situación.



Figura 66: Área en la que se encuentran contenidos todos los tiros de la prueba de desapuntado.

Comprobamos las dimensiones del área en el programa y vemos que se trata de una elipse con un diámetro mayor de 3 metros y un diámetro menor de 1 metro que se encuentran a 10 metros del personaje (fig. 66). Aplicando la fórmula de la tangente se obtiene que el



ángulo de desvío horizontal es de 24° de amplitud y de 8° de desvío vertical lo que para un arma de estas características es más que aceptable ya que lo típico para ellas se trata de un 20% lo que se ve cumplido gracias al ruido de medida del giróscopo y la vibración obtenida en el agarre del gatillo y el apoyo en el hombro de los motores de DC.

- ***Prueba de desapuntado por muelle***

Para la comprobación del desajuste producido por el retroceso que ocasiona disparar si se ha cargado el muelle se ha realizado la grabación de otra prueba grabada de donde se obtienen los resultados pertinentes para el análisis.

En la prueba se disponía de un usuario con el arma centrada en posición de equilibrio apuntando siempre a un mismo punto medio. A este usuario se le pediría que disparase 10 veces intentando siempre que este acierte en los cubos que se le presentan inmediatamente delante al principio del test. En caso de que el desajuste fuese demasiado grande se le permitiría mover el arma para que así pudiese reajustar la mira al objetivo que se le había puesto. Si el desajuste era demasiado grande se podría dar la prueba concluida antes de disparar todas las balas.

Se repitió la prueba 10 veces con 5 usuarios distintos con el fin de poder determinar si la diferencia de fuerza con la que era posible sostener el arma representaba un indicador importante del desajuste de la mira obtenido. En todos los casos ninguno de los sujetos de la prueba fue capaz de llegar a disparar las 10 veces que se les pedía. El mejor resultado lo obtuvo un varón caucásico de 1.95m y 87kgs que llegó a realizar 7 disparos mientras que el peor fue de una mujer caucásica de 1.71m y 62kgs que solo llegó a realizar 4 tiros.

En todos los casos el retroceso obtenido describía un desajuste vertical del arma que llegaba a ser de casi 180 grados en el peor de los resultados obtenidos ya que una persona apuntando completamente vertical hacía arriba encontraba su mira en la pantalla apuntando directamente al suelo (fig. 67).

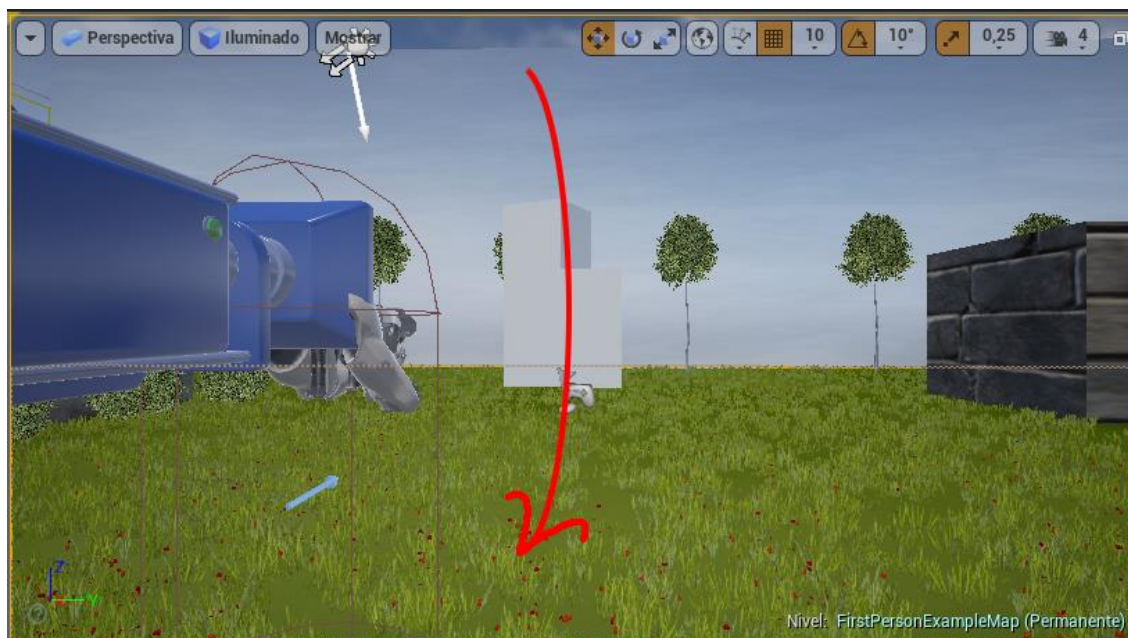


Figura 67: Trayectoria descrita por el arma debido al desapuntado producido por el muelle.

Se puede concluir que el desapuntado en este caso es más que suficiente y llega a producir un desajuste serio de la mira de apuntado virtual. Pero puesto que en el programa es viable realizar una reconfiguración del tipo de arma que se esté usando, este tipo de retroceso asociado al hecho de que para disparo es necesario volver a recargar permite realizar simulaciones no solo para carabinas y subfusiles sino también para rifles de francotirador y lanzacohetes.

Pruebas de oclusión

La oclusión se refiere a la obstrucción de un fluido por algún tipo de canal impidiendo su paso en su totalidad o al menos parcialmente. Lo que se pretende estudiar en este tipo de pruebas es el tipo de obstáculo que puede encontrarse por vía aérea nuestra señal sonora para así dar algún tipo de mal funcionamiento a los sensores de ultrasonidos.

Puesto que para el buen funcionamiento del arma esta debe utilizarse sin obstáculos entre ella y el proyector con el que se trabaje la prueba es meramente un compromiso para validar el grado de precisión que podemos encontrar.

La prueba consistirá en un usuario que mantenga fija el arma en una posición centrada a la altura del pecho apuntando en una línea recta a la simulación proyectada en la pantalla. Mientras este usuario se mantiene fijo, un asistente usará dos planchas planas de cualquier material que no tenga un alto grado de absorción sónica, como puede ser el corcho. Estas dos planchas se colocarán paralelas entre sí a 50cm de distancia con la línea de trayectoria de disparo en medio y se desplazarán hacia delante y atrás con el fin de

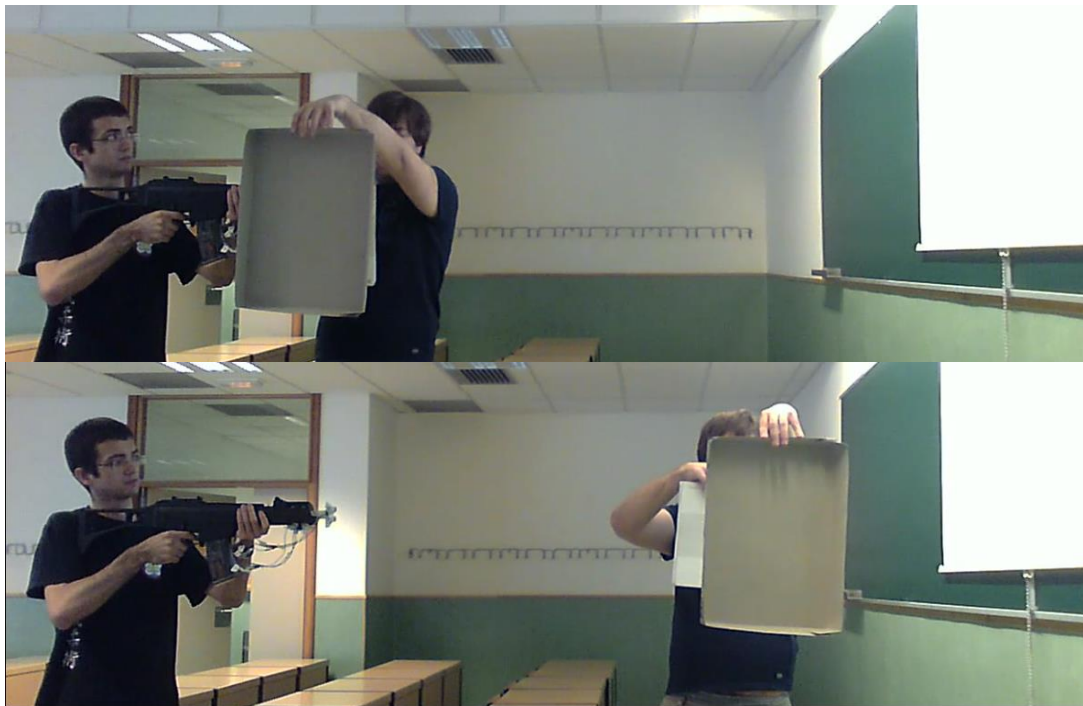


comprobar si la señal se queda atrapada entre ellas para así retornar al sensor a pesar de que no le corresponda y así dar una falsa sensación de movimiento (fig.68).

Esto se realizará tanto como para el sensor frontal como el vertical a pesar de que el vertical posee la condición de necesitar estar a más de dos metros de altura para que se obtenga algún tipo de respuesta.

Después de realizar la prueba se llega a la conclusión de que la única interacción se realiza cuando las planchas están colocadas enfrente del proyector por lo que en vez de tratarse de una oclusión de la señal que rebota de forma errónea se trata más bien de un mal funcionamiento debido a la aparición de unos obstáculos dentro del rango de rebote de la señal sonora que provoca un fallo en la onda de vuelta haciendo que unas veces perciba una distancia y otras veces otra y puesto que el funcionamiento del sistema de movimiento es mediante la detección de la variación de la posición eso provoca el baile de distancias dentro de la simulación (fig. 69).

En el caso del sensor vertical no existe problema de mal funcionamiento en el mismo pues para que este se produzca los obstáculos inferiores que puedan aparecer no se tratan de otros más que las piernas del usuario y no tienen un coeficiente de absorción sónica lo suficientemente alto como para retrasar la onda lo suficiente como para que se pueda considerar que existen dos metros entre el suelo y el sensor de ultrasonidos.



Figuras 68 y 69: Situación inicial de las planchas para la oclusión y situación de las mismas donde se produce conflicto en el rebote de la onda sonora.



Pruebas de repetibilidad

La puntería se define como “la destreza del tirador para dar en el blanco” en este caso será necesario para el arma que cuando una persona apunte con la misma el fallo que pueda haber al darle al objetivo se deba a la falta de puntería de ese usuario y no al hecho de que sea el propia arma la que carezca de dicha puntería y sea necesario rectificarla.

Es aceptable cierto grado de fallo en la precisión del arma ya que en condiciones normales existen diversos factores que pueden afectar a que el disparo de cualquier tipo de subfusil se vea desviado: como puede ser una ligera curvatura en el cañón o la existencia de corrientes de aire. Es por este motivo que si el arma tiene un desvío debido al error de los componentes electrónicos si este es lo suficientemente pequeño es aceptable, puesto que se debe a una característica inherente al propio aparato.

Para la realización de la prueba de repetibilidad se inmovilizará completamente el arma apoyándola contra una superficie y se inicializará la simulación. Puesto que el arma en la simulación siempre empieza en la misma posición independientemente como se encuentre está en la vida real lo que se pretende es eliminar cualquier tipo de variación en el movimiento de la misma incluida la provocada por cualquiera de los retrocesos del arma (fig. 70).



Figura 70: Arma inmovilizada contra la mesa para asegurar la ausencia de vibraciones, una mano se colocara en el gatillo y otra en el boton de recarga.

Una vez que el arma se encuentra inmovilizada se realizaran 40 disparos y se verificará que la dispersión entre los mismos es lo suficientemente pequeña para que sea aceptable el grado de precisión que presenta el arma.

