



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

K
I
S
A

I
C
S
I

University Master's Degree
**Computational Engineering and Intelligent
Systems**

Konputazio Zientziak eta Adimen Artifiziala Saila –
Departamento de Ciencias de la Computación e Inteligencia Artificial

Master's Thesis

Simulador de remo en banco fijo: Trainera

Etxebarria Iñarra, Ugaitz

Tutor

Makazaga Odria, Joseba

Konputazio Zientziak eta Adimen Artifiziala saila
Informatika Fakultatea



KZAA
/CCIA

September 2014

Índice general

1. Introducción	5
1.1. Resumen	5
1.2. Estado del arte	5
1.3. Estudios previos necesarios para el proyecto	6
2. Historia	7
2.1. Navío de pesca	7
2.2. Las primeras competiciones	8
2.3. Normativa	9
2.4. Competiciones	11
3. Diseño	13
3.1. El simulador	13
3.2. Elección de librerías	13
3.2.1. Candidatos para el motor físico	14
3.2.1.1. NVIDIA PhysX Physics	14
3.2.1.2. Bullet	14
3.2.1.3. Newton Game Dynamics	14
3.2.2. Candidatos para el motor gráfico	15
3.2.2.1. Ogre 3D	15
3.2.2.2. Unreal Development Kit	15
3.2.2.3. Unity3D	16
3.3. Elección de motores	16
4. Implementación	19
4.1. Creación de modelos en 3D	19
4.2. Física	25
4.2.1. Fuerza ejercida por el remero	25
4.2.1.1. Fuerza	25
4.2.2. Cantidad de rotación del remo	28
4.2.2.1. Momento de fuerza	29
4.2.2.2. Momento de Inercia	31
4.2.2.3. Velocidad y Aceleración Angular	33
4.2.2.4. Segunda Ley de Newton para la rotación	34

4.2.3.	Dirección de rotación del remo	35
4.2.4.	Drag	36
4.2.4.1.	Densidad	37
4.2.4.2.	Velocidad	37
4.2.4.3.	Área	38
4.3.	Fases de la palada	39
4.3.1.	Duración de la palada	39
4.3.1.1.	Frecuencia y periodo	40
4.3.2.	Velocidad de la fase de vuelta	41
4.4.	Ataque y final	42
4.5.	Movimiento de la trainera	43
4.5.1.	Palanca	43
4.6.	Cronómetro	44
5.	Uso de datos	45
5.1.	Datos despreciados en la física	45
5.2.	Datos utilizados como entrada	46
6.	Trabajo futuro	49
A.	Notas de programación	53
A.1.	Jerarquía de los objetos	53
A.2.	Códigos	55
A.3.	Flujograma	57
A.4.	Cambios de variable	58
B.	Resultado	59
C.	Terminología	61
D.	Codificación en Unity3D	63

Índice de figuras

2.1. Maquetas de traineras impulsadas por velas expuesta en el aquarium de San Sebastián.	8
2.2. Maqueta de una trainera de competición de 1920 expuesta en el aquarium de San Sebastián.	9
2.3. Maqueta de una trainera de competición de 1980 expuesta en el aquarium de San Sebastián.	11
2.4. Maqueta de una trainera de competición de 2010 expuesta en el aquarium de San Sebastián.	12
4.1. Casco en 3D.	19
4.2. Cuadernas en 3D.	20
4.3. Tostas en 3D.	20
4.4. Cremalleras en 3D.	21
4.5. Cuerno en 3D.	21
4.6. Apoyos del patrón en 3D.	22
4.7. Toleteras en 3D.	22
4.8. Toletes en 3D.	23
4.9. Pala del remo en 3D.	23
4.10. Tope del remo en 3D.	24
4.11. Caña del remo en 3D.	24
4.12. Remo de proa en 3D.	24
4.13. Remo de patrón en 3D.	24
4.14. Posición de los brazos y dirección de la fuerza en el ataque.	26
4.15. Posición de los brazos y dirección de la fuerza en el final.	26
4.16. Curva de fuerza vista en la parte inferior de la pantalla de un ergómetro.	27
4.17. Un ergómetro o remoergómetro.	27
4.18. Remos anclados en el tolete con el estrobo a la altura del tope.	28
4.19. Obtención del vector perpendicular al radio para el cálculo del momento de fuerza.	30
4.20. Representación gráfica de la regla de la mano derecha o sacacorchos.	31
4.21. Patinador girando sobre su propio eje con diferentes inercias y sus consiguientes velocidades.	32
4.22. Vistas del recorrido del remo siendo el eje estático.	35

4.23. Vistas del recorrido siendo el eje perfecto.	36
4.24. Pala utilizada en las competiciones de trainera.	38
4.25. Profundidad de la pala en el ataque.	42
4.26. Ángulo del remo en la vuelta.	43
4.27. La mecánica de la palanca en el remo.	44
5.1. Ciclo completo de la palada.	46
A.1. Representación de la jerarquía de los objetos tridimensionales. . .	54
A.2. Relación entre los diferentes bloques de código.	55
A.3. Flujograma.	57
A.4. Menús para poder cambiar valores de las variables.	58
B.1. La simulación de la trainera en el ataque.	59
B.2. La simulación desde el editor en el final.	60
B.3. La vista del editor durante una simulación.	60
D.1. Secuencia de ejecución de cada función.	64
D.2. Código escrito del módulo Trainera.	65

Capítulo 1

Introducción

1.1. Resumen

Las traineras son un tipo de embarcación a remo que se utilizan para competir en regatas. Con el fin de entender la física que impulsa estas embarcaciones, se ha llevado a cabo este proyecto que finaliza con un simulador de trainera. La primera parte del proyecto, la que tiene que ver con este documento, abarca la simulación del remo en sí y el cálculo de las fuerzas o físicas que lo rodeen. Dentro de este documento, comenzaremos con la fuerza ejercida por el remero y lo que conlleva en la rotación local del remo. Para esta rotación hay diferentes variables que se han estudiado y que pueden alterar la velocidad del remo y la fuerza que ejercen al conjunto de la trainera. La razón por la cual se mueve el remo depende de valores introducidos manualmente los cuales se pueden cambiar obteniendo diferentes resultados. Para una visualización rápida e intuitiva de los cálculos se ha utilizado la representación en 3D de la trainera y todos sus componentes. Con este simulador se pueden apreciar detalles en movimiento y en posición de cualquier componente de la trainera.

1.2. Estado del arte

La física del remo no es una ciencia investigada aunque hay artículos sobre el tema pero todos ellos en la modalidad olímpica o banco móvil, al contrario de las traineras que son banco fijo. En ellas se tiene en cuenta las físicas del remo y del cuerpo humano, y de esto se han creado estudios pero con diferentes resultados, ya que no hay un consenso en la reproducción física de los movimientos y fuerzas que entran en juego. Este proyecto se ha basado en varios estudios de la física del remo en banco móvil, entre otros los estudios de “Modeling the Dynamics of Rowing” de William C. Atkinson [3], las investigaciones de Marinus van Holst[32], Anu Dudhia[1] y Luca Formaggia [21]. Muchos de ellos han estudiado lo mismo con algunas diferencias. Algunos de estos estudios han creado modelos informáticos de la modalidad de banco móvil pero que han ayudado

a progresar con este proyecto. Puesto que hay diferencias entre el banco móvil y el banco fijo en este estudio, ha tenido que haber una investigación teniendo estos estudios como base. También se pueden encontrar en Internet estudios que no han sido publicados en revistas tecnológicas, como es el caso del estudio de Sander Rosendaal [28].

1.3. Estudios previos necesarios para el proyecto

Para poder llevar a cabo el proyecto se han tenido que aprender diferentes campos como son la mecánica, aerodinámica, hidrodinámica etc... Para ello se han utilizado diferentes estudios, entre otros el registro en algunos MOOC. Un MOOC (del inglés, Massive Open Online Course) es un curso a través de Internet en el que un profesor enseña globalmente. Particularmente los cursos adscritos han sido, “Physics 1 for Physical Science Majors” [13], un curso de la “University of Colorado Boulder” sobre física en general, en el que se ha aprendido lo necesario para una física básica, y “Sports and Building Aerodynamics” [7], un curso de la Universidad Tecnológica de Eindhoven sobre la aerodinámica en el deporte, en el que se ha aprendido lo necesario para la aerodinámica e hidrodinámica. Ambos MOOC se han encontrado en la web de cursos llamada Coursera [10], en el cual se ofrecen cursos de todo tipo de áreas y ofrecidos por múltiples universidades.

Capítulo 2

Historia

2.1. Navío de pesca

Las traineras en un principio era una embarcación de madera dedicada a la pesca, impulsada a remo y a veces a vela, propia de la costa cántabra. Según un viejo documento francés la primera trainera se construyó en Hondarribia hacia 1750. El peso de estas embarcaciones era de 600-700kg, un mástil principal de 6 metros y un mástil delantero de 5 metros. El nombre de dicha embarcación proviene del nombre “traina” cuyo significado es red de pesca. Antiguamente, los pescadores llevaban estas redes de pesca en la parte trasera (popa) de la trainera, junto al patrón y al avistar los bancos de peces remaban en círculo para poder ir lanzando la traina y así cercar al banco de peces.

Los tripulantes de estas traineras eran hombres fuertes y robustos ya que emprendiendo desde puerto tendrían que remar durante horas hasta llegar a los caladeros. Una vez allí, cada embarcación con sus tripulantes cogían la mayor cantidad de pescado para luego venderlo. Ya terminada la pesca, volvían a puerto para después, vender el pescado en subasta. La embarcación más veloz era la que antes llegaba a puerto y la que, por lo tanto, a mejor precio vendería lo que horas antes habían pescado, dando lugar a las primeras competiciones improvisadas.

A pesar de que esa sea la función más conocida de las traineras, también tenían otra. En puertos de difícil acceso como Bilbao o Pasajes, las traineras eran las que se ocupaban de ayudar a acceder a los grandes veleros al puerto. Tras avistar desde la atalaya a los grandes barcos o veleros, las traineras se apresuraban para llegar los primeros a éste y así obtener la mejor recompensa. Del mismo modo que la actividad pesquera, el atoaje también dio origen a las regatas de traineras de hoy en día.

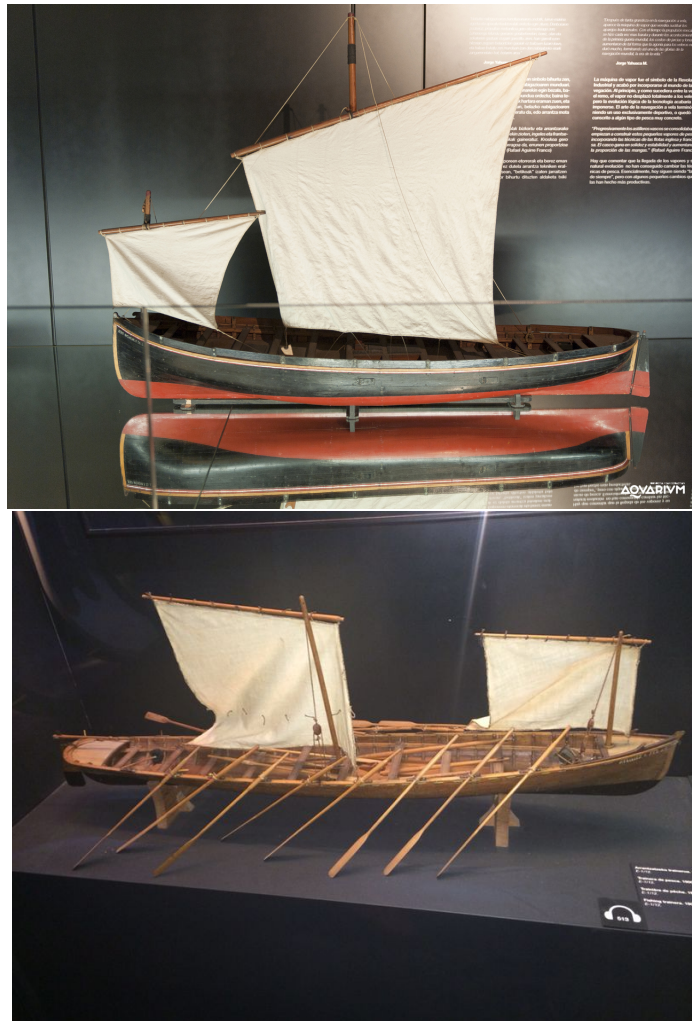


Figura 2.1: Maquetas de traineras impulsadas por velas expuesta en el aquarium de San Sebastián.

2.2. Las primeras competiciones

La primera regata escrita fue el 22 de Septiembre de 1871 en la que las traineras de Pasaia y Hondarribia fueron partícipes. La regata, organizada por razón de una apuesta entre los dos pueblos, se hizo desde Hondarribia a San Sebastián. La trainera “Avante” de Pasaia fue la vencedora de aquella regata. En el año 1879, ocho años más tarde de aquella apuesta, el ayuntamiento de San Sebastián, aprovechando que empezaba a disfrutar de un naciente turismo de alto nivel económico, con el fin de enseñar a los turistas las costumbres

tradicionales, organizó a través de los ayuntamientos de Pasaia y Hondarribia una regata para que traineras de puertos cercanos se uniesen a aquella fiesta marinera. Tras años de éxito de esta regata el ayuntamiento de San Sebastián la oficializó en 1901 como la regata de La Concha, siendo Orio la ganadora.

En vista de la cantidad de seguidores de las regatas de La Concha, cada vez eran más las regatas organizadas. Aprovechándose de tal situación y siendo consciente de que las traineras usadas eran muy pesadas y no estaban pensadas para competir, Vicente Olazabal construyó en 1916 en Mutriku una trainera estilizada de 400 kilos para el club de remo Getaria, a la que estos bautizaron como “Golondrina”. Paralelamente, los vapores de pesca arrinconaron definitivamente las embarcaciones de altura movidas a remo y empezaron a construirse traineras sólo para competiciones, mejorándolas cada vez más con el tiempo.

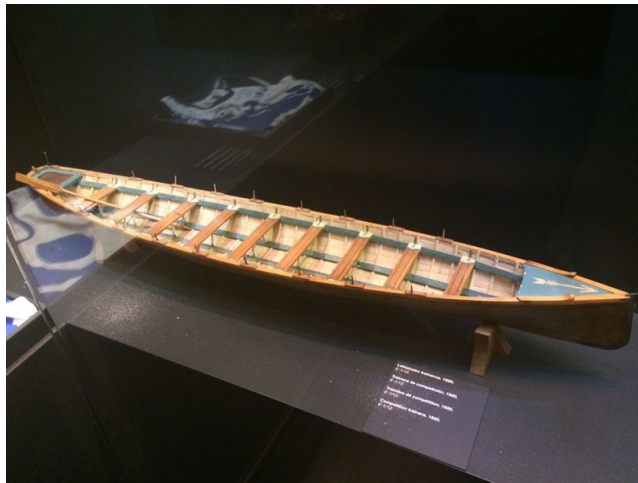


Figura 2.2: Maqueta de una trainera de competición de 1920 expuesta en el aquarium de San Sebastián.

