Un Simulador Ludificado de Instructor de Vuelo para Practicar el Entrenamiento Basado en Evidencia

A Gamified Flight Instructor Simulator to Practice Evidence-Based Training



Trabajo de Fin de Grado
Curso 2021–2022

Autor
Antonio Jesús Guerra Garduño
Javier Cruz López de Ochoa

Director
Federico Peinado Gil
Rubén Dapica Tejada

Grado en Desarrollo de Videojuegos
Facultad de Informática
Universidad Complutense de Madrid

Un Simulador Ludificado de Instructor de Vuelo para Practicar el Entrenamiento Basado en Evidencia

A Gamified Flight Instructor Simulator to Practice Evidence-Based Training

Trabajo de Fin de Grado
Departamento de Informática

Autor

Antonio Jesús Guerra Garduño
Javier Cruz López de Ochoa

Director

Federico Peinado Gil
Rubén Dapica Tejada

Convocatoria: *Junio 2022*

Grado en Desarrollo de Videojuegos
Facultad de Informática
Universidad Complutense de Madrid

30 de mayo de 2022

Agradecimientos

Agradecemos a David Rioja y Antonio Muñoz sus consejos sobre el simulador y el permitirnos utilizar los recursos e instalaciones de su empresa, Didactoons Games.

Agradecemos también a Miguel Ángel Cosano y Fernando Gómez su implicación en el proyecto a la hora de enseñarnos acerca del paradigma EBT y su uso en la actualidad, y también para resolver dudas y elaborar el guión de un escenario de entrenamiento realista.

Agradecemos a José Luis Parra y a todo el personal de Global Training Aviation Madrid el haber reunido a varios expertos en instrucción de vuelo y habernos permitirnos utilizar sus aulas para realizar allí las pruebas con usuario. También les agradecemos de corazón la visita guiada por sus instalaciones permitiéndonos conocer sus simuladores de vuelo de última generación.

Resumen

Hoy en día, el factor humano es una de las causas contribuyentes a la mayoría de accidentes aéreos.

Es por ello que durante los últimos años se ha estado haciendo más énfasis en mejorar el entrenamiento que reciben los pilotos. Para ello, desde la industria aeronáutica se está impulsando un nuevo paradigma que recopila datos e integra información sobre el rendimiento de los pilotos en todas las competencias necesarias, tanto técnicas como no técnicas. A este paradigma se le conoce como Entrenamiento Basado en Evidencia.

Los simuladores de vuelo son una herramienta fundamental en la formación de un piloto. En un simulador el piloto se enfrenta a escenarios especialmente diseñados por un instructor para probar a fondo sus competencias. Para evaluar las competencias basándose en la evidencia, el instructor trata de percibir e identificar correctamente unos determinados *Comportamientos Observables* que pueden darse o no en el piloto.

La evaluación de las competencias no técnicas podría resultar imprecisa y demasiado subjetiva si el instructor no está bien formado para realizar esta identificación. Este trabajo presenta la producción de un simulador ludificado, sencillo e intuitivo, del trabajo de un instructor de vuelo, con el objetivo de que estos profesionales se familiaricen con ese trabajo de reconocimiento de comportamientos en los pilotos.

Los objetivos establecidos en el proyecto finalizaron en la elaboración de un prototipo jugable denominado como Flight Instructor Simulator. Además, se realizaron pruebas con instructores reales de vuelo para comprobar su eficacia. El código fuente del prototipo se haya en un repositorio público en GitHub.

Para llevar a cabo la elaboración del proyecto, se investigaron diferentes juegos serios basados en la enseñanza y se mantuvieron charlas con expertos en la instrucción de pilotos. Después de hacer bocetos y un primer prototipo sin interfaz gráfica, el siguiente paso fue trabajar en la elaboración del prototipo y probarlo de manera presencial con instructores de pilotos reales.

Palabras clave

Aeronáutica, Simulación, Ludificación, Evaluación, Competencias No Técnicas, Conducta Observable, Desarrollo de Videojuegos, Unity.

Abstract

Nowadays, the human factor is one of the contributing causes of most air accidents.

That is why in recent years more emphasis has been placed on improving the training pilots receive. To this end, the aviation industry is promoting a new paradigm that collects data and integrates information on pilot performance in all necessary competencies, both technical and non-technical. This paradigm is known as Evidence Based Training.

Flight simulators are a fundamental tool in pilot training. In a simulator the pilot is confronted with specially designed scenarios designed by an instructor to thoroughly test their competencies. In order to assess the competencies based on evidence, the instructor tries to perceive and correctly identify certain Observational Behaviours that may or may not be present in the pilot.

The assessment of non-technical competencies could be inaccurate and too subjective if the instructor is not well trained to make this identification. This work presents the production of a simple and intuitive gamified simulator of the work of a flight instructor, with the aim of familiarising these professionals with the work of recognising behaviour in pilots.

The objectives set out in the project resulted in the development of a playable prototype named Flight Instructor Simulator. In addition, tests were carried out with real flight instructors to check its effectiveness. The source code of the prototype is in a public repository on GitHub.

For the development of the project, research was carried out on different serious teaching-based games and discussions were held with experts in pilot training. After sketches and a first prototype without graphical interface, the next step was to work on the development of the prototype and to test it in person with real pilot instructors.

Keywords

Aeronautics, Simulation, Gamification, Assessment, Non-Technical Competencies, Observable Behavior, Video Game Development, Unity.

Índice

1. Introducción **1**
- 1.1. Motivación . . . 2
- 1.2. Propósito y alcance . . . 3
  - 1.2.1. Propósito . . . 3
  - 1.2.2. Alcance . . . 4
- 1.3. Asignaturas relacionadas . . . 4
- 1.4. Estructura del trabajo . . . 5

2. Estado de la cuestión **7**
- 2.1. Formación no técnica en aeronáutica . . . 7
- 2.2. Juegos serios para la formación . . . 11
- 2.3. Trabajo relacionado . . . 14
  - 2.3.1. Afghanistan Pre-deployment: Cultural Awareness . . . 14
  - 2.3.2. STAR Classroom Simulator . . . 15
  - 2.3.3. Proyecto Versa . . . 16

3. Objetivos y especificación **19**
- 3.1. Objetivos . . . 19
- 3.2. Especificación . . . 20
  - 3.2.1. Interfaz de usuario . . . 20
  - 3.2.2. Escenarios de entrenamiento . . . 21
  - 3.2.3. Sistema de evaluación . . . 23
  - 3.2.4. Tutorial . . . 23

4. Metodología y herramientas **25**
- 4.1. Metodología de desarrollo . . . 25
  - 4.1.1. Herramientas utilizadas . . . 26

5. Diseño y desarrollo de Flight Instructor Simulator **31**
- 5.1. Análisis y diseño . . . 31
  - 5.1.1. Navegación . . . 31
  - 5.1.2. Diseño de la arquitectura . . . 31

5.2. Implementación en C# 32
5.2.1. *Step* 33
5.2.2. *Script* 34
5.2.3. *TableCompetencesToOB* 35
5.2.4. *LevelManager* 35
5.2.5. *Step Displayer* 35
5.2.6. *Evaluator* 36
5.2.7. *ResultsTracker* 36
5.2.8. *OBSelector* 37
5.3. *Escenas de Unity* 37

6. Resultados **41**
6.1. Pruebas 42
6.2. Resultados 43
6.3. Discusión 44

7. Conclusiones **47**
7.1. Objetivos alcanzados 47
7.2. Trabajo futuro 49
7.3. Reflexión final 49

A. Contribuciones individuales **51**
A.1. Antonio Jesús Garduño Guerra 52
A.2. Javier Cruz López de Ochoa 54

B. Introduction **57**
B.1. Motivation 57
B.1.1. Purpose 59
B.1.2. Scope 59
B.2. Related subjects 60
B.3. Structure of the work 61

C. Conclusions **63**
C.1. Targets achieved 63
C.2. Future Work 64
C.3. Final thoughts 65

D. Cuestionario utilizado en las pruebas **67**

E. Resultados del cuestionario **75**

Bibliografía **81**

Índice de figuras

2.1. Piloto entrenando en el Link Trainer. 8
2.2. Captura de pantalla in-game del Microsoft Flight Simulator. . 9
2.3. Partida de Kriegsspiel en mitad de desarrollo. 12
2.4. Captura de pantalla del juego America's Army. 13
2.5. Una captura de uno de los menús de Afghanistan Pre-deployment: Cultural Awareness. 15
2.6. Una captura de Afghanistan Pre-deployment: Cultural Awareness donde el jugador debe elegir qué hacer. 16
2.7. Captura de STAR Classroom Simulator donde se elige la dificultad de la sesión. 17
2.8. Captura de STAR Classroom Simulator tras realizar una acción. 18
2.9. Captura del tutorial de Portal 2, uno de los juegos usados por el proyecto Versa, concretamente para desarrollar la flexibilidad cognitiva. 18

3.1. Croquis de la interfaz final deseada hecha en moqups. 20

4.1. Parte de las actas realizadas durante el desarrollo. 27

5.1. Diagrama de la navegación entre escenas. 32
5.2. Diagrama de clases detallando la lógica del simulador. 33
5.3. Diagrama de clases detallando los principales componentes de Unity. 33
5.4. Editor en Unity de la escena de Juego. 36
5.5. Logo del simulador. 37
5.6. Captura del simulador en mitad de partida. 39

6.1. Nosotros, nuestros directores del trabajo y David Rioja de Didactoons en Global Training Aviation Madrid. 41
6.2. Uno de los expertos jugando el tutorial. 43

Capítulo 1

Introducción

A pesar de todos los avances experimentados en el campo de la seguridad aeronáutica, el factor humano sigue siendo una de las causas contribuyentes a la mayoría de accidentes aéreos (Kharoufah et al., 2018).

Por dicha razón, en los últimos años, tanto la ICAO [1] (International Civil Aviation Organization) (Organization, 2013) como la IATA [2] (International Air Transport Association) (Association, 2013) han estado impulsando un nuevo paradigma de entrenamiento llamado Entrenamiento Basado en Evidencia (o EBT, del inglés *Evidence-Based Training*). Este paradigma trata de adaptar el entrenamiento a las necesidades reales de los pilotos, integrando las competencias tanto técnicas como no técnicas.

Los simuladores de vuelo son uno de los avances históricos más notables tanto en la Aeronáutica como la propia Computación, siendo estos a día de hoy una herramienta indispensable a la hora de entrenar a un piloto. No obstante, el cómo se usan estos simuladores es igual de importante, y en este sentido, el paradigma EBT permite la creación de escenarios en el simulador especialmente diseñados para mejorar competencias específicas que el piloto necesita entrenar.

El diseño de estos escenarios no es trivial y depende en gran medida de la habilidad y experiencia del instructor, quienes no disponen de tiempo para crear escenarios personalizados para cada piloto. El éxito del entrenamiento también depende en gran medida de la capacidad del instructor a la hora de valorar el comportamiento del piloto, para ver si este posee tal o cual competencia. Es por ello que una buena formación para los instructores de vuelo al final también es importante para la seguridad aeronáutica.

[1] https://www.icao.int/
[2] https://www.iata.org/

1.1. Motivación

Uno de los retos permanentes en el campo aeroespacial es la formación del personal. En el caso de la aviación comercial, esta formación la llevan a cabo las propias aerolíneas, y el modelo que se está imponiendo actualmente a la hora de entrenar a los pilotos es el mencionado EBT.

Este nuevo modelo de entrenamiento se preocupa no sólo de las capacidades técnicas del piloto (es decir, la correcta ejecución de maniobras, el conocimiento de los controles del avión, etc.), sino en sus competencias no técnicas, más relacionadas con los denominados Factores Humanos. Los instructores de vuelo que ahora están llamados a actuar como evaluadores de pilotos utilizando el modelo EBT, deben fijarse en ciertos patrones de comportamiento de los pilotos llamados Comportamientos Observables (u **OB**, del inglés *Observational Behaviour*) las cuales revelan si el piloto a evaluar tiene o no las correspondientes competencias, como pueden ser la Comunicación o el Liderazgo.

El problema es que es difícil seguir las reglas del proceso de evaluación en este nuevo paradigma y, a menudo, se realizan evaluaciones más subjetivas en la que el instructor no se limita a observar y a registrar los Comportamientos Observables del piloto. Los instructores se enfrentan por tanto a este nuevo paradigma de entrenamiento que es diferente al empleado tradicionalmente en aviación, a lo que hay que añadir también la falta de cierto consenso que aún existe en muchos detalles del modelo. Por ejemplo, según nos cuentan los especialistas, los instructores de una compañía como Iberia contemplan 9 competencias diferentes y las puntúan en un rango de 0 a 4, mientras que los de otra compañía pueden considerar sólo 8 competencias y calificarlas con un número de 0 a 5.

Es en esta situación cuando surge la pregunta "¿Quién instruye a los instructores?", el tema central de un artículo de Alfredo Hernández, Rubén Dapica y Federico Peinado (Hernández et al., 2020) que sirve de detonante para nuestro trabajo. Las compañías que eligen certificarse para entrenamiento EBT, deben de hecho formar a sus instructores en este paradigma mediante cursos específicos y deben también examinarlos cada 3 años para verificar que siguen capacitados para ejercer sus funciones. (Association, 2013)

Al tratarse de un paradigma bastante novedoso, y no haber apenas compañías que hayan certificado su formación en esta metodología, no existe mucho material para que los instructores aprendan a integrar este nuevo enfoque en su trabajo cotidiano.

Dentro del Desarrollo de Videojuegos, el uso de simuladores ha sido habitualmente una herramienta vital en programas de formación práctica, y de manera muy especial, como se ha dicho, en el ámbito de la Aeronáutica.

No sólo es que los simuladores hayan estado presentes desde los inicios

de la Informática, es que muchos se usan actualmente en formación, otros en entrenamiento de todo tipo de competencias, en terapia psicológica y muchísimos otros escenarios.

Por ejemplo, en el sector de automovilismo, para replicar el comportamiento del coche en un circuito, existen simuladores que permiten a los pilotos familiarizarse con sus controles, incluso a nivel de hardware, sin afrontar los riesgos y costes de una conducción real. Esto también permite a los ingenieros poner a prueba ciertos avances tecnológicos sobre la simulación del vehículo. [3]

El elevado coste y los riesgos para la seguridad que involucran el pilotaje de un avión justificaron enseguida la creación de simuladores de vuelo donde poder realizar prácticas y experimentos en un entorno seguro y con una inversión más moderada. Hoy día existen varios simuladores, con distintos enfoques, e incluso se comercializan videojuegos de simulación para el mercado masivo, con un realismo y un número de usuarios impresionante. El paradigma de esto podemos verlo en las últimas entregas de la conocida saga Microsoft Flight Simulator (Xbox Game Studios, 2020).

Es importante dejar claro que en este trabajo nos hemos interesado en el uso de simuladores para la formación de instructores de pilotos, no necesariamente realistas sino enfocados a las competencias no técnicas y siguiendo el paradigma EBT. Estas competencias complementan lo que es propiamente el control de la aeronave, y resultan especialmente desafiantes para definir y evaluar, al involucrar factores humanos.

1.2. Propósito y alcance

En esta sección tratamos de presentar el propósito y de acotar el alcance del proyecto sobre el que versa este trabajo.

1.2.1. Propósito

El propósito de este Trabajo de Fin de Grado (TFG) es diseñar y desarrollar un simulador ludificado que ayude en la formación de instructores de vuelo según el paradigma EBT. Este simulador no pretende sustituir el entrenamiento actual de los instructores de vuelo, sino complementarlo y enfatizar los aspectos más relevantes.

Con este trabajo obviamente también se pretende afianzar el conocimiento obtenido en el Grado en Desarrollo de Videojuegos y usarlo para desarrollar una herramienta útil orientada a un sector concreto con problemas importantes y reales como es el de la Aeronáutica.

En la Sección 1.3, explicamos las asignaturas del Grado en Desarrollo de Videojuegos que hemos considerado cruciales a la hora de llevar a cabo este

[3] https://www.mercedesamgf1.com/en/news/2020/04/behind-scenes-f1-simulator/

desarrollo.

1.2.2. Alcance

Con el simulador propuesto se pretende que los instructores de vuelo se familiaricen con los Comportamientos Observables de los pilotos, organizados según las competencias del modelo EBT y que practiquen su identificación en un entorno inmersivo e interactivo. El simulador ludificado no estará dirigido al público general, sino más bien a las aerolíneas, con el fin de que estas puedan implantarlo como complemento a la formación de sus instructores.

Para validar que el simulador posee esta utilidad, crearemos un escenario basado en un accidente real, en colaboración con expertos en aviación, en donde se representarán las acciones más importantes que dieron lugar al accidente. Y usaremos este escenario como prueba para enseñar nuestra aplicación a instructores reales de vuelo.

En ningún caso se ha pretendido construir un "simulador de vuelo" pues los aspectos físicos del avión y de su entorno quedan fuera del alcance de este proyecto. Tampoco será un simulador autocontenido que pueda usarse durante todo el proceso de entrenamiento de los instructores. Nuestro proyecto se centra exclusivamente en la detección de *OBs* de pilotos basados en un escenario guionizado, que el simulado sigue a rajatabla, dando oportunidad al usuario de reaccionar en cada momento relevante.