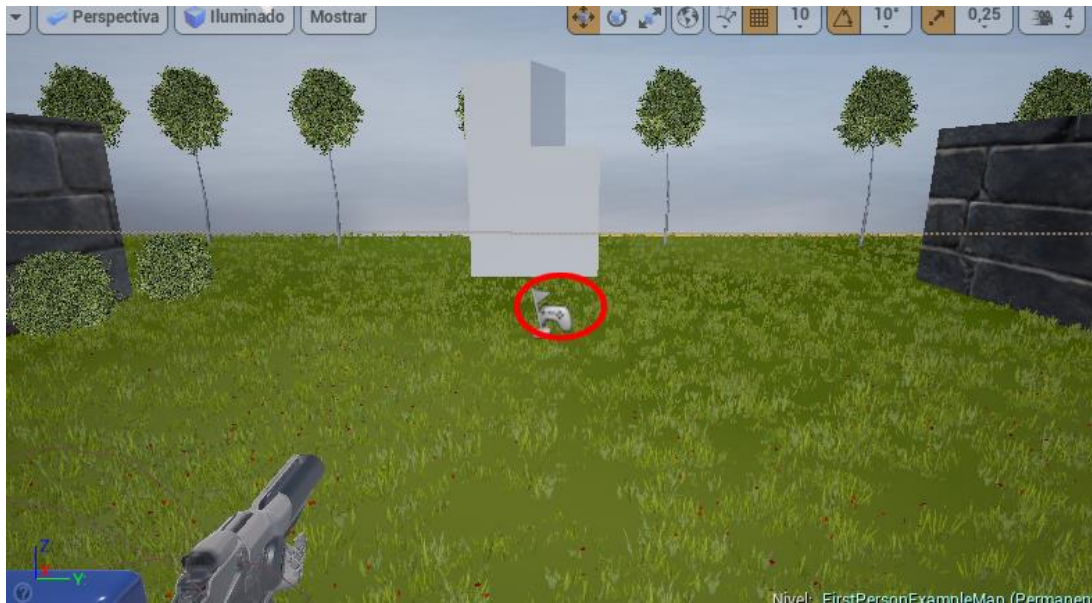


Figura 71: Área localizada donde se han producido los 40 disparos en la simulación

Se puede comprobar después de la prueba realizada que la desviación provocada en el arma es casi mínima siendo lo suficientemente aceptable como para que la precisión y repetibilidad de la misma sean consideradas propias de un modelo real.

En este caso el radio del círculo donde las balas se encontraban no supera el metro y se encuentra a una distancia simulada de 13m por lo que aplicando la tangente nos encontramos que la desviación máxima es de 4º para 5 cargadores lo que demuestra que el disparo repetido cuenta con una puntería aceptable (fig. 71).

Pruebas de autonomía y funcionamiento

Se han realizado comprobaciones prácticas del simulador funcionando de manera activa a lo largo del tiempo para poder validar su uso a lo largo de sesiones intensas y/o largas de simulación como suelen ser las propias de prácticas militares/sesiones de juego.

Para la autonomía se ha dejado encendida el arma durante un periodo de tiempo de 12 horas y se comprueba que esta sigue funcionando sin problemas a pesar de no poder interrumpirse nunca la comunicación bluetooth, los sensores de ultrasonidos y el giróscopo (no existe un modo stand-by, mientras esta encendida equivale a estar jugando a pesar de no pulsarse los botones). A partir de las 20 horas de uso el bluetooth empieza a ocasionar problemas si se enciende y apaga el arma, pues no recibe la intensidad suficiente para su funcionamiento y a partir de las 24h no se encenderá por lo que será necesario el recambio de la fuente de alimentación.



En el caso de los motores se han conectado directamente a la batería que los alimenta y se han dejado así hasta que esta ha sido insuficiente para que estos funcionen. Cosa que se produce superadas las 6 horas ininterrumpidas de uso.

En caso de que suceda que la alimentación se termine, simplemente es necesario abrir el arma para cambiar las pilas que alimentan los diferentes módulos.

En el caso del funcionamiento no autónomo, se ha dejado conectada el arma mediante USB a un ordenador personal al que se le ha desactivado el modo suspensión para asegurarse de que la prueba no se ve interrumpida.

Bajo estas condiciones se mantuvo 48 horas en funcionamiento el arma probándose en ciclos de 8 horas para comprobar que en caso de iniciar una simulación esta funcionaba correctamente. El resultado de las pruebas fue satisfactorio y se comprobó que en caso de ser necesario es posible mantener funcionando el arma durante periodos de varios días renunciando a la portabilidad y autonomía de movimiento al verse esta limitada por la presencia del cable.





CAPÍTULO 9: FUNCIONAMIENTO





El proceso para poder poner en marcha el simulador no requiere de conocimientos técnicos o propios de una formación especializada.

El primero de los pasos es encender el arma en el botón que se encuentra en el cargador. El indicador de que el arma se encuentra encendida es el parpadeo que vendrá por parte del módulo de *Bluetooth* situado en el interior del cargador.

Una vez el arma se encuentre encendida es necesario sincronizarla con el ordenador en el que se vaya a ejecutar la simulación. Para ello se encenderá el *Bluetooth* del ordenador y se realizará la búsqueda de dispositivos. Una vez se encuentre el dispositivo “ArmaBluetooth” se procederá al emparejamiento. El arma pedirá un código de identificación para poder sincronizarse. En caso de que el usuario no haya cambiado el código al configurar el *Bluetooth* de su arma el código de emparejamiento será “1234” (fig. 72).

Cuando se finalice el emparejamiento será necesario comprobar a cuál de los puertos se ha sincronizado el *Bluetooth* (fig. 73). El *Bluetooth* se sincroniza por defecto salvo que haya algún tipo de dispositivo en esos puertos COM con los puertos 5 y 6 siendo el que se configure como puerto saliente el que será motivo de atención. En el caso del ordenador en el que se ha realizado el proyecto se han tenido que usar los puertos 7 y 8 debido a que los 5 y 6 estaban ocupados por el puerto de red Ethernet.

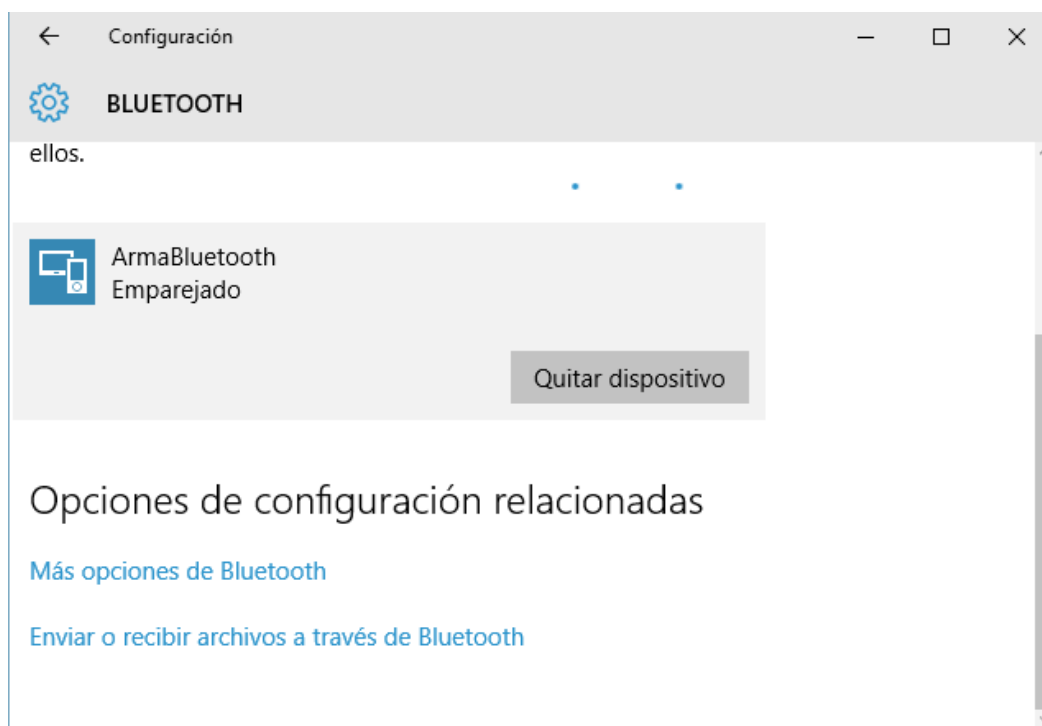


Figura 72: Menú de configuración de Bluetooth en Windows 10. Pulsando más opciones de Bluetooth surge la figura 73.

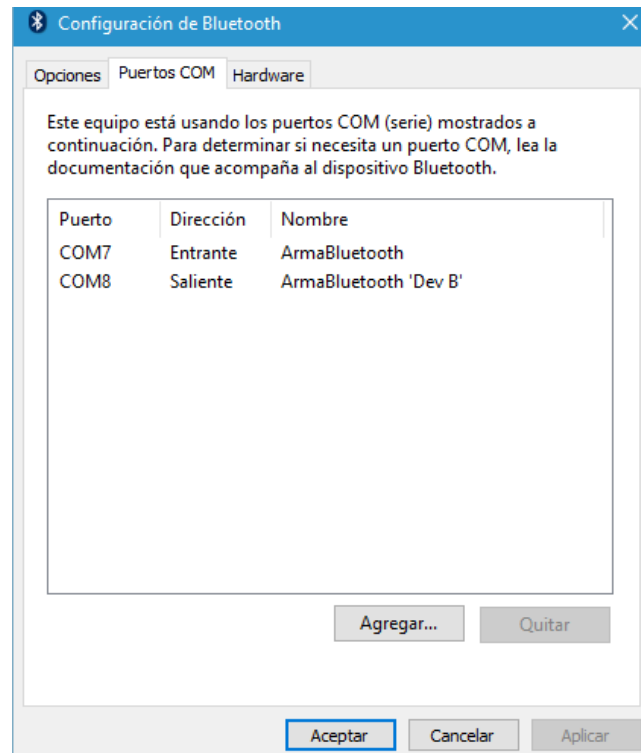


Figura 73: Menú donde se muestra la configuración de los puertos del arma Bluetooth.

Este paso solo es necesario la primera vez que se trabaje con el arma en un ordenador. Una vez el arma se ha sincronizado se conectará automáticamente con ese ordenador cuando este tenga activado el *Bluetooth*.

Una vez se ha comprobado a que puerto se encuentra conectado el arma de *Bluetooth* el siguiente paso es encender el *launcher* de *Epic Games* con nuestra cuenta y contraseña. *Epic Games* es una compañía que entre su gran variedad de productos se encuentran videojuegos además de programas de diseño. Por este motivo se mostrará un menú de inicio con todos los programas de *Epic Games* a pesar de que el único requisito para la ejecución de nuestra simulación es tener instalado el motor gráfico que provee esta compañía.

Para iniciar el motor gráfico se debe pulsar en la pestaña correspondiente al *Unreal Engine 4* mostrada como UE4 en la interfaz del *launcher*. Una vez el usuario se encuentra en la pestaña de *Unreal Engine* deberá pulsar en el botón correspondiente a la biblioteca donde se encuentran todos los proyectos con los que cuenta ese usuario (fig. 74).

El proyecto del simulador se provee en una carpeta que deberá estar encontrarse en la localización en la que el usuario tenga determinado la biblioteca de proyectos para que aparezca en el *launcher*.