2.3. **Normativa**

Visto el éxito de las competiciones se dictó una normativa estándar para toda España, así quedarían definidas las medidas para cualquier trainera de competición. Toda embarcación que no cumpla estas reglas no se puede considerar como trainera:

Descripción	Valor
Numero máximo de remeros	13 + 1 Patrón
Peso mínimo del casco	185 Kg
Peso mínimo de la embarcación (con accesorios no móviles)	200 Kg
Eslora máxima	12 metros
Puntal mínimo a proa	0,95 metros
Puntal mínimo a popa	0,75 metros
Puntal mínimo en la cuaderna maestra	0,605 metros
Manga mínima de la cuaderna maestra	1,72 metros
Manga exterior de la cuaderna maestra a 24 cm	1,18 metros
El bota-aguas no podrá alterar la eslora en más de 20 cm y la altura en 45 cm	

Con estas normas también llegó la definición de las embarcaciones de banco fijo como la trainerilla (6 remeros + 1 patrón) y el batel (4 remeros + 1 patrón).

No sólo las embarcaciones se estandarizaron sino también las competiciones ya que hasta entonces cada organizador definía sus normas.

Descripción	Valor
Distancia recorrida	3 millas náuticas (5556 metros)
Número de largos	4 largos (1389 metros cada una)
Número de giros o ciabogas	3 ciabogas
Número de calles	4 calles
Separación mínima entre calles	45 metros
Separación máxima entre calles	50 metros
Número de participantes	12 tripulaciones
Numero de tandas	3 tandas

La ciaboga es el giro de 180° que se ejecuta alrededor de una boya, dejándola ésta por babor (izquierda del bote a ojos del patrón). El giro dependía mayormente del patrón, que con su remo giraba completamente la trainera. Esto cambió en 1919, en la regata de la Bandera del Abra en Bilbao, Francisco Zubiaurre "Kiriko", patrón de la trainera donostiarra, dio la ciaboga utilizando un remo-timón en proa (parte delantera del bote), lo que conseguía girar de forma vertiginosa. Después de esto, al ver la ventaja que conllevaba, todos los equipos incluyeron en la trainera un "cuerno" en la aleta de estribor (parte delantera derecha a ojos del patrón), para facilitar esta maniobra.

Desde entonces a cada regata, organizada por ayuntamientos o empresas privadas, invitan a 12 tripulaciones para enfrentarse entre ellas y lograr diferentes premios, siendo una bandera la del ganador. Estas regatas iban desde junio hasta septiembre, aprovechando el buen tiempo del verano. Siendo las más importantes las banderas de la Concha[12], Bilbao y Santander.

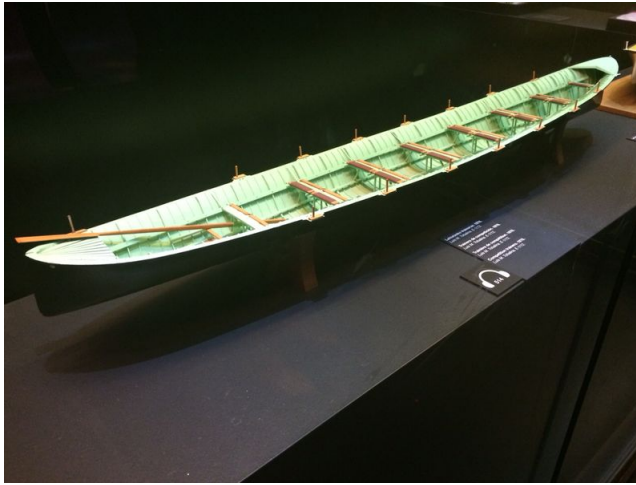


Figura 2.3: Maqueta de una trainera de competición de 1980 expuesta en el aquarium de San Sebastián.

A mediados de los 90, con la investigación de los materiales, llegó la innovación de las traineras, dando paso a la fibra de carbono tanto para el casco como para los remos haciendo estos últimos más ligeros y flexibles.

2.4. Competiciones

Tras un acuerdo con los Gobiernos de las comunidades autónomas de Galicia, Asturias, Cantabria y Euskadi se creó el 2 de Julio de 2003 una asociación llamada Asociación de Clubes de Traineras o ACT [11]. Esta asociación tendría como objetivo impulsar el remo en esta disciplina con los siguientes objetivos:

- Hacer de la Liga una competición de alto nivel amateur, gestionada profesionalmente, en la que participen los doce mejores equipos pertenecientes al área geográfica.
- Conseguir incrementar notoriamente el número de seguidores, potenciando la atención y cobertura que los medios de comunicación prestaban anteriormente.
- Diseñar la competición en formato de Liga Regular para que resultase más atractiva y comprensible para todos.
- Garantizar su estabilidad en el tiempo mediante una fase de ascenso y descenso con clubes provenientes de competiciones inferiores.
- Posicionar la Liga ACT como un producto deportivo moderno, sólido y de calidad, que fomente la práctica del remo y que en verano ofrezca la posibilidad de disfrutar de un deporte-espectáculo en un amplio ámbito

geográfico y en un entorno incomparable como es la costa de la cornisa cántabro-atlántica.

En el año 2006, visto el éxito de éste y el sistema de ligas, en ese mismo año se crearon ligas inferiores a la liga ACT para tener traineras que optasen por ascender y descender de ella, dividiéndose en dos ligas gallegas y dos para el resto del cantábrico.

Aparte de la categoría masculina, también hay categoría femenina en el mundo del remo. Para impulsar el remo femenino la ACT creó una liga femenina. Esta liga tiene algunas diferencias frente la masculina, entre otras, el recorrido es de dos largos de distancia que equivalen a 2778 metros y que cada año se eligen las 4 mejores, en 2 regatas clasificatorias, para optar a ganar la liga.

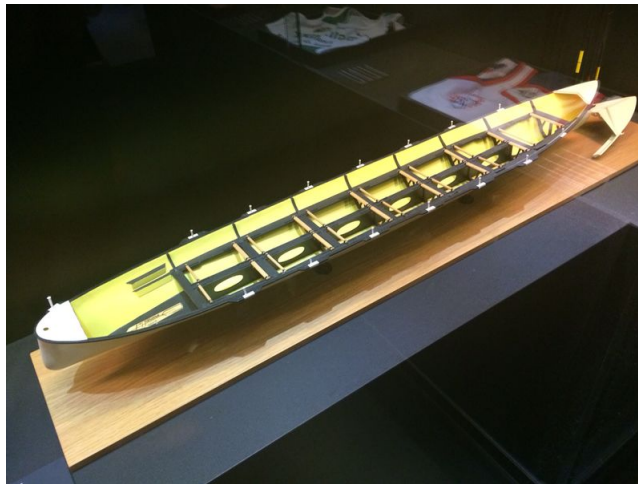


Figura 2.4: Maqueta de una trainera de competición de 2010 expuesta en el aquarium de San Sebastián.

Capítulo 3

Diseño

3.1. El simulador

Este proyecto tiene como fin crear una aplicación informática donde se implementen las leyes físicas que producen el movimiento de estas embarcaciones y así reproducir o simular la realidad, de ahí la clasificación de simulador. El proyecto tendrá dos partes que son la parte física y la parte gráfica. La parte física se encargará de todos los cálculos para predecir los movimientos, fuerzas, etc... que se necesiten en cualquier momento y la parte gráfica se encargará de visualizar la trainera y sus componentes para poder ver de una forma fácil e intuitiva todo cambio de la física. La función de esta aplicación es saber cómo la fuerza de los remeros influye en el movimiento de la trainera. Introduciendo los datos de cada remero en cada posición se logrará un comportamiento preciso, pero al cambiar los datos de estos, como por ejemplo el peso, la fuerza o la altura, produzca alteraciones en el comportamiento general del bote.

3.2. Elección de librerías

Un proyecto de estas dimensiones no se puede programar desde cero ya que sería demasiado trabajo y conllevaría un trabajo innecesario. Es por esto que se usan programas ya creadas por otros informáticos denominadas bibliotecas (o librerías por la similitud a la palabra biblioteca en inglés, “library”) que te ahorran un tiempo y un trabajo importante.

En este proyecto se usarán dos tipos de librerías que no son usadas muy a menudo en una simple aplicación informática, que son los denominados motores físicos y motores gráficos. El primero ayuda en el cálculo de las fuerzas y la física relacionada, y el segundo ayuda en la visualización de los gráficos e imágenes de la trainera.

3.2.1. Candidatos para el motor físico

Para cualquier simulador la programación de la física es imprescindible, ya que en la simulación deben de actuar las mismas leyes físicas que en la realidad. Un motor físico es un grupo de librerías que ofrecen las herramientas para lograr una física adecuada a cada proyecto pudiendo modificarlo de una forma fácil. Estas librerías ayudan a conseguir efectos físicos de una manera simple como por ejemplo la gravedad o las colisiones entre objetos. Aunque la realidad es siempre la misma hay varios motores físicos que pueden representarla, por esto hay que hacer una sabia elección para ver cuál es la que mejor se puede adecuar a nuestro proyecto. Se pueden clasificar los motores físicos de acuerdo a la capacidad de cálculo que requieren: aquellos de simulación en tiempo real y aquellos de alta precisión. Para nuestro proyecto usaremos la simulación en tiempo real ya que el cálculo de alta precisión se escapa de la media del nivel computacional de los PCs.

3.2.1.1. NVIDIA PhysX Physics

Este motor físico fue creado por AGEIA, la cual luego fue adquirida por la gigante Nvidia[22] para poder integrarlo en sus chips. Los sistemas operativos en los cuales se puede utilizar están Windows[35], Mac OS[24], Linux (con restricciones)[20], Wii[34], PlayStation 3[27] y Xbox 360[36]. PhysX[26] es de los motores más utilizados en el mercado gracias a que en 2005 Sony[29] firmó un acuerdo para usar el motor físico en su consola PlayStation 3, lo que le dio un gran empuje. El único punto negativo que tiene el motor físico es que tiene licencia propietaria aunque sólo haya que pagar cuando se quiera modificar el motor gráfico en sí y no al utilizarla.

3.2.1.2. Bullet

Este motor físico es de los más utilizados por la comunidad y es por esto que es la que más documentación tiene al respecto. Bullet[25] está implementado en C y C++ bajo la licencia zlib el cual es compatible con GNU General Public License con la cual se puede usar, estudiar, compartir (copiar) y modificar el software. Entre sus usos están PlayStation 3, XBox 360, Wii, PC, Linux, Mac OS, Android[2], iPhone[19] y más sorprendente aún, el cine.

3.2.1.3. Newton Game Dynamics

Otro de los candidatos a ser el motor físico utilizado en el proyecto es Newton Game Dynamics[14]. Este motor físico es de código abierto, con licencia zlib, lo que también lo hace editable y personalizable. Este motor físico contiene diferentes comportamientos implementados como son el manejo de la escena, detección de colisiones y el comportamiento dinámico. Las plataformas en la que se puede utilizar dicho motor físico son Windows, Mac OS X, iPhone OS y Linux, por lo que puede ser utilizado en las plataformas más populares. La versión más estable es la 2,36 la cual fue publicada en octubre de 2012 aunque

podemos encontrar una versión beta aún más avanzada que es la 3,11 publicada en octubre de 2013 por lo que a día de hoy puede estar un poco anticuada por falta de soporte.

3.2.2. Candidatos para el motor gráfico

Para llevar a cabo la parte gráfica del proyecto se podría crear todo programando pero eso no es lo que se va a innovar por lo que se puede utilizar un conjunto de librerías gráficas. Este conjunto es llamado motor gráfico o motor de juegos y es el encargado de crear de una forma “sencilla” los gráficos 3D y su animación. Es la base de todo juego o simulador que se encuentre en el mercado, sin ellos se debería de crear todo el mundo virtual a un nivel tan bajo de programación que haría de la producción de estos proyectos algo costoso y no rentable. Hay muchas empresas y grupos de programadores que han creado su propio motor gráfico, cada una con sus puntos fuertes y débiles. Por esta razón la elección de un motor gráfico que se adecue a nuestro proyecto es fundamental.

3.2.2.1. Ogre 3D

Ogre3D[23] (acrónimo del inglés Object-Oriented Graphics Rendering Engine) motor gráfico de software libre. Entre otras plataformas soporta Linux, Windows (las últimas versiones), OS X, NaCl, WinRT, Windows Phone 8, iOS y Android. Este motor gráfico no incluye un motor físico ni audio por lo que se deben de importar para poder usarlos. Aunque tiene una comunidad muy activa se está quedando atrás comparadas con otros motores gráficos. Este motor gráfico no es utilizado en juegos comerciales por estas razones. A su favor tiene que es GNU GPL por lo que puedes modificar a tu gusto el motor gráfico haciéndolo personalizado a tu proyecto. Al ser libre no tiene ningún tipo de coste a la hora de comercializar el producto.

3.2.2.2. Unreal Development Kit

El Unreal Engine[17] es desarrollado por Epic Games[18], creadores de conocidos juegos como las sagas Unreal[30] o Gears of War[33]. Es uno de los motores más utilizados para desarrollos AAA (clasificación excelente en todos los apartados del juego: sonido, gráficos y jugabilidad) y la lista de juegos que usan este motor es extensa, incluyendo títulos como BioShock[4] o Mass Effect[15]. Hay 2 opciones para usar el Unreal Engine. Por un lado está disponible el UDK (Unreal Development Kit), que permite desarrollar proyectos completos sin ninguna restricción usando las herramientas del motor y todas sus características de forma gratuita, pero sin poder modificar internamente el motor. Actualmente los términos de comercialización del producto final son los siguientes: Para poder vender el producto hay que adquirir primero una licencia que cuesta 99\$. Una vez en venta, no hay que pagar más mientras los ingresos no superen los 50.000\$, momento en el que deberá pagarse un 25 % de los siguientes ingresos a Epic. La segunda opción es adquirir una licencia del motor completo, con lo que

se obtendría acceso al código fuente del mismo, pudiendo modificarse para ajustarlo a las necesidades particulares. Al poseer la licencia completa del motor, no habría que pagar más por comercializar el producto final. Existen tutoriales, foros y una extensa documentación que son de gran utilidad.

3.2.2.3. Unity3D

Unity3D[31] es uno de los motores gráficos del mercado, éste incluye una interfaz gráfica con la que poder crear cualquier aplicación con gráficos en 3 dimensiones como juegos, simuladores, vídeos, etc... Unity3D además de ser muy potente como motor gráfico también es muy versátil ya que las aplicaciones creadas con ésta herramienta se pueden utilizar desde móviles a computadoras con cualquier sistema operativo pasando incluso por aplicaciones Web. El lenguaje de programación definido para el uso de esta herramienta es C# que es un lenguaje de programación creado por los creadores de Windows, Microsoft, aunque su núcleo esté programado en C++.

Uno de los mayores inconvenientes de usar Unity3D es que tiene licencia propietaria, por la cual hay que pagar un coste aproximado de 1.500 dólares americanos. También existe una versión gratuita de esta herramienta pero está limitada a uso privado y empresas pequeñas. A las diferentes versiones se les llama Unity Pro y Unity Free respectivamente. El segundo inconveniente que se le puede achacar a dicha herramienta es que sólo puede ser ejecutada en el sistema operativo Windows.

Las dos versiones de Unity3D incorporan el motor físico NVIDIA PhysX Physics por lo que facilita la creación de aplicaciones a la hora de enlazar todas las librerías.

3.3. Elección de motores

Lo valorado en los motores tanto físicos como gráficos ha sido que sea libre para poder editarlo y publicar este documento sin ningún tipo de impedimento. Sobre la plataforma a usar se acordó utilizar Linux por ser libre y ofrece una mayor maniobrabilidad a la hora de crear programas. Por eso, las opciones que se optaron fueron:

- 1. Opción: Bullet + Ogre3D.
- 2. Opción: PhysX + Unity3D.