1.3. Asignaturas relacionadas

En esta sección se describe la relación de este TFG con los contenidos impartidos a lo largo del Grado en Desarrollo de Videojuegos. Los conocimientos más relevantes que han sido aplicados en este trabajo vienen dados por las siguientes asignaturas:

- **Fundamentos de la Programación.** En esta asignatura se enseñan las bases de la programación que usamos a lo largo de todo el proyecto. Además, se utiliza C#, el cual es el lenguaje de *scripting* de Unity.
- **Motores de Videojuegos.** En esta asignatura se enseña lo básico para comprender el funcionamiento de un motor de videojuegos (en concreto, Unity), aprendiendo cuáles son los componentes más relevantes del motor y cómo usarlos.
- **Metodologías Ágiles de Producción.** En esta asignatura aprendemos cómo gestionar un proyecto con una metodología ágil como Scrum, lo que nos ha ayudado a que el desarrollo del proyecto haya sido bastante flexible, pudiéndonos adaptar a los cambios en la especificación que han ido surgiendo en nuestra interacción con los expertos.

- **Tecnologías de la Programación en Videojuegos.** En esta asignatura se estudian patrones de diseño que han sido aplicados al proyecto, como puede ser Singleton, además de afianzar conceptos importantes de Programación Orientada a Objetos como el polimorfismo.

- **Estructuras de Datos y Algoritmos.** En esta asignatura aprendemos sobre los órdenes de complejidad y sobre las estructuras que nos permiten controlar esta complejidad a fin de producir código eficiente.

- **Diseño de Sistemas Interactivos.** En esta asignatura se imparte conocimiento sobre el diseño de interfaces centradas en el usuario y el uso de ciertos patrones útiles para esta tarea.

- **Videojuegos para Dispositivos Móviles.** En esta asignatura se descubren los pormenores del desarrollo multiplataforma y se profundiza en el conocimiento sobre Unity más enfocado al desarrollo para dispositivos móviles.

1.4. Estructura del trabajo

La estructura de este trabajo se compone de los siguientes capítulos:

- En el Capítulo 1 se presenta el propósito y alcance que tiene este trabajo. Presentamos uno de los problema que existen actualmente en Aeronáutica y relacionamos este trabajo con asignaturas del grado.

- En el Capítulo 2 se revisa el estado de la cuestión en materia de formación de las *soft skills* y hablamos de los juegos serios como soporte. Finaliza con un pequeño estudio de trabajos relacionados.

- En el Capítulo 3 se presentan los objetivos principales del trabajo y se hace una especificación de todos los requisitos necesarios para poder desarrollar la herramienta propuesta.

- En el Capítulo 4 se explica la metodología usada para desarrollar el proyecto y qué herramientas concretas se han utilizado en el proceso.

- En el Capítulo 5 se realiza una explicación detallada del análisis, diseño e implementación de nuestro proyecto, y también del escenario tomado como ejemplo.

- En el Capítulo 6 se explican las pruebas y se discuten los resultados obtenidos de las pruebas con usuarios reales que hemos realizado para validar el funcionamiento y evaluar la utilidad de esta herramienta, siendo el resultado en general positivo.

- En el Capítulo 7 extraemos algunas conclusiones del proyecto, explorando si se han cumplido los objetivos y planteando líneas de trabajo futuro.

Capítulo 2

Estado de la cuestión

Dado el propósito del mismo, expuesto en el capítulo anterior, la revisión del estado de la cuestión que requiere este trabajo ha sido organizada en tres secciones principales. En la primera sección se explica el proceso de formación de competencias no técnicas en Aeronáutica. En la segunda, se aborda el tema de los juegos serios, con el enfoque en su uso como apoyo a la formación. Finalmente, se presentan otros trabajos relacionados con este.

2.1. Formación no técnica en aeronáutica

Para poder comprender las tendencias actuales en entrenamiento de pilotos, conviene explicar brevemente la historia de la formación aeronáutica y los simuladores de vuelo. (A, 2018) (Hernandez, 2012)

Desde el primer vuelo realizado por los hermanos Wright en 1903, se ha tratado de formar a los pilotos asumiendo el menor riesgo posible, por lo que pronto se empezaron a diseñar tecnologías con este propósito. Los pilotos comenzaron realizando ejercicios en aviones sencillos y con misiones simples, pero reales al fin y al cabo. A medida que estos iban adquiriendo competencias, pasaban a manejar aviones más complejos o a afrontar desafíos más ambiciosos.

Uno de los primeros simuladores de vuelo fue construido en 1910. Fue el *The Sanders "Teacher"*, una especie de maqueta a tamaño real, hecha con partes de aviones reales. Este simulador se colocaba sobre una articulación universal y se orientaba de forma que el viento lo golpeara frontalmente, haciendo posible que ciertas partes del simulador pudieran girar.
[1]

Aunque anteriormente existieron algunos simuladores como el *Antoinette Trainer* en 1909, estos necesitaban la presencia de asistentes para la realización de los ejercicios, ya que eran estos los que movían el simulador.

[1] http://havkar.com/en/blog/view/first-aircraft-simulator/90

Posteriormente se construirían simuladores, que seguían siendo limitados pero ya otorgaban la posibilidad de realizar giros en los tres ejes (cabeceo, guiñada y alabeo) sin necesidad de usar el viento.

En 1930, se desarrolla el *"Link Trainer"*, que al principio sólo era una cabina vacía que podía realizar movimientos mecánicos, pero posteriormente se le incluyeron instrumentos de control. Se puede apreciar mejor en la figura 2.1.

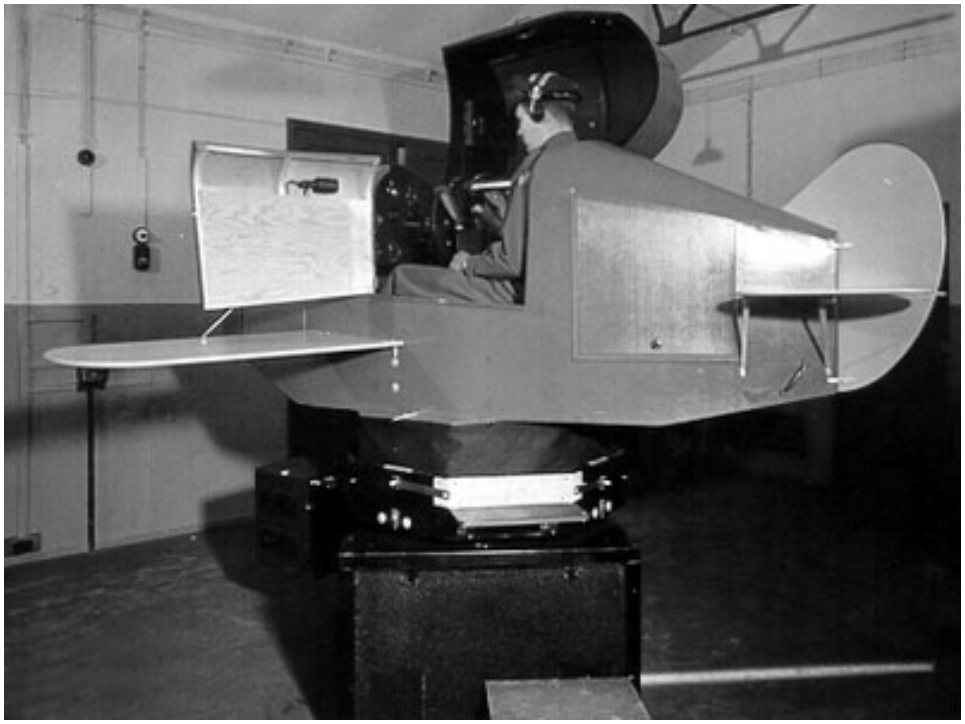


Figura 2.1: Piloto entrenando en el Link Trainer.

Posteriormente, durante la II Guerra Mundial, se desarrollaron los ordenadores analógicos, capaces de resolver ecuaciones relativas a los cálculos de vuelo, por lo que surgieron los primeros simuladores electrónicos.

El primer simulador comprado por una aerolínea (American Air) fue desarrollado por Curtiss-Wright en 1948 para el Boeing 377 Stratocruiser. Este simulador incluía el funcionamiento de toda la cabina y los controles asociados a este modelo de avión.

Por otra parte, a finales de los años 50, se desarrollan los primeros simuladores electrónicos con movimiento, permitiendo al principio un solo grado de libertad, pero llegando a permitir seis grados de libertad. En los años 70 se empezaron a colocar ordenadores y sistemas de visualización en las cabinas de estos simuladores que mostraban entornos en 3D, generalmente de las proximidades del aeropuerto.

En la actualidad, se han ido refinando estos sistemas hasta conseguir un gran nivel de realismo e inmersión para los pilotos (Véase figura 2.2), llegando incluso a desarrollarse programas de entrenamiento con realidad virtual oficiales, aprobados por EASA (*European Union Aviation Safety Agency*). [2]



Figura 2.2: Captura de pantalla in-game del Microsoft Flight Simulator.

Ahora la formación de los pilotos comienza en un simulador y posteriormente se les permite realizar pequeños vuelos en avioneta para familiarizarlos con una maquinaria relativamente sencilla.

Cuando se hace un curso completo sobre un modelo de avión concreto que va a pilotarse, el conocimiento y las competencias se van adquiriendo en distintas fases. El curso empieza por una fase teórica en la que se explican los fundamentos del manejo del correspondiente modelo de avión, para posteriormente pasar a una fase en simuladores sencillos, sin ningún tipo de movimiento en la cabina del mismo. Finalmente se pasa a entrenar en simuladores de vuelo más complejos, en los que se trata de simular la experiencia real de volar de la manera más fiel posible, incorporando incluso movimiento de la cabina. Cada seis meses se suelen hacer entrenamientos en simulador para refrescar el conocimiento sobre el avión y las posibles averías de éste. Además, cada tres años se hace un examen público para comprobar que, efectivamente, el piloto conserva sus capacidades para realizar su trabajo.

Durante este tipo de cursos, se imparte formación tanto técnica (funcionamiento del avión, protocolos de actuación, tipos de maniobras, etc) como no técnica, pues en los últimos años ha cobrado mayor importancia la cues-

[2] https://www.easa.europa.eu/newsroom-and-events/press-releases/easa-approves-first-virtual-reality-vr-based-flight-simulation

tión de los factores humanos. Esta tendencia está recogida en un paradigma actual llamado Entrenamiento Basado en Evidencias (*Evidence Based Training* ,EBT). Este modelo busca identificar y desarrollar las competencias requeridas por los pilotos, para operar de forma segura, efectiva y eficiente en entornos de aviación comerciales, planteando eventos durante el entrenamiento que están basados en la evidencia extraída de los errores producidos en sesiones de entrenamiento o vuelos reales.

Las competencias no técnicas que se contemplan como parte de este modelo son las siguientes:

- **Aplicación de procedimientos.** Identifica y aplica los procedimientos adecuados conforme a las instrucciones operativas publicadas y las regulaciones aplicables. Para todas las competencias existen comportamientos observables (*Observational Behavior*, *OB*) que demuestran cuando un piloto posee dicha competencia. Un ejemplo de comportamiento observable de esta competencia sería comprobar que el piloto "Monitoriza el estado de los sistemas de la aeronave".

- **Comunicación.** Demuestra que es capaz de comunicarse efectivamente de forma oral, no verbal y escrita, tanto en situaciones normales como anormales. Un ejemplo de *OB* sería "Se asegura de que el receptor es capaz de recibir información y está preparado para ello."

- **Manejo de la trayectoria de vuelo de la aeronave, control automático.** Controla la trayectoria de vuelo de la aeronave a través de la automatización, incluyendo el uso apropiado de los sistemas de manejo de vuelo y las guías. Un ejemplo de *OB* sería "Controla la trayectoria de vuelo para conseguir un rendimiento operativo óptimo."

- **Manejo de la trayectoria de vuelo de la aeronave, control manual.** Controla la trayectoria de vuelo de la aeronave a través del vuelo manual, incluyendo el uso apropiado de los sistemas de manejo de vuelo y de guía. Un ejemplo de *OB* sería "Mantiene la trayectoria de vuelo deseada durante vuelo manual mientras se ocupa de otras tareas y distracciones."

- **Liderazgo y trabajo en equipo.** Demuestra liderazgo y trabajo en equipo de forma efectiva. Un ejemplo de *OB* sería "Admite los fallos y toma la responsabilidad de dichos fallos."

- **Resolución de problemas y toma de decisiones.** Identifica riesgos de forma precisa y resuelve problemas. Usa los procesos adecuados para tomar decisiones. Un ejemplo de *OB* sería "Busca información precisa y adecuada de las fuentes apropiadas."

- **Conciencia de la situación.** Percibe y comprende toda la información relevante disponible y se anticipa a lo que pueda suceder que afecte

a la operación. Un ejemplo de *OB* sería "Identifica y gestiona amenazas hacia la seguridad tanto de la aeronave como de las personas a bordo."

- **Gestión de la carga de trabajo.** Administra los recursos disponibles de forma eficiente con el fin de priorizar y llevar a cabo las tareas a tiempo bajo cualquier circunstancia. Un ejemplo de *OB* sería "Ofrece y acepta ayuda, delega cuando es necesario y pide ayuda a tiempo." (Association, 2013)

  Aunque este modelo está ganando fuerza, aún es muy reciente y es un hecho que hay muy pocas compañías acreditadas para impartir EBT, pero es porque hay muchas otras compañías que aún están intentando conseguir dicha acreditación. De hecho, para poder impartir EBT, los instructores de pilotos deben realizar un curso específico. Además, el EBT tiene dos niveles (base y enhance), pero prácticamente ninguna compañía ha conseguido aún el enhance.

  También es cierto que hay poco consenso a la hora tanto de plantear las competencias como de evaluar, pues algunas compañías consideran que es necesario que existan otras competencias además de las especificadas anteriormente. Esto es debido a que no todas las compañías tienen las mismas necesidades, ya que por ejemplo, en el vuelo militar se necesitan pilotos que sepan reaccionar en condiciones más extremas que las que se suelen dar en vuelo comercial. Por otra parte, aunque la mayoría puntúan sobre 5 en las evaluaciones, también hay compañías que utilizan otras escalas.

  Para ayudarnos a profundizar en este tema, durante la realización de este trabajo hemos tenido acceso a varios instructores de pilotos y expertos en Aeronáutica, como nuestro codirector Rubén Dapica, que como piloto y profesor de Ingeniería Aeroespacial fue quien nos dió a conocer el problema y nos facilitó contactos en el sector, como el de los instructores Miguel Ángel Cosano y Fernando Gómez.

2.2. Juegos serios para la formación

De nuevo, comenzaremos esta sección dando cierto contexto sobre el tema a tratar. En este caso, los juegos serios.

Se conoce como juegos serios (del inglés *serious games*) a aquellos juegos cuya finalidad principal no es divertir o entretener a sus jugadores, sino otra que se considera más importante, como puede ser la trasmisión de conocimientos, la adquisición de habilidades o incluso la realización de trabajos como la propia investigación científica. Dos ejemplos de juegos serios podrían serían RescueSim (VSTEP, 2008) y Foldit(University of Washington, 2008). En el primero se simulan operaciones de rescate que es posible configurar

para que los profesionales del salvamento practiquen. En el segundo los jugadores deben plegar proteínas, y gracias a él se han hecho avances como el descubrimiento de la estructura de una enzima del VIH.

Los juegos serios se han utilizado desde el siglo XX, aunque sin recibir este nombre, como complemento para la educación, a menudo en forma de juegos en papel. Un ejemplo es el *Kriegsspiel*, un juego desarrollado en Prusia en siglo XIX para entrenar en estrategia y tácticas de combate a sus oficiales. Véase figura 2.3.



Figura 2.3: Partida de Kriegsspiel en mitad de desarrollo.

En 1970 Clarck C. Abt, un investigador americano, usó por primera vez el término y los definió como juegos con propósito educativo, aunque no por ello debían dejar de entretener.

A partir de los años 2000 empezaron a surgir juegos serios en formato electrónico. También empezaron a cobrar relevancia en más terrenos aparte de la educación. Un ejemplo de esto lo vemos en el videojuego America's Army (United States Army, 2002), un videojuego creado por la armada de Estados Unidos para reclutar soldados y entrenarlos. Véase figura 2.4.

Hoy en día vemos una gran diversidad de juegos serios, interesantes para servir a multitud de propósitos. Tenemos ejemplos como Dragon Box Elements (Kahoot DragonBox AS, 2014), diseñado para que los niños aprendan geometría, o Pacific (Gamelearn, 2015), un juego ambientado en una isla del pacífico con el que se pretende entrenar las *soft skills* necesarias para ser un buen líder, ya que el jugador tendrá que ponerlas en práctica para asignar ta-



Figura 2.4: Captura de pantalla del juego America's Army.

reas y prioridades o gestionar recursos en la isla virtual, entre otros desafíos. [3]

De hecho, existen videojuegos que, aunque su prioridad sí es divertir, pueden servir como recursos de aprendizaje para distintas personas en diferentes situaciones (De Gloria et al., 2014a). Un ejemplo podríamos verlo en juegos como Rocksmith 2014 (Ubisoft, 2013), con el que realmente se puede aprender a tocar la guitarra. Otro ejemplo es PC Building Simulator (The Irregular Corporation, 2018), con el que, además de divertirse, es posible aprender mucho sobre los componentes de un ordenador personal.

En el contexto de la educación, que veremos más en detalle en la siguiente sección, existen herramientas como e-Adventure (Torrente et al., 2010), creado en la Universidad Complutense de Madrid, que permiten crear juegos serios -en este caso del género de las aventuras *point & click*- de forma sencilla. También es posible encontrar un simulador para entrenar a profesores de escuela, llamado STAR Classroom Simulator[4].

Todos estos ejemplos nos dan a entender que el uso de juegos serios

[3] https://www.game-learn.com/en/resources/blog/all-you-need-to-know-serious-games-game-based-learning-examples/

[4] https://www.seriousgamemarket.com/2010/09/serious-games-improving-teacher.html

como complemento en la formación es bastante útil y ayuda a afianzar los conocimientos adquiridos en la teoría mediante la práctica diseñada desde un enfoque más ameno y recreativo.

