



Universidad
Carlos III de Madrid

TRABAJO FIN DE GRADO

Simulación de trayectorias de barcos y aplicación al control marítimo

Autor: Jaime Vázquez Coll (100284760)

Tutor: Antonio Berlanga de Jesús

Grado: Doble grado en Informática y ADE

Madrid, junio de 2016

Resumen

En el presente documento se lleva a cabo el estudio pertinente para la realización de un software que permite visualizar la trayectoria y ubicación de las naves y así, aumentar la seguridad en el mar y ayudar a los controladores navales a desempeñar su función.

En primer lugar, para la realización del simulador hemos establecido los objetivos mediante un análisis de la situación actual del mundo marítimo, que nos ha permitido identificar sus deficiencias, carencias y necesidades. También se ha realizado la búsqueda de las soluciones llevadas a cabo por otros organismos, para definir mejor las características de nuestro programa.

En segundo lugar, elegimos la plataforma más adecuada para la implementación. A partir de las capacidades de este software y de aquellas soluciones que hemos observado en otros proyectos similares, definiremos las características que nuestro programa debe cumplir.

Conforme a estas características implementaremos el software para que sea lo más fiel posible al objetivo que deseamos conseguir, que no es otro, que el de facilitar la navegación en el mar.

Para finalizar, detallamos las posibles mejoras aplicables a este software en un futuro, y usarlo como base. También se podría iniciar un nuevo proyecto para cada una

de estas funciones si se prevé que con la plataforma seleccionada puede resultar complicado.

Palabras clave: Simulador, motor gráfico, software, controlador naval y puerto.

Índice general

1. INTRODUCCIÓN, ESTRUCTURA Y OBJETIVOS.....	9
1.1 Introducción	9
1.1.1 Motivación.....	9
1.1.2 Ambientación.....	12
1.2 Estructura del documento	14
1.3 Objetivos.....	16
2. ESTADO DEL ARTE	17
2.1 Motores gráficos.....	17
2.1.1 Unreal Engine.....	18
2.1.2 Cry Engine.....	19
2.1.3 Panda3D	20
2.1.4 Ogre3D.....	20
2.1.5 Unity.....	21
2.1.6 Conclusiones.....	22
2.2 Simuladores navales	23
3. DESARROLLO.....	28
3.1 Herramientas utilizadas	28
3.1.1 Medios hardware	29
3.1.2 Medios software.....	29
3.2 Estructura del proyecto	31
3.3 Estudio de viabilidad del sistema	33
3.3.1 Establecimiento del alcance del sistema.....	34
3.3.2 Definición de requisitos del sistema	36
3.4 Diagrama de casos de uso.....	53
3.5 Casos de uso.....	54
3.6 Matriz de trazabilidad.....	58
4. DIFICULTADES ENCONTRADAS Y RESULTADOS	60
4.1 Dificultades encontradas.....	60
4.2 Resultados.....	61
5. CONCLUSIONES Y POSIBLES TRABAJOS FUTUROS.....	63

5.1 Conclusiones	63
5.2 Trabajos futuros	64
6. PLANIFICACIÓN Y PRESUPUESTO	66
6.1 Planificación	67
6.2 Presupuesto	68
6.2.1 Entorno de trabajo.....	68
6.2.2 Coste de licencias	69
6.2.3 Mano de obra	69
6.2.4 Coste del hardware	70
6.2.5 Coste del total.....	71
6.2.6 Coste con amortizaciones.....	71
7. REFERENCIAS	73
8. GLOSARIO.....	75
9. ANEXOS	76
10. ANEXO I: ENGLISH TRANSLATION	76
8.1 Introduction.....	79
8.1.1 Motivation	79
8.1.2 Setting.....	81
8.2 Objectives	83
8.3 Results	84
8.4 Conclusions and future works.....	85
8.4.1 Conclusions	85
8.4.2 Future Works.....	86
11. ANEXO II: SOFTWARE IMPLEMENTADO Y MANEJO	87

Índice de ilustraciones y *figures*

Ilustración 1: VTS.....	14
Ilustración 2: captura del videojuego Harpoon 3	24
Ilustración 3: captura del videojuego Silent Hunter.....	25
Ilustración 4: Hardware simulando puente de mando	26
Ilustración 5: captura del European Ship Simulator	27
Ilustración 6: Ciclo de vida del proyecto.....	32
Ilustración 7: Diagrama de casos de uso	53
Ilustración 8: Diagrama de tiempos previstos.....	67
Ilustración 9: Captura de pantalla inicial del software	87
Ilustración 10: captura de pantalla de barco manual.....	88
Ilustración 11: captura de pantalla de los barcos con piloto automático.....	88
Ilustración 12: Captura de pantalla de ruta de embarcación manual	89
Ilustración 13: Captura de pantalla del cambio de la cámara	90
Ilustración 14: captura de pantalla de fichero de barcos y rutas	90
Ilustración 15: Captura de pantalla de fichero de modificación de cámara.....	91
Ilustración 16: Inserción de nuevas embarcaciones.....	92
Figure 1:AIS and VTS	83

Índice de tablas y *tables*

Tabla 1: Flota mercante mundial	11
Tabla 2: Resumen de motores gráficos	22
Tabla 3: Especificaciones portátil	29
Tabla 4: Modelo tabla.....	37
Tabla 5: RF-F-1.0.....	38
Tabla 6: RF-F-1.1.....	39
Tabla 7: RF-F-1.2.....	39
Tabla 8: RF-F-1.3.....	40
Tabla 9: RF-F-1.4.....	40
Tabla 10: RF-F-2.0.....	41
Tabla 11: RF-F-3.0.....	41
Tabla 12: RF-F-3.1.....	41
Tabla 13: RF-F-4.0.....	42
Tabla 14: RF-F-4.1.....	42
Tabla 15: RF-F-5.0.....	42
Tabla 16: RF-F-6.0.....	43
Tabla 17: RF-F-7.0.....	43
Tabla 18: RF-F-7.1.....	43
Tabla 19: RF-F-7.2.....	44
Tabla 20: RF-F-7.3.....	44
Tabla 21: RNF-R-1.0.....	45
Tabla 22: RNF-R-1.1.....	45
Tabla 23: RNF-I-1.0.....	46
Tabla 24: RNF-I-1.1.....	46
Tabla 25: RNF-I-2.0.....	47
Tabla 26: RNF-I-2.1.....	47
Tabla 27: RNF-I-3.0.....	47
Tabla 28: RNF-I-3.1.....	48
Tabla 29: RNF-I-3.2.....	48
Tabla 30: RNF-I-3.3.....	48
Tabla 31: RNF-I-3.4.....	49
Tabla 32: RNF-I-4.0.....	49
Tabla 33: RNF-I-4.1.....	50
Tabla 34: RNF-I-4.2.....	50

Tabla 35: RNF-I-5.0	50
Tabla 36: RNF-I-6.0	51
Tabla 37: RNF-S-1.0	51
Tabla 38: RNF-P-1.0	52
Tabla 39: RNF-P-2.0	52
Tabla 40: RNF-P-3.0	52
Tabla 41: modelo tabla CU	54
Tabla 42: CU-1	55
Tabla 43: CU-2	56
Tabla 44: CU-3	57
Tabla 45: CU-4	57
Tabla 46: CU-5	58
Tabla 47: Matriz de trazabilidad	59
Tabla 48: precio licencias	69
Tabla 49: Coste de la mano de obra	70
Tabla 50: Coste hardware	71
Tabla 51: Coste Total	71
Tabla 52: amortizaciones	72
Tabla 53: coste total amortizado	72
Table 1: world merchant fleet	80

Capítulo 1

Introducción, estructura y objetivos

1.1 Introducción

En este punto explicaremos cual ha sido la motivación que hemos encontrado para realizar este proyecto. Además, introduciremos cuál es el tema que se va a tratar en este proyecto, se describirán tanto las necesidades navales actuales como en el control marítimo en los puertos. Así mismo, se detallarán las características que se esperan del software que se implementará.

1.1.1 Motivación

La idea principal es construir un sistema de simulación de trayectorias navales completo y flexible, que permita diversas aplicaciones. Las principales funciones para las que está pensado este software son:

1. Servir de base para probar los siguientes algoritmos:

1.1. Algoritmos de colisiones entre barcos

1.2. Algoritmos de diversas situaciones atmosféricas

1.3. Algoritmos de movimiento de los barcos

2. Probar escenarios tanto hipotéticos como realistas y observar las consecuencias

Esta función podría resultar útil para realizar las pruebas de los diferentes títulos navales y comprobar si efectivamente los candidatos tienen las aptitudes necesarias para dirigir las embarcaciones para las que se les examina.

Esta misma utilidad, se podría extender al caso de los prácticos del puerto, cuyas pruebas actualmente se realizan en papel¹. La implementación de una herramienta como el simulador podría incrementar el nivel de eficiencia y veracidad de estas pruebas.

3. Ayudar a los controladores navales en sus funciones y facilitarles su trabajo

Aunque no sólo existen estas aplicaciones éstas son las principales y para las cuales ha sido diseñado el sistema. No obstante, hay que tener en cuenta que las funcionalidades del sistema pueden ir variando, y muy seguramente lo hagan, a medida que se descubran nuevas aplicaciones para las que el software pudiese ser útil. Incluso los avances tecnológicos, como pueden ser las gafas de realidad virtual, pueden suponer importantes y prometedoras posibilidades para el simulador, y de esta manera encontrar nuevas funcionalidades y aplicaciones.

Actualmente el número de barcos que se encuentran navegando aumenta considerablemente año tras año, como podemos observar en la tabla 1. Esto hace evidente que cada vez se necesitarán tecnologías más avanzadas y eficaces para el control de las embarcaciones, sobre todo en las zonas de tránsito muy elevado, como son las zonas portuarias y sus alrededores.

¹ <http://www.practicosdepuerto.es>, «<http://www.practicosdepuerto.es>,» [En línea]. Available: <http://www.practicosdepuerto.es/colegio-federacion/cualificacion>. [Último acceso: 25 03 2016].

Flota mercante									
YEAR		2008	2009	2010	2011	2012	2013	2014	2015
World	Total fleet	1117779	1192317	1276137	1415110	1532114	1625750	1691628	1749222
	Oil tankers	407881	418266	450053	439932	454349	472890	482017	489388
	Bulk carriers	391127	418356	456623	547192	624022	686635	726319	760468
	General carg	105492	108881	108232	81159	78138	77589	77552	76731
	Container sh	144655	161919	169158	183691	196821	206547	216345	227741
	Other types	68624	84895	92072	163135	178784	182090	189395	194893

fuente:UNCTAD

Tabla 1:Flota mercante mundial

Fuente: UNCTAD

Adicionalmente, cada vez se incrementarán más las necesidades de simular situaciones, tanto para probar distintos escenarios posibles, como para valorar que otras soluciones podían haberse aplicado en situaciones pasadas.

Gracias a estas simulaciones podrá comprobarse que efectivamente los controladores marítimos son aptos para desempeñar su función. Esto podría llevarse a cabo probándolos tanto en circunstancias reales como hipotéticas, en las que, los candidatos puedan demostrar su pericia en situaciones normales, e incluso en escenarios de peligro.

Además, podrán usarse estas simulaciones para comprobar que los patrones de las embarcaciones, tanto militares como civiles son capaces de entender y manejar una embarcación siguiendo las órdenes que se les indiquen y evitar cualquier situación de peligro.

Los simuladores navales son capaces de reducir enormemente los costes en los que incurren todas las actividades relacionadas con el control marítimo, mediante las simulaciones antes mencionadas de manejo de las embarcaciones o de regulación del tráfico. Esta reducción del coste se debe principalmente a que, aunque las personas en prácticas se equivoquen, no se incurre en ningún tipo de gasto o desperfecto, además de evitar cualquier peligro contra la integridad de las personas.

Asimismo, pueden realizarse estudios de cuáles son las mejores rutas a seguir por las embarcaciones para que en el trayecto se consuman la menor cantidad de recursos, como podrían ser tanto el combustible como el tiempo, con el consiguiente ahorro.

1.1.2 Ambientación

Actualmente los controladores navales son un colectivo reducido y altamente cualificado, siguen las Recomendaciones IALA² (*International Association of Marine Aids to Navigation and Lighthouse Authorities*) V-103 y Resolución 857 Asamblea 20 OMI³ (Organización Marítima Internacional), en la actualidad este colectivo cuenta con la ayuda de 241 prácticos en España repartidos en los 56 puertos que cuentan con estos servicios⁴, estos prácticos cuentan con una gran preparación y experiencia ya que las Recomendaciones IALA así lo determinan. Algunos de los sistemas de apoyo con los que cuentan estos prácticos en la actualidad son los *Vessel Traffic Services* (en adelante, VTS), de los que también se sirven los controladores navales.

Los controladores proveen a las embarcaciones de información relevante como puede ser la posición de otras embarcaciones, información meteorológica, etc. Además, se encargan del manejo del tráfico marítimo, dando fluidez a la circulación y evitando situaciones de riesgo.

Cuando una embarcación entra en la zona de un VTS informa a las autoridades por radio de esta entrada y estas se encargarán de monitorizar a la embarcación mediante sus instrumentos de seguimiento.

Las tareas que debe desempeñar un controlador de VTS son:

1. Comunicar, con suficiente antelación, a los navíos de cualquier tipo de información relevante concerniente al área, como podría ser el estado del tráfico, para que de esta manera pueda procederse a la toma de decisiones por parte de los capitanes de los navieros ayudados por el controlador, evitando así, cualquier tipo de situación de peligro en la medida de lo posible.
2. Toma de decisiones: los controladores marítimos deben contribuir a la toma de decisiones de las embarcaciones y seguir las consecuencias de cada una de estas, las atribuciones del controlador dependerán de la legislación de cada país.

² IALA, Recomendaciones IALA V-103, Sant-Germain-en-Laye, 2013.

³ OMI, Resolución 857 Asamblea 20 OMI, 1997.

⁴ <http://www.practicosdepuerto.es/el-colegio>, «<http://www.practicosdepuerto.es/el-colegio>,» 2016. [En línea]. Available: <http://www.practicosdepuerto.es/el-colegio>.