Una vez el proyecto fue iniciado hubo problemas con la implementación del proyecto de Ogre 3D, ya que las instalaciones y los enlaces a las librerías externas (como en nuestro caso Bullet) dan problemas. Otro de los inconvenientes es que los modelos en 3D tienen que ser en un formato muy específico que necesita diferentes tratamientos para hacer posible la lectura de estos. Viendo los inconvenientes que tenía la elección hecha se decidió hacer un cambio con lo que conllevaba en tiempo perdido.

En este caso se decidió dejar los valores prioritarios aparte para centrarse en la simplicidad a la hora de configurarlo todo completamente, ya que no se podía perder más tiempo configurando programas en vez de empezar con el propio proyecto en sí. Para esto se pasó a la segunda opción, emplear Unity3D, ya que con la interfaz gráfica que incluye simplifica mucho las cosas al tener incorporado los motores gráficos y físicos, centrándote sólo en la implementación de nuestro proyecto. Con esta elección lo único que se ha perdido comparando con los valores iniciales ha sido la plataforma en la que implementar ya que Unity3D sólo se usa en Windows. Al tener la versión gratuita y libre de este motor aún se puede publicar esta documentación sin entrar en conflictos económicos.

Capítulo 4

Implementación

4.1. Creación de modelos en 3D

En una simulación todo objeto o persona de la vida real debe de ser representado mediante un objeto en 3 dimensiones (3D). Hay diferentes programas que sirven para la creación y edición de dichos modelos, tanto comerciales como libres. Al igual que los motores se ha optado por utilizar un programa de modelado que sea libre, y el más conocido y usado entre estos es Blender [5]. Con este programa podemos, no sólo crear objetos en 3D sino también iluminación, animación, e incluso con su motor físico crear juegos. Es un programa que está en constante mejora ya que es una amplia comunidad la que da soporte a este programa. Una de las razones que impulsaron el uso de esta aplicación es el uso previo en algunas asignaturas de la universidad.

Con este programa se han creado diferentes modelos, entre ellos:

- Trainera
 - Casco

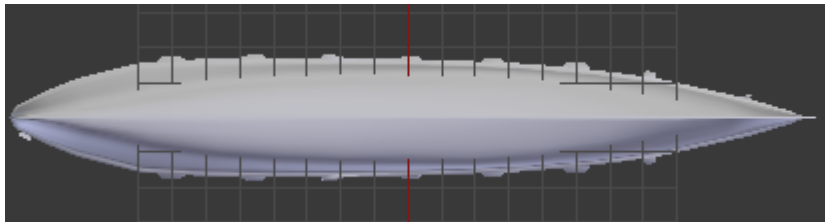


Figura 4.1: Casco en 3D.

- Cuadernas

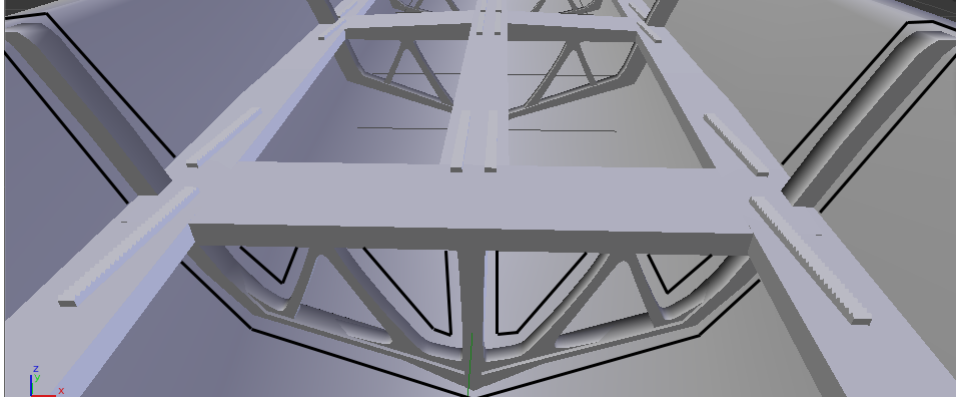


Figura 4.2: Cuadernas en 3D.

- Tostas

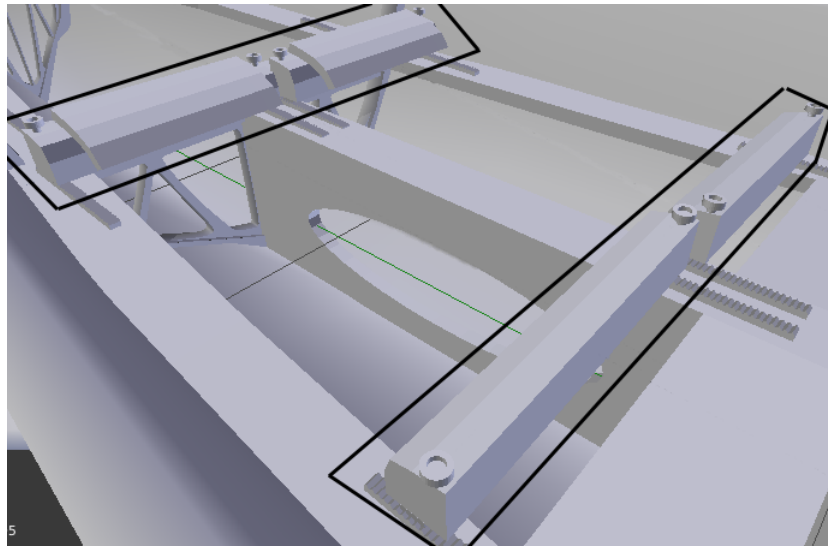


Figura 4.3: Tostas en 3D.

- Cremalleras

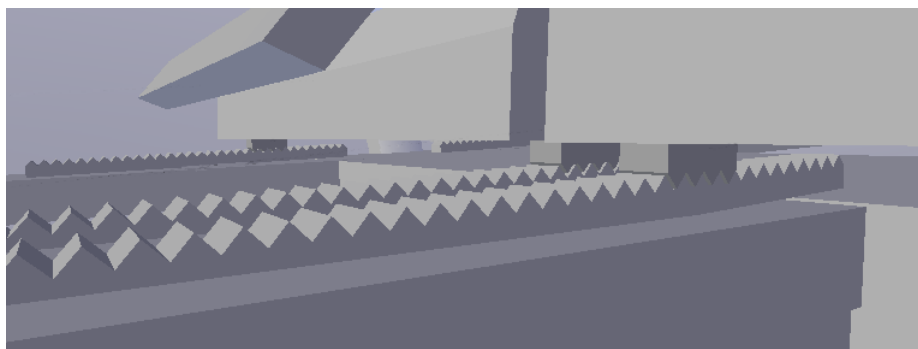


Figura 4.4: Cremalleras en 3D.

- Cuerno

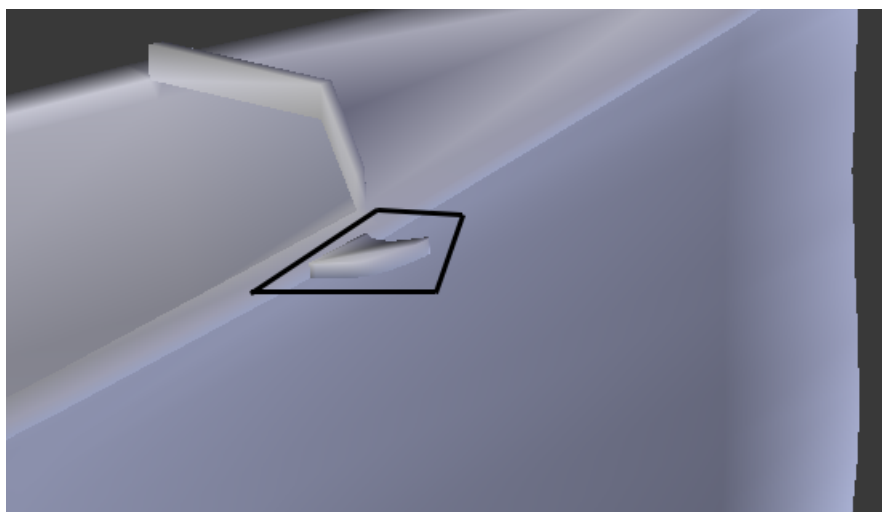


Figura 4.5: Cuerno en 3D.

- Apoyos del patrón (escaleras para subirse y tener mayor estabilidad)

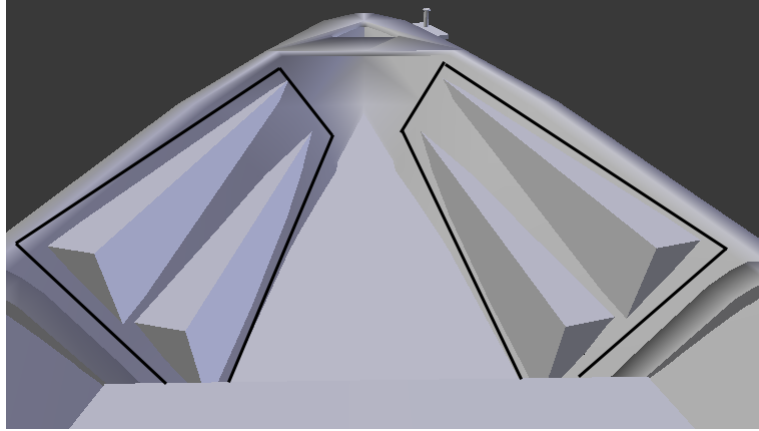


Figura 4.6: Apoyos del patrón en 3D.

- Toleteras

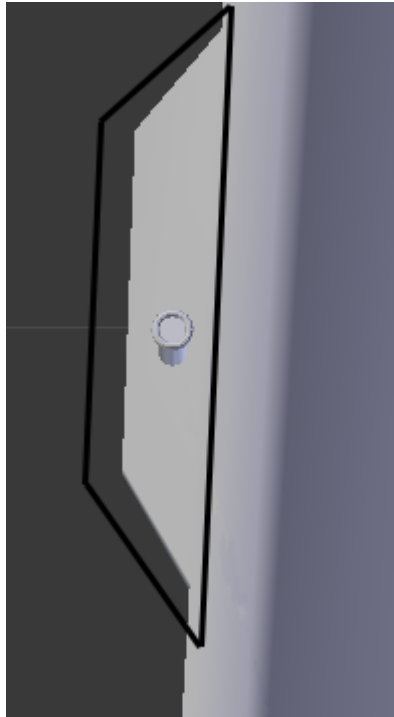


Figura 4.7: Toleteras en 3D.

- Toletes

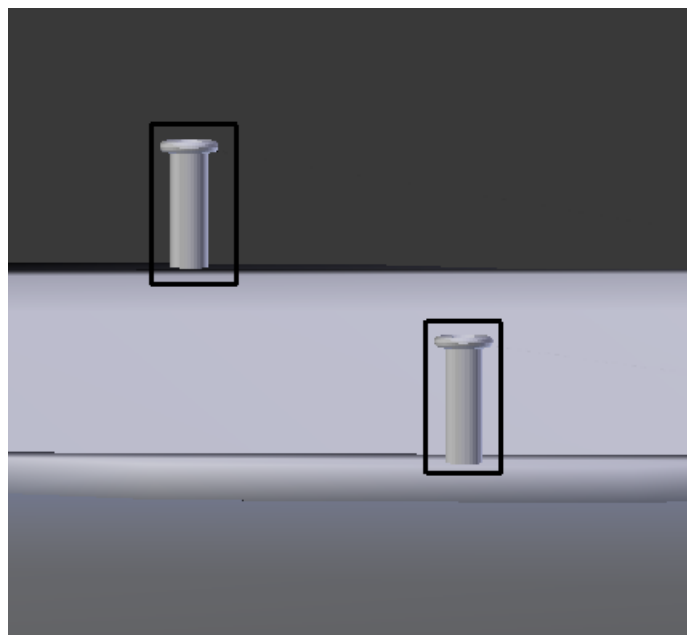


Figura 4.8: Toletes en 3D.

■ Remos

• 12 Remos convencionales

○ Pala

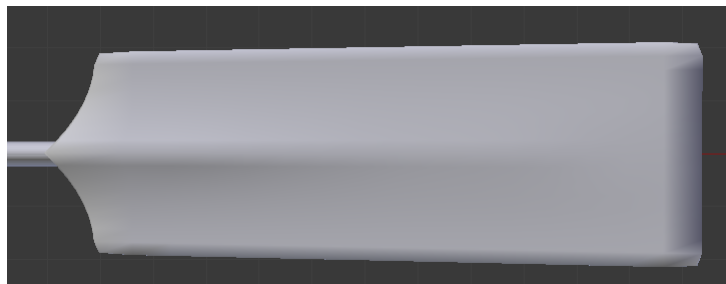


Figura 4.9: Pala del remo en 3D.

○ Tope

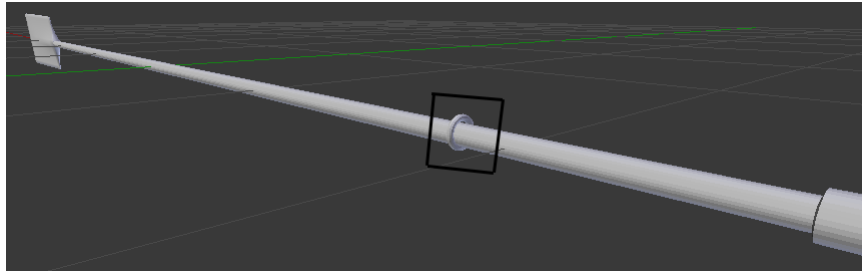


Figura 4.10: Tope del remo en 3D.

- Caña

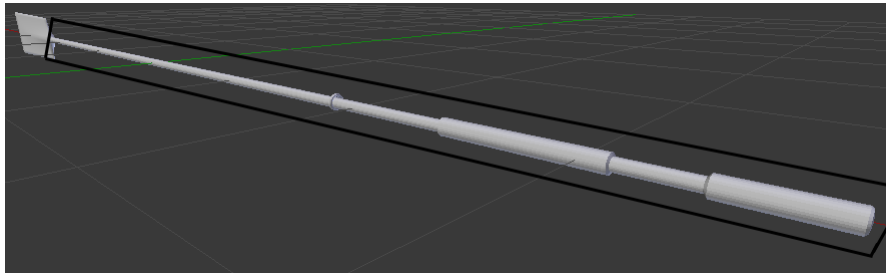


Figura 4.11: Caña del remo en 3D.

- 1 Remo de proa (más largo de lo normal)



Figura 4.12: Remo de proa en 3D.

- 1 Remo de patrón (más largo y con una pala mucho más larga)

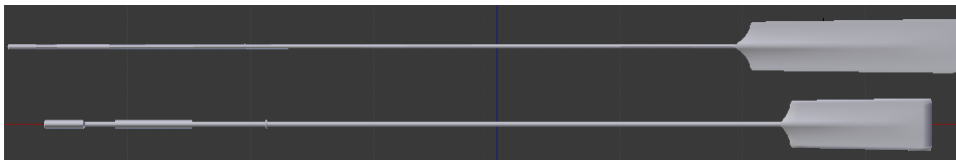


Figura 4.13: Remo de patrón en 3D.

No todos los modelos han sido creados, por ejemplo, las personas y el entorno se han descargado de Internet, de la página web danim.tv[6] con licencia CC[8]

(Creative Commons) pero las personas serán introducidas en el sistema en un futuro. En esta fase del proyecto se ha optado por crear objetos redondeados para hacer referencia al lugar en donde serían colocados los remeros.

4.2. Física

Quitando lo que el motor de física nos ofrece se deben de hacer cálculos para simular movimientos específicos. Como la fuerza que ejerce movimientos a toda la trainera es el propio remero, se ha empezado por la fuerza ejercida por éste, y en el primer objeto en el que se reproduce esa fuerza, que es el remo.