2.3. Trabajo relacionado

Como hemos visto anteriormente, en Aeronáutica existe una gran trayectoria en el uso de simuladores para entrenar a los pilotos. Sin embargo, no hay antecedentes de herramientas similares para el entrenamiento de aquellos que ejercen de instructores de estos pilotos. Las competencias de un instructor van más allá de las de un piloto, y habitualmente se demuestran en la interacción real con sus alumnos. Por ello, actualmente está muy extendida la práctica de evaluar la habilidad de un instructor es a través de otro instructor, usualmente con más experiencia, que valora el trabajo de su compañero

Para este proyecto vamos a enfocarnos en trabajos relacionados, aunque no necesariamente del mundo aeronáutico, sino sobre la enseñanza de *soft skills* y la consideración de factores humanos en tareas críticas (De Gloria et al., 2014b). A continuación destacamos los proyectos que hemos considerado más relevantes.

2.3.1. Afghanistan Pre-deployment: Cultural Awareness

Afghanistan Pre-deployment: Cultural Awareness es un juego serio empleado por las fuerzas armadas estadounidenses para entrenar a sus soldados antes de ser desplegados en Afganistán en los comienzos de este conflicto (2001-2021). Fue desarrollado en 2007 por LINE Communication, una empresa que trabaja de forma cercana al Ministerio de Defensa del Reino Unido. LINE trabajó con el ministerio de defensa en recrear escenarios realistas e inmersivos. El juego servía para instruir a militares en la cultura y sociedad afgana con el fin de promover la cooperación entre ambas culturas. (Barbieri et al., 2014)

El juego comienza con una introducción que tiene el propósito de informar sobre la geografía afgana, su población y su cultura. Después hay una serie de escenarios donde los jugadores encarnan a soldados destinados a la zona y pueden llevar a la práctica y poner a prueba los conocimientos adquiridos durante dicha introducción.

El juego se programó en flash pensado para ser utilizado dentro de un sistema de gestión de aprendizaje (*Learning Management System*, LMS, en inglés) o directamente en un CD. Al estar dentro de un LMS, permitía trackear el progreso de los jugadores. Además, podía ser utilizado desde dispositivos móviles que soportasen tecnología flash.

En el juego existe una serie de escenarios, donde al jugador se le presentan



Figura 2.5: Una captura de uno de los menús de Afghanistan Pre-deployment: Cultural Awareness.

fotos y videos. La interfaz es estática y muy básica. La forma de evaluar es mediante preguntas de respuesta múltiple, en el que el jugador elige una de ellas y recibe una respuesta inmediata sobre un simple indicador visual que indica la tensión social. Véase figura 2.6.

2.3.2. STAR Classroom Simulator

STAR Classroom Simulator es un juego serio donde un profesor puede mejorar sus habilidades como tutor interactuando con alumnos virtuales. El juego es una herramienta que enseña los mismos comportamientos a los profesores para que estos ejerzan su labor de la mejor manera. Este juego fue creado en 2007.

El juego fue pensado para ser jugado en web y desarrollado en flash usando el IDE de Visual Studio 2005. En 2007, eso significaba que solo podía ser jugado desde un ordenador.

En el juego, al jugador se le presenta una pantalla donde elige la dificultad y si quieren que el tiempo sea un factor a considerar a la hora de evaluar la toma de decisiones del jugador. Véase figura 2.7.

El juego funciona a base de ver una pequeña película hecha en flash que proporciona información sobre la mente de Jamal (un estudiante) y después elegir una opción entre tres. Al tomar la decisión, después se ofrece una

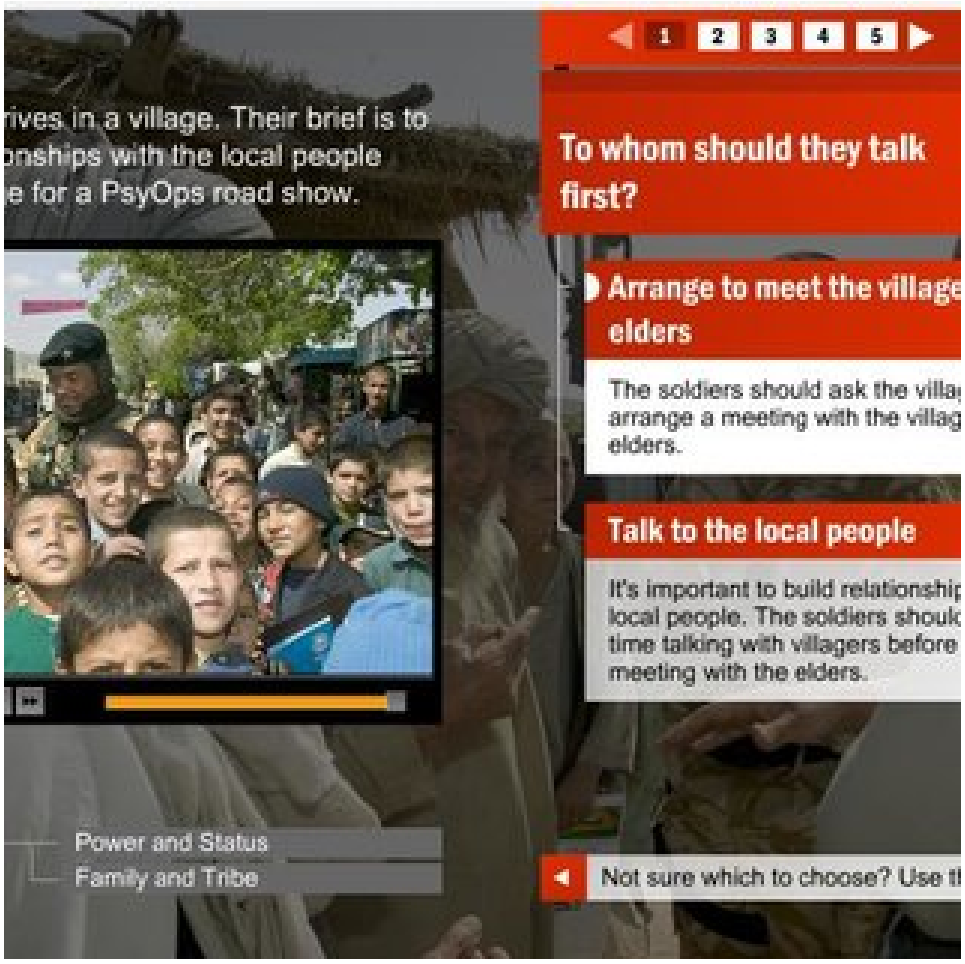


Figura 2.6: Una captura de Afghanistan Pre-deployment: Cultural Awareness donde el jugador debe elegir qué hacer.

respuesta inmediata de parte del juego en la que informa al jugador de cómo lo está haciendo con el color de una manzana. Este color transiciona de verde a amarillo a rojo. Este proceso se repite varias veces hasta que el escenario finaliza.

Estos dos proyectos tienen cierta similitud al nuestro en cuanto a la evaluación, pues en ambos se presenta una elección entre múltiples respuestas y se proporciona *feedback* inmediato. Véase figura 2.8.

2.3.3. Proyecto Versa

Proyecto Versa[5], también llamado *Video gamEs foR Skills trAining*, es un proyecto centrado en desarrollar las *soft skills* mediante el uso de video-

[5]https://gecon.es/versa-project/

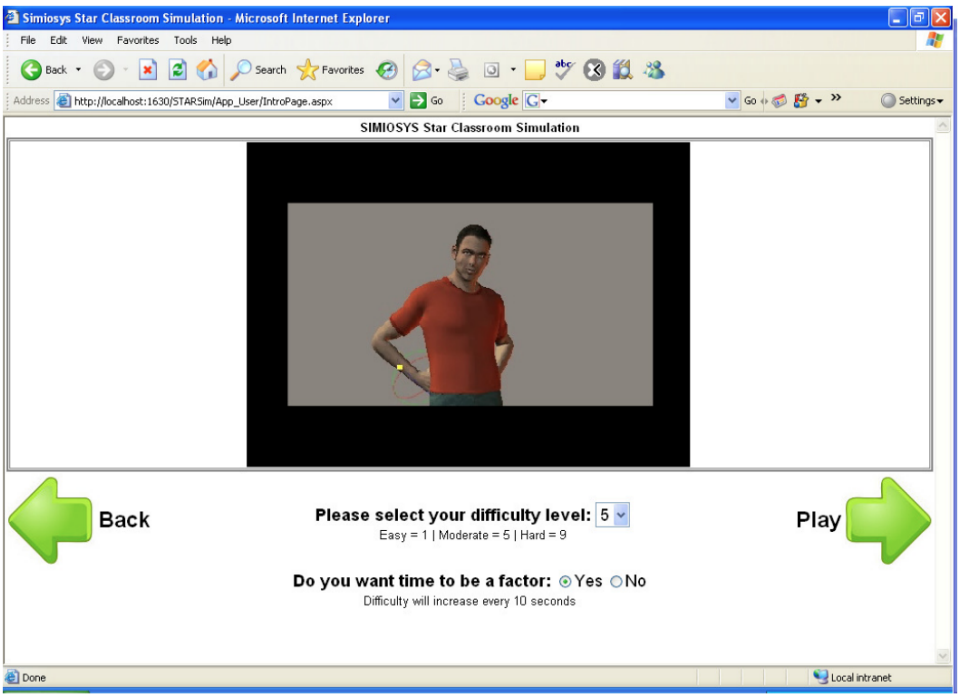


Figura 2.7: Captura de STAR Classroom Simulator donde se elige la dificultad de la sesión.

juegos comerciales. El proyecto está enfocado a los estudiantes de doctorado de diferentes universidades de la red *Aurora* que participan en el proyecto. Estas universidades son principalmente la Universidad Rovira i Virgili, la Universidad de Innsbruck y la Universidad Pública de Amsterdam.

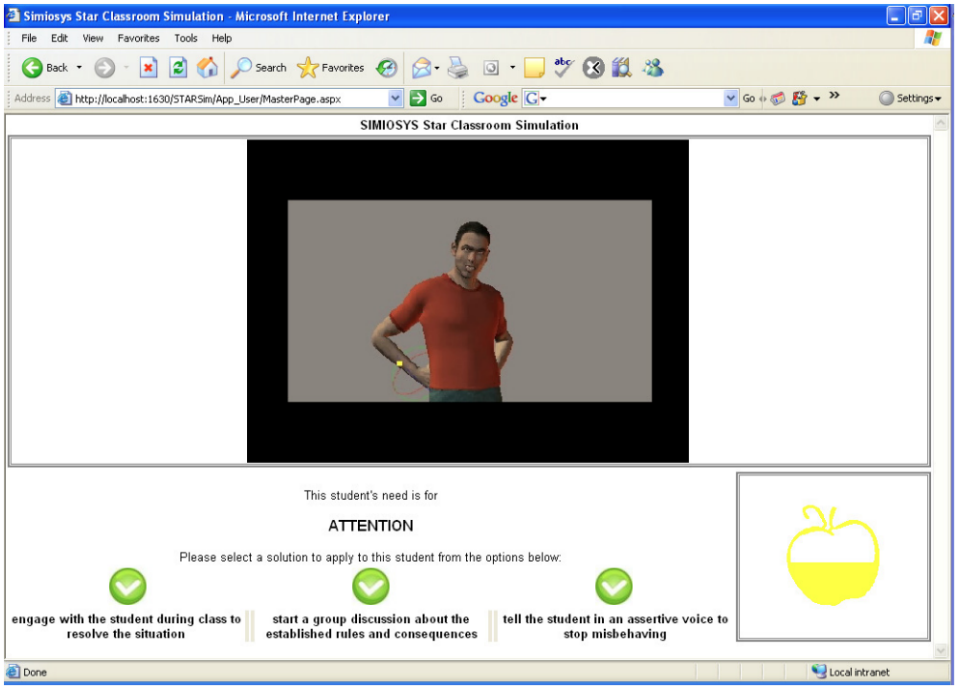


Figura 2.8: Captura de STAR Classroom Simulator tras realizar una acción.



Figura 2.9: Captura del tutorial de Portal 2, uno de los juegos usados por el proyecto Versa, concretamente para desarrollar la flexibilidad cognitiva.

Capítulo 3

Objetivos y especificación

El propósito de este proyecto es desarrollar un simulador ludificado para instructores de vuelo, que ayude a estos a reconocer comportamientos observables en los pilotos cuyas competencias están evaluando. Una parte importante de este proyecto es tener la capacidad de crear nuevos escenarios personalizados de forma consistente y eficaz, y otra que la aplicación final resulte útil para la formación de instructores reales.

3.1. Objetivos

Una vez conocido cuál es el estado de la cuestión, con las oportunidades y retos que esto implica, este proyecto se propone una serie de objetivos concretos:

- Modelar computacionalmente escenarios de simulación de vuelo basados en Entrenamiento Basado en Evidencia con suficiente realismo y profundidad como ser aceptados por los especialistas en aviación.

- Diseñar y desarrollar un simulador de instructores de vuelo donde experimentar, sobre los escenarios anteriormente mencionados, lo esencial de su trabajo de evaluación. Esto consiste en reconocer comportamientos observables de cada piloto que revelen si este ha adquirido o no las competencias no técnicas necesarias para realizar la misión que plantea el escenario de forma segura y eficaz.

- Producir una aplicación final compatible con dispositivos móviles que permita validar su funcionalidad y usabilidad con usuarios reales que tengan experiencia en la instrucción de pilotos de avión.

3.2. Especificación

En las próximas subsecciones se realiza una especificación inicial de los diferentes aspectos que deberá tener el simulador.

3.2.1. Interfaz de usuario

Al ser un simulador ludificado, la interfaz del simulador es algo importante. Esta interfaz debe ser intuitiva y dinámica. Para ello, cuando el usuario inicia el simulador debería verse el logo del simulador, seguido del menú principal. En el menú, para mantener un dinamismo, se vería un vídeo de un aeropuerto que se repite en bucle. Tal vídeo sería renderizado sobre modelos en 3D. En este menú, el jugador podrá acceder a la pantalla de selección de escenarios, a la de opciones, a la de créditos o salir del simulador.

En la pantalla de selección de escenario, verá en una pizarra todos los escenarios a los que podrá acceder, y podrá hacer *scroll* vertical para verlos todos. Habrá un botón para jugar a cada escenario y un botón para volver al menú principal en la parte superior de la pantalla.

Cuando el jugador está jugando un escenario, debería ver una cabina de avión desde detrás, observando a los pilotos desde la misma posición que un instructor evaluaría en una sesión real. Véase figura 3.1.



Figura 3.1: Croquis de la interfaz final deseada hecha en moqups.

Podrá ver un botón para volver al menú en la parte superior y un recuadro con la etapa de vuelo en la que se encuentra. Cuando se evalúe a uno de los pilotos se dará *feedback* inmediato sobre dicha evaluación, pues aparecerá un -1"de color verde si era correcta o un 1rojo si era incorrecta.

Si la situación lo requiere, aparecerá un botón para abortar la misión en la parte superior de la pantalla.

Una vez terminado cada escenario, el usuario verá una pantalla en la que aparecerá una clase y podrá ver el porcentaje de aciertos y de fallos, además de una frase indicando si ha superado el escenario. Aparecerá también un botón para volver al menú de selección de escenarios.

3.2.2. Escenarios de entrenamiento

La construcción de escenarios en el simulador se implementará de forma que los escenarios sean ficheros de texto que el simulador pueda leer e interpretar para desarrollar el escenario. Estos escenarios estarán compuestos principalmente de información sobre el contexto de la misión, y una serie de **pasos** o eventos, que ser irán produciendo secuencialmente.

Pasos

Todos los **pasos** tienen unas características comunes. Estas son:

- **Autoría**. Indica quién hace el paso. Puede ser tanto el piloto, el copiloto o incluso la torre de control, que puede dialogar con los pilotos. La torre de control se utiliza en el tutorial como personaje narrador que da instrucciones al jugador, y también proporciona datos básicos de la misión.
- **Tiempo** y **duración**. Indica cuándo reproducir este paso y por cuánto tiempo. Algunos **pasos** pueden solaparse en un mismo espacio de tiempo y estar reproduciéndose a la vez, como es el caso de las animaciones.
- **Resultado** y ***OB***. Indican qué comportamiento observable (*OB*) está representando el actor en ese paso y si es algo positivo, negativo o neutral. No siempre tienen por qué tener un *OB* asociado, pues muchas veces los pilotos realizan acciones comunes que no representan ningún *OB*.
- **Modificador**. A veces es necesario ejecutar código especial para algún paso. Tanto para resaltar algo en pantalla como para detener la ejecución del escenario. Sobretodo se usa en el tutorial.

Los pasos que hemos considerado que cualquier piloto necesita hacer y son necesarias para cualquier escenario y hemos implementado son las siguientes:

- **Diálogo**. Contiene el diálogo escrito que el actor del paso va a decir, en forma de texto. No se usa audio por el tiempo extra requerido a la hora de generar un escenario, pues haría falta grabar decenas de diálogos con diferentes voces para cada escenario. Por ello, pensamos que poner un bocadillo con texto es la mejor opción para nuestro simulador ludificado.

- **Interacción con un sistema del avión**. El actor del paso cambia el estado de alguno de los sistemas del avión, activándolos o desactivándolos. Para simplificar las animaciones y no recrear fielmente cada cabina de avión con todos sus sistemas, y para que se vea más cómodamente, aparecería un pequeño vídeo genérico con texto por encima que indique de qué sistema se trata.

- **Lenguaje no verbal**. El actor reproduce una animación como puede ser estar tranquilo, aparentar nerviosismo o señalar alguna nube.