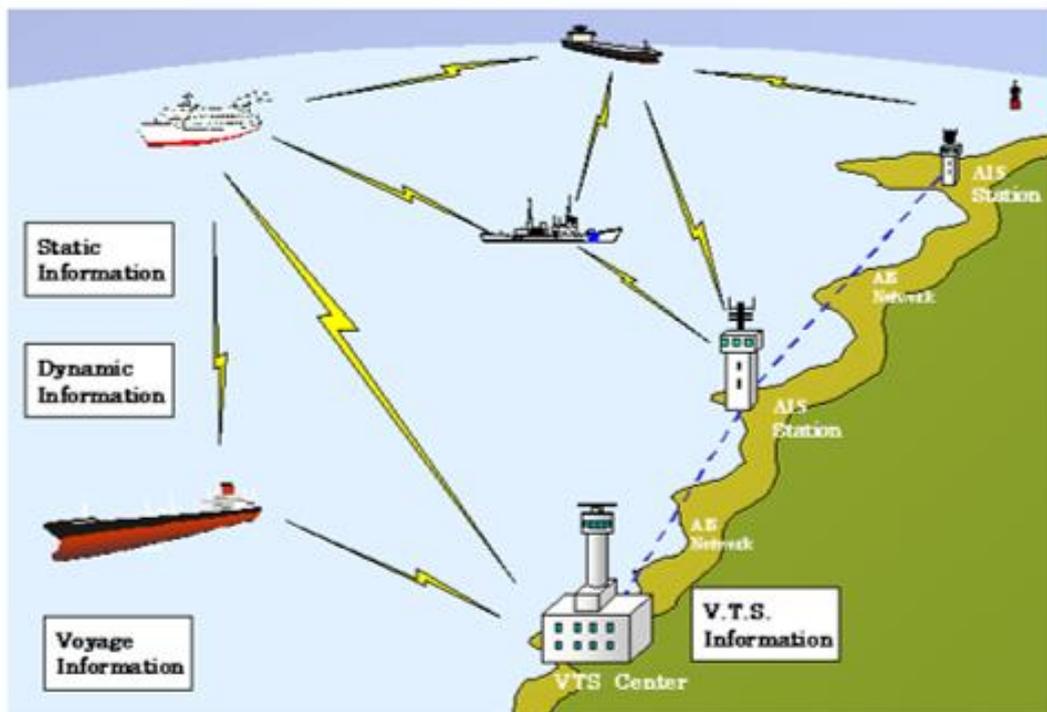
3. Organización, los controladores deberán facilitar el movimiento fluido y seguro de los buques, favoreciendo el tráfico en zonas de una alta concentración de flujo naval
4. Análisis de datos, la última tarea que debe desempeñar un controlador marítimo es la de coordinar la recopilación de datos, evaluarlos y distribuirlos debidamente para que sean lo más útiles y relevantes posibles.

Para desarrollar estos servicios el controlador cuenta dependiendo del tipo de puerto con los siguientes instrumentos:

1. *Automatic Identification System (AIS)*
2. Radar
3. *Radio Direction Finder (RDF)*
4. *Real-time Tracking*
5. *VHF Radio Position Reporting*

El uso de estos instrumentos está regulado por el Convenio SOLAS (*Safety of Life At Sea*), dictado por la OMI (Organización Marítima Internacional), en este convenio se especifica qué tipo de embarcaciones deben tener cada uno de estos sistemas a bordo, y de forma operativa, como también determina qué tipo de zonas costeras deben estar equipadas con sistemas de control para estos.

Ejemplo de AIS y VTS



Fuente: http://www6.kaiho.mlit.go.jp/kanmon/eng/mg_2.htm

Ilustración 1: VTS

Fuente: kaiho.mlit.go

1.2 Estructura del documento

Este documento está estructurado de manera gradual desarrollándose de manera análoga al proceso de diseño e implementación del software.

El documento está dividido en las siguientes secciones:

1. Introducción: en esta sección se determina en sus dos subsecciones cuales han sido las motivaciones encontradas para realizar el proyecto y el estado actual del mundo del control de embarcaciones marítimas y brevemente se muestran algunas instituciones encargadas de regular el control del tráfico marítimo.
2. Estructura del documento: En esta sección pretende explicarse cada una de las secciones del documento y así facilitar tanto la lectura y comprensión como la búsqueda de información dentro de este.

3. Objetivos: como bien indica el nombre de esta sección busca determinar e informar al lector sobre cuál es la finalidad de este proyecto y cuáles son las metas que se desean alcanzar.
4. Estado del arte: en esta sección se ha buscado la información relativa a los motores actuales, lo que ha servido para tomar una decisión sobre cual usar y cuál es la situación actual de otros simuladores navales y cuáles han sido sus puntos de vista del problema.
5. Desarrollo: En esta unidad se describe detalladamente el método en el que se ha ido desarrollando el trabajo.
6. Dificultades encontradas: aquí detallaremos cuales han sido las dificultades que se han presentado y como se han solucionado, si es que se ha podido solucionar.
7. Resultados: Se expone el resultado obtenido en la labor realizada y también se exponen los problemas encontrados y las partes no finalizadas en caso de haberlas.
8. Conclusiones y trabajos futuros: Se exponen las conclusiones que se han extraído de la realización del proyecto y se exponen futuras mejoras y ampliaciones que se le podrían realizar a este proyecto. Adicionalmente se estudiarán las posibles consecuencias de la implantación del sistema.
9. Planificación y presupuesto: En esta sección primeramente se detallará la planificación que se ha seguido para desarrollar este software y posteriormente para la realización del presupuesto se hará una estimación de los costes totales en los que se incurriría para la realización de este proyecto.
10. Bibliografía: Se hace referencia a todas las fuentes de las que se ha extraído información para la realización del proyecto.
11. Glosario: en esta sección únicamente se exponen todos los acrónimos y abreviaturas que se han utilizado en el documento y así en caso de que el lector no recuerde alguno o tenga alguna duda pueda consultarlos fácilmente.
12. Anexos: en esta sección se añaden un anexo con 10 páginas con la traducción de 4 las secciones que se debían escribir en inglés y otro anexo en el que se explica brevemente el funcionamiento del software acompañado con algunas imágenes para su mejor comprensión.

1.3 Objetivos

En este documento se estudiará cual es la situación actual del mundo marítimo, como se encuentran actualmente los controladores navales y cuáles son las ayudas de las que disponen. Asimismo, también se estudiarán otros simuladores navales para poder guiarnos en la realización del proyecto, y ver algunas posibles soluciones para nuestro proyecto.

Este proyecto pretende aportar una nueva herramienta al mundo marítimo, haciendo especial hincapié en la utilidad que podría suponer para los controladores marítimos, permitiéndoles tanto mejorar su observación, como la capacidad de predecir con mayor fiabilidad situaciones complejas de tráfico.

En este proyecto no nos centramos únicamente en estas utilidades, sino que pretende servir además como base a cualquier trabajo futuro en el ámbito de la simulación naval.

Por lo tanto, los objetivos de este proyecto son:

- El análisis de cómo podría mejorarse la capacidad de observación de los controladores del tráfico naval.
- Desarrollar ayudas mediante software para que los controladores detecten situaciones de peligro lo antes posible.
- Ayudar a la formación de estos controladores de una manera económica y segura
- Formar a los patrones de las embarcaciones para afrontar cualquier situación adversa.
- Desarrollar e implementar un software que sirva para futuros programas similares.

Capítulo 2

Estado del arte

En esta sección se pretende analizar y determinar cuál es el estado tanto de los motores gráficos actuales y elegir la mejor opción para desarrollar nuestro proyecto, así como comparar las diversas soluciones adoptadas en otros simuladores navales parecidos y aprender tanto de sus puntos fuertes como de sus puntos débiles.

Finalmente, esto nos permitirá aventurar los futuros problemas que podremos encontrarnos y sus soluciones, tanto las buenas como las no tan buenas y elegir la que mejor nos convenga en cada situación o encontrar una nueva solución más adecuada para cada problema.

2.1 Motores gráficos

Un motor gráfico es un software que consta de una serie de funcionalidades que permiten tanto el renderizado de imágenes como la inclusión de físicas, colisiones, sonidos, animación inteligencia artificial, administración de memoria, escenario gráfico y muchas otras extensiones dependiendo de la finalidad del sistema.

En este apartado, se realizará un breve repaso histórico a través de la evolución de estos sistemas, desde su origen hasta la actualidad.

Es importante destacar que la historia de los motores gráficos, está estrechamente ligada a los videojuegos. Esto se debe a que es la principal aplicación de estos sistemas y que se trata de un mercado altamente competitivo y en el que se facturan actualmente millones de dólares. Esta dependencia, puede llevar incluso a denominar a los motores gráficos como motores de videojuego.

Las empresas tratan de obtener una ventaja competitiva, desarrollando motores gráficos más y más avanzados que permitan a los nuevos juegos superar a la competencia.

Los primeros motores gráficos, surgidos a finales de los años 80, estaban desarrollados generalmente para un único videojuego, de manera que una vez este era lanzado al mercado el sistema era abandonado y no era actualizado salvo en el caso de una secuela del título anteriormente mencionado.

Existen excepciones destacadas, como es el motor SCUMM, que si bien fue creado para el juego Maniac Mansion (1987) finalmente y tras sucesivas actualizaciones fue implementado en una gran cantidad de aventuras gráficas de la compañía LucasArts.

La mayoría de estos motores eran difícilmente adaptables, pues estaban diseñados para un propósito muy específico, la creación del entorno y las físicas de un videojuego en concreto.

2.1.1 Unreal Engine

Todo esto cambió radicalmente con el motor Unreal Engine, si bien inicialmente fue desarrollado para el videojuego Unreal, del que toma nombre, su portabilidad y la facilidad de adaptación de la que hace gala este motor gráfico, lo convirtió en uno de los primeros motores gráficos genéricos, que podía ser implementado en una gran variedad de juegos y de diversa índole.

Su popularidad fue en aumento, sobre todo cuando tras una actualización incluso usuarios no profesionales lograban entornos gráficos de gran calidad utilizando este motor.

Sus principales bazas eran estar programado en C++, el lenguaje de programación más extendido y dominado por un mayor número de personas, así como la gran

portabilidad, que le ha llevado a estar presente en multitud de plataformas desde ordenadores de sobremesa a nuevas tecnologías como Oculus Rift.

Hasta la fecha existen 4 versiones de este motor de renderizado, si bien en ocasiones los cambios realizados por la empresa Epic Games, han sido tan extensos que prácticamente reescribían todo el código existente. En la actualidad Unreal Engine 4 continúa siendo un motor de referencia en la industria del videojuego.

Características técnicas:

Compatible con DirectX12, es junto con el motor de Crytek, que veremos a continuación el más potente del mercado, permite renderizados con más de un millón de partículas y goza de numerosas funcionalidades útiles a la hora de desarrollar vastos escenarios. Además de los apartados visuales, incluye físicas, sonido e incluso una inteligencia artificial muy avanzada.

Una característica muy importante del motor Unreal Engine es que se trata de una plataforma de código abierto, es decir una vez comprado el software, tendremos acceso a todo el código del motor gráfico permitiendo añadir tantas modificaciones y ampliaciones como deseemos.

2.1.2 Cry Engine

Desarrollado por la empresa CryTek, es el gran rival del Unreal Engine, surgió de la necesidad de crear un motor potente para el videojuego FarCry (2004), que mostrase la naturaleza de una isla de la manera más realista posible. El apartado gráfico de este título fue internacionalmente alabado, favoreciendo la aplicación de este motor en otros videojuegos para lograr gráficos de alto rendimiento. Sin embargo, la compra por parte de Ubisoft de la compañía redujo la utilización del motor solo a juegos propios. No fue hasta la tercera versión del motor cuando su popularidad en el mercado global aumentó, siendo considerado el motor con mayores prestaciones del mercado. Su principal inconveniente es la elevada dificultad de manejo, así como su alto coste que restringe el acceso a los grandes estudios.

Juegos como Crysis 3 o Ryse: Son of Rome son una buena prueba de la capacidad de este sistema llevado al límite.

Actualmente la empresa está trabajando en una nueva versión, denominada Cry Engine V, que espera situarse como el mejor motor gráfico del mercado.

Al contrario de Unreal Engine, el código no es abierto y la licencia para conseguir la versión completa del software de CryTek es ostensiblemente más elevada que la de cualquier otro motor gráfico del mercado.

Para conseguir la versión completa del software de CryTek es ostensiblemente más elevada que la de cualquier otro motor gráfico del mercado.

2.1.3 Panda3D

Diseñado por Disney su objetivo inicial era el de crear entornos para sus atracciones en 3D, posteriormente este motor fue utilizado para algunos videojuegos bajo la licencia Disney, no fue hasta 2002 cuando se liberalizó el código permitiendo su libre acceso.

Este motor gráfico funciona bajo una licencia de software libre, programado en C++, permite diseñar además de en este lenguaje en Python. Esto reduce significativamente la curva de aprendizaje de Panda 3D haciéndolo uno de los más populares en estudios independientes y en el ámbito académico. Una de las grandes bazas de este software es su activa comunidad que hace posible resolver cualquier duda del diseñador en poco tiempo funcionando como un servicio de ayuda en línea gratuito.

Su rendimiento si bien es inferior al logrado por las últimas versiones de Cry Engine o Unreal sigue siendo aceptable y permite crear escenarios 3D con un alto grado de detalle. Debido a sus orígenes está especialmente orientado a la creación de estéticas de dibujos animados.

2.1.4 Ogre3D

El software OGRE, fue diseñado como un software de renderizado en 3D, que permitiese la creación de entornos complejos ayudándose de librerías de objetos y texturas previamente diseñados. Al tratarse de Software libre se ha ido actualizando paulatinamente, incluyendo nuevas funcionalidades como físicas sencillas, gestor de memoria, etc. que le han acercado paulatinamente a los motores gráficos propiamente dichos.

El software original está diseñado en C++, aunque se pueden encontrar versiones en Python y Java. Su sistema modular de creación de objetos, permite generar entornos

muy variados y ricos en detalle, aunque sufre ciertas debilidades a la hora de incorporar movimientos y sonido en la escena.

2.1.5 Unity

Se trata de un motor gráfico relativamente joven, su primera versión se lanzó hace 10 años. El principio básico sobre el que se diseña este software es crear una herramienta de diseño gráfico portable que pueda ser implementada en la gran mayoría de plataformas existentes. Unity consigue este objetivo al poder ser utilizado en consolas, ordenadores de sobremesa, teléfonos inteligentes y Tablet. Esta característica le ha abierto un nicho de mercado y lo ha convertido rápidamente en uno de los motores gráficos más usados del momento.

Actualmente el motor se encuentra en su versión 5 y compite con el motor de CryTek y Unreal por la supremacía en el mercado, supliendo sus ligeramente peores prestaciones con su gran portabilidad y sencillez. Además, se trata de un sistema robusto programable en C++, que lo hace ideal para la creación de pequeños proyectos destinados a un gran número de usuarios como pueden ser aplicaciones móviles o juegos web.

