
**TEORIKART: EL VIDEOJUEGO COMO APOYO EN LA
PREPARACIÓN DEL EXAMEN TEÓRICO COMÚN DEL
PERMISO B**

**TEORIKART: THE VIDEO GAME AS A SUPPORT TOOL
TO PREPARE FOR THE TYPE B DRIVING LICENCE
THEORY TEST**



TRABAJO FIN DE GRADO

CURSO 2024-2025

AUTORES

**ADRIÁN MONTERO CASTRILLO
SIMONA ANTONOVA MIHAYLOVA**

DIRECTOR

ANTONIO CALVO MORATA

GRADO EN DESARROLLO DE VIDEOJUEGOS

FACULTAD DE INFORMÁTICA

UNIVERSIDAD COMPLUTENSE DE MADRID

**TEORIKART: EL VIDEOJUEGO COMO APOYO EN LA
PREPARACIÓN DEL EXAMEN TEÓRICO COMÚN DEL
PERMISO B**

**TEORIKART: THE VIDEO GAME AS A SUPPORT TOOL TO
PREPARE FOR THE TYPE B DRIVING LICENCE THEORY
TEST**

**TRABAJO DE FIN DE GRADO EN DESARROLLO DE VIDEOJUEGOS
AUTORES**

**ADRIÁN MONTERO CASTRILLO
SIMONA ANTONOVA MIHAYLOVA**

**DIRECTOR
ANTONIO CALVO MORATA**

CONVOCATORIA: JUNIO 2025

**GRADO EN DESARROLLO DE VIDEOJUEGOS
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID**

DÍA DE MES DE 2025

DEDICATORIA

A mi padre, por estar siempre a mi lado y no dejarme nunca tirar la toalla.- Adrián

A mi familia por creer en mí y apoyarme siempre. A Bulgaria, por ser el origen de mi historia, y a España, por ser el escenario donde sigo construyéndola.- Simona

AGRADECIMIENTOS

A nuestro tutor Toni por su labor tanto en las asignaturas que nos ha impartido como en la consecución de este trabajo y a todos los profesores que nos han transmitido sus conocimientos durante estos años. Agradecimientos a la Autoescuela Marca por permitirnos probar la eficacia de nuestro trabajo con su alumnado.

RESUMEN

TeoriKart: El videojuego como apoyo en la preparación del examen teórico común del permiso B

La educación vial es un componente esencial para garantizar la seguridad en las carreteras y formar a los futuros conductores. Los métodos tradicionales para preparar el examen teórico del permiso de conducir tipo B en España pueden resultar monótonos y poco motivadores. En este contexto los juegos serios son una buena opción para crear una experiencia de aprendizaje más dinámica y atractiva.

Este Trabajo de Fin de Grado consiste en un videojuego serio, con el objetivo de crear una herramienta complementaria para ayudar a los estudiantes a prepararse para el examen teórico, de una manera divertida. También se realizan pruebas con usuarios y se analizan los datos recogidos para evaluar en qué medida el juego contribuye a mejorar el proceso de aprendizaje de los estudiantes. Se centra en tres temas específicos del manual teórico, cada uno de ellos compuesto por diferentes niveles diseñados para ayudar a comprender las peculiaridades en estas áreas.

Palabras clave

Conducir, Examen, Videojuego, Juego Serio, Aprender, Autoescuela (máx. 10 palabras separadas por comas)

ABSTRACT

TeoriKart: The video game as a support tool to prepare for the type B driving licence theory test

Road safety education is an essential component for ensuring safety on the roads and for training future drivers. Traditional methods used to prepare for the type B driving licence theory test can be monotonous and unmotivating. In this context serious games are a good option to create a more dynamic and engaging learning experience.

This Final Degree Project focuses on the development of a serious videogame, with the aim of providing a complementary tool to help students in Spain prepare for the test in a more engaging way. User testing is also conducted, and the collected data is analyzed to evaluate the extent to which the game contributes in enhancing students' learning outcomes. It features three specific topics from the official theory manual, each having various levels designed to help users understand these areas more effectively.

Keywords

Driving, Test, Videogame, Serious Game, Learning, Driving school(10 keywords max., separated by commas).

ÍNDICE DE CONTENIDO

Capítulo 1. Introducción	1
1.1 Motivación.....	1
1.2 Objetivos.....	1
1.3 Plan de trabajo	2
Capítulo 2. Estado de la cuestión	5
2.1 Examen Teórico en España	5
2.2 Juegos serios	6
2.3 Juegos serios y simuladores de conducción.....	6
2.3.1 DriveSim.....	6
2.3.2 Virtual Simulator for the Taking and Evaluation of Psychometric Tests to Obtain a Driver's License.....	8
2.3.3 The Virtual Reality Serious Game for Learning Driving Skills before Taking Practical Test	10
2.3.4 Serious gaming: user acceptance of the simulated motorbike license training system.....	12
2.3.5 Car Driving School Simulator.....	13
2.4 Resumen y carencias.....	14
Capítulo 3. Desarrollo de TeoriKart	15
3.1 Estructura del videojuego	15
3.1.1 Menú principal.....	15
3.1.2 Menú de niveles	16
3.1.3 Niveles	19
3.2 Desarrollo	22
3.2.1 Jugador.....	22
3.2.2 Mapas	25
3.2.3 Coches controlados por inteligencia artificial.....	35

3.2.4	Elementos de tráfico	40
3.2.5	Condiciones climáticas y ambientales.....	40
3.2.6	Editor de Niveles	41
3.2.7	Telemetría	50
Capítulo 4.	Pruebas de usuario	53
4.1	Prueba del videojuego.....	53
4.1.1	Cuestionario Previo	54
4.1.2	Prueba del juego.....	54
4.1.3	Cuestionario Posterior	55
4.2	Prueba del editor.....	55
Capítulo 5.	Resultados de las pruebas de usuario	57
5.1	Resultados de la prueba del videojuego	57
5.1.1	Demografía de los usuarios.....	57
5.1.2	Análisis de las trazas de interacción con el videojuego	60
5.1.3	Comparativa entre cuestionarios	66
5.1.4	Opinión del videojuego.....	69
5.2	Resultados de la prueba del editor	70
Capítulo 6.	Conclusiones y trabajo futuro	72
Introduction.....		73
Motivation		73
Goals.....		73
Work plan.....		74
Conclusions and future work		75
Contribuciones Personales.....		76
Adrián Montero Castrillo		76
Simona Antonova Mihaylova.....		78
Bibliografía.....		80
Apéndice A - Preguntas de los cuestionarios de las pruebas con usuarios		82

Preguntas demográficas	82
Preguntas tipo test	82
Preguntas de opinión	93
Apéndice B - Guía de uso del editor de niveles	94
Interfaz general del editor	94
Tipos de nivel	95
Apéndice C - Preguntas del cuestionario de la prueba del editor usuario	98

ÍNDICE DE FIGURAS

Figura 2.1: Imagen del software de DriveSim	7
Figura 2.2: Imagen de un nivel del simulador DriveSim	7
Figura 2.3: Volante LOGITECH G29	8
Figura 2.4: Imagen de la prueba de Coordinación Mano-Ojo.....	9
Figura 2.5: Gráfica de las monedas obtenidas por intento en la prueba de Maniobrabilidad del vehículo	9
Figura 2.6: Imagen de la prueba de Precaución ante Accidentes, donde el usuario debe frenar ante la presencia de un peatón.....	10
Figura 2.7: Imagen de los tres sectores del juego	11
Figura 2.8: Imagen tomada durante las pruebas de usuario del juego	11
Figura 2.9: Imagen de un nivel del juego	12
Figura 2.10: Imagen del menú de Car Driving School Simulator	13
Figura 2.11: Imagen de un nivel de Car Driving School Simulator	14
Figura 3.1: Imagen del menú principal del videojuego	15
Figura 3.2: Imagen del menú del videojuego Overcooked 2	16
Figura 3.3: Boceto inicial del menú del videojuego	16
Figura 3.4: Imagen del menú de selección de nivel del videojuego	17
Figura 3.5: Imagen del menú de selección de nivel del videojuego, donde el jugador se acerca al primer nivel de la categoría prioridad	18
Figura 3.6: Imagen del texto explicativo del nivel 1 de la categoría prioridad..	18
Figura 3.7: Imagen del nivel 3 de la categoría prioridad	19
Figura 3.8: Imagen del nivel 1 de la categoría manejo	20

Figura 3.9: Imagen del nivel 3 de la categoría luces seleccionando las luces del vehículo.....	21
Figura 3.10: Imagen del nivel 3 de la categoría luces durante la conducción	22
Figura 3.11: Vista del prefab del coche del jugador en Unity	23
Figura 3.12: Sistema GPS en una versión previa del videojuego.....	24
Figura 3.13: Vista del minimapa en la interfaz del nivel 3 de manejo	24
Figura 3.14: Vista del del minimapa ampliado del nivel 3 de manejo	25
Figura 3.15: Imagen explicativa de las dimensiones de las piezas	26
Figura 3.16: Cabecera del archivo nivel10.json.....	27
Figura 3.17: Cabecera del archivo nivel18.json.....	27
Figura 3.18: Ejemplo de colocación de piezas en el JSON.....	28
Figura 3.19: Estructura de la carpeta Assets/Resources/PiezasPrefabs.....	29
Figura 3.20: Ejemplo de estructura del jugador en archivo	29
Figura 3.21: Ejemplo de estructura del objetivo en archivo.....	30
Figura 3.22: Ejemplos de estructura de varias señales en archivo	30
Figura 3.23: Señales y marcas viales disponibles en el videojuego	31
Figura 3.24: Creación de vehículos en el nivel 2 de prioridad	32
Figura 3.25: Imagen del nivel 1 de prioridad, donde el jugador debe decidir el orden de prioridad en un cruce	33
Figura 3.26: Ejemplo de creación de un semáforo	34
Figura 3.27: Diálogos definidos para el nivel 4 de prioridad	35
Figura 3.28: Pieza vertical en el inspector de Unity	36
Figura 3.29: Vista del script Waypoint en el inspector de Unity	37
Figura 3.30: Imágenes de los Waypoints de las piezas Glorieta e Intersección	38

Figura 3.31: Vista de la herramienta de creación de Waypoints en Unity	38
Figura 3.32: Vista del prefab de los coches de Inteligencia Artificial en Unity.	39
Figura 3.33: Vista del prefab del semáforo, destacan 2 triggers (rectángulos verdes)	40
Figura 3.34: Imagen de la escena del editor	41
Figura 3.35: Mecánica de arrastrar y soltar una pieza en el mapa.....	42
Figura 3.36: Panel ITEMS del editor.....	43
Figura 3.37: Vista del editor seleccionando la pieza (2,1) del mapa	44
Figura 3.38: Vista del editor con la pieza de la Figura 3.37 aumentada	44
Figura 3.39: Vista de pieza ampliada con elementos colocados en ella	45
Figura 3.40: Vista del componente InteractivePoint en la jerarquía de Unity ..	45
Figura 3.41: Vista de nivel de manejo en el editor	46
Figura 3.42: Panel de configuración de nivel.....	46
Figura 3.43: Archivo JSON correspondiente al nivel de la Figura 3.41	47
Figura 3.44: Panel con la lista de niveles del editor.....	48
Figura 3.45: Escena de juego del nivel de la Figura 3.42.....	48
Figura 3.46: Editor con nivel de luces, con su panel de configuración resaltado	49
Figura 3.47: Editor durante la creación de un nivel de prioridad, con el nuevo panel de configuración resaltado	49
Figura 3.48: Panel de configuración para el vehículo colocado en el mapa	50
Figura 3.49: Vista de la sección de identificadores en el entorno de Simva ...	51
Figura 3.50: Vista de los assets de Simva y Xasu en la ventana Package Manager de Unity	52
Figura 3.51: Vista del objeto XasuTracker en el inspector de Unity.....	52

Figura 3.52: Ejemplo de envío de traza de nivel completado correctamente

53

Figura 4.1: Fotografías tomadas durante las pruebas de usuario54

Figura 4.2: Fotografías tomadas durante las pruebas del editor55

Figura 5.1: Distribución de edad de los usuarios57

Figura 5.2: Frecuencia de tiempo dedicado a videojuegos de los usuarios ...58

Figura 5.3: Distribución de tiempo de matriculación en la autoescuela de los usuarios59

Figura 5.4: Distribución de métodos de preparación para el examen de los usuarios60

Figura 5.5: Distribución de aciertos entre cuestionario previo y posterior68

ÍNDICE DE TABLAS

Tabla 5.1: Estadísticas de los niveles del videojuego de todos los usuarios.....	61
Tabla 5.2: Estadísticas de los niveles del videojuego de los usuarios que más juegan a videojuegos.....	63
Tabla 5.3: Estadísticas de los niveles del videojuego de los usuarios que nunca juegan a videojuegos.....	64
Tabla 5.4: Comparativa entre media de tiempo entre pulsaciones entre los usuarios que más juegan a videojuegos y los que no juegan nunca.....	65
Tabla 5.5: Comparativa entre medias de las luces encendidas por usuario en los niveles de luces	66
Tabla 5.6: Comparativa entre aciertos en los cuestionarios de todos los usuarios..	67
Tabla 5.7: Media de las valoraciones del videojuego.....	69

1.1 Motivación

En los últimos años, es cada vez más frecuente encontrar alumnos en las autoescuelas de España que se limitan a prepararse para la obtención del permiso B realizando únicamente tests de examen, sin leer el manual ni asistir a clases impartidas por las propias autoescuelas. Esto es preocupante, ya que se fomenta un aprendizaje superficial basado en memorizar las respuestas, limitando la comprensión de las normas de tráfico, algo que puede causar problemas futuros, especialmente a la hora de examinarse del examen práctico.

Esta tendencia se aprecia especialmente en el alumnado más joven, debido a una clara falta de motivación, ya que los métodos tradicionales son repetitivos y poco dinámicos, además de no aprovechar bien las tecnologías modernas a las que están acostumbrados los jóvenes. Muchas personas aprenden más fácilmente de manera visual, o les cuesta visualizar algunas situaciones típicas de tráfico al no ser conductores.

Lo que motiva este trabajo es buscar nuevas maneras de estudiar el examen teórico, de una forma más amena y divertida. Los videojuegos se han visto como un medio para practicar las normas de circulación y la conducción en sí, pero enfocados en ejercicios libres y el carné práctico. Sin embargo, el potencial de los videojuegos no está limitado a estas áreas, podrían utilizarse para abordar la preparación del examen teórico.

1.2 Objetivos

El objetivo de este trabajo es desarrollar un videojuego que sirva como herramienta complementaria para la preparación del examen teórico del permiso de conducir tipo B en España. Para conseguir este objetivo, se abordan los siguientes subobjetivos:

- Entender el campo. Es necesario revisar el contenido curricular del examen teórico para identificar lo que se desea enseñar, al igual que estudiar cuales son las dificultades más frecuentes de los alumnos durante la preparación del examen. Además, se deben analizar los videojuegos y aplicaciones existentes relacionadas con este ámbito.

Introducción

- Desarrollar un videojuego que aborde el tema. Se propone desarrollar un videojuego que aborde el temario del examen de conducir teórico.
- Estudiar la eficacia del videojuego. Requiere evaluar la efectividad del videojuego mediante pruebas de usuario con alumnos de una autoescuela, se analizarán los resultados de dichas pruebas y se recabará retroalimentación por parte de los usuarios sobre posibles mejoras para el videojuego.

1.3 Plan de trabajo

Durante el transcurso de este trabajo, se van a realizar reuniones cada 2 semanas, tanto presenciales como en la plataforma Google Meet¹, el desarrollo del videojuego se realizará en un repositorio público de GitHub². A continuación, se describe el plan de trabajo a seguir para la consecución de los objetivos descritos en el apartado anterior:

- Investigación. Durante esta etapa, se investigarán tanto el contenido del examen teórico, como los diferentes videojuegos/herramientas relacionadas con el ámbito de los permisos de conducción. Se analizarán los aspectos clave, como el diseño educativo y las mecánicas de juego de las aplicaciones existentes.
- Diseño del videojuego. Se llevará a cabo un primer diseño del videojuego, realizando bocetos de cómo podrían ser los distintos niveles y pensando las mecánicas que permitan cumplir el objetivo educativo. También será necesario recabar información de los temas más difíciles de entender, por lo general, para el alumnado de las autoescuelas y analizar cuales se podrían adecuar mejor para ser abordados por un videojuego.
- Desarrollo del videojuego. Se desarrollará el videojuego siguiendo el diseño inicial. A medida que se avance, es posible que sea necesario adaptar dicho diseño. También se desarrollará un editor de niveles, para agilizar el proceso de creación de estos.
- Pruebas de usuario. Se realizarán pruebas con usuarios del videojuego en una autoescuela, siendo los usuarios alumnos matriculados preparándose para el

¹ <https://meet.google.com/>

² <https://github.com/>

Introducción

examen teórico. El objetivo de estas pruebas es evaluar la aceptación y la eficacia del videojuego, recogiendo las opiniones de los jugadores y midiendo los conocimientos adquiridos antes y después de jugar.

- Análisis de los datos. Tras las pruebas con usuarios, se analizarán los datos recogidos para evaluar si el juego es efectivo para preparar el examen teórico de conducir. Además, se estudiará su aceptación por parte de los usuarios y la dificultad y diseño de los niveles en base a los datos obtenidos.

2.1 Examen Teórico en España

El examen teórico en España consiste en una prueba tipo test con 30 preguntas con tres opciones, con una única correcta (Dirección General de Tráfico, 2024). Permite un máximo de 3 fallos para aprobar. Existe la posibilidad de presentarse al examen por libre, pero lo habitual es acudir a una autoescuela para una mejor formación. En el año 2024, un 52% de los exámenes teóricos fueron aptos, sin embargo, no debe interpretarse como un porcentaje de aprobados y suspensos, ya que una misma persona que suspende más de una vez sigue contabilizando en esta estadística. Pese a ello, sigue siendo una proporción muy baja que indica una necesidad de mejorar la comprensión del contenido del examen.

Debido a esto, para elegir correctamente los temas que se van a incluir en el videojuego, se debe investigar sobre qué preguntas se fallan más en los exámenes teóricos y cuáles son los temas que más cuestan comprender al alumnado.

La editorial de formación vial PONS Seguridad Vial³ realiza frecuentemente informes anuales acerca de las preguntas que más errores provocan en los test de preparación para la obtención del permiso B de conducir. La editorial analiza multitud de test online realizados en su plataforma PONS DIGITAL, con la cual trabajan más de 2300 autoescuelas en España.

Se han examinado informes de los últimos años, en los cuales los temas más frecuentemente fallados son seguridad vial, señales de tráfico, adelantamientos, limitaciones de velocidad y sanciones (COMUNICACIONPONS, 2024).

Por otra parte, y de cara a desarrollar ese trabajo, también se ha consultado con profesores de la Autoescuela Marca⁴ sobre los temas que suelen causar confusión en sus alumnos. Los profesores concluyeron en que los dispositivos de iluminación en los vehículos son un tema que siempre le cuesta mucho comprender a los alumnos, especialmente en situaciones climatológicas adversas. Además, al igual que refleja PONS Seguridad Vial en sus informes, los profesores de la autoescuela afirman que el tema de seguridad vial se les hace complicado a los alumnos, llegando a ser el que

³ <https://ponsseguridadvial.com/>

⁴ <https://autoescuelamarca.blogspot.com/>

Estado de la cuestión

más fallan en los exámenes. En añadido, indicaron que incluir en el juego señales sería un buen elemento. En los últimos años ha aumentado el desarrollo de diferentes juegos serios y simuladores enfocados en el ámbito de la conducción.

2.2 Juegos serios

Los juegos serios son juegos diseñados para un propósito principal formativo, a diferencia del objetivo de los juegos tradicionales, que se trata del entretenimiento (Laamarti, Eid and El Saddik, 2014). Tienen multitud de campos de usos, como pueden ser el sector educativo, científico, sanitario, etc. En los últimos años destacan, entre otros, varios juegos para aprender matemáticas, idiomas, programación o para prevenir el acoso escolar.

Entre sus principales ventajas destaca la posibilidad de integrar un aspecto recreativo durante la formación, el aumento de la motivación, la mejora en la retención de contenidos y la posibilidad de aprender de los errores. Además, los juegos serios tienen un coste medio de producción muy bajo y su distribución se limita a accesos mediante un sitio web dedicado (Morayma Estrada, 2024).

2.3 Juegos serios y simuladores de conducción

A continuación, se hace un análisis de varios juegos serios y simuladores de conducción.

2.3.1 DriveSim

DriveSim⁵ es un software de simulación especialmente diseñado para uso educativo en autoescuelas, centros de formación o cursos de amaxofobia y prevención de riesgos laborales. El simulador incluye situaciones realistas que permite a los usuarios practicar como si estuvieran conduciendo un vehículo real.

⁵ <https://drivesimsimulator.com/>

Estado de la cuestión



Figura 2.1: Imagen del software de DriveSim

En el simulador se puede configurar el tipo de vehículo, el clima, la hora y el idioma. Además, dispone de una configuración para controlar la evolución de los alumnos mediante informes y listados. Para utilizar DriveSim se necesita el DS-PAD, un componente del simulador que dispone de palancas de iluminación, intermitentes y limpiaparabrisas, cinturón de seguridad y freno de estacionamiento.

Obtuvo el premio Fun & Serious Games 2014 en la categoría de Mejor Juego de Estrategia Empresarial (Belén Gómez, 2014). Ese mismo año, fue finalista de los Unity Awards⁶ en dos categorías, Best VizSim Project y Community Choice.



Figura 2.2: Imagen de un nivel del simulador DriveSim

⁶ <https://unity.com/es/awards/2014>

Estado de la cuestión

2.3.2 Virtual Simulator for the Taking and Evaluation of Psychometric Tests to Obtain a Driver's License

Esta aplicación de realidad virtual está diseñada para administrar pruebas psicométricas a conductores, tanto profesionales como no profesionales (Sánchez et al., 2019). Fue creada por investigadores de dos universidades ecuatorianas, la Universidad de las Fuerzas Armadas ESPE y la Universidad Internacional del Ecuador.

Las pruebas están dirigidas a aquellos que obtienen su permiso de conducir por primera vez, están en proceso de renovarlo o de cambiar la categoría. La aplicación utiliza el motor Unity 3D para simular pruebas prácticas de conducción y evalúa las reacciones de los conductores ante diversos estímulos audiovisuales. El objetivo principal es medir las habilidades sensoriales, la capacidad motora y la destreza cognitiva y psicológica de los conductores. Se llevaron a cabo pruebas en 4 etapas para evaluar la capacidad de una persona para obtener una licencia de conducir. Para las pruebas se utilizó un kit de dispositivos hápticos, como el volante LOGITECH G29, pedales y palanca de cambios, permitiendo que sea de la manera más cercana a la realidad⁷.



Figura 2.3: Volante LOGITECH G29

Las pruebas son las siguientes:

- Coordinación Mano-Ojo. Esta prueba consiste en un circuito para evaluar la coordinación mano-ojo del conductor, utilizando la prueba de coordinación

⁷ <https://www.logitechg.com/es-es/products/driving/driving-force-racing-wheel.html>

Estado de la cuestión

bimanual tradicional. Se crea una trayectoria deseada mediante puntos que el usuario debe seguir, y luego se compara con la trayectoria real del usuario para calcular el error y la calificación al final del circuito.



Figura 2.4: Imagen de la prueba de Coordinación Mano-Ojo

- Maniobrabilidad del vehículo. Diseñada para medir la capacidad de discriminar estímulos y la concentración del conductor en la carretera. El circuito incluye varios objetos que el vehículo debe esquivar mientras recoge monedas, lo que representa la trayectoria deseada. Al final, se calcula el error y el número de monedas obtenidas.

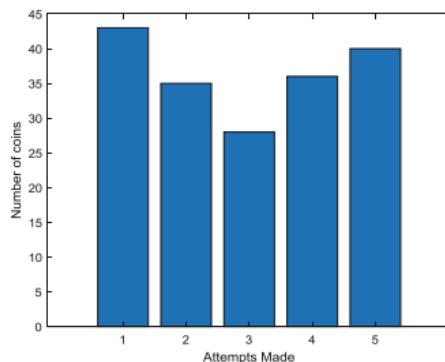


Figura 2.5: Gráfica de las monedas obtenidas por intento en la prueba de Maniobrabilidad del vehículo

- Control de Distancia. Evalúa la percepción del tiempo/espacio del usuario y su capacidad de autocontrol. El vehículo del usuario debe mantener una distancia estándar de otro vehículo en un circuito con varias curvas, evaluando la destreza y prudencia del conductor.
- Precaución ante Accidentes. Evalúa la capacidad del individuo para reaccionar ante factores externos en la carretera, como peatones o animales

Estado de la cuestión

que aparecen inesperadamente. El conductor debe decidir cómo frenar según la distancia del obstáculo.



Figura 2.6: Imagen de la prueba de Precaución ante Accidentes, donde el usuario debe frenar ante la presencia de un peatón.

