



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO EN SISTEMAS E INFORMÁTICA**

**TEMA: DESARROLLO DE UN SISTEMA DE SIMULACIÓN 3D SOBRE
EDUCACIÓN VIAL UTILIZANDO TÉCNICAS DE INTELIGENCIA
ARTIFICIAL. CASO DE ESTUDIO: NIÑOS DE LA ESCUELA “VÍCTOR
MANUEL PEÑAHERRERA”**

AUTOR: PROAÑO VALLES, CARLOS ANDRÉS

DIRECTOR: VILLACÍS SILVA, CÉSAR JAVIER

SANGOLQUÍ

2019



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

CERTIFICACIÓN

Certifico que el trabajo de titulación “**DESARROLLO DE UN SISTEMA DE SIMULACIÓN 3D SOBRE EDUCACIÓN VIAL UTILIZANDO TÉCNICAS DE INTELIGENCIA ARTIFICIAL. CASO DE ESTUDIO: NIÑOS DE LA ESCUELA VÍCTOR MANUEL PEÑAHERRERA**” fue realizado por el señor **PROAÑO VALLES, CARLOS ANDRÉS** el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 11 de junio de 2019

Una firma manuscrita en tinta azul, que parece ser "César Villacís Silva", escrita sobre una línea horizontal.

Ing. Villacís Silva, César Javier MSc.

CC. 1704892726



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORÍA DE RESPONSABILIDAD

Yo, Proaño Valles, Carlos Andrés, con cédula de identidad No. 1722372560, declaro que el contenido, ideas y criterios del trabajo de titulación **“DESARROLLO DE UN SISTEMA DE SIMULACIÓN 3D SOBRE EDUCACIÓN VIAL UTILIZANDO TÉCNICAS DE INTELIGENCIA ARTIFICIAL. CASO DE ESTUDIO: NIÑOS DE LA ESCUELA VÍCTOR MANUEL PEÑAHERRERA”** ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respaldado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Sangolquí, 11 de junio de 2019

Carlos Andrés Proaño Valles

CC. 1722372560



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

AUTORIZACIÓN

Yo, Proaño Valles, Carlos Andrés, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **“DESARROLLO DE UN SISTEMA DE SIMULACIÓN 3D SOBRE EDUCACIÓN VIAL UTILIZANDO TÉCNICAS DE INTELIGENCIA ARTIFICIAL. CASO DE ESTUDIO: NIÑOS DE LA ESCUELA VÍCTOR MANUEL PEÑAHERRERA”** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 11 de junio de 2019

Carlos Andrés Proaño Valles

CC. 1722372560

DEDICATORIA

A mis padres Víctor y Cecilia que siempre estuvieron apoyándome en todas las decisiones que he tomado. Además de siempre haberme brindado su cariño y amor para llevarme en la dirección correcta.

A mi hermana Carolina por siempre enseñarme con el ejemplo para llegar a cumplir mis metas. Incluso por enseñarme las actitudes que se tiene que tener en la vida y con la comunidad.

A mi familia en general por estar siempre para mí en los momentos en los que se me complicaba seguir luchando, y darme fuerzas para superarlos.

Carlos Proaño

AGRADEMIMIENTO

Agradezco a Dios por permitirme llegar a cumplir con mi objetivo y estar siempre presente en todas las actividades que he realizado.

Agradezco a mis padres y a mi hermana por brindarme todo lo necesario para cumplir con las actividades que he necesitado realizar y siempre estar apoyándome para que cumpla con las actividades que debía realizar.

Agradezco a mis compañeros por siempre ser un apoyo para cumplir con los trabajos y actividades de la universidad, y también brindarme su apoyo en las tareas incomprendidas en las clases.

Agradezco a mi tutor el ingeniero César Villacís por brindarme su tiempo y su apoyo para culminar con éxito la elaboración de mi trabajo de titulación.

Agradezco a mis profesores por brindarme nuevos conocimientos de las actividades que me gustan, y enseñarme lo que se debe hacer en la vida profesional.

Agradezco a la universidad por permitirme el ingreso para poder mostrar mi capacidad en las actividades realizadas.

Carlos Proaño

ÍNDICE

CERTIFICACIÓN	i
AUTORÍA DE RESPONSABILIDAD	ii
AUTORIZACIÓN	iii
DEDICATORIA	iv
AGRADEMIMIENTO	v
ÍNDICE	vi
ÍNDICE DE TABLAS	ix
ÍNDICE DE FIGURAS	x
RESUMEN	xiii
ABSTRACT	xiv
CAPÍTULO I	1
INTRODUCCIÓN	1
1.1 Antecedentes	1
1.2 Planteamiento del Problema.....	2
1.3 Objetivos	3
1.3.1 Objetivo General.....	3
1.3.2 Objetivos Específicos.....	3
1.4 Justificación	3
1.5 Alcance.....	4
CAPÍTULO II	5
MARCO TEÓRICO	5
2.1 Seguridad Vial.....	5
2.1.1 Los Involucrados en la vía.....	5
2.1.2 Las Señales de Tránsito.....	6
2.1.3 La dinámica de los vehículos.....	9
2.2 Cognición situada y aprendizaje en contextos escolares	9
2.2.1 El aprendizaje situado y las nuevas tecnologías de información y comunicación	11
2.3 Herramientas para el desarrollo del sistema.....	12
2.3.1 Máquinas de Estados.....	12
2.3.2 Sistema de Navegación de Unity: NavMesh.....	13
2.3.3 Seguimiento de una trayectoria.....	13
2.3.4 Controladores de Personajes (<i>Characters Controllers</i>).....	14

2.3.5	Colisionador de rueda (WheelCollider)	15
2.3.6	Canvas	15
2.3.7	Line Renderer.....	16
2.3.8	Scripts.....	17
2.4	La Inteligencia Artificial en los Sistemas de Simulación.....	18
2.5	Hardware.....	20
2.5.1	Volante Gigasus.....	20
2.6	Software	20
2.6.1	Unity.....	20
2.6.2	MakeHuman	22
2.6.3	Audacity.....	23
2.5.4	Zotero	24
2.5.4	Text2Speech	26
2.7	Metodología de Desarrollo del Sistema de Simulación	27
CAPÍTULO III		31
ANÁLISIS Y DISEÑO		31
3.1	Requerimientos Funcionales y No Funcionales	31
3.1.1	Requerimientos Funcionales	31
3.1.2	Requerimientos No Funcionales	35
3.2	Diagrama de Casos de Uso.....	36
3.3	Diagramas de Secuencia	37
3.3.1	Ingresar a la Aplicación	37
3.3.2	Seleccionar roles	38
3.3.3	Seleccionar la ruta.....	39
3.3.4	Seleccionar el Personaje.....	40
3.3.7	Realizar la actividad	40
3.3.8	Ingresar Nombre del Estudiante.....	44
3.4	Diagramas de Clases y Componentes.....	45
3.4.1	Diagramas de Clases.....	45
3.4.2	Diagrama de Componentes	51
DESARROLLO E IMPLEMENTACIÓN.....		52
4.1	El entorno del sistema de simulación.....	52
4.1.1	La ciudad.....	52
4.1.2	Coches automáticos.....	56

4.1.3	Semáforos.....	64
4.1.4	Peatones.....	65
4.2	Desarrollo de la interfaz gráfica del rol de peatón.....	71
4.3	Desarrollo de la interfaz gráfica del rol de conductor.....	73
4.4	Desarrollo de la interfaz gráfica del rol de espectador.....	76
4.5	Pruebas de caja blanca y caja negra.....	77
4.5.1	Pruebas de caja blanca.....	77
4.5.2	Pruebas de caja negra.....	82
CAPÍTULO V.....		91
RESULTADOS.....		91
5.1	Validación y Pruebas.....	91
5.2	Resultados.....	92
CAPÍTULO VI.....		102
CONCLUSIONES Y RECOMENDACIONES.....		102
REFERENCIAS BIBLIOGRÁFICAS.....		106

ÍNDICE DE TABLAS

Tabla 1 <i>Señales Informativas.</i>	7
Tabla 2 <i>Señales Preventivas.</i>	8
Tabla 3 <i>Señales Regulatorias.</i>	8
Tabla 4 <i>Propiedades de los WheelColliders.</i>	15
Tabla 5 <i>Plantilla de Documento Huddle.</i>	28
Tabla 6 <i>Descripción requerimiento funcional 1: Ingreso a la aplicación.</i>	31
Tabla 7 <i>Descripción requerimiento funcional 2: Rol conductor.</i>	32
Tabla 8 <i>Descripción requerimiento funcional 3: Rol peatón.</i>	32
Tabla 9 <i>Descripción requerimiento funcional 4: Rol espectador.</i>	32
Tabla 10 <i>Descripción requerimiento funcional 5: Seleccionar la ruta.</i>	33
Tabla 11 <i>Descripción requerimiento funcional 6: Seleccionar el personaje.</i>	33
Tabla 12 <i>Descripción requerimiento funcional 7: Realizar la actividad.</i>	34
Tabla 13 <i>Descripción requerimiento funcional 8: Guardar puntaje.</i>	34
Tabla 14 <i>Audios Señales de tránsito.</i>	75
Tabla 15 <i>Preguntas de usabilidad escuela.</i>	91
Tabla 16 <i>Información de la Aplicación.</i>	98
Tabla 17 <i>Aspectos Pedagógicos y Funcionales de la Aplicación.</i>	99
Tabla 18 <i>Aspectos Técnicos y Estéticos de la Aplicación.</i>	100
Tabla 19 <i>Recursos Didácticos que Utiliza la Aplicación.</i>	100
Tabla 20 <i>Esfuerzo Cognitivo que Exige la Aplicación.</i>	100

ÍNDICE DE FIGURAS

Figura 1. Componentes del Line Renderer.....	17
Figura 2. Ejemplo de Inicio de Script.	18
Figura 3. Volante Gigasus.....	20
Figura 4. Logo de Unity.....	21
Figura 5. Pantalla de Unity.....	22
Figura 6. Logo de MakeHuman.	22
Figura 7. Pantalla Principal de MakeHuman.....	23
Figura 8. Logo de AudaCity.....	23
Figura 9. Pantalla de Audacity.....	24
Figura 10. Logo de Zotero.....	24
Figura 11. Ventana de Zotero.....	26
Figura 12. Logo de Text2Speech.	26
Figura 13. Entorno de ingreso del texto a convertirse en voz.	26
Figura 14. Características de la aplicación Text2Speech.	27
Figura 15. Diagrama de Casos de Uso.	37
Figura 16. Diagrama de Secuencia: Ingresar a la aplicación.....	38
Figura 17. Diagrama de Secuencia: Seleccionar el modo de aplicación conductor.....	38
Figura 18. Diagrama de Secuencia: Seleccionar el modo de aplicación peatón.	39
Figura 19. Diagrama de Secuencia: Seleccionar el modo de aplicación espectador.....	39
Figura 20. Diagrama de Secuencia: Seleccionar la ruta.....	40
Figura 21. Diagrama de Secuencia: Seleccionar el personaje.....	40
Figura 22. Diagrama de Secuencia: Rol Conductor-Alcanza la meta.....	41
Figura 23. Diagrama de Secuencia: Rol Conductor-Infracción-Semáforo Rojo.....	42
Figura 24. Diagrama de Secuencia: Rol Conductor-Infracción-Contravía.....	42
Figura 25. Diagrama de Secuencia: Rol Conductor-Infracción-Exceso de Velocidad.....	43
Figura 26. Diagrama de Secuencia: Realizar la actividad: Rol Peatón.....	43
Figura 27. Diagrama de Secuencia: Realizar la actividad: Rol espectador.....	44
Figura 28. Diagrama de Secuencia: Ingresar Nombre del Estudiante.....	44
Figura 29. Diagrama de Clases del Rol de Peatón.....	45
Figura 30. Diagrama de Clases del Rol de Conductor.....	46
Figura 31. Diagrama de Clases CarAi.....	47
Figura 32. Diagrama de Clases de Autos Automáticos.....	48
Figura 33. Diagrama de Clases de Peatones Automáticos.....	49
Figura 34. Diagrama de Clases Semáforos.....	49
Figura 35. Diagrama de clases del Rol de Espectador.....	50
Figura 36. Diagrama de Clases para Guardar el Puntaje.....	50
Figura 37. Diagramas de clases de Navegación.....	51
Figura 38. Diagrama de Componentes de la Aplicación.....	51
Figura 39. Ciudad Despintada.....	52
Figura 40. Extracción Materiales Ciudad.....	53
Figura 41. Materiales Ciudad Pintados.....	53
Figura 42. Ciudad Pintada.....	53
Figura 43. Texturas edificios originales.....	54
Figura 44. Texturas edificios cambiadas.....	54
Figura 45. Texturas Techo Edificios.....	55

Figura 46. Ciudad con Paths.	55
Figura 47. Nodos y Paths.	56
Figura 48. Autos en Ciudad.....	56
Figura 49. Modelos de Autos.	57
Figura 50. Modelos de Autos Pintados.	57
Figura 51. Características Autos.	57
Figura 52. Componentes de los WheelColliders.	58
Figura 53. Componentes del body.....	58
Figura 54. Wheels Moving.....	59
Figura 55. Características CarAi.	59
Figura 56. Máquina de Estados.	60
Figura 57. Semáforo.....	61
Figura 58. Señal Pare	62
Figura 59. Límite de Velocidad.....	63
Figura 60. Modos de funcionamiento del semáforo.....	64
Figura 61. Semáforos Peatonales.	65
Figura 62. Tipos de hombres.....	65
Figura 63. Tipos de Mujeres.	66
Figura 64. Niño y Niña.....	66
Figura 65. Modelo básico humano.	66
Figura 66. Características físicas.....	67
Figura 67. Ropa.....	67
Figura 68. Tipos de Ropa.	68
Figura 69. Poses.	69
Figura 70. Navmesh en Ciudad.....	69
Figura 71. Personas en path.....	70
Figura 72. Personajes en Unity.	71
Figura 73. ThirdPersonController.	72
Figura 74. Ruta peatón	72
Figura 75. Señales Peatones.	73
Figura 76. Ruta Bomberos-Escuela.....	73
Figura 77. Ruta Estadio-Aeropuerto.	74
Figura 78. Ruta Clínica - Parqueadero Chaquiñán.....	74
Figura 79. Inputs Volante.....	75
Figura 80. Rol de Espectador.	76
Figura 81. Código de control de Manejo en Curvas.....	78
Figura 82. Código para evitar colisión entre coches.	80
Figura 83. Código de detección de contravía.	81
Figura 84. Pantalla Principal.	82
Figura 85. Modo de Juego.....	83
Figura 86. Peatón.	83
Figura 87. Personaje Seleccionado.....	84
Figura 88. Instrucciones Juego-Peatón.....	84
Figura 89. Simulación peatón-Niño.	85
Figura 90. Simulación peatón-Niña.	85
Figura 91. Pantalla Conductor.....	85
Figura 92. Ruta seleccionada.	86
Figura 93. Instrucciones de Juego-Conductor.....	86
Figura 94. Simulación ruta bomberos-escuela.	87