4.2.1. Fuerza ejercida por el remero

El remero es la única parte activa de la trainera, y gracias a esto es por lo que se mueve. La fuerza ejercida por el remero hace que se mueva el remo que con la resistencia del agua produce el movimiento de la trainera. Esta fuerza se produce con la espalda del remero que es aplicada al remo a través de los brazos. La colocación de los brazos es muy importante ya que si los colocas muy cerca del eje de rotación del remo se debe de producir más fuerza para poder moverlo. La posición de cada mano es un dato a introducir en el sistema ya que el punto medio entre las dos manos será el punto donde se ejerza la fuerza al remo. Antes de entrar en materia será mejor definir lo que es la fuerza.

4.2.1.1. Fuerza

La fuerza (F) es un valor vectorial que mide la intensidad del intercambio de momento lineal entre dos partículas o sistemas de partículas. La magnitud del vector o la cantidad de fuerza en el sistema internacional es el Newton (N) aunque también es muy usado el kilogramo-fuerza (kgf) o kilopondio que es equivalente a 9,81N. La fórmula utilizada para definir la fuerza es ésta:

$$\vec{F} = m \times \vec{a}$$

Lo que viene a decir lo que por lógica entendemos, para poder lograr una aceleración (a) a una masa (m) debemos aplicar una fuerza (F), y cuanto mayor sea el peso mayor es la fuerza para lograr la misma aceleración.

Sabiendo que la fuerza es vectorial en nuestro caso debemos definir el punto en el que el remero ejerce la fuerza al remo; el simulador se ha diseñado de forma que este lugar sea el punto exacto entre las dos manos. Como valor de entrada se tomará la distancia en centímetros hasta la punta del remo, siendo 0 cm el extremo contrario a la pala. Teniendo el punto en el que se aplica se necesita saber hacia qué dirección es ejercida esa fuerza. En un principio la dirección que se eligió fue el punto central entre los dos hombros, ya que en el ataque los dos brazos están rectos pero cada brazo empuja en una dirección diferente y había que buscar un punto medio. El problema vino cuando el remo, tras haber rotado lo suficiente, llega al punto en el que el remero queda detrás del remo,

o mas bién en la dirección del remo, momento en el que el remero no ejercería ninguna fuerza que hiciera rotar al remo ya que la fuerza tiraría del remo hacia el remero. Como se puede apreciar en la figura 4.14 los brazos están representados por líneas blancas y la dirección de la fuerza por la línea azul, y concuerda en cierto modo a la realidad, pero en la figura 4.15 vemos cómo el remo ha pasado el punto en el que la línea azul hace que la fuerza se ejerza hacia el lado que no debería.

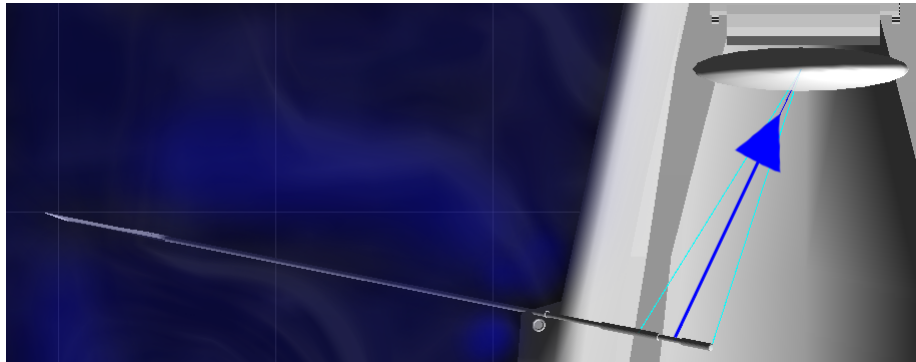


Figura 4.14: Posición de los brazos y dirección de la fuerza en el ataque.

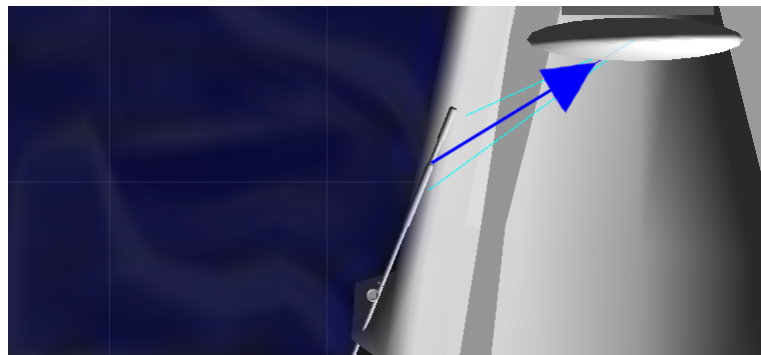


Figura 4.15: Posición de los brazos y dirección de la fuerza en el final.

Para arreglar este problema se optó por cambiar la dirección y que éste apuntase siempre a la parte en la que la fuerza es ejercida en la realidad estando siempre en el plano horizontal. Simulando la fuerza en la misma dirección como se aprecia en la figura 4.18.

En el caso de la cantidad de fuerza, al no ser todas las personas iguales, la fuerza ejercida por cada remero varía, pero además, la fuerza ejercida en cada momento de la palada también varía. La representación gráfica de la fuerza en una palada es parecida a la distribución gaussiana. Lo normal suele ser que en el ataque crezca hasta alcanzar el máximo de fuerza y que luego disminuya, dando

lugar a la representación de la figura 4.16. Estos datos se consiguen utilizando un ergómetro (Figura 4.17), esta máquina es una máquina para realizar una simulación de la acción de remar, con propósito de realizar ejercicio o entrenar para remo. El productor del software y ergómetros es Concept2 [9], gracias al software que está disponible en su página web es posible extraer estos datos.

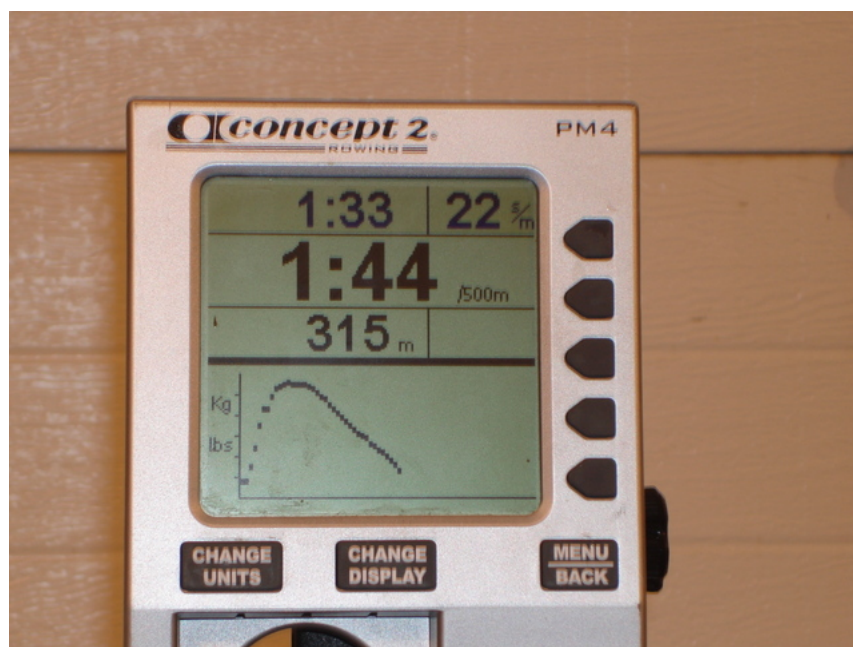


Figura 4.16: Curva de fuerza vista en la parte inferior de la pantalla de un ergómetro.



Figura 4.17: Un ergómetro o remoergómetro.

Los ergómetros están pre-diseñados para la modalidad de banco móvil, por lo que no son fiables al 100 % para la modalidad del banco fijo. La extracción de los datos del ergómetro no ha sido posible antes de la finalización de esta parte del proyecto por lo que se ha definido una fuerza constante de la fuerza máxima de un remero de banco móvil en velocidad de crucero, que es de 200 N.

4.2.2. Cantidad de rotación del remo

El remo se fija a la trainera en un cilindro vertical llamado tolete usando un aro que suele ser de nailon o esparto. Para que el remo se inserte a la distancia precisa en la que el entrenador prefiera, para aumentar la palanca y así controlar la dureza de la remada, se instalan unos aros en el remo con mayor diámetro que éste.



Figura 4.18: Remos anclados en el tolete con el estrobo a la altura del tope.

El remero al ejercer una fuerza al extremo del remo y, éste, al estar inmovilizado por el estrobo en el punto donde el tope sea colocado produce un movimiento rotacional, teniendo como eje el propio tope. El movimiento rotacional se produce cuando una fuerza es ejercida en cualquier lugar (excepto el centro exacto) y sea perpendicular al radio. De esta relación entre la cantidad

de fuerza y la posición en la que se aplica surge el momento.

4.2.2.1. Momento de fuerza

El momento de fuerza (τ) es una magnitud que da a conocer en qué medida existe la capacidad de una fuerza para cambiar el estado de rotación de un objeto. El momento se mide en unidades de Newton * metro (N*m o Nm) y se representa como un vector perpendicular a los vectores del radio (R) y la fuerza (F). Y el momento se define así:

$$\vec{\tau} = \vec{R} \times \vec{F}$$

En Unity3D a la hora de hacer los productos vectoriales utiliza la regla de la mano izquierda por lo que estas multiplicaciones dan un resultado en la dirección contraria lo que es un dato a tener en cuenta.

Como la fuerza (F) debe ser perpendicular al radio (R) podemos definir la siguiente fórmula para cualquier fuerza, utilizando el ángulo entre los dos vectores (α):

$$|\tau| = RF \sin \alpha$$

Insertando en la fórmula el seno del ángulo que crean los dos vectores tenemos la fuerza perpendicular. Se puede apreciar mejor gráficamente:

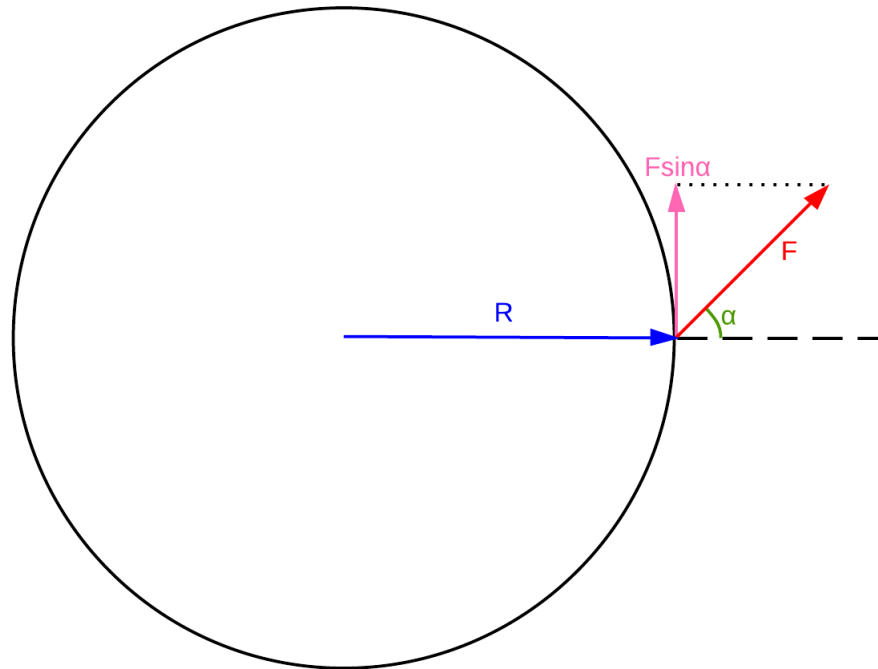


Figura 4.19: Obtención del vector perpendicular al radio para el cálculo del momento de fuerza.

La dirección del vector del momento de fuerza depende de la dirección de rotación del objeto, obedeciendo a la regla de la mano derecha o sacacorchos. Esta regla se llama así ya que se cumple utilizando la mano derecha con la mano cerrada y colocando los dedos en la dirección de la rotación y el pulgar apunta automáticamente al vector del momento de fuerza como podemos apreciar en la siguiente imagen:



Figura 4.20: Representación gráfica de la regla de la mano derecha o sacacorchos.

Como detalle técnico a comentar, en Unity3D se utiliza, al igual que en los productos vectoriales, la regla de la mano izquierda en todas las rotaciones por lo que es un dato a tener en cuenta, ya que es el contrario al que se obtiene con la regla de la mano derecha.

Una vez que se sabe la dirección de la rotación y la fuerza ejercida se debe conseguir la cantidad de rotación del objeto a la que corresponde dicha fuerza y para eso hay que valerse de otro fenómeno físico llamado momento de inercia:

4.2.2.2. Momento de Inercia

El momento de inercia (I) de un cuerpo indica su resistencia a adquirir una aceleración angular. Este fenómeno es debido a que los pesos (m) lejanos al eje de rotación son más difícil de acelerar. Se define en forma discreta y continua, respectivamente de la siguiente manera:

$$I = \sum m_i r_i^2$$

$$I = \int r^2 dm$$

El momento de inercia es una magnitud escalar y que da un valor que aumenta en mayor medida con mayores distancias (r), ya que cuanto más lejos la inercia aumenta cuadráticamente. Un ejemplo simple se puede ver en el patinaje sobre hielo (Fig. 4.21) y cómo al rotar sobre su propio eje al acercar los brazos al cuerpo aumenta la velocidad de rotación.

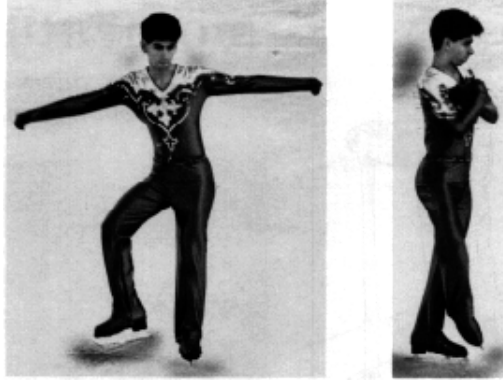


Figura 4.21: Patinador girando sobre su propio eje con diferentes inercias y sus consiguientes velocidades.

Volviendo al caso particular que se describe, tenemos el remo con dos masas diferenciables, la caña y la pala. La inercia total se debe calcular sumando las diferentes inercias del cuerpo compuesto, así:

$$I_{Total} = I_{Caña} + I_{Pala}$$

Para poder facilitar los cálculos se evita el cálculo de la integral resolviéndola de una forma generalizada, tomándolo como si fuese una varilla donde el peso de cada componente se divide uniformemente por todo él. Esto es, el diferencial de la masa (dm) se puede sustituir por la masa uniforme que equivale a esa porción, que sería como en cualquier lugar de la varilla, la masa (M) entre la longitud (L), así:

$$I = \int r^2 dm = \int r^2 \frac{M}{L} dr$$

Una vez hecho este cambio, la integral se simplifica a la hora de resolverla, ya que la masa (M) y la longitud (L) de la varilla son constantes :

$$I = \frac{M}{L} \int r^2 dr$$

Por lo que lo único que falta para poder resolver la integral serían los límites de éstas, que para cada caso es diferente:

$$I_{Caña} = \frac{M_{Caña}}{L_{Caña}} \int_{L_{Tiborta}}^{L_{Pala}} r^2 dr$$

para la caña, dónde,

- $M_{Caña}$ es la masa total desde la punta de la tiborta (lugar en el que se posicionan las manos al remar) hasta la parte dónde coinciden la caña con la pala,

- $L_{Caña}$ es la longitud total desde la punta de la tiborta hasta la parte dónde coinciden la caña con la pala,
- $L_{Tiborta}$ es la longitud desde el punto de rotación hasta la punta de la tiborta, en negativo por ser en la dirección opuesta,
- L_{Pala} es la longitud desde el punto de rotación hasta la parte dónde coinciden la caña con la pala.