2.3.3 The Virtual Reality Serious Game for Learning Driving Skills before Taking Practical Test

Este juego serio tiene como objetivo aumentar la conciencia de los conductores y reducir el riesgo de tener accidentes en carretera, mediante la enseñanza de las reglas de conducción antes de la subida a un coche real (Likitweerawong and Palee, 2018). Se centra en las reglas de conducción de Tailandia. Se ha realizado en Unity 3D y es compatible con dispositivos VR y joysticks de volante.

Se evalúan las habilidades del jugador calculando una puntuación en cada punto de control. Cada punto de control tiene diferentes tipos de evaluación de habilidades, como estacionamiento, velocidad y manejo de curvas. El curso de conducción se desarrolla basándose en el curso del examen práctico del Departamento de Transporte de Tailandia.

Las puntuaciones de cada sección se informan después de que el jugador completa todas las partes. Los resultados muestran el rendimiento del jugador en cada parte del curso de conducción. En el examen práctico, el conductor necesita pasar por tres áreas de pruebas que son: avanzar y retroceder en línea recta, estacionamiento y prueba de estacionamiento lateral. El juego contiene estos tres sectores en los que un jugador puede practicar o hacer una prueba antes de tomar un examen práctico real. El jugador también recibe retroalimentación sobre qué tan bien o mal conduce y en qué áreas necesita mejorar.

Estado de la cuestión



Figura 2.7: Imagen de los tres sectores del juego

El juego fue evaluado con 100 personas, mostrando que puede proporcionar una idea básica de la conducción y aumentar la confianza antes de conducir un coche real. Los resultados indican que el juego ayuda a mejorar las habilidades de conducción y proporciona conocimientos básicos. Además, el uso de dispositivos VR y joysticks de volante aumenta la inmersión y la emoción del juego, especialmente para usuarios jóvenes que desean aprender a conducir antes de obtener su licencia.

Los resultados también muestran que el juego es más atractivo para los hombres (75%) que para las mujeres (25%), y que tanto los jugadores con licencia de conducir (63.7%) como sin ella (36.3%) pueden beneficiarse del juego. Las dos mayores proporciones de jugadores tienen entre 19 y 24 años y más de 24 años, lo que indica que el juego es interesante y proporciona inmersión incluso para jugadores adultos.



Figura 2.8: Imagen tomada durante las pruebas de usuario del juego

Estado de la cuestión

2.3.4 Serious gaming: user acceptance of the simulated motorbike license training system

El contenido de este juego se basa principalmente en el proceso de obtención de licencias de conducción de motocicleta desarrollado por el Ministerio de Transporte de Taiwán (Li, Lu and Wei, 2023). Incluye actividades como conducir en línea recta, cruzar pasos de peatones, hacer giros en dos etapas en intersecciones, cambiar de carril, girar en ángulos rectos, estacionar y cruzar pasos a nivel. El diseño de los niveles y del juego se planifica para acercarse a situaciones de la vida real, y la jugabilidad se basa en los hábitos de uso del teclado comúnmente utilizados en los juegos de PC. Está desarrollado con el motor Unity 3D.



Figura 2.9: Imagen de un nivel del juego

El objetivo principal del juego es proporcionar a los jugadores múltiples oportunidades de práctica para evitar consecuencias graves causadas por comportamientos peligrosos. El juego utiliza mensajes de voz y texto para guiar a los jugadores, similar a operar una motocicleta en un campo de prácticas.

El primer nivel prueba la conducción equilibrada en línea recta, donde la motocicleta debe pasar por un rango específico en más de 7 segundos sin presionar las líneas. Si el jugador falla, se muestra una pantalla de misión fallida con la razón del fallo.

El experimento se llevó a cabo con la participación de 126 usuarios, principalmente estudiantes universitarios y de secundaria en el centro de Taiwán. Se recogieron datos mediante introducción al sistema, juego de los usuarios, distribución de cuestionarios y recolección de estos para análisis. Los resultados mostraron que el

Estado de la cuestión

46.8% de los participantes eran hombres y el 53.2% mujeres, con la mayoría mayores de 18 años y estudiantes universitarios. El 91.3% de los participantes consideró que la motocicleta será su principal medio de transporte en el futuro.

2.3.5 Car Driving School Simulator

Este juego está disponible para descargar en la Play Store ha sido desarrollado por BoomBit Games⁸ para la plataforma Android. Se trata de un videojuego educativo cuyo objetivo es el aprendizaje de las normas de tráfico. El juego se centra en las mecánicas de conducción, y parece tener un enfoque más para la parte práctica del permiso B.

Tiene 5 modos de juego:

- Lecciones: tutorial con 12 niveles en los que se enseñan las mecánicas básicas, y algunas normas de la conducción.
- Adventures: packs de misiones con vehículos especiales.
- Club de arcade: mini juegos de arcade y lógica para ganar recompensas adicionales.
- Multijugador: jugar con más jugadores, misiones especiales.
- Viaje libre: explorar mapas con diferentes coches.



Figura 2.10: Imagen del menú de Car Driving School Simulator

Los 4 modos de juego se desbloquean al completar el tutorial. Los mapas son de varias ciudades del mundo, sobre todo de Estados Unidos. El coche se controla mediante botones en pantalla. Se puede regular la sensibilidad, ajustar los controles según seas zurdo o diestro, las marchas, el lado de la carretera, etc.

⁸ <https://boombit.com/games/car-driving-school-simulator-en/>

Estado de la cuestión

En cada nivel aparece una misión y unas flechas azules que indican la dirección que hay que seguir.



Figura 2.11: Imagen de un nivel de Car Driving School Simulator

2.4 Resumen y carencias

Tras investigar los juegos y aplicaciones expuestos anteriormente, se puede apreciar una tendencia a realizarlos en Realidad Virtual (VR), un elemento que puede resultar atractivo para la gente más joven y además más inmersivo. En cambio, desarrollar el videojuego en VR, además de tener un mayor coste de desarrollo, supone un problema a la hora de aplicar el juego en autoescuelas con varios estudiantes a la vez, tanto en espacio, como por la falta de dispositivos y hardware específico en las propias autoescuelas.

Asimismo, los juegos y aplicaciones analizadas se enfocan más en el manejo del propio vehículo que en la comprensión de las normativas, lo que los hace una herramienta útil para la preparación del examen práctico, pero no para el examen teórico. Hay muchos datos, y una gran parte de normativa, que estos juegos no llegan a abordar, ya que no están relacionados con la conducción en sí.

Por último, al igual que los juegos analizados, se ha tomado la decisión de desarrollar el videojuego en 3D, permitiendo representar situaciones reales de una forma más fidedigna, además de proporcionar una visión del entorno más completa, y permitiendo a los jugadores visualizar lo que ocurre alrededor del avatar del jugador. Pero en este caso, las características del juego están enfocadas a poder ser ejecutadas en ordenadores con hardware antiguo y de características medio-bajas.

Para el desarrollo de TeoriKart se ha elegido el motor de videojuegos Unity ⁹ debido a que cumple todos los requisitos para el desarrollo de un videojuego serio, tiene una documentación extensa y detallada, existen muchos recursos específicos para Unity, se puede tener un repositorio del proyecto fácilmente, y se han dado numerosas asignaturas usando este motor, por lo que la manera de trabajar y la API es familiar y sencilla. Todo el desarrollo del trabajo se encuentra en el siguiente repositorio de GitHub: <https://github.com/admont02/TFG-SimonaAntonova-AdrianMontero>

3.1 Estructura del videojuego

Para poder representar el temario fielmente, se ha optado por un juego por niveles agrupados por temas. En este trabajo se han elegido 3 de los temas más difíciles para los estudiantes: prioridades, luces y señalización.

3.1.1 Menú principal

El menú principal tiene un diseño coherente con la temática del videojuego, tiene un fondo con señales de tráfico, y aparece el título y cuatro botones: Jugar, Editor, Créditos y Salir.



Figura 3.1: Imagen del menú principal del videojuego

⁹ <https://unity.com/>

Desarrollo de TeoriKart

3.1.2 Menú de niveles

Para el diseño del menú de niveles, por el que se accede a cada nivel, se ha tomado inspiración del popular videojuego Overcooked 2¹⁰ en el cual el usuario controla un autobús por un mapa sencillo, en el que se encuentran los niveles.



Figura 3.2: Imagen del menú del videojuego Overcooked 2

Inicialmente se hizo un boceto para visualizar la idea.

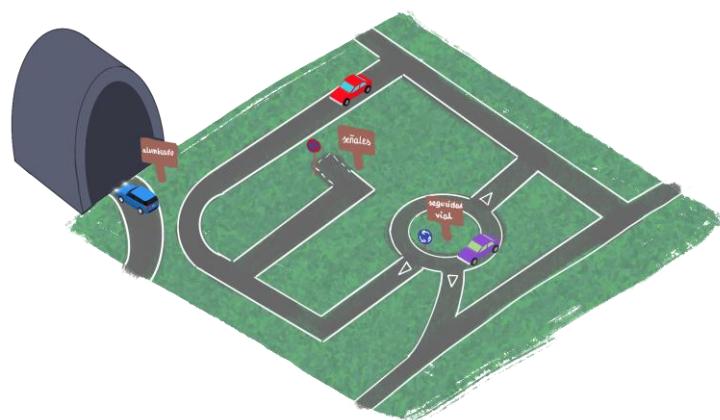


Figura 3.3: Boceto inicial del menú del videojuego

¹⁰ <https://www.team17.com/games/overcooked-2>

Desarrollo de TeoriKart

Al iniciar el menú de niveles comienza con un cuadro de texto con una explicación de los controles, tipos de niveles y objetivos.

El mapa se ha diseñado en vista ortográfica y consiste en una ciudad con 3 carreteras, las cuales corresponden a los tres temas seleccionados. Estos se identifican mediante carteles que indican el nombre de los temas. A lo largo de cada carretera, se encuentra una hilera de señales que muestran los números de cada nivel, los cuales aparecen ordenados.



Figura 3.4: Imagen del menú de selección de nivel del videojuego

Para acceder a un nivel, hay que mover el coche haciendo clic sobre la carretera, hasta acercarse a la señal correspondiente al nivel deseado. Cuando se hace clic, aparece un indicador temporal sobre la posición a la cual se dirige el coche. Una vez se está lo suficientemente cerca de la señal, esta aumenta de tamaño y aparece un indicador visual y sonoro para que el jugador sea conocedor de que puede clicar sobre ella.

Desarrollo de TeoriKart



Figura 3.5: Imagen del menú de selección de nivel del videojuego, donde el jugador se acerca al primer nivel de la categoría prioridad

Al hacer clic sobre la señal, se muestra un texto informativo sobre el nivel, y dos botones para jugar el nivel y volver al mapa.

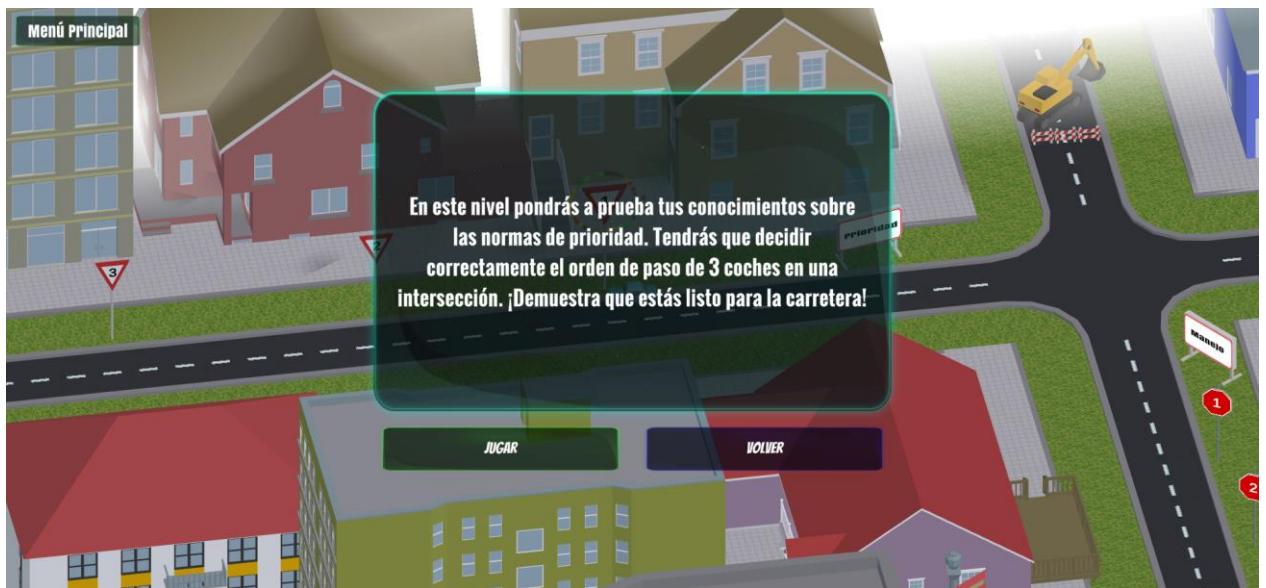


Figura 3.6: Imagen del texto explicativo del nivel 1 de la categoría prioridad

En cuanto a la interfaz de usuario, en la esquina superior izquierda se encuentra un botón para regresar al menú principal, el cual se explicará más adelante.

Desarrollo de TeoriKart

3.1.3 Niveles

Al entrar a cualquier nivel, siempre aparece un recuadro de texto con la explicación del funcionamiento del nivel. El diseño de los niveles se va a dividir en 3 categorías:

3.1.3.1 Prioridad

Estos niveles están centrados en las reglas de prioridad entre vehículos. El usuario tiene que decidir el orden de prioridad según las posiciones, el tipo de vehículo, la dirección a la que van, la señalización o si están en situación de emergencia. Hay varios tipos de vehículos: turismo, autobús y ambulancia. La escena consiste en una intersección con varios vehículos situados en distintos lados del cruce, que se ve con una vista fija de la intersección desde arriba.



Figura 3.7: Imagen del nivel 3 de la categoría prioridad

En la interfaz de usuario se dispone de un botón que al ser pulsado activa una animación que permite visualizar el punto de vista de cada uno de los vehículos, facilitando la comprensión de la situación. Para seleccionar la solución, el jugador debe hacer clic en cada vehículo en el orden de prioridad deseado. Al hacerlo, aparece un número encima del vehículo que indica su orden de prioridad. Si se vuelve a hacer clic en el mismo vehículo, este se deselecciona, actualizando automáticamente los números de los demás vehículos seleccionados. Cuando se seleccionen todos, si la solución es correcta, los vehículos se desplazarán uno a uno

Desarrollo de TeoriKart

en el orden seleccionado. Si la respuesta es incorrecta, todos los vehículos arrancarán de manera simultánea.

3.1.3.2 Manejo

Se centran en el aprendizaje del tema de señalización. Consisten en mapas con diferentes señales de tráfico y vehículos que circulan libremente. El usuario debe manejar el coche por el mapa como si fuese un conductor, respetando las normas de tráfico hasta llegar al destino. Este se podrá visualizar con una estrella en el mapa y en el minimapa de la interfaz de usuario situado en la esquina superior derecha, el cual se puede ampliar haciendo clic o pulsando la tecla M. Además, existe un dial en la parte inferior de la interfaz que indica la velocidad a la que circula el coche, ya que el usuario tendrá que respetar el límite de velocidad en caso de encontrar las señales pertinentes.



Figura 3.8: Imagen del nivel 1 de la categoría manejo

Las señales verticales y marcas viales incluidas son: semáforo, señal de stop, señal de ceda al paso, señal de prohibido el paso, cuadrícula amarilla, señal de glorieta, señal de únicas direcciones permitidas, línea continua y discontinua, zona de paso de peatones, señal de velocidad máxima. Al llegar al destino el nivel terminará y si hay errores aparecerán listados en un cuadro de texto.

Desarrollo de TeoriKart

3.1.3.3 Luces

Se centra en los tipos de luces y cuándo usarlas. Habrá mapas adaptados a situaciones en las que sea necesario encender alguna luz. Las situaciones que se han implementado son lluvia y niebla, túneles, vías insuficientemente iluminadas y deslumbramientos. Las luces que puede encender el jugador son:

- Posición.
- Corto alcance y largo alcance. Al encender esta se encienden las de posición automáticamente.
- Antinieblas traseras y delanteras.

Al empezar un nivel de luces, aparece un texto explicativo que describe la situación en la que te encuentras y a continuación, se muestran en pantalla los botones de las luces. Cuando se escogen las luces y se pulsa el botón "Aceptar", se minimiza el panel a la parte inferior izquierda de la interfaz, pudiendo cambiarlas en cualquier momento.

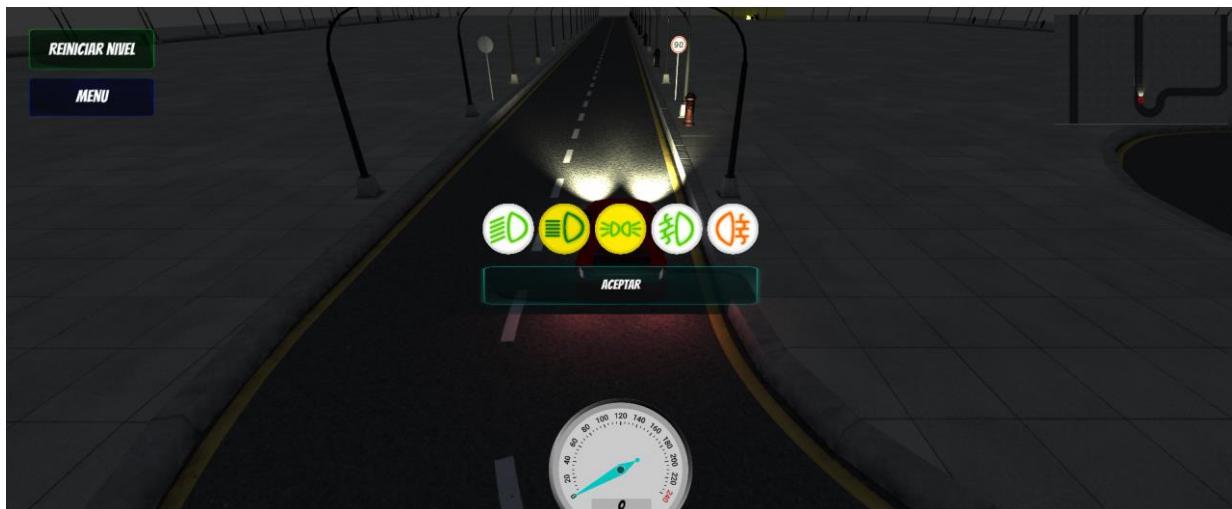


Figura 3.9: Imagen del nivel 3 de la categoría luces seleccionando las luces del vehículo

Desarrollo de TeoriKart



Figura 3.10: Imagen del nivel 3 de la categoría luces durante la conducción

En estos niveles también existe un destino objetivo al que llegar, al igual que los niveles de manejo, así como la interfaz de usuario de la velocidad y el mini mapa. Al finalizar el nivel, aparece un cuadro de texto con retroalimentación y botones para reiniciar el nivel o volver al menú.

3.2 Desarrollo

3.2.1 Jugador

Según el tipo de nivel el jugador tiene diferentes mecánicas.

3.2.1.1 En los niveles de Manejo y Luces

El jugador controla un coche por un mapa y tiene que llegar a un destino final. Se mueve con las teclas W, A, S, D y las flechas del teclado. El coche se desplaza verticalmente, pudiendo girar de forma realista, imitando el movimiento de un coche real. Con la tecla ESPACIO se frena. Puede llegar a un máximo de 120 Km/h. La velocidad se ve reflejada en la interfaz de usuario mediante un dial.

En los niveles de luces, estas se pueden encender haciendo clic en los botones de la interfaz de usuario, o mediante las teclas 1, 2, 3, 4, 5 - luces de posición, de cruce, de carretera, antinieblas delanteras, antinieblas traseras.

Desarrollo de TeoriKart

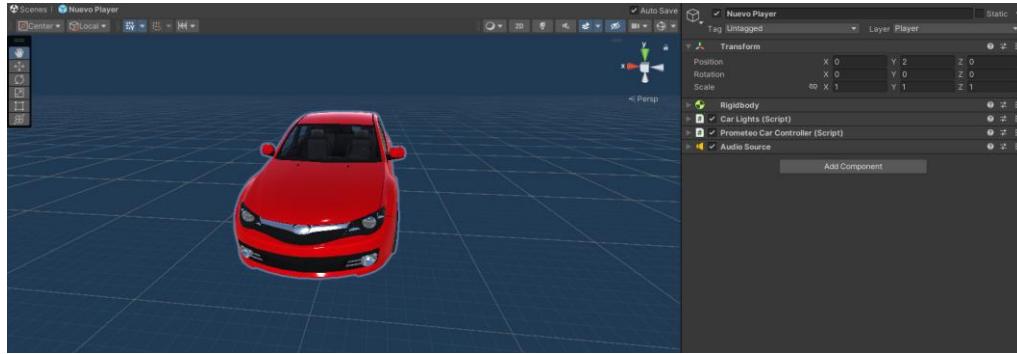


Figura 3.11: Vista del prefab del coche del jugador en Unity

En la Figura 3.11 se muestra el prefab del jugador en Unity. Para diferenciarlo del resto de coches, se le ha asignado el color rojo. Posee los siguientes scripts:

- CarLights.cs. Este script se encarga de la gestión de las luces del vehículo en los niveles pertinentes.
- PrometeoCarController.cs. Este script se descargó de la Asset Store de Unity para la conducción del vehículo. Inicialmente, se disponía de otro script para manejar la conducción del jugador, pero tras muchos cambios y ajustes no se conseguía una suavidad adecuada al girar en las curvas con suavidad, por lo que fue sustituido por este asset, el cual ha recibido varios ajustes para estar más adecuado al videojuego.

Inicialmente se había implementado un sistema GPS, con el cual se pintaban unas líneas azules que guiaban al jugador al destino del nivel por la mejor ruta desde su posición. Sin embargo, fue eliminado por dos motivos, el primero es que distraía demasiado y quitaba visibilidad a elementos del nivel, como por ejemplo las señales. El otro motivo viene motivado por el cambio del funcionamiento de los coches no controlables por el jugador, ya que estos vehículos se desplazaban siguiendo la ruta del GPS. Esta decisión se encuentra explicada en detalle en la sección 3.2.3.

Desarrollo de TeoriKart

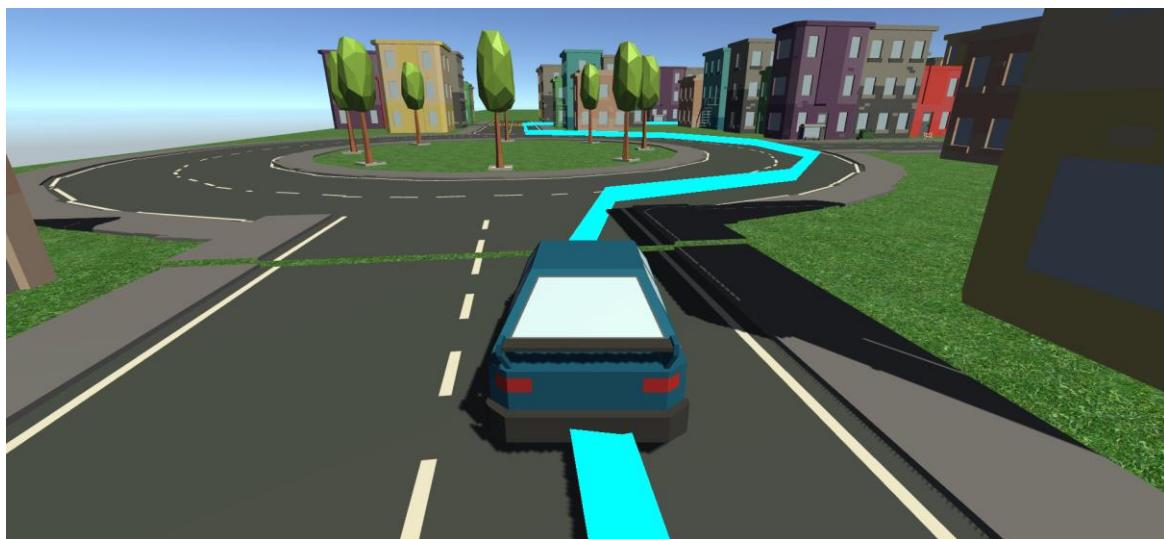


Figura 3.12: Sistema GPS en una versión previa del videojuego

Para que el jugador sepa hacia dónde tiene, se ha implementado un minimapa en la esquina superior derecha. Se puede agrandar con la tecla M y muestra el mapa desde arriba con todos los elementos con iconos representativos, incluyendo el destino con una estrella.



Figura 3.13: Vista del minimapa en la interfaz del nivel 3 de manejo



Figura 3.14: Vista del del minimapa ampliado del nivel 3 de manejo

3.2.1.2 En los niveles de Prioridad

El jugador elige el orden de prioridad mediante clics en los vehículos.