Figura 95. Simulación ruta estadio-aeropuerto.	87
Figura 96. Simulación ruta clínica - parqueadero chaquiñán.	88
Figura 97. Pantalla Game Over.	88
Figura 98. Pantalla Ganador.	89
Figura 99. Tabla de Puntajes.	89
Figura 100. Instrucciones de Juego-Espectador.	90
Figura 101. Pantalla modo espectador.	90
Figura 102. Pantalla modo espectador – Auto Seleccionado.	90
Figura 103. Características Computadora Escuela.	92
Figura 104. Pregunta 1.	92
Figura 105. Pregunta 2.	93
Figura 106. Pregunta 3.	93
Figura 107. Pregunta 4.	94
Figura 108. Pregunta 5.	95
Figura 109. Pregunta 6.	95
Figura 110. Pregunta 7.	96
Figura 111. Pregunta 8.	96
Figura 112. Pregunta 9.	97
Figura 113. Pregunta 10.	97
Figura 114. Resultados Encuesta Profesores.	101

RESUMEN

Este proyecto de titulación consiste en el desarrollo de un sistema de simulación en 3D que permite el aprendizaje de las señales de tránsito y que ha sido probado con niños de la Escuela Víctor Manuel Peñaherrera. La aplicación utiliza un modelo de ciudad que incluye muchos detalles como edificios, calles, avenidas, iglesias, hoteles, hospitales, estación de bomberos, estaciones de gasolina, paradas de bus, automóviles. Sobre esta base se construye la animación de un sistema de tránsito que incluye dos actores principales: coches y peatones. Esos objetos se constituyen en los PNJ (Personajes no jugadores) del sistema de simulación y están dotados de inteligencia artificial para circular por la ciudad. Los coches automáticos respetan los semáforos, frenan si un coche se detiene enfrente, respetan la señal de pare y actúan adecuadamente en las curvas y en los ascensos y descensos presentes en la ciudad simulada. La aplicación permite que el jugador tome el rol de peatón o conductor de un coche y realizar una misión básica que consiste en dirigirse de un lugar a otro de la ciudad. Para hacer atractiva la aplicación se implementó un sistema de puntaje que penaliza tres tipos de infracciones que comete el conductor: no respetar el semáforo, exceder el límite de velocidad, circular en contravía. En el rol de peatón la aplicación permite que el usuario se desplace a través de la ciudad y en puntos específicos recibe información audible del comportamiento correcto de un peatón al circular por la ciudad.

PALABRAS CLAVE

- **SISTEMA DE SIMULACIÓN**
- **COCHES AUTOMÁTICOS**
- **SISTEMA DE TRÁNSITO**
- **SEÑALES DE TRÁNSITO**
- **APRENDIZAJE SITUADO**

ABSTRACT

This degree project consists of the development of a 3D simulation system that allows the learning of traffic signs and has been tested with children from the Victor Manuel Peñaherrera School. The application uses a city model that includes many details such as buildings, streets, avenues, churches, hotels, hospitals, fire stations, gas stations, bus stops, automobiles. On this basis, the animation of a transit system is constructed that includes two main actors: cars and pedestrians. These objects are constituted in the NPCs (non-player characters) of the simulation system and are endowed with artificial intelligence to circulate in the city. The automatic cars respect the traffic lights, brake if a car stops in front of them, respect the stop sign and act appropriately in the curves and in the ascents and descents present in the simulated city. The application allows the player to take on the role of pedestrian or driver of a car and carry out a basic mission that consists of going from one place to another in the city. To make the application attractive, a scoring system was implemented that penalizes three types of infractions committed by the driver: not respecting the traffic light, exceeding the speed limit, driving against the road. In the role of pedestrian the application allows the user to move through the city and at specific points receives audible information of the correct behavior of a pedestrian when driving through the city.

KEY WORDS

- **SIMULATION SYSTEM**
- **AUTOMATIC CARS**
- **TRANSIT SYSTEM**
- **TRAFFIC SIGNALS**
- **LOCALIZED LEARNING**

CAPÍTULO I

INTRODUCCIÓN

1.1 Antecedentes

En la actualidad la educación vial tiene gran importancia por el incremento de accidentes de tránsito que ocurre diariamente. Hoy en día las distancias son cada vez más cortas entre pueblos y ciudades debido a la construcción de carreteras, autopistas y vías de circulación rápida que llevan al pasajero a su destino, por lo que es más alto el porcentaje de peligro de sufrir un accidente de tránsito de los peatones y conductores, debido a la falta de conocimiento sobre el uso correcto de utilización de la vías, calles, pasos peatonales conforme a leyes, normas y reglamentos de tránsito (Arias Ramos, 2015).

En el Ecuador en el año 2013 en el Pleno de la comisión de lo Económico se determinó la creación de la **Cátedra de Educación Vial** para que sea dictada de manera obligatoria en todos los establecimientos de educación públicos y privados del país en todos sus niveles. En esta cátedra se tratan temas relacionados con la prevención y seguridad vial, así como los principios, disposiciones y normas fundamentales que regulan el tránsito, su señalización, el uso de las vías públicas, de los medios de transporte terrestre, de conformidad con los programas de estudios elaborados conjuntamente por la Comisión Nacional de Transporte Terrestre, Tránsito y Seguridad Vial y el Ministerio de Educación (Telégrafo, 2014). Como resultado, en el pensum académico del año 2016 en el currículo del área de ciencias sociales consta la información sobre Seguridad Vial como una destreza básica deseable («Ciencias Sociales – Ministerio de Educación», s. f.).

Existen aplicaciones desarrolladas en el ámbito de seguridad vial para niños, que les explican lo positivo y lo negativo de sus acciones al encontrarse en calles o avenidas y algunas reglas de tránsito, sin embargo, estas aplicaciones no le permiten interactuar al niño para realizar la acción correcta cuando se encuentre con estos percances en la vida real. Para que las situaciones aparenten ser reales es importante incorporar personajes que produzcan la ilusión de inteligencia en su comportamiento. Ante tal situación se plantea desarrollar un **Sistema de Simulación de Educación Vial 3D con Inteligencia Artificial** que le permita al niño ubicarse en un entorno en

el cual él pueda realizar las acciones de forma correcta para lograr el objetivo de mantener su vida a salvo al momento de encontrarse en este tipo de situaciones.

1.2 Planteamiento del Problema

Una de las principales causas de muerte en el Ecuador son los accidentes de tránsito, los cuales implican peatones y conductores. Solamente en el año 2017, los accidentes de tránsito en Ecuador dejaron 2.153 fallecidos y lesionaron a más de 22 mil personas. Pere Navarro, exdirector general de tránsito de España comentó “Ecuador tiene más del doble de muertos por siniestros de tránsito que por homicidios y tiene 128 muertos por accidentes de tránsito por cada millón de habitantes” (Telégrafo, 2018).

El factor humano es una de las principales causas de este tipo de accidentes ya que existe un desconocimiento de las leyes de tránsito del Ecuador, o se da poca importancia a las mismas, ya que no existen mecanismos que logren que la persona tenga mayor interés en este tema, especialmente en los niños a los cuales únicamente se les brinda información con medios tradicionales como trípticos o pancartas o de vez en cuando una presentación de las reglas de tránsito y la forma de actuar en las calles. Sin embargo, no se le permite al niño demostrar el conocimiento obtenido lo que se observa cuando este sigue incumpliendo las normas por no prestarle la importancia necesaria.

Además, existen carreteras que se encuentran frente a escuelas en las cuales el niño tiene que buscar la mejor forma de cruzar para llegar a su destino, sin la ayuda de una persona mayor, situación que incrementa el peligro de un accidente. La falta de herramientas pedagógicas enfocadas en la educación vial en las escuelas provoca un escaso aprendizaje, debido a la falta de motivación y atención de los niños. Por este motivo se plantea desarrollar un Sistema de Simulación Vial que tenga una interfaz amigable la cual incentive al niño a cumplir con las reglas de tránsito y entender las actitudes que debe tener al transitar por las vías de la ciudad.

La escuela Víctor Manuel Peñaherrera, se encuentra en el cantón Quito, en la Parroquia de Tumbaco, pertenece a las instituciones fiscales y fiscomisionales. En la escuela los mecanismos tradicionales no han logrado los mejores resultados en los

niños, por lo que se les ha planteado el desarrollo de esta aplicación, mediante la cual se busca que los niños le tomen mayor importancia al tema de la educación vial.

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar un Sistema de Simulación 3D sobre Educación Vial Utilizando Técnicas de Inteligencia Artificial, aplicando las mejores prácticas de la Ingeniería de Software y validarlo con los niños de la Escuela “Víctor Manuel Peñaherrera”, de la parroquia Tumbaco del cantón Quito.

1.3.2 Objetivos Específicos

- Revisar el estado del arte de las aplicaciones de multimedia educativa enfocada a la educación vial.
- Realizar el análisis de requerimientos aplicando la norma ISO 830.
- Aplicar la metodología Huddle para el diseño y desarrollo del sistema de simulación.
- Realizar la implementación y las pruebas del sistema de simulación en la escuela “Víctor Manuel Peñaherrera”.

1.4 Justificación

Los accidentes de tránsito conllevan un gasto al Estado por la pérdida de futuros ciudadanos que serán el motor productivo y el eje económico del país. Con este precedente es indispensable educar a la ciudadanía en general, sobre la importancia de prevenir los accidentes de tránsito para evitar la calamidad doméstica o económica que deja secuelas graves en la vida de los afectados o que en algunos produce la muerte.

Con la creación de un software de simulación 3D que promueva la Educación en Seguridad vial, se logra mejorar el aprendizaje—educación de las normas básicas y la ley de tránsito para generar peatones con buenas actitudes con respecto al cumplimiento y respeto a la ley. La meta que se persigue es enseñar conductas para que las personas procedan de manera prudente para evitar y prevenir accidentes.

La principal área de dominio de la Universidad de las Fuerzas Armadas “ESPE” en el desarrollo de la investigación es la Seguridad y Defensa. El desarrollo de un software aplicado para la educación en seguridad vial contribuye de manera directa a alcanzar los objetivos institucionales de investigación y vinculación con la sociedad en esta área.

A los niños que se les enseña desde pequeños la educación vial crecen en un entorno en el que cumplir las normas básicas es lo normal. La herramienta de simulación 3D se muestra como una herramienta útil para complementar la metodología tradicional de enseñanza de educación vial para los niños en edad escolar. Para demostrar su utilidad, el sistema ha sido probado en los estudiantes de cuarto, quinto y sexto año de educación básica de la Escuela Víctor Manuel Peñaherrera.

1.5 Alcance

Se ha desarrollado un Sistema de Simulación 3D sobre Educación Vial que permite incorporar información de las leyes de tránsito del Ecuador donde el usuario puede actuar en dos roles: como conductor o como peatón.

Se incorporaron técnicas de Inteligencia artificial para simular un entorno animado 3D atractivo donde los personajes no jugadores (PNJ) realizan acciones parecidas a las reales. Los PNJ tienen animación de inteligencia artificial usando las técnicas de búsqueda de ruta y toma de decisiones. La animación de los PNJ incluye el respeto de los pasos peatonales y pasos cebras y la actuación adecuada con los semáforos.

El sistema consiste en el desarrollo de escenas en 3D con animación que contienen los elementos de educación de seguridad vial siguiendo una secuencia de aprendizaje definida que permite influir en la conducta del niño a través de la interiorización de la importancia de conocer y respetar las normas de tránsito. Para el rol del conductor, la secuencia de aprendizaje en las escenas se trabaja desde el punto de vista de las señales de tránsito, considerando las señales informativas, luego las señales preventivas y las señales regulatorias. En su rol de peatón el usuario aprende las buenas actitudes que debe tener para una circulación segura en la calle.

CAPÍTULO II

MARCO TEÓRICO

2.1 Seguridad Vial

En el anuario de estadísticas de transporte publicado por el Instituto Nacional de estadísticas y Censos se publica una información bajo el apartado Accidentes de tránsito. Textualmente se expresa: “Durante el año 2016, se registraron 30.269 siniestros por accidentes de tránsito. El promedio por mes fue de 2.522, siendo enero el mes con mayor número de accidentes con el 10,1% respecto del total. La provincia del Pichincha registró el mayor número de accidentes con el 35,6%, siguiéndole la provincia del Guayas con el 26,1% y en tercer lugar la provincia de Imbabura con el 5,1%. Las causas que provocaron el 77,7% de los accidentes fueron: impericia e imprudencia del conductor (51,9%), no respeta las señales de tránsito, (13,4%) y exceso de velocidad (12,4%). Con respecto a la clase de accidentes, los choques constituyen el mayor número de accidentes con el 45,4%, siguiéndole los atropellos con el 16,8% y estrellamientos el 13,3%, los mismos que constituyen el 75,5% de accidentes ocurridos en el país.” («Anuario De Estadística De Transporte 2016 Resumen ...», s. f.). Estas cifras revelan la alarmante situación que se presenta en las vías de nuestro país por la poca importancia que la población presta al conocimiento y respeto de las normas de tránsito.

2.1.1 Los Involucrados en la vía

Los principales actores viales son el conductor y el peatón. El conductor es una persona facultada para desempeñar tal actividad por la autoridad competente de tránsito. Para ello debe cumplir los requisitos legales. El conductor debe conocer, cumplir y acatar las normas de tránsito. Al conducir el vehículo debe respetar el paso de los peatones, ciclistas y motociclistas. Debe estar atento a cualquier situación cuando el coche se encuentra en movimiento. Un peatón, según la Real Academia de la Lengua Española, es una persona que va a pie por una vía pública. El tránsito de peatones en Ecuador está delimitado “por fuera de las zonas destinadas al tránsito de vehículos”.

Las obligaciones de los conductores son:

- Respetar las señales de tránsito.
- Manejar con precaución.
- Tener el certificado médico y la licencia vigente.
- Usar el cinturón de seguridad.