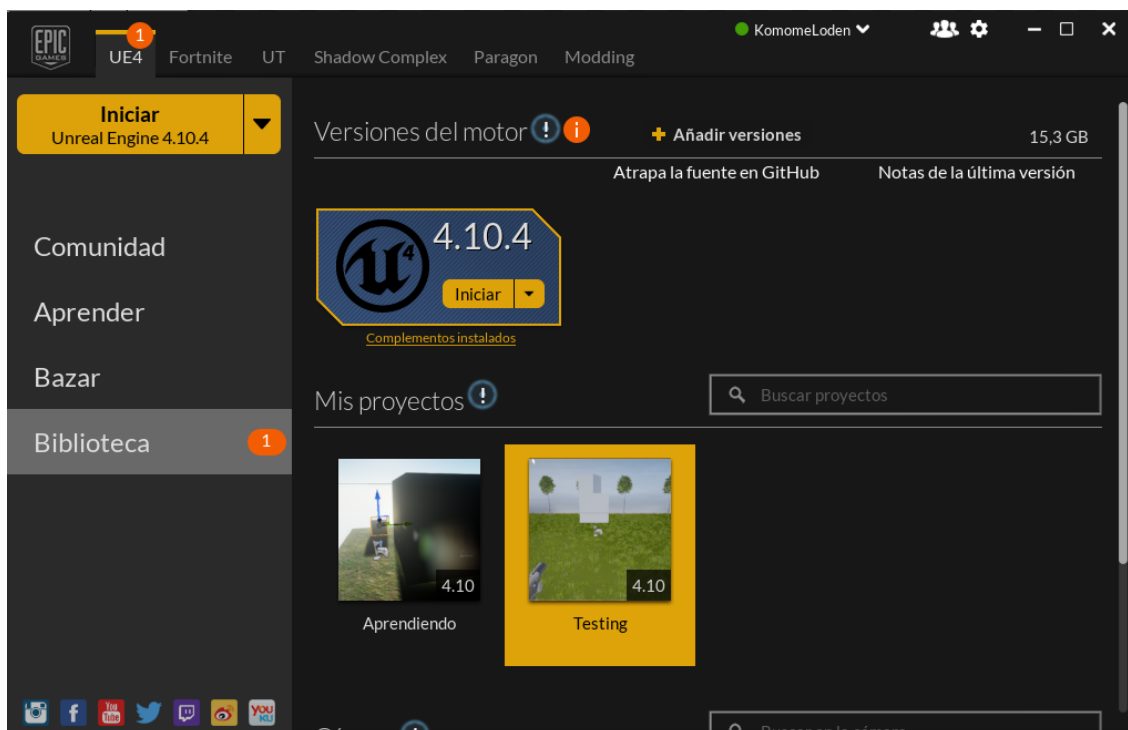
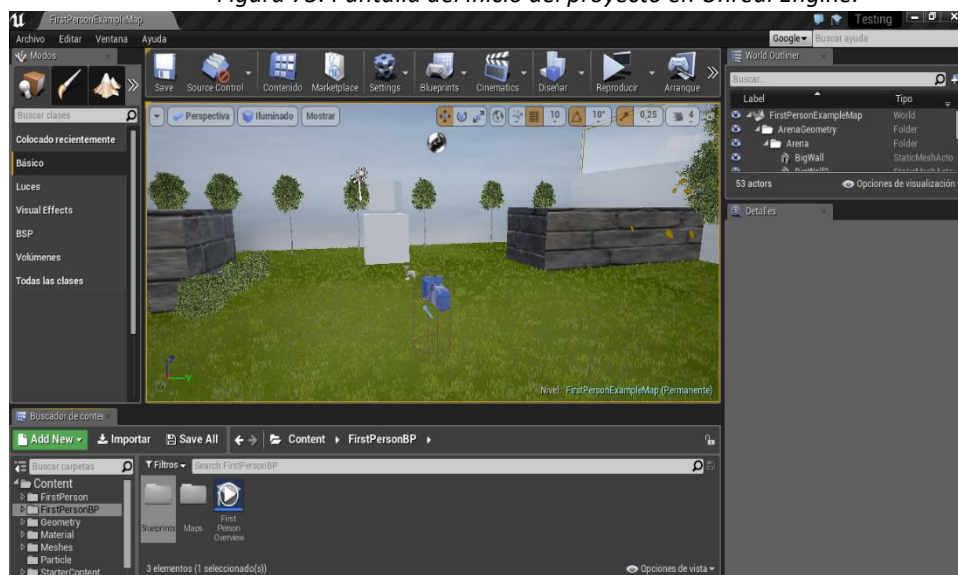


Figura 74: Launcher de Epic Games Company

Haciendo doble clic sobre el proyecto se lanzará el motor gráfico apareciendo la interfaz desde el punto en el que se la haya dejado al guardar. En este caso aparecerá siempre sobre el punto en el que se ha definido al jugador y a su vez punto de partida del juego (fig. 75).

Figura 75: Pantalla del inicio del proyecto en Unreal Engine.





Una vez en este menú será necesario navegar por el gestor de contenido pulsando en la carpeta FirstPersonBP (fig. 76).

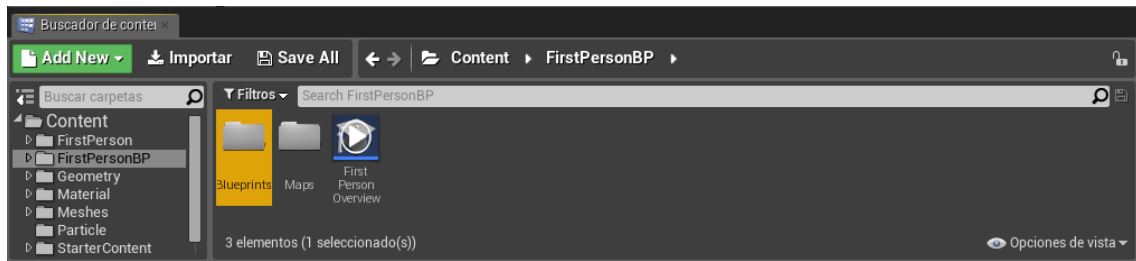


Figura 76: Gestor de contenido en la carpeta FirstPersonBP.

Desde ese punto se accederá a los Blueprints del programa y se abrirá el correspondiente al First Person Character (fig. 77) (fig. 78).

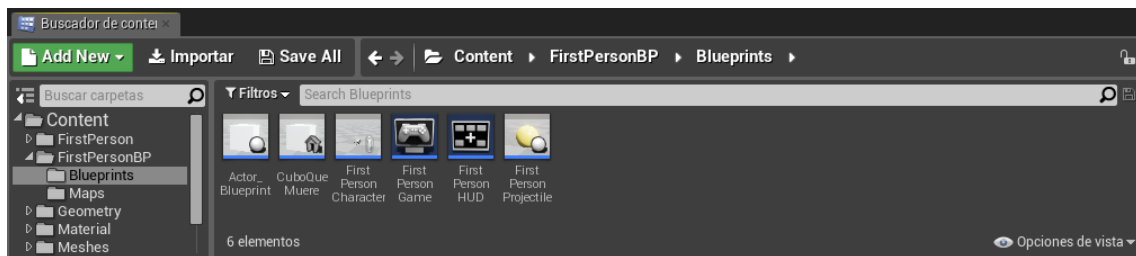


Figura 77: Gestor de contenido en la carpeta de Blueprints

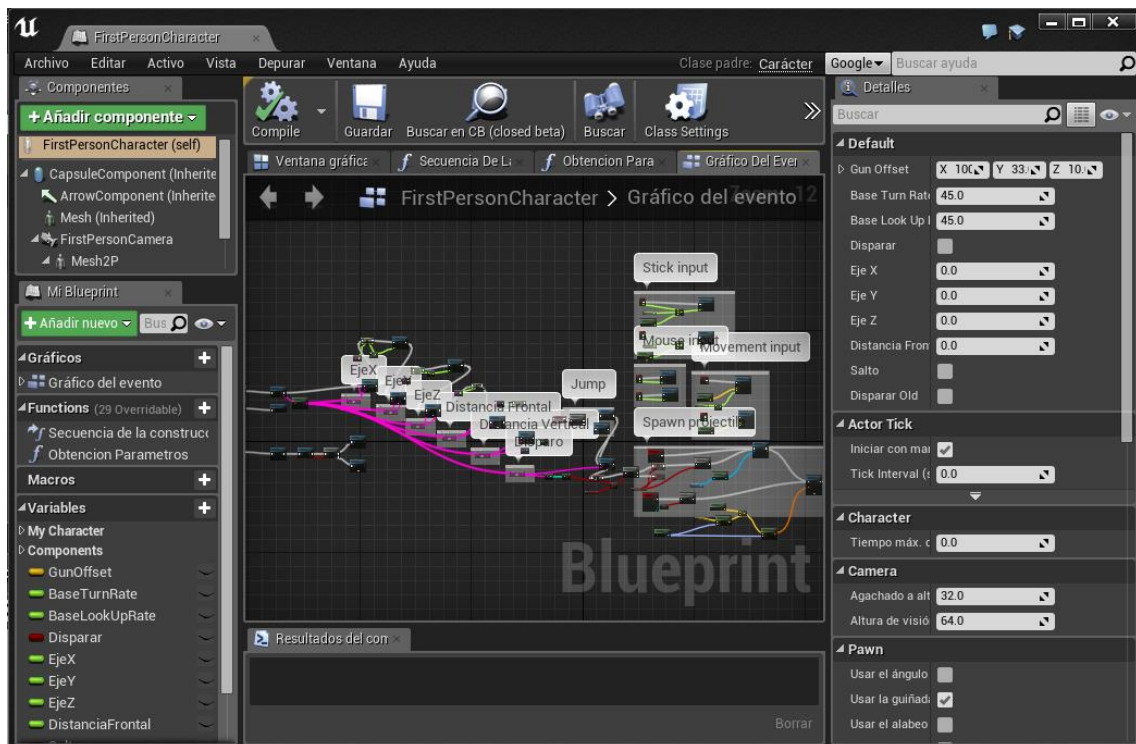


Figura 78: Ventana emergente del First Person Character.



Una vez dentro de la configuración del personaje se buscará el bloque con el nombre *OpenSerial*. Para desplazarse por la interfaz se usará el *scroll* del ratón para aumentar y reducir el zoom y el botón derecho del ratón para mover lo que se encuentra en pantalla. Una vez localizado el bloque se comprobará que el puerto corresponde al saliente que se identificó en el menú de *Bluetooth* y la velocidad es la misma que se configuró en su momento en el módulo de comunicaciones, en este caso debería ser 9600. En caso de que no coincidan se modificarán por los valores correctos (fig. 79).

Para el funcionamiento del programa es necesario que el motor gráfico reconozca los bloques correspondientes a la apertura de puertos del ordenador. Para ello, el usuario debe tener instalado en su motor gráfico un *plugin* desarrollado por el usuario “grizly32” de los foros de *Unreal Engine* denominado UE4duino.

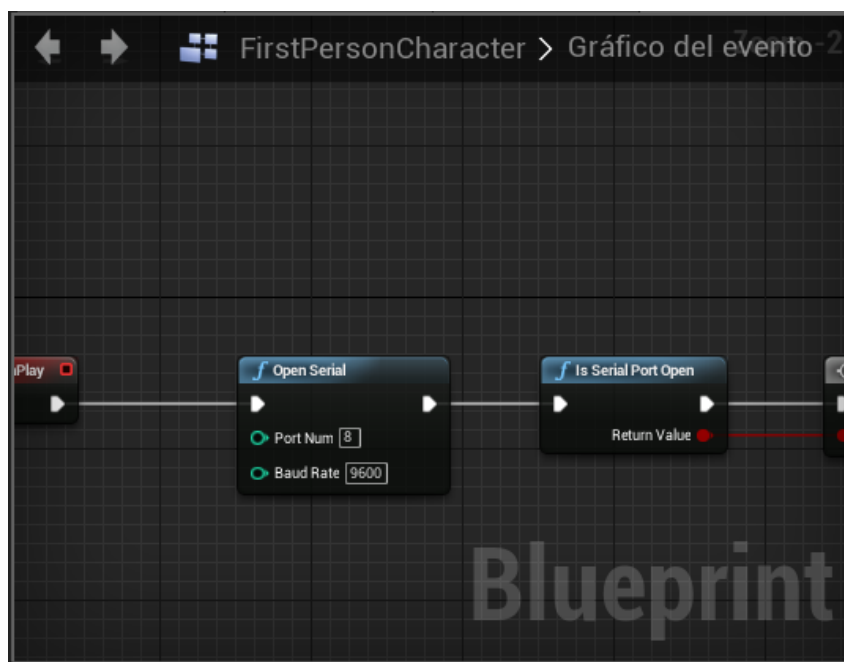


Figura 79: Bloque blueprint *Open Serial*.

Esta verificación al igual que la del *Bluetooth* una vez se ha realizado por primera vez no es necesario volverla a hacer si se guarda el programa después de haberlo modificado.

Una vez realizado todas estas verificaciones es posible encender el simulador. Para su correcto funcionamiento se recomienda estar en una sala de 4mx4m y trabajar con un proyector, si bien es viable hacerlo con un ordenador normal en una sala cualquiera mientras no exista ningún tipo de obstáculo entre el usuario y las paredes o variaciones a media altura provocadas por elementos como sillas, mesas, estanterías, etc.



Para encender el simulador es necesario desde la pantalla principal de *Unreal Engine* pulsar el botón reproducir, que salvo modificación por parte del usuario de las ventanas de *Unreal* se encontrará encima a la derecha de la previsualización del nivel (fig. 80).

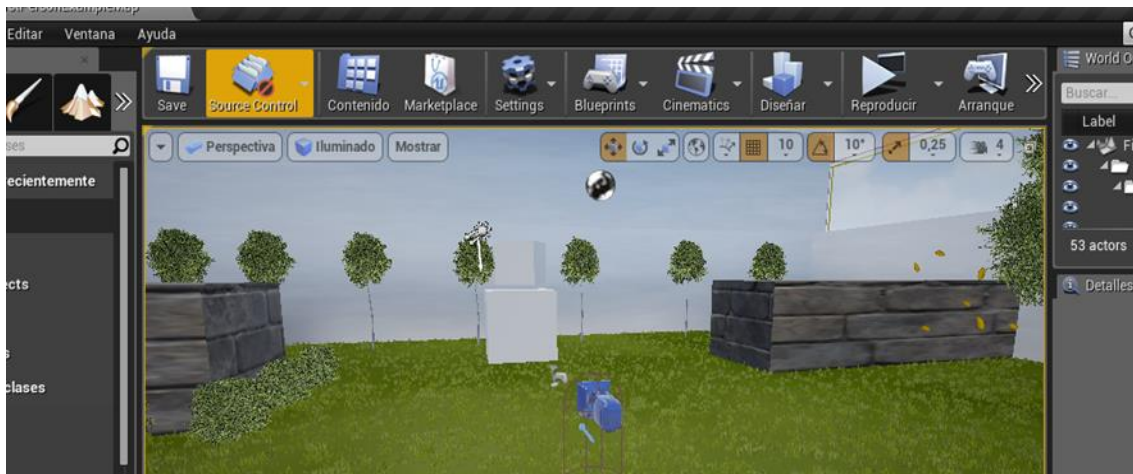


Figura 80: Previsualizador del nivel donde se aprecia el botón reproducir.