3.2.2 Mapas

Los mapas están conformados por distintas piezas colocadas en una cuadrícula, y forman las carreteras. Cada pieza tiene unas dimensiones de 50 x 50 unidades, y están divididas en 20 celdas, cada una de ellas con una dimensión de 2,5 x 2,5 unidades. Las celdas se organizan en una matriz de 20 filas y 20 columnas, lo que permite colocar los elementos con precisión dentro de cada pieza.

Desarrollo de TeoriKart

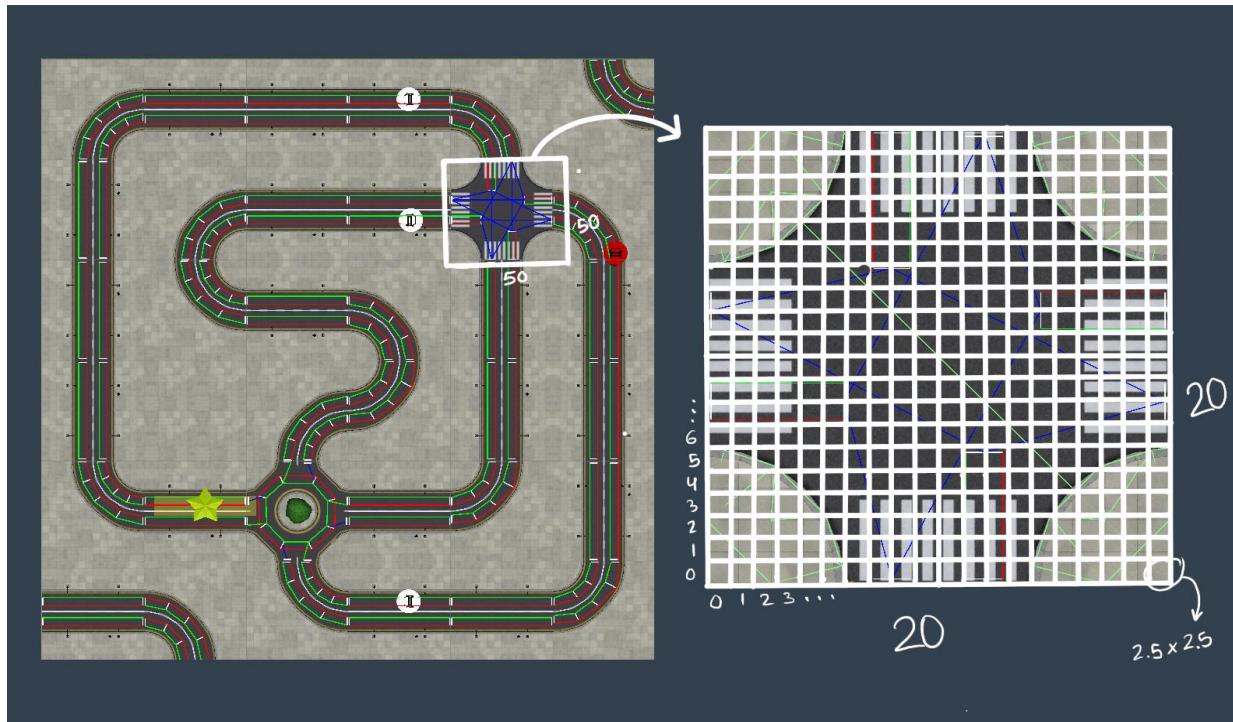


Figura 3.15: Imagen explicativa de las dimensiones de las piezas

Estas especificaciones de dimensiones se definen para la correcta colocación tanto de las piezas como de elementos dentro de ellas (coches, semáforos, señales...) dentro de la escena de Unity. Para guardar los mapas se ha implementado un editor que guarda los datos en un archivo JSON, que también se puede editar manualmente. Debe seguir la siguiente estructura:

- Descripción del nivel
- Piezas
- Jugador
- Punto de destino del jugador
- Señales verticales y marcas viales
- Coches controlados por Inteligencia Artificial
- Semáforos
- Diálogos

Es necesario indicar el tipo del nivel, y dependiendo de qué tipo es, otras especificaciones. Todos los niveles deben de tener un número identificador que debe coincidir con el del nombre del archivo.

```
"type": "Prioridad",
"nivel": 10,
"correctOrder": [
    2,
    1,
    0
],
```

Figura 3.16: Cabecera del archivo nivel10.json

En la Figura 3.16 se puede apreciar que el nivel número 10 es de tipo prioridad, especificado con la variable “type”. El array “correctOrder” es necesario en estos niveles para establecer la solución correcta de estos. Dentro del array se deben poner los identificadores de los vehículos en el orden en el que debe seleccionarlos el jugador, en este caso primero el vehículo con índice 2, luego el de índice 1 y finalmente el de índice 0.

```
"type": "Luces",
"nivel": 18,
"night": false,
"deslumbramiento": false,
"fog": true,
"rain": false,
```

Figura 3.17: Cabecera del archivo nivel18.json

En el archivo del nivel mostrado en la Figura 3.17 se indica mediante la variable type que es de la categoría luces, en esta clase de niveles se puede especificar diversas condiciones específicas mediante las siguientes variables:

- fog: si se le asigna true, aparece una niebla intensa en el cielo del nivel.
- rain: al ser true, llueve intensamente en el nivel.
- night: si se le da el valor true, el nivel es en horario nocturno.
- deslumbramiento: esta variable sólo puede activarse si se activa también la variable anterior, ya que está pensada para niveles nocturnos en los cuales el jugador tiene que evitar deslumbrar a otros vehículos.

Desarrollo de TeoriKart

En cuanto a las piezas, estas se agrupan dentro del objeto “mapa”, cada una dentro de un array según sus tipos, es decir, las rectas horizontales se definen juntas, las verticales también, y así sucesivamente. Dentro del objeto mapa, antes de las piezas, es necesario indicar cuantas piezas, filas y columnas tiene el mapa.

```
"mapa": {
    "numPiezas": 49,
    "filas": 7,
    "columnas": 7,
    "Vertical": [
        {
            "fil": 4,
            "col": 3
        },
        {
            "fil": 3,
            "col": 3
        }
    ]
}
```

Figura 3.18: Ejemplo de colocación de piezas en el JSON

En la Figura 3.18 se puede apreciar un mapa de dimensiones 7 filas x 7 columnas, con 49 piezas, indicado en “numPiezas”. Aparecen definidas 2 piezas de tipo “Vertical”, la primera de ellas colocada en la fila 4 columna 3 y la segunda en la fila 3 columna 3.

Según su colocación en el mapa, por código, a cada pieza se le asigna un identificador, la pieza situada en fila 0 columna 0 tiene el identificador 0, la situada en fila 0 columna 1 tiene el identificador 1, y así sucesivamente. No es necesario que estén definidas en orden en el archivo. Todos los modelos de las piezas deben guardarse en la carpeta Assets/Resources/PiezasPrefabs, que se muestra en la Figura 3.19.

Desarrollo de TeoriKart

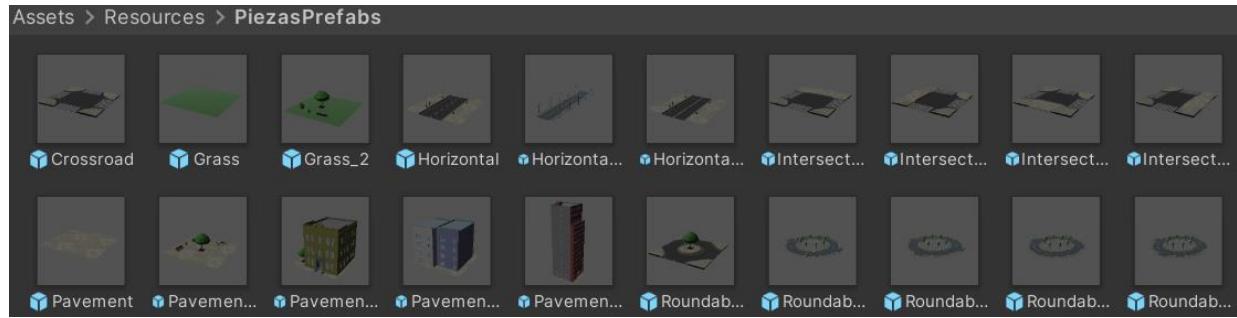


Figura 3.19: Estructura de la carpeta Assets/Resources/PiezasPrefabs

Antes de hacer hincapié en los tipos de elementos que se pueden crear dentro del nivel, se describen los componentes básicos que muchos de estos elementos comparten. En general, estos elementos siguen un esquema común que incluye los siguientes atributos:

- “pieza”: indica en qué pieza del mapa se colocará el elemento. Cada pieza tiene un índice único que la identifica dentro del mapa.
- “subPosicion”: define la posición exacta del elemento dentro de la pieza, utilizando coordenadas de fila (fil) y columna (col).
- “orientacion”: determina la dirección en la que estará orientado el elemento, con “arriba”, “abajo”, “izquierda” y “derecha” como posibles valores.

En algunos de los niveles, los usuarios tendrán que conducir su propio coche, el cual se define en el archivo de la siguiente manera:

```
"jugador": {
    "pieza": {
        "index": 5
    },
    "subPosicion": {
        "fil": 11,
        "col": 6
    },
    "orientacion": "arriba"
},
```

Figura 3.20: Ejemplo de estructura del jugador en archivo

En la Figura 3.20, el coche del jugador se coloca en la pieza con índice 0, en la subposición correspondiente a la fila 10 y columna 9, con una orientación hacia arriba. Este elemento no añade parámetros al esquema descrito anteriormente.

Desarrollo de TeoriKart

En ciertos niveles puede que el objetivo principal del nivel sea que el jugador vaya de un punto A a un punto B. En ese caso, se puede definir en el archivo el siguiente elemento:

```
"targetJugador": {  
    "pieza": {  
        "index": 8  
    },  
    "subPosicion": {  
        "fil": 11,  
        "col": 9  
    }  
},
```

Figura 3.21: Ejemplo de estructura del objetivo en archivo

En el ejemplo mostrado en la Figura 3.21, el objetivo del jugador se coloca en la pieza con índice 1, en la subposición correspondiente a la fila 9 y columna 0. Este elemento se representa en el nivel mediante un halo de luz amarilla con una estrella que indica al jugador dónde debe dirigirse.

Todas las señales y marcas viales siguen la estructura del esquema, las únicas con una variación son las señales de prohibición de velocidad, que poseen un parámetro extra.

```
"stops": [  
    {  
        "pieza": {  
            "index": 29  
        },  
        "subPosicion": {  
            "fil": 0,  
            "col": 14  
        },  
        "orientacion": "derecha"  
    }  
],  
  
"maxVelocidad": [  
    {  
        "velocidad": 90,  
        "pieza": {  
            "index": 14  
        },  
        "subPosicion": {  
            "fil": 14,  
            "col": 18  
        },  
        "orientacion": "abajo"  
    }  
],
```

Figura 3.22: Ejemplos de estructura de varias señales en archivo

La Figura 3.22 muestra la creación de dos señales, una señal de Stop y otra de Velocidad Máxima, en la cual se especifica que es 90 km/h mediante la variable "velocidad". Las señales y marcas viales que se pueden crear son las siguientes:

- Señal R2 Detención Obligatoria: se define dentro del array "stops".

Desarrollo de TeoriKart

- Señal R301 Velocidad Máxima: se define dentro del array "maxVelocidad". Mediante la variable "velocidad" se especifica la velocidad máxima de la señal, pudiendo este valor encontrarse entre 20 km/h y 120 km/h, siendo siempre múltiplo de 10.
- Señal R101 Entrada Prohibida: se define dentro del array "prohibidos".
- Señal R1 Ceda el Paso: se define dentro del array "cedas".
- Señal R413 Alumbrado corto alcance: se define dentro del array "iniLuz".
- Señal R400c Sentido Obligatorio: se define dentro del array "frente".
- Señal R403a Únicas Direcciones Permitidas: se define dentro del array "frenteDcha".
- Señal R403b Únicas Direcciones Permitidas: se define dentro del array "frentelzq".
- Cuadrícula de marcas amarilla M-7.10: indica que no se debe entrar en la intersección si se corre el riesgo de quedar detenido en ella, obstaculizando la circulación. Es el único de estos elementos que no tiene el parámetro "orientación" ya que es una marca vial.

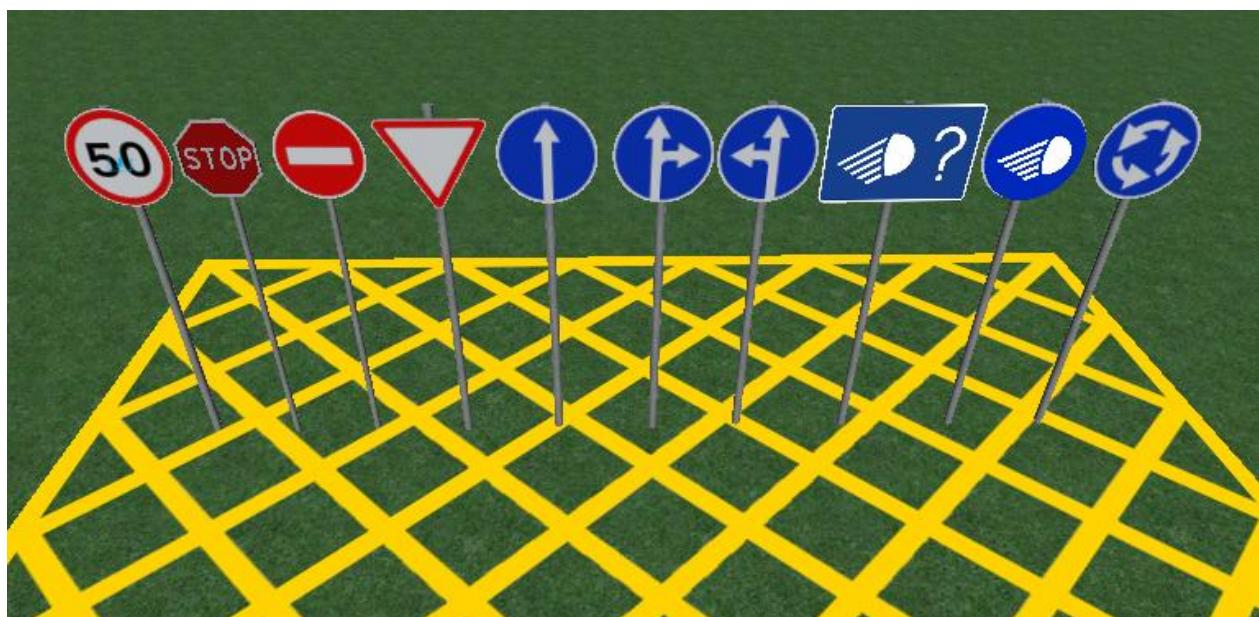


Figura 3.23: Señales y marcas viales disponibles en el videojuego

Desarrollo de TeoriKart

Los coches que no son controlados por el jugador presentan la siguiente estructura:

```
"IACars": [
    {
        "pieza": {
            "index": 31
        },
        "subPosicion": {
            "fil": 11,
            "col": 14
        },
        "orientacion": "arriba",
        "branchTo": 1,
        "vehicle": "ambulance"
    },
    {
        "pieza": {
            "index": 41
        },
        "subPosicion": {
            "fil": 3,
            "col": 11
        },
        "orientacion": "izquierda",
        "branchTo": 1
    }
],
```

Figura 3.24: Creación de vehículos en el nivel 2 de prioridad

Además de los componentes básicos, se aprecian otras dos variables. Por un lado, “vehicle” especifica la clase de vehículo que se crea, pudiendo ser los valores “ambulance” para una ambulancia, “bus” para un autobús, y “car” para un turismo, aunque en este caso no es necesario especificarlo ya que es el valor por defecto.

La variable “branchTo” sólo se utiliza en los niveles de tipo prioridad para indicar en qué dirección saldrá cada uno de los coches de la intersección. En los niveles de otro tipo, esta variable no se usa, en los de prioridad, el valor 0 para ir hacia el Este del coche, 1 hacia el Norte y 2 hacia el Oeste.

Desarrollo de TeoriKart



Figura 3.25: Imagen del nivel 1 de prioridad, donde el jugador debe decidir el orden de prioridad en un cruce

En la Figura 3.25 podemos apreciar 3 vehículos en una intersección. En la parte delantera de los vehículos se puede apreciar una flecha azul para indicar hacia dónde quieren ir, estas flechas no aparecen en el videojuego, simplemente es para ayudar a esta explicación. Cuando un vehículo va a torcer, dispone de intermitentes para indicar al jugador hacia donde se dirige.

Tanto el vehículo gris como el amarillo quieren continuar recto (hacia su Norte), por lo que su valor de la variable branchTo se define con el valor 1. El coche amarillo, al querer torcer a la izquierda, su branchTo tiene el valor 2.

Se pueden colocar semáforos en los niveles de la siguiente manera:

Desarrollo de TeoriKart

```
"semaforos": [
    {
        "pieza": {
            "index": 13
        },
        "subPosicion": {
            "fil": 4,
            "col": 2
        },
        "orientacion": "abajo",
        "greenSeconds": 20.0,
        "amberSeconds": 2.0,
        "redSeconds": 15.0,
        "initialLight": "green"
    }
],
```

Figura 3.26: Ejemplo de creación de un semáforo

En este ejemplo mostrado en la *Figura 3.26*, el semáforo se coloca en la pieza con índice 13, en la subposición correspondiente a la fila 4 y columna 2, con orientación hacia abajo. Además, este elemento posee los siguientes parámetros adicionales al esquema básico:

- “greenSeconds”: este parámetro define la duración de la luz verde en segundos. En este ejemplo, la luz verde dura 20 segundos.
- “amberSeconds”: este parámetro define la duración de la luz ámbar (amarillo) en segundos. En este ejemplo, la luz ámbar dura 2 segundos.
- “redSeconds”: este parámetro define la duración de la luz roja en segundos. En este ejemplo, la luz roja dura 15 segundos.
- “initialLight”: este parámetro define el estado inicial del semáforo cuando comienza el nivel. En este ejemplo, el estado inicial del semáforo es “green” (verde).

El juego cuenta con un sistema de diálogos indicativos para guiar al jugador durante los niveles. Los diálogos se dividen en dos grupos principales:

- “levelDialogs”: estos diálogos se muestran al inicio del nivel y proporcionan información, instrucciones o contexto adicional al jugador.

Desarrollo de TeoriKart

- “completedDialogs”: estos diálogos aparecen cuando el jugador completa el nivel correctamente, ofreciendo felicitaciones, resúmenes o cualquier mensaje final.
- “wrongDialogs”: aparecen en combinación con mensajes de errores específicos. La finalidad de estos diálogos es dar pequeñas pistas al jugador, especialmente en los niveles de prioridad.

```
"levelDialogs": [  
    "En este nivel, te encuentras con una intersección en la que participan un autobús y un coche",  
    "Tu tarea será analizar cuidadosamente la situación y decidir el orden de prioridad según las normas de tráfico.",  
    "Observa bien los detalles desde sus perspectivas y toma la decisión adecuada. ¡Buena suerte!"  
],  
"completedDialogs": [  
    "¡Buen trabajo! Has tomado la decisión correcta.",  
    "Recuerda que, aunque uno de los vehículos sea un autobús, la prioridad la tiene el coche porque continúa recto, según"  
],  

```

Figura 3.27: Diálogos definidos para el nivel 4 de prioridad

Debido a que la creación de los niveles es un poco tediosa, se ha desarrollado un editor de niveles para facilitar este proceso, se explica en detalle en la sección 3.2.6.

3.2.3 Coches controlados por inteligencia artificial

Los coches complementarios que se mueven por inteligencia artificial suponen un papel importante en el videojuego, ya que permiten que los niveles sean más variados y aportan realismo. Inicialmente, los coches funcionaban mediante el sistema NavMesh¹¹ de Unity, eran agentes y en las piezas se generaba una carretera por la cual se podían mover. Según ha avanzado el desarrollo y mejorado el videojuego, se ha tomado la decisión de prescindir de todo lo relacionado con NavMesh por varios motivos. El principal es que mediante el sistema GPS, tras muchas horas, no se ha conseguido que los vehículos distingan por qué carril deben circular. Otros problemas han sido la detección de otros vehículos y la posibilidad de hacer bake correctamente en ejecución, ya que el mapa, como se ha explicado anteriormente, se crea por archivo.

¹¹ <https://docs.unity3d.com/ScriptReference/AI.NavMesh.html>

Desarrollo de TeoriKart

El funcionamiento final consiste en un sistema de puntos llamados Waypoints, en cada una de las piezas que componen los mapas. Cada pieza cuenta con una sucesión de Waypoints por los cuales se desplazan los coches, en función del sentido por el que conducen.

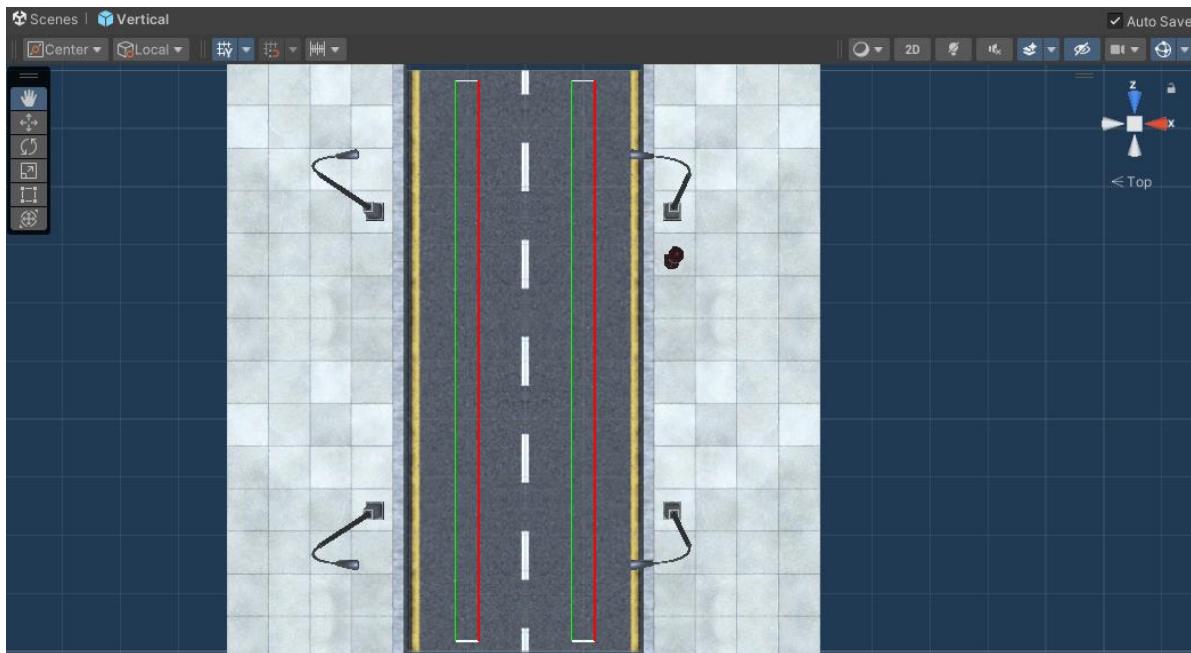


Figura 3.28: Pieza vertical en el inspector de Unity

En la Figura 3.28 se puede observar una de las piezas, en la que se aprecian dos caminos, uno por cada carril de la vía. Cada uno de los caminos de esta pieza cuenta con dos Waypoints, uno en cada extremo del carril. Los coches, una vez llegan a un Waypoint, buscan el siguiente en la dirección en la que van, dentro de la pieza actual. En caso de que la pieza actual no disponga de más Waypoints por los que pueda continuar su recorrido, el coche busca la siguiente pieza en la dirección que sigue, y se dirige al Waypoint del extremo por el que va a entrar.