Aunque la versión completa de Unity tiene un precio considerable de aproximadamente 1350€ o 67€/mes, la versión 5.3.0f4 Personal, que es la que se ha utilizado, resulta totalmente gratuita y ofrece una amplia gama de opciones y funcionalidades que la hacen apta para cualquier aplicación sin fines comerciales.

2.1.6 Conclusiones

Tabla resumen de los motores gráficos:

Motor gráfico	Versión gratuita	Físicas	Lenguajes de programación	Portabilidad
Unreal Engine	Si	Si	C++	Windows, Mac, iOS, Android, Xbox, PlayStation
Cry Engine	Si	Si	C++	Windows, Linux, PlayStation, Xbox, VR
Panda3D	Si	Si	Phyton/C++	Windows, Mac y Linux
Ogre3D	Si	No	C++	Windows, Mac, Linux, Android, iOS
Unity	Si	Si	Csharp	Windows, Mac, Linux, Android, VR

Tabla 2: Resumen de motores gráficos

Fuente: Elaboración propia, con datos de cada web

Tras una evaluación y análisis exhaustivo de los diferentes motores gráficos existentes, se han elaborado las siguientes conclusiones, gracias a las cuales, se ha podido elegir el motor gráfico más adecuado para el desarrollo del proyecto.

En primer lugar, es necesario un software gratuito o de muy bajo coste, debido a los limitados recursos económicos. De los analizados anteriormente, todos ellos cuentan con licencias gratuitas para productos sin fines comerciales, e incluso algunos como Unreal Engine, Panda3D u Ogre ofrecen el software completo de manera gratuita.

Otro aspecto relevante consiste en las funcionalidades requeridas, puesto que el proyecto consiste en un simulador naval orientado al ámbito profesional, no se exige un apartado gráfico vistoso o realista, pero si unas físicas precisas y un sistema fiable y robusto. Estos requisitos descartan al OGRE enfocado principalmente a la renderización de escenarios 3D.

La complejidad del motor es también un factor clave a la hora de elegirlo para el diseño del sistema, se cuenta con un único programador para realizar el proyecto, y además se espera facilidad en caso de futuras ampliaciones y actualizaciones. Debido a estas razones, el motor de CryTek no es especialmente recomendable puesto que consta de una curva de aprendizaje muy pronunciada y no está dirigido a pequeños proyectos sino a equipos expertos.

Por último, pero no menos importante, la portabilidad es considerada un aspecto clave en el diseño puesto que se valora la opción de aplicarlo tanto a ordenadores

personales como a tabletas e incluso a dispositivos portátiles como las Oculus Rift o similares. De todo el software analizado Unity es el que más destaca en este aspecto.

Si a todos los requisitos mencionados anteriormente unimos el hecho de que se puede partir de una versión muy básica del sistema ya implementado en Unity, este es el motor gráfico elegido para el desarrollo del proyecto.

Su versión 5.3.0f4 Personal resulta gratuita, proporciona todas las funcionalidades necesarias para el proyecto. Además, el lenguaje de programación empleado es básicamente C++, un lenguaje que el programador ya conoce y domina por lo que no es necesario un gran proceso de adaptación.

2.2 Simuladores navales

Un simulador es, según la Real Academia Española (en adelante, RAE), “*un aparato que reproduce el comportamiento de un sistema en determinadas condiciones, aplicado generalmente para el entrenamiento de quienes deben manejar dicho sistema.*”⁵, por ello un simulador naval es un aparato que reproduce el comportamiento de un sistema naval en unas determinadas condiciones para el entrenamiento de las destrezas de los que deben manejar el sistema.

En este apartado analizaremos los principales simuladores navales actuales realizando un análisis de cómo es cada uno de ellos y cuáles son sus puntos fuertes y débiles.

Los primeros simuladores virtuales, surgieron en la década de los 80, estaban centrados principalmente en el ámbito aéreo y más concretamente en el entrenamiento de pilotos militares. El algoritmo diseñado por Roberts, que permitía eliminar y ocultar superficies ocultas en una imagen, favoreció la expansión de sistemas de simulación más complejos y realistas.

En el marco de la simulación naval, uno de los pioneros fue la versión computarizada del juego de guerra Harpoon, utilizada por la marina estadounidense y que permitía la representación de buques, sistemas de detección y proyectiles. Fue a principios de la década de los noventa, cuando comenzaron a aparecer diversos títulos

⁵ RAE, <<http://www.rae.es/>> [En línea]. Available: <http://dle.rae.es/?id=Xw14yph>. [Último acceso: 27 03 2016].

comerciales de simulación diseñados para PC, inicialmente en un entorno minimalista en 2D en el que primaba el componente estratégico.



Ilustración 2: captura del videojuego Harpoon 3

Fuente: <http://www.matrixgames.com/products/323/details/Harpoon.3-.Advanced.Naval.Warfare>

El mayor impulso de los simuladores lo proporcionó la saga Silent Hunter, las primeras versiones se basaban en la guerra submarina, sin embargo, pronto se expandió la franquicia a barcos convencionales y se pasó del 2D, a la simulación realista en 3 dimensiones, aún hoy la saga es un referente en su campo.



Ilustración 3: captura del videojuego Silent Hunter

Fuente: <http://www.silenthunteronline.com/>

Hay que señalar el hecho de que los mayores avances tecnológicos del momento en este ámbito pertenecen a la investigación militar y se encuentran bajo un alto grado de secretismo. En España la empresa con mayor inversión en este ámbito es la tecnológica Indra, que recientemente ha presentado un nuevo simulador, capaz de modelizar fielmente el manejo de un buque, además este sistema incluye un hardware de control que refleja fielmente el puente de mando de un buque. Este producto es el empleado por la marina española para la formación de sus hombres.



Ilustración 4: Hardware simulando puente de mando

Fuente: <http://www.indracompany.com/es/simuladores-terrestres-navales>

En un nivel más simple, la empresa SAES ha presentado un simulador táctico naval, que permite recrear numerosas situaciones de emergencia y evaluar las aptitudes de los usuarios.

Por último en el marco civil, los simuladores se orientan principalmente al mercado de juegos de PC, en los que se deben realizar misiones civiles, tales como pesca en alta mar, transporte de mercancías o viajeros, remolcador de un puerto etc. En este mercado destacan títulos como European Ship Simulator, que permite al jugador elegir entre una enorme galería de barcos y realizar todo tipo de interacciones con ellos, una parte importante del juego se desarrolla en los puertos en donde se puede observar el funcionamiento de un gran puerto comercial e incluso es posible dirigir el tráfico marítimo. Es quizás esta última funcionalidad, en la que, salvando las distancias, el juego se asemeje más al software desarrollado en este documento.



Ilustración 5: captura del European Ship Simulator

Fuente: <http://store.steampowered.com/app/299250/?l=spanish>

Dado que todos los títulos analizados o bien pertenecen a la industria militar o bien son juegos destinados a la venta, sus sistemas de modelado y diseño son confidenciales y por tanto no resultan de gran ayuda a la hora de acometer un nuevo proyecto de simulación naval, aparte de establecer el nivel de calidad y rendimiento de los competidores.

Capítulo 3

Desarrollo

En esta sección se expondrá de qué forma se ha ido desarrollando el proyecto y todas las cuestiones relacionadas con este, tanto las herramientas usadas para su desarrollo, como la estructura del proyecto. En el marco normativo se analizarán todas las partes del proyecto para ver cuales tienen algún tipo de normativa legal que las regula.

3.1 Herramientas utilizadas

Las herramientas utilizadas para realizar este proyecto han sido tanto medios software como hardware, por lo tanto, los separaremos en estas dos categorías y especificaremos en cada caso cuales han sido, así como la funcionalidad que han tenido para el desarrollo del proyecto.

3.1.1 Medios hardware

Como medios hardware únicamente se ha hecho uso de un ordenador portátil personal con las siguientes especificaciones técnicas:

Especificaciones técnicas del portátil

Sistema Operativo	Procesador	Memoria RAM	Disco Duro
Windows 10 home edition	Intel® Core™ i7-5500U CPU @ 2.40 GHz 2.40GHz	4,00 GB	450 GB

Tabla 3: Especificaciones portátil

Fuente: Elaboración propia, con datos obtenidos del propio ordenador

3.1.2 Medios software

En cuanto a los medios software podemos encontrarnos varias aplicaciones que nos han ayudado a realizar este proyecto:



La versión del software que se ha utilizado es la versión gratuita, 5.3.0f4 Personal, que se otorga a los estudiantes para que puedan desarrollar libremente sus habilidades. Esta versión para los estudiantes cuenta con todas las funcionalidades deseadas para desarrollar este proyecto.

Se ha elegido Unity para desarrollar este proyecto como ya se ha mencionado antes debido principalmente a que se disponía de una licencia gratuita para desarrollarlo además de contar con todas las características deseables para la realización de este proyecto.

Adicionalmente el lenguaje empleado por Unity, es similar a C++, por lo que la curva de aprendizaje no debería ser muy pronunciada.

Por último, como ya se ha mencionado, se contaba con una versión que ya se disponía de las mecánicas del barco para poder implementar, así pues, todas las funcionalidades deseadas sin tener que desarrollar toda la mecánica del barco.



Esta aplicación ha servido para modificar los scripts que se utilizaban dentro del programa de Unity. Se ha usado la versión 5.9.6 de este software, este software es gratuito y viene integrado dentro de Unity, siendo esta una versión personalizada por Unity del software.

Además, trae un depurador de código dentro del propio editor de scripts, lo que resulta muy útil para testear que no exista ningún problema antes de ejecutar el programa en Unity de nuevo.

En la propia página de Unity recomiendan que se utilice este software, por lo que al ser libre y ser gratuito se ha determinado que este software era adecuado.



Este software, únicamente ha sido utilizado para crear y modificar los archivos del formato .txt que determinaban la situación deseada que debía mostrarse en Unity.

Se ha utilizado la versión 6.9.1 y como en el caso anterior se trata de un software libre y gratuito.



Este software, únicamente ha sido utilizado para crear el diagrama de la planificación, se ha usado la versión online del software ya que era más que suficiente para realizar la planificación. Posteriormente se ha exportado la planificación y se ha introducido en la memoria.

3.2 Estructura del proyecto⁶⁷

En esta sección se describirá el proceso que se ha llevado a cabo para realizar el proyecto, para realizarlo se ha seguido un modelo estructurado de la manera de un ciclo de vida iterativo e incremental guiado por casos de uso. Esto quiere decir que en cada ciclo de vida se obtiene un prototipo funcional y a partir de este se recibe una retroalimentación por parte del usuario/s final/es y se vuelve a empezar desde la fase de análisis.

El modelo de ciclo de vida que se ha seguido para este proyecto ha sido el siguiente:

⁶ M.-I. Sanchez Segura y A. Mora-Soto, «ocw.uc3m.es,» 01 08 2012. [En línea]. Available: <http://ocw.uc3m.es/ingenieria-informatica/metodologia-de-desarrollo-visual/course-files>.

⁷ R. Colomo, «OCW - UC3M,» 12 03 2013. [En línea]. Available: http://ocw.uc3m.es/ingenieria-informatica/ingeniera-del-software-iii/materialclase/ISIII_07_ASI.pdf.

Ciclo de vida de un proyecto

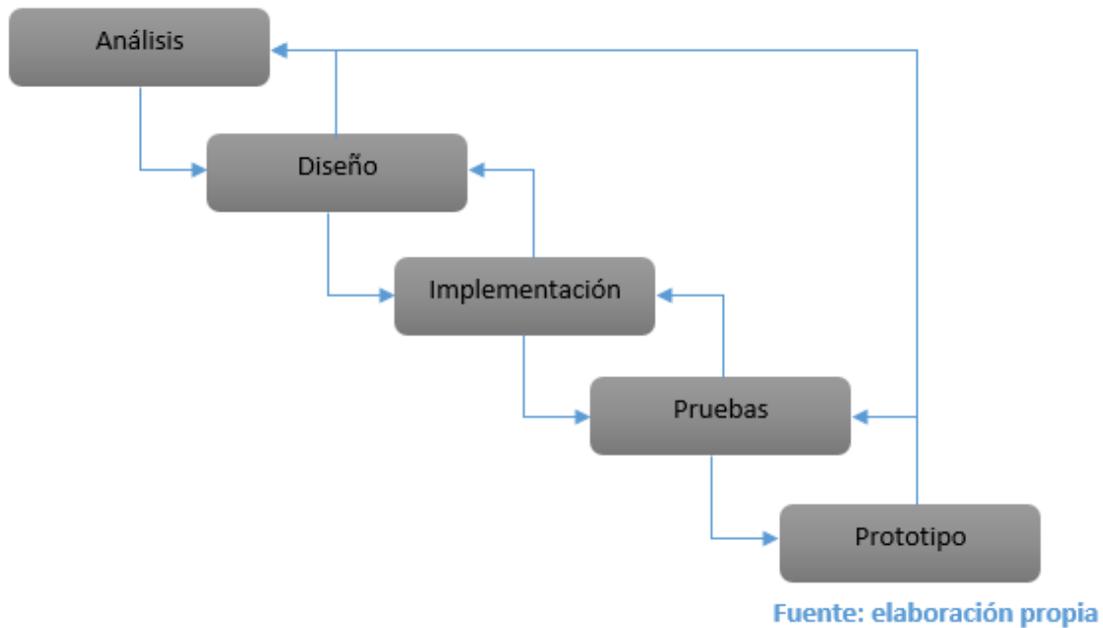


Ilustración 6: Ciclo de vida del proyecto

Fuente: Elaboración propia

Como podemos observar en la ilustración superior se trata de un modelo del tipo ciclo de vida iterativo e incremental, en el que poco a poco se va pasando por cada una de las fases de arriba abajo, pero con el punto añadido de que en cada fase se realizan revisiones de las anteriores y se modifican en caso de ser necesario.

Cuando se llega a la última fase de este ciclo de vida se dispone de un prototipo funcional básico el cual se verifica y una vez verificado se vuelve a empezar corrigiendo los errores encontrados y ampliando sus funcionalidades.

Las fases de las que consta este modelo son las de la imagen superior y ahora se detallará cual es la función de cada una de ellas:

- **Análisis**

En esta fase se estudia el problema hasta comprenderlo adecuadamente. Se analizan tanto los requerimientos de este como las necesidades que debe satisfacer el sistema.

En esta fase se establecen y determinan los requisitos del sistema.

- **Diseño**

Se diseñará la estructura de la solución y se identificarán los distintos módulos de los que debe constar el proyecto y así poder estructurar el trabajo en bloques para poder realizar el trabajo por funcionalidades.