Y para la pala:

$$I_{Pala} = \frac{M_{Pala}}{L_{Pala}} \int_{L_{Interior}}^{L_{Exterior}} r^2 dr$$

dónde,

- M_{Pala} es la masa total desde la parte dónde coinciden la caña con la pala hasta el extremo de la pala,
- L_{Pala} es la longitud total desde la parte dónde coinciden la caña con la pala hasta el extremo de la pala,
- $L_{Interior}$ es la longitud desde el punto de rotación hasta la parte dónde coinciden la caña con la pala,
- $L_{Exterior}$ es la longitud desde el punto de rotación hasta el extremo de la pala.

Una vez definido todo se puede proceder a la resolución de la integral:

$$I = \frac{M}{L} \int_{L_I}^{L_F} r^2 dr = \left[\frac{M}{L} \frac{1}{3} r^3 \right]_{L_I}^{L_F} = \frac{M}{3L} (L_F)^3 - \frac{M}{3L} (L_I)^3 = \frac{M}{3L} (L_F^3 - L_I^3)$$

por lo que unificando todo el momento de inercia de la pala será:

$$I_T = \frac{M_{Caña}}{3L_{Caña}} (L_{Pala}^3 - L_{Tiborta}^3) + \frac{M_{Pala}}{3L_{Pala}} (L_{Exterior}^3 - L_{Interior}^3)$$

El momento de inercia junto con el momento de fuerza nos ofrece lo necesario para poder saber la aceleración angular del remo, pero para eso mejor definir lo que es la velocidad y aceleración angular.

4.2.2.3. Velocidad y Aceleración Angular

La velocidad angular (ω) es la velocidad con la que varía el ángulo (ϑ) en el tiempo y se mide en radianes por segundos (rad/s). Matemáticamente se define así:

$$\omega = \frac{d\vartheta}{dt}$$

La velocidad angular es un vector paralelo al eje de rotación, la magnitud es la conseguida con la fórmula y la dirección de ésta es la conseguida por la regla de la mano derecha. Para un objeto que gira alrededor de un eje, cada punto del objeto tiene la misma velocidad angular. A la variación de la velocidad angular en el tiempo se le llama aceleración angular (α) que se mide en radianes por segundo al cuadrado (s^{-2}) y se define así:

$$\alpha = \frac{d\omega}{dt}$$

La aceleración angular también es un valor vectorial pero éste no debe de estar en el eje de rotación.

Si la aceleración es constante se puede predecir la velocidad en el tiempo con esta fórmula, que se usará más adelante:

$$\omega = \omega_0 + \alpha t$$

4.2.2.4. Segunda Ley de Newton para la rotación

Las leyes de Newton, también conocidas como leyes del movimiento de Newton, son tres principios a partir de los cuales se explican la mayor parte de los problemas planteados por la mecánica, en particular, aquellos relativos al movimiento de los cuerpos. En este caso de las tres leyes de Newton se va a utilizar la segunda, que dice:

“El cambio de movimiento es proporcional a la fuerza motriz impresa y ocurre según la línea recta a lo largo de la cual aquella fuerza se imprime.”

lo que matemáticamente sería:

$$Fuerza = masa * aceleración$$

Esta fórmula es la que nos da la relación entre la fuerza ejercida a un objeto y la aceleración a la que conlleva. Llevándolo al área rotacional logramos lo llamado como la Segunda ley de Newton para la rotación, que se define así:

$$\tau = I\alpha$$

Esta fórmula viene a decir lo que en la lógica sería que a un objeto con mucho momento de inercia, o en otras palabras un gran peso lejos del eje, para acelerarlo en una pequeña cantidad necesitaría mucha fuerza.

En nuestra aplicación tenemos el momento de fuerza y el momento de inercia del remo, por lo que ya sabemos cuál es la aceleración en cada instante, con una simple operación matemática conseguimos la aceleración:

$$\alpha = \frac{\tau}{I}$$

Esta aceleración equivale a la aceleración angular que tiene el remo en un fotograma, es por eso que necesitamos la fórmula de la velocidad angular para saber cuál es la velocidad angular que tiene en ese mismo instante. La única

diferencia que el tiempo a multiplicar por la aceleración será el diferencial del tiempo, o en otras palabras, el tiempo que tarda de fotograma a fotograma. Quedando así:

$$\omega = \omega_0 + \alpha * dt$$

Como bien se ha dicho antes ésta fórmula sólo es válida cuando la aceleración es constante, por lo que se deduce que el tiempo entre fotograma a fotograma se mantiene constante al calculado en el fotograma previo. Así es posible utilizar esta fórmula, que es la que nos da la velocidad angular, que es lo necesario para saber el ángulo de rotación equivalente a la fuerza ejercida por el remero.

Como se ha dicho anteriormente la velocidad angular es la variación del ángulo en el tiempo, por lo que con otra simple operación matemática sabemos cuál va a ser la variación para el próximo fotograma.

$$d\vartheta = \omega dt$$

Con esta fórmula ya tenemos el ángulo que se va a rotar para el próximo fotograma, en el cual se volverá a ejercer una fuerza y se repetirán los mismos cálculos hasta lograr el ángulo.

4.2.3. Dirección de rotación del remo

Con todos los cálculos físicos hechos hasta ahora se sabe que velocidad de giro tiene el remo en cada fotograma, pero no se sabe cuál es el eje de rotación. Este eje no queda inmóvil ya que la fuerza no es ejercida en el mismo plano. La fuerza ejercida produce una rotación en el remo para que la pala siempre quede a una altura precisa de la superficie de agua, a lo que se le llamará eje dinámico. Por el contrario, está el eje estático, que teniendo una rotación en un plano inclinado al agua, la rotación del remo tendrá un recorrido en el cual la pala se irá sumergiendo hasta un máximo y vuelva a salir a la superficie.

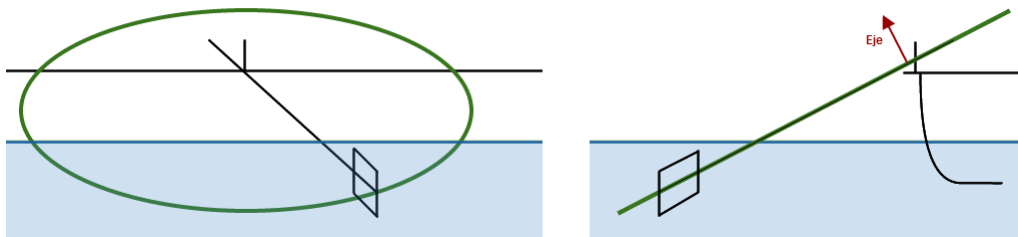


Figura 4.22: Vistas del recorrido del remo siendo el eje estático.

Es por esto que para una palada efectiva el eje de la rotación del remo debe de cambiar de dirección para mantener la pala en el plano horizontal. La rotación perfecta supondría que el eje de rotación fuese el vector vertical, manteniendo la altura óptima para cualquier momento de la palada.

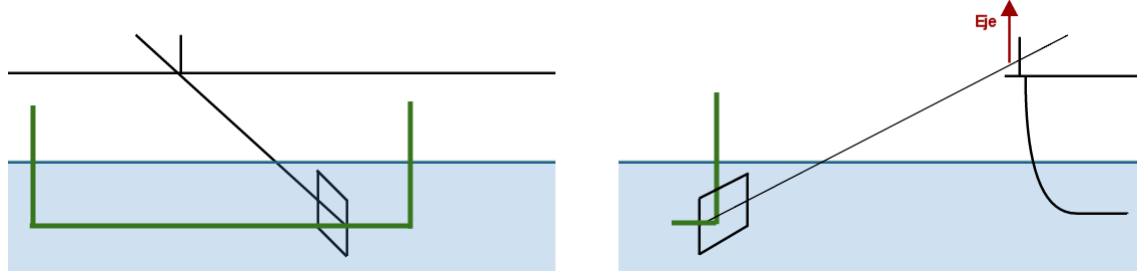


Figura 4.23: Vistas del recorrido siendo el eje perfecto.

Para poder lograr que el eje de rotación sea vertical, la fuerza debe de aplicarse en el plano horizontal en todo momento. Por lo que en la simulación la fuerza ejercida por el remero se ha predefinido para que siempre sea en este plano, simulado así la fuerza aplicada por un buen remero. Como en la vida real, en la simulación se deben definir los ángulos de entrada para poder aprovechar mejor la fuerza ejercida. El ángulo de salida puede depender de dos valores: que el remo llegue a un ángulo máximo de salida o que el tiempo para esa palada finalice, sea cual sea el ángulo. En caso de que la fuerza sea muy pequeña no recorre toda la distancia necesaria para llegar al ángulo máximo por lo que si lleva un ritmo de palada muy rápida obligará a sacar la pala antes. El ritmo de palada será un dato que se exige, y es medida por paladas por minuto (ppm).

4.2.4. Drag

Hasta el momento la única fuerza descrita que ejerza una fuerza en el remo es el remero, pero esta fuerza únicamente no traslada ninguna fuerza a la trainera como para producir movimiento. El movimiento de la trainera se basa en la resistencia de la pala en el agua. El área de la pala produce mucha resistencia al fluir en el agua ya que éste es denso, utilizando esta resistencia como punto de apoyo, la fuerza ejercida por el remero impulsa a la trainera. Es por esto que esta fuerza es necesaria para cualquier cálculo de la trainera. Esta fuerza es llamada drag (del inglés, “arrastre”), y es la fuerza o fricción que sufre un objeto cuando atraviesa un fluido. La forma de calcularla es ésta:

$$\vec{D} = \frac{1}{2} \rho v^2 C_D A$$

dónde,

- ρ , es la densidad del fluido,
- \vec{v} , es la velocidad del objeto relativa al fluido,
- A , es el área de la cara del objeto que atraviesa el fluido,

- C_D , es el Coeficiente de Drag, que es un valor que se debe de calcular experimentalmente.

El Coeficiente de Drag es un valor escalar que se usa para cuantificar la resistencia de un objeto en un entorno fluido, esto significa que la forma total del objeto produce cambios a la hora de calcular la fuerza del drag. Cuanto menos sea el valor de este coeficiente, más aerodinámico o hidrodinámico será y menos resistencia ofrecerá al fluido. Es un valor difícil de calcular ya que puede cambiar con la velocidad del fluido, pero ha sido calculado por Atkinson [3] y lo define así:

$$C_D = A_D * (\sin(a))^2$$

dónde,

- A_D , es la amplitud máxima de la pala, que en la pala plana (la pala más parecida a la utilizada en la trainera) es del valor de 2.0,
- a , es el ángulo que hay entre la dirección del fluido y el vector normal (perpendicular al área) de la pala, o en otras palabras, la inclinación de la pala.

Por lo que uniendo las dos ecuaciones lograremos la ecuación utilizada en la simulación:

$$\vec{D} = \frac{1}{2} \rho v^2 2(\sin(a))^2 A = \rho v^2 (\sin(a))^2 A$$

Esta fuerza es aplicada en el punto medio de la pala en dirección relativa del fluido respecto al objeto, por lo que será en la misma dirección que la fuerza aplicada al remo por la persona. Para calcular la fuerza total hay que calcular los momentos de fuerza, que al estar uno en cada lado del eje se contrarrestan. La fuerza ejercida del remero crea una aceleración angular en un sentido pero el drag siempre creará una aceleración en la dirección opuesta a ésta, por lo que el remo gira lentamente relativo a la trainera.

4.2.4.1. Densidad

En la simulación se han implementado dos tipos diferentes de drags que son el drag en agua y drag en el aire. Puesto que el agua es mucha más densa que el aire, la rotación local es mucho más lenta. En la simulación con el drag en el aire se ha utilizado la densidad del aire a la altitud del nivel del mar, que es 1,19 kg/m³. En cambio para el agua, se ha utilizado la densidad del agua de mar a una temperatura de 10°C, que es de 1027 kg/m³.

4.2.4.2. Velocidad

En el caso de la velocidad, se calcula la suma de la corriente y la velocidad tangencial, esto es así porque en caso de que el agua esté estancada (la velocidad sea 0) el agua no va hacia la pala sino que la pala es la que va hacia el agua. En este caso la velocidad que tiene la pala es la que lleva el fluido en contra de

la pala, que en este caso es la velocidad tangencial, pero al sentido contrario. La velocidad tangencial es la velocidad que tiene una partícula al rotar, esto es, que al contrario de la rotación angular, la velocidad tangencial sí varía con la distancia al eje, cuanto más lejos más velocidad tendrá. Como bien se aprecia en la definición:

$$\vec{v}_t = \vec{\omega} R$$

En caso de haber alguna corriente en el agua, éste sumaría la velocidad a este valor. Así, si la corriente y el contrario de la velocidad tangencial van en la misma dirección estos se suman. Dando la suma utilizada en la simulación:

$$\vec{v} = \vec{v}_t + \vec{v}_c$$

4.2.4.3. Área

La pala tiene forma de trapecio, esto es, la pala en su extremo es más ancha que en su interior. Las palas utilizadas en las competiciones de trainera no son completamente planas ya que tienen una zona más gruesa y resistente en el centro siguiendo la dirección de la caña, como si fuera la columna vertebral de la pala como se aprecia en la figura 4.24 y haciéndose más fino gradualmente hasta los bordes:



Figura 4.24: Pala utilizada en las competiciones de trainera.

Puesto que el área es lo que se necesita, la forma de ésta no aplica cambio en esta variable aunque si en el coeficiente de drag, aunque eso debería de descubrirse experimentalmente. Por lo tanto el área de esta superficie será la del trapecio que se calcula de la siguiente manera:

$$A = \frac{(B + b) \times h}{2}$$

siendo,

- B, la distancia en metros de la parte del exterior de la pala, o la distancia de la parte ancha de la pala,
- b, la distancia en metros de la parte interior de la pala, o la distancia de la parte estrecha de la pala,
- h, la longitud en metros de la pala, o la distancia entre la parte estrecha a la parte ancha.

4.3. Fases de la palada

4.3.1. Duración de la palada

Una vez sabida la fuerza que aplica el remero y el movimiento que eso conlleva en la rotación del remo, se debe saber durante cuánto tiempo o cuántos grados se ejercerán la palada. En la vida real cada entrenador según su propia experiencia elige una duración del recorrido. Pero en este caso el recorrido será una consecuencia del ritmo de palada. La razón de esto es la elección de una variable la que debería de ser definida por el usuario y la otra la consecuencia de esa elección. Por ejemplo, si se definen las dos variables, siendo la frecuencia 40 paladas por minuto (ppm) y el recorrido 90° , un remero con muy poca fuerza no llegaría a completar todo el recorrido, ya que necesitaría de mucha fuerza para poder llegar en tan poco tiempo a un ángulo muy largo. Es por eso que se ha decidido que el ritmo sea el valor a definir por el usuario y el recorrido sea consecuencia del tiempo para dar una palada y la fuerza que tiene el remero.

En el caso de que un remero sea demasiado fuerte o pueda fallar la palada creando una rotación demasiado rápida se definirá una rotación máxima en la que el remero deberá quedar inmóvil a la espera del resto de la tripulación para poder volver.