Además, fue necesario implementar más tipos de pasos para recrear los eventos externos de cualquier escenario:

- **Cambio de etapa de vuelo**. Dado que un escenario puede darse durante varias etapas de vuelo, este paso es necesario para indicar que la fase del vuelo ha cambiado y la situación ya no es la de antes.

- **Cambio de clima**. Cambia el entorno para simular lluvia, niebla, soleado u otros climas que pueden alterar el escenario.

- **Alerta sonora**. Se reproduce una alerta sonora que indica el estado del avión o de su trayectoria.

- **Abortar escenario**. Da la oportunidad al instructor de terminar la sesión en ese instante.

El paso de **Abortar misión** es importante ofrecerlo en cierto momento, aunque su uso sea opcional. Algunos instructores, cuando se da una situación en el simulador donde los pilotos ya no pueden hacer nada para salvar el avión, deciden acabar la sesión y volver a un estado donde se puede arreglar. No obstante, también existen instructores que opinan que la mejor manera de enseñar es, aunque ya no se pueda salvar el avión, dejar a los pilotos experimentar un accidente ficticio.

Por esta razón, en nuestro simulador ludificado debería ser opcional abortar el escenario, pues nosotros no pretendemos dictar la filosofía a seguir para instruir a los pilotos.

3.2.3. Sistema de evaluación

Para evaluar a los pilotos, el jugador deberá seleccionar el piloto a evaluar, o su acción. Si el actor está hablando, el jugador tendrá la opción de seleccionar el bocadillo del diálogo; si está interactuando con el sistema del avión, el jugador también podrá pinchar sobre dicha acción. Además, el jugador siempre podrá pinchar en el propio piloto. Al hacer cualquiera de estas acciones, se empieza la evaluación del piloto.

Al empezar la evaluación, aparecen seis posibles respuestas sobre el piloto que **no** se ha seleccionado. Esto se debe a que es necesario que el jugador pueda observar al evaluado sin ningún estorbo en la pantalla.

Como es necesario leer las seis opciones antes de elegir alguna opción, el tiempo se detendrá. En una sesión real el tiempo nunca se detiene, pero las sesiones pueden durar horas, dando tiempo entre paso y paso a que el evaluador anote en su tableta, o en su libreta, el comportamiento observable que cree haber visto. En nuestro simulador ludificado, las sesiones duran minutos, por lo que al compactar los pasos de los pilotos, tener que leer las seis opciones que ofreceremos cuando el jugador piensa evaluar, congelar el tiempo es algo necesario.

No obstante, no se congela el tiempo para todos las entidades del simulador para mantener dinamismo durante la partida.

3.2.4. Tutorial

Para enseñar al jugador las mecánicas básicas del simulador y sus metáforas, se desarrollará un escenario especial que servirá de **tutorial**.

En este tutorial sería, a diferencia de una partida normal, avanzado a base de pulsar un botón de continuar. Esto se haría para que se pudiese leer con comodidad y tranquilidad. También se resaltarían los aspectos importantes de la pantalla para guiar los ojos del usuario.

El tutorial no será obligatorio, pues queremos que el usuario tenga total libertad de la ejecución de su partida y del simulador ludificado.

Además, para exponer la capacidad que tiene nuestro simulador ludificado para entrenar a los evaluadores, se hará un escenario en colaboración con expertos como una demostración de los escenarios que se pueden llegar a crear. El escenario se basará en el accidente real en 2018 del vuelo Air Nuigini 73. (Commission, 2018)
[1]

[1]https://www.abc.net.au/news/2019-07-18/air-niugini-crash-flight-73-pilots-ignored-alerts/11322486

Capítulo 4

Metodología y herramientas

En este capítulo se habla del modelo de proceso y planificación que se ha llevado a cabo para la correcta realización del proyecto, las iteraciones que ha tenido el proyecto y todas las herramientas utilizadas para poder ejercer esta metodología.

4.1. Metodología de desarrollo

Nosotros hemos hecho uso de metodología ágiles, cuales hemos aprendido en la asignatura de *Metodologías ágiles de Producción* y aplicando extensivamente a lo largo de nuestra carrera, sobre todo en las asignaturas *Proyecto 1*, *Proyecto 2* y *Proyecto3*.

Por definición, las metodologías ágiles son aquellas que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno. [1]

Las ventajas que ofrecen las metodologías ágiles sobre las clásicas son varias, como la mayor satisfacción del cliente porque está más envuelto en proceso, haciendo que el proyecto no diste de su visión. Las tomas de decisiones son más rápidas e intenta minimizar el *crunch* al acabar los hitos.

Nosotros hemos usado *SCRUM*, pues es con la que tenemos más experiencia trabajando. Para ello se han hecho reuniones cada dos semanas con nuestro tutor, además de tener reuniones con expertos de forma más esporádica y mantenido un *sprint-log* y un *back-log*. De estas particularidades hablaremos más adelante.

Por limitaciones naturales del curso académico, hemos estructurado la realización del proyecto en tres grandes etapas, llamadas hitos, las cuales han comprendido los periodos: desde comienzo de curso en septiembre a Navidad,

[1] https://gestion.pe/economia/management-empleo/son-metodologias-agiles-ayudar-organizacion-nnda-nnlt-263341-noticia

desde Navidad a Semana Santa, y desde Semana Santa hasta hasta el Hito Final (que incluye la revisión y la posterior defensa del Trabajo Fin de Grado) ya en mayo. Cada uno de estos periodos tuvo una duración aproximada de tres meses.

Dentro de cada hito, se han elaborado tareas cada *sprint*, teniendo además una reunión con nuestro tutor. Los *sprints* son periodos cortos de tiempo, normalmente una semana o dos (Véase Figura 4.1). Nosotros decidimos hacerlas de dos semanas de duración. Por lo tanto, teníamos una reunión con nuestro tutor cada dos semanas de forma general. A veces hacíamos una reunión a mitad de *sprint* para aclarar algunas dudas. En estas reuniones, nuestro tutor revisaba lo que habíamos hecho y juntos debatíamos las tareas que debíamos hacer para el siguiente *sprint*. Una vez acabada la reunión, se añadían las tareas del *back-log* al *sprint-log* y uno de nosotros se adquiría la autoría de la tarea para realizarla.

Un *back-log* en *SCRUM* consiste en un montón de tareas, o historias de usuario, donde se describen los aspectos que debería tener el proyecto al acabar el hito. Un *sprint-log* son las tareas o historias de usuario a realizar en el *sprint*.

Durante los distintos hitos se han tenido objetivos distintos:

- **Hito de Navidad**. Nuestro objetivo en este hito fue la recaudación de información, desarrollar la idea y desarrollar un pequeño prototipo con la lógica final que funcione directamente en consola. Se ha investigado sobre trabajos relacionados, tanto en el mundo de la aeronáutica como en el de los juegos serios. En este hito se definió la interfaz con pequeñas pruebas con nuestros expertos.
- **Hito de Semana Santa**. Este hito estuvo orientado a desarrollar la herramienta y dejarla lo más acabada posible.
- **Hito Final**. Este hito hemos arreglado errores en nuestra herramienta, y sobre todo realizado test con usuarios.

4.1.1. Herramientas utilizadas

Para seguir esta metodología y el correcto desarrollo de este proyecto, se ha hecho uso de diferentes programas y herramientas. Ello nos ha permitido registrar las tareas, tener el proyecto actualizado o mantener una comunicación directa con nuestro tutor y los expertos.

GitHub

GitHub [2] es una plataforma de desarrollo colaborativo para alojar repositorios de Git. Git es un herramienta de control de versiones que hemos usado

[2] https://github.com/

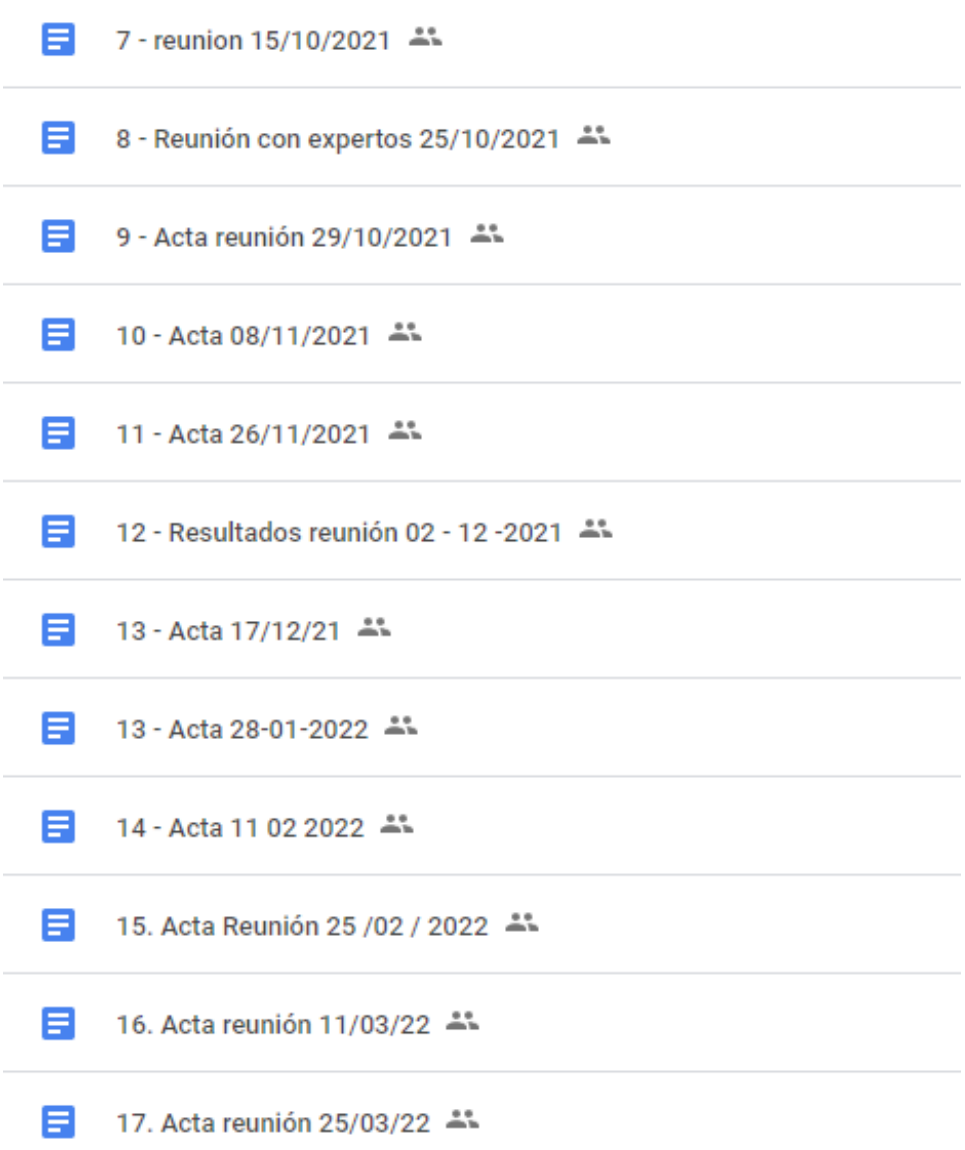


Figura 4.1: Parte de las actas realizadas durante el desarrollo.

ampliamente y que aprendimos a usar en la asignatura de *Metodologías Ágiles de Producción*. En GitHub, subimos nuestro proyecto de consolas y de Unity, donde se fueron actualizando continuamente a lo largo del desarrollo mediante *commits*. El proyecto puede encontrarse en: https://github.com/Narratech/TFG-Cruz-Guerra

Git se puede usar por consola, pero por comodidad nosotros usamos GitHub Desktop como interfaz gráfica. Este programa tiene Git completamente integrado, pudiendo hacer pleno uso de Git de forma sencilla.

GitHub Projects

GitHub Projects es una herramienta que ofrece GitHub (Véase 4.1.1) para organizar tareas. Este utiliza un sistema de tarjetas virtuales que se mueven desde el *back-log* hasta finalmente al *done*.

Drive

Drive[3] es un servicio de alojamiento de ficheros de Google en el que se pueden compartir entre todos los integrantes del equipo. Lo hemos utilizado para almacenar archivos pesados que no deberían ser subidos al repositorio Git, archivos propios de la metodología, como pueden ser documentos de reuniones, archivos propios de la investigación y algunos bocetos de interfaces.

Para la creación de estos ficheros de texto se ha hecho uso de Documentos de Google, dada su rápida sincronización en la nube y su capacidad de trabajar varios a la vez en el mismo documento. Estos documentos han sido usado sobre todo para elaborar las actas de reuniones.

Forms

Forms[4] es una herramienta de Google para crear cuestionarios y visualizar las respuestas dadas en forma de gráficos. Lo hemos usado para crear el cuestionario dado a los expertos en las pruebas con usuarios.

Photoshop

Photoshop[5] es un software de edición de imágenes creado por Adobe. Lo usamos para crear algunas de las imágenes del simulador, como los botones de Play o la animación de la radio, ya que también permite la creación de archivos .gif con animaciones.

Blender

Blender[6] es un software gratuito de creación de mallas 3D, animación, simulación y renderizado. Lo usamos para modificar las texturas de los modelos de los pilotos, para bajar la resolución de ciertas mallas y para crear los modelos, animaciones y vídeo de, por ejemplo, la pulsación de un botón en la cabina.

[3] https://www.google.com/intl/es/drive/
[4] https://www.google.es/intl/es/forms/about/
[5] https://www.adobe.com/es/products/photoshop.html
[6] https://www.blender.org/

Unity

Unity [7] es un motor de videojuego multiplataforma creado por Unity Technologies. Unity está disponible como plataforma de desarrollo para Microsoft Windows, Mac OS, Linux. La plataforma de desarrollo tiene soporte de compilación con diferentes tipos de plataformas.

Unity nos ha permitido crear un simulador ludificado de manera sencilla, tanto por sus capacidad como motor como por nuestra previa experiencia usándolo en el grado.

Para mejorar el apartado visual hemos usado varios módulos específicos de Unity, como el módulo de animaciones. Ejemplos de este uso serían el *feedback* que aparece tras evaluar, que está hecho con el *Animator*; o el menú principal, que está hecho con el *Timeline* (una herramienta de Unity para crear cinemáticas) y grabado con el módulo de grabaciones para mejorar la eficiencia en dispositivos móviles, pues los *assets* utilizados tenían demasiada calidad para que estos dispositivos los simularan de manera fluida.

También usamos uno de los sistemas de partículas propios de Unity, *Shuriken*, para simular la lluvia. Este sistema de partículas usa la CPU para la simulación, pero permite la compatibilidad con dispositivos móviles.

Usamos también los *Scriptable Objects* para generar assets de configuración del tiempo atmosférico.

Además, para serializar y deserializar ciertos elementos de la lógica, como los escenarios, hemos usado la librería de *JSON*.

Visual Studio

Visual Studio [8] es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para Windows y macOS. Es compatible con múltiples lenguajes de programación al igual que entornos de desarrollo web.

Lo hemos usado para programar el pequeño proyecto de consola y sobre todo para programar en C# los scripts de *Unity*.

Overleaf

Overleaf [9] es un editor de LaTeX online. LaTeX es un sistema de composición de textos, orientado a la creación de documentos. Lo hemos utilizado para la construcción de la memoria del TFG.

[7] https://unity.com/
[8] https://visualstudio.microsoft.com/es/
[9] https://www.overleaf.com/

Slack

Slack [10] es una herramienta de comunicación en equipo. Es una herramienta muy usada por Narratech. Nosotros la utilizamos para comunicarnos con el tutor y los expertos, preguntar dudas y organizar las reuniones.

Google Meet

Google Meet [11] es un servicio de videotelefonía desarrollado por Google. Google Meet permite hacer videollamadas online sin necesidad de instalarse nada excepto un navegador.

Hemos hecho uso extenso de Google Meet durante el desarrollo del trabajo. Todas las reuniones con nuestros tutores, como las reuniones con los expertos, las hemos hecho a través de Google Meet.

[10] https://slack.com/intl/es-es/
[11] https://meet.google.com/

Capítulo 5

Diseño y desarrollo de Flight Instructor Simulator

En este capítulo se detallan detalles técnicos sobre el simulador ludificado en cuestión, el cuál tiene como nombre **FI-SIM**, siendo estas las siglas en inglés de *Flight Instructor Simulator*. También se explica el análisis, diseño e implementación en *Unity*.

Para ello, se detalla la arquitectura diseñada, y la implementación de algunas de las clases y métodos más relevantes para entender las decisiones tomadas en la creación del sistema interno de la aplicación. También se explican todas las escenas creadas en Unity para entender el contenido que ofrece el prototipo del simulador en su estado actual.

5.1. Análisis y diseño

A continuación se describe la arquitectura a más alto nivel de Flight Instructor Simulator.

5.1.1. Navegación

Para el desarrollo de una partida completa de Flight Instrutor Simulator, el usuario debería poder navegar entre escenas de manera completa, pudiendo acceder a todas las escenas del simulador con suficiente fluidez. Esta navegación se describe en la siguiente figura 5.1. En dicha figura se puede apreciar que el tutorial como una partida normal son escenas diferentes. La razón de esto se explica a continuación.