- **Implementación**

Se implementará el código fuente de la futura aplicación siguiendo la estructura del diseño realizado y haciendo especial hincapié en los requisitos indispensables que se han determinado en la fase de análisis.

En esta fase se buscan soluciones para poder realizar el trabajo de la forma más satisfactoria posible y se intentará reutilizar código.

- **Pruebas**

Se unirán todas las partes del sistema para componer el sistema final, esto permitirá comprobar que el sistema funciona correctamente y que cumple con los requisitos.

Después de esta última fase si el sistema ya ha realizado todas las iteraciones necesarias y cumple con todos los requisitos se entra en la fase de verificación que está integrada en la fase de pruebas, en esta fase se entregará la versión final al usuario final para que realice pruebas exhaustivas y de esta forma verificar que el sistema funciona de una manera satisfactoria cumpliendo todos los objetivos.

3.3 Estudio de viabilidad del sistema^{8 9}

En esta sección se analizarán las necesidades del usuario y se propondrá una solución acorde al problema planteado por este mediante la creación de los requisitos que debe cumplir el sistema que se implemente como solución a este.

⁸ R. Colomo, «OCW - UC3M,» 12 03 2013. [En línea]. Available: http://ocw.uc3m.es/ingenieria-informatica/ingeniera-del-software-iii/materialclase/ISIII_06_EVS.pdf

⁹ F. Berzal Galiano, «Universidad de granada,» [En línea]. Available: <http://elvex.ugr.es/idbis/db/docs/design/2-requirements.pdf>

3.3.1 Establecimiento del alcance del sistema

Se presenta, a continuación, la solicitud presentada por parte del usuario y el análisis realizado respecto a este, determinando así los requisitos que debe cumplir el proyecto.

El establecimiento del alcance del sistema se divide en tres fases bien diferenciadas. El estudio de la solicitud, que es una descripción de las necesidades, requisitos y restricciones con los que deberá cumplir el sistema final. Seguidamente la identificación del alcance del sistema, punto en el que se analizara el contexto del proyecto y los usuarios a los que se destina. Finalmente, se realizará la especificación del alcance del sistema, donde se estipularán los objetivos y el plan de trabajo que deberá llevarse a cabo.

1. Estudio de la solicitud:

Se ha solicitado que se desarrolle un software que permita la simulación de trayectorias navales.

Se espera que este software sea capaz de contener varias embarcaciones y la ruta que debe seguir cada una. Cada embarcación debe contar con un piloto automático para que siga la ruta especificada sin ningún tipo de interacción humana. Las posiciones y las rutas de los barcos deben ser introducidas en el sistema mediante el uso de un fichero externo en el formato .txt.

En cuanto a los aspectos visuales, el sistema debe mostrar la velocidad, la potencia a la que actualmente se encuentra el motor y la posición del timón. Además, esta información debe encontrarse en un panel visible sobre cada uno de los barcos, que sea distinguible y que siempre se encuentre mirando en dirección a la cámara.

El sistema debe simular la navegación de las embarcaciones de una manera similar a la real, en un entorno en el que solo se encuentren las embarcaciones. Este software debe estar abierto a nuevas modificaciones para probar nuevos algoritmos de todo tipo o distintas embarcaciones.

El sistema debe intentar tener funcionalidades abiertas para interactuar con las gafas de realidad virtual. Y en cualquier caso se implementen o no estas funcionalidades el desarrollo de la simulación debe ser fluido y natural.

Si el sistema puede ser implementado en puertos comerciales, el coste podría ser elevado, pero hay que tener en cuenta que debe ser muy fiable y robusto pues un fallo en el software puede suponer pérdidas millonarias.

En cuanto al marco normativo se ha realizado una búsqueda de posible normativa aplicable a este software en concreto, pero después de realizar una búsqueda exhaustiva no se ha encontrado ninguna referencia a ninguna legislación particular que deba seguir y cumplir este software.

2. Identificación del alcance del sistema:

El sistema de simulación de trayectorias de barcos será fácil de usar para cualquier tipo de usuario que desee realizar una simulación naval.

En el software resulta bastante sencillo realizar modificaciones de la cantidad de embarcaciones, la posición de esta y las rutas de estas únicamente modificando el fichero .txt para que se realicen estas operaciones.

Si se desea realizar el pilotaje de un único barco y deshabilitar el piloto automático bastará con pulsar el botón, en ese momento aparecerá un barco en la pantalla y podrá ser pilotado de manera manual.

Para testear nuevas embarcaciones y distintos algoritmos únicamente hay que sustituir el *prefab* o modificar el script para usar otro *prefab* o ambos. En el caso de querer probar otros algoritmos, como podría ser el del piloto automático, habría que modificarlo directamente en el código, tarea sencilla para los usuarios que tengan conocimientos de informática. Al verse capacitados para desarrollar este nuevo piloto automático u otro algoritmo, se les presupone aptos.

3. Especificación del alcance del sistema:

El análisis de la situación actual se ha realizado antes de iniciar el estudio de la viabilidad del sistema, se han analizado los posibles motores gráficos que se podían usar, eligiéndose el motor gráfico de Unity. Además, se ha analizado otros simuladores navales y el marco actual en el que se encuentran actualmente las normativas de control del tráfico marítimo.

Este estudio, se ha realizado para detallar cuáles eran las mejores soluciones que se podían tomar en cada caso, tanto para saber cuál era el motor gráfico más adecuado, como para establecer qué soluciones se han tomado en otros proyectos similares.

Los clientes del sistema serían principalmente las autoridades portuarias civiles, que buscarán un sistema fiable y eficaz para mejorar el tráfico marítimo, así como un software para realizar test para evaluar a futuros patrones de barco o controladores.

Los *stakeholders* del sistema son principalmente los controladores del tráfico marítimo, que pueden realizar prácticas realísticas y simulaciones para mejorar en su desempeño, además que este sistema puede ayudarles a desempeñar su trabajo proveyéndoles de una visión ampliada de lo que acontece en su zona de control. También encontramos a usuarios que esperan poder testear las habilidades de los futuros controladores de tráfico o de los patrones de las embarcaciones.

3.3.2 Definición de requisitos del sistema

En esta actividad se fijarán los requisitos con los que debe cumplir el sistema, estos requisitos se planificarán a partir de la definición del estudio de la solicitud realizado anteriormente que ha sido obtenida de los usuarios participantes en el proyecto.

Los requisitos se dividirán en dos categorías principales, los requisitos funcionales y los requisitos no funcionales, estas categorías cuentan con subcategorías que se determinan a continuación:

1. Requisitos funcionales: estos requisitos especifican que es exactamente lo que se espera que sea capaz de realizar nuestro software, estos requisitos se subdividen en otros subgrupos:
 - 1.1. Requisitos de funcionamiento: Determinan que funcionalidades debe tener el sistema y cuáles no.
 - 1.2. Requisitos de interfaz: Determinan de qué forma se muestra la información al usuario y como puede interactuar este usuario con el sistema.
2. Requisitos no funcionales: estos requisitos especifican como ser el sistema, dentro de este grupo existen varios subgrupos que son:
 - 2.1. Requisitos de rendimiento: en estos requisitos se determinarán aspectos relacionados con el rendimiento de la aplicación.
 - 2.2. Requisitos de seguridad: requisitos conforme a la forma en que se trataran los datos sensibles y la información por parte de nuestro sistema.
 - 2.3. Requisitos de portabilidad: requisitos referentes a la portabilidad de nuestro sistema.

Para definir los requisitos tanto los funcionales como los no funcionales se usará una tabla basada en la que se proporciona por el estándar IEEE 830 pero modificada ligeramente, quedando finalmente una tabla de la siguiente forma:

Modelo de tabla para los requisitos

Id del requisito	RT-Smn.n
Nombre del requisito	Nombre único
Tipo	Funcionamiento, interfaz, rendimiento, seguridad o portabilidad
Fuente del requisito	Cliente o Analista
Descripción del requisito	Descripción
Prioridad del requisito	Esencial, deseado u opcional

Tabla 4: Modelo tabla

Fuente: Elaboración propia

Id del requisito, aquí especificaremos un identificador único para cada requisito, que funcionará de la siguiente manera:

RT-S-n.n

La R se mantendrá para todos los requisitos ya que indica que se trata de un requisito.

La T determina el tipo de requisito del que se trata, esto puede ser:

1. F = funcional.
2. NF= no funcional

La S indicará el subtipo al que hace referencia el requisito, existen las siguientes posibilidades para esta variable:

1. F = funcionamiento
2. I = interfaz
3. R = rendimiento
4. S = seguridad
5. P = portabilidad

Las n indican la numeración de los requisitos dentro de cada subtipo, el ‘n’ indica que dentro de un requisito pueden encontrarse especificaciones más detalladas de este requisito.

El nombre del requisito identifica inequívocamente al requisito y determina resumidamente el contenido del mismo.

Tipo, el tipo de requisito hace referencia a el subtipo de requisito en el que esta englobado.

Fuente del requisito, puede tomar los siguientes valores: cliente o analista, que indican quien ha fijado ese requisito para el sistema.

Descripción indica cual es el requisito detalladamente para ayudar a los programadores a entenderlo y a ponerlo en marcha.

Prioridad, tendrá los siguientes valores:

1. Esencial
2. Deseado
3. Opcional

3.3.2.1 Requisitos funcionales

Requisitos de funcionamiento:

Id del requisito	RF-F-1.0
Nombre del requisito	Ficheros de datos
Tipo	Funcionamiento
Fuente del requisito	Cliente
Descripción del requisito	El sistema debe ser capaz de obtener los datos de un fichero externo
Prioridad del requisito	Esencial

Tabla 5: RF-F-1.0

Fuente: Elaboración propia

Id del requisito	RF-F-1.1
Nombre del requisito	Formato de datos: crear embarcación
Tipo	Funcionamiento
Fuente del requisito	Analista
Descripción del requisito	El sistema debe obtener los datos de la siguiente forma: nombre posición x, posición y, dirección que determinara una nueva embarcación con su posición y su dirección
Prioridad del requisito	Esencial

Tabla 6: RF-F-1.1

Fuente: Elaboración propia

Id del requisito	RF-F-1.2
Nombre del requisito	Formato de datos: ruta embarcación
Tipo	Funcionamiento
Fuente del requisito	Analista
Descripción del requisito	El sistema debe obtener los datos de la siguiente forma: posición x, posición y fijará un nuevo punto en la ruta del ultimo barco creado.
Prioridad del requisito	Esencial

Tabla 7: RF-F-1.2

Fuente: Elaboración propia

Id del requisito	RF-F-1.3
Nombre del requisito	Formato de datos: posición cámara
Tipo	Funcionamiento
Fuente del requisito	Analista
Descripción del requisito	El sistema debe obtener los datos de la siguiente forma: Cámara posición x, posición y, posición z fijará un nuevo punto donde se situará la cámara.
Prioridad del requisito	Esencial

Tabla 8: RF-F-1.3

Fuente: Elaboración propia

Id del requisito	RF-F-1.4
Nombre del requisito	Formato de datos: rotación cámara
Tipo	Funcionamiento
Fuente del requisito	Analista
Descripción del requisito	El sistema debe obtener los datos de la siguiente forma: rotación x, rotación y, rotación z fijará la dirección en la que mira la cámara.
Prioridad del requisito	Esencial

Tabla 9: RF-F-1.4

Fuente: Elaboración propia

Id del requisito	RF-F-2.0
Nombre del requisito	Movimientos
Tipo	Funcionamiento
Fuente del requisito	Cliente
Descripción del requisito	El movimiento de las embarcaciones debe ser realista
Prioridad del requisito	Esencial

Tabla 10: RF-F-2.0

Fuente: Elaboración propia

Id del requisito	RF-F-3.0
Nombre del requisito	Piloto automático
Tipo	Funcionamiento
Fuente del requisito	Cliente
Descripción del requisito	Las embarcaciones deben contar con un piloto automático.
Prioridad del requisito	Esencial

Tabla 11: RF-F-3.0

Fuente: Elaboración propia

Id del requisito	RF-F-3.1
Nombre del requisito	Piloto automático dirección
Tipo	Funcionamiento
Fuente del requisito	Cliente
Descripción del requisito	El piloto automático debe modificar tanto la velocidad como la dirección de la embarcación para seguir los puntos de la ruta
Prioridad del requisito	Esencial

Tabla 12: RF-F-3.1

Fuente: Elaboración propia

Id del requisito	RF-F-4.0
Nombre del requisito	Movimiento cámara
Tipo	Funcionamiento
Fuente del requisito	Cliente
Descripción del requisito	La cámara debe tener la posibilidad de moverse.
Prioridad del requisito	Esencial

Tabla 13: RF-F-4.0

Fuente: Elaboración propia

Id del requisito	RF-F-4.1
Nombre del requisito	Rotar cámara
Tipo	Funcionamiento
Fuente del requisito	Cliente
Descripción del requisito	La cámara debe tener la posibilidad de rotar.
Prioridad del requisito	Esencial

Tabla 14: RF-F-4.1

Fuente: Elaboración propia

Id del requisito	RF-F-5.0
Nombre del requisito	Colisiones
Tipo	Funcionamiento
Fuente del requisito	Cliente
Descripción del requisito	El sistema debe detectar las colisiones entre dos embarcaciones y no permitir que se atraviesen.
Prioridad del requisito	Esencial

Tabla 15: RF-F-5.0

Fuente: Elaboración propia

Id del requisito	RF-F-6.0
Nombre del requisito	Inicializaciones
Tipo	Funcionamiento
Fuente del requisito	Cliente
Descripción del requisito	El sistema debe permitir que se acceda varias veces al fichero para traer nuevos barcos o mover la cámara.
Prioridad del requisito	Esencial

Tabla 16: RF-F-6.0

Fuente: Elaboración propia

Id del requisito	RF-F-7.0
Nombre del requisito	Embarcación manual
Tipo	Funcionamiento
Fuente del requisito	Cliente
Descripción del requisito	El sistema debe permitir que se cree como máximo una embarcación con control manual.
Prioridad del requisito	Esencial

Tabla 17: RF-F-7.0

Fuente: Elaboración propia

Id del requisito	RF-F-7.1
Nombre del requisito	Embarcación manual manejo velocidad
Tipo	Funcionamiento
Fuente del requisito	Cliente
Descripción del requisito	El usuario debe ser capaz de manejar la velocidad de la embarcación.
Prioridad del requisito	Esencial

Tabla 18: RF-F-7.1

Fuente: Elaboración propia

Id del requisito	RF-F-7.2
Nombre del requisito	Embarcación manual manejo dirección
Tipo	Funcionamiento
Fuente del requisito	Cliente
Descripción del requisito	El usuario debe ser capaz de manejar la dirección (timón) de la embarcación.
Prioridad del requisito	Esencial