Una vez pulsado ese botón se abrirá una nueva ventana donde se muestra la simulación. Destacar que la posición en la que se encuentre el arma una vez lanzada la simulación es la que se considera como de equilibrio y corresponderá a un apuntado de frente paralelo al suelo en lo que refiere a la simulación. Una vez abierta los controles de la misma son los siguientes:

- **Avanzar:** El sistema permite el avance lineal hacia delante y atrás, para realizarlo el usuario solo tiene que caminar hacia delante o hacia atrás para realizar dicho movimiento dentro de la simulación.
- **Saltar:** El salto de la simulación se reconoce cuando la posición del arma es a más de dos metros del suelo. Para ello el usuario si tiene bien colocada el arma simplemente tendrá que saltar para que se le reconozca el salto. En caso de contar con un usuario de estatura reducida, como puede ser un infante, tendrá que elevar el arma con sus brazos a la altura requerida para realizar el salto.
- **Disparo:** Para disparar el usuario solo tiene que apretar el gatillo del arma que tiene en las manos.
- **Recarga:** El arma solo cuenta con 8 balas así que para recargar es necesario pulsar el botón que se encuentra debajo del riel de la mira justo enfrente de la palanca de recarga mecánica.
- **Apuntado vertical:** Para cambiar la altura en el eje de la Y de aquello a lo que este apuntando el arma, el usuario solo tendrá que variar la posición de la boca del arma en dicho eje inclinándola.
- **Apuntado horizontal:** Para cambiar la orientación del arma en el eje de las X será necesario que el usuario haga roll del arma sobre el eje en el que se



encuentra situado el cañón. Si se realiza giro en sentido horario el arma empezará a apuntar hacia la derecha y si es en sentido antihorario lo hará hacia la izquierda.

El entorno de simulación se corresponde con un jardín de 50mx50m delimitado por 4 muros de cerámica en el que se pueden encontrar 14 árboles de distintos tamaños y 10 arbustos algunos de los cuales cuentan con un sistema de simulación de partículas dando la sensación de que se les caen las hojas. Por el mismo se encuentran colocados también 6 bloques blancos que son elementos dinámicos. Cuando a estos bloques se les dispara se les transmite la energía cinética de la bala siendo posible sacarlos de su posición. Según las dimensiones de los bloques también cambia el peso que poseen siendo los más voluminosos aquellos con más peso. Los bloques más pesados son más difíciles de mover de forma que es necesario un mayor número de balas y o una mayor puntería hacia las esquinas para conseguir que estos vuelquen.

El entorno de simulación no tiene ningún tipo de obstrucción en su uso así que una vez ejecutado el usuario puede estar dentro de este entorno todo el tiempo que quiera.

Como pruebas de habilidad al usuario se le proponen los siguientes juegos:

- **Basurero:** ¿Cuánto tiempo y cuantas balas gastas en agrupar todos los bloques en la misma esquina del jardín?
- **Lucky Luke:** ¿Cuánto tiempo tardas en volcar a todos los bloques por primera vez?
- **Torre de Babel:** ¿Tienes la habilidad suficiente para montar una torre con los bloques a base de disparos?





CAPÍTULO 10: RESULTADOS Y CONCLUSIONES





RESULTADOS

Se ha obtenido con la consecución de este proyecto un prototipo viable de un periférico funcional con aspecto realista para un entorno de simulación virtual que cumple el requisito de tener un bajo coste en comparación a los otros elementos del mercado.

- Sistema de retroalimentación háptica: se ha conseguido gracias a los motores y al muelle del arma un sistema de retroceso que consigue transmitir una sensación semejante a lo que es disparar un arma. Los motores por si solos resultan escasos para el usuario más allá de una ligera vibración, sin embargo, si provocan un desapuntado en el arma que es necesario reajustar por lo que se da por valido el resultado.
- Sistema de apuntado digital: Los sensores de ultrasonidos junto al giróscopo resultan un método bastante robusto en conjunto para definir el movimiento y posición del arma en el espacio. Se ha cumplido también el requisito propuesto de la movilidad del usuario por el entorno que era una de las novedades del proyecto. El usuario cuando se desplaza por la sala en la que está ejecutándose la simulación comprueba como su personaje se mueve con él dentro de la misma.
- Entorno virtual: Se ha desarrollado un entorno de juego virtual con una apariencia realista limitado tan solo por la potencia gráfica del computador que lo ejecute. Las características del entorno también se han delimitado de forma que si la simulación se corriese en una cueva virtual de 3 paredes como la dispuesta en la Universidad Rey Juan Carlos la sensación de inmersión sería totalmente realista al corresponderse las paredes con las propias del entorno. La simulación también cuenta con efectos de físicas simuladas con lo cual los objetos responden según los parámetros propios de las acciones físicas realizados contra ellos. También se ha incluido un sistema de partículas a los elementos naturales y se han utilizado algoritmos de distribución de elementos aleatorios como la hierba para garantizar un mayor efecto de inmersión.
- Bajo coste: Gracias a los materiales utilizados en el proyecto se ha conseguido mantener el coste del simulador en 113.45€ solo con el coste de los materiales. Esto se encuentra dentro de los estándares del Low Cost. Siendo su precio total de 100€ según el estudio económico realizado en el anexo II, lo cual lo sitúa en una muy buena posición dentro del mercado para usuarios particulares.



- Inalámbrico y autónomo: La utilización de tecnología *Bluetooth* ha servido para que el simulador no dependa de ningún tipo de cable o atadura a la hora de funcionar. Esto da total autonomía al arma pudiendo moverse el usuario por donde desee sin ningún impedimento. Se ha comprobado también que se dispone de una autonomía considerable y que el tiempo de uso continuado es el suficiente para sesiones intensivas.
- Aspecto: No se ha modificado la apariencia exterior de la réplica utilizada en el proyecto, lo que lo dota de un mayor realismo a la hora de utilizarlo. El peso en el mango y los botones colocados en posición estratégica contribuyen a que, durante su utilización, el usuario perciba el uso del arma como algo natural y propio de un arma de fuego real y no de una réplica.

CONCLUSIONES

Finalizado el proyecto se cuenta con una réplica de arma que tiene implementado un sistema de retroceso electrónico que transmite una sensación de realismo cuando dispara. Disparar se ve reflejado en un entorno de realidad virtual programado a través de un motor gráfico gratuito. El movimiento del arma es libre y totalmente inalámbrico con una autonomía probada. El arma ha pasado diversas pruebas de simulación las cuales ha pasado sin complicación alguna.

Las pruebas a las que se ha visto sometido el simulador han servido para comprobar que el grado de precisión de la misma es propio de un arma real, que no se trata de un elemento 100% preciso si no que se ve afectado por los diferentes elementos atmosféricos y ambientales, y eso se ve reflejado. El desapuntado producido por el retroceso se ha visto probado en las pruebas y se comprueba que el movimiento de la mira es el suficiente para necesitar la recolocación propia del apuntado que se pierde al disparar un arma.

Se ha verificado también la capacidad de disparo repetido y la cadencia que es capaz de mantener cuando se ve sometida a una serie repetida de recargas y si es capaz de mantener la precisión en esas circunstancias. Cosa que se ha visto superada sin problemas en las pruebas realizadas.

El coste del arma se conseguido mantener alrededor de los 100€ viéndose ligeramente superado, pero que en caso de iniciar un proceso de producción su precio de venta al público no se vería lo suficientemente incrementado como para considerarse un producto de ocio u formativo de coste elevado y que resulta accesible para cualquier público que demuestre interés en el mismo.



El grado de innovación incluido en el proyecto ha sido satisfactorio en cuanto a la implementación de tecnología sónica en el mismo se ha comprobado funcional y que en un entorno de simulación libre de obstáculos, que es una característica propia de este tipo de entornos, es totalmente funcional y no presenta errores serios.

La autonomía que se ha conseguido en el mismo permite operar el arma durante varios días a lo largo de sesiones de uso intensas y que no se ve afectada por el cambio del nivel de exigencias de la simulación. Por lo que se considera un logro considerable en un elemento de estas características el que sean necesarias muchas sesiones para tener que realizar un cambio de las baterías.

Por todo lo expuesto con los resultados más detallados en el apartado anterior se puede concluir que el proyecto ha sido exitoso con unos resultados favorables y que dejan un grado de margen suficiente para la ampliación y profundización del mismo en caso de ser pertinente.





CAPÍTULO 11: FUTUROS DESARROLLOS





La amplitud de este trabajo deja muchos campos abiertos para posibles nuevos desarrollos, una mayor profundización en cualquiera de los elementos aquí desarrollados o la inclusión de funciones no previstas en el proyecto inicial.

Referente al sistema de apuntado es necesario el desarrollo de un filtro de mayor precisión que el presentado en este proyecto con el fin de poder corregir los errores de medida que no se compensan al utilizar el retroceso provocado por el muelle en vez de los motores. También es requisito la optimización de la captación de valores validos en el proceso para disminuir los microdesplazamientos provocados por ligeras variaciones en el arma mezcladas con ruido eléctrico.

La utilización de un filtro de Kalman podría solucionar este tipo de problemas, pero para eso se requeriría de la implementación de los datos a través de programas externos como Matlab para utilizar la potencia de cálculo del ordenador en vez de la del microcontrolador. Sin embargo, este tipo de desarrollo resulta muy poco práctico para poder ser utilizado dentro del motor gráfico por lo que habría que pasar la información a través de otros programas como *Processing* para que el motor pudiese trabajar con ellos. Utilizar 3 programas en vez de uno incrementaría de forma considerable la potencia del microprocesador y gráfica necesaria por el ordenador en el que trabaje la simulación lo que se convierte en un aumento de coste indirecto para el producto.

Se encuentra necesario la creación de un módulo que consiga integrar los valores de medida del sensor de distancia con la posición del giróscopo, para que al inclinar el arma en una variación mayor a 45° con respecto a su posición horizontal base, no perciba la variación de distancia del debida al arco como un desplazamiento del usuario (fig. 81).



Figura 81: Imagen del problema de reconocimiento de distancia del ultrasonidos al variar la inclinación. Trayectoria en posición horizontal (Azul), Variación de distancia percibida (Negro).

Este problema podría verse solucionado escribiendo en el código que el personaje solo se pueda desplazar si el arma se encuentra en posición horizontal, pero eso le restaría realismo al simulador, por lo que en proyectos futuros se recomienda la integración de los datos de ambos elementos en un mismo modelo matemático.



El retroceso presenta el problema realismo/coste/espacio, un mayor realismo para el usuario implica un mayor desembolso en el proyecto buscando un tipo de motor o percutor que si tenga las dimensiones y los requisitos de potencia adecuados para encajar dentro del modelo del arma utilizado en el simulador. Por este motivo la creación de algún producto de bajo coste que sea capaz de proporcionar impactos percutidos de gran potencia es relevante para este proyecto.

La solución del problema del salto para personas demasiado bajas como para colocar el arma de forma cómoda a una altura de 2m viene dada por una pequeña calibración antes de usar el producto. Se ofrece a continuación una pequeña corrección de las líneas de código calibrando la altura del usuario hasta que este realiza el primer disparo. Sin embargo, se recomendaría otro método distinto para calibrar la altura que no limite ninguna de las características de movimiento. Se presentan solo las líneas del código que han sufrido alguna variación y aquellas que se han integrado nuevas.

```
long alturaUser, alturaSalto;
int primerdisparo;

void setup() {

primerdisparo=0;
alturaUser=170;

}

void loop() {

if(primerdisparo!=0) {
    alturaUser=(alturaUser+distanciaV)/2;
    alturaSalto=alturaUser+30;
}

if(distanciaV>=alturaSalto) {
    Salto=1;
} else {
    Salto=0;
}

}
```

Uno de los inconvenientes que presenta el arma viene dado por su utilización en habitaciones demasiado amplias en las que la distancia a las paredes paralelas a la línea de tiro es mayor a la distancia definida como altura de salto. En ocasiones estas condiciones de uso provocan que el personaje de la simulación comience a saltar en contra de la voluntad del usuario. Una posible solución conlleva la limitación del salto a la posición horizontal base del arma limitando la libertad de movimientos del usuario. La solución más adecuada implica colocar dos sensores de distancia en los laterales del arma para que según



esta rote cambie el sensor del que dependa el salto. Esto permitiría el salto en cualquiera de las posiciones del arma, pero implicaría un aumento del coste del simulador.