El tránsito de peatones se restringe a puentes peatonales, aceras, pasos cebra. Los pasos cebra son unas líneas longitudinales demarcadas en la calle paralelas al flujo del tráfico, de colores oscuro y claro generalmente alternados. Por allí deben cruzar los peatones, generalmente en calles de alto flujo vehicular. Con ellas, el peatón tiene prioridad la hora de cruzar sobre el conductor de vehículos.

La principal regla con respecto a los peatones es que “Cuando un peatón requiera cruzar una vía vehicular, lo hará respetando las señales de tránsito y cerciorándose de que no existe peligro para hacerlo”. Los peatones tienen prohibidas una serie de conductas. Entre ellas están: invadir la zona destinada al tránsito de vehículos, ni transitar en ésta en patines, monopatines, patinetas o similares. Llevar, sin las debidas precauciones, elementos que puedan obstaculizar o afectar el tránsito.







El peatón debe meditar sobre las situaciones que ponen en peligro su vida como colocarse delante o detrás de un vehículo que tenga el motor encendido. En los parqueaderos es usual que esta situación se presente y por eso es importante tener cuidado de no ubicarse detrás de ellos. Otras posibles situaciones de riesgo son: remolcarse de vehículos en movimiento, cruzar la vía atravesando el tráfico vehicular en lugares en donde existen pasos peatonales, subirse o bajarse de los vehículos, estando éstos en movimiento, cualquiera que sea la operación o maniobra que estén realizando.

2.1.2 Las Señales de Tránsito

Las señales de tránsito son imágenes puestas en carteles en la vía pública, cuyos símbolos tienen un significado especial y tienen diferentes colores y formas. El significado de cada una de éstas es informar al conductor sobre las normas de tránsito que debe respetar así como también alertar sobre las situaciones que se presentan en la vía pública.








- Señales informativas. Estas tienen la función como su nombre lo indica de informar y al mismo tiempo alertar al peatón y conductor sobre la presencia de sitios que tienen interés público como son los parqueaderos, parada de bus, restaurantes, iglesias, hospitales, aeropuerto, etc., como se detalla en la Tabla 1.

Tabla 1*Señales Informativas.*

 AEROPUERTO	Aeropuerto: Señal Informativa que indica que el camino conduce al aeropuerto.
 ESTACIÓN DE SERVICIO	Estación de servicio: Señal informativa que señala la proximidad de una estación de abastecimiento de combustible.
 RESTAURANT	Restaurante: Señal informativa que indica la cercanía de un restaurante.
 MECÁNICO	Mecánica: Señal informativa que señala la proximidad de una mecánica.
 IGLESIA	Iglesia: Señal informativa que señala la cercanía a un templo católico.
 TELÉFONO	Teléfono público: Señal informativa que indica la cercanía a un teléfono público

- Señales preventivas. Las señales de tránsito preventivas también pueden llamarse de prevención, estas señales tienen la función de prevenir al peatón o conductor de situaciones peligrosas o no. Su forma es un rombo de color amarillo con imágenes en negro, como se detalla en la Tabla 2.



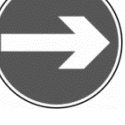
Tabla 2*Señales Preventivas.*

	Zona escolar: Advierte sobre una zona de escuela cercana para que el conductor reduzca su velocidad.
	Semáforo: Este señalamiento advierte sobre la presencia de un semáforo próximo, normalmente en cruces o en zonas donde no se espera hallarlos.
	Doble circulación: El señalamiento se utiliza para marcar el cambio de circulación de un sólo sentido a circulación en doble sentido.
	Curva cerrada: Este tipo de señal de tránsito se usa para indicar curvas a la derecha o izquierda con una curvatura de 90 grados
	Empalme T: Esta señal indica la presencia de una calle que se empalma con la principal. La línea gruesa muestra el camino principal y la angosta el camino secundario en donde los autos deben ceder el paso.
	Cruce: Advierte sobre el cruce de dos caminos. La línea más ancha señalará el camino principal mientras las más angosta es el camino secundario.
	Paso peatonal: Indica un camino con constante paso peatonal o un cruce peatonal en específico.

- Señales regulatorias. Estas tienen la función como su nombre lo indica de obligar al conductor, a cumplir con la información señalada por las mismas.

La Tabla 3 resume las principales señales regulatorias

Tabla 3*Señales Regulatorias.*

	Límite de velocidad: Indica la velocidad a la cual se debe circular en la vía.
	Pare: Es una señal de tráfico reglamentaria que indica en las intersecciones la obligación de detenerse antes de continuar la marcha.
	Sentido Circulación: Indica la dirección de la vía en la cual se debe circular.

2.1.3 La dinámica de los vehículos

Es importante reconocer que cada vehículo es un sistema que reacciona en el ambiente siguiendo los principios fundamentales de la física. Las ideas básicas que se deben tener en cuenta son las relaciones entre las variables de los vehículos. Algunas de estas ideas se enuncian a continuación:

- El movimiento del coche se produce debido a la fuerza de deslizamiento entre las llantas y el piso. La fuerza de deslizamiento se produce por la aplicación de un torque a las llantas que proviene del motor. El torque se convierte en fuerza de deslizamiento al dividirlo para el radio de la llanta. Esta fuerza produce una aceleración directamente proporcional y la constante de proporcionalidad es la masa del vehículo.
- Cuando el vehículo adquiere velocidad se tiene que ha almacenado energía en forma de energía cinética. La energía cinética de un cuerpo es directamente proporcional al cuadrado de la velocidad. Para detener el vehículo mediante la aplicación del freno, la energía cinética debe convertirse en energía de rozamiento. Esta transformación de la energía permite determinar la distancia requerida para detener un vehículo.
- Las distancias de parada a partir de determinada velocidad pueden obtenerse considerando que la energía cinética del coche se transforma en energía de rozamiento según la Ecuación 1.

$$E_c = E_r \quad (1)$$

La energía cinética depende de la masa del coche y del cuadrado de la velocidad. (Ver Ecuación 2)

$$\frac{1}{2}mv^2 = mgd_f\mu \quad (2)$$

En la energía de rozamiento intervienen la masa del coche (m) la gravedad (g), la distancia de paro (d_f) y el coeficiente de rozamiento del piso (μ).

2.2 Cognición situada y aprendizaje en contextos escolares

El aprendizaje situado es una de las tendencias actuales más importantes y con mayores posibilidades de éxito para la adquisición de conocimiento y es un tema

que adopta varios nombres como aprendizaje cognitivo, aprendizaje artesanal o cognición situada. La práctica del aprendizaje situado se contrapone a las prácticas educativas arraigadas en el sistema escolar donde se abstrae el conocimiento de las situaciones donde se lo emplea. El aprendizaje situado parte de la premisa que el conocimiento es situado, es parte y producto de la actividad, el contexto y la cultura en que se desarrolla y utiliza.

La instrucción que se imparte en la enseñanza situada destaca la importancia de la actividad y el contexto para el aprendizaje y asume que el aprendizaje es un proceso de integrarse a una cultura de prácticas sociales, por lo tanto es importante que los aprendices sean situados en el contexto pertinente.

Los expertos que defienden el aprendizaje situado critican la forma como se promueve el aprendizaje en forma tradicional y en particular, cuestionan la forma en que se enseñan aprendizajes declarativos abstractos y descontextualizados, conocimientos inertes, poco útiles y escasamente motivantes, de relevancia social limitada (Arceo, 2003). En estas prácticas se manifiesta una separación notable entre el saber qué (Know what) y el saber cómo (Know how) donde se asume que el conocimiento es autosuficiente e independiente de las situaciones de la vida real. Esto ha producido que los alumnos de los sistemas escolares adquieran aprendizajes poco significativos, carentes de significado y que sean incapaces de aplicar y generalizar lo que han aprendido.

En el aprendizaje situado se busca una enseñanza centrada en la realización de actividades educativas auténticas, es decir, "prácticas ordinarias de la cultura". Los educandos aprenden involucrándose en el mismo tipo de actividades que enfrentan los profesionales de los diferentes campos culturales. En la educación tradicional, se intenta simular las prácticas científicas y culturales de los expertos de cada área de conocimiento, y se pretende que los alumnos tengan las mismas habilidades que los matemáticos biólogos o historiadores, pero como la enseñanza no se aplica en un contexto real, los aprendices no reflexionan ni son capaces de generalizar lo que aprenden. Además se ha demostrado que los expertos realizan su actividad de una manera muy diferente a la que los alumnos realizan sus trabajos de matemáticas, historia (Hendricks, 2001). El experto ha adquirido un conocimiento que no solo es más abundante o profundo que el de los aprendices sino que tiene mucha más calidad por

que ha sido sometido a procesos de reflexión, análisis y ajustado a los procesos reales.

En el aprendizaje situado, el aprendizaje implica el entendimiento e interiorización de los símbolos y señales de la cultura. Los aprendices se apropian de las prácticas a través de la interacción con miembros más experimentados. Es importante en este paradigma la relación alumno-alumno o aprendizaje social que permite la construcción de significados en forma colaborativa. En esta perspectiva del aprendizaje, la interacción entre los individuos que aprenden y el contexto en que realizan la actividad es más importante que el conocimiento “frio”.

El aprendizaje situado puede considerarse como un sistema donde la entrada es el sujeto que aprende, existen herramientas para realizar la actividad donde se privilegia el aprendizaje de los símbolos, un objeto de aprendizaje que determina los saberes o los contenidos a apropiarse, una comunidad a la que pertenece el aprendiz con sus respectivas normas. Se concibe el aprendizaje como un cambio de comportamiento que involucra el pensamiento, la afectividad y la acción. Para Dewey "toda auténtica educación se efectúa mediante la experiencia, el aprendizaje experiencial es activo y genera cambios en la persona y en su entorno y no sólo va al interior del cuerpo y alma del que aprende, sino que utiliza y transforma los ambientes físicos y sociales para extraer lo que contribuya a experiencias valiosas y establecer un fuerte vínculo entre el aula y la comunidad”.

Una de las estrategias para el aprendizaje significativo se centra en el aprendizaje experiencial y situado, que se enfoca en la construcción del conocimiento en contextos reales, en el desarrollo de las capacidades reflexivas, críticas y en el pensamiento de alto nivel, así como en la participación en las prácticas sociales auténticas de la comunidad y es el aprendizaje mediado por las nuevas tecnologías de la información y comunicación (NTIC).

2.2.1 El aprendizaje situado y las nuevas tecnologías de información y comunicación

Aquello que se aprende a través de la pasión y la motivación raramente se olvida (Mark Prensky, 2011). En la misma línea, (Francisco Mora, 2013) afirma que “solo se puede aprender aquello que se ama”. La escuela debe generar escenarios y

situaciones divertidas y emocionantes a fin de despertar la curiosidad y deseo de aprender del alumnado (Richard Gerver, 2012). De este modo, conectar con los intereses y emociones del alumnado permitirá implicarlo en su aprendizaje y hacerlo más efectivo (David Istance, 2012).

Las tecnologías de la información y comunicación (TICs) están cambiando las formas de enseñar, aprender, participar, trabajar y divertirse (César Coll, 2010). Las TICs se han incorporado en forma notable en todos los campos de actividad de las personas y por ello son una herramienta de uso común que debe ser aprovechada de la mejor manera posible. El aprendizaje mediante las TICs determina un cambio de comportamiento tanto del profesor como del alumno y un cambio en la forma como ellos se relacionan. El profesor deberá adoptar otros mecanismos de interacción con sus alumnos y deberá diseñar actividades que construyan experiencias educativas. (Ullastres, 2009) sostiene que incorporar las TICs en la educación supone repensar los retos y objetivos de la educación actual y decidir como las TICs pueden contribuir a su consecución.

2.3 Herramientas para el desarrollo del sistema

Para la implementación del sistema de simulación es necesario utilizar los recursos más variados que están disponibles en el entorno de desarrollo Unity que permite disponer de muchas herramientas que se adaptan a la necesidad del problema a resolver. A continuación se mencionan algunos de los elementos más importantes que han sido utilizados para la aplicación desarrollada.

2.3.1 Máquinas de Estados

Es un modelo de sistema con entradas y salidas en el que las salidas no dependen sólo de las entradas actuales sino también de estados anteriores. Las máquinas de estados se representan mediante diagramas de estado. Una máquina de estados se denomina máquina de estados finitos (FSM finite state machine) si el conjunto de estados de la máquina es finito, este es el único tipo de máquinas de estados que se puede modelar en un computador en la actualidad.

2.3.2 Sistema de Navegación de Unity: NavMesh

El sistema de navegación permite crear personajes que pueden moverse inteligentemente en el mundo del juego y utiliza mallas de navegación para guiarse en el medio ambiente, que se crean automáticamente a partir de la geometría de la escena. Los obstáculos dinámicos permiten alterar la navegación de los personajes en tiempo de ejecución. Los enlaces off-mesh le permiten crear acciones específicas como abrir puertas, o saltar desde una cornisa.

Este sistema de navegación les otorga a los personajes la habilidad de entender que necesitan subir escaleras para llegar al segundo piso, o saltar para cruzar una zanja.

El sistema Unity NavMesh consta de las siguientes piezas:

- **NavMesh** (abreviatura de Navigation Mesh) es una estructura de datos que describe las superficies transitables del mundo del juego y permite encontrar el camino de un lugar caminable a otro en el juego mundo. La estructura de datos se construye, o se hornea, automáticamente a partir de la geometría de su nivel.
- **El componente Agente Malla de Navegación (*NavMesh Agent*)** ayuda a crear caracteres que se evitan entre sí mientras se mueven hacia su objetivo. Los agentes razonan sobre el mundo del juego usando la malla de navegación *NavMesh* y saben cómo evitarse unos a otros, así como los obstáculos en movimiento.
- **El componente Enlaces fuera de Malla (*Off-Mesh Link*)** permite incorporar accesos directos de navegación que no pueden ser representados mediante una superficie transitable. Por ejemplo, saltar por encima de una zanja o una cerca, o abrir una puerta antes de pasar a través de ella, puede ser descrita como enlaces *Off-mesh*.

2.3.3 Seguimiento de una trayectoria

La característica básica de los coches inteligentes es que puedan seguir una trayectoria definida (waypoints) que incluye un conjunto de puntos en el camino y cada punto en el camino es una coordenada en el espacio 3D que representa una posición clave en las vías de la ciudad. Este sistema ha sido utilizado por muchos de los primeros juegos de carreras de coches debido a su eficacia y simplicidad. “En Racer,

los puntos en el camino se almacenan como una lista ordenada de posiciones en el espacio 3D utilizando la lista genérica del espacio de nombres y el sistema de software Thegame lee esta lista de puntos en el camino como entrada y luego itera a través de la lista hasta que todos los puntos que tenga el camino haya sido pasado por el coche controlado por el juego” (Chan, Chan, & Gelowitz, 2016). El sistema de juego sólo itera al siguiente punto en el camino de la lista cuando detecta que el coche controlado por el juego está dentro de una distancia especificada desde su actual punto de referencia.