Tabla 19: RF-F-7.2

Fuente: Elaboración propia

Id del requisito	RF-F-7.3
Nombre del requisito	Embarcación manual ruta
Tipo	Funcionamiento
Fuente del requisito	Cliente
Descripción del requisito	El usuario debe ser capaz de insertar una ruta manualmente a su embarcación mediante el uso del ratón.
Prioridad del requisito	Esencial

Tabla 20: RF-F-7.3

Fuente: Elaboración propia

3.3.2.2 Requisitos no funcionales

Requisitos de rendimiento:

Id del requisito	RNF-R-1.0
Nombre del requisito	Renderizado
Tipo	Rendimiento
Fuente del requisito	Analista
Descripción del requisito	La simulación debe ejecutarse con fluidez en los entornos para los que se desarrolla.
Prioridad del requisito	Deseable

Tabla 21: RNF-R-1.0

Fuente: Elaboración propia

Id del requisito	RNF-R-1.1
Nombre del requisito	Datos de los barcos
Tipo	Rendimiento
Fuente del requisito	Analista
Descripción del requisito	La simulación debe acceder cada <i>frame</i> a la información de las embarcaciones para actualizarla.
Prioridad del requisito	Esencial

Tabla 22: RNF-R-1.1

Fuente: Elaboración propia

Requisitos de interfaz:

Id del requisito	RNF-I-1.0
Nombre del requisito	Botón inicio de la simulación
Tipo	Interfaz
Fuente del requisito	Cliente
Descripción del requisito	El sistema debe contar con un botón con el nombre inicial para que se creen las embarcaciones.
Prioridad del requisito	Esencial

Tabla 23: RNF-I-1.0

Fuente: Elaboración propia

Id del requisito	RNF-I-1.1
Nombre del requisito	Botón inicio de la simulación posición y dirección
Tipo	Interfaz
Fuente del requisito	Cliente
Descripción del requisito	El sistema mostrará las embarcaciones en la posición y en la dirección que se le indica por fichero.
Prioridad del requisito	Esencial

Tabla 24: RNF-I-1.1

Fuente: Elaboración propia

Id del requisito	RNF-I-2.0
Nombre del requisito	Botón embarcación manual
Tipo	Interfaz
Fuente del requisito	Cliente
Descripción del requisito	El sistema debe ser capaz de generar una embarcación para ser majeada manualmente
Prioridad del requisito	Esencial

Tabla 25: RNF-I-2.0

Fuente: Elaboración propia

Id del requisito	RNF-I-2.1
Nombre del requisito	Desaparición botón embarcación manual
Tipo	Interfaz
Fuente del requisito	Analista
Descripción del requisito	Si se crea dicha embarcación debe desaparecer el botón
Prioridad del requisito	Esencial

Tabla 26: RNF-I-2.1

Fuente: Elaboración propia

Id del requisito	RNF-I-3.0
Nombre del requisito	Cuadro informativo
Tipo	Interfaz
Fuente del requisito	Cliente
Descripción del requisito	Cada embarcación contará con un cuadro informativo sobre esta.
Prioridad del requisito	Esencial

Tabla 27: RNF-I-3.0

Fuente: Elaboración propia

Id del requisito	RNF-I-3.1
Nombre del requisito	Cuadro informativo diferenciado
Tipo	Interfaz
Fuente del requisito	Cliente
Descripción del requisito	La embarcación de control manual contará con un cuadro con un color distinto y diferenciable del resto de embarcaciones.
Prioridad del requisito	Esencial

Tabla 28:RNF-I-3.1

Fuente: Elaboración propia

Id del requisito	RNF-I-3.2
Nombre del requisito	Velocidad, latitud, longitud y rumbo
Tipo	Interfaz
Fuente del requisito	Cliente
Descripción del requisito	El sistema debe mostrar la velocidad, latitud, longitud y rumbo de cada embarcación en el cuadro informativo
Prioridad del requisito	Esencial

Tabla 29: RNF-I-3.2

Fuente: Elaboración propia

Id del requisito	RNF-I03.3
Nombre del requisito	Dirección de los cuadros
Tipo	Interfaz
Fuente del requisito	Cliente
Descripción del requisito	Los rótulos que muestren información sobre las embarcaciones deben mirar siempre hacia la cámara
Prioridad del requisito	Esencial

Tabla 30: RNF-I-3.3

Fuente: Elaboración propia

Id del requisito	RNF-I-3.4
Nombre del requisito	Cuadros visibles
Tipo	Interfaz
Fuente del requisito	Cliente
Descripción del requisito	Los rótulos que muestren información sobre las embarcaciones deben tener un color que permita su visibilidad.
Prioridad del requisito	Esencial

Tabla 31: RNF-I-3.4

Fuente: Elaboración propia

Id del requisito	RNF-I-4.0
Nombre del requisito	Ruta
Tipo	Interfaz
Fuente del requisito	Cliente
Descripción del requisito	El sistema debe mostrar una línea saliente de cada embarcación y pasando por cada uno de los puntos de la ruta de este
Prioridad del requisito	Esencial

Tabla 32: RNF-I-4.0

Fuente: Elaboración propia

Id del requisito	RNF-I-4.1
Nombre del requisito	Ruta embarcación manual
Tipo	Interfaz
Fuente del requisito	Cliente
Descripción del requisito	El sistema debe mostrar una línea saliente de la embarcación manual con su ruta de color diferente al resto de embarcaciones
Prioridad del requisito	Esencial

Tabla 33: RNF-I-4.1

Fuente: Elaboración propia

Id del requisito	RNF-I-4.2
Nombre del requisito	Rutas coincidentes
Tipo	Interfaz
Fuente del requisito	Cliente
Descripción del requisito	En el caso de que dos rutas pasen por el mismo punto estas deben cambiar de color.
Prioridad del requisito	Deseable

Tabla 34: RNF-I-4.2

Fuente: Elaboración propia

Id del requisito	RNF-I-5.0
Nombre del requisito	Embarcación
Tipo	Interfaz
Fuente del requisito	Cliente
Descripción del requisito	Las embarcaciones deben tener un aspecto similar al de una embarcación real.
Prioridad del requisito	Esencial

Tabla 35: RNF-I-5.0

Fuente: Elaboración propia

Id del requisito	RNF-I-6.0
Nombre del requisito	Agua
Tipo	Interfaz
Fuente del requisito	Cliente
Descripción del requisito	El agua deberá tener una apariencia realista.
Prioridad del requisito	Deseable

Tabla 36: RNF-I-6.0

Fuente: Elaboración propia

Requisitos de seguridad:

Id del requisito	RNF-S-1.0
Nombre del requisito	Datos
Tipo	Rendimiento
Fuente del requisito	Analista
Descripción del requisito	Los datos de cada ejecución deben eliminarse al finalizar la ejecución y no almacenarse en ningún caso.
Prioridad del requisito	Deseable

Tabla 37: RNF-S-1.0

Fuente: Elaboración propia

Requisitos de portabilidad:

Id del requisito	RNF-P-1.0
Nombre del requisito	Windows
Tipo	Portabilidad
Fuente del requisito	Cliente
Descripción del requisito	El programa debe funcionar en cualquier equipo que cuente con el sistema operativo Windows 7 o superiores.
Prioridad del requisito	Esencial

Tabla 38: RNF-P-1.0**Fuente: Elaboración propia**

Id del requisito	RNF-P-2.0
Nombre del requisito	Mac
Tipo	Portabilidad
Fuente del requisito	Cliente
Descripción del requisito	El programa debe funcionar en cualquier equipo que cuente con el sistema operativo MAC os 9 o superiores.
Prioridad del requisito	Deseable

Tabla 39: RNF-P-2.0**Fuente: Elaboración propia**

Id del requisito	RNF-P-3.0
Nombre del requisito	VR
Tipo	Funcionamiento
Fuente del requisito	Cliente
Descripción del requisito	El sistema debe intentar ser compatible con el uso de gafas de realidad virtual.
Prioridad del requisito	Opcional

Tabla 40: RNF-P-3.0**Fuente: Elaboración propia**

3.4 Diagrama de casos de uso

Diagrama de casos de uso

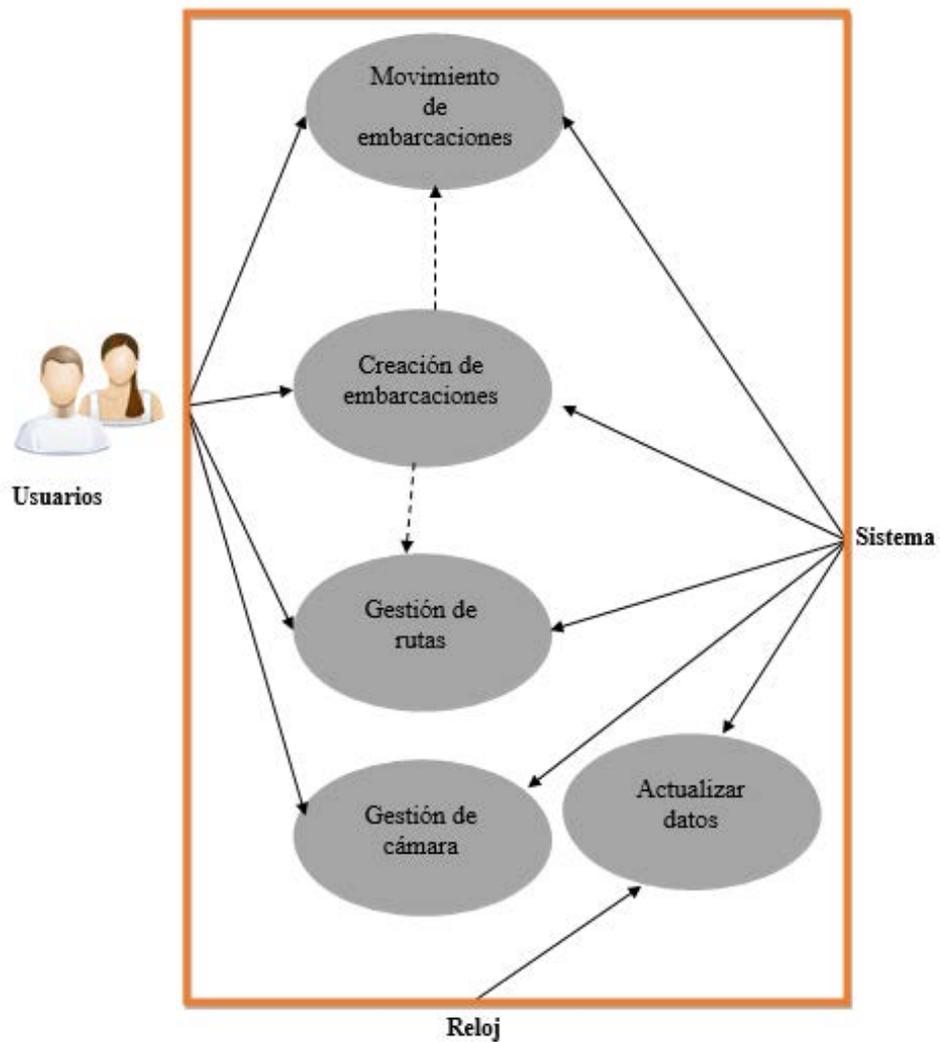


Ilustración 7: Diagrama de casos de uso

Fuente: Elaboración propia

3.5 Casos de uso

En este apartado describiremos los casos de uso del software que deseamos implementar.

La descripción de los casos de uso se realizará mediante unas tablas que seguirán el siguiente formato:

Modelo de tabla para los Casos de uso

Id del caso de uso	CU-X
Nombre del caso de uso	Nombre
Actores	Actores implicados
Precondición del caso de uso	Descripción de precondiciones
Postcondición del caso de uso	Descripción de postcondiciones
Descripción del caso de uso	Descripción

Tabla 41: modelo tabla CU

Fuente: Elaboración propia

En la tabla podemos observar que el identificador de los casos de uso ira incrementándose de uno en uno para la identificación de cada uno inequívocamente, cada caso de uso además contará con un nombre que haga referencia a su funcionalidad.

Se describirán los actores principales de cada caso de uso para que este se pueda llevar a cabo, así como también las precondiciones que deben cumplirse para que este pueda ser llevado a cabo y las postcondiciones, es decir las consecuencias que tendrá esta acción, finalmente una descripción de que sucede realmente en este caso de uso.

Id del caso de uso	CU-1
Nombre del caso de uso	Creación de embarcaciones
Actores	Usuario,
Precondición del caso de uso	El sistema debe encontrarse encendido y funcionando correctamente. El botón para crear las embarcaciones debe encontrarse activo y en caso de querer introducir un barco manual no debe haber ningún otro barco manual en la escena.
Postcondición del caso de uso	Aparecerá en la escena el/los barcos indicados y en caso de ser un barco manual ya no se encontrará disponible el botón de “1 manual”
Descripción del caso de uso	El usuario deberá apretar en el botón “1 manual” o “inicial” El sistema generará un único barco de control manual y eliminará el botón o generará los barcos indicados por script manteniendo el botón.

Tabla 42: CU-1

Id del caso de uso	CU-2
Nombre del caso de uso	Movimiento de embarcaciones
Actores	Usuario
Precondición del caso de uso	El sistema debe encontrarse encendido y funcionando correctamente. Deben encontrarse creadas las embarcaciones que se desean mover ya sean manual o automáticas, lo único que cambiará es quien lo maneja, ya sea el usuario o el piloto automático.
Postcondición del caso de uso	Las embarcaciones realizaran el movimiento de forma realista según las indicaciones del usuario o del piloto automático.
Descripción del caso de uso	El usuario deberá indicar con las teclas “i” y “k” si desea que el movimiento sea hacia adelante o hacia atrás y su potencia y las teclas “j” y “l” para indicar el giro del timón, y en caso de ser una embarcación que deba ser guiada por el piloto automático indicara la ruta que debe seguir en el fichero de creación de las embarcaciones.