Desarrollo de TeoriKart

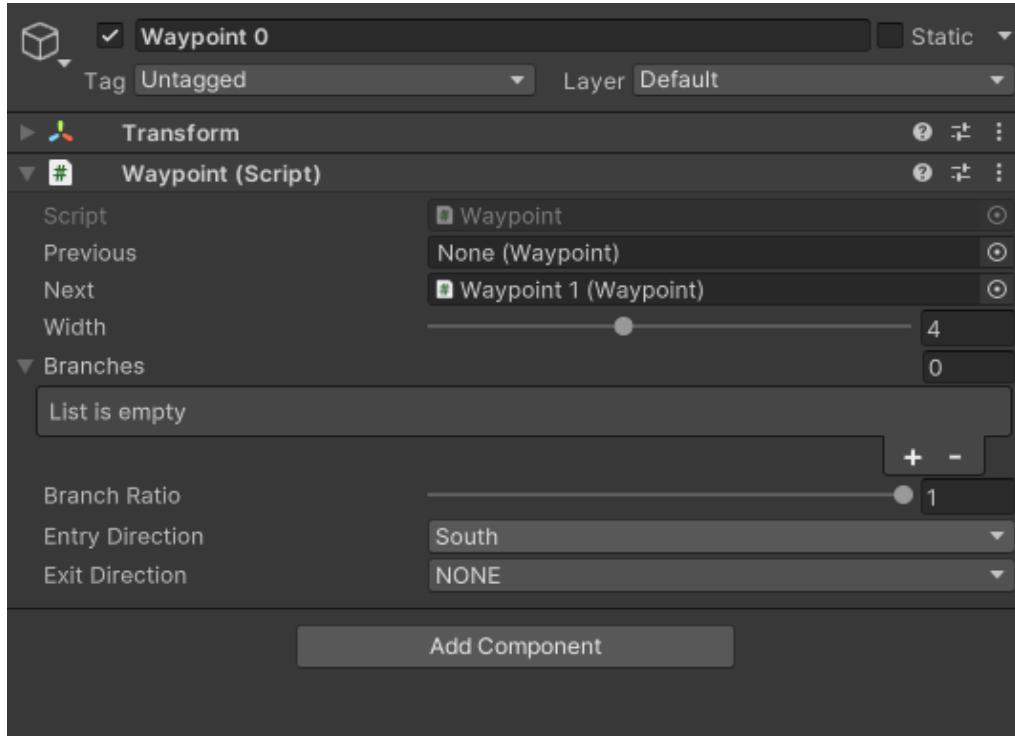


Figura 3.29: Vista del script Waypoint en el inspector de Unity

Para llevar a cabo ese cambio de pieza, mediante código, se compara la dirección de salida “Exit Direction” del Waypoint en el que se encontraba el coche (el último de esa pieza) y se busca un Waypoint en la siguiente pieza con una dirección de entrada “Entry Direction” opuesta. Las direcciones son los cuatro puntos cardinales del sistema de referencia cartesiano. Todas las piezas que tienen carretera tienen Waypoints, y por defecto las carreteras son de doble sentido.

En caso de que existan varias salidas posibles, por ejemplo, en una intersección o una glorieta, se han implementado Branches, que son Waypoints que salen del mismo punto anterior y conectan con otro camino. Tienen una variable Branch Ratio, que es un porcentaje de desvío del camino principal hacia las ramas.

Desarrollo de TeoriKart

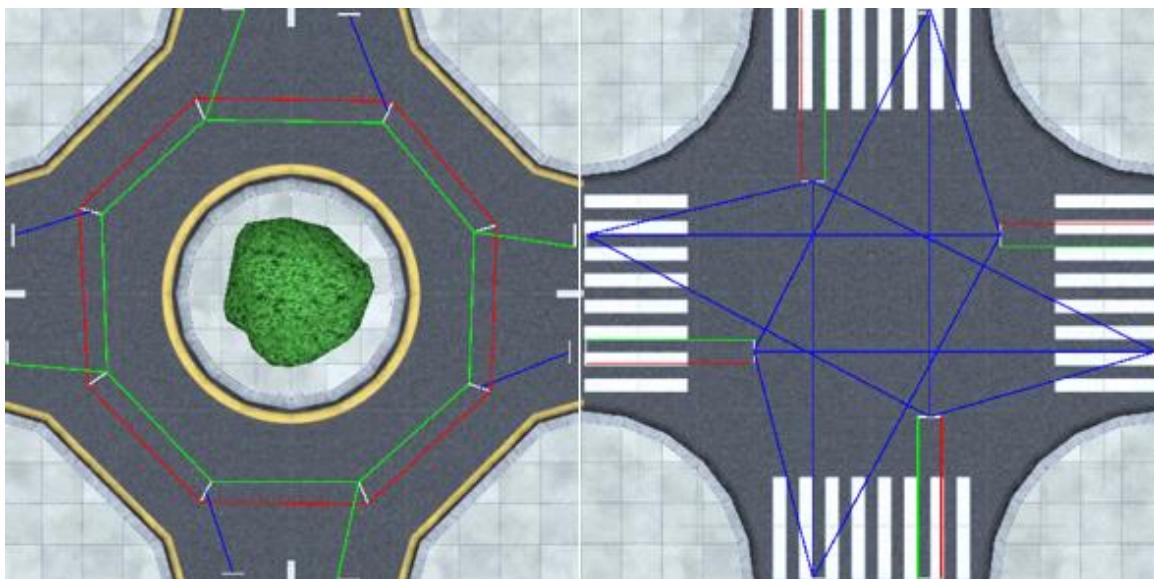


Figura 3.30: Imágenes de los Waypoints de las piezas Glorieta e Intersección

En la Figura 3.30 se pueden observar dos piezas con varias salidas, las líneas de color azul representan las Branches, mientras que las verdes y rojas son un camino de Waypoints consecutivos entre sí.

Para agilizar la creación y colocación de estos Waypoints, se ha creado una herramienta dentro de Unity llamada WaypointEditor.

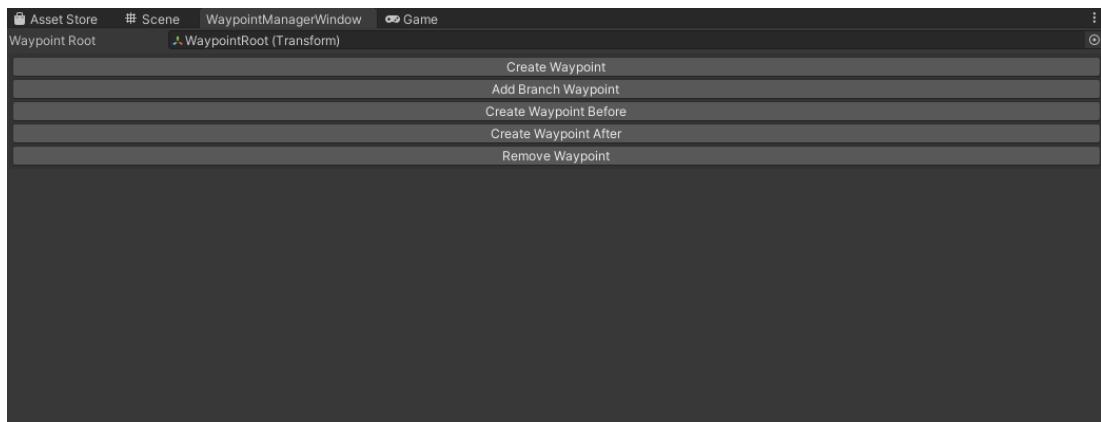


Figura 3.31: Vista de la herramienta de creación de Waypoints en Unity

La herramienta necesita que se coloque en la variable “Waypoint Root” al padre de los Waypoints de la pieza, que es un objeto vacío. Una vez colocado, permite crear un Waypoint nuevo, ya sea sin conexiones o establecerlo como anterior o posterior de otro Waypoint ya existente, eliminar un Waypoint y crear un Waypoint de tipo Branch. Además, permite colocarlos en la pieza y ajustar su tamaño.

Desarrollo de TeoriKart

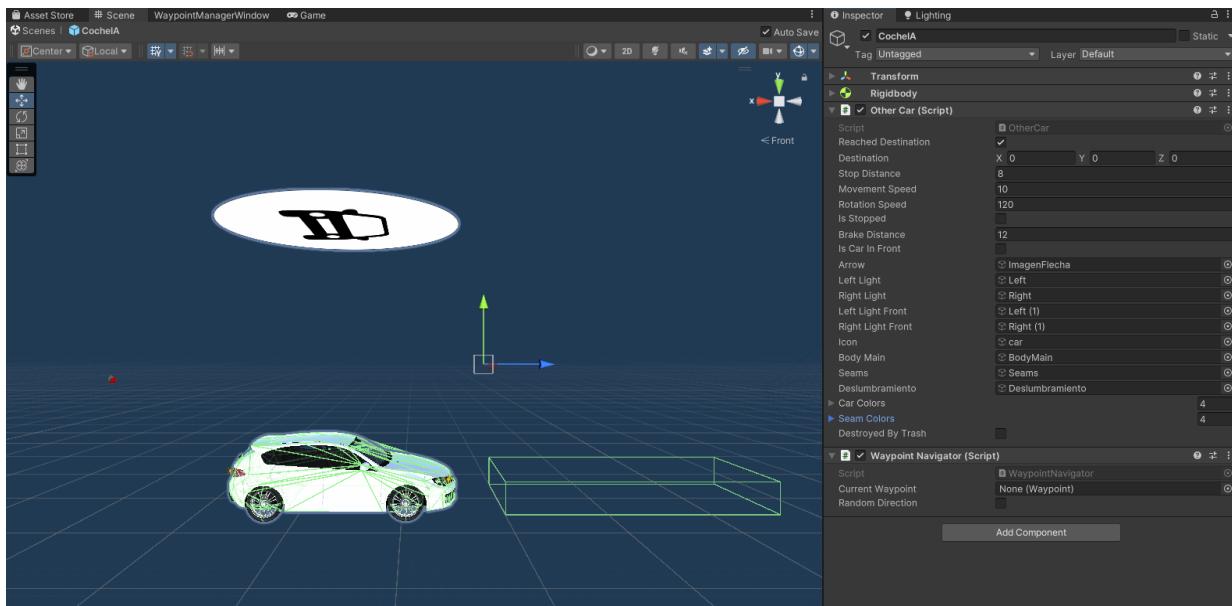


Figura 3.32: Vista del prefab de los coches de Inteligencia Artificial en Unity

El prefab mostrado en la Figura 3.32 contiene los siguientes scripts:

- OtherCar.cs. Este script se encarga del comportamiento general del coche, su movimiento, respuesta a eventos y cuando se seleccionan en los niveles de prioridad.
- WaypointNavigator.cs. Este script gestiona toda la lógica relacionada con los Waypoints, determinando el siguiente Waypoint al que debe ir el coche y gestionando las transiciones entre piezas.
- PreventCrash.cs. Este script está asociado a un objeto hijo con un collider¹² de tipo trigger en la parte delantera del vehículo, que se encarga de prevenir choques con el jugador y otros coches.
- Deslumbramiento.cs. Asociado a otro objeto hijo, este script se emplea en algunos niveles de la categoría luces para comprobar si el jugador deslumbra a un vehículo de la IA con las luces largas.

Por otro lado, tiene una imagen para los iconos del minimapa y un Mesh Collider para que puedan chocar con el jugador. Estos vehículos, además de turismos, presentan una variante para ambulancia y autobús con el mismo funcionamiento.

¹² <https://docs.unity.cn/520/Documentation/ScriptReference/Collider.html>

Desarrollo de TeoriKart

La ambulancia se puede crear tanto en situación de emergencia como normal, para aportar variedad especialmente a los niveles de prioridad.

3.2.4 Elementos de tráfico

Los elementos de tráfico del juego (señales, semáforos y marcas viales) se han desarrollado siguiendo un esquema similar. El funcionamiento de todos ellos se controla mediante un trigger que, en caso de entrar en contacto con el jugador, verifica si ha respetado la norma de tráfico asociada al elemento. En caso de que el contacto se produzca con un vehículo no controlable, el elemento le obligará a hacer la opción correcta, por ejemplo, detenerse en un semáforo en rojo y no reanudar su camino hasta que pase a estar verde.

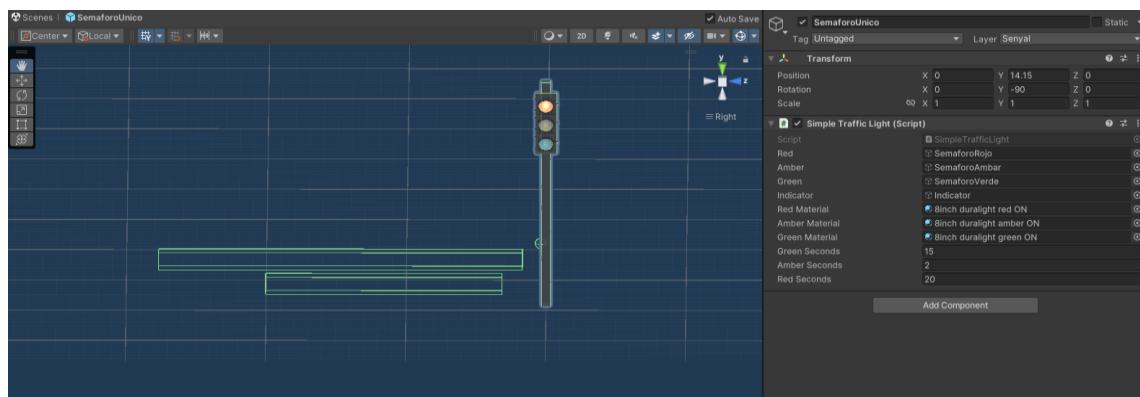


Figura 3.33: Vista del prefab del semáforo, destacan 2 triggers (rectángulos verdes)

3.2.5 Condiciones climáticas y ambientales

En los niveles de la categoría luces, como se ha explicado en la sección 3.2.2, existe la posibilidad de configurar los niveles con distintas condiciones climáticas y ambientales. Las posibles variaciones que se pueden incluir son las siguientes:

- Niebla Intensa: En caso de activar la niebla, se añade a la cámara una imagen blanquecina que ocupa toda la interfaz para simular el efecto. En caso de escoger las luces correctas del nivel, la niebla se hace menos espesa para otorgar retroalimentación al jugador.
- Lluvia intensa: Para la lluvia finalmente se dispone de un asset de Unity que al crearse en la escena aporta un feedback tanto sonoro como visual.

Desarrollo de TeoriKart

- Noche. Los niveles pueden configurarse para tener una ambientación nocturna, al igual que las configuraciones anteriores, para aportar diversidad a los posibles niveles relacionados con preguntas del tema de alumbrado y dispositivos de iluminación. Para poner la escena del juego de noche, se ha utilizado la ventana de Unity RenderSettings¹³ para sustituir el Skybox¹⁴ por uno más oscuro y DynamicGI¹⁵ para actualizar la iluminación de la escena. Esta configuración, además de servir para diseñar niveles en los que hay que seleccionar correctamente las luces, también se puede complementar con la posibilidad de deslumbrar a otros vehículos que circulen por la vía si se activa la opción correspondiente.

3.2.6 Editor de Niveles

Este editor es una escena de Unity que permite crear fácilmente el archivo JSON de un nivel sin necesidad de colocar todas las piezas y elementos manualmente mediante texto.

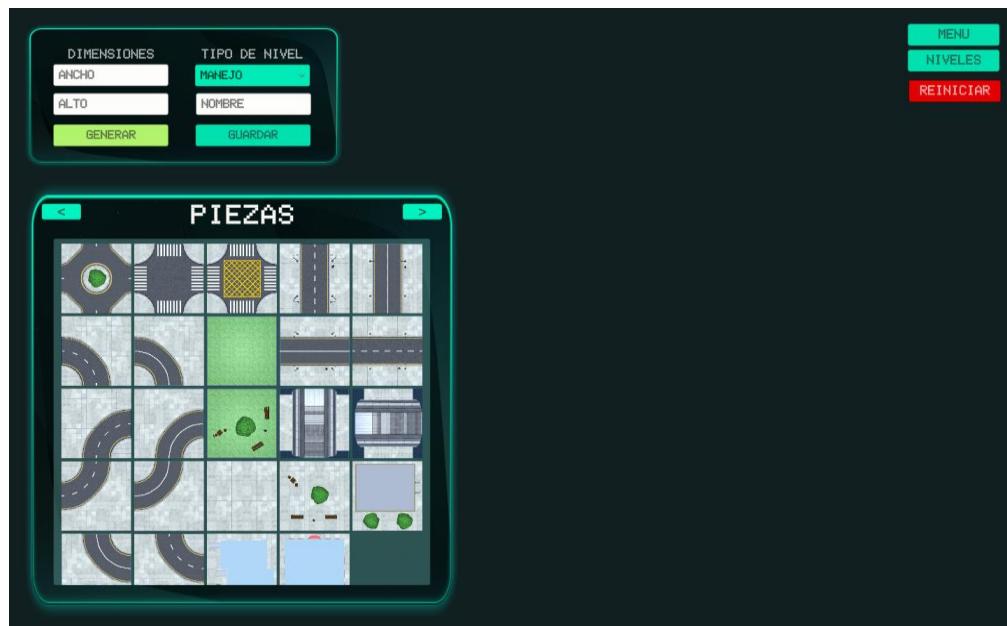


Figura 3.34: Imagen de la escena del editor

En la Figura 3.34 se aprecia la vista general del editor. En la esquina superior izquierda se encuentran diversos botones para realizar la configuración del mapa.

¹³ <http://docs.unity3d.com/6000.1/Documentation/ScriptReference/RenderSettings.html>

¹⁴ <https://docs.unity3d.com/6000.0/Documentation/ScriptReference/Skybox.html>

¹⁵ <https://docs.unity3d.com/es/530/ScriptReference/DynamicGI.html>

Desarrollo de TeoriKart

Por un lado, para las dimensiones, existen dos cuadros de texto “ANCHO” y “ALTO”, que sirven para establecer las filas y columnas del mapa (entre 3 y 10) en los niveles de manejo y luces. Al lado se encuentra un desplegable de color azul con el que se puede seleccionar el tipo de nivel a crear, un cuadro de texto para introducir el nombre deseado para el nivel y un botón “GUARDAR” para guardarlo en archivo.

En los niveles de las categorías manejo y luces es necesario introducir unas dimensiones en los cuadros de ancho y alto y generar el mapa. Una vez hecho esto, se crea una cuadrícula con las dimensiones escogidas, que representa el mapa. En caso de querer cambiar de dimensiones o empezar de nuevo el mapa, se puede pulsar el botón rojo “REINICIAR”, situado en la esquina superior derecha.

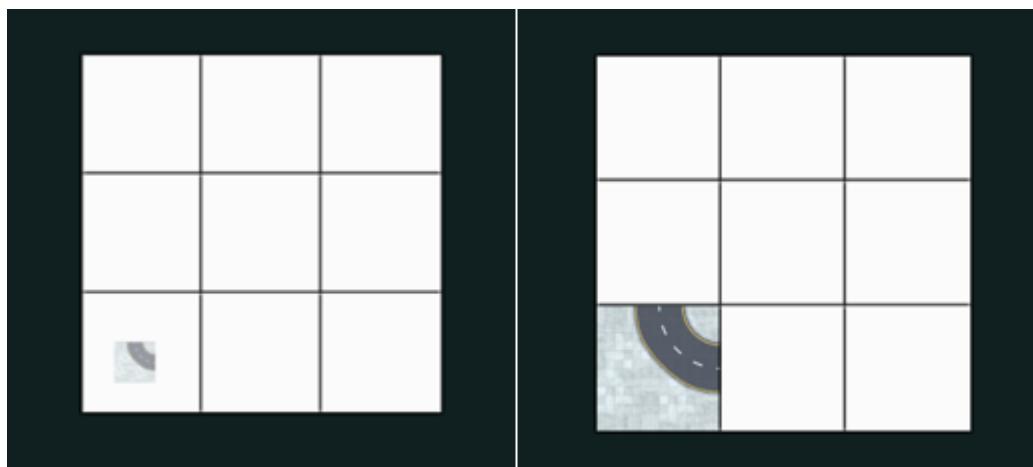


Figura 3.35: Mecánica de arrastrar y soltar una pieza en el mapa

Para colocar las piezas del mapa, se tienen que arrastrar individualmente desde la sección “PIEZAS” y soltarlas en la baldosa deseada de la cuadrícula. Una vez soltada, la baldosa se rellena con la pieza correspondiente.

Para incluir elementos como vehículos, señales o semáforos en las piezas, primero se tiene que cambiar al panel de “ITEMS” haciendo clic en las flechas del panel.



Figura 3.36: Panel ITEMS del editor

Con el objetivo de dar retroalimentación a los usuarios que quieran crear un mapa, al colocar el cursor del ratón encima de cualquier pieza y elemento, aparece un letrero azul con un identificador.

Para colocar los elementos en la pieza deseada, se tiene que hacer clic sobre ella, ya que aumenta considerablemente de tamaño y se hace más sencilla dicha colocación.

Desarrollo de TeoriKart

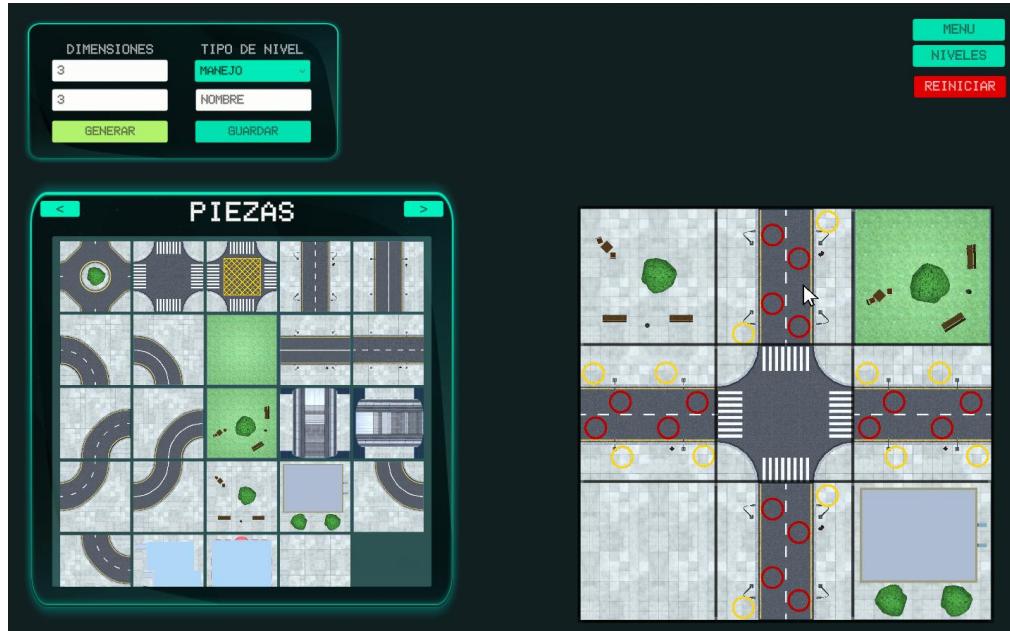


Figura 3.37: Vista del editor seleccionando la pieza (2,1) del mapa

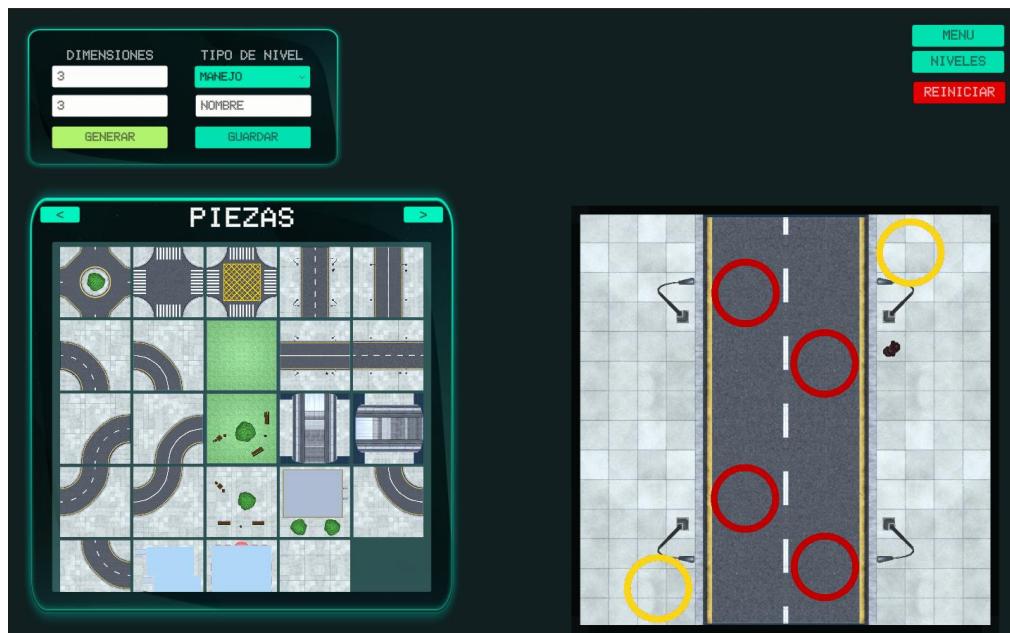


Figura 3.38: Vista del editor con la pieza de la Figura 3.37 aumentada

Una vez ampliada la pieza, en algunas de ellas se pueden observar unas circunferencias de color rojo y amarillo, como se aprecia en la Figura 3.38. Estas circunferencias representan los lugares donde se pueden colocar los elementos, en las de color rojo se pueden arrastrar aquellos que se pueden crear en carretera, ya sean vehículos o el punto objetivo de los niveles. En las amarillas, se pueden colocar los demás elementos, conformados por semáforos y varias señales de tráfico. Para

Desarrollo de TeoriKart

cambiar entre las piezas y los elementos, se tiene que hacer clic en las flechas de los paneles que los contienen.

Para colocar estos elementos se tienen que arrastrar desde la sección “ITEMS” y soltar en dichas circunferencias, siguiendo la misma mecánica que con las piezas.