2.3.4 Controladores de Personajes (*Characters Controllers*)

El personaje en un juego de primera o tercera persona va a necesitar físicas basadas en colisiones para que no caiga a través del suelo o camine a través de las paredes. Sin embargo, usualmente la aceleración del personaje y el movimiento no serán físicamente reales, por lo que podría acelerar, frenar y cambiar de dirección casi instantáneamente sin ser afectado por el momento.

En físicas 3D, este tipo de comportamiento puede ser creado usando un Controlador de Personaje (*Character Controller*). Este componente le da al personaje un simple colisionador (*collider*) en forma de capsula que siempre está en posición vertical. El colisionador es como un muro que cubre al personaje y le permite ser golpeado por otros objetos pero especialmente por rayos emitidos por otros objetos que permiten detectarlo. El controlador tiene sus propias funciones especiales para establecer la velocidad y dirección del objeto pero a diferencia de los verdaderos colisionadores (*colliders*), un cuerpo rígido (*rigidbody*) no es un objeto requerido y los efectos del momento no son realistas.

Un controlador de personaje (*character controller*) no puede caminar a través de colisionadores estáticos (*colliders static*) en una escena, de igual forma seguirá caminos por el suelo y será obstruido por paredes. Puede empujar hacia un lado objetos con un cuerpo rígido (*rigidbody*), mientras se mueve, pero no será acelerado por colisiones entrantes. Esto significa que se puede usar los colisionadores (*colliders*) 3D estándar para crear una escena alrededor de la cual el controlador caminará pero no estará limitado por el comportamiento físico realista del personaje en sí.

2.3.5 Colisionador de rueda (WheelCollider)

El colisionador de la rueda (*Wheel collider*) es un colisionador especial para vehículos en tierra que lleva incorporado un sistema de detección de colisiones y un modelo de fricción de neumáticos basado en el deslizamiento. Puede utilizarse para objetos que no sean ruedas, pero está diseñado específicamente para vehículos con ruedas. La Tabla 4 resume las principales propiedades de los colisionadores de rueda con su función respectiva.

Tabla 4

Propiedades de los WheelColliders.

Propiedad	Función
Mass	La masa de la rueda.
Radius	Radio de la rueda.
Wheel Damping Rate	Amortiguador de la velocidad de suspensión.
Suspension Distance	Distancia máxima de extensión de la suspensión de las ruedas, medida en el espacio local. La suspensión siempre se extiende hacia abajo a través del eje local 'y'.
Force App Point Distance	Este parámetro define el punto en el que se aplicarán las fuerzas de la rueda y debe estar en metros desde la base de la rueda en posición de reposo a lo largo de la dirección de marcha de la suspensión. Si <code>forceAppPointDistance = 0</code> , las fuerzas se aplicarán en la distancia entre ejes en reposo. Un vehículo mejor tendría que aplicar fuerzas ligeramente por debajo del centro de masa del vehículo.
Center	Centro de la rueda en el espacio local del objeto.
Suspension Spring	La suspensión intenta alcanzar una Posición Objetivo añadiendo resortes y fuerzas de amortiguación.
Spring	La fuerza del resorte intenta alcanzar la Posición de Objetivo. Un valor mayor hace que la suspensión alcance más rápidamente la posición de destino.
Damper	Amortiguador de la suspensión. Un valor mayor hace que el resorte de suspensión se mueva cada vez más lento.
Target Position	La distancia de descanso de la suspensión a lo largo de la Distancia de Suspensión (<code>Suspension Distance</code>), donde el valor de 1 mapea la suspensión completamente extendida, y el valor de 0 mapea la suspensión completamente comprimida. El valor predeterminado es 0,5 que coincide con el comportamiento de la suspensión de un automóvil normal.

2.3.6 Canvas

El Canvas es un *gameobject* que permite realizar interfaces de usuario *UI* donde se colocan todos los elementos gráficos que se desea mostrar en la pantalla directamente para comunicar al usuario con la aplicación.

El Canvas se muestra como un rectángulo en la pantalla y se lo observa como un objeto 2D. Los diferentes elementos se ubican dentro de esta superficie y se los

posiciona según la conveniencia de la aplicación. Los elementos que pueden ser incluidos en el Canvas son:

- **Text:** Este es un *GameObject* que permite ingresar texto directamente en la pantalla y al crear este objeto, se puede cambiar el tipo de fuente, el tamaño de la letra y el formato del texto ingresado.
- **Image:** Este es un *GameObject* que se refiere a las imágenes donde para ingresar una imagen, se tiene que dar un formato a la misma de tipo *Sprite*, para cargarla en el componente *image*, del objeto.
- **RawImage:** A diferencia del objeto *image*, este objeto utiliza texturas y puede aceptar la imagen enfocada de una cámara, lo que permite observar la escena con un enfoque distinto.

2.3.7 Line Renderer

El *line renderer* es un componente de *Unity* que permite crear trayectorias para guiar al jugador sobre el camino que debe seguir y llegar a la meta. Para construir la trayectoria se utiliza una serie de puntos en el espacio, los cuales se colocan en un arreglo. El *line renderer* toma las posiciones escritas en el arreglo y los utiliza como los vértices de la trayectoria. En la Figura 1 se muestran los componentes del *line renderer* con la configuración por defecto.

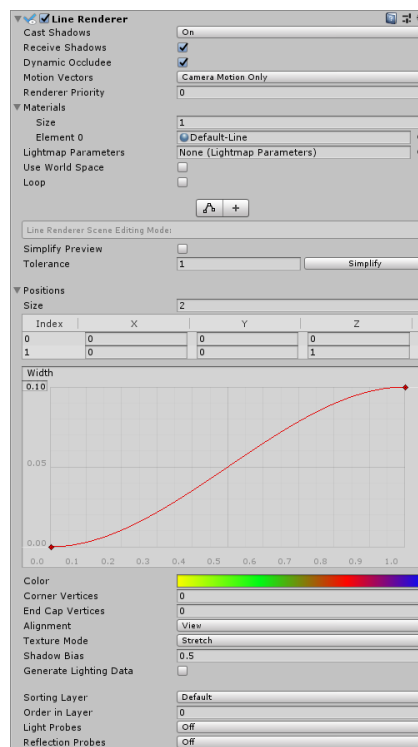


Figura 1. Componentes del Line Renderer.

2.3.8 Scripts

La programación de *scripts* es un ingrediente esencial en el desarrollo de sistemas tipo videojuegos. En todas las aplicaciones se necesita scripts para responder a las entradas (*input*) del jugador y asegurar que los eventos se ejecuten en el momento adecuado. Los *scripts* se utilizan para crear efectos gráficos, controlar el comportamiento físico de objetos o para implementar la inteligencia artificial en los personajes del juego.

Los *Scripts* se consideran como componentes de comportamiento en *Unity*, al igual que otros componentes pueden ser aplicados a objetos y ser vistos en la ventana del Inspector de Unity. Los *Scripts* se crean en lenguaje C# o en JavaScript y añaden comportamiento a los objetos que manejan la funcionalidad de la aplicación.

Los *Scripts* se escriben en un editor de texto que para *Unity* es Visual Studio.NET. La Figura 2 muestra los contenidos iniciales del archivo de un script, donde se observa que existen dos funciones definidas dentro de la clase NewBehaviourScript.


```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class NewBehaviourScript : MonoBehaviour {
6
7      // Use this for initialization
8      void Start () {
9
10     }
11
12     // Update is called once per frame
13     void Update () {
14
15     }
16 }
17

```

Figura 2. Ejemplo de Inicio de Script.

La función *Update* es el lugar para colocar el código que se encargará de la actualización por cuadro (*frame*) del objeto del juego (*GameObject*). El objeto *GameObject* puede incluir movimiento, acciones de disparo (*trigger*), responder a la entrada (*input*) del usuario y básicamente cualquier cosa que necesite ser manejada en el tiempo durante el juego (*gameplay*). Para que la función *Update* haga su trabajo, a veces es útil configurar variables, leer preferencias y hacer conexiones con otros *GameObjects* antes de que cualquier acción del juego tome lugar. La función *Start* es llamada por *Unity* antes de que el *gameplay* comience (es decir, antes de que la función *Update* sea llamada por primera vez) y es un lugar ideal para hacer cualquier inicialización.

2.4 La Inteligencia Artificial en los Sistemas de Simulación

La inteligencia artificial es la inteligencia exhibida por las máquinas; en ciencias de la computación, una máquina «inteligente» ideal es un agente racional flexible que percibe su entorno y lleva a cabo acciones que maximicen sus posibilidades de éxito en algún objetivo o tarea (David Poole, s. f.).

La inteligencia artificial ha sido un tema bastante tratado desde los inicios de las computadoras. En los últimos años se han desarrollado algoritmos capaces de superar a los mejores jugadores del mundo de ajedrez. Éstos últimos han sido realmente importantes, ya que marcan un antes y un después en la determinación de objetivos alcanzables mediante el uso de inteligencia artificial (Gutiérrez Utande, 2017).

Si bien la Inteligencia Artificial actualmente presenta herramientas poderosas como las redes neuronales, los sistemas de lógica difusa, los sistemas de computación evolutiva, debe también notarse que con la dotación de un sistema de actuación lógica a los objetos de un sistema, se puede lograr agentes que demuestran cierto grado de inteligencia. En un sistema de simulación, la aplicación de la Inteligencia Artificial se hace evidente en el control de los PNJ (Personajes No Jugadores) que es una característica deseable en la que estos realizan la búsqueda de una ruta, es decir, buscan un camino para moverse de un punto en un mapa a otro, teniendo en cuenta el terreno y evitando los obstáculos. La IA del juego se preocupa de dotar a los PNJ de la capacidad de navegar en su entorno, la búsqueda de un camino hacia un objetivo, evitando colisiones con otras entidades o colaborar con ellos.

En este contexto, se busca que los PNJ puedan tomar decisiones basados en las condiciones del entorno. Para ello es necesario que los PNJ dispongan de sensores que les permitan monitorear las condiciones según las cuales deben actuar. Esta característica de monitorear el ambiente y actuar en consecuencia le otorga al PNJ la capacidad de tomar decisiones lógicas como las que tomaría un individuo inteligente. Un sistema que simule un sistema de tráfico real, inevitablemente requiere la presencia de inteligencia artificial en los entes que forman el sistema de tránsito que son los peatones y los conductores. El aprendizaje de señales de tránsito en un entorno similar a la realidad con coches y peatones circulando y obedeciendo las leyes de tránsito generará una motivación importante en los aprendices.

Tal sistema permitirá adicionalmente analizar ciertos fenómenos que están presentes en el marco del tráfico real lo que representa una gran ventaja frente a realizar experimentos con los autos en los caminos lo que provocaría inconvenientes. Se concluye que es de gran importancia el modelado y simulación de un sistema de tránsito de vehículos puesto que permite obtener conocimientos adicionales como la dinámica de los sistemas de tráfico sin necesidad de molestar a los conductores ni provocar un caos vial.

2.5 Hardware

2.5.1 Volante Gigasus

La Figura 3 muestra el tipo de volante utilizado en este proyecto, que es un volante de marca Gigasus, y cuya descripción se indica a continuación.

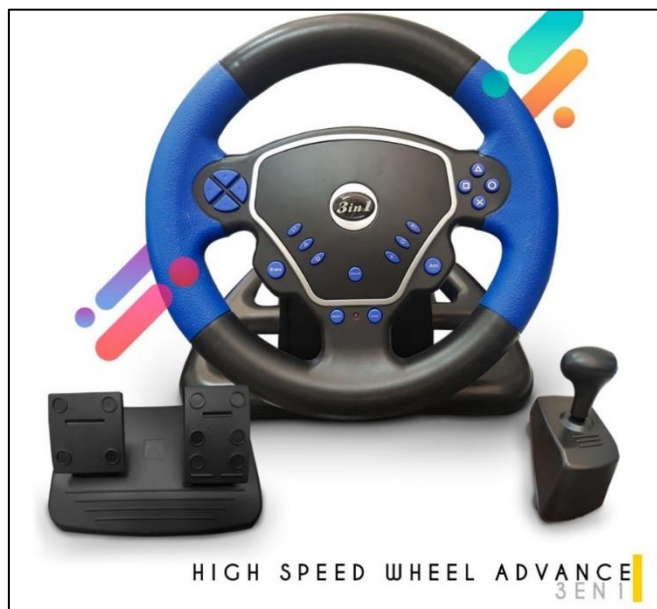


Figura 3. Volante Gigasus.

DESCRIPCIÓN:

- Volante: 10 pulgadas
- Vibración: Vibración de motores gemelos, vibración de sensibilidad
- Compatibilidad: Windows 7, 8, 10, 98, 2000, Windows XP, ME, Vista
- Rotación de la dirección: 250 grados
- Control direccional digital de 8 vías
- Cable de datos de entrada: Cable PS2 / USB

2.6 Software

2.6.1 Unity

Es un motor de juego desarrollado por Unity Technologies para diversas plataformas que incluye la herramienta de desarrollo Mono DEVELOP con la que podremos crear scripts en JavaScript, C#. (Aroca, 2012). La Figura 4 muestra el logo del motor de juegos Unity.



Figura 4. Logo de Unity.

Unity está disponible como plataforma de desarrollo para Microsoft Windows, OS X y Linux, y permite crear juegos para Windows, OS X, Linux, Xbox 360, PlayStation 3, Playstation Vita, Wii, Wii U, iPad, iPhone, Android y Windows Phone. Gracias al plugin web de Unity, también se pueden desarrollar videojuegos de navegador para Windows y Mac (Leañez, 2016).

Unity es un ecosistema completo para todo aquel que busque desarrollar un negocio a partir de la creación de contenido de alta gama y conectarse con sus jugadores y clientes más fieles y entusiastas (Spiritu3.com, 2015).