Tabla 43: CU-2

Id del caso de uso	CU-3
Nombre del caso de uso	Gestión de rutas
Actores	Usuario
Precondición del caso de uso	El sistema debe encontrarse encendido y funcionando correctamente.
Postcondición del caso de uso	El sistema debe almacenar y mostrar las rutas de cada una de las embarcaciones con su color correspondiente
Descripción del caso de uso	<p>El usuario deberá haber generado el fichero adecuadamente para que el sistema reciba las rutas de cada embarcación y/o deberá indicar la ruta de la embarcación manual mediante el ratón.</p> <p>El sistema recibirá, almacenará y mostrará por pantalla la ruta de cada una de las embarcaciones con el color más adecuado para cada una</p>

Tabla 44: CU-3

Id del caso de uso	CU-4
Nombre del caso de uso	Actualizar datos
Actores	Reloj
Precondición del caso de uso	El sistema debe encontrarse encendido y funcionando correctamente con al menos una embarcación.
Postcondición del caso de uso	Los datos se habrán actualizado.
Descripción del caso de uso	El sistema actualizara cada <i>frame</i> los datos de la velocidad, latitud, longitud, rumbo y la ruta de cada embarcación, actualizando el color de las rutas según corresponda

Tabla 45: CU-4

Id del caso de uso	CU-5
Nombre del caso de uso	Gestión de cámara
Actores	Usuario
Precondición del caso de uso	El sistema debe encontrarse encendido y funcionando correctamente.
Postcondición del caso de uso	La cámara modificara tanto su posición como su dirección según lo indicado ya sea por teclado o por fichero
Descripción del caso de uso	<p>El usuario deberá usar las teclas “w” y “s” para hacer avanzar o retroceder la cámara respectivamente y las teclas “a” y “d” desplazarla a derecha o izquierda respectivamente para rotarla usará el ratón o bien indicará en un fichero la posición que debe tener y la dirección en la que debe mirar la cámara.</p> <p>El sistema modificará la posición y la dirección de la cámara según se lo indique el usuario por estos métodos.</p>

Tabla 46: CU-5

3.6 Matriz de trazabilidad

En esta sección representaremos la matriz de trazabilidad entre los requisitos y los casos de uso, para ello pondremos en las columnas a los 7 casos de uso y en las filas a los requisitos.

	CU-1	CU-2	CU-3	CU-4	CU-5
RF-F-1.0	X				
RF-F-1.1	X				
RF-F-1.2	X		X		
RF-F-1.3					X
RF-F-1.4					X
RF-F-2.0		X		X	
RF-F-3.0		X			
RF-F-3.1		X			
RF-F-4.0					X
RF-F-4.1					X
RF-F-5.0		X			
RF-F-6.0	X				
RF-F-7.0	X				
RF-F-7.1		X		X	
RF-F-7.2		X		X	
RF-F-7.3			X	X	

Tabla 47: Matriz de trazabilidad

Capítulo 4

Dificultades encontradas y resultados

4.1 Dificultades encontradas

Este proyecto ha resultado difícil de implementar debido al desconocimiento previo que se tenía tanto del funcionamiento como de la programación en Unity. Si bien es verdad que la curva de aprendizaje es bastante corta, puesto que el lenguaje para programar los scripts, el CSharp, es muy similar a java y C++ (lenguajes que ya se dominaban) por lo que ha resultado relativamente sencillo. Además, al ser Unity una herramienta muy intuitiva y visual, tampoco se han encontrado grandes dificultades a la hora de aprender a usarlo.

En cuanto al desarrollo del software para ser utilizado con gafas de realidad virtual no se ha podido realizar puesto que, para poder desarrollarlo y testarlo en la aplicación, se requería tener unas gafas conectadas. Al no disponerse de unas y al no tratarse de un elemento indispensable del software, sino de una característica adicional, se ha optado por no implementarlo.

Otro problema que se encontró fue hacerse a la idea de cómo estaba programada la parte ya implementada del software y entender como estaban relacionadas todas las partes de este.

Finalmente, la última de las dificultades encontradas, hace referencia a la elaboración de la memoria. Estas dificultades residen en que este tipo de trabajos de investigación no se realizan habitualmente y no se sabía muy bien cómo empezar la memoria ni que estructura debía seguirse en la misma. En especial el apartado del marco normativo, en el que no se ha puesto ninguna legislación aplicable, esto puede deberse a que no existe ninguna normativa que se le aplique a este tipo de software o por el contrario que esta no se haya sabido encontrar.

4.2 Resultados

En este proyecto se habían establecido como uno de los objetivos que se estudiase la situación actual del mundo marítimo. Este objetivo se ha cumplido y se ha podido observar que cada año la flota mundial que surca los mares asciende año tras año, por lo que, cada vez se necesita mayor control del tráfico marítimo, así como la incorporación de nuevas tecnologías.

También se ha marcado como meta el estudio de los controladores marítimos y las herramientas con las que cuentan. Esta meta ha sido alcanzada y se ha podido observar que en el mundo marítimo ya se cuentan con una buena regulación en cuanto al control del tránsito y cuentan con una gran cantidad de instrumentación para poder realizar este control, pero se puede mejorar y poner las cosas más fáciles para los controladores.

En cuanto al propósito de analizar otros simuladores navales, se han analizado no solo los navales sino también simuladores de todo tipo, pero este análisis no ha ayudado demasiado ya que estos simuladores tienen sus sistemas de modelado y diseño son confidenciales y por tanto no resultan de gran ayuda a la hora de acometer el nuevo proyecto.

En cuanto al objetivo de aportar una nueva herramienta, se ha realizado un estudio de viabilidad determinando los requisitos que debe cumplir la aplicación y se ha implementado, cumpliéndose así además la meta de tener un software que pueda servir como base para otros proyectos navales.

Este software implementa la mejora de la capacidad de observación de los controladores mediante la representación de las embarcaciones y dibujando una línea imaginaria con la trayectoria de cada uno, lo que ayuda a que estos puedan ver cuándo se puede sobrevenir una situación de peligro.

Adicionalmente, este software también cumple con las metas de abaratar los costes de adiestramiento de los controladores y los patrones de embarcaciones mediante sus simulaciones.

Por lo tanto, se puede afirmar que los objetivos propuestos para este proyecto se han visto completados en su totalidad, y se puede decir que existe un amplio mercado en el mundo marítimo ya que sigue expandiéndose día a día, y necesitando así de nuevas tecnologías continuamente.

Capítulo 5

Conclusiones y posibles trabajos futuros

En esta sección, expondremos las conclusiones finales del trabajo realizado y, además, expondremos cuales podrían ser los trabajos futuros que podrían realizarse tomando como base este proyecto.

5.1 Conclusiones

Inicialmente se ha realizado un análisis de la situación actual del mundo naval y la cantidad de embarcaciones que surcan las aguas, de este análisis se extrae la conclusión de que año tras año, aun habiendo épocas de crisis parece que el número de embarcaciones aumenta, pareciendo no verse afectado por las crisis y seguir su avance impasible.

Debido a este aumento del número de embarcaciones se necesitan mejoras cada cierto tiempo en el control del tránsito de estas embarcaciones, en especial de las zonas de alta densidad de tráfico como los puertos.

Por ello, se ha detallado un software que permite mejorar el control de este incipiente tráfico marítimo, ayudando a los controladores navales a mejorar su capacidad de vislumbrar la situación actual en la que se encuentran todas las embarcaciones y su ruta.

Además, con este software se reducirá el coste tanto del adiestramiento de los controladores como el de los patrones de las embarcaciones, así como se mejorará su conocimiento de las situaciones más adversas.

Por lo tanto, el software que aquí se detalla se pretende que llegar a suponer una de esas mejoras que se incluyen cada cierto tiempo. Aunque sabemos que esta tarea es complicada debido, a que como ya hemos visto, ya se cuenta con muchos sistemas sofisticados que ayudan a los controladores marítimos a desempeñar su función debidamente.

5.2 Trabajos futuros

El software implementado es bastante básico y podrían realizarse muchas mejoras tanto visuales como funcionales, así como también perfeccionamientos en cuanto a la usabilidad, las más destacadas son:

La inclusión de la simulación de las inclemencias del tiempo en el que las embarcaciones tienen problemas para controlar tanto la velocidad como la dirección de los navíos. Esto incluiría tanto la variabilidad del tamaño de las olas como el de su frecuencia, incluso podría añadirse una opción para modificar la fuerza y la dirección del viento.

También podría desarrollarse un algoritmo para que algunas embarcaciones simulasen estar averiadas y fuesen a la deriva, dependiendo tanto de las corrientes marinas, el oleaje y el viento.

Adicionalmente, se podría modificar la profundidad del lecho marino para poder simular el real.

En cuanto a las representaciones de las embarcaciones podría ampliarse los diferentes tipos de embarcaciones que pudiesen representarse, no solo el que disponemos actualmente, y con ello, también implementar nuevos algoritmos de movimiento de estas nuevas embarcaciones, proporcionando así un repertorio más

amplio de embarcaciones y todo lo que afecta y de este modo facilitar la identificación y el reconocimiento de estas entre la simulación y la realidad.

Finalmente, se podría mejorar el sistema de colisiones entre los barcos, permitiendo incluso determinar los daños que sufriría cada embarcación, teniendo en consideración para ello la velocidad, la dirección, el tamaño, el peso y el ángulo de los barcos que colisionasen, mostrando los daños que sufriese cada uno de ellos, llegando incluso a poder hundirse alguno de los buques.

Capítulo 6

Planificación y presupuesto

En esta sección se mostrará cual ha sido la planificación que se ha seguido a la hora de desarrollar este proyecto mediante un diagrama de Gantt. También se mostrará el presupuesto que se estima para el desarrollo de este proyecto completo.

En la ilustración 7 podemos observar cómo ha sido la planificación para desarrollar este proyecto, se trata de un diagrama de Gantt en el que se especifica la duración estimada de cada una de las tareas a realizar.

En el primer punto se observa la propuesta del proyecto por parte de Antonio Berlanga, y la posterior aceptación de este. Nada más realizarse la propuesta se comienza a realizar un análisis de la propuesta, mediante este análisis se comprueba que era una buena propuesta y se acepta dicha propuesta. Posteriormente se empieza a investigar más profundamente sobre el tema y sobre las posibles soluciones.

Podemos observar como después de esta búsqueda de información se realiza el diseño de la solución mediante la definición del problema y la definición de los requisitos.

Una vez se tiene el diseño de la solución se empieza a implementar dicha solución, y a medida que se va implementando se realizan periódicamente revisiones de todos los puntos del proceso escalonadamente para que la solución sea lo más completa y correcta. Corrigiendo y comprobando si la implementación es adecuada y es consecuente con el diseño previo.

Paralelamente a este trabajo se va realizando la memoria del trabajo que se va desarrollando.

Finalmente, se realizarán al final del proceso las pruebas y la evaluación del trabajo final, comprobándose así que todo está implementado adecuadamente, al superarse esta actividad se procede a la entrega del proyecto final a la espera de la confirmación de la fecha de presentación del mismo.

6.2 Presupuesto

El presupuesto de este proyecto se estima para que la realización del proyecto se realice en 4 meses por 2 trabajadores trabajando 6 h diarias.

6.2.1 Entorno de trabajo

El entorno de trabajo necesario para poder desarrollar adecuadamente el proyecto debe constar como mínimo de los siguientes elementos:

- Ordenadores Portátiles para cada trabajador, cada uno de estos debe disponer de las características siguientes:
 - Microsoft office
 - Windows 7 o superior
 - Conexión a internet
 - Unity (Licencia)
 - Navegador (Chrome)
- Material de oficina

6.2.2 Coste de licencias

Las licencias de las que se debe disponer son la del sistema operativo Windows 7 o superior, la licencia del Office para empresas y finalmente la licencia de Unity. El coste total de todas estas licencias quedaría de la siguiente manera:

Precio licencias

Nombre del software	Precio Licencia completa	Precio licencia mensual
Windows 10	539,00€	-
Office	211,20€=8,80€/mes*usuario(anual)	211,20=8,80€/mes*usuario(anual)
Unity	1320,00€	67,00€
Total	2070,20€	1018,20€

Tabla 48:precio licencias
Fuente: Elaboración propia

6.2.3 Mano de obra

Como ya se ha dicho antes para el desarrollo del software se utilizarán a 2 empleados durante 4 meses 6 horas diarias 5 días a la semana lo que hacen un total de

480 horas (4 meses * 4 semanas * 5 días * 6 horas) cada empleado para la realización del proyecto completo.

Coste de los trabajadores

Trabajador	Salario por hora	Salario mensual	Salario total
Jefe de proyecto	18,00€/h	2160,00€/mes	8640,00€
Programador	13,00€/h	1560,00€/mes	6240,00€
Total	31,00€/h	3720,00€/mes	14880,00€

Tabla 49: Coste de la mano de obra

Fuente: Elaboración propia

6.2.4 Coste del hardware

Los medios hardware que se necesitarán serán:

- Un ordenador portátil para cada empleado, en total dos ordenadores portátiles.
- Un teléfono inteligente para cada empleado, en total dos teléfonos inteligentes.

El coste total del hardware aquí descrito ascenderá a:

Coste hardware

Dispositivo	Precio individual	Precio total
Portátil: Acer Aspire E5-573G-57EZ - NX.MVPEB.004 ¹⁰	545,00€	1090,00€
Teléfono inteligente: Bq Aquaris E4 ¹¹	120,00€	240,00€
Total		1330,00€

Tabla 50: Coste hardware

Fuente: Elaboración propia

6.2.5 Coste del total

Aquí obtenemos el coste total de cada una de las partes sin prorrateos.

Coste Total

Elemento	Coste total
Salarios	14880,00€
Licencias	1018,20€
Hardware	1330,00€
Total	17228,20€/h

Tabla 51: Coste Total

Fuente: Elaboración propia

6.2.6 Coste con amortizaciones

En esta sección se analiza que elementos pueden ser amortizados y se amortizan. En este caso las licencias tanto de Windows como de Office se amortizan a 5 años. El hardware en cambio se prorratea a distintos valores, los portátiles que tienen una vida

¹⁰aceronline, «aceronline.es,» acer, [En línea]. Available: <http://www.aceronline.es/portatil-ultrabook-acer-aspire-e5-573g-57ez-nx-mvpeb-004.html>.

¹¹ BQ, «bq.com/,» BQ, [En línea]. Available: <http://www.bq.com/es/aquaris-e4>.

útil más elevada se les imputa una vida útil de 4 años y en cambio los teléfonos móviles al tener una vida útil inferior se les imputan únicamente 3 años.

Amortizaciones

Elemento	Periodo de amortización	Coste total periodo
Licencia Windows 10	60	36,00€
Licencias Office	12	70,40€
Portátiles	48	90,90€
Teléfonos inteligentes	36	26,70€
Total		224,00€

Tabla 52: amortizaciones

Fuente: Elaboración propia

Una vez obtenido los valores amortizados procederemos a calcular el valor real de este proyecto durante los 4 meses que dura.