Otra variable a tener en cuenta es la diferencia de tiempo que hay entre el tiempo que la pala está en el agua y la que está volviendo a la postura inicial. En general la velocidad que transcurre en el aire depende de la velocidad que se lleva en el agua, si se rema rápidamente la vuelta suele ser rápida. Tanto en ritmo de regata como en velocidad de crucero el tiempo ejerciendo fuerza suele ser un poco menor al de vuelta, y ese valor se ha implementado definiendo un porcentaje de la palada en la que se tiene la pala en el agua. Por ejemplo, si una palada dura 1 segundo y el porcentaje de la diferencia es 50% el tiempo ejerciendo fuerza será la misma que volviendo, 0.5 segundos, pero si cambiamos el porcentaje a 25%, la pala estará en el agua 0.25 segundos contra los 0.75 segundos que estará en el aire.

Con todos estos datos logramos saber la duración de una palada y el tiempo que conlleva cada movimiento, pero para entenderlo mejor es imprescindible saber lo que son la frecuencia y el periodo.

4.3.1.1. Frecuencia y periodo

La frecuencia (f) es una magnitud que mide el número de repeticiones por unidad de tiempo de cualquier fenómeno o suceso periódico. En el sistema internacional la frecuencia se mide en hercios (Hz) que es la cantidad de repeticiones en un segundo:

$$1Hz = \frac{1}{s}$$

La frecuencia viene de la mano siempre con el periodo, que es lo inverso. El periodo (T) es el tiempo transcurrido para poder una repetición completa, y se mide en segundos (s). Por lo que la definición matemática entre estos dos valores es:

$$f = \frac{1}{T} \text{ ó } T = \frac{1}{f}$$

Una vez descubierta esta definición el descubrimiento de la duración de cada palada o periodo es muy simple, pero no sin antes adecuar los valores al sistema internacional.

La forma de definir la frecuencia en el remo es el ppm o paladas por minuto y la medición del sistema internacional son las repeticiones por segundo por lo que hay que pasar de minutos a segundos:

$$1ppm \times \frac{1pps}{60ppm} = \frac{1}{60}pps$$

Con esta conversión ya estamos listos para saber el tiempo que se invierte en cada palada dependiendo de la frecuencia de la palada, que por ejemplo a una frecuencia de 30 ppm sería:

$$f = 30ppm = \frac{30}{60}pps = 0,5pps \implies T = \frac{1}{f} = \frac{1}{0,5} = 2s$$

Así, con la duración total de la palada sólo debemos distinguir la duración del tiempo en la que el remero rema y en la que está volviendo, que como bien hemos dicho antes es determinado por un porcentaje. Si al ejemplo anterior le añadimos el porcentaje de 40 % en la que se está ejecutando la remada, logramos la duración de ésta, que será:

$$2s \times 40\% = 0,8s$$

y por lo tanto el tiempo que dura la retirada de la parada podría calcularse de diferentes manera, ambas computacionalmente fáciles:

$$2s \times (100 - 40)\% = 1,2s \text{ ó } 2s - 0,8s = 1,2s$$

Con todos los datos necesarios descubiertos ya se puede deducir en qué momento de la palada está en cada instante, pero para eso se necesita un contador de tiempo ininterrumpido. Así, cada segundo equivaldría a alguna de las fases

de la palada. Como en la vida real la palada comienza en el ataque (con la pala sumergida), el punto de inicio se ha definido igualmente, por lo que la fase de cada segundo se puede lograr con una simple condición:

Si el resto de la división entre el tiempo total y el periodo, esto es,

$$t_{Total} \bmod T$$

es más pequeño que el tiempo de duración de la pala en el agua, esto es,

$$T \times \frac{\text{Porcentaje}}{100},$$

entonces ésta estará en la fase de remada, y si por lo contrario no cumple esa condición la fase sería la de retirada.

4.3.2. Velocidad de la fase de vuelta

Para este proyecto se contó con la ayuda del entrenador del equipo Orio Arraunketa Elkartea[16], Igor Makazaga. Con su ayuda se despejaron incógnitas que un entendido en las traineras sabe, entre otras la diferencia de las fases de la palada, información detallada de las distancias y pesos estándares, la velocidad de vuelta relativa a la palada, e incluso la ayuda en corrección de errores.

Volviendo al proyecto, sabemos el tiempo que dura cada fase, entre ellas la fase de vuelta o retirada de la palada. En la vida real un remero debe de llegar en el tiempo exacto hasta el ángulo de ataque, ya que si llega tarde, el recorrido de la palada será menor, con su consiguiente pérdida de fuerza ejercida y la menor velocidad para la trainera, y por lo contrario, si llega demasiado pronto al ángulo de ataque, éste deberá esperar a que el resto de tripulación llegue a este punto de la palada. En este caso la pérdida de velocidad para la trainera se debe a que el levantamiento del tronco es más brusco y esto crea una fuerza contraria a la de la dirección de navegación de la trainera, es por eso que cuanto más regular sea el levantamiento del tronco del remero menos fuerza negativa se crea.

En la simulación se ha optado por hacer a todos los remeros perfectos en este sentido ya que la fuerza negativa de los pesos de los remeros y sus inercias no ha sido implementada aún. Además el cálculo de la velocidad de rotación del remo se hace mucho más simple, ya que tenemos el tiempo que debe de durar la retirada y el ángulo hasta el punto de ataque. Con estos datos logramos la velocidad media utilizando simplemente la definición de velocidad angular:

$$\omega = \frac{d\theta}{dt}$$

Poniendo como ejemplo dos paladas diferentes que con la fuerza ejercida han rotado 30° y 60° respectivamente, y siendo el periodo el mismo del ejemplo utilizado anteriormente, 2 segundos con una diferencia del 40%. Tendremos dos velocidades angulares diferentes, pero el momento en el que llegarán al punto

de ataque será el mismo. Por lo que la velocidad angular de la vuelta de cada remero será:

$$\omega_{Débil} = \frac{30^{\circ} \times \frac{\pi}{180}}{1,2s} = 0,43rad/s$$

$$\omega_{Fuerte} = \frac{60^{\circ} \times \frac{\pi}{180}}{1,2s} = 0,87rad/s$$

Como la distancia recorrida es el doble, lógicamente la velocidad necesaria para llegar al mismo punto en una cierta cantidad de tiempo debe ser el doble, por lo que se cumple lo esperado.

4.4. Ataque y final

El ataque de una palada es cuando la pala se introduce en el agua y lógicamente el final es cuando la pala se extrae del agua. En la vida real el ataque simplemente se consigue dejando caer la pala por su propio peso ayudándose de la gravedad o aplicando una leve fuerza con las manos hacia arriba haciendo así que la pala baje, aunque en el caso del final el remero es el que levanta el remo con su propia fuerza. Esta fuerza no ha sido calculada, y puesto que no afecta mucho en la palada se han despreciado dichos datos. Aunque para la física no tenga mucho efecto, para la parte visual o gráfica sí que lo tiene ya que ver la pala volver mientras la pala está sumergida no tiene mucho sentido, por lo que por arreglar este sentido se han hecho unos retoques. A la hora de la remada se ha definido un ángulo en el que la pala quede sumergida en su totalidad, pero sin hundirse mucho en el agua (en la figura 4.25 se ha disminuido el ángulo de ataque un poco para poder ver la profundidad de la pala).

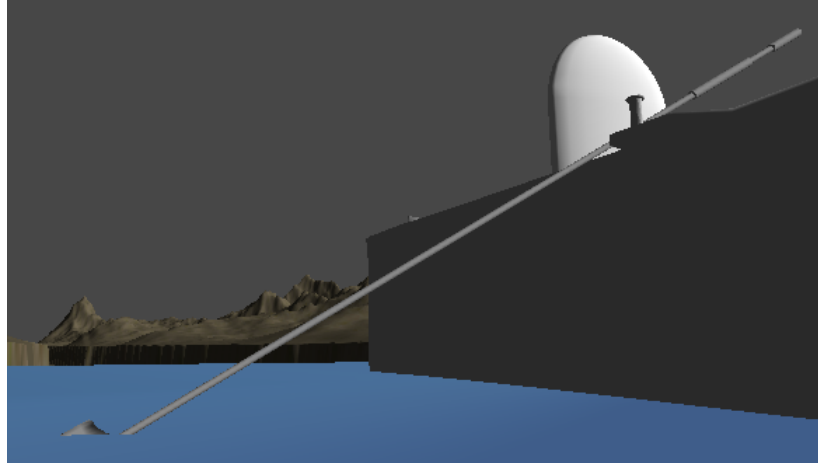


Figura 4.25: Profundidad de la pala en el ataque.

Una vez iniciada la fuerza que aplica el remero, este ángulo deja de hacer efecto dejando al ángulo moverse libremente, aunque por la fuerza ejercida por el remero y la falta de la elevación producida por la inclinación del ataque hacen que se mantenga casi en el mismo ángulo.

En la vuelta también se ha predefinido, pero en este caso el ángulo debe de ser menor para que la pala quede fuera del agua, como se puede apreciar en la figura 4.26.

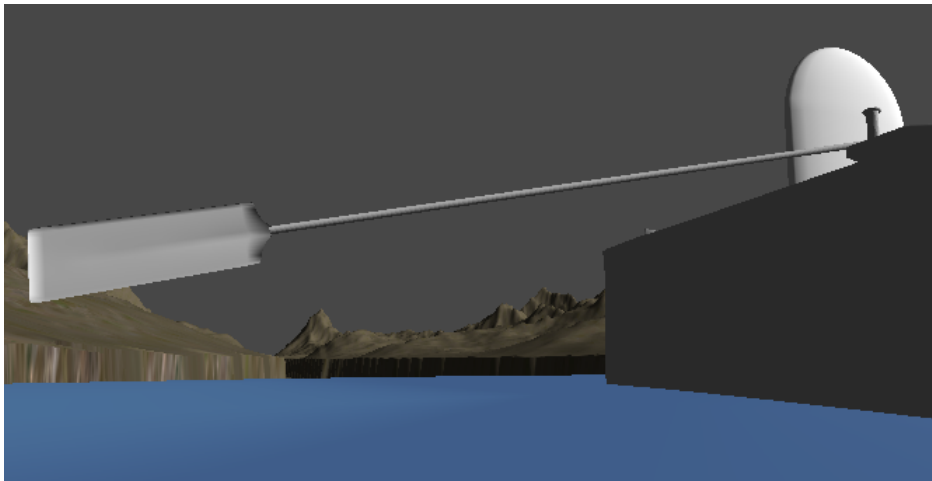


Figura 4.26: Ángulo del remo en la vuelta.

4.5. Movimiento de la trainera

Cada remero aporta su parte para poder hacer avanzar la trainera y para poder transferir la fuerza del remo a la trainera se utiliza la fórmula de la palanca.

4.5.1. Palanca

Una palanca es una máquina simple cuya función es transmitir fuerza y desplazamiento. La palanca se compone de tres componentes, potencia (P), resistencia (R) y fulcro (F). Como para nuestro caso particular lo importante es la fuerza que se ejerce el fulcro o punto de apoyo, por lo que en el caso de un remo es así:

$$F = P + R$$

Y la representación gráfica de esto es:



Figura 4.27: La mecánica de la palanca en el remo.

Con esta simple formula se ha logrado transmitir la fuerza del remero a la trainera que exactamente es la fuerza del remero y la del drag en la dirección contraria. Aunque hay que mejorar la transmisión de esta fuerza se ha conseguido el primer paso para tener una velocidad producida por los remeros. Esta velocidad se consigue sumando la fuerza de todos los remeros, y al igual que el remo se ha calculado el momento logrando así una aceleración. Puesto que no hay gravedad, la trainera tiende a ir hacia arriba, se toma como válido ya que una trainera cabecea cuando avanza. Al haber un remero más por una banda que de otra tiende a moverse al lado contrario de la banda en la que más remeros hay. En la vida real esto se contrarresta con el patrón y su remo, pero que aún no se han implementado.

4.6. Cronómetro

En una regata el valor a minimizar es el tiempo, ya que cuanto menos tiempo te lleve completar el recorrido más posibilidades tienes de ganar. Por esta razón la necesidad de contabilizar el tiempo es necesaria, por eso la creación de un cronómetro. Unity 3D te da la opción de saber el tiempo en segundos desde el inicio de la aplicación, con esa ayuda basta con unos simples cálculos matemáticos para crear el cronómetro:

$$fraccion = (tiempo \times 100) \text{ mod } 100$$

$$segundos = tiempo \text{ mod } 60$$

$$minutos = \frac{tiempo}{60}$$

Una vez teniendo todos los valores no hay más que presentarlos en el formato adecuado para que todo el mundo lo entienda, como es el estándar de un cronómetro:

$$mm : ss : fff$$

siendo,

- m, los minutos,
- s, los segundos,
- f, la fracción de segundo.

Capítulo 5

Uso de datos

5.1. Datos despreciados en la física

En una simulación de cualquier entorno real hay múltiples variables a identificar ya que la suma de todas las variables es la que determina un comportamiento de cualquier objeto, pero hay algunas variables que influyen poco en el comportamiento de un objeto por lo que en la física se suelen despreciar, dejándolos fuera de todo cálculo. En este proyecto se han despreciado algunas variables que pueden implementarse en un futuro o por la complejidad y el poco aporte al comportamiento queden sin implementar. Los datos despreciados han sido estos:

- Centro de rotación del remo durante la palada.
- Fricción del estrobo al remo.
- Drag en la caña.
- Fuerza ejercida durante el ataque y el final.
- Área real de la pala. Falta por sumar al área la unión de la pala a la caña.
- El coeficiente real de drag de una pala real de trainera.
- Elevación creada por la inclinación de la caña. La pala si se inclina demasiado hacia delante produce una elevación pudiendo hacer que la pala salga de agua y fallar la palada, y a la inclinación contraria sumergir demasiado la pala y en consecuencia endurecer la palada. Algo parecido ocurre si la pala no se sumerge lo suficiente, que termina emergiendo completamente y quedarse sin punto de apoyo.
- Repaleo: Al acabar la palada en la vida real por la anatomía propia de la mano y el exceso de rotación produce una inclinación de la pala. Este fenómeno es aprovechado para girar horizontalmente la pala una vez acabada la palada y no frenar el bote por el viento.

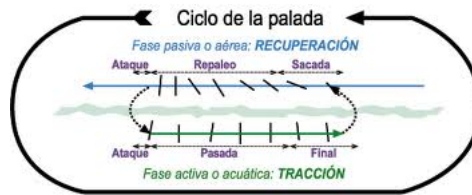


Figura 5.1: Ciclo completo de la palada.