El editor de Unity 3D es uno de los más sencillos y potentes del mercado. Se divide en 5 vistas principales («El desarrollo de juegos con la herramienta Unity 3D», s. f.):

1. **Explorador:** Lista todos los elementos (o activos) de los proyectos y permite ordenar de forma sencilla la aplicación. En esta vista se encuentran las imágenes, escenas, scripts, audios, *prefabs*, texturas y todos los elementos que se usará en la aplicación. Los **Prefab** permiten crear, configurar y almacenar los objetos de juego (*GameObject*) en forma completa con todos sus componentes, valores de las propiedades y Objetos hijos (*ChildObjects*) como un *asset* reusable. El *asset Prefab* actúa como una plantilla desde la cual se puede crear otras instancias del *prefab* en la escena, donde un *asset* es la representación de cualquier ítem que puede ser usado en el proyecto.
2. **Inspector:** Muestra y define las propiedades de los elementos del proyecto, modifica valores de forma rápida, cambia texturas arrastrando ficheros desde el Explorador, añade scripts, guarda *prefabs*, etc.
3. **Jerarquía:** Lista jerárquica de los elementos de la escena del juego.
4. **Escena:** Diseño y maqueta del juego completo o una pantalla o sección de éste. Cada escena representa un nivel o sección diferente del juego, donde se arrastra los activos desde el Explorador y se edita sus variables desde el Inspector.

5. **Juego:** Visualiza el juego a distintas resoluciones.

La pantalla de inicio de la aplicación de Unity se distribuye en varias secciones. (Ver Figura 5)

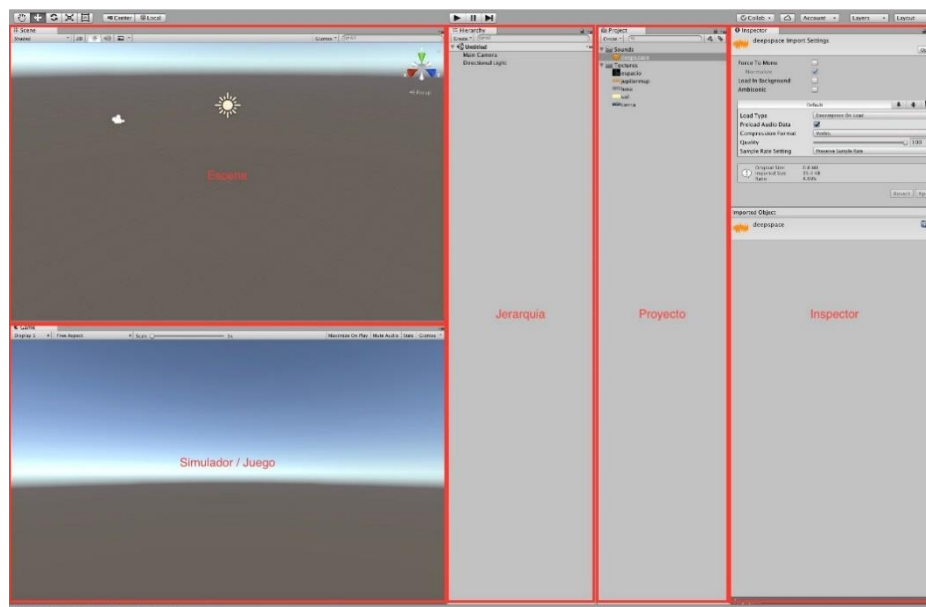


Figura 5. Pantalla de Unity.

2.6.2 MakeHuman

Es un software escrito completamente en C++ y Python que está disponible para todas las principales plataformas (Windows, OSX, Linux, etc.). Makehuman es software libre, las bases de datos y el código están liberados bajo una licencia GNU Licence Affero GPL, cuyo objetivo es el modelado de humanoides en 3 dimensiones («MakeHuman - EcuRed», s. f.). En la Figura 6 se presenta el Logo del software MakeHuman.



Figura 6. Logo de MakeHuman.

MakeHuman se desarrolla usando la tecnología morphing 3D, donde partiendo de una malla de base humana andrógina estándar (única) puede transformarse en

una gran variedad de caracteres (macho y hembra), mezclándolos con interpolación lineal («Tutoriales gratis de Makehuman en español», s. f.).

La pantalla de inicio de la aplicación se observa en la Figura 7 donde se presenta el objeto original sobre el que se realiza las modificaciones para obtener las características deseadas. Entre las características que se configuran están el tamaño, edad, contextura, sexo y ropa utilizada.

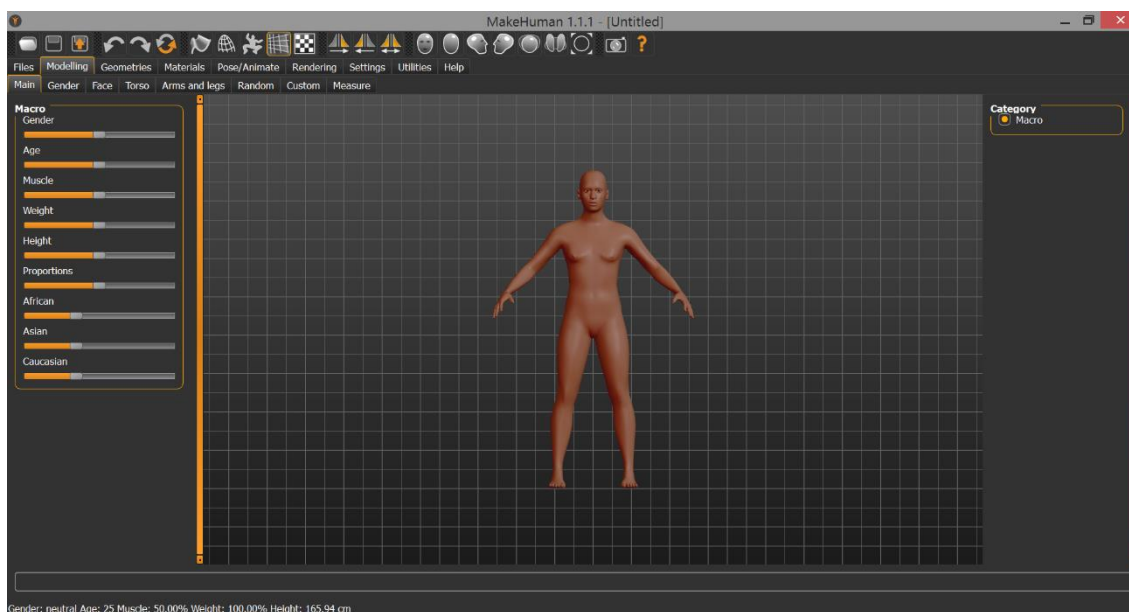


Figura 7. Pantalla Principal de MakeHuman.

2.6.3 Audacity

Es un editor de audio gratuito. Puedes grabar sonidos, reproducir sonidos, importar y exportar archivos WAV, AIFF, y MP3, y más («Audacity, programa para editar sonido digital. Software Libre», 2006). La Figura 8 muestra el logo del software Audacity.



Figura 8. Logo de Audacity.

Se puede usar Audacity para («Audacity», s. f.):

- Grabar audio en vivo.
- Convertir cintas y grabaciones a sonido digital o CD.

- Editar archivos MP3, WAV y AIFF.
- Cortar, copiar, unir y mezclar sonidos.
- Cambiar la velocidad o el tono de una grabación.

Audacity fue creado en otoño de 1999 por Dominic Mazzoni y Roger Dannenberg en la Universidad de Carnegie Mellon. Tras lo cual fue publicado en SourceForge.net como software libre en mayo de 2000 (H. Antonio Jiménez R, 2011).

La pantalla de inicio de la aplicación se muestra en la Figura 9 donde se dispone de un área donde se coloca los archivos de audio a ser editados.

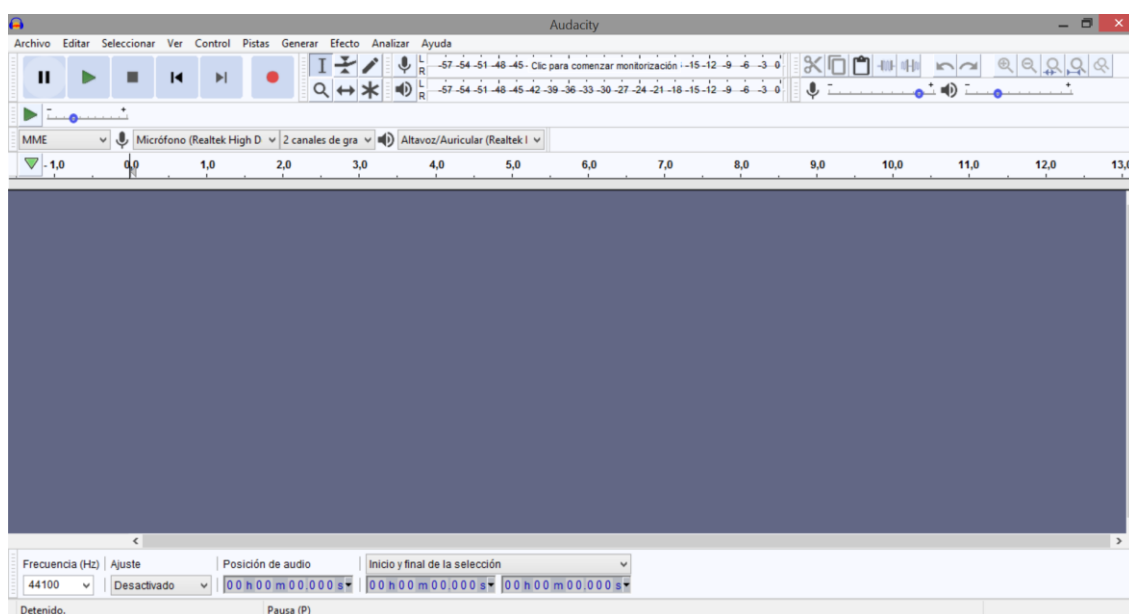


Figura 9. Pantalla de Audacity.

2.5.4 Zotero

Es un gestor o manejador de referencias bibliográficas, de acceso libre. Permite a los usuarios crear su propia base de datos (biblioteca) donde puedes: guardar, manejar, buscar, organizar y citar fuentes de todo tipo (*Zotero: Guía básica*, s. f.). La Figura 10 muestra el logo del programa Zotero.



Figura 10. Logo de Zotero.

Esta aplicación permite guardar referencias conforme se va descubriendo en un navegador (Torres, s. f.). Entre sus beneficios están («Gestor de citas bibliográficas Zotero - Universidad de Chile», s. f.):

- Permite crear una colección de citas bibliográficas de almacenamiento ilimitado.
- Acceder a estas citas bibliográficas a través de cualquier navegador de Internet.
- Almacenamiento en línea para archivos adjuntos, 300 MB de capacidad.
- Maneja una gran variedad de formatos (referencia PDF, vídeos, imágenes).
- Archiva páginas web.
- Almacena PDFs.
- Importar citas bibliográficas desde bases de datos: Catálogo Bello, Web of Knowledge, ScienceDirect, EBSCOhost, JSTOR, PubMed, entre otros.
- Detecta duplicados.
- Crea grupos personalizados (o carpetas) y compartirlos con cualquier usuario de Zotero.
- Disponer de variados (más de 6.000) estilos bibliográficos.
- Cuenta con un complemento para Microsoft Word y LibreOffice / OpenOffice / NeoOffice que permite editar documentos insertando citas en el texto y crear bibliografías al final.
- Dispone de aplicaciones para Android y Apps, para trabajar con dispositivos móviles.
- Cuenta con RSS.

La pantalla de inicio de la aplicación se muestra en Figura 11, donde se tiene un área de trabajo que guarda la bibliografía.

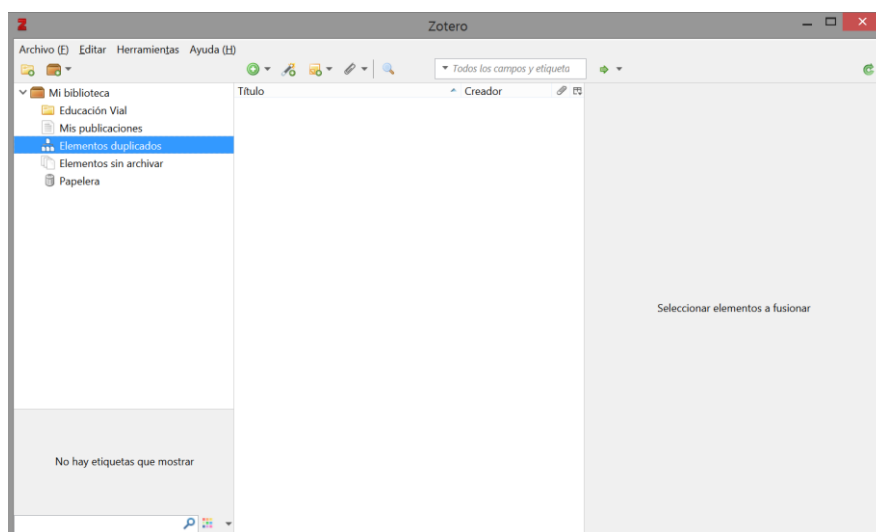


Figura 11. Ventana de Zotero.

2.5.4 Text2Speech

La aplicación Text2Speech permite generar archivos de audio a partir de texto. La Figura 12 muestra el logo de la aplicación.



Figura 12. Logo de Text2Speech.

El número máximo de caracteres que se admite es de 50000. La generación de archivos de audio mediante esta aplicación es un procedimiento sencillo. En la Figura 13 se muestra la caja de texto donde se coloca la información que se desea transformar a voz.

Convert text to speech

Simply copy and paste your text in the box below :

"Cuando un peatón requiera cruzar una vía vehicular, lo hará respetando las señales de tránsito y cerciorándose de que no existe peligro para hacerlo"

Figura 13. Entorno de ingreso del texto a convertirse en voz.

En la figura 14 se observa los cuadros de diálogo donde se selecciona el idioma y el personaje que habla que para el idioma español puede ser Mateo o Isabella.

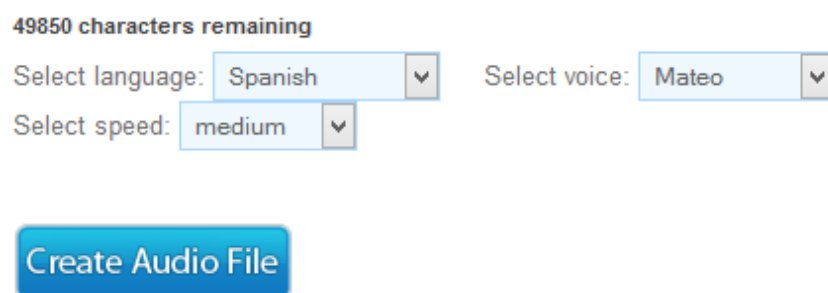


Figura 14. Características de la aplicación Text2Speech.