Coste Total Amortizado

Elemento	Coste total
Salarios	14880,00€
Licencias	374,40€
Hardware	117,60€
Total	15372,00€/h

Tabla 53: coste total amortizado

Fuente: Elaboración propia

Referencias

aceronline, s.f. *aceronline.es*. [En línea]

Available at: <http://www.aceronline.es/>

Berzal Galiano, F., s.f. *Universidad de granada*. [En línea]

Available at: <http://elvex.ugr.es/idbis/db/docs/design/2-requirements.pdf>

BQ, s.f. *bq.com*/. [En línea]

Available at: <http://www.bq.com/>

Colomo, R., 2013. *OCW - UC3M*. [En línea]

Available at: http://ocw.uc3m.es/ingenieria-informatica/ingeniera-del-software-iii/materialclase/ISIII_07_ASI.pdf.

Colomo, R., 2013. *OCW - UC3M*. [En línea]

Available at: http://ocw.uc3m.es/ingenieria-informatica/ingeniera-del-software-iii/materialclase/ISIII_06_EVS.pdf

CryEngine, 2015. *cryengine.com*. [En línea]

Available at: <https://www.cryengine.com/>

<http://www.practicosdepuerto.es/el-colegio>, 2016. <http://www.practicosdepuerto.es/el-colegio>. [En línea]

Available at: <http://www.practicosdepuerto.es/el-colegio>

<http://www.practicosdepuerto.es>, s.f. <http://www.practicosdepuerto.es>. [En línea]

Available at: <http://www.practicosdepuerto.es/colegio-federacion/cualificacion> [Último acceso: 25 03 2016].

IALA (International Association of Marine Aids to Navigation and Lighthouse Authorities), IAPH (International Association of Ports and Harbors) and IMPA (International Maritime Pilots' Association) IFSMA (International Federation of Shipmasters' Associatio, s.f. *worldvtsguide.org*. [En línea]

Available at: <http://www.worldvtsguide.org/>

IALA, 2013. *Recomendaciones IALA V-103*. Sant-Germain-en-Laye: s.n.

Microsoft, s.f. *microsoftstore.com*. [En línea]

Available at: <http://www.microsoftstore.com/>

Office, s.f. <https://products.office.com>. [En línea]

Available at: <https://products.office.com>

ogre3d, s.f. *ogre3d.org*. [En línea]
Available at: <http://www.ogre3d.org>

OMI, 1997. *Resolución 857 Asamblea 20 OMI*. s.l.:s.n.

panda3d, 2015. *panda3d.org*. [En línea]
Available at: <https://www.panda3d.org/>

RAE , s.f. *http://www.rae.es/*. [En línea]
Available at: <http://dle.rae.es/?id=Xw14yph>
[Último acceso: 27 03 2016].

Sanchez Segura, M.-I. & Mora-Soto, A., 2012. *ocw.uc3m.es*. [En línea]
Available at: <http://ocw.uc3m.es/ingenieria-informatica/metodologia-de-desarrollo-visual/course-files>.

Unity, 2015. *unity3d.com/es*. [En línea]
Available at: <https://unity3d.com/es>

Unreal Engine, 2016. *unrealengine.com*. [En línea]
Available at: <https://www.unrealengine.com/>

WordReference.com, s.f. *http://www.wordreference.com*. [En línea]
Available at: <http://www.wordreference.com/definicion/simulador>
[Último acceso: 27 03 2016].

Glosario

RAE	<i>Real Academia Española</i>
VTS	<i>Vessel Traffic Services</i>
AIS	<i>Automatic Identification Service</i>
RDF	<i>Radio Direction Finder</i>
VHF	<i>Very High Frequency</i>
SOLAS	<i>Safety of Life At Sea (seguridad de la vida humana en el mar)</i>
OMI	<i>Organización Marítima Internacional</i>
IMO	<i>International Maritime Organization</i>
IALA	<i>International Association of Marine Aids to Navigation and Lighthouse Authorities.</i>

Anexos

Anexo I: English translation

In that appendant we will show five specific parts of the project translated from Spanish into English, we will do it to show our knowledge of that language. The sections to be translated are:

- Abstract
- Introduction
- Objectives
- Results
- Conclusions

Abstract

The current document performs the pertinent study for the realization of a software that allows to increase the security in the sea and helps especially the naval controllers to perform their duties.

For the realization of that software have had to determine the objectives which the software must fulfill, because of that it has carried out an analysis of the current situation of the maritime world to see which are their shortcomings, gaps and needs were.

Once we have seen which are that necessities we made another analysis, that one about the solutions that other organisms have carried out, if there is any solution, and also we analyze which is going to be the platform that we are going to use to realize the implementation of the solution.

Then, once we have determined both the necessities of the software and the software that we will use, we use the capabilities of the chosen software and the solutions that we have seen in another similar projects and we define the characteristics that this software must meet.

According to these characteristics the software will be implemented to become as faithful as possible to the objective that we want to achieve, which is no other than facilitate as much as possible the life at sea.

Finally, we will detail possible improvements that could be made to this software in the future and using it as a base, but also could be started a new project for everyone of

this functions, if it's anticipated that the chosen platform could overcome an extra difficulty for the implementation

Keywords: Simulator, graphics engine, software, naval controller and harbor.

8.1 Introduction

In this part we will introduce what the main character of the project is, we will describe the current naval necessities and the necessities of the maritime control near the harbors. Additionally, we will describe the software's needs. Finally, we'll explore the motivations behind the creation of this particular Project.

8.1.1 Motivation

The principal idea is to build a complete and flexible simulation system for naval trajectories, which should allow various applications. The principal functions for which the software has been created are:

1. Being a base to test the following algorithms:
 - 1.1. Algorithms for collisions between ships
 - 1.2. Algorithms that simulate weather conditions
 - 1.3. Algorithms for boat movements
2. Test hypothetical and realistic scenarios and observe their consequences.

This could serve as a test for the different naval titles and check whether or not the candidates meet the technical capabilities that are required to sail the vessels.

Additionally, it could also serve to achieve the tests to the harbor pilots, which currently are implemented in paper, that easily could increase the level of efficiency and truthfulness of that tests.

3. Assist naval controllers in their functions and facilitate their work.

Nevertheless, these aren't the only functionalities, so the software could be used for other functions, the ones explored here are the main ones. It is also true that the system functionality may change as new applications are developed for which the software is shown to be useful. Even with the new technological improvements that appear, like virtual reality glasses, the software could find new functionalities and applications.

Currently, the number of ships that are sailing the seas is raising considerably year after year, as we can see in the table 1. This makes it clear that more efficient and more advanced technologies to control the boats are always needed, especially in the areas with high rates of traffic, like the harbors and their surroundings.

Flota mercante									
YEAR		2008	2009	2010	2011	2012	2013	2014	2015
World	Total fleet	1117779	1192317	1276137	1415110	1532114	1625750	1691628	1749222
	Oil tankers	407881	418266	450053	439932	454349	472890	482017	489388
	Bulk carriers	391127	418356	456623	547192	624022	686635	726319	760468
	General carg	105492	108881	108232	81159	78138	77589	77552	76731
	Container sh	144655	161919	169158	183691	196821	206547	216345	227741
	Other types	68624	84895	92072	163135	178784	182090	189395	194893

fuente:UNCTAD

Table 1: world merchant fleet

Source: UNCTAD

Additionally, the necessity to simulate various situations is increasing, both to test possible scenarios, and to evaluate other solutions that could have been applied in past situations.

Thanks to such simulations, the maritime controllers will be able to demonstrate if they have the capabilities to perform their duty as controllers. This could be done by testing them in realistic circumstances, and hypothetically the candidates should demonstrate their proficiency in a normal situation, or even in some dangerous situations.

Moreover, these simulations may be used to make sure that the boat master, even the military or civilians are able to understand and handle a boat following the orders that the controllers send to them and avoid any kind of dangerous situation.

The naval simulators are able to enormously reduce all the costs that incur in any activity related to maritime control, through the simulations aforementioned of piloting vessels or traffic regulation. That cost reduction is mainly because, even though the trainees make a big mistake, in the simulation, it doesn't incur any expenses or imperfection, also it prevents any danger against the integrity of the people.

Likewise, studies could be performed to find the best routes to follow the ships to make the boats consume less resources in the journey, as could be the fuel, or the time thus saved.

8.1.2 Setting

Nowadays the naval controllers are a small group and they are highly qualified, because they follow the International Association of Marine Aids to Navigation and Lighthouse Authorities (hereafter, IALA) Recommendations V-103¹² and the 20 International Maritime Organization (hereafter, IMO) Assembly Resolution 857¹³, that naval controllers are assisted by the harbor pilots, in Spain there are 241 harbor pilots that support them spread in the 56 ports that count with that service¹⁴, the harbor pilots have a great preparation and experience inasmuch as IALA recommendations establish. Some of the support systems that are at the disposal of the harbor pilots currently are the Vessel Traffic Services (hereafter, VTS), which are systems that are at the disposal of the naval controllers too.

The naval controllers provide the boats with relevant information like the position of other close ships, weather information, etc. Furthermore, the naval controllers take charge of handling the maritime traffic, giving fluidity to the traffic, and avoiding risky situations.

When a vessel enters in the area of a VTS, the vessel has to inform the authorities of that entrance via radio. Then the authorities have the task of monitoring the vessel using their instruments.

The duties of a naval controller using the VTS are:

1. Communicate, with enough time, to the boats about any kind of information that could be relevant about the area, it could be the state of the traffic, to make decision-making possible for the commander of the vessels assisted by the naval controller, in that way avoiding any kind dangerous situation.
2. Decision-making: the naval controllers should help in the decision-making of the vessels and track their consequences in all of them, that contributions depends on the legislation of every country.
3. Organization: the naval controllers should help to have a fluid and safe movement, stimulating the traffic flow in the areas with a high concentration of ships.

¹² IALA, Recomendaciones IALA V-103, Sant-Germain-en-Laye, 2013.

¹³ OMI, Resolución 857 Asamblea 20 OMI, 1997.

¹⁴ <http://www.practicosdepuerto.es/el-colegio>, «<http://www.practicosdepuerto.es/el-colegio>,» 2016. [En línea]. Available: <http://www.practicosdepuerto.es/el-colegio>.

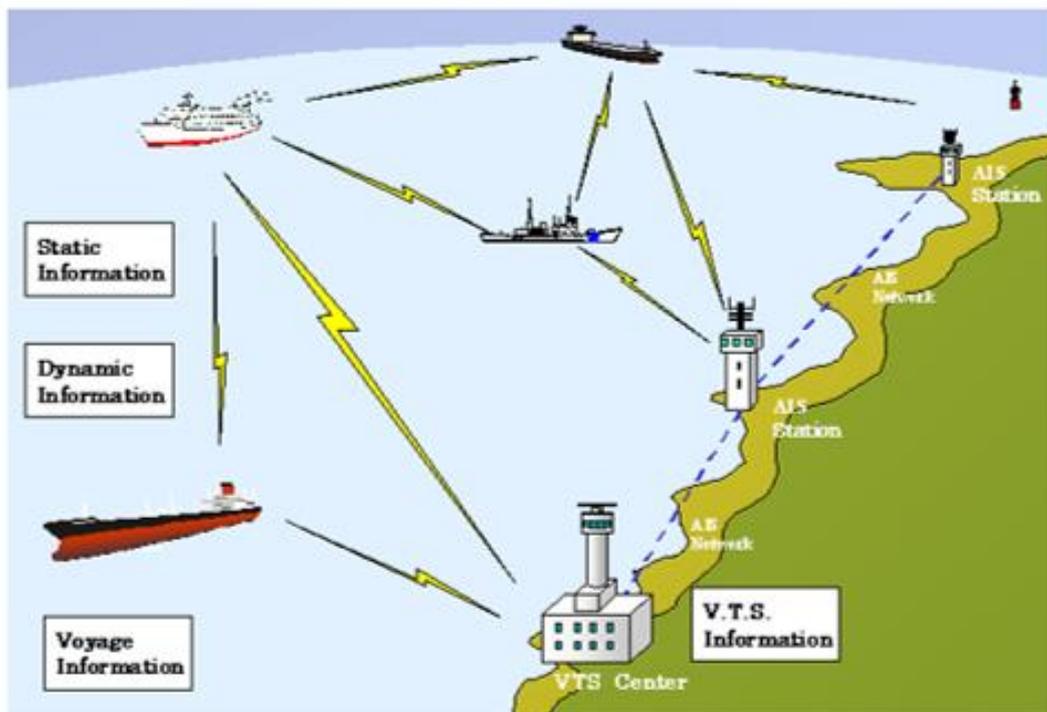
4. Data analysis: the last task that the controllers must do is to coordinate the gathering of the data, evaluate it and distribute it properly to make it useful and relevant.

To develop these services, the controller could count on, depending on the kind of port, the following instruments:

1. Automatic Identification System (AIS)
2. Radar
3. Radio Direction Finder (RDF)
4. Real-Time Tracking
5. VHF Radio Position Reporting

The use of that instruments is regulated by the International Convention of Safety of Life at Sea (hereafter, SOLAS) dictated by the IMO. This convention specifies which kind of vessels must have all of those systems on board, and they must be operative, and so on it determines which coastal areas should be equipped with control systems for that systems.

Ejemplo de AIS y VTS



Fuente: http://www6.kaiho.mlit.go.jp/kanmon/eng/mg_2.htm

Figure 1: AIS and VTS

Source: kaiho.mlit.go

8.2 Objectives

In this document we will study what the current situation in the maritime world is, how are current the naval simulators and what are the aids that they have. Likewise, other naval simulators will be studied to guide us in the completion of the project, and see some possible solutions for our project.

This project wants to offer a new tool to the maritime world, with particular emphasis on the usefulness that it could mean for maritime controllers, allowing both improvement in their observations, and their ability to predict, with greater reliability, complex traffic situations with the vessels.

In this project we won't focus solely on those utilities, but it is intended to also serve as the basis of any future work in the field of naval simulation.

Therefore, the objectives of this project are:

- The analysis of how the capability of observation of the naval controllers could be improved.
- Develop aids with software to make the controllers detect dangerous situations as soon as possible.
- Assist the formation of the naval controllers in an economic and safe way
- Train the vessel controllers to face any adverse situations.
- Develop and implement a software that will serve as a basis for future similar programs.

8.3 Results

In this project it has been established as one of the main objectives to study the current situation of the maritime world. That objective has been achieved and it could be observed that every year the world fleet is increasing year by year, so, every time it needs more control of the maritime traffic, and the incorporation of new technologies.

It has also been stated that the objective of the study of the maritime controllers and the tools that they have. That goal has been achieved and it has been observed that the maritime world already has a good level of regulation regarding the control of traffic and they have a lot of instruments to be able to perform that control, but it can be improved upon and things can be made easier for the controllers.