5.2. Datos utilizados como entrada

En el simulador hay diferentes valores que se pueden cambiar para así crear una simulación personalizada, cada valor tiene diferentes reacciones en el simulador, por lo que para un uso más fácil se definirán todos ellos:

- El remero
 - Peso: El peso del remero. Este valor influye en el peso total de la trainera.
 - Mano exterior: Colocación de la mano exterior (la más cercana al tolete). Cantidad de metros desde el extremo de la tiborta. Valor para calcular el punto donde se ejerce la fuerza.
 - Mano interior: Colocación de la mano interior (la más cercana a la tiborta). Cantidad de metros desde el extremo de la tiborta. Valor para calcular el punto donde se ejerce la fuerza.
 - Ángulo de ataque: El ángulo en el que empieza el ataque. Siendo el valor "0" la perpendicular a la trainera y siendo a positivo hacia la proa.
 - Inclinación de ataque: Inclinación de la pala en el ataque. El valor influye en la fuerza de drag, siendo el ángulo "0" completamente vertical.
- La trainera
 - Peso: El peso de la trainera. Este valor influye en el peso total de la trainera.
 - Diferencia de la palada: Diferencia entre el tiempo que la tripulación transcurrirá con el remo en el agua y el aire. El valor asignado será el porcentaje de la palada completa en la que la pala estará en el agua. El valor influye en el tiempo que el remero aplica fuerza.
 - Paladas por minuto: Cantidad de paladas que se dan por minuto. El ritmo influye en el tiempo que el remero ejerce fuerza.
- El remo

- Distancia: Distancia en metros desde la tiborta hasta la pala. Este valor influye en la palanca del remo.
 - Distancia al tope: Distancia en metros desde la pala al punto de rotación (tope). Este valor influye en la palanca del remo.
 - Distancia de la caña: Distancia en metros desde la tiborta hasta el inicio de la pala. Este valor se usa para el cálculo del momento de inercia.
 - Distancia de la pala: Distancia en metros de la longitud de la pala. Este valor se usa para el cálculo del momento de inercia.
 - Ancho del exterior de la pala: Distancia en metros de la parte más lejana de la pala. Este valor se usa en el cálculo del área de la pala.
 - Ancho del interior de la pala: Distancia en metros de la parte más cercana a la tiborta de la pala. Este valor se usa en el cálculo del área de la pala.
 - Peso de la caña: Peso en kilogramos desde la tiborta hasta el inicio de la pala. Este valor se usa en el cálculo del momento de inercia.
 - Peso de la pala: Peso en kilogramos de la pala. Este valor se usa en el cálculo del momento de inercia.
 - Ángulo máximo de la palada: Ángulo máximo que podrá recorrer una palada desde el ataque hasta el final. El ritmo influye en el tiempo que el remero ejerce fuerza.
- El entorno
- Vector de viento: La dirección y la fuerza del viento representada por un vector. Este valor influye en el drag cuando la pala está fuera del agua.
 - Vector de corriente: La dirección y la fuerza de la corriente representada por un vector. Este valor influye en el drag cuando la pala está en el agua.

Capítulo 6

Trabajo futuro

En el futuro hay una gran cantidad de variables a introducir o mejoras en lo ya creado y entre otros trabajos futuros tenemos:

- Introducción de gravedad.
- Implementar la flotabilidad.
- La elevación de la pala: La pala cuando se inclina produce una fuerza de inclinación que es muy parecida al drag, es éste el valor que se busca aumentar en el caso de los aviones, aunque en el caso del remo se busca más el drag que la elevación, al contrario que en la aviación. Como todavía queda la gravedad y la dirección de la fuerza del remero por introducir en el sistema, la pala terminaría saliendo del agua y no quedaría reflejado con la realidad.
- La mejora de la fuerza que en cada instante le aplica el remo a la trainera haciendo que éste también se desplace.
- La obtención de una representación fiable de la fuerza ejercida por el remero.
- La dirección en la que se ejerce la fuerza al remo, para mantener una misma altura de la pala en el agua.
- Rozamiento de la trainera con el agua.
- Calcular el centro de gravedad de la trainera.
- Pesos de los remeros y su posición o postura, cambiando notablemente el centro de gravedad lo que puede producir alteraciones en los botes.
- El efecto en la dureza de la palada dependiendo de la profundidad del agua.
- El efecto del viento.

- La variación del peso cuando hay lluvia.
- El movimiento de la trainera y el agua en su interior que oscila produciendo fuerzas por el cambio de peso.
- Introducción de las olas, esto conlleva diferentes roces con el agua al no ser uniforme y el cambio de velocidad al coger una ola.
- El efecto del remo de patrón.
- La contribución del patrón en el movimiento de la trainera, evitando escorarse.
- Paladas iniciales o salidas.
- La ayuda en el giro del remo de proa.
- El giro de las ciabogas en general.
- La introducción de agua en la trainera por culpa del oleaje.
- La ayuda para evitar el punto anterior utilizando diferentes objetos.
- La variación en el escoraje por razón de los remeros.
- La diferencia en la dureza de la palada dependiendo del sitio en el que reme.
- Creación de la interfaz para facilitar los cambios de las variables durante la ejecución.
- Mejoras en lo gráfico como son los materiales y texturas.

Este trabajo es un proyecto de tesis de máster. Un simulador de trainera que tenga en cuenta todo lo que se ha mencionado en este apartado requeriría un tiempo de programación e investigación muy superior a un proyecto de tesis de máster, podría considerarse como una tesis doctoral completa y pensamos buscar vías de financiación para la realización de dicha tesis.

Bibliografía

- [1] Dudhia A. The physics of rowing. *www.atm.ox.ac.uk/rowing/physics*, 2001.
- [2] Android. *www.android.com*.
- [3] William C. Atkinson. Modeling the dynamics of rowing. *www.atkinsophd.com/row/rowrpage.htm*, 2002.
- [4] BioShock. *www.2kgames.com/bioshock*.
- [5] Blender. *www.blender.org*.
- [6] Blender-Models. *www.blender-models.com*.
- [7] Bert Blocken. *Sports and Building Aerodynamics*. Eindhoven University of Technology, *www.coursera.org/course/spobuildaerodynamics*.
- [8] Creative Commons. *es.creativecommons.org*.
- [9] Concept2. *www.concept2.com*.
- [10] Coursera. *www.coursera.org*.
- [11] Asociación de Clubes de Traineras. *www.ligasanmiguel.com*.
- [12] Bandera de la Concha. *www.donostiasansebastian.com/regatas*.
- [13] Ed Johnsen Dr. Michael Dubson, David H. Lieberma. *Physics 1 for Physical Science Majors*. University of Colorado Boulder, *www.coursera.org/course/physics1*.
- [14] Newton Dynamics. *www.newtondynamics.com*.
- [15] Mass Effect. *masseffect.bioware.com*.
- [16] Orío Arraunketa Elkartea. *www.orio-ae.com*.
- [17] Unreal Engine. *www.unrealengine.com*.
- [18] Epic Games. *epicgames.com*.
- [19] Apple iPhone. *www.apple.com/es/iphone*.

- [20] Linux. *www.linux.com*.
- [21] Andrea Mola Luca Formaggia, Edie Miglio and Antonio Montano. A model for the dynamics of rowing boats. *International journal for numerical methods in fluids*, (DOI: 10.1002/fld.1940), 2008.
- [22] Nvidia. *www.nvidia.es*.
- [23] Ogre3D. *www.ogre3d.org*.
- [24] Apple Mac OS. *www.apple.com/es/osx*.
- [25] Bullet Physics. *bulletphysics.org*.
- [26] Nvidia PhysX. *www.nvidia.es/object/nvidia-physx-es.html*.
- [27] Sony PlayStation. *es.playstation.com*.
- [28] Sander Rosendaal. A model of rowing. *http://sanderroosendaal.wordpress.com*, 2011.
- [29] Sony. *www.sony.es*.
- [30] Unreal Tournament. *www.unrealtournament.com*.
- [31] Unity3D. *unity3d.com*.
- [32] van Holst M. On rowing. *http://home.hccnet.nl/m.holst/RoeiWeb.html*, 2004.
- [33] Gears Of War. *gearsofwar.xbox.com*.
- [34] Nintendo Wii. *www.nintendo.com/wiiu/es*.
- [35] Microsoft Windows. *windows.microsoft.com*.
- [36] Microsoft Xbox. *www.xbox.com/es-es*.

Apéndice A

Notas de programación

A.1. Jerarquía de los objetos

La organización de los objetos tridimensionales también tiene que ver en la programación ya que si hay una jerarquía entre los objetos significa que un objeto se mueve dependiendo de lo que se mueva el que esté sobre el objeto jerárquicamente o “padre”. Por lo que en nuestro caso, si la trainera se mueve, eso significa que la gente que está sentada en ellos deben de moverse lo mismo que la trainera. Por eso, la relación jerárquica que se ha definido ha sido ésta:

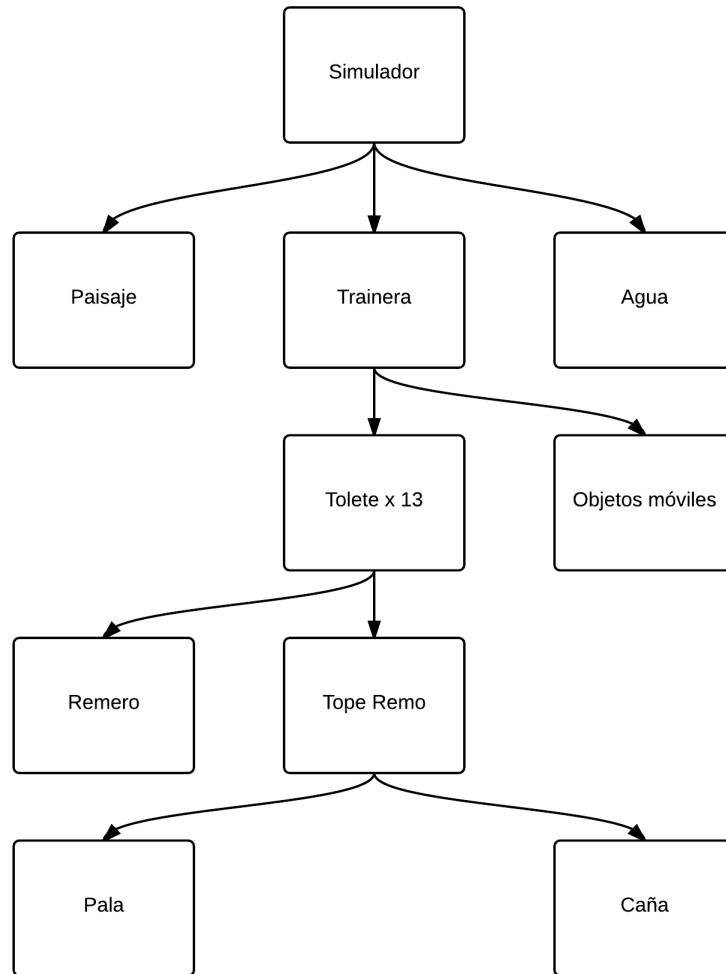


Figura A.1: Representación de la jerarquía de los objetos tridimensionales.

Dentro del simulador, o lo que sería el mundo virtual, se ha definido el paisaje como lo más alto en la jerarquía junto con la trainera ya que no dependen de ningún otro movimiento para saber su posición. Dentro de la trainera se han colocado los objetos móviles como son los toletes y tostas que se pueden calibrar para tener un mejor rendimiento a la hora de lograr la velocidad de la trainera. Dependiendo dónde esté la tosta se coloca el remo, por lo que depende del tolete para saber su posición final. En lugar de colocar el remo se establece la posición del tope y una vez también definida la longitud la que tendrá la palanca del remo se coloca la caña y la pala.

A.2. Códigos

En Unity3D se debe de aplicar cada código de programación a un objeto por lo que cada objeto tendrá su bloque de programación asignada a él. Aunque son independientes entre ellos hay posibilidad de conseguir datos del resto de códigos. Por esto es fácil definir los códigos ya en ellos se procesa lo necesario para ese objeto. Así, el resultado ha sido éste:

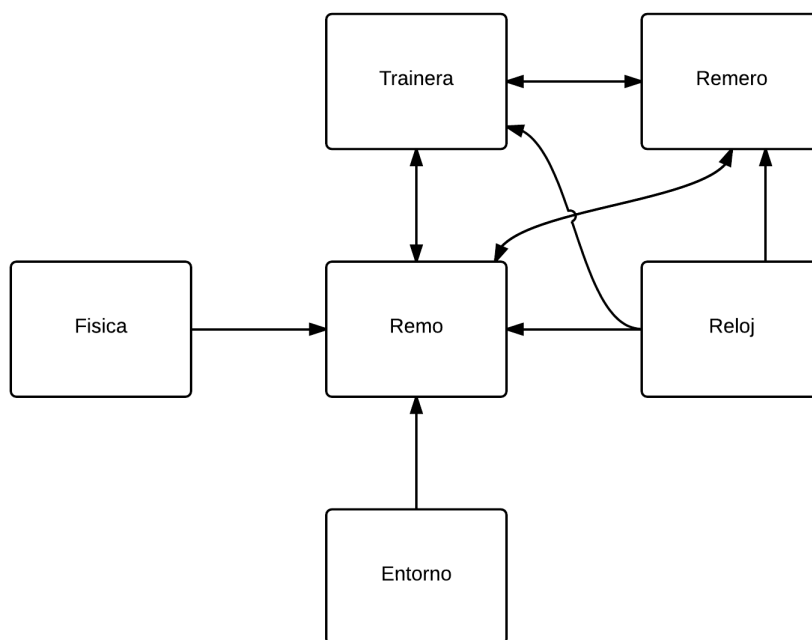


Figura A.2: Relación entre los diferentes bloques de código.

Se puede apreciar que hay algunos módulos que necesitan la información de otros módulos para poder hacer sus cálculos y otros que simplemente están para servir de datos a otros. A continuación se definen cada módulo y su uso:

- Entorno: En este módulo sólo se guardan dos valores, que son vectores de viento y corriente.
- Física: Es un bloque abstracto ya que no se usa para mover o rotar objetos, solamente tiene funciones de la física descrita en temas anteriores para poder ser utilizadas por cualquier objeto, aunque de momento sólo el módulo “Remo” es el que hace uso de estas funciones. Al no usar transformaciones se puede asociar a cualquier objeto, y como el único objeto obligatorio en cualquier escena es la cámara se le ha asignado a éste. Las funciones que tiene este módulo son:

- **MomentoFuerzaResultante:** Calcula el momento de fuerza resultante en un objeto con diferentes fuerzas aplicadas en él.
 - **Momento2Radian:** Del momento de fuerza resultante calcula cuantos grados rota el objeto con la fuerza ejercida.
 - **MomentoInerciaVarilla:** El cálculo del momento de inercia de la varilla recibiendo las medidas del objeto.
- **Reloj:** Este módulo contiene el cronometro y todo lo correspondiente al tiempo. El uso más importante lo hace el bloque “Remero” para calcular las paladas, ya que las fases de ésta dependen del tiempo.
 - **Remero:** Aquí es donde se definen la mayoría de las variables que se definen para ver los diferentes resultados en la simulación. Estas deben de ser introducidas manualmente ya que aún no se ha programado la interfaz para facilitar los cambios, o incluso hacerlos mientras se ejecuta. En este módulo sólo se ha implementado una función:
 - **Palada:** Esta función decide, dependiendo del tiempo, si el remero está en la fase de tirar o de recuperación. Cuando se ha decidido la fase, llama al módulo denominado “Remo” para que haga los cálculos necesarios y rote la cantidad equivalente.
 - **Remo:** Es el módulo más importante hasta la fecha por su cantidad de código implementado. Este bloque se encarga de calcular el drag y otros factores para saber cuál es el movimiento para el próximo fotograma. Y las funciones más importantes de este bloque son:
 - **Drag:** Es una función en la que se calcula la ecuación de drag. A esta función se le llama de otras funciones menos importantes, que simplemente hacen esta llamada con diferentes densidades, que se llaman DragAire y DragAgua.
 - **Tirar:** Calcula todo lo necesario para poder rotar la pala una cantidad de grados, entre otros cálculos están el drag, momento de fuerza, ángulo máximo, etc...
 - **Volver:** Se calcula la velocidad que se debe de tener para poder volver al ataque en el momento exacto.
 - **Trainera:** Éste es el modulo que, al ser jerárquicamente el más superior, hace llamadas al resto. En Unity3D hay funciones especiales que trae el motor gráfico, como por ejemplo la función “Update” que se ejecuta en cada fotograma, y es ahí donde hace todas las llamadas para poder calcular todo lo necesario para que la imagen sea acorde a la realidad. Hay otra función predefinida en Unity3D que se ejecuta una vez antes de iniciarse, es por esto que es el mejor sitio para poder inicializar todas las variables.
 - **Mover:** Ésta es la función que se ejecuta una vez por fotograma y encadena una serie de llamadas en distintos módulos que crean el movimiento deseado.