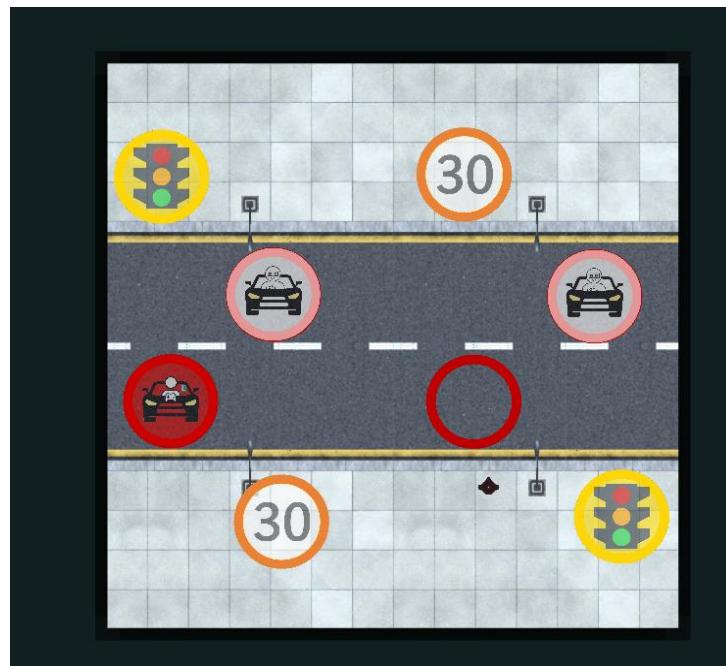


Figura 3.39: Vista de pieza ampliada con elementos colocados en ella

Las circunferencias están configuradas para que, al tener un objeto en ellas, se guarde en el JSON ese tipo de objeto en dicha pieza, mediante el script “InteractivePoint”. Este script posee 4 variables públicas para gestionar dicha creación. Fil y Col, para la subposición {fil,col} dentro de la pieza, Orientacion para la orientación del objeto. Accepted Type establece el tipo de elemento aceptado, que puede ser de tipo Traffic Elem (colocable en las circunferencias amarillas) o de tipo Car (colocable en las circunferencias rojas).

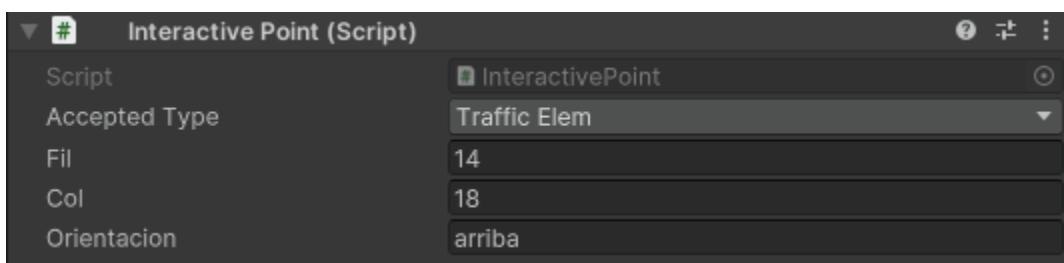


Figura 3.40: Vista del componente InteractivePoint en la jerarquía de Unity

Desarrollo de TeoriKart

Si se suelta un elemento de tráfico en la circunferencia asociada al script InteractivePoint mostrado en la Figura 3.40, se crea el objeto en la fila 14 columna 18 de la pieza asociada con una orientación hacia arriba. Para guardar el nivel en un archivo, se debe hacer clic en el botón “Guardar”, si hay espacios vacíos en la cuadrícula, se añade un pavimento por defecto.

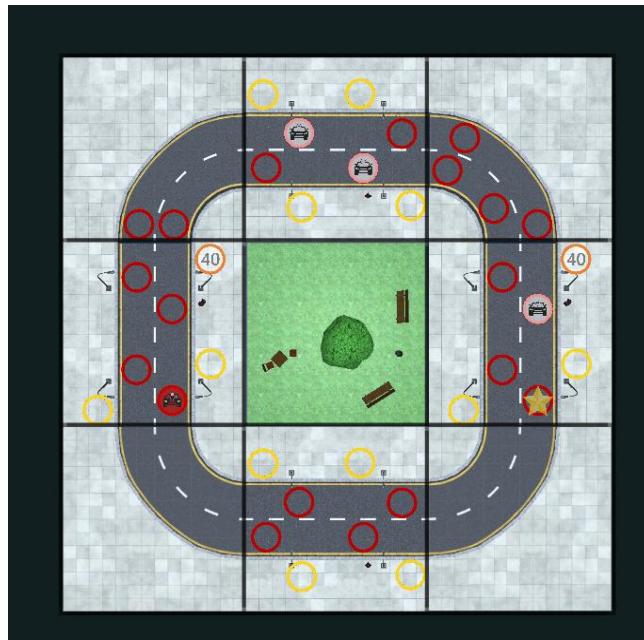


Figura 3.41: Vista de nivel de manejo en el editor

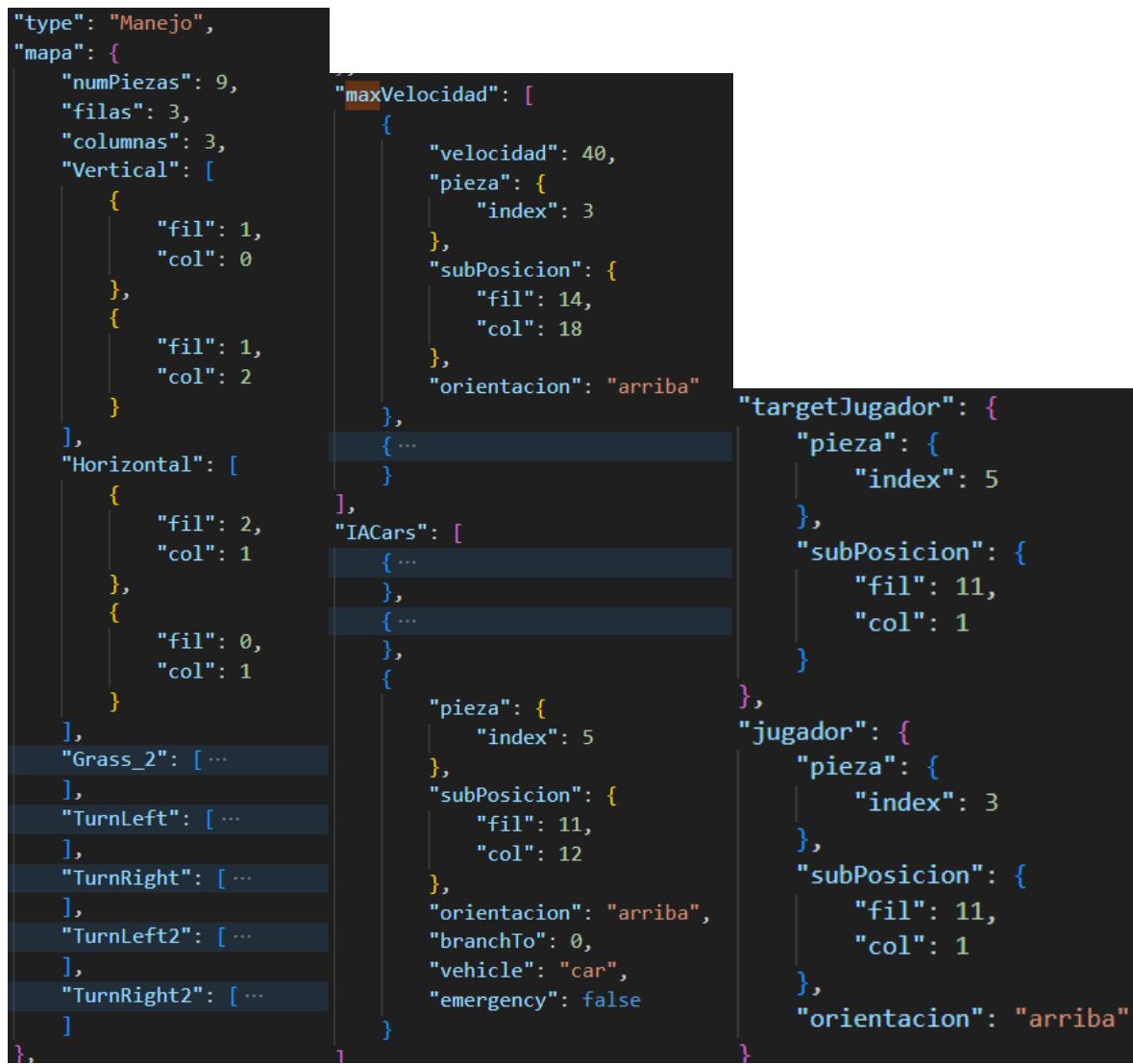
En el mapa de ejemplo de la Figura 3.41 se crea un nivel de la categoría manejo con unas dimensiones de 3x3. Se han colocado en total 2 señales de velocidad, 3 coches manejados por Inteligencia Artificial, el jugador y el objetivo al que tiene que llegar.



Figura 3.42: Panel de configuración de nivel

Desarrollo de TeoriKart

Al igual que en la Figura 3.42, una vez esté listo el nivel hay que guardarla con un nombre identificativo del mismo. En los niveles tanto de manejo como de luces es necesario que haya en el mapa tanto un jugador (coche rojo) como un objetivo (estrella), en caso contrario no se puede guardar el nivel. En cuanto a los niveles de prioridad, deben tener mínimo dos coches no manejables, sino no podrán guardarse, además, no es recomendable colocar ni al jugador ni un objetivo, aunque si se colocan en estos niveles, no se crean.



```
"type": "Manejo",
"mapa": {
    "numPiezas": 9,
    "filas": 3,
    "columnas": 3,
    "Vertical": [
        {
            "fil": 1,
            "col": 0
        },
        {
            "fil": 1,
            "col": 2
        }
    ],
    "Horizontal": [
        {
            "fil": 2,
            "col": 1
        },
        {
            "fil": 0,
            "col": 1
        }
    ],
    "Grass_2": [...],
    "TurnLeft": [...],
    "TurnRight": [...],
    "TurnLeft2": [...],
    "TurnRight2": [...]
},
"maxVelocidad": [
    {
        "velocidad": 40,
        "pieza": {
            "index": 3
        },
        "subPosicion": {
            "fil": 14,
            "col": 18
        },
        "orientacion": "arriba"
    },
    ...
],
"IACars": [
    ...
],
{
    "pieza": {
        "index": 5
    },
    "subPosicion": {
        "fil": 11,
        "col": 1
    }
},
"targetJugador": {
    "pieza": {
        "index": 5
    },
    "subPosicion": {
        "fil": 11,
        "col": 12
    },
    "orientacion": "arriba",
    "branchTo": 0,
    "vehicle": "car",
    "emergency": false
},
"jugador": {
    "pieza": {
        "index": 3
    },
    "subPosicion": {
        "fil": 11,
        "col": 1
    },
    "orientacion": "arriba"
}
```

Figura 3.43: Archivo JSON correspondiente al nivel de la Figura 3.41

Al guardar el mapa, se crea un archivo con todos los elementos que se han colocado en el editor siguiendo la estructura definida en el apartado anterior.

Desarrollo de TeoriKart

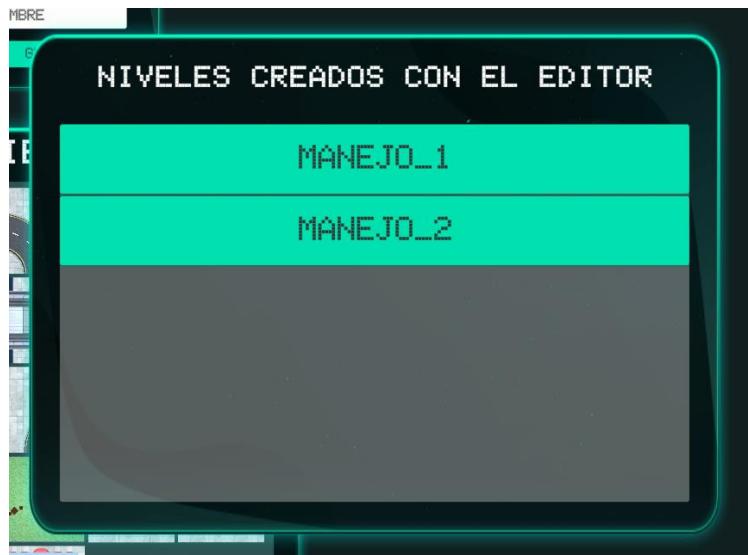


Figura 3.44: Panel con la lista de niveles del editor

Hay un botón “Niveles” en la esquina superior derecha, que abre el panel de la Figura 3.36, y muestra una lista con todos los niveles creados con el editor. Al hacer clic en uno se abre el nivel para jugar.

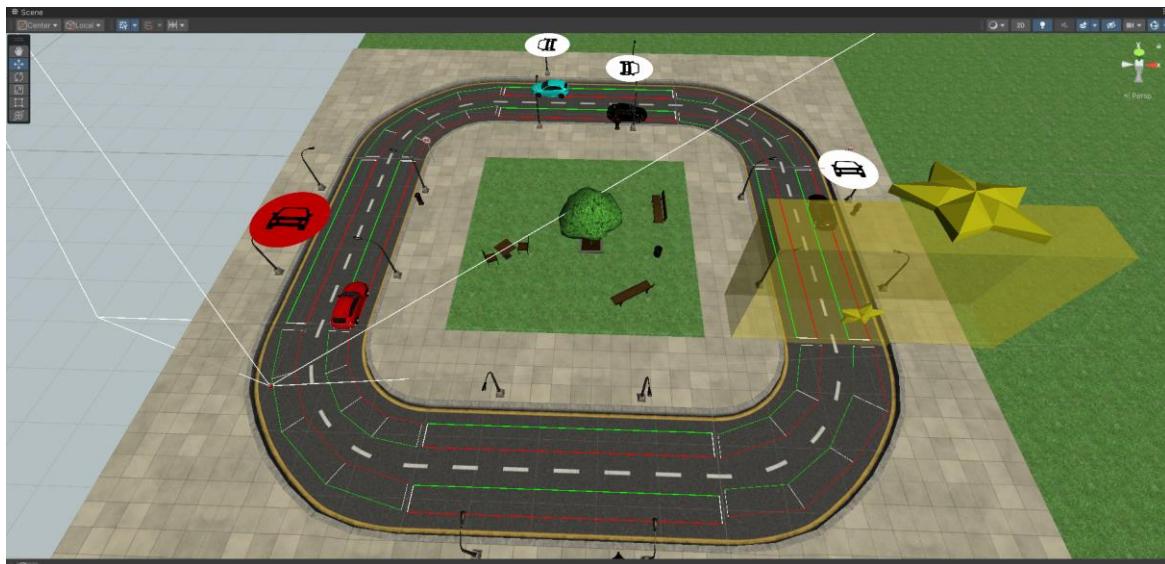


Figura 3.45: Escena de juego del nivel de la Figura 3.42

En caso de que el nivel sea de la categoría luces, aparece en la parte superior del editor un panel con diferentes casillas para poder añadir al nivel diferentes condiciones ambientales y climáticas. En este panel también se debe seleccionar las luces correctas para completar el nivel.

Desarrollo de TeoriKart



Figura 3.46: Editor con nivel de luces, con su panel de configuración resaltado

En la Figura 3.46, el nivel se configura de noche, controlando el evento de deslumbramiento y las luces de la solución son las largas y las de posición.

Por otra parte, la creación de un nivel de prioridad es diferente. Una vez se selecciona el tipo de nivel en el desplegable, se crea automáticamente un mapa de dimensiones 9x9 con la estructura que se ha seguido para este tipo de niveles.

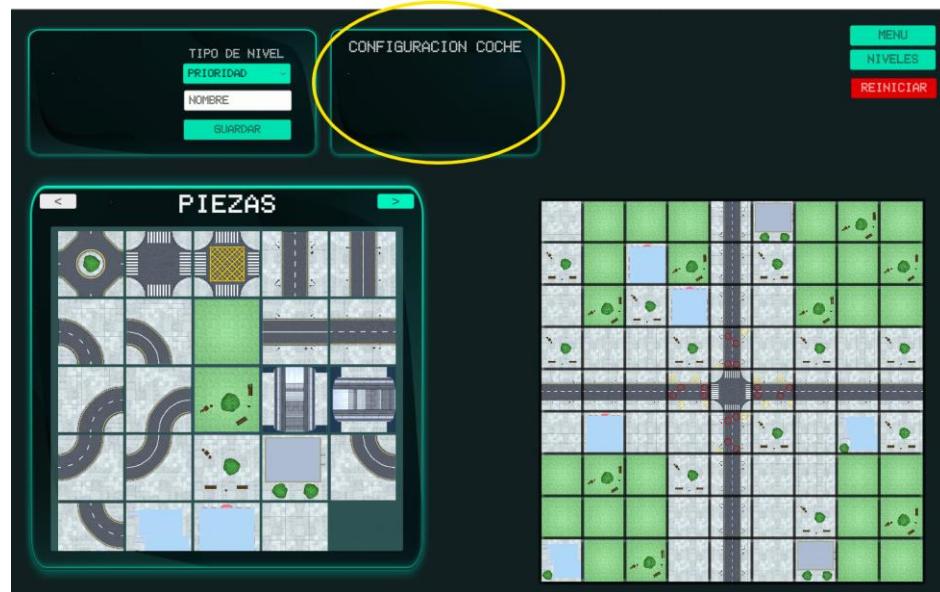


Figura 3.47: Editor durante la creación de un nivel de prioridad, con el nuevo panel de configuración resaltado

Desarrollo de TeoriKart

Como se puede observar en la Figura 3.47, además del mapa mencionado, desaparece el área para introducir las dimensiones y aparece un nuevo panel para configurar los coches no manejables por el jugador. Una vez arrastrados a la pieza deseada, en el panel aparecen sus opciones de configuración, aunque haciendo clic sobre el coche deseado se pueden volver a cambiar las opciones en cualquier momento.

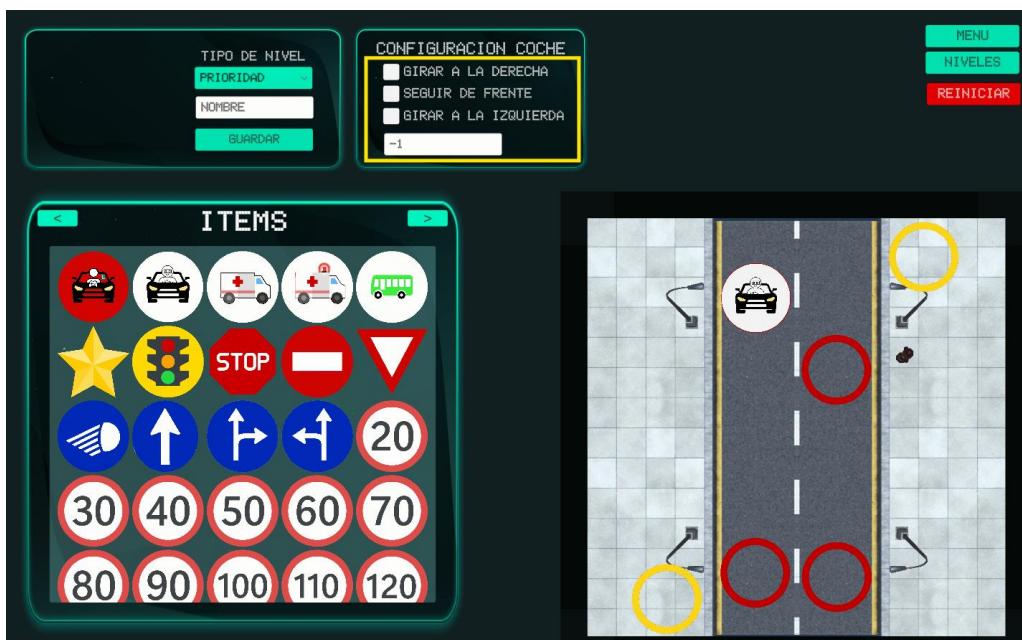


Figura 3.48: Panel de configuración para el vehículo colocado en el mapa

Como se muestra en la Figura 3.48, al colocar el vehículo, en el panel se muestran diversas opciones. Las tres primeras casillas representan la dirección que se quiere que el vehículo tome en la intersección, recto, girar a la izquierda o girar a la derecha. El cuadro de texto restante es para introducir el índice de ese vehículo en la solución, es decir, el vehículo con índice 0 es el primero en la solución, el que tiene índice uno es el segundo en la solución, y así sucesivamente.

El editor posee una guía de creación de los distintos tipos de nivel, que se encuentra en el Apéndice B.

3.2.7 Telemetría

Para la recopilación de datos del videojuego durante las pruebas de usuario, es necesario integrar un sistema de telemetría dentro de Unity. El objetivo es registrar eventos a tener en cuenta para llevar a cabo un análisis sobre la eficacia del

Desarrollo de TeoriKart

videojuego, como los niveles jugados, los errores cometidos, el tiempo de juego de cada nivel, el orden en el que se seleccionan los vehículos en los niveles de la categoría prioridad y las luces que se encienden en los niveles de la categoría luces. Para ello se ha utilizado Simva, un conjunto de herramientas desarrolladas por el grupo de investigación de la Universidad Complutense de Madrid, e-UCM¹⁶.

Simva¹⁷ es una herramienta creada para simplificar la tarea de validación de juegos educativos, integrando herramientas para realizar encuestas y sistemas de analíticas de aprendizaje. Durante las pruebas de usuario se utiliza Simva para la recopilación de datos. Por un lado, los cuestionarios son encuestas enlazadas de la aplicación de software libre LimeSurvey¹⁸. Entre medias de los cuestionarios, se realiza la prueba del juego en Unity, durante la cual se envían trazas para recopilar datos sobre el comportamiento de los usuarios. Esta herramienta facilita la cohesión de los datos mediante la creación automática de identificadores aleatorios y totalmente anónimos, que se distribuyen de manera física a los participantes. Mediante estos identificadores se puede acceder tanto a las respuestas de los cuestionarios como a las trazas del juego, para su posterior análisis.

Group: Grupo_Autoescuela						
Owners		Participants				
		Edit Group Name				
Pic	Username	Is Token	Token	Email	Role	Add Participants
	67f52f0490d7c302b9aebce2_namv	true	namv	67f52f0490d7c302b9aebce2_namv@example.com	student	
	67f52f0490d7c302b9aebce2_yrhs	true	yrhs	67f52f0490d7c302b9aebce2_yrhs@example.com	student	
	67f52f0490d7c302b9aebce2_uwap	true	uwap	67f52f0490d7c302b9aebce2_uwap@example.com	student	
	67f52f0490d7c302b9aebce2_cwrr	true	cwrr	67f52f0490d7c302b9aebce2_cwrr@example.com	student	
	67f52f0490d7c302b9aebce2_ashs	true	ashs	67f52f0490d7c302b9aebce2_ashs@example.com	student	
	67f52f0490d7c302b9aebce2_vkcq	true	vkcq	67f52f0490d7c302b9aebce2_vkcq@example.com	student	
	67f52f0490d7c302b9aebce2_fxsa	true	fxsa	67f52f0490d7c302b9aebce2_fxsa@example.com	student	
	67f52f0490d7c302b9aebce2_kbuj	true	kbuj	67f52f0490d7c302b9aebce2_kbuj@example.com	student	
	67f52f0490d7c302b9aebce2_cmyf	true	cmyf	67f52f0490d7c302b9aebce2_cmyf@example.com	student	
	67f52f0490d7c302b9aebce2_kpmi	true	kpmi	67f52f0490d7c302b9aebce2_kpmi@example.com	student	
	67f52f0490d7c302b9aebce2_gmlt	true	gmlt	67f52f0490d7c302b9aebce2_gmlt@example.com	student	

Figura 3.49: Vista de la sección de identificadores en el entorno de Simva

Xasu es un asset de Unity creado también por el grupo e-UCM, que consiste en un tracker, un sistema que una vez integrado en un juego, puede enviar y/o almacenar

¹⁶ <https://www.e-ucm.es/es/>

¹⁷ <https://www.e-ucm.es/es/portfolio-item/simva/>

¹⁸ <https://www.limesurvey.org/es>

Desarrollo de TeoriKart

eventos significativos para su posterior análisis. Xasu se encarga de simplificar el uso de analíticas de aprendizaje xAPI¹⁹ .que es una especificación de aprendizaje virtual que permite recopilar datos sobre las experiencias que una persona tiene dentro de las actividades de capacitación en línea y fuera de línea.

Para utilizarlo en el videojuego, se tienen que añadir los packages de Unity correspondientes, tanto de Xasu como de Simva.



Figura 3.50: Vista de los assets de Simva y Xasu en la ventana Package Manager de Unity

Una vez añadidos, se crea un GameObject²⁰ empty XasuTracker en la escena de Menú, con el script Simva Plugin asociado.