Una de las limitaciones del proyecto viene dada por el tiempo de retorno de la onda sonora desde los sensores de ultrasónidos. El desarrollo de un programa que permita la ejecución de ordenes mientras aún no se ha producido el regreso del pulso incrementaría la eficacia del simulador debido a un aumento de la velocidad de respuesta del mismo ante los sentidos del cuerpo humano. Una mayor fluidez del programa ante los cambios externos conllevan una mayor naturalidad al disminuir el número de escalones entre los reinicios del mismo que son causa del *Uncanny Valley*.

Si se realiza una producción en cadena del producto aquí prototipado sería necesaria la creación de una placa de circuito impreso que tenga las dimensiones del Arduino para que esta se pueda colocar como si se tratase de un módulo de ampliación de puertas digitales. Esta especificación es debida al limitado espacio dentro del cargador y la colocación de los cables es un proceso muy delicado que debe realizarse a mano para garantizar una correcta conexión. La creación de esta placa permitiría el soldado de cables a la misma de una manera más segura y menos vulnerable ante las tensiones vibratorias ante las que se ve sometida el arma.





CAPÍTULO 12: BIBLIOGRAFÍA





- [1] Definición de Quimera. Diccionario Real Academia Española.
<http://dle.rae.es/?id=UrsQsDY>
- [2] "The Matrix Trilogy" (Trilogía Matrix. Dir. Hermanos Wachowski). [DVD-BlueRay]. Warner Bros. 1999-2003
- [3] Paglia, J y Cosby, A. (2006). *Eureka*. Estados Unidos. SyFy.
- [4] KAWAHARA, R. (2009). *Sword Art Online*. Tokyo. ASCII Media Works.
- [5] .HACK// (2002) CyberConect2.
- [6] Head Mounted Display Definition. Dictionary of Engineering
<http://www.dictionaryofengineering.com/definition/head-mounted-display.html>
- [7] CAVE Automatic Virtual Environment. Visbox
<http://www.visbox.com/products/cave/>
- [8] Burleigh, T. J.; Schoenherr, J. R.; Lacroix, G. L. (2013). "Does the uncanny valley exist? An empirical test of the relationship between eeriness and the human likeness of digitally created faces". *Computers in Human Behavior* **29**: 3.
- [9] Virtual Gloves Definition. Dictionary.com
<http://www.dictionary.com/browse/data-glove>
- [10] 3DJuegos (2015) Playstation VR. Miedo Virtual.
<http://www.3djuegos.com/juegos/avances/18840/4539/0/playstation-vr/>
- [11] Robles De La Torre, G. (2006). The Importance of the sense of touch in virtual and real environments. *IEEE Multimedia*, 13 (3), 24-30.
- [12] Schiffman, Harvey (2001). «7». *La Percepción Sensorial*. Limusa Wiley. p. 152.
- [13] Anderson, Joseph and Barbara Fisher. "The Myth of Persistence of Vision." *Journal of the University Film Association* XXX:4 (Fall 1978): 3-8.
<http://web.archive.org/web/20090213030209/http://www.uca.edu/org/ccsmi/ccsmi/classicwork/Myth1.htm>
- [14] Aasias (2015). Simulador de Tiro Avanzado Virtra.
<http://aasias.info/simulador-de-tiro-avanzado-virtra/>
- [15] Kotaku (2010). Playstation Move: Everything you need to know.
<http://kotaku.com/5490632/playstation-move-everything-you-need-to-know>
- [16] Crysis 3 (2013). Crytek.
- [17] Morelle, R. (6 de mayo de 2013). Working Gun Made With 3D Printer. BBC.
<http://www.bbc.com/news/science-environment-22421185>
- [18] Normativa de la Federación Española de Airsoft (2016).
<http://federacion-espa-airsoft.es/wp-content/uploads/2016/05/NORMATIVA-FEA.pdf>
- [19] Program Executive Order Soldier (2015). Portfolio p. 199-200.
<http://www.peosoldier.army.mil/portfolio/#199>



- [20] Heckler-Koch HK G36 assault rifle (Germany) (sf). World Guns.
<http://world.guns.ru/assault/de/hk-g36-e.html>
- [21] The PDW/MP7 Newest Innovation (sf). HK Pro.
http://www.hkpro.com/index.php?option=com_content&view=article&id=88:mp7&catid=9:the-submachine-guns&Itemid=5
- [22] Hasting.M (2010). The most influential weapon of our time. The New York Review Of Books.
<http://www.nybooks.com/articles/2011/02/10/most-influential-weapon-our-time/>
- [23] Williams.H (2010). How did the Ak-47 become the most abundant weapon on Earth. Independent.
<http://www.independent.co.uk/news/world/how-did-the-ak-47-become-the-most-abundant-weapon-on-earth-2124407.html>
- [24] Mossberg 500 shotguns (USA) (sf). World Guns.
<http://world.guns.ru/shotgun/usa/mossberg-500-e.html>
- [25] The Sight M1911 (2011).
<http://www.sightm1911.com/>
- [26] D. K. Bhatnagar; "Position Trackersfor Head Mounted Display systems: a survey", March 29th, 1993.
- [27] Hackmann, Willem D. «Sonar Research and Naval Warfare 1914-1954: A Case Study of a Twentieth-Century Science.» *Historical Studies in the Physical and Biological Sciences* 16#1 (1986) 83-110
- [28] Diccionario de la Real Academia Española. Definición de Ecolocalización.
<http://dle.rae.es/?id=EKxeH6s>
- [29] Espadero. JM. (sf). QUIMERA: La cueva de realidad virtual de la URJC
<http://www.gmr.es/~jespa/quimera.html>
- [30] Álvarez. J. (2013). ¿CÓMO FUNCIONA? (II) La física de un puñetazo. CanalDeCiencias.
<http://www.canaldeciencias.com/2013/02/02/c%C3%B3mo-funciona-ii-la-f%C3%ADsica-de-un-pu%C3%B1etazo/>
- [31] Sistema de posicionamiento global.Wikipedia.
https://es.wikipedia.org/wiki/Sistema_de_posicionamiento_global
- [32]Starlino(2009). A Guide To using IMU (Accelerometer and Gyroscope Devices) in Embedded Applications. Starlino.
http://www.starlino.com/imu_guide.html
- [33] SODIAL(R) Modulo MPU-6050 de 3 ejes giroscopio + 3 Modulo Acelerometro GY-521 Eje para Arduino. Amazon.
https://www.amazon.es/gp/product/B00K67X810/ref=oh_aui_detailpage_o04_s00?ie=UTF8&psc=1



[34] Módulo ultrasónico hc-SR04 distancia transductor de medición del sensor para Arduino. Amazon.

https://www.amazon.es/gp/product/B00GGPYTSW/ref=oh_aui_detailpage_o03_s00?ie=UTF8&psc=1

[35] Fuentes.S.(2007). Novedades en la vibración del sixaxis Dualshock 3. Xataka.

<http://www.xataka.com/videojuegos/novedades-en-la-vibracion-del-sixaxis-dualshock-3>

[36] El actuador lineal. (sf). Linak.

<http://www.linak.es/about/?id3=4283>

[37] M. J. Escalera Tornero, A.J. Rodríguez Fernández; “Actuadores neumáticos” Ingeniería Industrial, Universidad de Huelva.

[38] What it is Arduino? (sf). Arduino.

<https://www.arduino.cc/en/Guide/Introduction>

[39] About (sf). Raspberry Pi.

<https://www.raspberrypi.org/about/>

[40] Vensengineer. (sf). Programming PIC Microcontrollers. Instructables.

<http://www.instructables.com/id/Programming-PIC-Microcontrollers/>

[41] Sensores Capacitivos (2016). SensorTecnics.

<http://www.sensortecnics.net/es/productos/category/96/sensores-y-transmisores/sensores-capacitivos>

[42] Fotorresistor. (sf). Wikipedia.

<https://es.wikipedia.org/wiki/Fotorresistor>

[43] What is Bluetooth Technology? (sf). Bluetooth.

<https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth>

[44] Discover Wi-Fi. (sf). Wi-Fi.org.

<http://www.wi-fi.org/discover-wi-fi>

[45] What is ZigBee? (sg). ZigBee.org.

<http://www.zigbee.org/what-is-zigbee/>

[46] Diymall Esp8266 Módulo de transceptor de WiFi Esp-01 para Arduino UNO R3. Amazon.

https://www.amazon.es/Diymall-Esp8266-oacute-transceptor-Arduino/dp/B01533Q3J0/ref=sr_1_1?ie=UTF8&qid=1463855477&sr=8-1&keywords=esp8266

[47] Arduino XBee Shield. Amazon.

https://www.amazon.es/ARDUINO-Arduino-XBee-Shield/dp/B00L7Z0BGU/ref=sr_1_4?ie=UTF8&qid=1463855524&sr=8-4&keywords=xbee

[48] Ecloud Shop® HC-05 Bluetooth Arduino host per modulo transceiver seriali. Amazon.



https://www.amazon.es/Ecloud-Bluetooth-Arduino-transceiver-seriali/dp/B015CJJPR2/ref=sr_1_1?ie=UTF8&qid=1463855099&sr=8-1&keywords=hc+05

[49] Ward.J.(2008). What is a game engine? GameCareerGuide.

http://www.gamecareerguide.com/features/529/what_is_a_game_.php

[50] What is Unreal Engine 4? (sf). Unreal Engine.

<https://www.unrealengine.com/what-is-unreal-engine-4>

[51] Gonzales.A.(2016). El desarrollo de Gears Of War 4 podría haberle costado 100 millones de dólares a Epic Games.Vandal.

<http://www.vandal.net/noticia/1350676348/el-desarrollo-de-gears-of-war-4-podria-haberle-costado-100-millones-de-dolares-a-epic-games/>

[52] List Of Unreal Engine Games. Wikipedia.

https://en.wikipedia.org/wiki/List_of_Unreal_Engine_games

[53] Unity (Software). Wikipedia.

[https://es.wikipedia.org/wiki/Unity_\(software\)](https://es.wikipedia.org/wiki/Unity_(software))

[54] Unity. Unity.

<https://unity3d.com/es/unity>

[55] List Of Unity Games. Wikipedia.

https://en.wikipedia.org/wiki/List_of_Unity_games

[56] Unity Showcase of Videogames. Unity.

<https://unity3d.com/es/showcase/gallery/games?platform=windows&genre=&gametype=t-all>

[57] Washenko.A (2015). 6 indie games that pushed boundaries at E3. Mashable.

<http://mashable.com/2015/06/25/innovative-indies-e3-2015/#PXAszXLe5sql>

[58] Angry Birds (2009). Rovio Entertainment.

[59] Library Software Serial. Arduino.

<https://www.arduino.cc/en/Reference/SoftwareSerial>

[60] Library Wire. Arduino.

<https://www.arduino.cc/en/Reference/Wire>

[61] WireEndTransmission. Arduino.

<https://www.arduino.cc/en/Reference/WireEndTransmission>

[62] Tr4duc7or. (2015). Tutorial de Arduino y MPU 6050. Robologs.

<http://robologs.net/2014/10/15/tutorial-de-arduino-y-mpu-6050/>

[63] Sloth.K (2012). A practical approach to Kalman filter and how to implement it. TKJ Electronics.