A.3. Flujograma

Viendo cómo un módulo es el que encadena todo se podría crear un flujograma para ver gráficamente los cálculos de un simple fotograma:

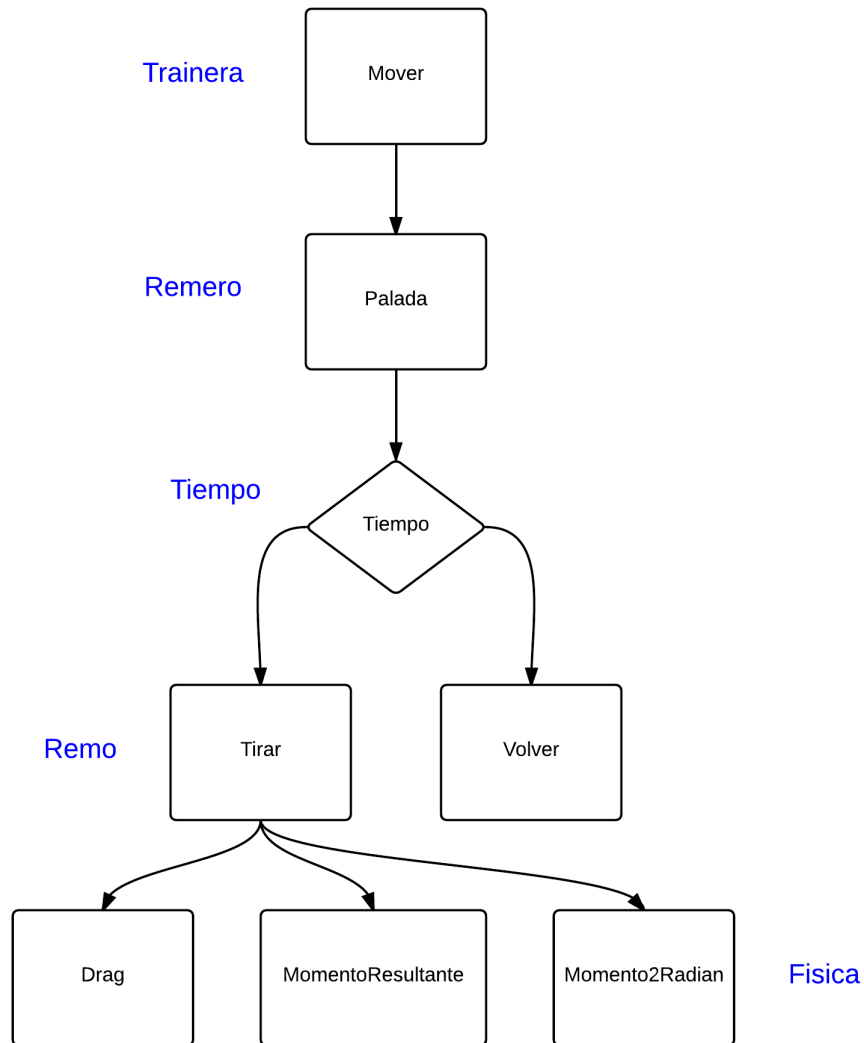


Figura A.3: Flujograma.

Se ha definido así para que venga concordando con la jerarquía de los objetos que existe. Así, las funciones del módulo “Trainera” estaría por encima del resto,

aunque intercambiando “Remo” y “Remero”.

A.4. Cambios de variable

Lo importante de un simulador es poder cambiar los valores de objetos para ver cómo eso influye en el sistema completo, es por eso que se ha habilitado unos menús en el propio Unity3D (Fig. A.4). Los cambios realizados en este menú se reflejarán en la simulación en tiempo real aunque el simulador esté ejecutándose. La parte mala de utilizar esto, es que una vez creado el simulador este menú no quedaría visible ya que es parte de Unity3D y no parte del proyecto por lo que la incorporación de unos menús similares en el simulador será un trabajo para el futuro.



Figura A.4: Menús para poder cambiar valores de las variables.

Apéndice B

Resultado

Una vez terminado el trabajo de este proyecto de fin de máster, se ha logrado simular la rotación local del remo en la mayor parte, ya que aún hay algunos factores que también influyen en la velocidad de rotación como puede ser la propia velocidad de la trainera. A continuación se expondrán unas imágenes del producto resultante:



Figura B.1: La simulación de la trainera en el ataque.

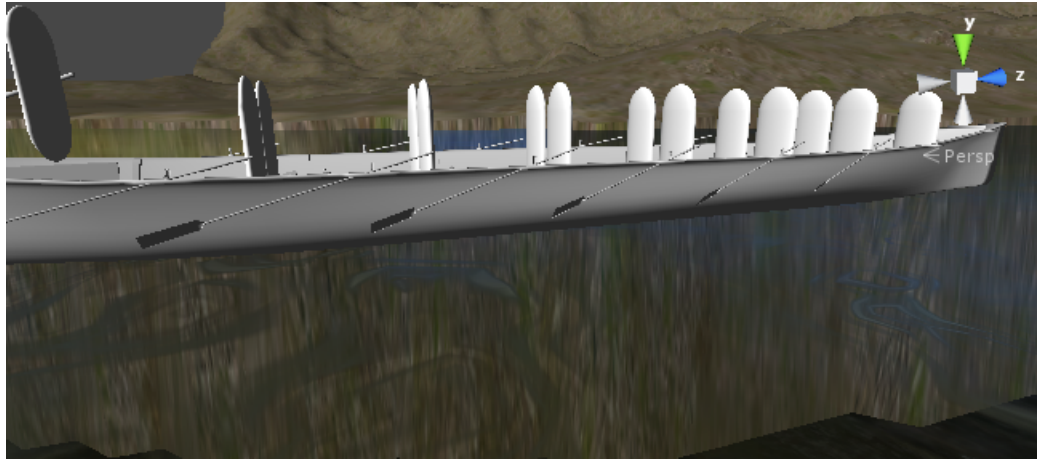


Figura B.2: La simulación desde el editor en el final.

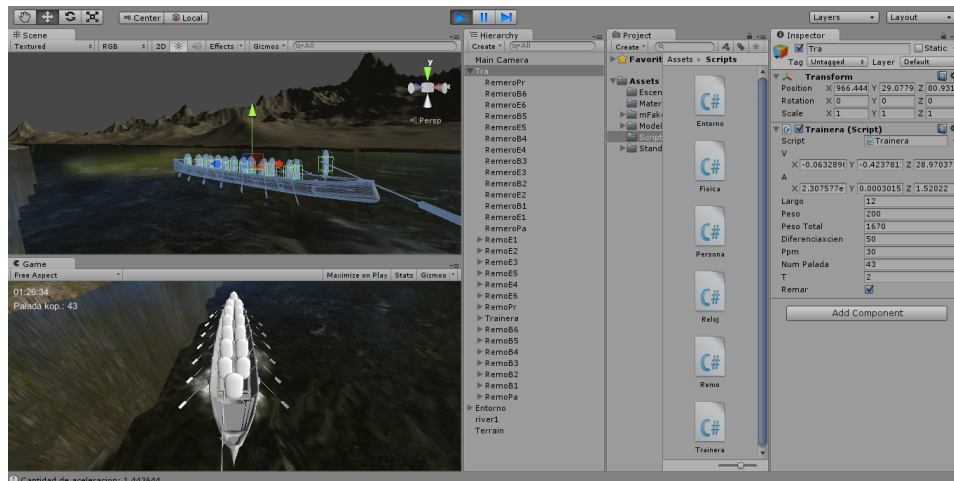


Figura B.3: La vista del editor durante una simulación.

Apéndice C

Terminología

- Carel: Borde superior de una embarcación pequeña, la cual rodea todo el bote.
- Cuadernas: Refuerzos transversales, parecidos a las costillas humanas.
- Cuerno: Soporte para hacer la maniobra de la ciaboga que se sitúa en la aleta de estribor (parte delantera derecha a ojos del patrón), también llamada oreja.
- Cremalleras: Pieza dentada en la que se fijan las tostras. La fijación puede hacerse en diferentes dientes por lo que los hacen móviles, personalizado para cada remero.
- Tiborta: Parte del remo situada en la punta donde se colocan las manos a la hora de aplicar fuerza.
- Tolete: Pieza vertical situada en el carel, en la cual se introducen y fijan el estrobo y remo para transmitir la fuerza del remero al bote.
- Toleterera: Parte en la que se apoya el tolete. Es la parte más ancha en el carel, en el caso de las traineras.
- Tosta: Pieza de madera o aluminio en el que el remero apoya la parte baja de la espalda y los pies.
- Trainera: Embarcación a remo utilizada para competir en regatas.

Apéndice D

Codificación en Unity3D

Como se ha dicho anteriormente, los códigos son asignables a cada objeto de la escena, pero no es lo único que asignable a los objetos, ya que Unity3D ofrece de una forma fácil asignar todo lo necesario para crear un juego, como son la física, materiales, efectos, etc... Esto lo hace un sistema fácil de usar e intuitivo. Los elementos asignables se llaman complementos y cada objeto puede tener infinidad de complementos.

Los códigos asignables tienen unas funciones predefinidas, cada una con su uso y función que se va a explicar a continuación:

- **Awake:** función que se llama antes de representar los objetos en 3D. Se usa para inicializar datos que no tengan repercusión con la escena.
- **Start:** al igual que Awake se llama una única vez al iniciar el mundo pero en este caso se pueden inicializar datos que usen objetos en 3D.
- **Update:** esta función es llamada cada fotograma, por lo que es dónde se implementa la mayoría del código.
- **FixedUpdate:** tiene la misma función que Update pero en esta función el tiempo entre fotogramas se define, lo que hace un sitio perfecto para implementar la física de la escena.
- **OnGUI:** Esta función se llama una vez por cada fotograma y se usa para renderizar interfaces y para eventos que estos

Hay muchas más funciones y cada una tiene su función y momento de ejecución, aunque las anteriormente descritas son las únicas utilizadas en el proyecto, para más detalles:

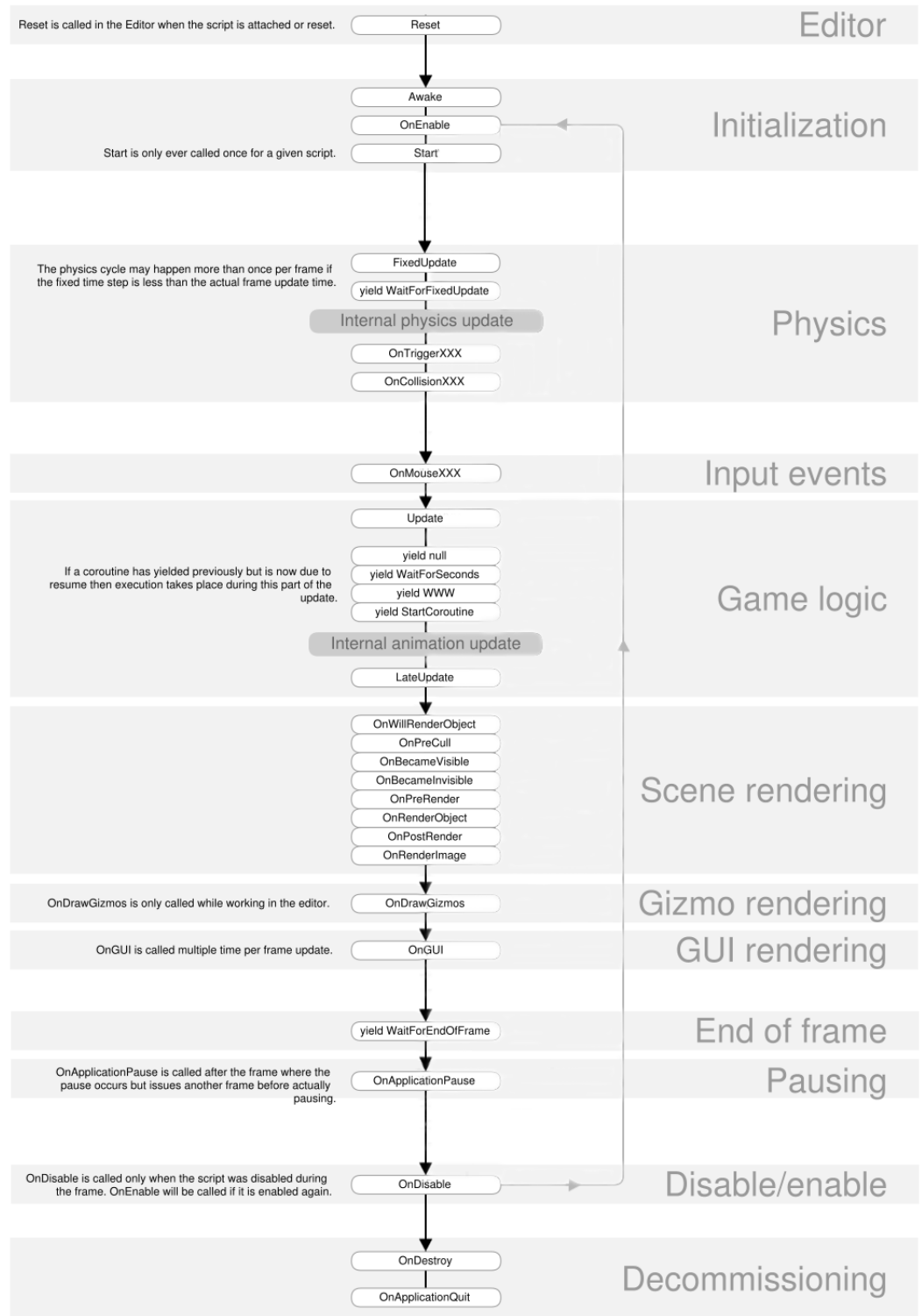


Figura D.1: Secuencia de ejecución de cada función.

```

void Update () {

    Reloj reloj = GameObject.Find ("Main Camera").GetComponent<Reloj> ();
    if (Input.GetKeyDown (KeyCode.Space)) {
        this.remar = !this.remar;
        reloj.tInicio = Time.time;
    }
}

// Update is called once per frame
void FixedUpdate () {

    Reloj reloj = GameObject.Find ("Main Camera").GetComponent<Reloj> ();

    if (this.remar) {

        float t = reloj.t;|
        this.numPalada = (int)(t / this.T);
        if (this.numPalada != this.aurreko)
            Debug.Log ("***** " + this.numPalada + ". Palada *****");
        this.aurreko = this.numPalada;
        this.mover();
    }
}

void OnGUI() {
    GUI.Label(new Rect(10, 25, 100, 20), "Palada kop.: " + this.numPalada.ToString());
}

```

Figura D.2: Código escrito del módulo Trainera.

Como ejemplo podemos apreciar funciones implementadas en el módulo trainera:

Aquí se puede apreciar que la función Update se utiliza solo para pausar la simulación, en cambio FixedUpdate, después de calcular la palada en la que está, llama a la función mover que es la que inicia la secuencia de cálculos para predecir la rotación. En el fondo de la imagen D.2 se utiliza la interfaz con la función OnGUI para hacer saber al usuario en la palada en la que se encuentra la simulación.