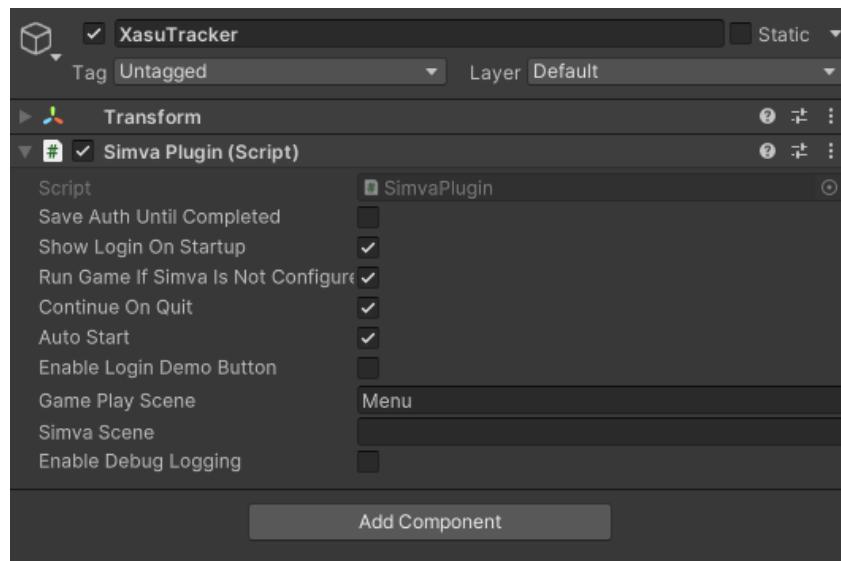


Figura 3.51: Vista del objeto XasuTracker en el inspector de Unity

Para el envío de las trazas xAPI, en el código se llaman a las instrucciones correspondientes con la información deseada que se quiere recopilar en cada evento.

¹⁹ <https://xapi.com/>

²⁰ <https://docs.unity3d.com/ScriptReference/GameObject.html>

Pruebas de usuario

```
CompletableTracker.Instance.Completed("nivel" + currentLevel.ToString()).WithSuccess(true);
```

Figura 3.52: Ejemplo de envío de traza de nivel completado correctamente

Para evaluar la efectividad del videojuego y la usabilidad del editor, es necesario realizar pruebas con el objeto de obtener datos para analizar el aprendizaje tras jugar al videojuego de alumnos de autoescuela preparándose el examen teórico. Y por otro, evaluar la usabilidad del editor con una profesora de autoescuela, observando si es capaz de crear niveles de manera intuitiva. Los resultados obtenidos se describen en el Capítulo 5.

4.1 Prueba del videojuego

Las pruebas se han llevado a cabo en la autoescuela Marca de Madrid el día 9 de abril del año 2025. Todos los participantes fueron citados el día anterior por la autoescuela durante una clase teórica.

Estas pruebas han consistido en la realización de dos cuestionarios, uno al inicio de las pruebas y otro al final, y en jugar a una serie de niveles del videojuego entre medias de ambos. Las pruebas se han organizado en 2 turnos: uno por la mañana y otro por la tarde. En la sesión matutina acudieron 8 personas, en 4 turnos de 2 personas, y por la tarde 9 personas en 3 turnos de 3 personas. El tiempo máximo de duración de la sesión se ha limitado a 35 minutos por usuario, incluyendo tanto los cuestionarios como la parte del videojuego. El hardware que se ha utilizado consiste en ordenadores portátiles con un sistema operativo Windows 11²¹. Estos ordenadores los proporcionamos nosotros, ya que los equipos de los que disponía la autoescuela eran muy antiguos. Todas las preguntas de los cuestionarios se muestran en el Apéndice A.

²¹ <https://www.microsoft.com/es-es/windows/windows-11?r=1>

Pruebas de usuario



Figura 4.1: Fotografías tomadas durante las pruebas de usuario

4.1.1 Cuestionario Previo

En este cuestionario se presentan 19 preguntas de tipo test de examen teórico, cada una de ellas con tres opciones. El objetivo de este primer cuestionario es evaluar los conocimientos de los usuarios antes de probar el videojuego. Estas preguntas se componen de 3 temas distintos del teórico: Prioridad de paso; Dispositivos de alumbrado; Señalización óptica de los vehículos y Señales de Indicación. La mayoría de las preguntas forman parte de la plataforma de test en línea Matferline²², empleada por la Autoescuela Marca. También se han obtenido preguntas de la página PracticaTest²³. Algunas de las preguntas elegidas corresponden con situaciones de los niveles del videojuego, para observar si se mejora entre los cuestionarios al jugarlos, las otras preguntas no se han añadido como niveles para observar el contraste.

4.1.2 Prueba del juego

Tras el cuestionario inicial, los usuarios pasarán a jugar al videojuego. La prueba consistirá en jugar una serie de niveles de las tres secciones disponibles: 4 niveles de prioridad, 4 de luces y 5 de manejo, siendo uno de estos últimos un tutorial.

La idea es recopilar datos de las partidas de los usuarios, como el tiempo que han tardado en completar cada nivel, si lo han completado de manera correcta o no, qué errores han tenido o las luces que han encendido. El objetivo de recoger esta

²² <https://matferline.com/>

²³ <https://practicatest.com/>

Pruebas de usuario

información es analizar el comportamiento de los usuarios, observar posibles errores de diseño del juego y evaluar la eficacia.

4.1.3 Cuestionario Posterior

En este cuestionario final se repiten las 19 preguntas del primer cuestionario, para evaluar si después de la sesión de juego, los participantes conseguían más aciertos. Por otro lado, se llevarán a cabo 10 preguntas de opinión sobre el videojuego, acerca de la satisfacción, dificultad y comprensión de los contenidos.

4.2 Prueba del editor

El editor de niveles se ha probado el día 12 de mayo de 2025 con una profesora de la Autoescuela Marca.

La prueba ha consistido en crear un nivel de cada categoría (manejo, luces, prioridad), teniendo acceso a la guía del editor que se muestra en el Apéndice B. La estructura de los niveles creados ha sido totalmente decidida por la profesora. Una vez creado cada nivel, se ha jugado para comprobar que el resultado es el deseado.

El hardware utilizado consiste en un ordenador portátil de las características de los utilizados en las pruebas del videojuego.

Una vez finalizado este proceso, ha llevado a cabo un cuestionario de opinión sobre el editor, cuyas preguntas se encuentran en el Apéndice C.

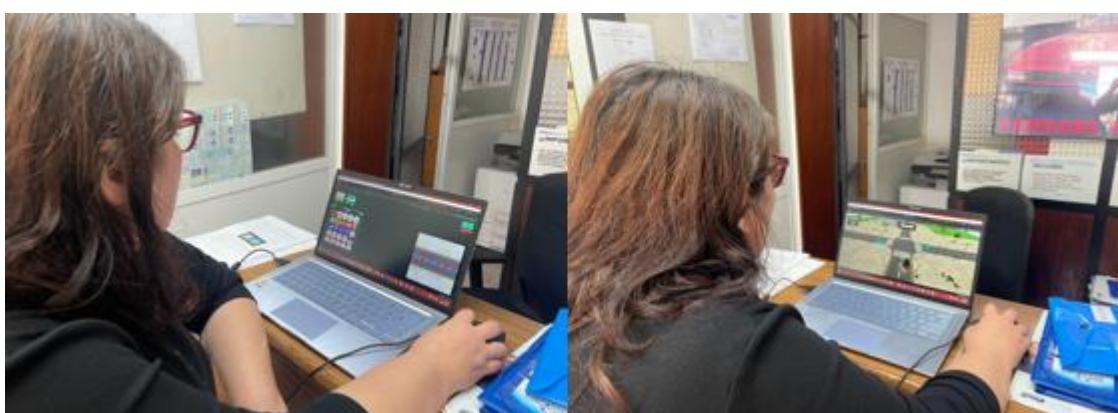


Figura 4.2: Fotografías tomadas durante las pruebas del editor

Capítulo 5. Resultados de las pruebas de usuario

En este capítulo se muestran los resultados obtenidos de ambas pruebas descritas en el Capítulo 4.

5.1 Resultados de la prueba del videojuego

Tras las pruebas realizadas se han analizado tanto los datos de interacción recogidos de la prueba con el videojuego, como las respuestas a los cuestionarios previo y posterior. Este análisis se ha realizado utilizando Jupyter Notebook²⁴ como entorno de desarrollo, empleando el lenguaje de programación Python²⁵ y la biblioteca especializada en análisis de datos Pandas²⁶.

5.1.1 Demografía de los usuarios

Las pruebas fueron realizadas por 17 usuarios, de los cuales 9 eran mujeres y 8 eran hombres, consiguiendo así una distribución equilibrada. Todos los usuarios eran alumnos de la Autoescuela Marca y se encontraban preparándose para presentarse al examen teórico de conducir del permiso B. La edad media de los usuarios que han participado es de 26.29 años (ver figura 5.2 con la distribución de las edades).

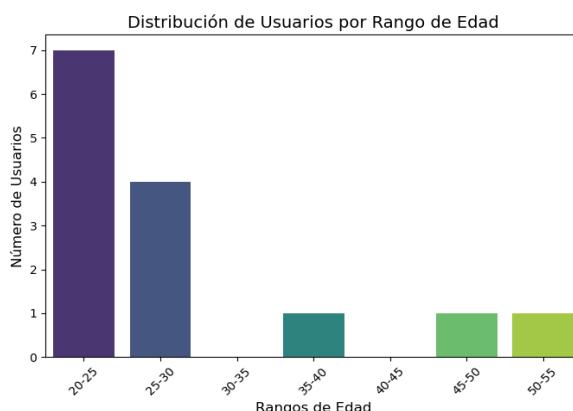


Figura 5.1: Distribución de edad de los usuarios

²⁴ <https://jupyter.org/>

²⁵ <https://www.python.org/>

²⁶ <https://pandas.pydata.org/>

Resultados de las pruebas de usuario

También se ha analizado cuánto de habituados están los usuarios a jugar a videojuegos, ya que cabe la posibilidad de que los menos acostumbrados experimenten mayores dificultades jugando.

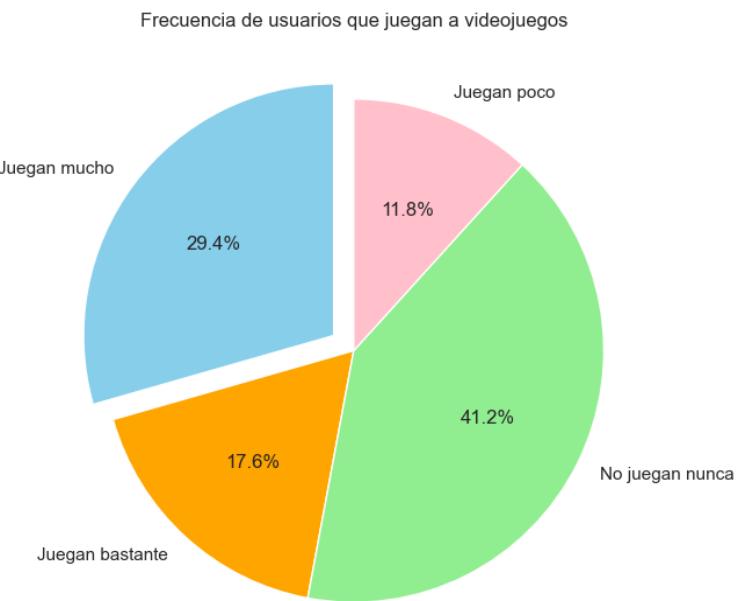


Figura 5.2: Frecuencia de tiempo dedicado a videojuegos de los usuarios

La gráfica de la Figura 5.2 muestra que existe bastante variedad en cuanto a la frecuencia con la que los usuarios de las pruebas juegan a videojuegos. Por una parte, 8 personas (47% del total) juegan con bastante o mucha frecuencia, 2 usuarios (11.8%) juegan poco y los otros 9 usuarios (53%) indican que no juegan nunca.

Resultados de las pruebas de usuario

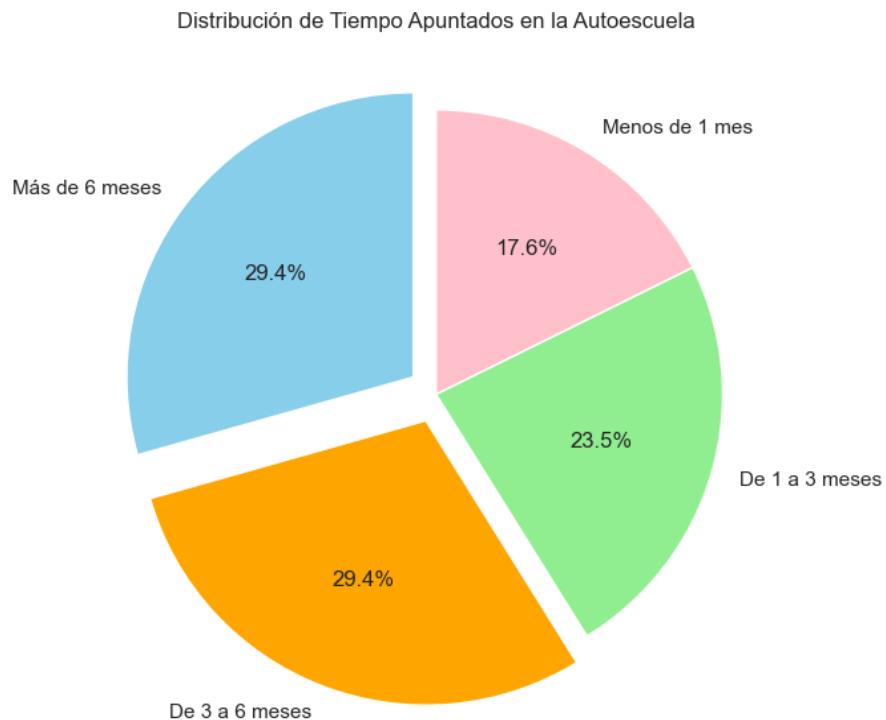


Figura 5.3: Distribución de tiempo de matriculación en la autoescuela de los usuarios

La gráfica en la Figura 5.3 muestra que existe bastante variedad en cuanto al tiempo que llevan los usuarios apuntados en la autoescuela. Un 58.8% del total lleva más de 3 meses matriculado, mientras que el resto de los usuarios tiene una antigüedad menor.

Resultados de las pruebas de usuario

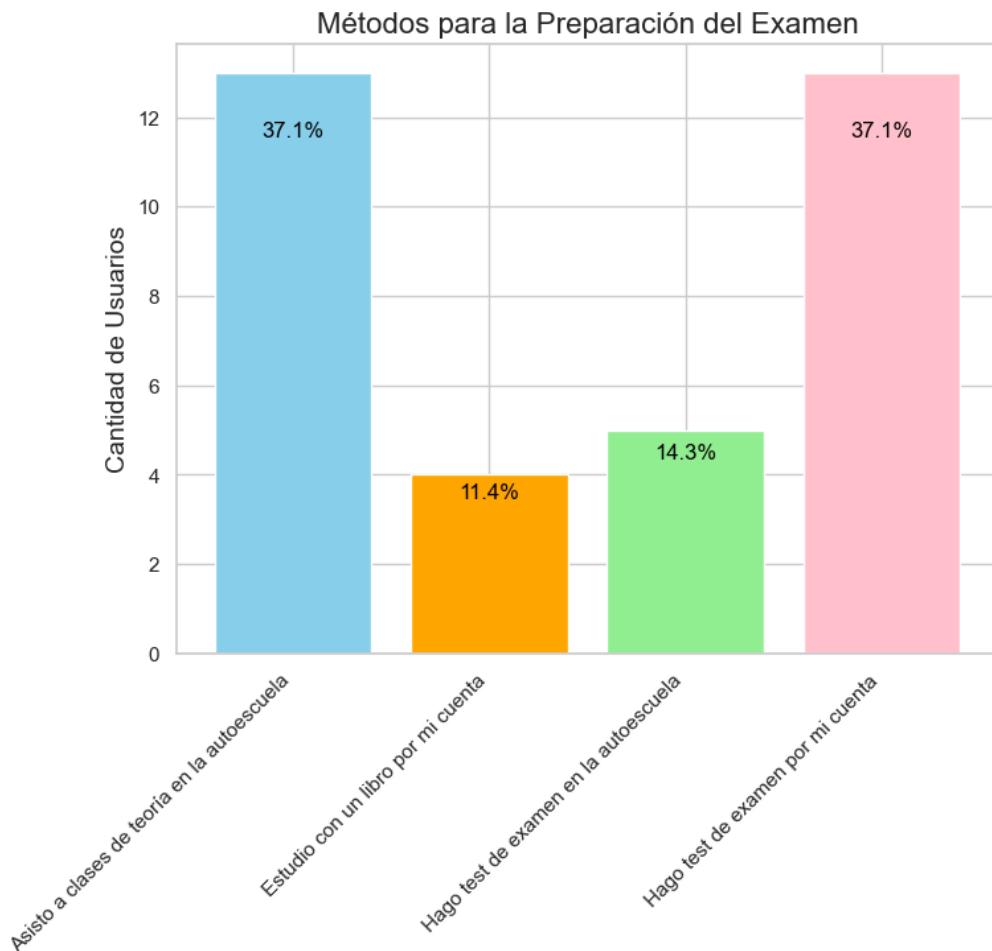


Figura 5.4: Distribución de métodos de preparación para el examen de los usuarios

Respecto al estudio del examen, se preguntó a los participantes sobre los métodos que empleaban, permitiendo seleccionar varias opciones. Como se puede apreciar en la Figura 5.4, la mayor parte de los usuarios se preparan asistiendo a clases teóricas de la autoescuela y haciendo test por su cuenta.

5.1.2 Análisis de las trazas de interacción con el videojuego

A continuación, se exponen varias tablas con datos sobre los niveles completados durante las pruebas.

Resultados de las pruebas de usuario

Nivel	Tiempo medio de cada intento (s)	Desviación típica del tiempo en cada intento (s)	Usuarios Únicos que Acertaron	Usuarios Únicos que Fallaron	Veces Fallado/ Usuarios Totales	Usuarios Totales que jugaron el nivel
Manejo1	71.97 s	20.73 s	11	8	0.69	13
Manejo2	57.96 s	9.98 s	8	8	0.91	11
Manejo3	82.33 s	29.29 s	9	3	0.36	11
Manejo4	87.67 s	22.81 s	7	2	0.22	9
Prioridad1	69.95 s	35.08 s	16	6	0.65	17
Prioridad2	33.90 s	10.31 s	16	4	0.25	16
Prioridad3	42.21 s	12.27 s	14	1	0.29	14
Prioridad4	36.59 s	17.32 s	13	5	0.38	13
Luces1	62.56 s	24.81 s	17	2	0.12	17
Luces2	44.87	18.94 s	12	9	0.8	15
Luces3	88.28 s	37.35 s	5	8	0.91	11
Luces4	53.84 s	23.51 s	9	9	1.17	12

Tabla 5.1: Estadísticas de los niveles del videojuego de todos los usuarios

En la Tabla 5.1 se recogen los datos de los niveles jugados por los 17 usuarios que han realizado las pruebas. Como se puede apreciar, los niveles con mayor número de fallos por usuario total son los niveles Luces3 y Luces4. El nivel Luces3 corresponde a una situación en la cual el jugador se encuentra conduciendo de noche en una vía interurbana insuficientemente iluminada. En primera instancia, se deben escoger las luces con las que debe circular un vehículo en dicha situación (Luces Largas y Luces

Resultados de las pruebas de usuario

de Posición), más tarde, otro vehículo se cruza con el jugador en la carretera, por lo que lo correcto es poner las Luces Cortas para no deslumbrar a este vehículo. La mayoría de los intentos incorrectos vienen porque una vez los usuarios se alejaron del vehículo, no volvieron a poner las Luces Largas para retomar la conducción y acabar el nivel.

En cuanto al nivel Luces4, muchos usuarios se han confundido en la selección de las luces correctas, debido a que en el nivel se hace inciso que se escojan aquellas luces que se pueden llevar encendidas en una situación de niebla intensa, y los usuarios encendían únicamente las obligatorias.

Por otro lado, respecto a los niveles de manejo, se puede apreciar que el nivel Manejo2 también hay un alto número de errores, con 0.91 veces fallado por usuario. La inmensa mayoría de estos errores provienen de saltarse un semáforo en rojo (2 en algunos casos), lo que es importante tener en cuenta ya que la clave de este nivel reside en la gestión de los semáforos y el tráfico, debido a la existencia de una cuadrícula de marcas amarilla.

En el nivel Manejo1, 5 de los intentos incorrectos son por chocar con otro vehículo, el cual aparece delante del jugador en el mapa. Cabe resaltar que en el nivel Manejo3 solamente se hizo mal una vez la detención en una señal de Stop que aparece al principio del nivel.

Los niveles de la categoría prioridad han sido en líneas generales los menos fallados por los usuarios y los que más usuarios únicos han podido probar.

En la siguiente tabla se expone la misma información que en la del apartado anterior, pero en lugar de estar reflejados los datos de todos los usuarios, aparecen datos de aquellos que más juegan a videojuegos, los 5 usuarios que indicaron que juegan con mucha frecuencia a videojuegos y los 3 que indicaron que juegan bastante, haciendo un total de 8 usuarios. El objetivo de esta tabla es analizar en qué aspectos influye tener una mayor experiencia jugando a videojuegos en esta prueba.

Resultados de las pruebas de usuario

Nivel	Tiempo medio de cada intento(s)	Desviación típica del tiempo en cada intento (s)	Usuarios Únicos que Acertaron	Usuarios Únicos que Fallaron	Veces Fallado/ Usuarios Totales	Usuarios Totales que jugaron el nivel
Manejo1	66.68 s	9.11 s	6	7	1.0	8
Manejo2	59.12 s	8.30 s	6	5	0.88	8
Manejo3	79.53 s	25.33 s	7	2	0.25	8
Manejo4	89.53 s	24.85 s	6	1	0.14	7
Prioridad1	59.27 s	28.56 s	7	2	0.63	8
Prioridad2	30.36 s	11.23 s	8	1	0.13	8
Prioridad3	37.68 s	8.76 s	8	1	0.5	8
Prioridad4	36.59 s	15.66 s	7	4	0.57	7
Luces1	55.04 s	15.27 s	8	0	0	8
Luces2	43.85 s	23.90 s	7	4	0.75	8
Luces3	88.89 s	38.60 s	4	4	0.86	7
Luces4	52.28 s	24.57 s	5	8	1.63	8

Tabla 5.2: Estadísticas de los niveles del videojuego de los usuarios que más juegan a videojuegos

Lo destacable de la Tabla 5.2 es que la mayoría de estos 8 usuarios han podido probar todos los niveles del videojuego, a excepción de 1 nivel de cada categoría, en los cuales 1 usuario no tuvo tiempo de probarlo, siendo este usuario diferente. Los tiempos medios de cada intento son bastante similares a los vistos en la tabla anterior, pero en general se aprecia una ligera disminución de la desviación típica, por lo que los usuarios completan los intentos en tiempos más similares. Respecto a

Resultados de las pruebas de usuario

los fallos en los niveles, solamente es notorio que en el nivel Luces4 se han equivocado todos estos usuarios y en Manejo1, 7 de ellos, suponiendo un gran porcentaje del total de usuarios que tuvieron intentos incorrectos en estos dos niveles.

A continuación, la siguiente tabla corresponde únicamente a los 7 usuarios que afirmaron que no juegan nunca a videojuegos.

Nivel	Tiempo medio de cada intento(s)	Desviación típica del tiempo en cada intento (s)	Usuarios Únicos que Acertaron	Usuarios Únicos que Fallaron	Veces Fallado/ Usuarios Totales	Usuarios Totales que jugaron el nivel
Manejo1	89.09 s	35.68 s	4	1	0.25	4
Manejo2	62.24 s	2.77 s	1	2	1.0	2
Manejo3	97.61 s	44.53 s	1	1	1.0	2
Manejo4	94.32 s	-	0	1	1.0	1
Prioridad1	94.54 s	36.95 s	7	2	0.43	7
Prioridad2	38.98 s	7.80 s	6	2	0.33	6
Prioridad3	47.85 s	12.52 s	5	0	0	5
Prioridad4	44.5 s	20.21 s	5	1	0.2	5
Luces1	77.03 s	29.24 s	7	1	0.14	7
Luces2	51.86 s	10.08 s	4	3	0.6	5
Luces3	103.70 s	44.73 s	0	3	1.0	3
Luces4	62.77 s	22.16 s	3	1	0.33	3

Tabla 5.3: Estadísticas de los niveles del videojuego de los usuarios que nunca juegan a videojuegos

Resultados de las pruebas de usuario

En la Tabla 5.3 se pueden apreciar dos aspectos considerables. El primero, el tiempo medio de cada intento, que en líneas generales es sustancialmente mayor que en las tablas anteriores. Debido a esto, surge el otro aspecto destacable, que es el número de usuarios que pudieron probar los niveles. Especialmente en 3 de los niveles de la categoría manejo, apenas 1-2 usuarios pudieron jugarlos debido a la falta de tiempo. Esta problemática también surge porque muchos de estos usuarios se demoraron mucho más tiempo del esperado realizando el cuestionario previo.

Hay 2 usuarios que destacan negativamente ya que han podido jugar muy pocos niveles. Estos dos usuarios son de 2 de los 3 que más edad tienen de aquellos que han completado las pruebas de usuario. El primero, de 38 años, probó únicamente 4 niveles. El otro usuario, de 45 años, jugó nada más que 3 niveles. Ninguno de estos 2 usuarios jugó ningún nivel de la categoría de manejo.