2.7 Metodología de Desarrollo del Sistema de Simulación

La definición del proceso de desarrollo de software es importante puesto que le proporciona una consistencia y estructura al conjunto de actividades necesarias para realizar el proyecto. Para el presente proyecto fue necesario adoptar una metodología que permita detallar el avance del proyecto mediante diagramas y plantillas para documentar el desarrollo. Se realizó una búsqueda de posibles metodologías y se encontró que el término de metodologías ágiles es el más adecuado para la realización de este proyecto que debe ser realizado por solo una persona. A continuación se resumen algunas características de las metodologías ágiles donde se busca privilegiar:

- A los individuos y sus interacciones frente a procesos y herramientas.
- Software en funcionamiento frente a documentación exhaustiva.
- Colaboración del cliente frente a una negociación de contrato.
- Respuesta al cambio frente a seguir un plan.

Una de las metodologías analizadas tiene el nombre de SUM para videojuegos y persigue como objetivo desarrollar videojuegos de calidad en tiempo y costo, así como la mejora continua del proceso para incrementar su eficacia y eficiencia. Pretende obtener resultados predecibles, administrar eficientemente los recursos y riesgos del proyecto, y lograr una alta productividad del equipo de desarrollo. SUM fue concebida para que se adapte a equipos multidisciplinarios pequeños (de tres a siete integrantes que trabajan en un mismo lugar físico o estén distribuidos), y para proyectos cortos (menores a un año de duración) con alto grado de participación del

cliente. Del análisis de la descripción de esta metodología se observa que no se ajusta al requerimiento de que no habrá participación más que de un estudiante encargado de llevar a feliz término el proyecto.

En vista de que para la realización del proyecto lo que se requiere es contar con herramientas que permitan realizarlo y que faciliten su documentación se ha encontrado que una posible herramienta que se ajusta al desarrollo es una metodología conocida como Huddle que ha generado una plantilla para documentar la fase de diseño del sistema de simulación.

Plantilla del Documento de Diseño

En la Tabla 5 se muestra la plantilla junto con el diseño desarrollado para el presente proyecto.

Tabla 5

Plantilla de Documento Huddle.

CAMPO	DESCRIPCIÓN
	CONCEPTO
TÍTULO	<p><i>El título del juego debe ser un nombre que capte la atención del jugador y del lector del documento. A grandes rasgos debe incluir el concepto del juego. El título debe ser algo memorable.</i></p> <p>El título que se ha escogido es "Save your Life" (Salva tu vida) en alusión a la gran cantidad de muertes que provocan los accidentes de tránsito. Se ha escogido el idioma inglés por la sensación que produce el producto extranjero de que es de mejor calidad</p>
DISEÑADOR	<p><i>El nombre del Diseñador del documento.</i></p> <p>Estudiante Carlos Proaño Valles</p>
GÉNERO	<p><i>El género abarca que tipo de juego será: Simulación, Rol, etc.</i></p> <p>En este caso el proyecto será de tipo simulación.</p>
PLATAFORMA	<p><i>Que hardware se requiere para jugarlo. Computadora personal, Xbox 360, PS3, etc.</i></p> <p>La plataforma para el juego "Save your life" será una computadora personal con Windows y sistema operativo de 32 bits.</p>
VERSION	<p><i>La versión del documento. Debe ser un número y no una fecha</i></p> <p>La versión es 1.0</p>

CAMPO	DESCRIPCIÓN
SINOPSIS DE LA JUGABILIDAD Y CONTENIDO	<p data-bbox="448 309 1323 342">CONCEPTO</p> <p data-bbox="448 353 1323 432"><i>En uno o dos párrafos escribir la esencia del juego. Incluir un poco del contenido que tendrá, historia, personajes, objetivo, etc.</i></p> <p data-bbox="448 443 1323 611">La esencia del juego es el respeto de las señales de tránsito y el aprendizaje de las normas de conducta de un peatón. En el juego se presentarán las señales de tránsito más comunes situadas en una ciudad junto con los edificios, iglesias, calles, luminarias, parques, puentes.</p> <p data-bbox="448 622 1323 981">El objetivo que se persigue es el aprendizaje en un contexto cercano a la realidad y el personaje principal de las escenas es el jugador que debe enfrentarse a la situación de manejo de un coche en un ambiente donde otros coches están circulando y respetar las leyes de tránsito esenciales como son el respeto a los semáforos, los pasos cebra, las señales de vías de un solo sentido y evitar los choques con los otros vehículos a partir del respeto de los límites de velocidad y respeto a la distancia mínima con los otros coches.</p> <p data-bbox="448 992 1323 1120">En el rol de peatón el objetivo es que el jugador reconozca las virtudes que hacen que se considere a una persona un buen ciudadano al respetar la norma más básica de comportamiento en la ciudad.</p>
CATEGORÍA	<p data-bbox="448 1137 1323 1216"><i>Comparar con uno o varios juegos existentes y enfatiza en las diferencias y características principales de este juego.</i></p> <p data-bbox="448 1227 1323 1579">La comparación se realiza con respecto a un videojuego en 2D producido por una empresa española MAPFRE que dispone en una página web un simulador para el aprendizaje de las normas de tránsito. Frente a ese sistema el presente proyecto tiene la ventaja significativa que genera un ambiente mucho más cercano a la realidad puesto que es un ambiente 3D. Por otro lado el mecanismo del juego en aquella aplicación no realiza un aprendizaje en el contexto de las situaciones que enfrenta un conductor en las calles lo que si se presenta en el proyecto presente</p>
LICENCIA	<p data-bbox="448 1597 1323 1675"><i>Describir si el juego está basado en un libro o en una película. Si es original se puede omitir este campo.</i></p> <p data-bbox="448 1686 1323 1765">El juego está basado en la realidad y trata de simularla en la forma más aproximada posible con la herramienta de desarrollo utilizada</p>

CAMPO	DESCRIPCIÓN
	CONCEPTO
MECÁNICA	<p><i>Describir la jugabilidad y el control del juego. ¿Qué hace el jugador?. ¿Qué usa para lograr sus objetivos?</i></p> <p>El juego se controla mediante el teclado de la computadora y los botones de avanzar y retroceder y también se puede interconectar un volante comercial. El jugador debe conducir un coche por una ciudad que es inicialmente desconocida, pero que con el avance del juego se puede llegar a reconocer todos los elementos que la componen. La ciudad tiene aproximadamente 12 ejes longitudinales y 16 ejes transversales para simular un ambiente complejo y que se produzcan muchas situaciones de conflicto. Lo importante es que el jugador cumpla las leyes de tránsito, el programa penaliza tres tipos de infracciones: irrespeto a los semáforos, señales de una sola vía y exceso de velocidad. El jugador inicia con 900 puntos y por cada infracción que comete se le restan puntos. El objetivo es que termine su recorrido con la mayor cantidad de puntos y la mayor cantidad de señales de tránsito aprendidas.</p>
TECNOLOGÍA	<p><i>Enlistar que Hardware y software se requiere para producir el juego. Desde lenguaje de programación hasta editor de sonidos.</i></p> <p>El juego ha sido desarrollado en un sistema operativo de 64 bits mediante el programa Unity versión 2018.2.18f1. Los sonidos que se reproducen son de tipo wav y las imágenes de las señales de tránsito son de tipo png.</p>
PÚBLICO	<p><i>¿A quién va dirigido el juego? ¿Quién lo jugará? Tipo de jugador casual, competitivo.</i></p> <p>El juego está dirigido a niños de 10 a 12 años y el juego es competitivo.</p>

CAPÍTULO III

ANÁLISIS Y DISEÑO

3.1 Requerimientos Funcionales y No Funcionales

3.1.1 Requerimientos Funcionales

Un requerimiento funcional define una función del sistema de software o sus componentes. Una función es descrita como un conjunto de entradas, comportamientos y salidas. Los requerimientos funcionales pueden ser: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que se supone, un sistema debe cumplir. Los requisitos de comportamiento para cada requerimiento funcional se muestran en los casos de uso. Son complementados por los requerimientos no funcionales, que se enfocan en cambio en el diseño o la implementación (Eugene. Wiegers, 2003).

Requerimiento funcional 1: Ingreso a la aplicación

En la Tabla 6 se describe las características del requerimiento funcional de ingreso a la aplicación.

Tabla 6

Descripción requerimiento funcional 1: Ingreso a la aplicación.

Nombre	Ingresar a la aplicación
Descripción	Permitirá al estudiante ingresar a la aplicación
Salidas	Muestra la pantalla inicial de la aplicación.
Proceso	Dar click en el botón <i>Play</i> .
Post condiciones	Se muestra la pantalla de los roles de la aplicación
Prioridad	Media
Rol que lo ejecuta	Estudiante

Requerimiento funcional 2: Rol conductor

En la Tabla 7 se describen las características del requerimiento funcional del rol conductor.

Tabla 7

Descripción requerimiento funcional 2: Rol conductor.

Nombre	Seleccionar el rol de conductor
Descripción	Permitirá al estudiante seleccionar el rol de conductor.
Salidas	Se muestra la pantalla para el rol de conductor.
Proceso	Dar click en el botón <i>Conductor</i> .
Precondiciones	Haber ingresado a la aplicación como estudiante.
Post condiciones	Se mostrará la pantalla del rol de conductor.
Prioridad	Alta
Rol que lo ejecuta	Estudiante

Requerimiento funcional 3: Rol peatón

En la Tabla 8 se describen las características del requerimiento funcional del rol peatón.

Tabla 8

Descripción requerimiento funcional 3: Rol peatón.

Nombre	Seleccionar el rol de peatón
Descripción	Permitirá al estudiante seleccionar el rol de peatón.
Salidas	Se muestra la pantalla para el rol de peatón.
Proceso	Dar click en el botón <i>Peatón</i> .
Precondiciones	Haber ingresado a la aplicación como estudiante.
Post condiciones	Se mostrará la pantalla del rol de peatón.
Prioridad	Alta
Rol que lo ejecuta	Estudiante

Requerimiento funcional 4: Rol espectador

En la Tabla 9 se describen las características del requerimiento funcional del rol peatón.

Tabla 9

Descripción requerimiento funcional 4: Rol espectador.

Nombre	Seleccionar el rol de espectador
Descripción	Permitirá al estudiante seleccionar el rol de espectador.
Salidas	Se muestra la pantalla para el rol de espectador.
Proceso	Dar click en el botón espectador.
Precondiciones	Haber ingresado a la aplicación como estudiante.

Post condiciones	Se mostrará la pantalla del rol de espectador.
Prioridad	Alta
Rol que lo ejecuta	Estudiante

Requerimiento funcional 5: Seleccionar la ruta

En la Tabla 10 se describen las características del requerimiento funcional de Seleccionar la ruta.

Tabla 10

Descripción requerimiento funcional 5: Seleccionar la ruta.

Nombre	Seleccionar la ruta
Descripción	Permitirá al estudiante escoger la ruta en la que desea cumplir la actividad
Salidas	Se selecciona la actividad y la ruta
Proceso	Seleccionar la ruta a realizar.
Precondiciones	1. Haber ingresado al sistema como estudiante. 2. Haber seleccionado el rol de conductor.
Post condiciones	Se seleccionará la actividad y la ruta
Prioridad	Alta
Rol que lo ejecuta	Estudiante

Requerimiento funcional 6: Seleccionar el personaje

En la Tabla 11 se describen las características del requerimiento funcional de Seleccionar el personaje.

Tabla 11

Descripción requerimiento funcional 6: Seleccionar el personaje.

Nombre	Seleccionar el personaje
Descripción	Permitirá al estudiante escoger el personaje con el cual desea cumplir la actividad
Salidas	Se selecciona la actividad y el personaje
Proceso	Seleccionar el personaje.
Precondiciones	1. Haber ingresado al sistema como estudiante. 2. Haber seleccionado el rol de peatón.
Post condiciones	Se seleccionará la actividad y el personaje
Prioridad	Alta
Rol que lo ejecuta	Estudiante

Requerimiento funcional 7: Realizar la actividad

En la Tabla 12 se describen las características del requerimiento funcional de realizar la actividad: conducir, caminar o visualizar.

Tabla 12

Descripción requerimiento funcional 7: Realizar la actividad.

Nombre	Realizar la actividad.
Descripción	Permitirá al estudiante realizar las actividades según el rol escogido.
Entradas	Volante, joystick, pedales o mouse.
Salidas	Conducción, caminata o visualización de la aplicación.
Proceso	Conducir el vehículo. Caminar por la ciudad Visualizar la ciudad
Precondiciones	<ol style="list-style-type: none"> 1. Haber ingresado al sistema como estudiante. 2. Haber seleccionado el rol de conductor o peatón y haber dado click en <i>Play</i>. 3. Haber seleccionado el rol de espectador.
Post condiciones	<ol style="list-style-type: none"> 1. Si se cumple la actividad y llega a la meta se mostrará una pantalla para ingresar el nombre del estudiante. 2. Si no se cumple la actividad (no llega a la meta) se mostrará una pantalla de <i>Game Over</i>.
Prioridad	Alta
Rol que lo ejecuta	Estudiante

Requerimiento funcional 8: Guardar puntaje

En la Tabla 13 se describen las características del requerimiento funcional de guardar puntaje.

Tabla 13

Descripción requerimiento funcional 8: Guardar puntaje.

Nombre	Ingresar nombre del estudiante
Descripción	Permitirá al estudiante colocar su nombre para guardar su puntaje
Salidas	Se muestra la tabla de puntajes
Proceso	Ingresar nombre del estudiante
Precondiciones	<ol style="list-style-type: none"> 1. Haber ingresado al sistema como estudiante. 2. Haber seleccionado el rol de conductor y

	haber dado click en <i>Play</i> . 3. Haber culminado la actividad (llegar a la meta).
Post condiciones	Se mostrará la tabla de puntajes obtenidos
Prioridad	Alta
Rol que lo ejecuta	Estudiante

3.1.2 Requerimientos No Funcionales

Un requisito no funcional o atributo de calidad es, en la ingeniería de sistemas y la ingeniería de software, un requisito que sabe bien y especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, ya que éstos corresponden a los requisitos funcionales. Por tanto, se refieren a todos los requisitos que no describen información a guardar, ni funciones a realizar, sino características de funcionamiento, por eso suelen denominarse Atributos de calidad de un sistema.

Queda entonces el requisito no funcional, que son las restricciones o condiciones que impone el cliente al programa que necesita, por ejemplo el tiempo de entrega del programa, el lenguaje o la cantidad de usuarios (Stellman & Greene, 2005).