5.1.2. Diseño de la arquitectura

A continuación, mostramos en las figuras 5.2 y 5.3 la arquitectura del simulador, tanto de la lógica como de los componentes de Unity. En la sección

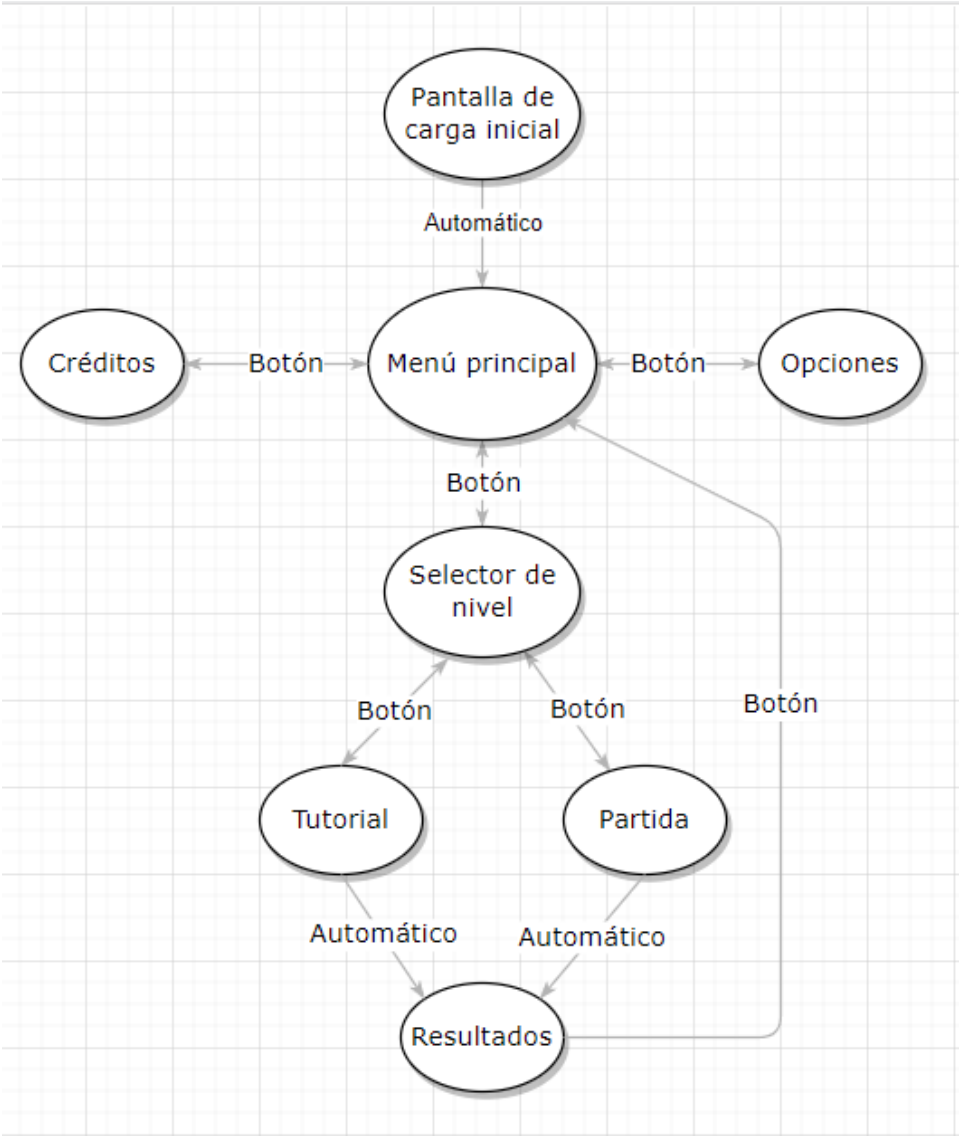


Figura 5.1: Diagrama de la navegación entre escenas.

posterior se explica con más detalle cada clase y su funcionalidad.

5.2. Implementación en C#

A continuación, se explican las clases de código en C# que componen los aspectos más básicos de la funcionalidad del simulador ludificado.

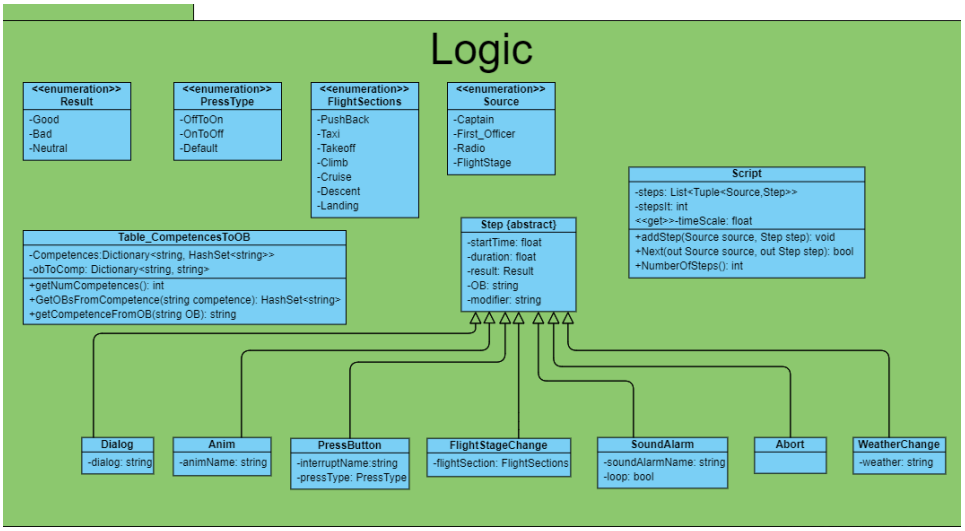


Figura 5.2: Diagrama de clases detallando la lógica del simulador.

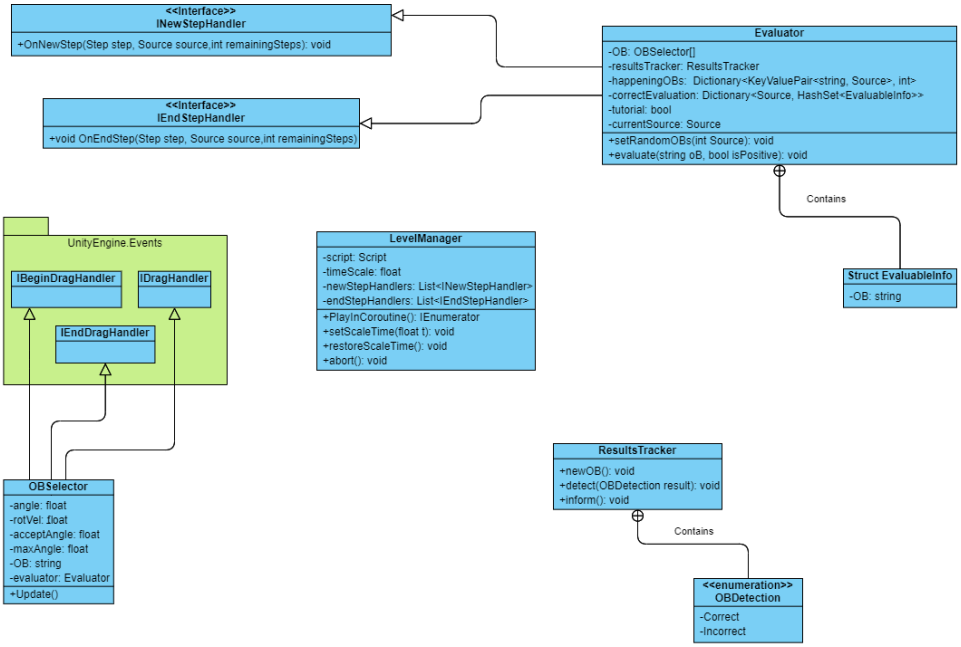


Figura 5.3: Diagrama de clases detallando los principales componentes de Unity.

5.2.1. *Step*

Esta clase es la clase base de las que todos los pasos heredan como hijos. Contiene información básica de todo paso, como se ha detallado previamente en

5.2.2. *Script*

Esta clase es la encargada de almacenar el escenario en memoria a partir de un fichero de texto *json*. Más adelante se explica qué clase es la encargada de leer el escenario y ejecutarlo.

Estos ficheros de texto guardan toda la información necesaria para que se pueda formar el escenario. Esa información es:

1. **Name**. Este campo de tipo *string* es meta información sobre el nombre del escenario.

2. **CaptainName**. Este campo de tipo *string* es meta información sobre el nombre del piloto.

3. **FirstOfficerName**. Al igual que el capitán, este campo de tipo *string* es meta información sobre el nombre del copiloto.

4. **CaptainVariant**. Este campo de tipo *string* es meta información sobre el modelo 3D a usar para representar al capitán. Además, finaliza con un número que indica la variante sobre el modelo 3D.

5. **FirstOfficerVariant**. Al igual que la variante del capitán, este campo de tipo *string* es meta información sobre el modelo 3D a usar para representar al copiloto. Además, finaliza con un número que indica la variante sobre el modelo 3D.

6. **TimeScale**. Este campo de tipo *float* es meta información que indica a qué velocidad debe ser ejecutado el escenario. Dado que cambiar todas las marcas temporales de todos los pasos del escenario podría resultar tedioso, si están bien proporcionalmente unas con respecto a otras, este campo facilita cambiar la velocidad de reproducción en caso de que no el jugador no disponga del suficiente tiempo para leer los diálogos y reaccionar para evaluar, o si en cambio pasa demasiado tiempo entre pasos.

7. **Pasos**. Este campo es el grueso del fichero, pues se trata de una lista que contiene todos los pasos. Dado que estos pasos contienen marcas temporales, pueden aparecer en desorden. Además de contener un *Step*, también contienen el actor que va a interpretar el paso. Es decir, contienen un número del 0 al 2 (inclusive). El 0 representa al capitán; el 1 al primer oficial y el 2 a la radio.

El fichero de texto del escenario de ejemplo que desarrollamos junto a expertos tiene una duración de cinco minutos, conteniendo un total de 868 lineas.

5.2.3. *TableCompetencesToOB*

Esta clase es la encargada de almacenar una tabla que relaciona cada competencias con sus *OB*. Mediante esta clase se pueden conseguir tanto los *OB* de una competencia como la competencia a la que pertenece un *OB*, ya que guardamos la información en los dos sentidos.

5.2.4. *LevelManager*

Este componente es el encargado de ejecutar los scripts. Para ello, lanza una corrutina que ejecuta el método principal, el cuál explicaremos brevemente a continuación.

Primero, dado que las parejas de pasos del *Script* pueden estar desordenadas, se ordenan los pasos en una cola de prioridad donde el orden se establece por la marca de tiempo.

Después se empieza el bucle principal, donde se espera hasta que algún paso empieza. Cuando esto sucede, se inserta en otra cola de prioridad para los pasos en ejecución, con la prioridad puesta con su marca de tiempo más la duración del evento. Es decir, con la marca de tiempo de finalización.

Cuando un paso empieza o acaba, se avisa todos aquellos objetos que se hayan apuntado a la lista de observadores. Este patrón se le conoce como *patrón observer/listener*. Los observadores podrán tratar entonces el paso.

Para ello, entre sus atributos se encuentran una lista estática de observadores para cuando se produce el paso y otra para cuando se acaba. Todos los pasos que se meten en la primera cola se van sacando cuando empiezan, y todos los pasos que se meten en la segunda cola se sacan cuando el paso finaliza.

Cuando ya no quedan pasos que ejecutar en ambas colas de prioridad, se concluye la ejecución del bucle principal y del escenario. Después se cambia la escena de Juego a la de resultados.

5.2.5. *Step Displayer*

Este componente es el encargado de escuchar al LevelManager y hacer que los actores interpreten los pasos.

Este componente tiene acceso a todos los actores, tanto a los pilotos como a la radio y al sistema. Cuando el LevelManager llama al método de OnNewStep, StepDisplayer distingue entre los pasos y llama a los métodos privados apropiados para reproducir el paso. Esto puede ser mostrar un diálogo, una animación, un cambio de etapa de vuelo, cambiar el tiempo atmosférico o activar la posibilidad de pulsar el botón de *Abort*



Figura 5.4: Editor en Unity de la escena de Juego.

5.2.6. *Evaluator*

Este componente es el encargado tanto de preparar las opciones para que el jugador elija, como de evaluar la elección del jugador. Para ello cuenta con un diccionario que vincula a los personajes con un set de respuestas correctas, ya que pueden estar sucediendo varios *OB* simultáneamente.

Cuando el jugador pincha en uno de los elementos que permiten evaluar (el personaje elegido, la burbuja de texto, etc) prepara las opciones observando cuáles son los *OB* que se están desarrollando en la escena y almacenando un diccionario que vincula las competencias de estos *OB* con la lista completa de *OB*s de esa competencia para elegir las opciones de ahí. En caso de que no esté sucediendo ningún *OB*, elige una competencia al azar. Una vez hecho esto, divide el número de posiciones del evaluator entre el número de *OB*s correctos que están sucediendo para seleccionar *OB*s falsos de las mismas competencias que los *OB*s correctos y distribuirlos de forma equitativa. También introduce en el diccionario personaje-set de respuestas correctas las opciones y respuestas (*OB*s positivos o negativos) correctas vinculadas al personaje que el jugador seleccionó.

A la hora de evaluar la selección del jugador mira si la opción y la respuesta seleccionados están en ese set.

5.2.7. *ResultsTracker*

Este componente es el encargado de contabilizar cuándo se produce un *OB* y cuándo el jugador evalúa, de forma que esta información dé lugar a los porcentajes de fallo y acierto mostrados en la pantalla de resultados.

5.2.8. *OBSelector*

Este componente lo tienen todas las opciones que conforman el evaluator y es el responsable de rotarlas cuando el jugador las arrastra. Cada una de las opciones tiene el punto de pivote muy por debajo del centro, lo que permite la rotación hacia los lados. Cada opción tiene un ángulo límite a partir del cual no se podrá rotar más hacia la derecha o hacia la izquierda, y cuentan también con un ángulo de aceptación, que es el mínimo que hay que girar para que la respuesta se considere seleccionada. Si el jugador no llega a este ángulo, la opción volverá a su posición inicial. En caso contrario, le mandará la selección al evaluator para que la evalúe.

5.3. *Escenas de Unity*

A continuación, se explican las diferentes escenas que componen el simulador.

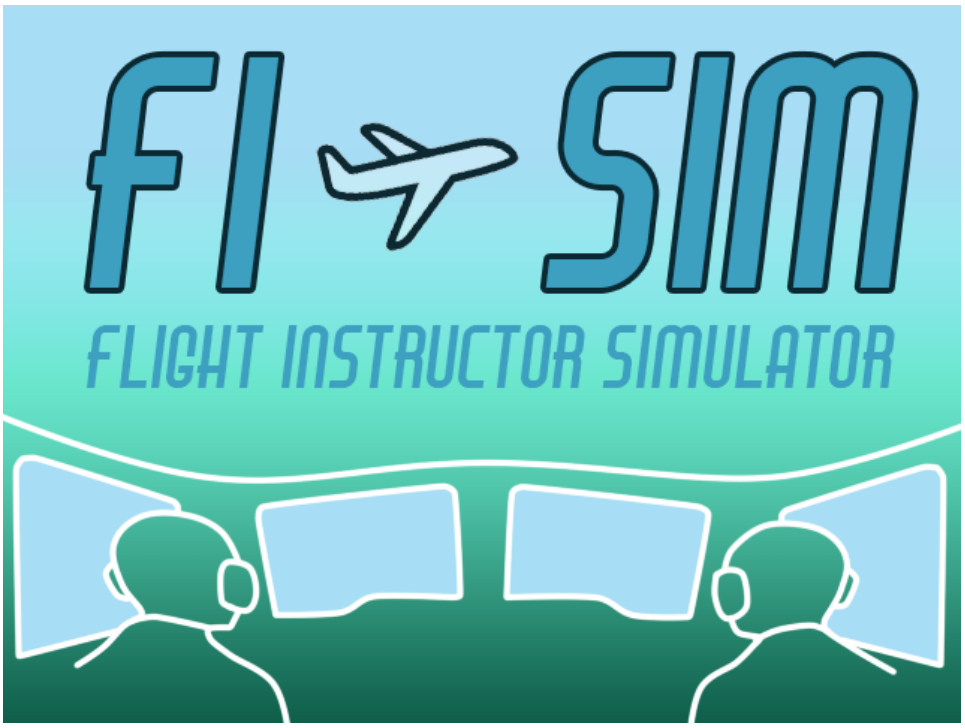


Figura 5.5: Logo del simulador.

1. *Cargador*. Esta escena solo se muestra al principio de la ejecución, pues se encarga de mostrar el logo del simulador (véase figura 5.5) y carga por detrás de manera asíncrona la siguiente escena: el menú principal.

2. *Menú principal.* Esta escena sirve como punto central de la aplicación. Contiene un vídeo de fondo prerenderizado de un aeropuerto a partir de otra escena que no se incluye en la build. Esto es debido a que los modelos y texturas eran demasiado pesados y no conseguíamos que los dispositivos móviles simularan la escena en tiempo real de forma fluida. Lo más importante, contiene unos botones que permiten el acceso a diferentes escenas, como a la escena de elegir nivel, la de opciones o créditos. También contiene un botón para cerrar el simulador.

3. *Opciones.* Esta escena es sencilla, pues no hay muchas opciones en el simulador. La única opción que ofrece cambiar es el volumen del simulador. De fondo, se muestra una pizarra de un aula renderizada. Tiene además un botón de retorno para volver al menú principal.

4. *Créditos.* Esta escena solo muestra un fondo de color negro y texto blanco superpuesto que indica quiénes han formado parte de la elaboración del simulador ludificado. Tiene además un botón de retorno para volver al menú principal.

5. *Seleccionar nivel.* Esta escena, accedida tras pulsar el botón de jugar del menú principal, muestra una pizarra con opciones. El jugador puede desplazarse por los escenarios en vertical para elegir la que él desee pulsando la flecha pequeña a la derecha de cada opción. Tiene además un botón de retorno para volver al menú principal. Cuando el jugador indica al simulador que desea el jugar un escenario, se pasa a la escena de Juego.