<http://blog.tkjelectronics.dk/2012/09/a-practical-approach-to-kalman-filter-and-how-to-implement-it/>



- [64] fighter5347. Foliage Starter Kit. Forums Unreal Engine.
<https://forums.unrealengine.com/showthread.php?59812-FREE-Foliage-Starter-Kit>
- [65] Blambot Comic Fonts. Firefight BB. 1001 Fonts.
<http://www.1001fonts.com/firefight-bb-font.html>
- [66] grizly32. UE4Duino - Arduino to UE4 plugin Release! Forums Unreal Engine.
<https://forums.unrealengine.com/showthread.php?68643-UE4Duino-Arduino-to-UE4-plugin-Release!>
- [67] El módulo Bluetooth HC_05. Prometec.
<http://www.prometec.net/bt-hc05/>
- [68] Configuración del módulo bluetooth HC-05 usando comandos AT.
NyalampMechatronics.
http://www.naylampmechatronics.com/blog/24_Configuraci%C3%B3n--del-m%C3%B3dulo-bluetooth-HC-05-usa.html
- [69] Zoomboomcrash. (2015). Las cifras que mueven los videojuegos hacen temblar a la industria del cine. La información
<http://blogs.lainformacion.com/zoomboomcrash/2015/01/12/las-cifras-que-mueven-los-videojuegos-hacen-temblar-a-la-industria-del-cine/>
- [70] Nuevos hábitos de consumo de ocio digital en movilidad. (sf). Fundetec.
<http://www.fundetec.es/reportajes/la-movilidad-cambia-los-habitos-de-consumo-del-ocio-digital/>
- [71] Facebook compra Oculus Rift por 2.000 millones de dólares. (sf). Europapress.
<http://www.europapress.es/portaltic/empresas/noticia-facebook-compra-oculus-rudy-2000-millones-dolares-20140325234721.html>
- [72] Chiconuclear. (2016). Las reservas de Oculus Rift vuelan a pesar de su elevado precio. AnaitGames.
<http://www.anaitgames.com/noticias/oculus-reservas-agotadas>
- [73] España-Gasto público de defensa. Expansión Datos Macro.
<http://www.datosmacro.com/estado/gasto/defensa/espana>
- [74] Presupuestos oficiales del estado español de defensa del año 2015.
<http://www.defensa.gob.es/Galerias/presupuestos/presupuesto-defensa-2015.pdf>
- [75] Barnes, Frank C. (1972). *Cartridges of the World*. Northfield II: DBI Books. p. 37. [ISBN 0-695-80326-3](#) (peso de balas)





ANEXO I:

CONFIGURACIÓN DEL BLUETOOTH





Los módulos de *Bluetooth* presentan una configuración estándar en la que vienen determinados por defecto las diversas características de comunicación del mismo. La realización del proyecto conlleva unas especificaciones concretas de velocidad de transmisión de datos y de paridad que es necesario verificar que se cumplen.

El acceso a la configuración de *Bluetooth* se realiza mediante el Arduino siguiendo una serie específica de pasos [67] [68].

1. Se ha de conectar el módulo de *Bluetooth* al Arduino tal y como se indica en la figura 82.

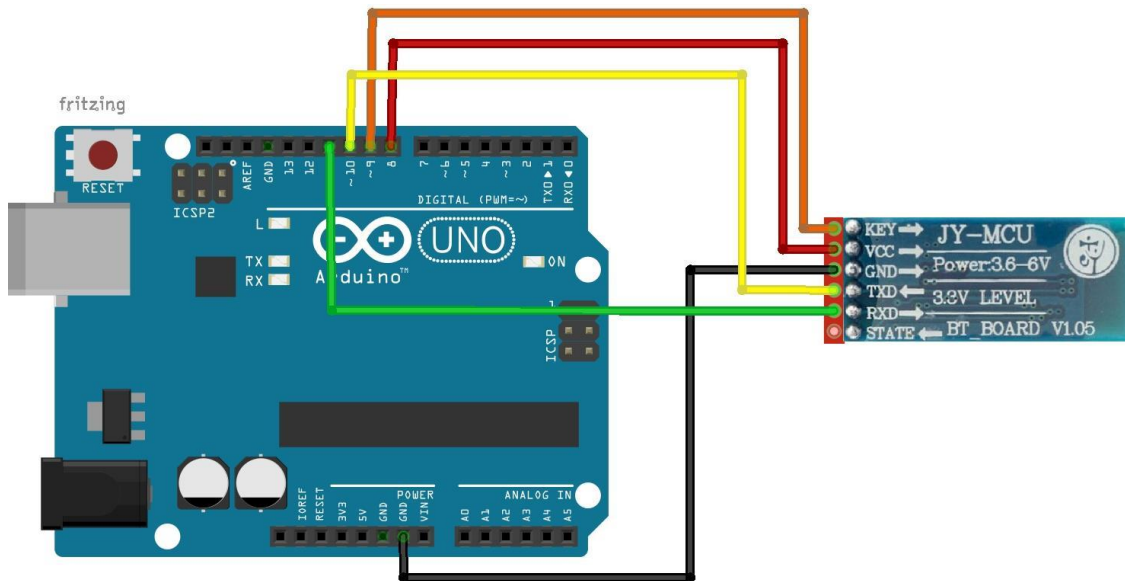


Figura 82: Conexión de Bluetooth para su configuración.

2. Se ha de cargar el siguiente código provisto en el Arduino.

```
#include <SoftwareSerial.h>

SoftwareSerial BT1(10, 11); // RX | TX

void setup()
{
  pinMode(8, OUTPUT);          // Se inicia primero el KEY
  pinMode(9, OUTPUT);          // La alimentación del modulo también
  vendrá de aqui
  digitalWrite(9, HIGH);
  delay(500);                  // Tiempo necesario para que se alimente
  antes el key que el VCC
  Serial.begin(38400);
  Serial.println("Levantando el modulo HC-06");
  digitalWrite(8, HIGH);      //Enciende el modulo
  Serial.println("Esperando comandos AT:");
  BT1.begin(38400);
}
```



```
void loop()
{
  if (BT1.available()) {
    Serial.write(BT1.read());
    Serial.println("El Bluetooth serial funciona");
  }
  if (Serial.available()) {
    BT1.write(Serial.read());
    Serial.println("El serial esta disponible");
  }
}
```

3. Cuando se vaya a cargar el código ha de pulsarse el botón que viene provisto en el módulo *Bluetooth* (fig. 83) en el caso de que lo haya mientras el programa se ejecuta. En caso de no haber botón se cargará el programa.

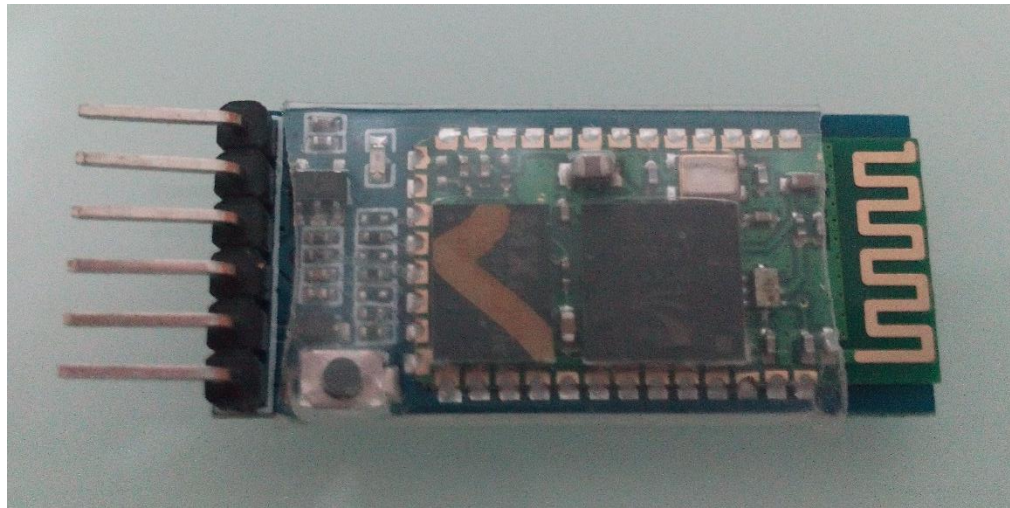


Figura 83: Parte trasera de un módulo Bluetooth con botón.

4. Después de cargarse el programa el *Bluetooth* entrará en modo programación pasados dos segundos. Se puede verificar que el *Bluetooth* esta en modo programación porque el LED que incluye en la placa parpadeará a razón de 0.5Hz. Una vez el *Bluetooth* esté en modo programación se podrá soltar el botón del módulo de *Bluetooth* en caso de que este lo tuviese.
5. A continuación, es necesario sincronizar el ordenador con el módulo de *Bluetooth*. Para ello cuando el ordenador trate de sincronizarse con él su identidad será HC-05 y se le pedirá una contraseña que por defecto es "1234".
6. Sincronizado el *Bluetooth*, se abre el monitor serial que es el medio por el que se realizará la configuración. Es necesario configurar el monitor a 38400



baudios con nueva línea y retorno de carro ya que son las características por defecto que presentan los módulos de *Bluetooth* recién adquiridos.

7. La comunicación con el módulo se realiza mediante comandos AT, lo primero que se debe hacer es comprobar que la conexión está bien realizada por lo que se le enviará al *Bluetooth* el comando "AT" a lo cual el módulo debería responder con un "OK". Si esta prueba de comunicación ha fallado repetir el proceso desde el paso 3.
8. Una vez realizada la comprobación se le pedirá al módulo la información concerniente a su identidad mediante los comandos "AT+NAME" y "AT+PSWD" a lo que la consola debería responder NAME=HC-05 y PSWD=1234.
9. Comprobada la información sobre su identidad se procederá a su reconfiguración. Se usan los comandos "AT+NAME=XXXX" y "AT+PSWD=XXXX" para introducir los nuevos valores a los cuales se responderá con sendos "OK". En el proyecto la configuración de estos elementos será "AT+NAME=ArmaBluetooth" y "AT+PSWD=1234".
10. Realizada la configuración de identidad del *Bluetooth* se pasará a comprobar los parámetros de comunicación. Para ello se usarán los comandos "AT+ROLE", "AT+UART". El *Bluetooth* debería devolver un "ROLE=0" indicando que está configurado como esclavo y "UART=38400,0,0" indicando que funciona a 38400 baudios y que no tiene ni bit de stop ni paridad.
11. En caso de que el rol por defecto no sea esclavo debe procederse a modificarlo mediante el comando "AT+ROLE=0". El baudrate es necesario modificarlo y para eso el *Bluetooth* tiene un código de asignación a todas las velocidades de baudrate disponibles. El código que se usará es "AT+UART=4,0,0" para asegurar que en el caso de que el bit de stop y la paridad no fuesen 0 ahora si lo sean.
12. En caso de haber cometido algún error en la programación es posible reiniciar el estado original del *Bluetooth* para volver a configurarlo desde cero mediante el comando "AT+RESET".





ANEXO II: ESTUDIO ECONÓMICO





En la actualidad el ocio digital está en boga llegando a ser una de las fuentes de ingresos que más beneficios ingresa, igualando a medios dominantes durante las últimas décadas del anterior milenio como es el cine [69].

Las novedades del mercado que más expectación están generando son aquellas que permiten añadir a las consolas clásicas un mayor grado de inmersión del videojuego en el usuario. Proyectos como *PlayStation VR* son fuentes de inversión en la actualidad que prometen una tasa de retorno de la inversión que comprometen a multinacionales con presupuestos millonarios [70]. Un ejemplo con cifras viene dado por el gigante *Facebook* que realizó la compra del proyecto de Kickstarter *Oculus Rift* por 2.000 millones de dólares [71].

Este tipo de desembolso no se produce solo por parte de las compañías que realizan el desarrollo del producto. El perfil de usuario medio de ocio digital es un varón con una edad entre los 25 y 44 años y una capacidad de desembolso económico equivalente a una renta media-alta. Este tipo de usuario goza de todos los productos multimedia que suelen salir al mercado, sino al menos de un alto porcentaje de los mismos. Es habitual que posea un *Smartphone* de gama alta sino uno de gama media acompañado de una *Tablet*, así como de un ordenador portátil, otro de sobremesa y una videoconsola conectada a un televisor de no más de 10 años.

Este usuario está acostumbrado a realizar un desembolso por sus productos de ocio que muy difícilmente baja de los 200€ y la aparición de este nuevo tipo de producto que son las gafas de realidad virtual sin duda va a ser foco de atención [72].

Las *Headsets* de realidad virtual son dispositivos de muy alto coste, las *HTC Vive* que son las únicas que se encuentran en el mercado a fecha de publicación de este proyecto tienen un coste de 900€ que, además, requieren de unas especificaciones técnicas por parte del ordenador que las vaya a utilizar propias de un producto de gama alta. Por lo que el usuario posiblemente se vea forzado a actualizar piezas de su computadora para disfrutar de la experiencia que se le propone.

La popularización en el mercado de un modo de juego que también se puede comercializar al sector profesional a muy bajo coste conseguiría atraer a usuarios de esta creciente corriente de adicción tecnológica.