Movimiento	Media de diferencia entre pulsaciones Grupo Juega (s)	Media de diferencia entre pulsaciones Grupo No Juega (s)
Avanzar	6.05 s	11.66 s
Giro Izquierda	8.27 s	15.95 s
Giro Derecha	7.27 s	9.67 s

Tabla 5.4: Comparativa entre media de tiempo entre pulsaciones entre los usuarios que más juegan a videojuegos y los que no juegan nunca

En la Tabla 5.4 se expone una comparativa entre la diferencia media en segundos de las pulsaciones de las teclas de movimiento del coche del jugador en los niveles de manejo y luces. En una columna aparecen las medias del grupo conformado por los 8 usuarios que juegan mucho y bastante a videojuegos y en la otra aquellas relativas a los 7 jugadores que no juegan nunca. Se aprecia una diferencia significativa entre ambos grupos, los usuarios que no juegan nunca pulsan con menor frecuencia las teclas de movimiento, por lo que en lugar de pulsar las teclas poco a poco para moverse, realizan pulsaciones de mayor duración.

Resultados de las pruebas de usuario

Nivel	Luces Encendidas/ Usuarios que han jugado (Grupo juega)	Luces Encendidas/ Usuarios que han jugado (Grupo no juega)
Luces1	4.13	3.57
Luces2	5.25	4.4
Luces3	8.43	8.67
Luces4	10.63	8.33

Tabla 5.5: Comparativa entre medias de las luces encendidas por usuario en los niveles de luces

En la Tabla 5.5 se expone una comparativa de la media de las luces encendidas en cada nivel por usuario que lo ha jugado, tanto de los 8 usuarios que más juegan como de los 7 usuarios que no juegan nunca. No se aprecia una diferencia sustancial entre ambos grupos.

5.1.3 Comparativa entre cuestionarios

A continuación, se expone una tabla con una comparativa de los cuestionarios realizados, apareciendo el porcentaje de aciertos en cada una de las preguntas de ambos cuestionarios, así como la diferencia obtenida entre ellos.

Pregunta	Porcentaje de aciertos (Test Previo)	Porcentaje de aciertos (Test Posterior)	Diferencia
Pregunta 1	88.24%	88.24%	0
Pregunta 2	88.24%	82.35%	-5.89%
Pregunta 3	100%	100%	0%
Pregunta 4	88.24%	94.12%	+5.89%
Pregunta 5	100%	94.12%	-5.89%

Resultados de las pruebas de usuario

Pregunta 6	58.82%	64.71%	+5.89%
Pregunta 7	47.06%	76.47%	+29.41%
Pregunta 8	76.47%	76.47%	0%
Pregunta 9	94.12%	100%	+5.89%
Pregunta 10	47.06%	76.47%	+29.41%
Pregunta 11	82.35%	88.24%	+5.89%
Pregunta 12	76.47%	82.35%	+5.89%
Pregunta 13	76.47%	88.24%	+11.76%
Pregunta 14	82.35%	88.24%	+5.89%
Pregunta 15	64.71%	76.47%	+11.76%
Pregunta 16	52.94%	58.82%	+5.89%
Pregunta 17	70.59%	82.35%	+11.76%
Pregunta 18	100%	100%	0
Pregunta 19	47.06%	52.94%	+5.89%

Tabla 5.6: Comparativa entre aciertos en los cuestionarios de todos los usuarios

Las preguntas con mayor mejoría entre el cuestionario previo y el posterior son la Pregunta 7 y la Pregunta 10, con un aumento del 29.41% de los aciertos. Además, ambas presentaban la peor tasa de aciertos en el cuestionario previo, con un 47.06% cada una. Ambas preguntas aparecen representadas en niveles del videojuego. La Pregunta 10 se corresponde con el Nivel 3 de la categoría luces del videojuego, el cual, como se mencionó en el apartado anterior, tuvo la peor tasa Aciertos/Fallos de todos los niveles del juego (50%). Se aprecia una mejora en la puntuación final del test después de jugar al juego.

Resultados de las pruebas de usuario

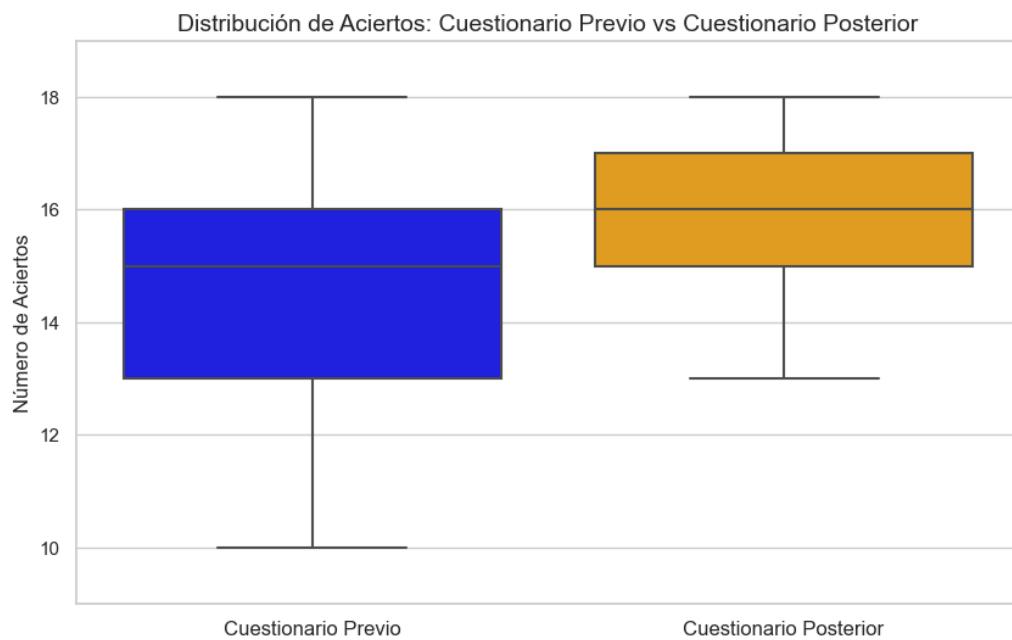


Figura 5.5: Distribución de aciertos entre cuestionario previo y posterior

En el cuestionario previo, la media de aciertos total es de 14.41 aciertos, mientras que, en el posterior, la media es de 15.71 aciertos por lo que se aprecia una mejora significativa. En cuanto a la variabilidad de los resultados, como se puede apreciar en la Figura 5.5, los resultados en el cuestionario posterior son menos dispersos, pasando de estar los aciertos contenidos entre 10 y 18 en el cuestionario previo, a estar entre 13 y 18 en el cuestionario posterior. Se ha producido un descenso de la desviación estándar de 2.45 (previo) a 1.57 (posterior).

También es interesante analizar los resultados divididos en 4 grupos, según el tiempo que llevan los usuarios matriculados en la autoescuela, como se explicó anteriormente en la sección de demografía. Todos estos grupos presentan una mejora similar en la media de aciertos y en la disminución de la desviación estándar, lo que es una buena señal ya que significa que han aprendido todos los usuarios independientemente de su tiempo de matriculación.

Los 4 usuarios que llevan 1-3 meses matriculados y los 3 usuarios que llevan menos de 1 mes coinciden en la media de aciertos del cuestionario final con 15.01 aciertos. Los 5 usuarios entre 3-6 meses matriculados cuentan con 15.39 aciertos y, los 5 restantes que llevan más de 6 meses, una media de 16.91 aciertos. Se aprecia una evolución progresiva en los aciertos según el tiempo de matriculación en la autoescuela.

Resultados de las pruebas de usuario

5.1.4 Opinión del videojuego

Este cuestionario posee una serie de afirmaciones para expresar la opinión sobre el videojuego, que se pueden evaluar con valores entre 0 y 5 (ambos incluidos). Las respuestas tienen las siguientes medias:

Afirmación	Media (sobre 5)
El juego me pareció útil para aprender los conceptos teóricos del carné de conducir	4.95
Los controles del coche con teclado fueron fáciles de manejar	4.29
En los niveles de prioridad, las instrucciones sobre cómo determinar la prioridad entre los coches fueron claras	4.88
En los niveles de manejo, el destino al que hay que llegar es fácil de visualizar	4.76
En los niveles de luces, las instrucciones sobre cómo encender y apagar las distintas luces fueron claras	4.76
La retroalimentación que se otorga en los niveles es adecuada	4.76

Tabla 5.7: Media de las valoraciones del videojuego

Además de estas preguntas de opinión, se formula a los usuarios la siguiente cuestión:

“¿Hay algún tema/sección del examen teórico que te cueste más comprender? ¿Cuál?”

7 usuarios (41.18% del total) indican específicamente que el tema que más les cuesta comprender es el relativo a las Luces.

3 usuarios (17.65% del total) afirman que lo que más les cuesta comprender son las Velocidades según el tipo de Vehículo y Carreteras. 2 de estos usuarios también mencionan que les resulta difícil el tema de Distancias de Seguridad.

Resultados de las pruebas de usuario

Otros temas mencionados son las ITVs y las revisiones y los pesos de camiones y sus medidas.

Para finalizar el cuestionario, se les ha pedido a los usuarios que contesten con texto las siguientes preguntas:

“¿Qué te gustó más del juego y qué cambiarías?” y “¿Qué aspectos mejorarias del juego para hacerlo más útil o divertido?”

Todas las respuestas mencionan que les gusta mucho el juego y que les parece bastante útil y divertido. La mayoría de los usuarios con menos experiencia en videojuegos indican que les ha resultado complicada la conducción del coche en los niveles que lo requerían. Un usuario sin experiencia en videojuegos comentó durante las pruebas que le resultaría más sencillo si el juego fuese para dispositivos móviles.

En cuanto a otras mejoras recomendadas, la mayoría afirma que les gustaría tener más secciones de temas distintos, otros elementos (como peatones) y que se redujese la cantidad de texto en los niveles.

5.2 Resultados de la prueba del editor

Durante la prueba del editor con la profesora de la autoescuela, pudo crear satisfactoriamente un nivel de cada categoría y probarlo sin problemas. Afirma que la interfaz del editor es bastante intuitiva, especialmente porque dispone de todas las configuraciones al mismo tiempo en pantalla, sin necesidad de ir creando los niveles paso a paso. Por otra parte, señala que echa en falta los siguientes elementos:

- Añadir luces de emergencia y que el jugador pueda encender los intermitentes
- Una nueva pieza de plaza de circulación giratoria (raqueta)
- Nuevas señales de peligro que marcan prioridad

Además, indica que una posible mejora es permitir arrastrar una pieza ya colocada en el mapa a otra baldosa. Por último, afirma que este editor permite crear niveles útiles para complementar la enseñanza de los temas de prioridades, señalización y alumbrado a sus alumnos.

Una vez finalizado el análisis de las pruebas, se puede concluir que el videojuego ha gustado entre los usuarios y les ha parecido útil y divertido. Los usuarios afirman que lo utilizarían para estudiar y se ha visto una mejora sustancial entre cuestionarios previo y posterior, por lo que se puede afirmar que el videojuego ha sido eficaz en las pruebas.

Por otro lado, el manejo del vehículo del jugador ha generado problemas en los usuarios menos experimentados en los videojuegos, por lo que habría que realizar cambios en los controles en un futuro. Es interesante reevaluar la forma en la que se otorga la retroalimentación al comenzar un nivel, ya que a muchos usuarios les parece que existe una cantidad demasiado grande de texto en las explicaciones.

A pesar de los resultados es necesario recalcar que las pruebas deberían realizarse con un mayor número de usuarios y debería de medirse el recuerdo, es decir, si tras varias semanas siguen acordándose de los contenidos que han aprendido. También es importante probar el editor de niveles con un mayor número de profesores de autoescuela, para poder evaluarlo con más opiniones.

En cuanto a nuevos cambios a incluir, se encuentra la incorporación de nuevos temas del manual del teórico en el videojuego, además de aumentar la cantidad de niveles por categoría y añadir nuevas señales de tráfico. También es interesante agregar peatones, para aportar una mayor diversidad a estos niveles. Asimismo, el editor debe de aumentar en consecuencia con el videojuego, permitiendo agregar los nuevos elementos a los niveles.

Además, el videojuego puede adaptarse a una versión para las plataformas de Android e iOS, para hacerlo más accesible a aquellos jugadores que no disponen de un ordenador personal o que no están muy habituados a jugar a videojuegos en ellos.

Motivation

In recent years, it has become increasingly common to find students in Spanish driving schools to prepare for the type B driving licence theory test solely by taking practice tests, often without reading the manual or attending classes taught by the driving schools themselves. This is concerning, as it promotes superficial learning based on memorizing answers, limiting understanding of traffic regulations, which can cause problems later on, especially when taking the practical driving test.

This trend is especially noticeable among younger students, due to a clear lack of motivation, as traditional methods are slow and repetitive, and they also fail to fully leverage the modern technologies they are accustomed to. Many people are visual learners or struggle to picture typical traffic situations accurately, especially because they have no prior driving experience .

The motivation behind this work is to find new, more engaging and enjoyable ways to study for the theoretical driving test. Video games have been seen as a tool to practice traffic regulations and driving itself, but they have mostly been focused on free-form exercises and preparation for the practical driving test. However, the potential of video games extends beyond these areas; they could also be used to address the preparation of the theoretical driving exam.

Goals

The objective of this project is to develop a video game that serves as a complementary tool for preparing for the type B driving licence theory test in Spain. To achieve this, the following sub-objectives are addressed:

- Understand the field. It is necessary to review the curriculum content of the theoretical driving test to identify what is intended to be taught, as well as to study the most common difficulties students face during exam preparation. In addition, existing video games and applications related to this field must be analyzed.
- Develop a video game that addresses the topic. The proposal is to develop a video game that addresses the syllabus of the theoretical driving test.

Introduction

- Study the effectiveness of the video game. This requires evaluating the effectiveness of the video game through user testing with students from a driving school. The results of these tests will be analyzed, and feedback from users will be collected on possible improvements to the video game.

Work plan

During the course of this work, meetings will be held every 2 weeks, both in person and on the Google Meet platform, the development of the video game will be carried out in a public GitHub repository. The following section outlines the work plan designed to achieve the objectives described above:

- Research. During this stage, both the content of the theoretical exam and the various video games/tools related to the field of driving licenses will be researched. Key aspects, such as the educational design and gameplay mechanics of existing applications, will be analyzed.
- Video Game Design. An initial design of the video game will be carried out, with sketches of what the different levels could be like and the mechanics they may contain. To do this, it will be necessary to gather information on the topics that are generally most difficult for driving school students to understand and analyze which ones would be best suited for a video game.
- Video Game Development. The development will follow the initial design. This design may be adapted as progress is made. A level editor will also be developed to streamline the level creation process.
- User Testing. User tests of the video game will be conducted at a driving school, with the users being enrolled students preparing for the theoretical driving test. The objective of these tests is to evaluate the effectiveness of the video game by measuring the knowledge acquired through questionnaires. It is also necessary to analyze whether it is intuitive and engaging for users.
- Data analysis. After user testing, the collected data will be analyzed to assess whether the game is effective in preparing users for the theoretical driving test. In addition, user acceptance, level difficulty and design will be studied based on the data obtained.

Once the test analysis is complete, it can be concluded that users liked the video game and found it useful and fun. Users stated they would use it for studying, and a substantial improvement was seen between pre- and post-tests, so it can be stated that the video game has been effective in testing.

On the other hand, the handling of the player's vehicle has caused problems for less experienced video game users, so changes to the controls would be necessary in the future. It is interesting to reevaluate the way feedback is given at the beginning of a level, as many users feel there is too much text in the explanations.

Despite the results, it is necessary to emphasize that the tests should be conducted with a larger number of users and that recall should be measured, that is, whether they still remember the content they have learned after several weeks. It's also important to test the editor with a larger number of driving instructors so we can evaluate it with more feedback.

Regarding new changes to be included, there is the incorporation of new topics from the theory manual into the video game, in addition to increasing the number of levels per category and adding new traffic signs. It is also interesting to add pedestrians to provide greater diversity to these levels. Also, the editor must grow accordingly with the game, allowing new elements to be added to the levels.

Additionally, the game can be adapted for Android and iOS platforms, making it more accessible to players who don't have a personal computer or aren't very familiar with playing video games on them.

Adrián Montero Castrillo

Por mi parte, una vez definimos la idea de este trabajo, acudí a la autoescuela

Contribuciones Personales

Marca a comentarles la idea y preguntar acerca de la posibilidad de llevar a cabo unas pruebas de usuario del videojuego con sus alumnos. Además, pregunté a los profesores acerca de los temas que más dificultades causaban al estudiarse, qué preguntas se fallaban más en los exámenes y qué elementos podrían ser adecuados para el videojuego.

Por otra parte, investigué artículos con información oficial sobre el examen teórico, como por ejemplo las estadísticas de aprobados y suspensos o las preguntas más falladas en los últimos años. En cuanto a los juegos serios de conducción que hemos investigado para conocer el estado de la cuestión, me he encargado de hacer un análisis de Virtual Simulator for the Taking and Evaluation of Psychometric Tests to Obtain a Driver's License y de The Virtual Reality Serious Game for Learning Driving Skills before Taking Practical Test, ambos hechos en Realidad Virtual. Además, también me he encargado de analizar el simulador de conducción DriveSim.

Respecto al videojuego, inicialmente abordé la programación de los coches no manejables por el usuario, pero finalmente cuando se desechó la idea de usar NavMesh, mi compañera se encargó de programarlos de otra manera distinta. También ayudé a mi compañera con el sistema de GPS que utilizaban estos coches y el jugador, pero que también se descartó. En cuanto al funcionamiento actual de estos vehículos, me he encargado de gestionar la conexión de las piezas del mapa para la correcta circulación de los coches. También he añadido la posibilidad de que estos vehículos puedan ser ambulancias o autobuses, para aportar más dinamismo en los niveles de prioridad.

Me he encargado de la creación de los niveles mediante archivo JSON, inicialmente por coordenadas de Unity, y posteriormente mediante conversiones con divisiones matriciales en las piezas del mapa. Me he encargado de la lógica de los niveles de prioridad. Respecto a los elementos

Contribuciones Personales

de tráfico, he programado la lógica de la señal de Stop, y me he encargado de que los coches no manejables respeten tanto las señales como los semáforos.

Respecto a los niveles del juego, he diseñado y creado todos los de la categoría prioridad y algunos de la categoría manejo. Respecto al menú de selección de nivel, he implementado el movimiento del coche, el feedback visual y sonoro de las señales que representan los niveles y los textos explicativos de cada nivel.

Respecto al editor de niveles, desarrollé una primera versión que consistía en la creación de una cuadrícula con dimensiones especificables, que representa el mapa y la mecánica de arrastrar y soltar las piezas sobre ella. También programé los InteractivePoints de las piezas para colocar los elementos, además de configurarlos para guardarlos correctamente. Me encargué también del feedback que se aporta al colocar el ratón sobre las piezas y elementos. Finalmente hice que los mapas se guarden en archivo JSON siguiendo la estructura deseada y se creasen desde Unity leyendolos. Mi compañera siguió mejorando y desarrollando más funciones del editor.

En cuanto a la recopilación de datos, me he encargado de integrar el sistema de telemetría dentro de Unity y de enviar aquellos eventos de los que queríamos recabar información. También hice una selección de las preguntas que se iban a incluir en los cuestionarios de las pruebas de usuario. Durante las pruebas, estuve presente junto a mi compañera observando el comportamiento de los participantes y anotando información relevante.

Para finalizar, me he encargado del análisis de los datos recabados durante las pruebas, tanto de ambos cuestionarios como de los eventos que se han recopilado del videojuego. He utilizado Jupyter Notebook para este análisis, generando varias tablas y gráficas de comparativa de datos de los usuarios, tanto en conjunto como separándolos en varios grupos según el tiempo que llevan matriculados en la autoescuela, sus métodos de preparación para el examen y su experiencia jugando a videojuegos. He participado junto a mi compañera en la redacción de esta memoria.

Contribuciones Personales

Simona Antonova Mihaylova

Durante mi participación en este Trabajo de Fin de Grado, he contribuido en múltiples tareas tanto de diseño como de desarrollo.

Como primer paso, investigué junto a mi compañero los juegos serios existentes relacionados con el tema elegido: la conducción y la preparación del examen práctico para obtener el carné de conducir. Realicé el análisis del juego Car Driving School Simulator, lo cual me dió una visión más profunda sobre la manera de abordar la estructura y objetivos de nuestra idea. También me encargué de la elaboración de los bocetos y esquemas iniciales de la interfaz de usuario y el diseño del juego para reflejarlos visualmente.

Durante el desarrollo del sistema de coches dirigidos por inteligencia artificial, investigué la mejor manera de abordarlo. Inicialmente se consideró el uso de NavMesh, usando la inteligencia artificial de Unity, pero tras hacer varias pruebas finalmente optamos por descartarlo por la incompatibilidad con los mapas creados por archivo. Como alternativa desarrollé el sistema de Waypoints para la navegación de los coches, que permitiese una mayor flexibilidad a la hora de la navegación por distintas piezas. Simultáneamente desarrollé una herramienta para editor de Unity para añadir los waypoints de forma más sencilla. En relación a esto, me ocupé de la abstracción de las piezas y sus conexiones, lo que permitía recrear el movimiento de los vehículos de forma más realista y eficiente.

Me encargué de crear los elementos del mapa como todas las piezas, incluyendo carreteras, glorietas, intersecciones, y otras estructuras urbanas. Esto implicó ajustar las escalas de los modelos, colocar los waypoints necesarios para la IA de los coches y la selección de assets adecuados para el entorno del juego, incluyendo los vehículos, carreteras, edificios, señales de tráfico, semáforos, túneles y elementos para la interfaz de usuario en 2D. También me encargué de diseñar algunos de estos recursos desde cero.

Contribuciones Personales

Me encargué del diseño e implementación de las interfaces de usuario, asegurando la cohesión visual entre escenas y la correcta escalabilidad de los elementos en función la resolución de pantalla.

Junto a mi compañero, diseñamos inicialmente un sistema de GPS, sin embargo al comprobar que distraía al usuario opté por desarrollar un minimapa que guiara al jugador de forma más sutil y menos intrusiva. Participé también en el diseño y desarrollo del menú principal, el menú de selección de niveles y el editor de niveles. Utilizando este editor generé múltiples mapas para cada nivel del juego: todos los de luces y varios de manejo. Además, amplié las funcionalidades del editor para permitir la creación de niveles de cada tipo: prioridad, luces y manejo, más allá de la creación de mapas. En concreto los paneles de las luces, en los que se pueden elegir las condiciones del mapa y la solución. La adaptación correspondiente a los niveles de prioridad la realicé en colaboración con mi compañero.

Los scripts iniciales para el movimiento del vehículo los implementé yo, sin embargo, se decidió cambiarlos. Integré el sistema de control de movimiento, buscando un asset que emulase de forma realista el comportamiento de un vehículo real, con el objetivo de aumentar la inmersión en el juego.

Para los niveles de luces, implementé los botones de la interfaz que permiten encender las luces, y programé la lógica de los niveles de deslumbramiento y niebla. En los niveles de prioridad me ocupé de las cámaras en perspectiva de cada vehículo y de su animación, y las luces intermitentes que indican el giro de los vehículos. En cuanto a los niveles de manejo, implementé la lógica de funcionamiento de los semáforos, la marca vial cuadrícula amarilla, y la señal de velocidad máxima.

Por último, mi compañero y yo realizamos las pruebas con usuarios en la autoescuela Marca, donde observamos las sesiones de juego y tomamos notas sobre la experiencia de los usuarios, lo que nos permitió determinar posibles mejoras. He escrito la guía de uso del editor de niveles. Ambos nos hemos encargado de la corrección de errores y de la redacción de esta memoria.

Belén Gómez (2014) 'Los Mejores Juegos Serios de 2014 premiados en Bilbao', *On Serious Games* [Preprint]. Available at: <https://www.onseriousgames.com/mejores-juegos-serios-2014/> (Accessed: 14 May 2025).

COMUNICACIONPONS (2024) 'La confusión de las señales, entre las diez preguntas más falladas por los estudiantes del permiso B en 2024'. Available at: <https://ponsseguridadvial.com/la-confusion-de-las-señales-entre-las-diez-preguntas-mas-falladas-por-los-estudiantes-del-permiso-b-en-2024/> (Accessed: 16 May 2025).

Dirección General de Tráfico (no date) 'Requisitos, preparación y presentación a examen'. Available at: <https://www.dgt.es/nuestros-servicios/permisos-de-conducir/obtener-un-nuevo-permiso-de-conducir/requisitos-preparacion-y-presentacion-a-examen/> (Accessed: 16 May 2025).

Laamarti, F., Eid, M. and El Saddik, A. (2014) 'An Overview of Serious Games', *International Journal of Computer Games Technology*, 2014, pp. 1–15. Available at: <https://doi.org/10.1155/2014/358152>.