6. *Juego.* Es la escena encargada de la parte jugable del simulador. En esta escena se reproducirá el escenario y el jugador evaluará a los pilotos. Tiene además un botón de retorno para volver al selector de niveles. Cuando el jugador completa el escenario, se carga la escena de *resultados*. Para ver la escena normal, véase figura 5.6. Para verla desde el editor de Unity, véase figura 5.4

7. *Resultados.* Esta escena se muestra al acabar de jugar un escenario. En ella, se aprecia una animación en primera persona de sentarse en una silla de un aula. Superpuestos, se pueden ver los resultados del porcentaje de todos los *OB* que se ha evaluado positivamente y los que no. La escena tiene además un botón de retorno para volver al menú principal.

8. *Tutorial.* Esta escena es una copia de la escena de Juego con la diferencia de que tiene configurados algunos scripts de forma diferente para poder efectuar el tutorial. Un ejemplo de esto son los *OBSelector*, que solo permiten evaluar la opción correcta durante el tutorial.



Figura 5.6: Captura del simulador en mitad de partida.

9. *Scenario Creator*. Esta escena sirve para crear un guión añadiendo los diferentes pasos que se van a ejecutar tanto en el capitán, el primer oficial y la radio.

Capítulo 6

Resultados



Figura 6.1: Nosotros, nuestros directores del trabajo y David Rioja de Didactoons en Global Training Aviation Madrid.

En este capítulo trataremos el desarrollo de las pruebas con usuarios, los resultados obtenidos en dichas pruebas y discutiremos acerca de estos resultados, confrontándolos con lo que conocemos del estado de la cuestión.

6.1. Pruebas

Las pruebas se realizaron para comprobar la utilidad del simulador ludificado para representar un escenario de entrenamiento realista y versátil, a la vez que ameno y fácil de usar.

Se desarrolló un plan de pruebas en el que se organizaba la sesión de forma que cada uno de los dos autores hiciera el papel de entrevistador, dividiendo a los probadores en dos grupos para ser más eficientes. Primero, el usuario debía jugar el tutorial, pudiendo preguntar a los entrevistadores cualquier duda que tuviera. Después, el usuario jugaría a un nivel simple de demostración, ya preparado para familiarizarse con las mecánicas del simulador ludificado. Durante esta partida, el usuario también podía preguntar a los entrevistadores cualquier duda y estos tenían la posibilidad de ayudarle si detectaban algún bloqueo o confusión. Para finalizar el experimento con el simulador, el usuario debía jugar el escenario basado en un accidente real del Capítulo 3, ya de manera más autónoma. Al terminar la experiencia, el usuario debía rellenar un cuestionario, pudiendo elegir si hacerlo en español o en inglés, en el que respondía a preguntas centradas tanto la problemática de la formación de instructores como en la utilidad y usabilidad del prototipo mismo. Los dos entrevistadores debían anotar las opiniones exprsadas por los usuarios mientras usaba el simulador, y escribir acerca de cualquier otro problema o mejora que pudiera surgir.

Estas pruebas se realizaron de manera presencial, en las aulas de Global Training Aviation [1], con un total de 8 probadores, siendo todos especialistas en aeronáutica con experiencia como instructores de pilotos. Véase Figura 6.2.

Como se ha mencionado previamente, además de las anotaciones, se hizo uso de un cuestionario. Las diferentes secciones de tal cuestionario fueron las siguientes:

1. **Opinión sobre al ámbito tratado.** Este apartado contenía preguntas sobre la formación actual en Aeronáutica y la aplicación del paradigma EBT.

2. **Opinión sobre el juego y sugerencias sobre usabilidad.** Este apartado contenía preguntas sobre conceptos del simulador ludificado, las mecánicas y dinámica de juego, e incluso el contenido informativo de los escenarios y su representación estética con los recursos audiovisuales utilizados.

3. **Sugerencias generales sobre el proyecto.**

El cuestionario completo en español puede encontrarse en el Apéndice D.

[1] https://www.globaltrainingaviation.com/en



Figura 6.2: Uno de los expertos jugando el tutorial.

6.2. Resultados

La información detallada acerca de los resultados obtenidos en las pruebas están disponibles en el Apéndice E, pero a continuación vamos a resumir lo más relevante.

Como ya se mencionó previamente, todos los usuarios de las pruebas eran instructores de vuelo reales. Sus opiniones ante la formación actual en pilotaje fueron dispares. La mayoría opinó que es suficiente, sin embargo, una porción no despreciable de los encuestados (25 %) opinó que estaban en desacuerdo y que veía carencias o cuestiones mejorables.

Todos declararon útil el paradigma EBT, pues opinaron o bien que estaban "de acuerdo" con este paradigma o que estaban "totalmente de acuerdo". No obstante, una mayoría (75 %) contestó también que no estaban seguros sobre si la formación actual de los instructores de vuelo era suficiente para desempeñar correctamente su trabajo. Esto refuerza nuestra idea de que hay una oportunidad clara para que un proyecto como el nuestro facilite y mejore esa formación de los profesionales en este área de la Aeronáutica.

Sobre las preguntas en torno al simulador ludificado, la gran mayoría (75 %) entendió sin problemas el concepto de proyecto que les presentábamos.

Aun así, mostraron sus opiniones negativas en varias preguntas sobre la forma en la que el simulador representaba la información. Tanto en el cuestionario como en los apuntes manuscritos, se veía reflejado que la simulación se reproducía por error a más del doble de velocidad del que debería. Esto dificultó la lectura y comprensión de los diálogos, causando que no se pudiese reaccionar debidamente a lo que el escenario ofrecía.

Por otra parte, la mayoría de los usuarios nos transmitieron que les faltaba información para evaluar de forma justa a los pilotos del escenario. Algunos ejemplos de mejoras que nos sugirieron con respecto a esto fueron tener un *briefing* antes de empezar el escenario para que el evaluador sepa a qué se van a enfrentar los pilotos del escenario, que se muestre el estado del avión representándolo a tiempo real junto con diferentes mediciones que haría también un avión real, o que el evaluador sepa la experiencia previa del piloto. Esta última objeción es de particular interés, pues uno de los expertos nos aseguró que se cambia la manera de evaluar dependiendo del nivel de experiencia del piloto. Por llevar a cabo la misma acción, un piloto con poca experiencia puede obtener un 4 sobre 5, pero un piloto más experimentado puede obtener un 2 sobre 5.

Además, nos recomendaron que cada escenario se centrara en una o dos competencias, consiguiendo de esta forma dosificar mejor el conocimiento, lo cual se traduciría en un mayor rendimiento a la hora de ponerlo en práctica.

Otra mención más cualitativa que dieron varios expertos fue que, a la hora de evaluar y elegir solamente una opción, no sabían cuál elegir pues según ellos había varias opciones igual de válidas y argumentables.

El trabajo se puede encontrar desde la pagina de Narratech[2]

6.3. Discusión

La idea del trabajo desde el principio fue crear un simulador ludificado capaz de ayudar en la detección de comportamientos observables para el que fuese sencillo crear escenarios personalizados. Conviene realizar ahora una pequeña disertación sobre si la aplicación software que hemos creado podría considerarse un juego serio (concretamente un videojuego educativo) o, como nosotros defendemos, principalmente un simulador, aunque haya sido ludificado para hacer más ameno y atractivo su uso, al no tener la expectativa de que este software vaya a formar parte del proceso obligatorio de formación de los pilotos e instructores de vuelo, como ocurre con otros simuladores.

A primera vista, es fácil defender que estamos ante un simulador, ya que la interacción del jugador con el programa es análoga a la que un instructor realiza cuando está evaluando a un piloto. No hay muchas acciones posibles y básicamente se trata de tomar decisiones, lo cual no es una tarea especialmente divertida. Es cierto que no es un simulador "al uso", enfocado en la simulación física del avión ni en los posibles contratiempos y desafíos que requieren habilidad y control que pueden surgir durante una sesión de vuelo, sino que es un simulador más enfocado a las habilidades sociales de un instructor. En cuanto a la ludificación, destacamos que sobre este simu-

[2]https://narratech.com/es/game/flight-instructor-simulator/

lador se han incorporado paradigmas propios de los videojuegos, como la retroalimentación inmediata que damos al usuario al terminar cada sesión, el enfoque que reciben los tutoriales, la propia presentación cinemática que damos al simulador, como si de un juego de aviones se tratase, etc.

Ahora que hemos redefinido el resultado de nuestro trabajo, podemos decir que a pesar de que en el proyecto se han cometido algunos fallos, tanto de planteamiento inicial como de ejecución, los resultados obtenidos han demostrado ser esperanzadores y podrían tener utilidad de cara a su uso en cursos de formación de instructores. Nunca fue nuestra intención hacer ninguna contribución teórica al mundo de los protocolos aeronáuticos, pues el trabajo está basado en competencias y comportamientos ya existentes y bien documentados, pero sí ofrecer una herramienta que permita a esta comunidad reflexionar sobre sus propios procesos.

El escenario que creamos muestra las capacidades, y futuras posibilidades, que tiene este proyecto para crear escenarios útiles para mejorar la formación de los instructores de vuelo. La capacidad de expansión del proyecto es grande, pues con toda la información recibida por parte de los expertos sabemos qué es lo que puede mejorar y hasta dónde puede llegar.

En nuestro simulador ludificado, las evaluaciones se realizan en el acto. Aun facilitando la detección inmediata de un comportamiento observable y favoreciendo la retroalimentación (pues es inmediata también), tiene un contraargumento. Evaluar los comportamientos observables aplicables a una competencia, como el liderazgo, por ejemplo, es una tarea complicada, y además hacerlo de forma rápida y sin saber cómo van a reaccionar los pilotos a continuación es un reto irreal añadido al que los instructores en sesiones reales no se enfrentan, pues evalúan al finalizar la sesión tomando ayuda de apuntes realizados durante la misma. Esta decisión de diseño se puede implementar, haciendo que el jugador evalúe a los pilotos al finalizar la partida. De esta forma se mantendría una experiencia más cercana a la real. Esta forma de evaluación para nuestro simulador debería ser estudiada en un futuro.

También cabe destacar que según los instructores, la evaluación es relativa según la experiencia del piloto. Esto también se alinea con enseñar información básica del piloto y del escenario antes de la sesión, pues daría un esquema al jugador de a qué se va a enfrentar. Incluso también mostrando las competencias en las que debe fijarse y evaluar, con el fin de hacer una evaluación más profunda. Implementar esta característica sería trivial, aunque añadiría una nueva capa de desarrollo a la hora de diseñar un escenario.

Los expertos aseguraron una falta de información a la hora de evaluar. Esto se debe en parte a que se evalúa en el momento y no al finalizar la sesión, pero también porque en nuestro simulador no se pudo representar todas las métricas e información que un instructor podría ver en un sesión de evaluación real. Implementar en el simulador una cabina fiel a la real y que consecuentemente proporcione toda la información que debería, es costoso.

Por ello, una solución parcial es ir añadiendo más información sobre el estado del avión simulado iterativamente, hasta que la abstracción sobre un avión real sea suficientemente buena.

Capítulo 7

Conclusiones

En este capítulo final ofrecemos las conclusiones del trabajo realizado, valorando si los objetivos propuestos en el capítulo 3 se han cumplido o no. También se proporciona una línea de trabajo para el futuro desarrollo de este proyecto.

Los objetivos expuestos previamente en el capítulo 3 fueron:

- Modelar computacionalmente escenarios de simulación de vuelo basados en Entrenamiento Basado en Evidencia con suficiente realismo y profundidad como ser aceptados por los especialistas en aviación.
- Diseñar y desarrollar un simulador de instructores de vuelo donde experimentar, sobre los escenarios anteriormente mencionados, lo esencial de su trabajo de evaluación. Esto consiste en reconocer comportamientos observables de cada piloto que revelen si este ha adquirido o no las competencias no técnicas necesarias para realizar la misión que plantea el escenario de forma segura y eficaz.
- Producir una aplicación final compatible con dispositivos móviles que permita validar su funcionalidad y usabilidad con usuarios reales que tengan experiencia en la instrucción de pilotos de avión.

7.1. Objetivos alcanzados

A continuación se analiza lo alcanzado con respecto a cada objetivo:

- **Modelar computacionalmente escenarios de simulación de vuelo basados en Entrenamiento Basado en Evidencia con suficiente realismo y profundidad como ser aceptados por los especialistas en aviación**.

  Este objetivo fue cumplido como los resultados de las pruebas con expertos demuestra, pues gracias a la ayuda del reporte oficial del ac-

cidente (Commission, 2018), además de la ayuda de los expertos, pudimos realizar un escenario de 6 minutos en el que los diálogos de los pilotos son muy cercanos a los reales, y las evaluaciones que se deberían dar fueron hechas con la ayuda de varios expertos.

Es verdad que al escenario se le podría añadir más elementos para representar mejor el entorno, pero de esa cuestión se habla más en la sección 7.2.

- **Diseñar y desarrollar un simulador de instructores de vuelo donde experimentar, sobre los escenarios anteriormente mencionados, lo esencial de su trabajo de evaluación. Esto consiste en reconocer comportamientos observables de cada piloto que revelen si este ha adquirido o no las competencias no técnicas necesarias para realizar la misión que plantea el escenario de forma segura y eficaz**.

  Aunque hemos conseguido replicar lo esencial del trabajo de un instructor (dejando elegir al jugador cuándo, a quién y qué comportamientos evaluar), todavía hay margen de mejora, ya que el *feedback* que nos aportaron los expertos dice que en algunos casos la evaluación correcta es discutible, pudiendo darse el caso de que varias opciones o todas se podrían valorar como correctas.

- **Producir una aplicación final compatible con dispositivos móviles que permita validar su funcionalidad y usabilidad con usuarios reales que tengan experiencia en la instrucción de pilotos de avión**.

  El simulador ludificado funciona perfectamente tanto en Android como en iOS, debido a las facilidades que ofrece *Unity* para exportar proyectos multiplataforma. Para compilar y probar el simulador en Android, simplemente se requieren las herramientas de compilación y un dispositivo con sistema operativo Android. No obstante, para compilar y hacer pruebas en iOS hace falta un paso extra, que solamente puede realizarse en un ordenador con MacOS (Sistema Operativo de Apple para sus ordenadores Mac). Para realizar este paso, fue necesaria la ayuda de Didactoons [1], empresa que nos permitió utilizar sus instalaciones durante dos días para instalar el simulador en un iPad desde uno de sus Mac y probarlo.

  Lamentablemente, el día anterior a las pruebas con usuarios, descubrimos que los permisos para utilizar la aplicación en el dispotivio iPad habían caducado, y, por lo tanto, no pudimos hacer las pruebas utilizando un dispositivo iOS, por lo que usamos una Surface en su lugar.

[1] https://www.didactoons.com/

7.2. Trabajo futuro

Aunque creemos haber conseguido una aplicación sólida, todavía hay espacio para un posible trabajo futuro, pues al principio del desarrollo se planteó una característica bastante interesante que no se llegó a desarrollar del todo.

Esta consistiría en que los pilotos tuvieran un nivel de habilidad para cada competencia y que en los escenarios se pudiera dar un nivel de dificultad a cada *OB*. Esto permitiría que algunos pilotos pudieran mostrar un comportamiento negativo en ese *OB* y otros, por el contrario tendrían un comportamiento positivo. Esto implicaría diseñar escenarios en los que cada *OB* tuviera dos ramificaciones y también habría que modificar un poco la arquitectura del proyecto.

También, a la hora de aportar más información al jugador, se podría valorar implementar un pequeño *briefing* del escenario con información sobre el nivel de experiencia de los pilotos antes de jugarlo y también que en la escena del juego aparezcan unos gráficos indicando el estado actual del avión. Esto ayudaría a interpretar la situación y su contexto para evaluar mejor a los pilotos.

Como se mencionó previamente en el Capítulo 6 en el apartado de discusión, se podría valorar estudiar e implementar la evaluación al finalizar la sesión, aunque esto disminuya algunos aspectos lúdicos del simulador como la retroalimentación instantánea de una evaluación a mitad de sesión.

7.3. Reflexión final

Este proyecto ha tenido especial valor para nosotros, pues el esfuerzo que hemos realizado, tanto en el desarrollo del simulador como en conocer más sobre el mundo de la instrucción en la aeronáutica, ha resultado en un prototipo de simulador que, según los propios expertos con los que nos hemos reunido, tiene mucho potencial.

Durante todo el desarrollo de este trabajo de fin de grado hemos adquirido muchos conocimientos, mas allá de la informática, y conocido a multitud de personas encantadoras que se han ofrecido a ayudarnos y a compartir su conocimiento y su tiempo.

Apéndice A

Contribuciones individuales

Aquí expondremos las tareas más relevantes realizadas por cada uno de los integrantes. Es necesario mencionar que en muchas de estas tareas se colaboró mutuamente, dada por la complejidad de la tarea o por sencillamente la organización, se requería la ayuda del otro integrante para la realización de la tarea.

A.1. Antonio Jesús Garduño Guerra

- Producción. Tareas relacionadas con la producción de este proyecto.
  - Todas las reuniones con los directores del proyecto.
  - Organización de algunas de las reuniones con expertos.
  - Redacción en las actas de las reuniones.
  - Planificación de los objetivos a conseguir cada sprint.
  - Preparación de las presentaciones con el grupo de investigación cada hito, a parte de la exposición final
  - Creación del cuestionario para las pruebas de usuarios.
  - Traducción del cuestionario a inglés.
- Memoria. Tareas relacionadas con la redacción de esta memoria.
  - Búsqueda de trabajos y juegos serios relacionados.
  - Parte de la inserción de citas y pies de página.
  - Búsqueda de posibles escenarios en los que basarse para la realización del escenario final.
  - Cáscara Agradecimientos.
  - Cáscara Resumen
  - Capítulo 2, Formación de competencias no técnicas en aeronáutica y Juegos serios como apoyo a la formación.
  - Capítulo 3, parte de los objetivos y el apartado de Interfaz, que los hicimos juntos.
  - Capítulo 4, correcciones del capítulo.
  - Capítulo 5, subsecciones Evaluator, Scenario Creator y Tutorial.
  - Capítulo 6, correcciones varias.
  - Capítulo 7, parte de las conclusiones acerca de los objetivos.
- Juego. Tareas relacionadas con la programación del simulador ludificado.
  - Creación de bocetos de la Interfaz de Usuario.
  - Diseño de la lógica del prototipo del programa de consola.
  - Búsqueda de animaciones en Mixamo.
  - Adaptación de las animaciones a los modelos de los personajes.
  - Variantes de los personajes
  - Parte del guionizado del tutorial y del segundo escenario.