Proponiendo un coste inferior al que ellos están acostumbrados a gastar se puede atraer a este tipo de target mediante una estrategia de “moderno y novedoso, pero atractivo para el bolsillo” de forma que añadan como gasto complementario a ese tipo de experiencia que pretenden obtener mediante las *Headsets*, el simulador de arma, puesto que la experiencia del simulador no se limita solo al uso en una pantalla.

Contemplando también su posible explotación en la distribución de este producto como un método de entrenamiento para fuerzas de defensa profesionales, el bajo coste al



que se pretende vender al público resulta atractivo para aquellas entidades gubernamentales o privadas, a las que se les puede ofrecer el producto en lotes con el fin de vender una mayor cantidad ajustándoles el precio con respecto al de venta al público.

Desde el periodo correspondiente a la anterior legislatura política el presupuesto de defensa se vio reducido en 200 millones de Euros desde el periodo de 2012 a 2014 empezando a presentar un repunte en el año 2016. El año 2015 tuvo un gasto de formación de personal armado de 324.747,98€, si la tendencia de los años posteriores es seguir subiendo, el ministerio de defensa del país es un comprador potencial de este tipo de productos ya que con su bajo coste rivaliza frente a otras ofertas de formación en simuladores profesionales cuyo coste económico es considerablemente mayor [73] [74].

Empresas como Virtra ofrecen su simulador profesional a un precio medio de 200.000\$ y las estadísticas dicen que en el 2014 el número de profesionales de la seguridad formados mediante la simulación es de 2500 en los últimos 4 años. Ofreciendo un precio competente con de venta al público, el simulador aquí ofrecido podría formar a 1000 agentes por el mismo precio que el que ofrece Virtra disponiendo cada uno de esos agentes de su propio simulador exclusivo a su entera disposición.

La estrategia de mercado a seguir con un producto como este es hallar el método de una producción en serie que mantenga el coste al mínimo con el fin de poder comercializarse en ambos ámbitos con un único producto, pero un precio de venta ajustado al pedido de lotes.



MEDICIONES

En esta sección se realizará un inventario de los materiales, herramientas y personal necesario para poder llevar a cabo el proyecto, así como las horas imputables a cada parte del proceso.

COMPONENTES PRINCIPALES

Lista de los materiales necesarios que conforman el producto final del proyecto (Tabla 10).

COMPONENTES	CANTIDAD
Arduino UNO	1
Sensor de ultrasonidos HC_SR04	2
Pulsador	2
Módulo MPU-6500 (Giróscopo)	1
Ppn13lb11c (Motor DC de alta velocidad)	2
Pila 9V	2
Placa para soldadura	1
Cables conexión Macho-Hembra (Pack 40Uds)	1
Resistencia 2.2K Ω	2
Resistencia 330 Ω	1
Resistencia 3.3K Ω	1
Transistor NPN 2N222a	1
Conmutador de 2 posiciones.	1
Réplica Airsoft HK G36C	1
Módulo Bluetooth HC-05	1
Hilo de estaño para soldadura	1

Tabla 10: Medida de los materiales del proyecto.

EQUIPO Y HERRAMIENTAS

Herramientas necesarias para poder fabricar el prototipo (Tabla 11).

ELEMENTO	CANTIDAD	HORAS DE USO
Ordenador	1	250
Multímetro	1	50
Soldador	1	40
Destornillador, alicates, pelacables y herramientas varias	1	80
Protoboard	1	50
Fuente de Alimentación regulable	1	20
Cable de conexión USB-Arduino	1	40

Tabla 11: Medida del equipo y de las herramientas.



SOFTWARE

Software y programas de ordenador requeridos para la consecución del proyecto (Tabla 12).

PROGRAMA	HORAS DE USO
Arduino IDE	25
Unreal Engine 4	50
Microsoft Office 365	80
Adobe Photoshop CS5	2
Adobe Premiere Pro CC 2015	5
Fritzing	2

Tabla 12: Medida de Software

MANO DE OBRA

Actividades en las que se separa el proyecto y número de horas consumidas por cada actividad. Los procesos de documentación y estudios previos se realizarán por un Ingeniero de sistema que actúa como líder del proyecto. Los procesos que implican diseño de alguno de los elementos del proyecto, así como las pruebas del mismo se realizan por un ingeniero electromecánico y aquellos que requieran de conocimientos de programación por un ingeniero electrónico. El montaje del simulador y las partes necesarias de taller se le asignan a un graduado superior en electrónica (Tabla 13).

ACTIVIDAD	HORAS
Estudio y diseños previos	40
Diseño de Circuitos	50
Montaje	70
Programación	85
Ensamblado del simulador	20
Documentación	100

Tabla 13: Medida de la mano de obra.



PRECIOS UNITARIOS

Coste de cada uno de los elementos del proyecto por sí mismo.

COMPONENTES PRINCIPALES

COMPONENTES	COSTE UNIDAD(€)
Arduino UNO	23,68
Sensor de ultrasonidos HC_SR04	5,29
Pulsador	1,71
Módulo MPU-6500 (Giróscopo)	2,75
Ppn13lb11c (Motor DC de alta velocidad)	2,66
Pila 9V	4,78
Placa para soldadura	2,07
Cables conexión Macho-Hembra (Pack 40Uds)	3,99
Resistencia 2.2K Ω	0,04
Resistencia 330 Ω	0,04
Resistencia 3.3K Ω	0,04
Transistor NPN 2N222a	1,65
Conmutador de 2 posiciones.	2,7
Réplica Airsoft HK G36C	35,9
Módulo Bluetooth HC-05	3,9
Hilo de estaño para soldadura	7,77

Tabla 14: Precios unitarios de los componentes.

EQUIPO Y HERRAMIENTAS

Todo el equipo y herramientas necesarias han sido provistas por la universidad por lo que de cara al proyecto su coste ha sido de 0€. Los usos de estos elementos de cara a un proyecto comercial se valorarán como parte del proceso de manufacturación del material que se estima en 12€/hora.

SOFTWARE

Los softwares utilizados en el proyecto cuentan todos con licencia gratuita, provista por un acuerdo de la universidad con la empresa o condicionada al beneficio comercial. Salvo por los productos Adobe que tienen un precio de uso de 24.19€/mes.



MANO DE OBRA

ACTIVIDAD	PRECIO (€/hora)
Estudio y diseños previos	25
Diseño de Circuitos	20
Montaje	15
Programación	20
Ensamblado del simulador	15
Documentación	25

Tabla 15: Precio unitario de cada hora de trabajo en cada una de las actividades.



SUMAS PARCIALES

Coste de total de cada una de las secciones, actividades y componentes del proyecto.

COMPONENTES PRINCIPALES

COMPONENTES	CANTIDAD	COSTE UNIDAD (€)	COSTE TOTAL PRODUCTO (€)
Arduino UNO	1	23,68	23,68
Sensor de ultrasonidos HC_SR04	2	5,29	10,58
Pulsador	2	1,71	3,42
Módulo MPU-6500 (Giróscopo)	1	2,75	2,75
Ppn13lb11c (Motor DC de alta velocidad)	2	2,66	5,32
Pila 9V	2	4,78	9,56
Placa para soldadura	1	2,07	2,07
Cables conexión Macho-Hembra (Pack 40Uds)	1	3,99	3,99
Resistencia 2.2KΩ	2	0,04	0,08
Resistencia 330Ω	1	0,04	0,04
Resistencia 3.3KΩ	1	0,04	0,04
Transistor NPN 2N222a	1	1,65	1,65
Conmutador de 2 posiciones.	1	2,7	2,7
Réplica Airsoft HK G36C	1	35,9	35,9
Módulo Bluetooth HC-05	1	3,9	3,9
Hilo de estaño para soldadura	1	7,77	7,77
TOTAL			113,45

Tabla 16: Suma parcial del coste de los componentes.

EQUIPO Y HERRAMIENTAS

Se considerará el mayor tiempo de uso de uno de los elementos del taller como número total de horas de uso del mismo asumiendo que el resto de los elementos se usan de forma simultanea por los demás miembros del proyecto cuando es necesario. Siendo entonces el gasto total correspondiente a el coste unitario definido en el apartado anterior por las 80 horas de uso de herramientas varias dando un coste total de 720€.

SOFTWARE

Puesto que el estudio económico solo abarca hasta la compensación del coste del mismo no es necesario preocuparse por la tasa del 5% sobre el beneficio de *Unreal Engine*. El único coste asociado al software será el del alquiler durante un mes de los servicios de Adobe para el proceso de documentación que es de 24,19€.



MANO DE OBRA

ACTIVIDAD	HORAS	PRECIO (€/hora)	COSTE DE LA ACTIVIDAD (€)
Estudio y diseños previos	30	25	750
Diseño de Circuitos	40	20	800
Montaje	60	15	900
Programación	75	20	1500
Ensamblado del simulador	10	15	150
Documentación	90	25	2250
TOTAL			6350

Tabla 17: Suma parcial de los costes asociados a la mano de obra.



PRESUPUESTO GENERAL

Coste total que tiene el desarrollo del prototipo de simulador desarrollado durante el proyecto.

CONCEPTO	COSTE (€)
Componentes	113,45
Equipo y herramientas	720
Software	24,19
Mano de Obra	6350
TOTAL	7207,64

Tabla 18: Coste total asociado al proyecto.



RENTABILIDAD DEL PROYECTO

El coste obtenido en la sección anterior es el que viene asociado al desarrollo de un único prototipo. La producción en serie de un producto de estas características ve reducido su coste por desarrollo al distribuirse todo el coste entre las unidades fabricadas y posteriormente distribuidas.

Asumiendo que no existen más gastos asociados al producto que los expuestos arriba (se descartan gastos de licencias de propiedad intelectual, distribución, publicidad, etc.) se calculará el número de unidades que es necesario vender con el fin de retornar el dinero que conlleva el desarrollo de este prototipo.

Se definen como gastos fijos aquellos que serán invariables independientemente del número de productos que se vayan a fabricar. En estos gastos se encuentra el desarrollo del prototipo del arma del que se parte para hacer los demás modelos, el alquiler de la maquinaria y los contratos del personal que se va a ocupar de fabricar el producto. Se establecerá el gasto fijo en 10.000€, pues una vez terminado el prototipo solo se necesita la nómina del empleado con la formación suficiente para montarla y el alquiler del taller para poder fabricarlas.

Los gastos variables son los que dependen del número de unidades que se vayan a fabricar. Son los gastos asociados al coste de los materiales del arma. Se establecerá el gasto variable en 80€. Esto es porque a pesar de que el coste de los materiales es de 113.45€ al realizarse una compra al por mayor de una gran cantidad de productos para su manufacturación su coste se aproxima al precio de fábrica.

$$C_t = C_f + C_v * n$$

Ecuación 3: Ecuación del coste total de un producto.

Se obtiene el coste total del producto como la suma de costes fijos con los variables por el número de unidades manufacturadas como viene mostrado en la ecuación 3.

$$B = P_u * n - C_t$$

Ecuación 4: Ecuación del beneficio.

El beneficio se obtiene según indica la ecuación de la figura 4 donde B es el beneficio obtenido, P_u el precio de venta de cada unidad, n el número de unidades vendidas y C_t los costes totales.

Se establece un precio de venta al público de 150€ con el fin de marcar un precio atractivo a los compradores potenciales de este tipo de tecnologías que no les prive de otras adquisiciones semejantes y se pueda percibir como un gasto complementario.



$$B = P_u * n - C_f - C_v * n$$

Ecuación 5: Inclusión de la ecuación 3 en la ecuación 4.

Juntando las dos ecuaciones e igualando el beneficio a 0 se puede calcular el número de unidades necesarias para retornar la inversión del desarrollo de este producto. El número de unidades necesarias al público general es 143 unidades (Ecuación 5).

En caso de querer implementar la venta a organizaciones privadas y organismos gubernamentales se establecerá un precio de venta a estos compradores de 120€ por unidad sabiendo que estos comprarán en lotes de 10 armas y constituyen un 60% del mercado al que se pretende alcanzar.

$$B = P_{upublico} * 0.4 * n + P_{uprivado} * 0.6 * n - C_f - C_v * n$$

Ecuación 6: Ecuación del beneficio incluyendo la venta a particulares y organizaciones.

Con la venta a particulares y organizaciones el número mínimo de armas que se debe vender con el fin de recuperar la inversión realizada en el desarrollo del simulador ha de ser 193 unidades (Ecuación 6).