Li, W.-Y., Lu, T.-C. and Wei, H.-X. (2023) 'Serious gaming: user acceptance of the simulated motorbike license training system', *Multimedia Tools and Applications*, 83(17), pp. 52763–52794. Available at: <https://doi.org/10.1007/s11042-023-17532-x>.

Likitweerawong, K. and Palee, P. (2018) 'The virtual reality serious game for learning driving skills before taking practical test', in *2018 International Conference on Digital Arts, Media and Technology (ICDAMT)*. IEEE, pp. 158–161. Available at: <https://doi.org/10.1109/ICDAMT.2018.8376515>.

Morayma Estrada (2024) 'Los juegos serios: una combinación de educación y entretenimiento.', *Red Educa* [Preprint]. Available at: <https://www.rededuca.net/blog/actualidad-educativa/juegos-serios> (Accessed: 14 May 2025).

Sánchez, J.S. et al. (2019) 'Virtual Simulator for the Taking and Evaluation of Psychometric Tests to Obtain a Driver's License', in, pp. 138–149. Available at: https://doi.org/10.1007/978-3-030-25965-5_11.

APÉNDICES

Apéndice A - Preguntas de los

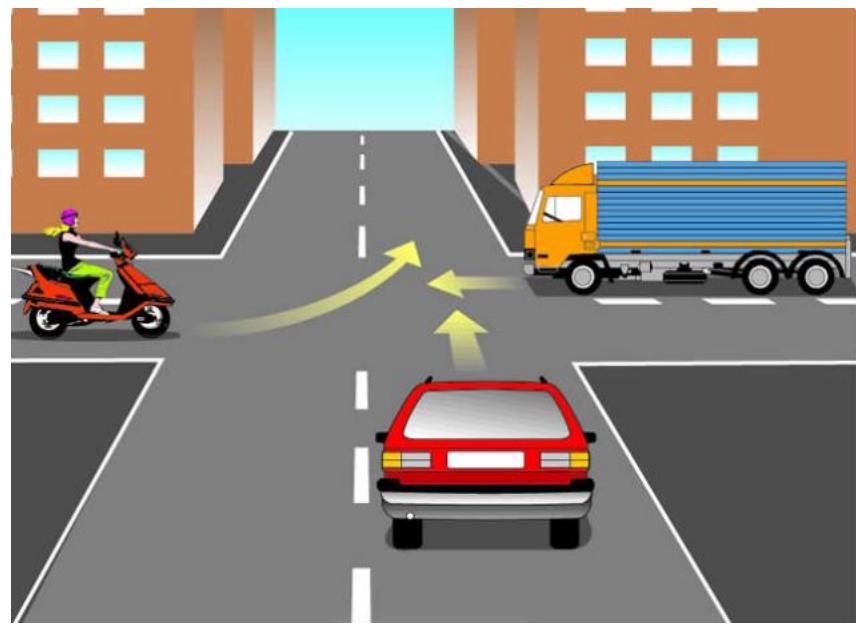
En este apéndice se exponen las preguntas que contestaron los participantes
cuestionarios de las pruebas con usuarios
de las pruebas de usuario en los cuestionarios.

Preguntas demográficas

1. Edad
2. Género
3. ¿Hace cuánto que te has apuntado a la autoescuela para sacarte el carné de conducir?
 - Menos de 1 mes
 - De 1 a 3 meses
 - De 3 a 6 meses
 - Más de 6 meses
4. ¿Cómo te preparas para el examen teórico? (puedes seleccionar varias opciones)
 - Asisto a clases de teoría en la autoescuela
 - Estudio con un libro por mi cuenta
 - Hago test de examen en la autoescuela
 - Hago test de examen por mi cuenta

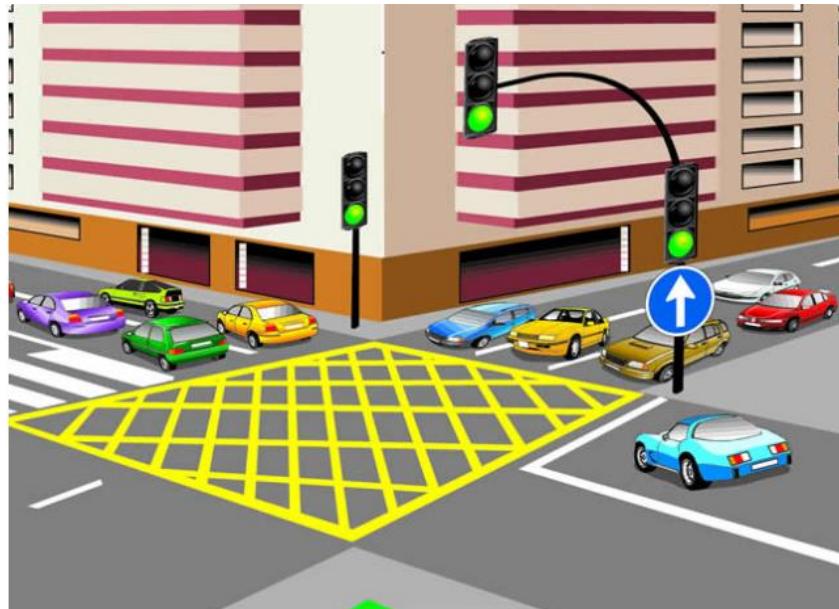
Preguntas tipo test

1. En esta intersección, ¿Quién debe pasar primero?



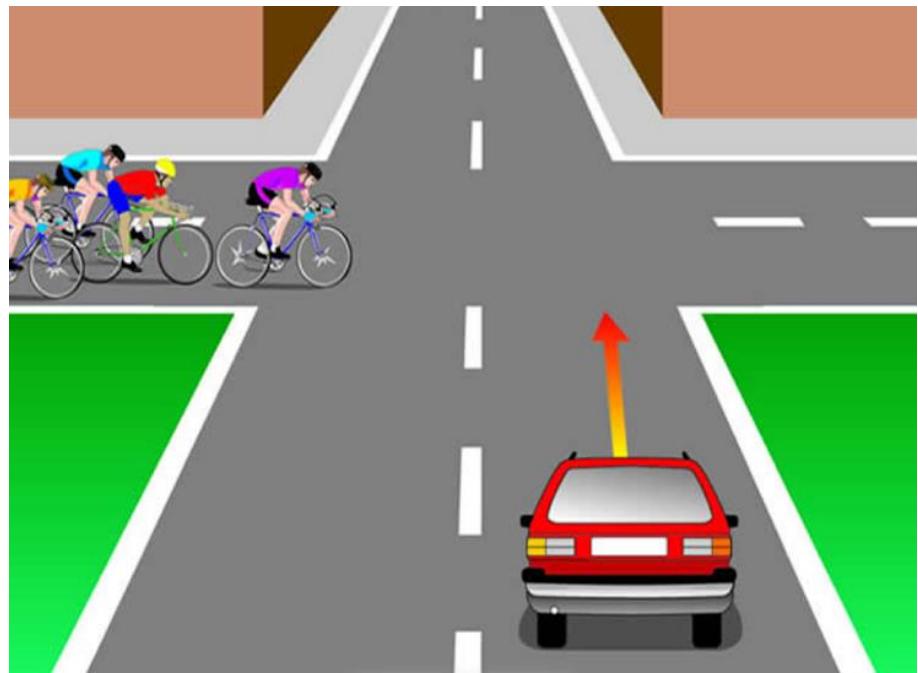
- Camión
- Turismo
- Ciclomotor

2. En este cruce, ¿puede pasar cuando se encienda la luz verde?



- Sí, pero debo esperar hasta poder pasar sin bloquearlo
- Sí, siempre, porque tengo preferencia
- Sí, pero debo ceder el paso a los vehículos de la derecha
- No

3. En esta intersección, ¿debe ceder el paso a los ciclistas?



- Sí, pero solo al primer ciclista
 - No, porque los ciclistas no tienen la prioridad
 - Sí, a todo el grupo de ciclistas
4. ¿Debe encender la luz de corto alcance en un túnel lo suficientemente iluminado?



- No, porque está iluminado
 - Sí, es obligatorio
 - No, debe encender la luz de largo alcance
5. Hay mucha niebla, ¿qué luces deben llevar encendidas los vehículos?

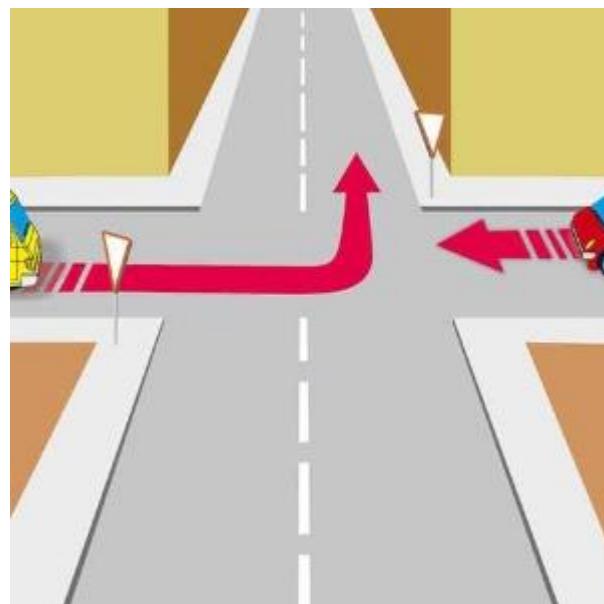


- Las luces de antiniebla junto con las luces cortas
 - Las luces de antiniebla junto con las luces de emergencia
 - Las luces de posición solamente
6. ¿Puede circular por una vía interurbana con el alumbrado de largo alcance en caso de lluvia intensa o niebla densa?



- Sí, porque en este caso la vía está insuficientemente iluminada

- No, está prohibido
 - Sí, pero no es recomendable porque el reflejo en la niebla o la lluvia le puede deslumbrar
7. En esta intersección, en la que los dos vehículos tienen de frente una señal de ceda el paso, ¿quién pasará primero?



- El conductor del autobús porque se aplica la regla general de prioridad de paso
 - El conductor del autobús porque tiene mayor dificultad de maniobra
 - El conductor del vehículo rojo porque tiene preferencia de paso
8. ¿Qué indica la señal?



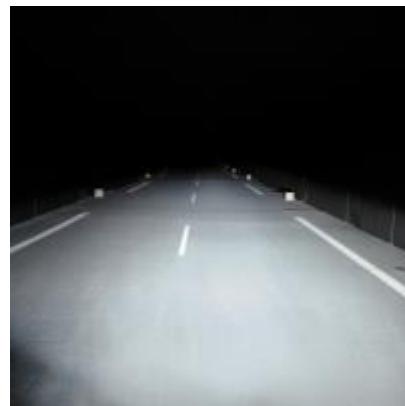
- Aconseja no circular a más de 80 km/h
- Obliga a circular a 80 km/h
- Obliga a dejar una distancia de 80 metros con el vehículo que circula delante

9. A partir de esta señal, está obligado a llevar encendido al menos, el alumbrado de...



- Corto alcance
- Posición
- Largo alcance

10. En esta vía es obligatorio utilizar la luz de corto alcance...



- Siempre
- Sólo cuando no se disponga de luz larga
- Cuando exista la posibilidad de deslumbrar a otros usuarios

11. ¿Qué indica la luz de posición de un vehículo?



- Que el vehículo está parado
- Que el vehículo está circulando
- El lugar que ocupa en la calzada y la anchura del vehículo

12. ¿Está permitido circular en poblado con la luz de largo alcance?



- No
- Sí, cuando la vía esté insuficientemente iluminada
- Sí, cuando circule a más de 40 km/h

13. A una ambulancia que usa únicamente la señal especial luminosa, ¿se debe ceder el paso?



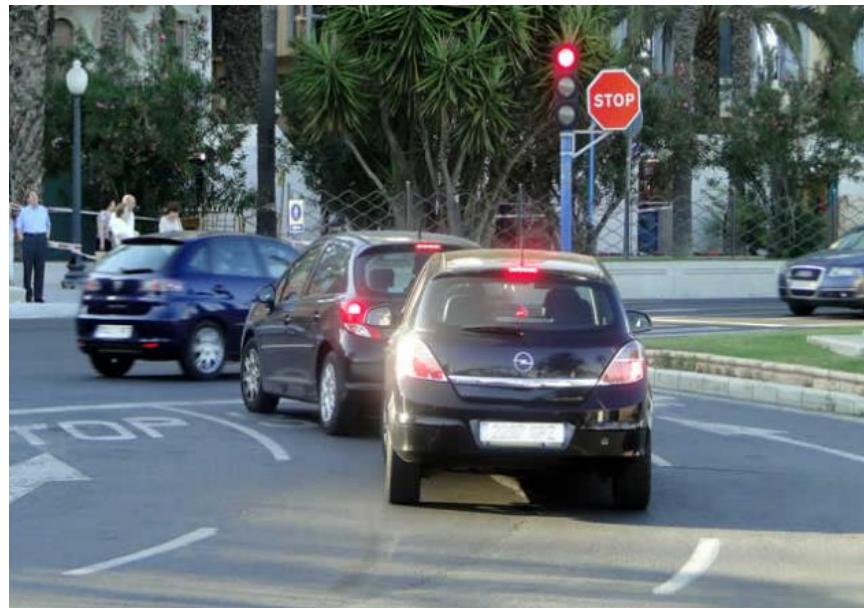
- No, ya que siempre debe emplearla conjuntamente con la acústica
- Sí, pero solo cuando se trate de vía urbana
- Sí

14. La señal indica que...



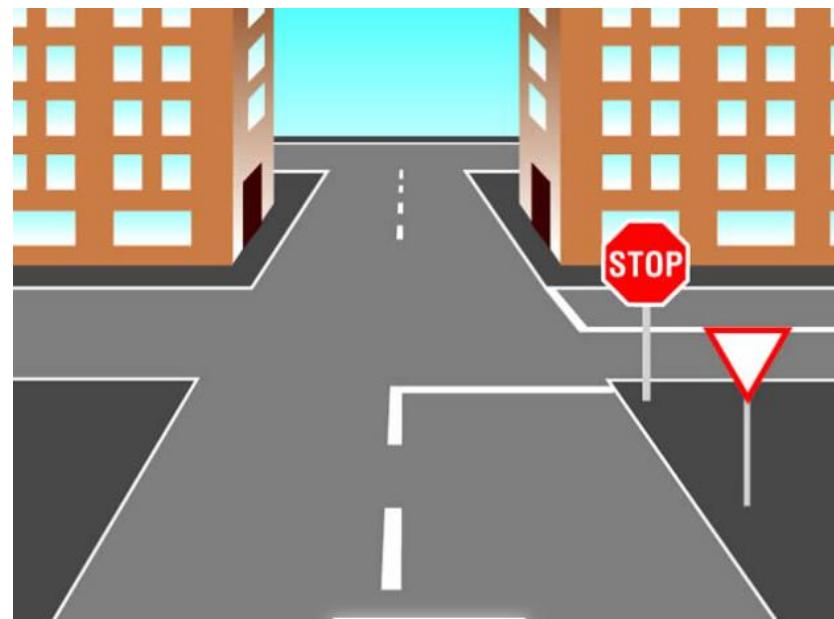
- Está prohibido el cambio de sentido
- No puede desplazarse lateralmente ni a la derecha ni a la izquierda
- Debe circular en el sentido indicado por la flecha, ya que es una calzada de sentido único

15. Cuando el semáforo se ponga en verde, ¿qué debe hacer?



- Obedecer el semáforo, pero tiene que ceder el paso
- Obedecer la señal de Stop
- Obedecer el semáforo, ya que tiene prioridad sobre la señal de Stop

16. En esta situación, ¿qué debe hacer?



- Obedecer la señal de Stop
- Ceder el paso y si es necesario detenerme
- Ceder el paso sin sobrepasar la línea de detención

17. ¿Qué indica la flecha verde del semáforo?



- Que puedo girar a la derecha con prioridad
- Que puedo girar a la derecha cuando se encienda la luz verde del semáforo principal

- Que puedo tomar la dirección que indica la flecha, con precaución y dejando pasar a los vehículos que circulen por el carril al que me incorporo

18. Entre los distintos tipos de señales, ¿cuál es el orden de prioridad?



- Agentes, semáforos y señales verticales
- Semáforos, agentes y señales verticales
- Señales verticales, agentes y semáforos

19. En esta situación, si pretende seguir de frente, ¿a cuál de los dos semáforos debe de obedecer?



- Izquierda

- Derecha
- Como la señalización es contradictoria, prevalece la más restrictiva

Preguntas de opinión

1. "¿Cómo de acuerdo estás con las siguientes afirmaciones?

Siendo: 1 - Muy en desacuerdo 2 - En desacuerdo 3 - No se 4 - De acuerdo 5 - Muy de acuerdo

- A. El juego me pareció útil para aprender los conceptos teóricos del carnet de conducir.
 - B. Los controles del coche con teclado fueron fáciles de manejar.
 - C. En los niveles de prioridad, las instrucciones sobre cómo determinar la prioridad entre los coches fueron claras.
 - D. En los niveles de manejo, el destino al que hay que llegar es fácil de visualizar.
 - E. En los niveles de luces, las instrucciones sobre cómo encender y apagar las distintas luces fueron claras.
 - F. La retroalimentación que se otorga en los niveles es adecuada.
2. ¿Con qué frecuencia juegas a videojuegos?
 3. ¿Qué te gustó más del juego y qué cambiarías?
 4. ¿Qué aspectos mejorarías del juego para hacerlo más útil o divertido?
 5. ¿Hay algún tema/sección del examen teórico que te cueste más comprender?

Apéndice B - Guía de uso del editor de niveles

A continuación se explica en detalle cómo utilizar el editor, dividido en secciones clave.

Interfaz general del editor

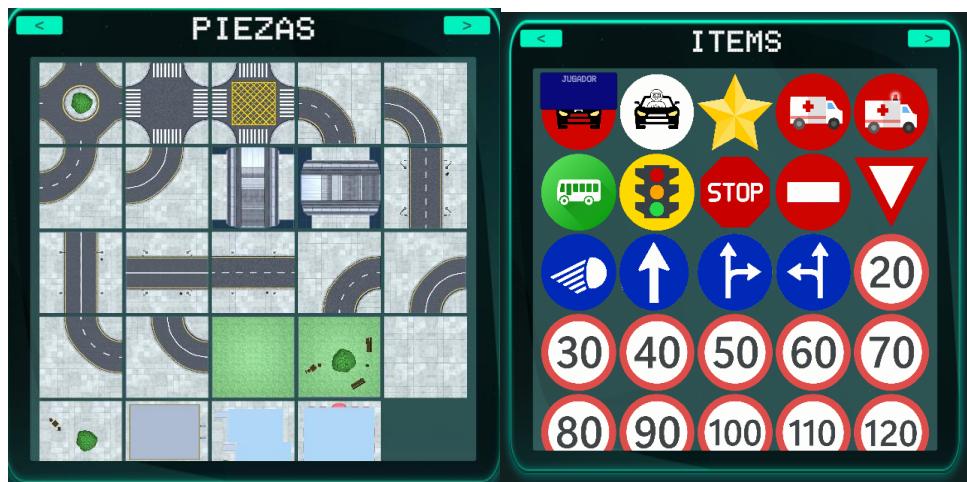
Al abrir el editor de niveles, encontrarás los siguientes elementos principales:

- Panel de configuración.



Incluye:

- Dimensiones del mapa. Deben ser NxN ya que los mapas son cuadrados.
- Botón Generar. Crea la cuadrícula vacía con las dimensiones especificadas.
- Desplegable de tipo de nivel. Permite seleccionar entre los tres tipos de niveles disponibles:
 - Manejo
 - Luces
 - Prioridad
- Campo de nombre del nivel. Este será el nombre con el que se guardará.
- Botón de guardar: Guarda el nivel en una lista con todos los demás niveles. Se encuentran en StreamingAssets/Editor.
- Panel Piezas e Items.



Piezas contiene elementos estructurales como carreteras, rotondas, intersecciones y pavimentos. Con las flechas laterales se puede navegar a Items, que contiene otros elementos como vehículos, señales de tráfico, el jugador y el destino (estrella). Estos ítems deben colocarse en los pequeños círculos disponibles sobre las piezas del mapa, arrastrándolos con el ratón y soltándolos.

- Botones de funcionalidad. Están en la esquina superior derecha.



- Menú. Regresa al menú principal, donde se pueden jugar los niveles prediseñados del juego.
- Niveles. Abre un panel con la lista de todos los niveles creados y guardados por el usuario.
- Reiniciar. Restablece el editor a su estado inicial.

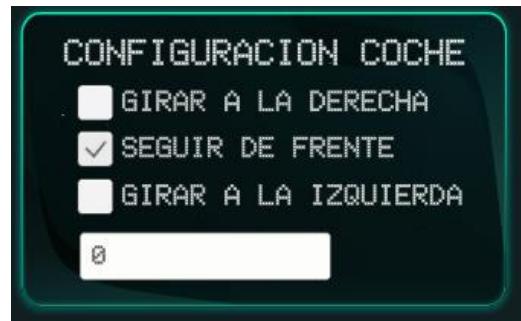
Tipos de nivel

- Manejo
 - Se puede diseñar el mapa libremente con las piezas disponibles.
 - Es obligatorio colocar:

- Un único jugador.
- Un objetivo (target) representado por una estrella.
- Estos elementos deben colocarse en los círculos de colocación sobre las piezas.
- El objetivo del jugador será llegar del punto inicial al destino superando los obstáculos o retos que el diseñador haya dispuesto.
- Luces. Funciona como el nivel de Manejo, pero con configuraciones adicionales específicas:



- Se puede elegir entre varios efectos ambientales:
 - Lluvia intensa
 - Niebla intensa
 - Noche
 - Deslumbramiento
- Se debe definir una solución luminosa, es decir, qué luces deben estar activadas correctamente para superar el nivel:
 - Cortas
 - Largas
 - Posición
 - Antinieblas delanteras
 - Antinieblas traseras
- Se puede seleccionar varias de estas simultáneamente.
- El jugador deberá activar las luces correctas en función de la situación planteada.
- Prioridad. Al seleccionar este tipo de nivel, el editor genera automáticamente un mapa predeterminado de dimensión 9x9. Este nivel está centrado en las reglas de prioridad en intersecciones.



El diseñador debe:

- Hacer clic en cualquiera de las cuatro piezas contiguas a la intersección central.
- Añadir vehículos. Se deben de añadir un mínimo de 2 vehículos, ya sean ambulancias, coches IA o autobuses. No se debe añadir al jugador. Por cada vehículo añadido, se abre un menú para configurar su comportamiento:
 - Girar a la derecha
 - Seguir recto
 - Girar a la izquierda

Esta configuración se puede cambiar en cualquier momento haciendo clic sobre el vehículo. Finalmente, hay un índice de solución: un área de texto donde se introduce la solución correcta para cada coche, usando los números 1, 2, 3, 4 que representan el orden en el que deben ser pulsados. Esta solución representa la respuesta correcta que el jugador deberá deducir o seleccionar al jugar. Por ejemplo, si dispones de dos coches en tu mapa, el coche que tiene la prioridad debe tener el índice 1, y el otro el índice 2.

Apéndice C - Preguntas del cuestionario

En este apéndice se muestran las preguntas del cuestionario realizado por la profesora de la Autopascola Marco tras probar el editor.

1. ¿Qué tan intuitiva te parece la interfaz del editor de niveles?

Siendo: 1 - Nada intuitiva 5 - Muy intuitiva

2. ¿Te ha resultado fácil e intuitiva la mecánica de arrastrar y soltar piezas en la cuadrícula?

- Sí
- No

3. ¿Echas en falta señales u otro tipo de elementos? ¿Cuáles?

4. ¿Crees que este tipo de niveles es una forma efectiva de enseñar el tema de señalización?

- Sí
- No

5. ¿Te resultó fácil definir tanto la solución luminosa correcta para el nivel como los efectos ambientales? ¿Harías algún cambio?

6. ¿Echas en falta algún elemento para la configuración de este tipo de nivel? ¿Cuál?

7. ¿Crees que este tipo de niveles es una forma efectiva de enseñar el tema de alumbrado?

8. ¿El mapa generado automáticamente de 9x9 te pareció adecuado para representar situaciones de prioridad en intersecciones?

- Sí
- No

9. ¿Has tenido dificultades al añadir y configurar el comportamiento de los vehículos?

10. ¿Te parece intuitiva la manera de definir la solución correcta mediante índices? ¿Qué cambiarías?

11. ¿Crees que este tipo de niveles es una forma efectiva de enseñar el tema de prioridades?
12. ¿Has sabido distinguir al vehículo del jugador con los vehículos no controlables?
- Sí
 - No, me he confundido alguna vez
13. Respecto a los niveles que has creado, ¿has podido probar todos correctamente?
- Sí
 - No, el nivel de manejo no ha funcionado bien
 - No, el nivel de luces no ha funcionado bien
 - No, el nivel de prioridad no ha funcionado bien
 - No ha funcionado bien ninguno
14. ¿Qué aspectos mejorarías para que el editor sea más intuitivo y fácil de usar para un profesor de autoescuela?