A) Requerimientos de Interfaz

- La aplicación tendrá una interfaz accesible para el usuario, además de tener muy claro cuáles son los procedimientos y funciones que realizara el sistema.
- La aplicación tendrá una interfaz amigable para guiarle al niño en la consecución de la actividad.

B) Requerimientos de Navegación

- La aplicación le permitirá al usuario comprender las actividades que debe cumplir para lograr llegar al objetivo.
- La aplicación tendrá botones que le indican los pasos a seguir para lograr cumplir una actividad.

C) Requerimientos de usabilidad

- La aplicación contará con una interfaz atractiva, simple y de fácil uso para los usuarios.

- La aplicación mostrará mensajes de información o de error antes de realizar cualquier procedimiento.
- La aplicación le enseñará al niño las definiciones de cada una de las reglas de tránsito mientras se encuentra realizando la actividad.

D) Requerimientos de escalabilidad

- La aplicación estará en capacidad de realizar cambios e implementar funcionalidades en el futuro.

E) Requerimientos operacionales

- La aplicación contará con vehículos y peatones automatizados que se desplazarán por la ciudad.
- La aplicación contará con señales de tránsito ubicadas en puntos donde se le guía al niño para comprender lo que debe realizar.

F) Requerimientos de seguridad

- La aplicación mantendrá un grado de confidencialidad de los datos de cada usuario que sea registrado.
- La aplicación mantendrá un grado de confidencialidad de los reportes de cada calificación, fallos y puntajes ingresados.
- La aplicación manejará de manera íntegra la información almacenada.

G) Requerimientos de hardware

- Para desarrollar e implementar la aplicación es necesario contar con un computador que tenga las siguientes características:
 - Sistema Operativo Windows 7, 8, 10.
 - Memoria RAM de por lo menos 12GB
 - Disponibilidad en disco de 100 GB.

3.2 Diagrama de Casos de Uso

Los diagramas de caso de uso son uno de los tipos de diagramas en UML para modelar aspectos dinámicos de sistemas. Los diagramas de casos de uso son

importantes para modelar el comportamiento de un sistema, un subsistema o una clase. Cada uno muestra un conjunto de casos de uso, actores y sus relaciones (Calleros, 2019).

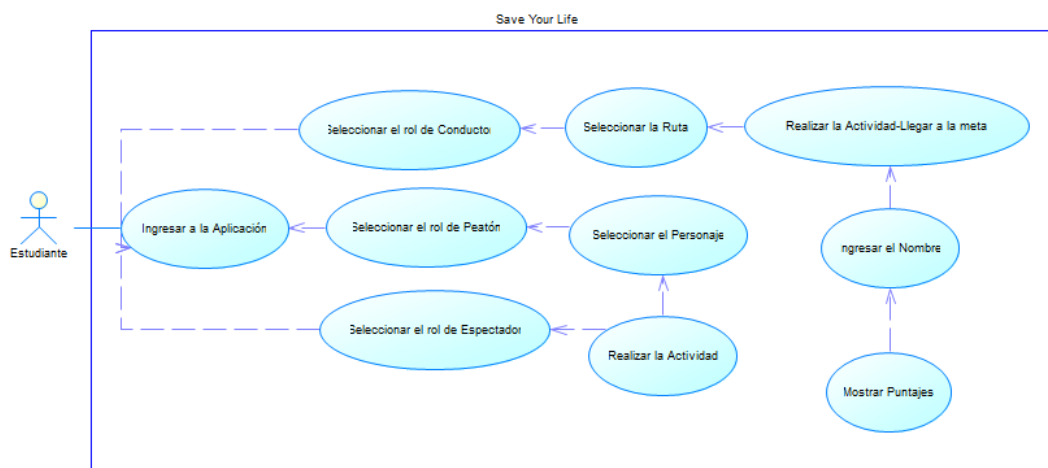


Figura 15. Diagrama de Casos de Uso.

3.3 Diagramas de Secuencia

El diagrama de secuencia es un tipo de diagrama de interacción cuyo objetivo es describir el comportamiento dinámico del sistema de información haciendo énfasis en la secuencia de los mensajes intercambiados por los objetos (Calleros, 2019).

3.3.1 Ingresar a la Aplicación

En la Figura 16 se muestra el diagrama de secuencia de ingreso a la aplicación desde la pantalla inicial.

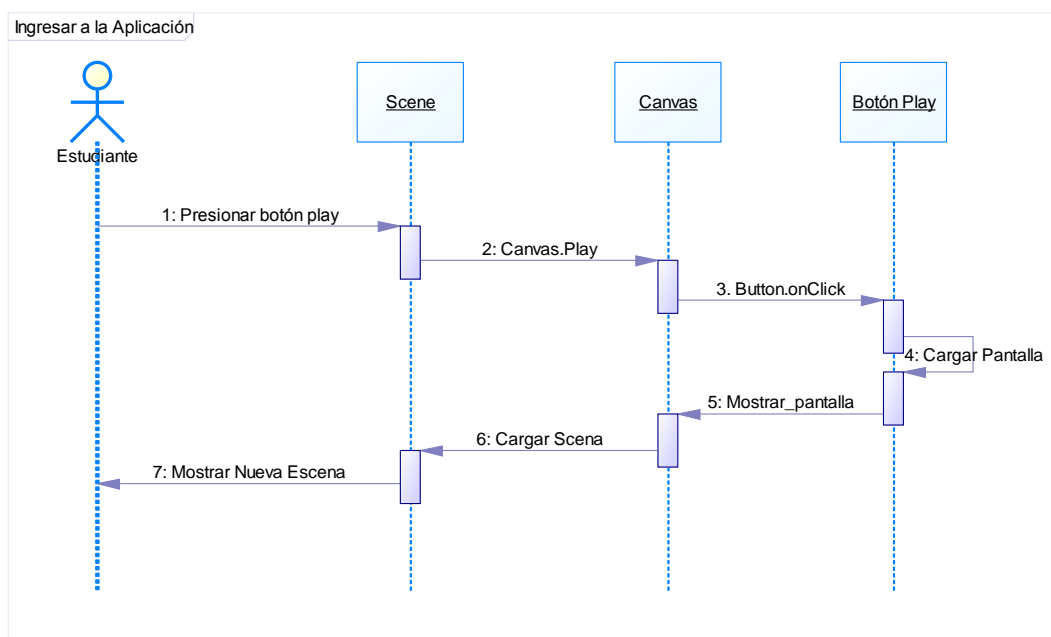


Figura 16. Diagrama de Secuencia: Ingresar a la aplicación.

3.3.2 Seleccionar roles

En la Figura 17 se muestra el diagrama de secuencia para escoger el rol conductor en la pantalla de selección de roles, en la Figura 18 aparece el diagrama de secuencia para escoger el rol de peatón y en la Figura 19 se presenta el diagrama de secuencia para escoger el rol espectador.

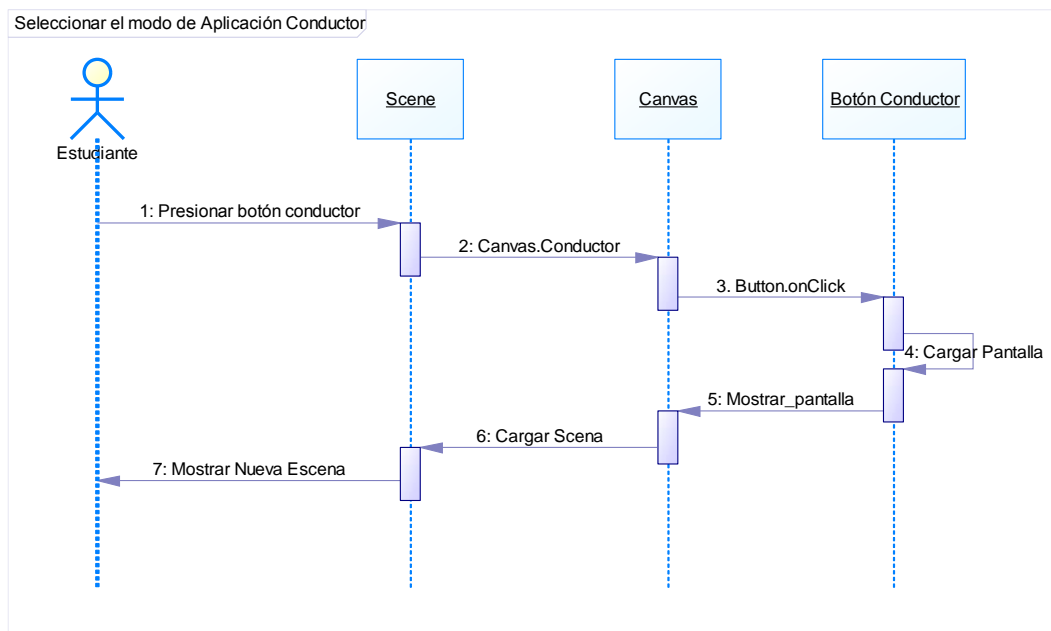


Figura 17. Diagrama de Secuencia: Seleccionar el modo de aplicación conductor.

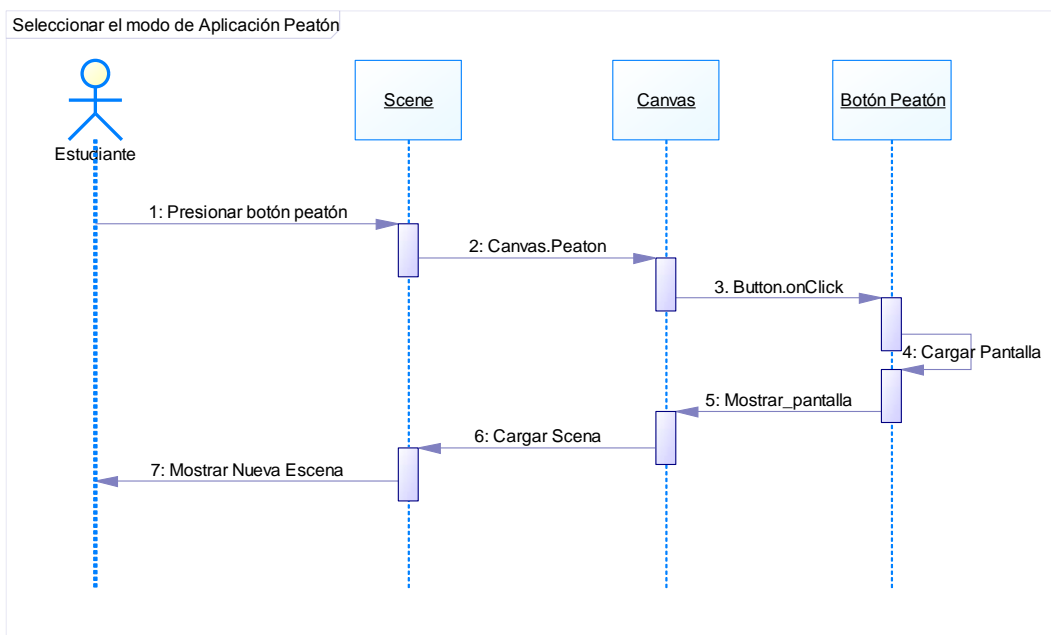


Figura 18. Diagrama de Secuencia: Seleccionar el modo de aplicación peatón.

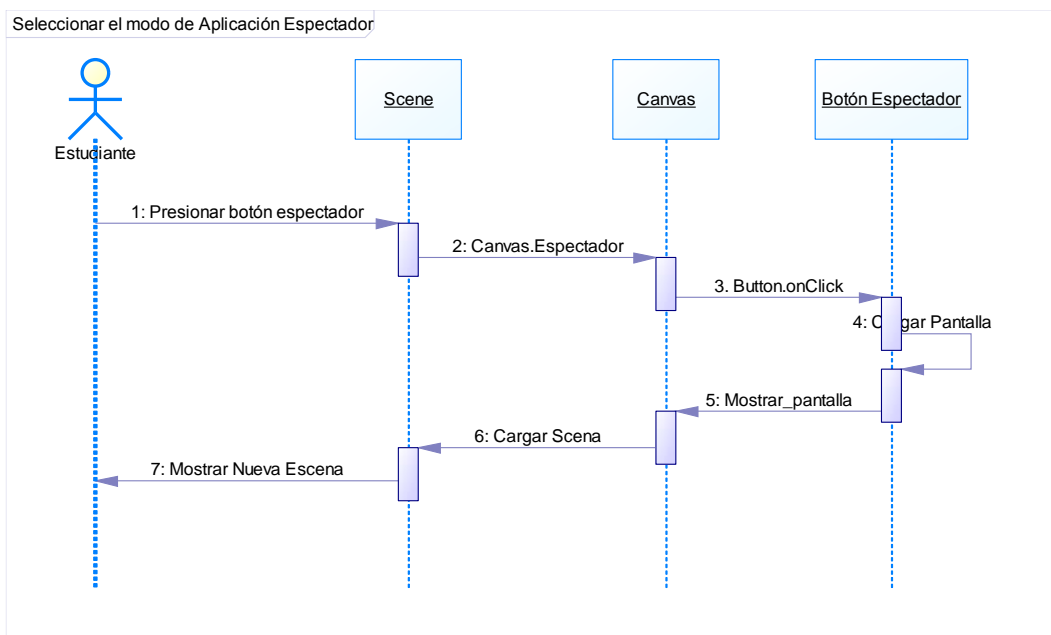


Figura 19. Diagrama de Secuencia: Seleccionar el modo de aplicación espectador.

3.3.3 Seleccionar la ruta

En la Figura 20 se muestra el diagrama de secuencia para seleccionar la ruta en el rol de conductor.