ANEXO III:

CALCULO TEÓRICO DE LA
VALIDEZ DEL USO DEL
GIRÓSCOPO Y LOS
MOTORES DC





Giróscopo

Considerando que un brazo de un deportista experimentado puede llegar a los 10m/s en su extremo [30] y el arma con el que nosotros trabajaremos tiene una longitud de 720mm [20] vamos a usar unos cálculos que nos permitan demostrar que el grado de precisión aportado es el suficiente:

Cuando una persona apunta de un extremo a otro describe prácticamente un cuarto de circunferencia, pues hay que restarle los grados correspondientes a apoyar el arma en el pecho. Sabiendo esto podemos deducir el ángulo que va a girar (Ecuación 7):

$$\theta = \frac{80}{360} * 2\pi = 1,396 \text{ rads}$$

Ecuación 7: Conversión de los ángulos de giro del arma a radianes.

Suponiendo que el arma en su extremo se moverá a la misma velocidad que la mano de un deportista poniéndonos en el caso más desfavorable obtendremos su velocidad angular (Ecuación 8).

$$\omega = 0,72 * 10 = 7,2 \frac{\text{rads}}{\text{s}}$$

Ecuación 8: Obtención de la velocidad angular de arma.

Con estos dos números podemos obtener el tiempo que tardara en realizar dicho cuarto de arco de trayectoria (Ecuación 9).

$$t = \frac{\theta}{\omega} = 0,1939 \text{ s}$$

Ecuación 9: Obtención del tiempo que tarda el brazo en realizar todo el recorrido.

Comprobamos que si multiplicamos ese tiempo por la menor de las tolerancias de nuestro producto seguimos obteniendo un ángulo tal que:

$$T * 250 = 48,48^\circ$$

Ecuación 10: Obtención de la cantidad de ángulos que se miden en la fracción de tiempo que se tarda en realizar todo el recorrido del brazo.

Por lo que seríamos capaz de obtener la medida de 48 grados a lo largo de esa misma fracción de tiempo solo con la menor de tolerancia que es solo la mitad de los puntos por los que hemos pasado así que podemos afirmar que este nivel de precisión es más que aceptable.



Motores DC.

Al disparar una bala se pretende enviar un pequeño proyectil en línea recta a lo largo de una distancia de 800m [20] en el mejor de los casos. Las balas de un subfusil son de poco calibre, en este caso seleccionaremos las propias de 6mm que tienen un peso que ronda los 4 gramos [75].

Consideraremos la explosión como una distribución uniforme de fuerzas con forma circular por lo que solo existirá la fuerza que empuja la bala hacia fuera y la reacción equivalente que empuja el arma hacia atrás.

Estas balas alcanzan una velocidad tope de 940m/s por lo que calcularemos la aceleración necesaria para conseguir que se alcance tal velocidad sabiendo que el punto en el que se alcanza es a la salida del cañón. Ya que es conocido el modelo de arma que vamos a utilizar se usará la longitud de su cañón que es 228mm [20].

Gracias a la forma de punta de lanza que tienen las balas se supondrá nula la fuerza que ejerce el aire para frenarla y ya que se está planteando este caso para la distancia máxima tampoco existirá un viento oponible que la frene.

$$940 \frac{m}{s} = a * t$$

Ecuación 11: Formula de la velocidad en función de la aceleración.

$$0.228m = a * t^2 * \frac{1}{2}$$

Ecuación 12: Formula de la distancia en función de la aceleración y velocidad inicial.

Del sistema que forman las ecuaciones 11 y 12 se obtiene que el tiempo que tarda en salir la bala del cañón del arma es de 48 milésimas de segundo lo que implica una aceleración de $1.95 \times 10^6 m/s^2$.

$$F = 0.004g * 1.95 \times 10^6 \frac{m}{s^2}$$

Ecuación 13: Formula de la fuerza obtenida como masa por aceleración.

Por lo tanto, la fuerza que se ha ejercido sobre la bala es de 7833N según la ecuación 13. Esta misma fuerza se verá repartida de manera uniforme contra el resto del arma. Si la superficie de la misma es de aproximadamente $756cm^2$ y la media de la superficie de la palma de una mano humana $64cm^2$ la fuerza que percibe la palma es de 560N. Si un motor puede llegar a dar una fuerza de 16kg/cm con una superficie de $24cm^2$ la fuerza es prácticamente equivalente.



ANEXO IV:

CÓDIGO IMPLEMENTADO

EN EL ARMA





CODIGO:

```
#include <SoftwareSerial.h>
#include <Wire.h>

//CONFIGURACION DE PINES DEL ARDUINO
SoftwareSerial blue(2, 3); //Asignacion de los pines reservados para la
comunicacion inalambrica Rx|Tx

//Direccion I2C de la IMU
#define MPU 0x68

//Ratios de conversion
#define A_R 16384.0
#define G_R 131.0

//Conversion de radianes a grados 180/PI
#define RAD_A_DEG = 57.295779

//Variables necesarias para la medida del tiempo en sensor vertical y
sensor frontal de ultrasonidos
long tiempoV, tiempoF;
long distanciaV, distanciaF;
long VariFront, Fprev;
int Salto;

//Valores sin refinar
int16_t AcX, AcY, AcZ, GyX, GyY, GyZ;

//Angulos
float AccY, AccX, AccZ; //Angulos del acelerometro
float GyyX, GyyY, GyyZ; //Angulos del giroscopio

//Estimacion del angulo mediante filtro complementario
double Xpost, Ypost, Zpost; //Angulos en el modelo a posteriori
double Xprev, Yprev, Zprev; //Angulos en el modelo previo
double VariX, VariY, VariZ; //Variacion de angulos que nos da los
diferenciales de cambio de posicion

//Disparador y retroceso
int dispararant;
int dispararact;
int disparar;
int lectura;
int retroceso;

//Cargador y balas
#define balasmax 8
int balasact;
int recarga;

void setup() {

    //CONFIGURACION DE LOS PINES DEL ARDUINO
    pinMode(9, OUTPUT); //Pin definido como salida para la emision del
pulso ultrasonico vertical
```



```
pinMode(8, INPUT); //Pin definido como entrada para la recepcion del
rebote del pulso ultrasonico vertical
pinMode(11, OUTPUT); //Pin definido como salida para la emision del
pulso ultrasonico frontal
pinMode(10, INPUT); //Pin definido como entrada para la recepcion del
rebote del pulso ultrasonico frontal
pinMode(5, INPUT); //Pin definido como entrada para saber cuando se
produce una recarga del arma
pinMode(4, INPUT); //Pin definido como entrada para saber cuando se
esta disparando
pinMode(6, OUTPUT); //Pin definido como salida para indicar cuando se
esta produciendo retroceso debido al disparo
pinMode(7, OUTPUT); //Pin definido como salida para indicar cuando se
esta produciendo retroceso debido al disparo

//CONFIGURACION DE LA TRANSMISION EN SERIE DEL GIROSCOPO
Wire.begin();
Wire.beginTransmission(MPU);
Wire.write(0x6B);
Wire.write(0);
Wire.endTransmission(true);

//INICIALIZACION DE LOS BAUDRATES DEL PROGRAMA
Serial.begin(9600);
blue.begin(9600);
blue.println("Conectado");
}
void loop() {

    //Actualizacion del valor previo del modelo
    Xprev=Xpost;
    Yprev=Ypost;
    Zprev=Zpost;

    //OBTENCION DE LOS ANGULOS DE POSICION DEL ARMA
    //Leer los valores del Acelerometro de la IMU
    Wire.beginTransmission(MPU);
    Wire.write(0x3B); //Pedir el registro 0x3B - corresponde al AcX
    Wire.endTransmission(false);
    Wire.requestFrom(MPU,6,true); //A partir del 0x3B, se piden 6
registros
    AcX=Wire.read()<<8|Wire.read(); //Cada valor ocupa 2 registros
    AcY=Wire.read()<<8|Wire.read();
    AcZ=Wire.read()<<8|Wire.read();

    //Leer los valores del Giroscopio
    Wire.beginTransmission(MPU);
    Wire.write(0x43); //Se salta el registro 0x41 porque ese nos ofrece el
valor de la temperatura
    Wire.endTransmission(false);
    Wire.requestFrom(MPU,6,true);
    GyX=Wire.read()<<8|Wire.read();
    GyY=Wire.read()<<8|Wire.read();
    GyZ=Wire.read()<<8|Wire.read();

    //Obtencion de los valores de angulos provistos por el acelerometro
```



```
AccX = atan((AcX/A_R)/sqrt(pow((AcY/A_R),2) +  
pow((AcZ/A_R),2))) *RAD_TO_DEG;  
AccY = atan((AcY/A_R)/sqrt(pow((AcX/A_R),2) +  
pow((AcZ/A_R),2))) *RAD_TO_DEG;  
AccZ = atan((AcZ/A_R)/sqrt(pow((AcX/A_R),2) +  
pow((AcY/A_R),2))) *RAD_TO_DEG;  
  
//Obtencion de los valores de velocidad angular provistos por el  
Giroscopio  
GyyX = GyX/G_R;  
GyyY = GyY/G_R;  
GyyZ = GyZ/G_R;  
  
//Filtro Complementario  
Xpost=0.6*(Xpost+GyyX*0.01)+0.4*AccX;  
Ypost=0.6*(Ypost+GyyY*0.01)+0.4*AccY;  
Zpost=0.6*(Zpost+GyyZ*0.01)+0.4*AccZ;  
  
//Comprobacion de cuanto se ha movido el arma con respecto a la  
anterior ejecucion  
VariX=-(Xpost-Xprev);  
if (VariX<=0.05 && VariX>=-0.05){  
    VariX=0;  
}  
VariY=Ypost-Yprev;  
VariZ=Zpost-Zprev;  
  
//OBTENCION DE LA DISTANCIA CON EL SENSOR VERTICAL  
digitalWrite(9,LOW); //Estabilizamos el sensor antes de comenzar la  
medida  
delayMicroseconds(5);  
digitalWrite(9, HIGH); //Envio del pulso ultrasonico  
delayMicroseconds(10);  
tiempoV=pulseIn(8, HIGH); //Funcion dedicada que cuenta el tiempo  
desde que se envia el pulso de ultrasonidos y el sensor recibe la onda  
rebotada  
distanciaV= int(0.017*tiempoV); //Ecuacion que nos da la distancia  
entre el objeto con el que haya rebotado el pulso y el sensor  
  
if (distanciaV>=200){  
    Salto=1;  
}else{  
    Salto=0;  
}  
  
//OBTENCION DE LA DISTANCIA CON EL SENSOR FRONTAL  
digitalWrite(11,LOW); //Estabilizamos el sensor antes de comenzar la  
medida  
delayMicroseconds(5);  
digitalWrite(11, HIGH); //Envio del pulso ultrasonico  
delayMicroseconds(10);  
tiempoF=pulseIn(10, HIGH); //Funcion dedicada que cuenta el tiempo  
desde que se envia el pulso de ultrasonidos y el sensor recibe la onda  
rebotada  
distanciaF= int(0.017*tiempoF); //Ecuacion que nos da la distancia  
entre el objeto con el que haya rebotado el pulso y el sensor  
VariFront=distanciaF-Fprev;
```



```
Fprev=distanciaF;

//Eliminacion de las variaciones de tolerancia minima y outliers que no
se podrían dar
if(VariFront>= -1 && VariFront<=1){
    VariFront=0;
}else if(VariFront>=20){
    VariFront=0;
}else if(VariFront<=-20){
    VariFront=0;
}
//Lectura del disparo
dispararant=dispararact;
lectura=digitalRead(4);

if(lectura==1){
    dispararact=1;
}else{
    dispararact=0;
}

if(dispararact==1 && dispararact!=dispararant && balasact!=0){
    disparar=1;
}else{
    disparar=0;
}
//Control del cargador y las balas
if(disparar==1){
    balasact=balasact-1;
    retroceso=3;
}

recarga = digitalRead(5);
if(recarga==1){
    balasact=balasmax;
}

//Implementacion del retroceso
if(retroceso > 0){
    digitalWrite(6,HIGH);
    digitalWrite(7,HIGH);
}else if(retroceso <= 0){
    digitalWrite(6,LOW);
    digitalWrite(7,LOW);
}
retroceso=retroceso-1;

//PRUEBA DE LAS MEDIDAS EN PANTALLA
blue.print(VariX);
blue.print(",");
blue.print(VariY);
blue.print(",");
blue.print(Zpost);
blue.print(",");
blue.print(VariFront);
blue.print(",");
blue.print(Salto);
```



```
blue.print(",");  
blue.print(disparar);  
blue.print(",");  
blue.print(balasact);  
blue.print("\n");  
}
```