- Parte del guionizado del escenario basado en el accidente real de Air Nuigini 73.
- Creación del evaluator
- Creación de la escena para crear escenarios
- Multiplicador de la velocidad de reproducción de los escenarios
- Posibilidad de elegir las variantes de los pilotos que aparecerán en cada escenario
- Lógica del tutorial
- Lógica de la pantalla de resultados, incluyendo el recuento de los resultados
- Interfaz de la pantalla de resultados
- Interfaz básica de la pantalla de juego
- Implementación de la lluvia
- Posibilidad de elegir desde el editor qué escenarios mostrar en la escena de selección de niveles
- Evaluación de los pilotos, desde la interacción con el piloto hasta el feedback de la elección
- Cinemática del menú principal
- Pantalla de créditos
- Optimización del menú principal para permitir reproducción en dispositivos móviles

A.2. Javier Cruz López de Ochoa

- Producción. Tareas relacionadas con la producción de este proyecto.
  - Todas las reuniones con los directores del proyecto.
  - Redacción en las actas de las reuniones.
  - Preparación de las presentaciones con el grupo de investigación cada hito, a parte de la exposición final.
  - Planificación de los objetivos a conseguir cada sprint.
  - Creación del cuestionario para las pruebas de usuarios.
  - Traducción del cuestionario a inglés.
- Memoria. Tareas relacionadas con la redacción de esta memoria
  - Búsqueda de trabajos y juegos serios relacionados.
  - Parte de la inserción de imágenes y cuotas.
  - Parte del apartado de agradecimientos.
  - Capítulo 1, hecho sobretodo el apartado de Estructura del trabajo.
  - Capítulo 2, Trabajo relacionado
  - Capítulo 3 en su gran mayoría, excepto el apartado de Interfaz, que lo hicimos juntos.
  - Capítulo 4 completo, exceptuando correcciones finales realizadas por mi compañero.
  - Capítulo 5, hechos las secciones de clases, excepto lo referente a la subsección del *Evaluator*. También he hecho la sección de Escenas.
  - Capítulo 6 completo, exceptuando correcciones finales realizadas por mi compañero.
  - Capítulo 7, la parte de análisis de los objetivos propuestos y su evaluación.
  - Inserción de las imágenes de la memoria.
- Juego. Tareas relacionadas con la programación del simulador ludificado.
  - Creación de bocetos de la Interfaz de Usuario.
  - Diseño de la lógica del prototipo del programa de consola.
  - Búsqueda de posibles escenarios en los que basarse para la realización del escenario final.
  - Serialización de las clases en el prototipo de programa de consola.
  - Creación de las tablas de *OB* en el prototipo de programa de consola.

- Búsqueda de animaciones en Mixamo.
- Búsqueda de sonidos de alerta.
- Sistema de reproducción de sonidos de alerta.
- Parte del guionizado del tutorial y del segundo escenario.
- Participación en el guionizado del escenario basado en el accidente real de Air Nuigini 73.
- Sistema de reproducción de un escenario.
- Carga asíncrona de las escenas.
- Creación del logotipo del juego usando Photoshop + Paint Tool Sai.
- Creación de la escena del juego encargada de la carga inicial.
- Creación de la escena del juego de Opciones, junto a la opción de volumen de sonido.
- Creación de la escena de selección de escenarios.
- En la escena de resultados, creación del nivel junto a la animación de entrada.
- Búsqueda de una cabina e incorporación de esta en la escena de juego.
- En la escena de juego, creación del movimiento de la cabina.
- En la escena de juego, creación de un shader que permite los cambios de tiempo atmosférico.
- Lógica de cambio de tiempo atmosférico
- Creación a mano en Blender de los videos de pulsación de un interruptor de la cabina.
- Creación de la representación en pantalla de la etapa de vuelo.
- Creación de la mecánica de abortar una misión junto al botón para hacerlo.
- Cambio del tutorial para que funcione a base de pulsar un botón para avanzar.
- Optimización del menú principal para permitir reproducción en dispositivos móviles

Appendix B

Introduction

Despite all the progress made in the field of aviation safety, the human factor is still one of the contributing causes of most air accidents.

For this reason, in recent years, both the ICAO [1] (International Civil Aviation Organization) (Organization, 2013) and IATA[2] (International Air Transport Association) (Association, 2013) have been promoting a new training paradigm called Evidence-Based Training (EBT). This paradigm attempts to adapt training to the real needs of pilots, integrating both technical and non-technical competencies.

Flight simulators are one of the most notable historical advances in both aeronautics and computing, and are nowadays an indispensable tool for pilot training. However, how these simulators are used is just as important, and in this sense, the EBT paradigm allows the creation of scenarios in the simulator specially designed to improve specific skills that the pilot needs to train.

The design of these scenarios is not trivial and depends largely on the skill and experience of the instructor, who does not have the time to create customised scenarios for each pilot. The success of the training also depends to a large extent on the instructor's ability to assess the pilot's behaviour to see if the pilot possesses this or that competence. That is why good training for flight instructors is ultimately also important for aviation safety.

B.1. Motivation

One of the ongoing challenges in the aerospace field is the training of personnel. In the case of commercial aviation, this training is carried out by the airlines themselves, and the model currently being used to train pilots is the EBT model.

This new training model is concerned not only with the pilot's technical

[1] https://www.icao.int/
[2] https://www.iata.org/

skills (i.e. the correct execution of manoeuvres, knowledge of the aircraft controls, etc.), but also with his or her non-technical skills, which are more related to the so-called Human Factors. Flight instructors who are now called upon to act as pilot assessors using the EBT model must look at certain patterns of pilot behaviour called Observable Behaviours (or **OB**) which reveal whether or not the pilot to be assessed has the corresponding competencies, such as Communication or Leadership.

The problem is that it is difficult to follow the rules of the assessment process in this new paradigm and often more subjective assessments are made in which the instructor does not simply observe and record the pilot's Observable Behaviours. Instructors are therefore faced with this new training paradigm that is different from that traditionally used in aviation, and there is also a lack of consensus on many details of the model. For example, according to specialists, the instructors of a company like Iberia consider 9 different competencies and rate them in a range from 0 to 4, while those of another company may only consider 8 competencies and rate them with a number from 0 to 5.

It is in this situation that the question "Who trains the trainers" arises, the central theme of an article by Alfredo Hernández, Rubén Dapica and Federico Peinado (Hernández et al., 2020) which serves as a trigger for our work. Companies that choose to become certified for EBT training must in fact train their instructors in this paradigm through specific courses and must also examine them every 3 years to verify that they are still qualified to carry out their functions. (Association, 2013)

As this is a fairly new paradigm, and there are hardly any companies that have certified their training in this methodology, there is not much material for trainers to learn how to integrate this new approach into their daily work.

Within Video Game Development, the use of simulators has usually been a vital tool in practical training programmes, and in a very special way, as mentioned above, in the field of Aeronautics.

Not only have simulators been around since the dawn of Computer Science, but many are currently used in education, in training for all kinds of skills, in psychological therapy, and in a host of other scenarios.

For example, in the motorsport sector, to replicate the behaviour of the car on a circuit, there are simulators that allow drivers to familiarise themselves with its controls, even at the hardware level, without facing the risks and costs of real driving. This also allows engineers to test certain technological advances on the vehicle simulation. [3]

The high cost and safety risks involved in flying an aircraft soon justified the creation of flight simulators where practice and experiments could be carried out in a safe environment and with a more moderate investment. Today there are several simulators, with different approaches, and even sim-

[3] https://www.mercedesamgf1.com/en/news/2020/04/behind-scenes-f1-simulator/

ulation video games are marketed for the mass market, with an impressive realism and number of users. The paradigm of this can be seen in the latest instalments of the well-known Microsoft Flight Simulator saga (Xbox Game Studios, 2020).

It is important to make it clear that in this work we have been interested in the use of simulators for the training of pilot instructors, not necessarily realistic but focused on non-technical competences and following the EBT paradigm. These competencies complement the actual control of the aircraft, and are particularly challenging to define and evaluate, as they involve human factors.

In this section we attempt to present the purpose and scope of the project on which this work is based.

B.1.1. Purpose

The purpose of this Final Degree Project (FDP) is to design and develop a gamified simulator that helps in the training of flight instructors according to the EBT paradigm. This simulator is not intended to replace the current flight instructor training, but to complement it and emphasise the most relevant aspects.

This work obviously also aims to consolidate the knowledge obtained in the Degree in Video Game Development and use it to develop a useful tool oriented to a specific sector with important and real problems such as Aeronautics.

In the Section B.2, we explain the subjects of the Degree in Videogame Development that we have considered crucial when carrying out this development.

B.1.2. Scope

The proposed simulator is intended to familiarise flight instructors with the Observable Pilot Behaviours, organised according to the competences of the EBT model, and to practice their identification in an immersive and interactive environment. The gamified simulator will not be aimed at the general public, but rather at airlines, so that they can implement it as a complement to their instructor training.

To validate that the simulator has this utility, we will create a scenario based on a real accident, in collaboration with aviation experts, where the most important actions that led to the accident will be represented. And we will use this scenario as a test to teach our application to real flight instructors.

In no case has it been intended to build a 'flight simulator' as the physical aspects of the aircraft and its environment are outside the scope of this project. Nor will it be a self-contained simulator that can be used throughout

the instructor training process. Our project focuses exclusively on the detection of Observational Behaviours of the pilots based on a scripted scenario, which the simulator follows to the letter, giving the user the opportunity to react at every relevant moment.

B.2. Related subjects

This section describes the relationship of this dissertation with the contents taught throughout the Bachelor's Degree in Video Game Development.

The most relevant knowledge that has been applied in this work comes from the following subjects:

- **Programming Fundamentals**. This subject teaches the basics of programming, which we use throughout the project. In addition, C# is used, which is Unity's scripting language.
- **Video Game Engines**. This subject teaches the basics of understanding how a video game engine works (specifically, Unity), learning which are the most relevant components of the engine and how to use them.
- **Agile Production Methodologies**. In this subject we learn how to manage a project with an agile methodology such as Scrum, which has helped us to make the development of the project quite flexible, being able to adapt to changes in the specification that have arisen in our interaction with the experts.
- **Video Game Programming Technologies**. In this subject we study design patterns that have been applied to the project, such as Singleton, as well as reinforcing important concepts of Object Oriented Programming such as polymorphism.
- **Data Structures and Algorithms**. In this course we learn about orders of complexity and the structures that allow us to control this complexity in order to produce efficient code.
- **Design of Interactive Systems**. In this subject, knowledge is imparted on the design of user-centred interfaces and the use of certain patterns useful for this task.
- **Video Games for Mobile Devices**. In this subject you will discover the ins and outs of cross-platform development and you will deepen your knowledge of Unity with a focus on development for mobile devices.

B.3. Structure of the work

The structure of this work is composed of the following chapters:

- In the Appendix B we present the purpose and scope of this work. We present one of the problems that currently exist in Aeronautics and we relate this work with subjects of the degree.
- In the Chapter 2 we review the state of the art in the field of soft skills training and we talk about serious games as a support. It ends with a short survey of related work.
- In the Chapter chapter 3 the main objectives of the work are presented and a specification of all the necessary requirements to be able to develop the proposed tool is made.
- In the Chapter 4, the methodology used to develop the project and the specific tools used in the process are explained.
- In the Chapter 5, a detailed explanation of the analysis, design and implementation of our project, and also of the scenario taken as an example, is given.
- In the Chapter 6, the tests are explained and the results obtained from the tests with real users that we have carried out to validate the functioning and evaluate the usefulness of this tool are discussed, with a generally positive result.
- In the Appendix C we draw some conclusions of the project, exploring whether the objectives have been met and proposing lines of future work.

Appendix C

Conclusions

In this final chapter we offer the conclusions of the work carried out, assessing whether or not the objectives proposed in chapter 3 have been met. We also provide a line of work for the future development of this project. The objectives set out previously in chapter 3 were:

- To computationally model flight simulation scenarios based on Evidence Based Training based flight with enough realism and depth to be accepted by aviation specialists.
- To design and develop a flight instructor simulator in which to experience, on the above-mentioned scenarios, the essentials of their assessment work. This consists of recognising observational behaviours of each pilot that reveal whether or not he/she has acquired the non-technical competencies necessary to carry out the mission posed by the scenario in a safe and effective manner.
- To produce a final application compatible with mobile devices that allows validation of its functionality and usability with real users who have experience in airline pilot training.

C.1. Targets achieved

The following is an analysis of what has been achieved with respect to each objective:

- **To computationally model flight simulation scenarios based on Evidence Based Training based flight with enough realism and depth to be accepted by aviation specialists.**

  This target was achieved as the results of the expert tests show, because thanks to the help of the official (Commission, 2018) crash report, as well as the help of the experts, we were able to build a 6-minute scenario

in which the dialogues of the pilots are very close to the real ones, and the evaluations that should be given were made with the help of several experts.

It's true that more elements could be added to the scenario to better represent the environment, but this is discussed further in section section C.2.

- **To design and develop a flight instructor simulator in which to experience, on the above-mentioned scenarios, the essentials of their assessment work. This consists of recognising observational behaviours of each pilot that reveal whether or not he/she has acquired the non-technical competencies necessary to carry out the mission posed by the scenario in a safe and effective manner.**

  Although we have managed to replicate the essentials of an instructor's work (letting the player choose when, whom and which behaviours to evaluate), there is still room for improvement, as the feedback we received from the experts says that in some cases, the correct evaluation is debatable, and it may be the case that several or all options could be judged as correct.

- **To produce a final application compatible with mobile devices that allows validation of its functionality and usability with real users who have experience in airline pilot training.**

  The gamified simulator works perfectly on both Android and iOS devices, due to the facilities offered by *Unity* for exporting cross-platform projects. To compile and test the simulator on Android, all that is required is the compilation tools and a device with an Android operating system. However, compiling and testing on iOS requires an extra step, which can only be done on a computer running MacOS (Apple's operating system for its Mac computers). To perform this step, we needed the help of Didactoons [1], a company that allowed us to use their facilities for two days to install the simulator on an iPad from one of their Macs and test it.

  Unfortunately, the day before user testing, we discovered that the permissions to use the application on the iPad had expired, and so we were unable to test using an iOS device, so we used a Surface instead.

C.2. Future Work

Although we believe we have developed a solid application, there is still room for possible future work, as at the beginning of the development a

[1] https://www.didactoons.com/

rather interesting feature was proposed but not fully developed.

This would be that pilots would have a skill level for each competence and that scenarios could be given a level of difficulty for each. This would allow some pilots to show a negative behaviour in that OB and other pilots, on the contrary, would have a positive behaviour. This would imply designing scenarios in which each OB would have two branches and the architecture of the project would also have to be modified.

Also, in order to provide more information to the player, it could be considered to implement a small briefing of the scenario with information about the experience level of the pilots before playing the game and also that in the game scene some graphics appear indicating the current state of the plane. This would help to interpret the situation and its context to better evaluate the pilots.

C.3. Final thoughts

This project has been of particular value to us, as the effort we have made, both in developing the simulator and in learning more about the world of aviation training, has resulted in a simulator prototype that, according to the experts we have met, has a lot of potential.

Throughout the development of this final degree project we have acquired a lot of knowledge, beyond computer science, and met many delightful people who have offered to help us and share their knowledge and time.

Apéndice D

Cuestionario utilizado en las pruebas

A continuación se presenta el cuestionario creado para la realización de las pruebas.

Cuestionario de evaluación de Flight Instructor Simulator

¡Hola!

Somos Javier Cruz y Antonio Jesús Guerra junto con nuestros tutores Federico Peinado y Rubén Dapica, de Narratech Laboratories, una iniciativa de la Facultad de Informática de la Universidad Complutense de Madrid (España).

Usaremos los datos generados por este cuestionario para analizar la utilidad de nuestro juego serio: Flight Instructor Simulator. Es juego está concebido con la finalidad de mejorar la capacidad de evaluación de los instructores, y no las habilidades de los pilotos en sí. En el juego podrás seleccionar el nivel al que quieras jugar y tendrás que evaluar a los dos pilotos, que se encontrarán realizando un ejercicio de EBT.

Link al cuestionario: https://forms.gle/tEuyDrdDb9LnecA9A

Con este gesto nos ayudas a mejorar nuestro juego serio y a probar su potencial como herramienta de aprendizaje adicional para los instructores. Todas las preguntas son opcionales y puedes responder tanto en español como en inglés.

Si tienes alguna duda o quieres saber más acerca de este proyecto, no dudes en contactar con nosotros por e-mail:
javcruz@ucm.es, antgue01@ucm.es, fpeinado@ucm.es

¡Gracias!

*Obligatorio

LLevando la prueba a cabo

Antes que nada, juega al tutorial para aprender las mecánicas del juego. Una vez concluido, selecciona "EscenarioTest" y juégalo tratando de detectar la mayor cantidad de "Observational Behaviours" que puedas.

En total, la prueba no debería durar más de 4 minutos.