Regarding the propose of analyzing another naval simulators have been analyzed not just naval also simulators of all kinds, but this analysis has not helped a lot because these simulators have their systems of modeling and design are confidential and therefore they didn't become a great resource for undertaking that new project.

Regarding the objective of providing a new tool, it has conducted a study of viability determining the requirements that the applications should meet and it has been implemented, fulfilling also the goal of having a software that could serve as a basis for other naval projects.

This software implements the improvement of the visual capability of the controllers by representing the crafts and drawing an imaginary line representing the trajectory of each line, which helps them to see when a dangerous situation may occur.

Additionally, this software also meets the goals of reducing the costs of training the drivers of the vessels and boaters through their simulations.

Therefore, it can be stated that the objectives for this project have been completed fully, and it can be said that there is a large market in the maritime world as it continues to expand every day, and so that it needs new technology to be applied.

8.4 Conclusions and future works

In this section, we will discuss the final conclusions of the work done and we will also discuss what could be the future work that could be done based on the work carried out in this project.

8.4.1 Conclusions

Initially an analysis of the current situation of world shipbuilding and the number of vessels plying the waters has been carried out, of this analysis the conclusion is drawn that year after year, even during times of crisis it seems that the number of vessels increases, seemingly unaffected by the crisis and continues this impassive progress.

Because of this increase in the number of vessels improvements are needed from time to time in the traffic control of these vessels, especially in areas with high traffic density such as ports.

Because of this, a software that allows a better control of this increasing maritime traffic has been detailed, helping naval drivers to improve their ability to discern the actual situation of all the vessels and their routes.

In addition, this software, here detailed, will reduce the cost of training both drivers and patterns of vessels, and they will improve the knowledge of the most adverse situations.

Therefore, the software detailed herein is intended to assume one of these improvements that are included periodically. Although we know that this task is complicated, as we have seen, it already has many sophisticated systems that help the drivers to develop their maritime function in a proper way.

8.4.2 Future Works

The software implemented is quite basic and many visual and functional improvements could be made, as well as improvements in terms of usability, the most important are:

The inclusion of the simulation of inclement weather in which the vessels have troubles controlling both the speed and the direction of the ships. This would include both the variability of the size of the waves as their frequency, and it might even be an option to modify the strength and the direction of the wind.

It also could develop an algorithm to simulate some boats that are damaged and simulate drifting depending on sea currents, waves and wind.

In addition, it could change the depth of the seabed to simulate the real one.

The representations of the vessels could be expanded with different types of boats that may be represented, not only the one that is currently available, and thereby, also implement new algorithms of movement for these new vessels, thereby providing a wider boat repertory and everything that affects it and thereby facilitate the identification and recognition of these between the simulation and the reality.

Finally, it could improve the collision system between ships, even allowing it to determine damages that each vessel will suffer, taking into consideration the speed, the direction, the size, the weight and angle of every vessel involved in the collision, showing the damage suffered by the boats, and even being able to sink some ships.

Anexo II: Software implementado y manejo

Se ha generado un ejecutable, “PFC_Jaime_Vázquez”, para poder ejecutar la aplicación, una vez lo ejecutamos nos encontramos esta pantalla:

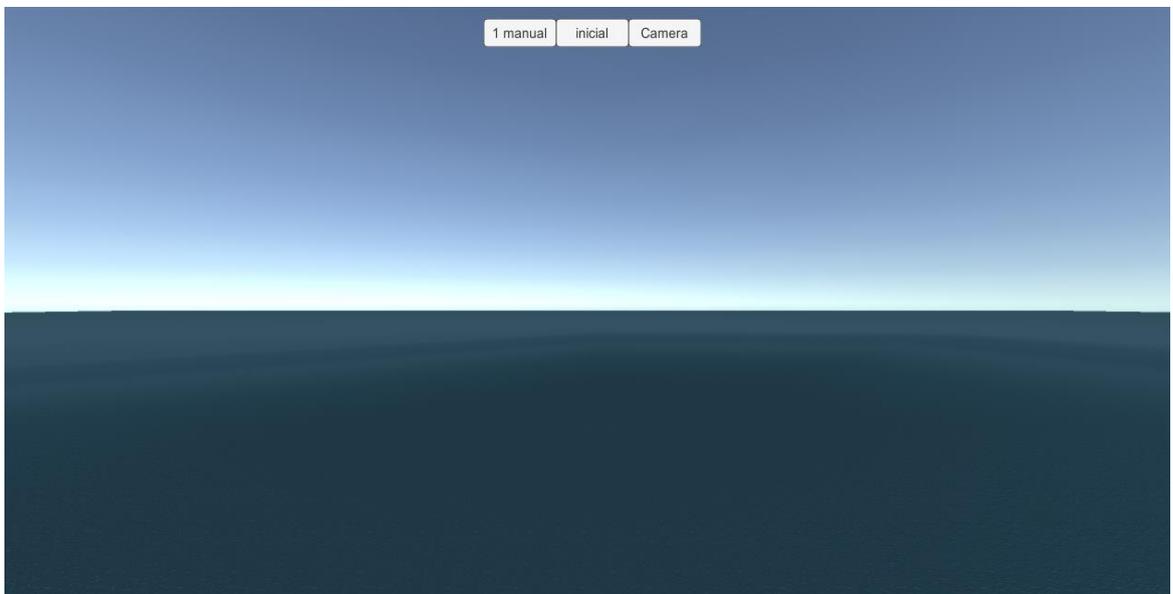


Ilustración 9: Captura de pantalla inicial del software

Fuente: Captura de pantalla

En esta pantalla disponemos de 3 botones, cada uno con su funcionalidad, el primero de ellos (1 manual) sirve para generar un barco de control manual, el segundo botón permite generar barcos con un piloto automático que les hace seguir una ruta, todo esto determinado en un fichero. Y finalmente un último botón que nos permite mover la cámara a una posición y en un ángulo especificado en un fichero.



Ilustración 10: captura de pantalla de barco manual

Fuente: Captura de pantalla

En la ilustración 9 podemos observar cómo se genera el barco de control manual y como desaparece el botón de generar otro para que únicamente pueda existir un único barco manual como máximo en cada ejecución.

Los controles para la embarcación manual son las teclas:

- “i” para acelerar
- “k” para reducir la potencia o ir marcha atrás
- “j” para girar el timón a babor
- “l” para girar el timón a estribor

Podemos observar que la embarcación cuenta con un rotulo informativo con sus coordenadas, su velocidad y su rumbo actual.



Ilustración 11: captura de pantalla de los barcos con piloto automático

Fuente: Captura de pantalla

En la ilustración 10 podemos observar que al apretar el botón inicial se generar 3 embarcaciones con sus respectivas rutas, la tercera embarcación se encuentra oculta a la derecha, pero podemos observar su ruta de color verde. Las líneas rojas indican peligro ya que hay dos rutas que se interseccionan y quiere decir que 2 embarcaciones pasaran por el mismo punto, la verde indica que no hay ninguna embarcación que vaya a pasar por ninguno de los puntos de la ruta seleccionada.

Podemos apreciar que el color del cuadro informativo tiene un color distinto al resto de embarcaciones para la embarcación manual, esto es para facilitar su reconocimiento.

En la imagen 11 podemos observar también que le podemos asignar una ruta a la embarcación manual mediante la combinación de teclas “*shift*” y “*r*” y haciendo clic derecho sobre la ubicación a la que queremos dirigirnos, podemos observar que el color de esta ruta es diferente a las demás para poder distinguirla de las de las embarcaciones dirigidas por piloto automático.



Ilustración 12: Captura de pantalla de ruta de embarcación manual

Fuente: Captura de pantalla

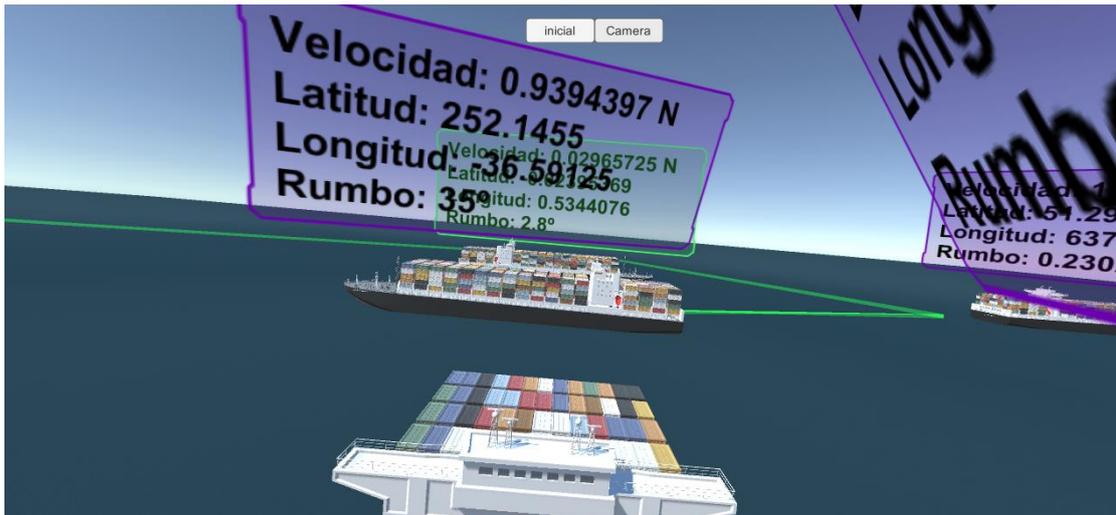


Ilustración 13: Captura de pantalla del cambio de la cámara

Fuente: Captura de pantalla

En la ilustración 12 podemos observar el cambio realizado por la cámara mediante el botón de camera y el fichero, podemos ver que la cámara ha rotado y se ha desplazado, y adicionalmente podemos observar que las rutas de los cargueros que antes estaban en rojo ahora se encuentran de color verde al no cruzarse la una con la otra.

Finalmente, los ficheros deberán tener la siguiente forma:

```

1 boat 590.0 1.0 90
2 590.0 500.5
3 590.0 1000.4
4 800.0 50.0
5 boat 52.0 422.0 0
6 50.2 800.1
7 333.2 1000.2
8 210.0 1200.0
9 boat 252.0 -22.0 180
10 60.2 452.1
11 -500.2 100.2
12 0.0 -1220.0
13

```

Ilustración 14: captura de pantalla de fichero de barcos y rutas

Fuente: Captura de pantalla

En la ilustración 13 podemos observar que el fichero indica con la palabra *boat* la creación de una embarcación, seguida por dos posiciones que determinaran su posición

en la simulación y finalmente unos grados, que indicaran cuanto debe estar girada esta embarcación.

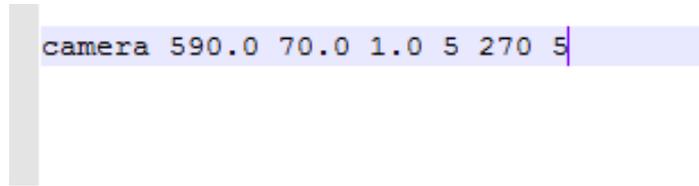


Ilustración 15: Captura de pantalla de fichero de modificación de cámara

Fuente: Captura de pantalla

En la ilustración 14 observamos que el fichero de modificación de la cámara debe indicar primeramente la cámara, posteriormente indica sus 3 coordenadas determinando su posición y finalmente sus 3 ángulos de giro para determinar cuál es la dirección en la que apunta esta.

Para la inclusión de otro tipo de embarcaciones se recomienda importar un *prefab* con esta nueva embarcación que se desea incluir en la simulación y modificar el script de inicial.cs e incluir que acepte otra palabra en el fichero en el que indicamos el tipo de embarcación su posición su ángulo y su ruta, adicionalmente se debería entrar en el formato desarrollador e indicado en el botón inicial que embarcaciones son las que debe usar como se muestra en la imagen inferior deberíamos introducir el *prefab* del nuevo barco en el campo donde pone Other.

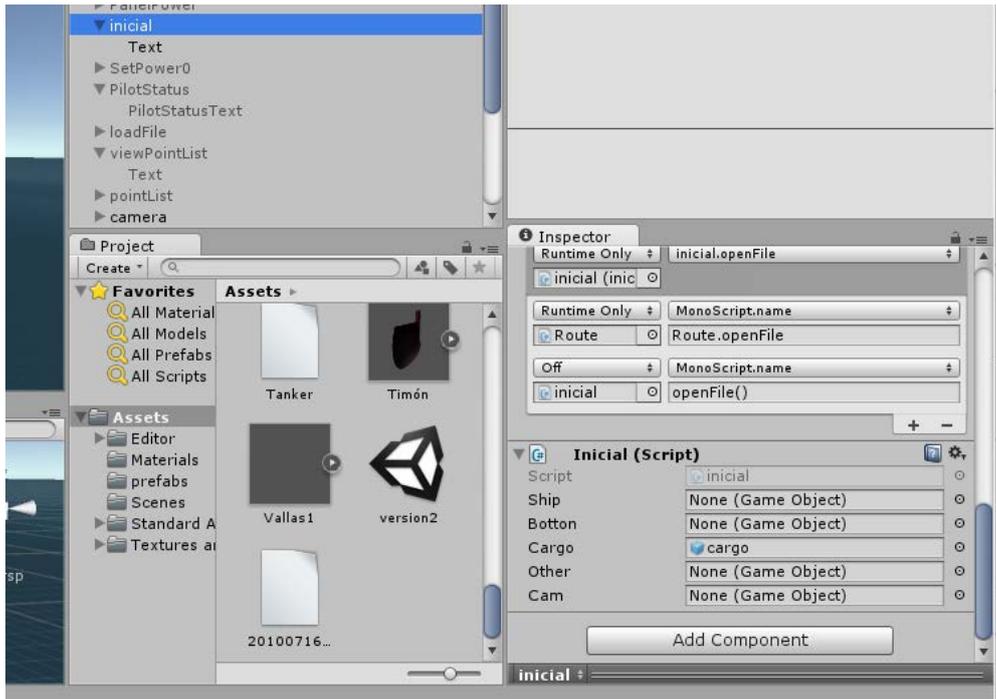


Ilustración 16: Inserción de nuevas embarcaciones

Fuente: Captura de pantalla

Una vez hecho esto el sistema ya podría incluir todas las embarcaciones deseadas incluyendo otros *prefabs* y modificando el fichero de introducción de embarcaciones.

También se podría realizar esta función únicamente modificando el código de *inicial.cs* y ahí especificar la ruta del objeto que deseamos introducir en la simulación, pero personalmente se recomienda el otro método debido a que es más sencillo y visual.