Salta a la pregunta 1

Opinión sobre el ámbito tratado

Las siguientes preguntas (o afirmaciones en las que manifestar grado de acuerdo) sirven para retratar como ve el evaluador la cuestión educativa en lo que respecta al pilotaje de aeronaves.

1. 1. La formación actual en pilotaje de aeronaves es suficiente.

   Marca solo un óvalo.

   - Totalmente en desacuerdo
   - En desacuerdo
   - Ni de acuerdo ni en desacuerdo
   - De acuerdo
   - Totalmente de acuerdo

2. 2. ¿Le parece útil el paradigma EBT para evaluar a los pilotos? *

   Marca solo un óvalo.

   - Totalmente en desacuerdo
   - En desacuerdo
   - Ni de acuerdo ni en desacuerdo
   - De acuerdo
   - Totalmente de acuerdo

3. 3. La formación actual de los instructores de vuelo es suficiente para desempeñar correctamente su trabajo. *

   Marca solo un óvalo.

   - Totalmente en desacuerdo
   - En desacuerdo
   - Ni de acuerdo ni en desacuerdo
   - Totalmente de acuerdo

4. 4. ¿Tiene alguna propuesta para mejorar la formación de los instructores de vuelo?

_____

_____

_____

_____

_____

Salta a la pregunta 5

Opinión sobre el juego serio

Se le preguntara al encuestado sobre lo jugado, de forma que sus respuestas puedan ayudar a entender la utilidad que podría tener este tipo de simuladores en el futuro o como podrían modificarse para que sean más útiles.

5. 1. ¿Se entiende el concepto del juego serio que presentamos?

Marca solo un óvalo.

- ◯ Sí
- ◯ No
- ◯ Otro: _____

6. 2. ¿Entiendes que abortar la misión es opcional? *

Marca solo un óvalo.

- ◯ Sí
- ◯ No
- ◯ Otro: _____

7. 3. El tutorial muestra correctamente las mecánicas del juego. *

Marca solo un óvalo.

- Totalmente en desacuerdo
- En desacuerdo
- Ni de acuerdo ni en desacuerdo
- De acuerdo
- Totalmente de acuerdo

8. 4. La información mostrada en los eventos contenidos en los escenarios es clara. *

Marca solo un óvalo.

- Totalmente en desacuerdo
- En desacuerdo
- Ni de acuerdo ni en desacuerdo
- De acuerdo
- Totalmente de acuerdo

9. 5. La información mostrada en los eventos contenidos en los escenarios es suficiente para evaluar correctamente a los pilotos. *

Marca solo un óvalo.

- Totalmente en desacuerdo
- En desacuerdo
- Ni de acuerdo ni en desacuerdo
- De acuerdo
- Totalmente de acuerdo

10. 6. ¿Qué información plantearía añadir a los escenarios?

11. 7. El desplegable de evaluación es lo suficientemente claro. *

Marca solo un óvalo.

- Totalmente en desacuerdo
- En desacuerdo
- Ni de acuerdo ni en desacuerdo
- De acuerdo
- Totalmente de acuerdo

12. 8. El desplazamiento entre las diferentes escenas y menús parece intuitivo.

Marca solo un óvalo.

- Totalmente en desacuerdo
- En desacuerdo
- Ni de acuerdo ni en desacuerdo
- De acuerdo
- Totalmente de acuerdo

13. 9. ¿Qué errores graves en la aplicación observa, que convendría arreglar de forma prioritaria?

14. 10. ¿Qué cosas cree que se podrían mejorar en la aplicación?

Salta a la pregunta 15

¡Gracias por completar el formulario!

Para terminar, hemos añadido un par de apartados en los cuales puede dejarnos su correo electrónico si quiere conocer los resultados de nuestro proyecto, además de un apartado libre donde poder dejar cualquier comentario que desee para ayudarnos a mejorar nuestra aplicación.

Al finalizar, no olvide pulsar en el botón de abajo para enviar sus respuestas.

15. 1. Dirección de correo electrónico (opcional, para recibir los resultados del proyecto)

Consulte más información sobre nuestros proyectos en http://narratech.com/

16. 2. En este apartado puede hacer cualquier comentario, sugerencia o aclaración que quiera. Su opinión nos ayudará a seguir mejorando.

Este contenido no ha sido creado ni aprobado por Google.

Google Formularios

Apéndice E

Resultados del cuestionario

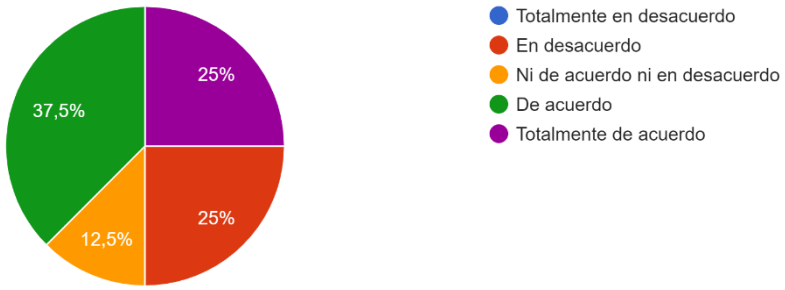
A continuación se presentan los resultados del cuestionario creado para la realización de las pruebas.

Resultados del cuestionario

Opinión sobre el ámbito tratado

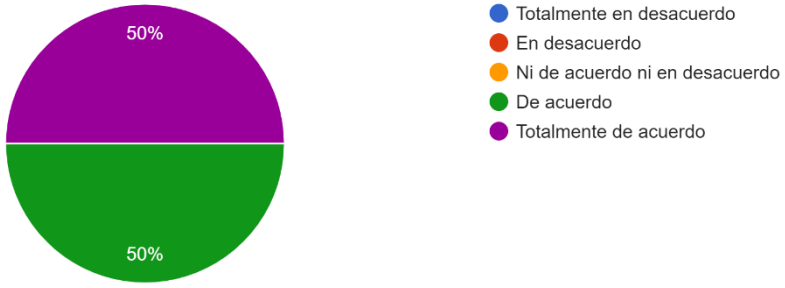
1. La formación actual en pilotaje de aeronaves es suficiente.

8 respuestas



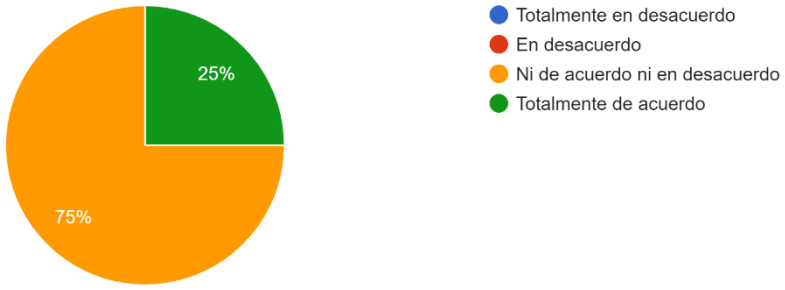
2. ¿Le parece útil el paradigma EBT para evaluar a los pilotos?

8 respuestas



3. La formación actual de los instructores de vuelo es suficiente para desempeñar correctamente su trabajo.

8 respuestas



4. ¿Tiene alguna propuesta para mejorar la formación de los instructores de vuelo? 7 respuestas

- Este juego es fundamental para ello.

- La figura del train the trainers.

- Soy partidario de cursos de formación prácticos basados en escenarios reales.

- Evaluación constante por evidencias y revisión de los procedimientos.

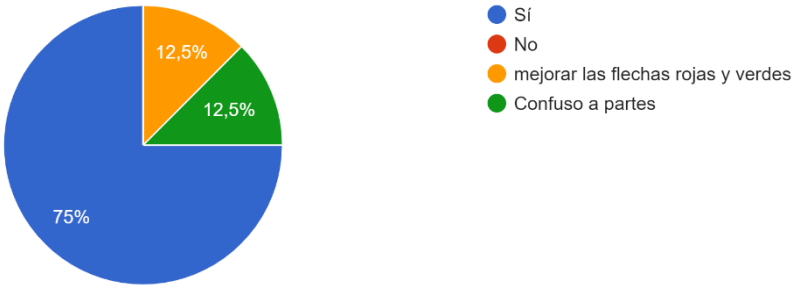
- Realizar más reuniones anuales.

- Que reciban instrucción continuada en nuevos métodos y procesos de enseñanza.

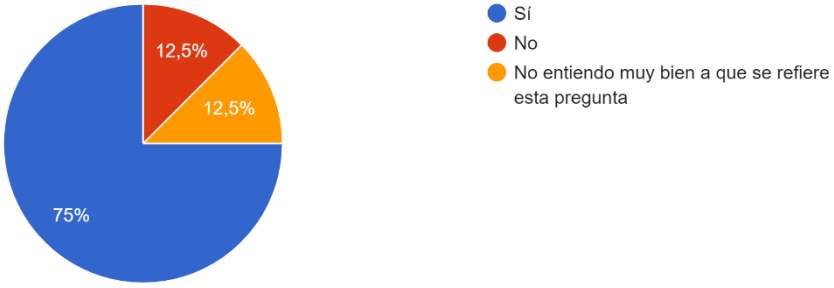
- Formación continua y mantenerse al día de sucesos, incidentes y de lo que ocurre en la industria en general.

Opinión sobre el juego serio

1. ¿Se entiende el concepto del juego serio que presentamos?
8 respuestas

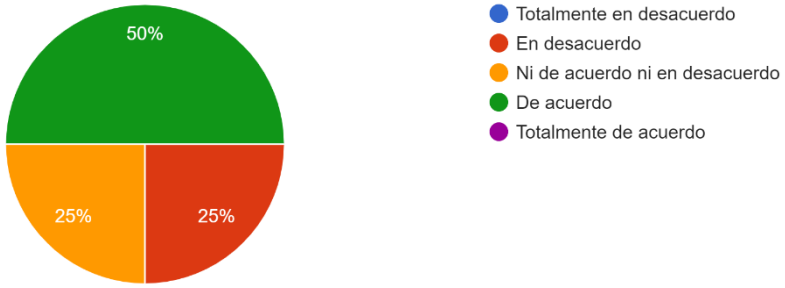


2. ¿Entiendes que abortar la misión es opcional?
8 respuestas



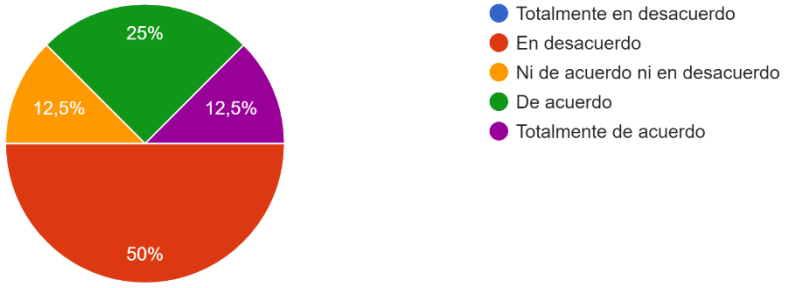
3. El tutorial muestra correctamente las mecánicas del juego.

8 respuestas



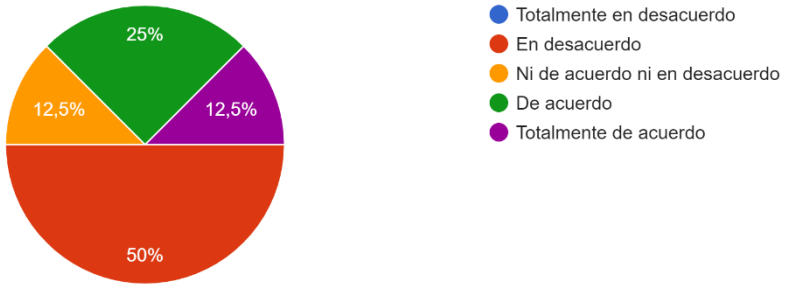
4. La información mostrada en los eventos contenidos en los escenarios es clara.

8 respuestas



5. La información mostrada en los eventos contenidos en los escenarios es suficiente para evaluar correctamente a los pilotos.

8 respuestas



6. ¿Qué información plantearía añadir a los escenarios? 7 respuestas

- No es necesaria más información.

- Disponer de una pequeña introducción sobre la escena previa al ejercicio y sin límite de tiempo.

- Más lentitud para asumir los textos. Son fases del vuelo críticas que requieren análisis claro y conciso. Las prisas no son buenas.

-No he estado de acuerdo con algunos de los procedimientos presentados.

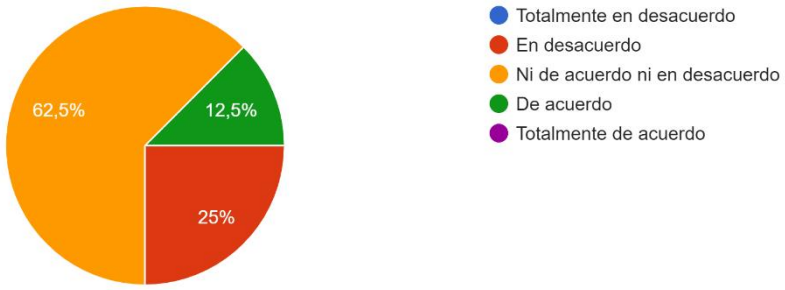
- Habría que completarla con información de la experiencia de cada piloto.

- Una estructura más organizada del evento desde el punto de vista del piloto.

- Información sobre instrumentos del avión y configuración para entender mejor el escenario.

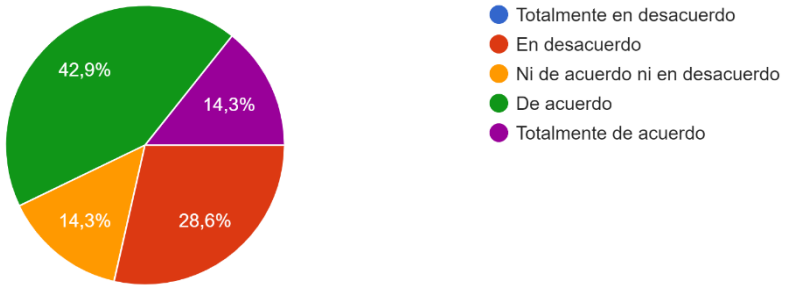
7. El desplegable de evaluación es lo suficientemente claro.

8 respuestas



8. El desplazamiento entre las diferentes escenas y menús parece intuitivo.

7 respuestas



9. ¿Qué errores graves en la aplicación observa, que convendría arreglar de forma prioritaria?

7 respuestas

- Los tiempos.

- El tiempo de dialogo es demasiado rápido. En ocasiones no permite interpretar la información, pues inmediatamente salta al otro piloto.

- Mucha rapidez en los textos e imágenes, poco tiempo para evaluar.

- La velocidad del juego.

- La sección de correcto o incorrecto de las flechas.

- La velocidad en que salen los comentarios, no da tiempo a leer. Se lo podría poner en audio. La estructura del evento, que junto con las imágenes lleva a confusión.

- Habría que dejar más claro en qué casos se espera de nosotros que hagamos una evaluación del comportamiento.

10. ¿Qué cosas cree que se podrían mejorar en la aplicación? 7 respuestas

- Opciones abiertas.

- Más que ampliar el tiempo entre diapositivas y/o mensajes, que esa opción fuera dirigida a través de un click por parte del evaluador.

- Más lentitud entre diapositivas.

- Me parece que las opciones de respuestas son bastante cerradas.

- La velocidad de los mensajes es muy rápida.

- Darle una visión de un piloto para que se pueda evaluar la secuencia de eventos.

- La representación de lo que está ocurriendo en la cabina.

Bibliografía

A, V. R. . L. C. . A. R. E. J. . T. M. . J. Simuladores de vuelo: una revisión. *Ciencia Y Poder Aéreo, 13(2), 138–149*, 2018.

Association, I. A. T. *Evidence-Based Training Implementation Guide*. 2013. ISBN 978-92-9252-191-2.

Barbieri, A., Tesei, A., Bruzzone, A., Ion, R. y Beligan, D. Learning impact evaluation of the serious game "cultural awareness – afghanistan pre-deployment". *International Journal of Serious Games*, vol. 1, 2014.

Commission, A. I. Final report. vol. 1, página 116, 2018.

De Gloria, A., Bellotti, F. y Berta, R. Serious games for education and training. *International Journal of Serious Games*, vol. 1, 2014a.

De Gloria, A., Bellotti, F. y Berta, R. Serious games for education and training. *International Journal of Serious Games*, vol. 1, 2014b.

Gamelearn. Pacific. 2015.

Hernandez, M. Historia de los simuladores de vuelo. 2012.

Hernández, A., Dapica, R. y Peinado, F. Who trains the trainers? towards a flight instructors simulator based on training scenarios. 2020.

Kahoot DragonBox AS. Dragonbox: Elements - Geometry Proofs. 2014.

Kharoufah, H., Murray, J., Baxter, G. y Wild, G. A review of human factors causations in commercial air transport accidents and incidents: From to 2000–2016. *Progress in Aerospace Sciences*, vol. 99, páginas 1–13, 2018. ISSN 0376-0421.

Organization, M. I. C. A. *Manual of evidence-based training*. 2013. ISBN 978-92-9249-242-7.

The Irregular Corporation. Pc Building Simulator. 2018.

Torrente, J., del Blanco, A., Marchiori, E. J., Moreno-Ger, P. y Fernández-Manjón, B. lt;e-adventure gt;: Introducing educational games in the learning process. En *IEEE EDUCON 2010 Conference*, páginas 1121–1126. 2010.

Ubisoft. Rocksmith 2014. 2013.

United States Army. America's Army. 2002.

University of Washington. Foldit. 2008.

VSTEP. RescueSim. 2008.

Xbox Game Studios. Microsoft Flight Simulator. 2020.