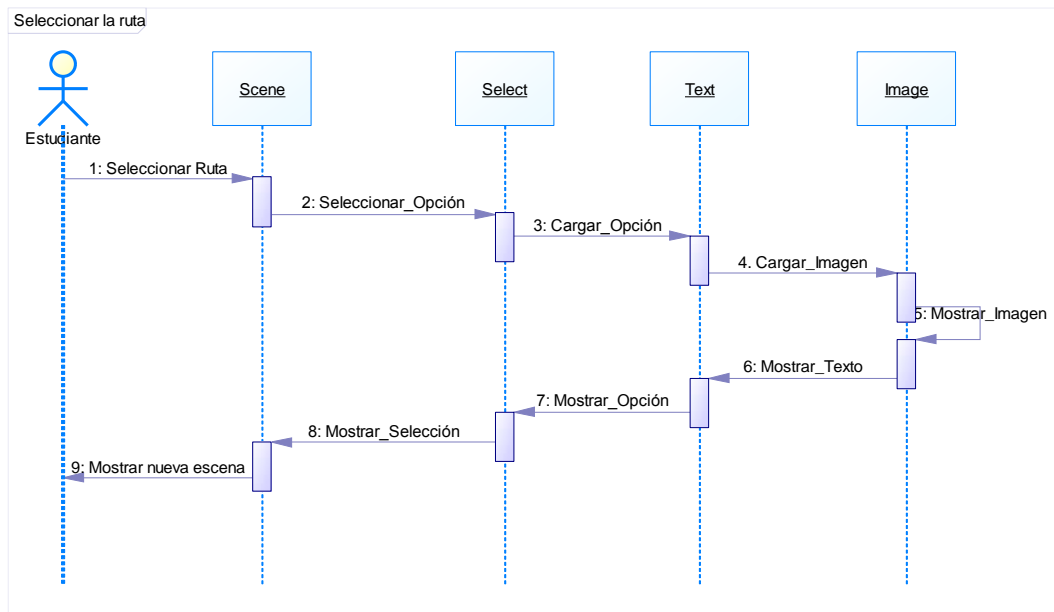


Figura 20. Diagrama de Secuencia: Seleccionar la ruta.

3.3.4 Seleccionar el Personaje

En la Figura 21 se muestra el diagrama de secuencia para seleccionar el personaje en el rol de peatón.

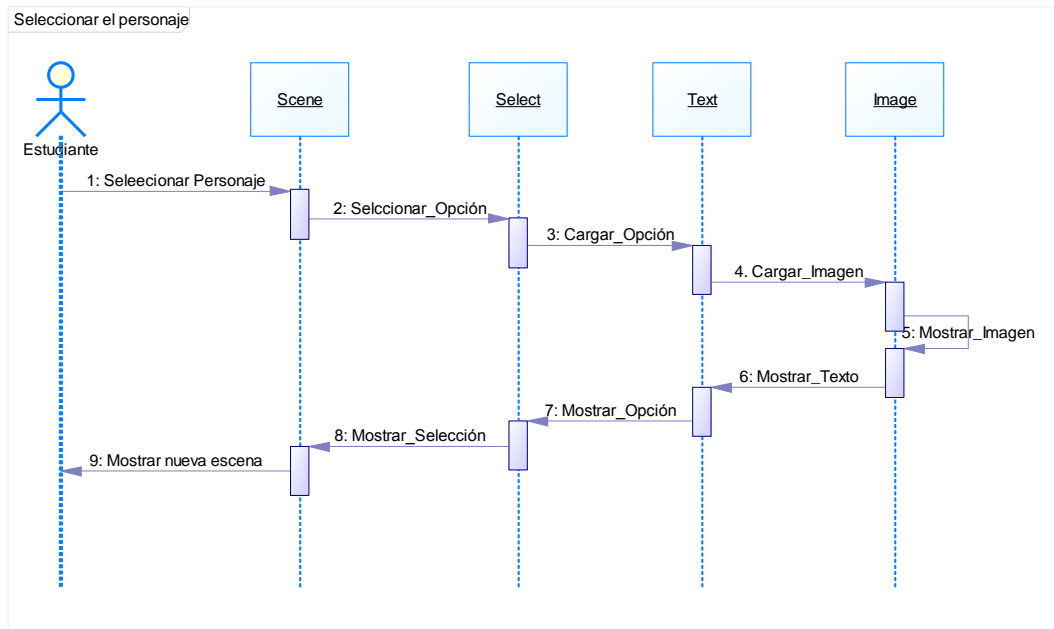


Figura 21. Diagrama de Secuencia: Seleccionar el personaje.

3.3.7 Realizar la actividad

Rol Conductor-Alcanza la meta

En la Figura 22 se muestra el diagrama de secuencia cuando el estudiante realiza la actividad completa y alcanza la meta.

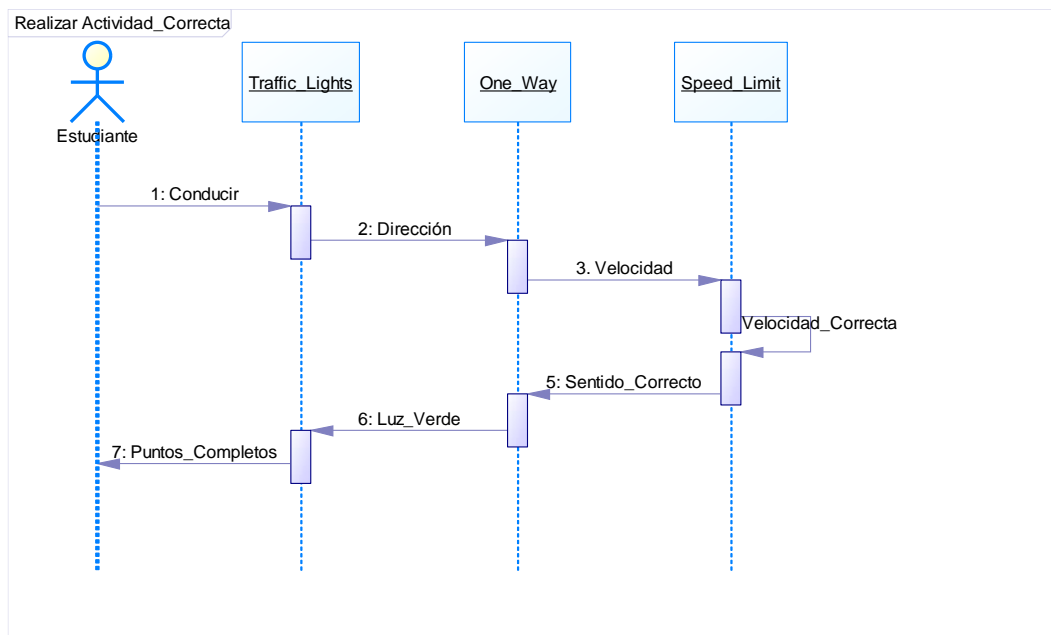


Figura 22. Diagrama de Secuencia: Rol Conductor-Alcanza la meta.

Rol Conductor-Infracciones

En la Figura 23 se muestra el diagrama de secuencia cuando se realiza la actividad en el rol de conductor y comete la infracción de pasar un semáforo en rojo, en la Figura 24 se presenta el diagrama de secuencia cuando se realiza la actividad de rol de conductor y comete la infracción de ir en contravía y en la Figura 25 se presenta el diagrama de secuencia cuando se realiza la actividad de rol de conductor y comete la infracción de superar la velocidad permitida

Semáforo Rojo

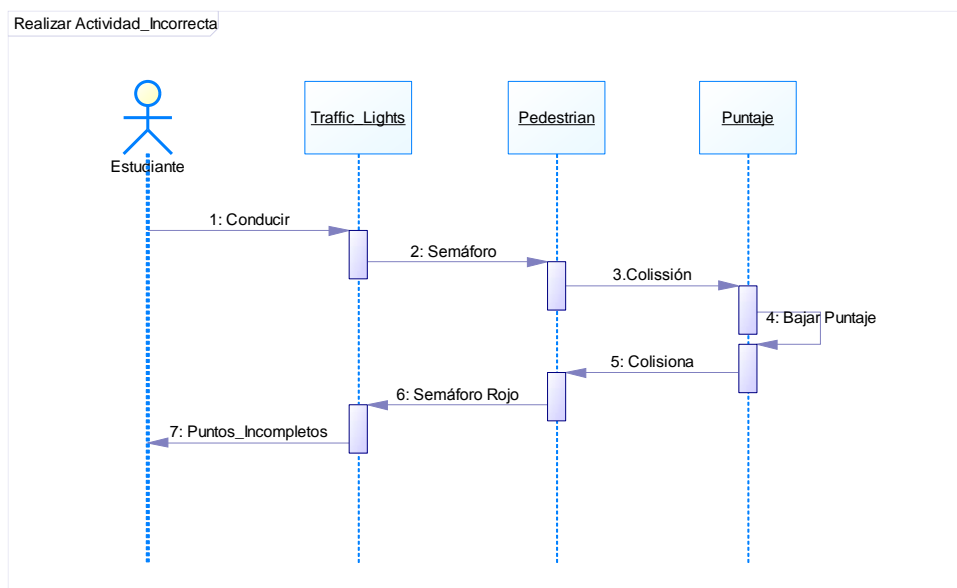


Figura 23. Diagrama de Secuencia: Rol Conductor-Infracción-Semáforo Rojo.

Contravía

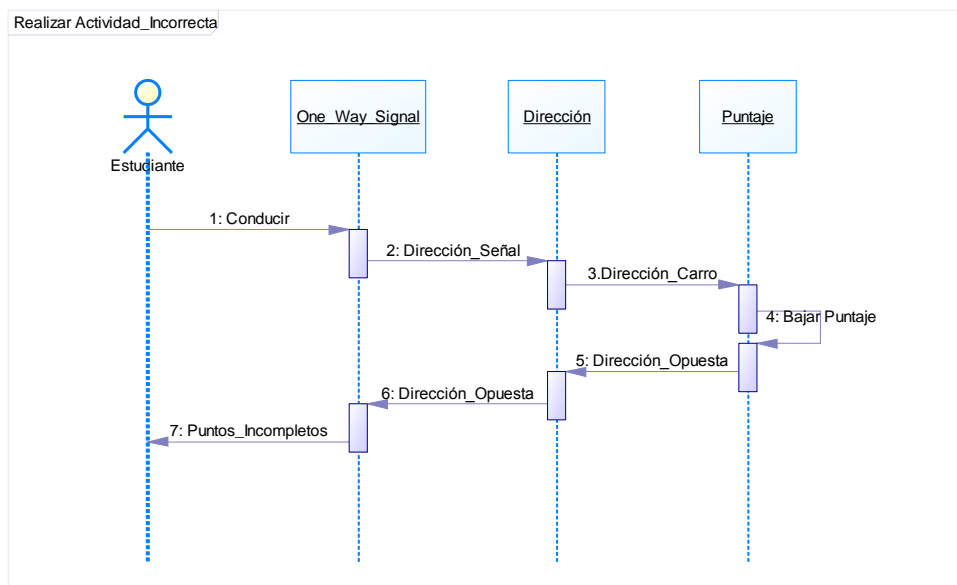


Figura 24. Diagrama de Secuencia: Rol Conductor-Infracción-Contravía.

Exceso de Velocidad

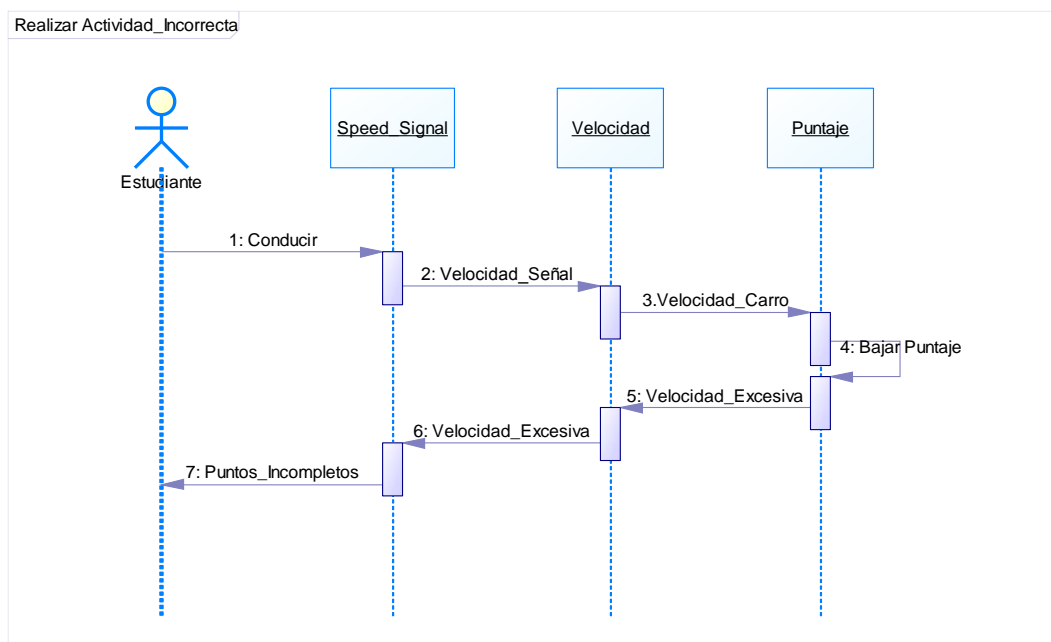


Figura 25. Diagrama de Secuencia: Rol Conductor-Infracción-Exceso de Velocidad.

Rol Peatón

En la Figura 26 se presenta el diagrama de secuencia para realizar la actividad de peatón donde se traslada por la ciudad observando las señales de tránsito que se le presentan.

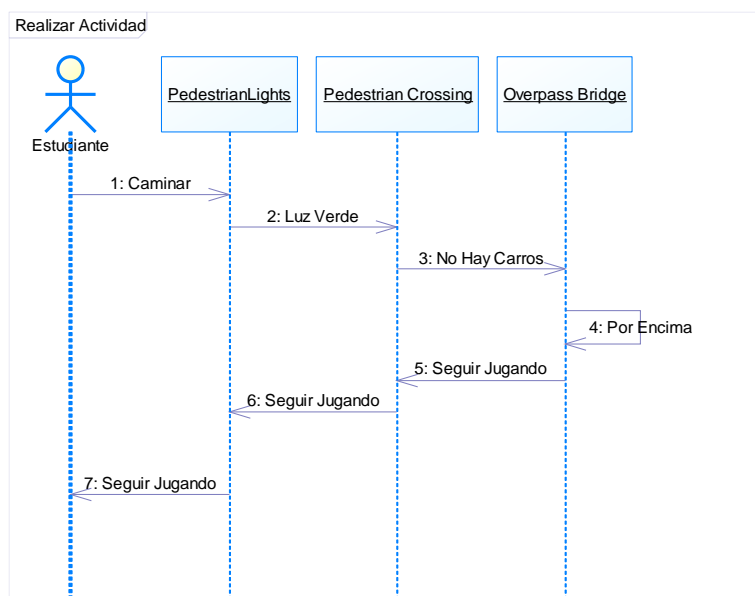


Figura 26. Diagrama de Secuencia: Realizar la actividad: Rol Peatón.

Rol Espectador

En la Figura 27 se presenta el diagrama de secuencia para realizar la actividad de espectador donde se selecciona el vehículo a visualizar mientras se traslada por la ciudad.

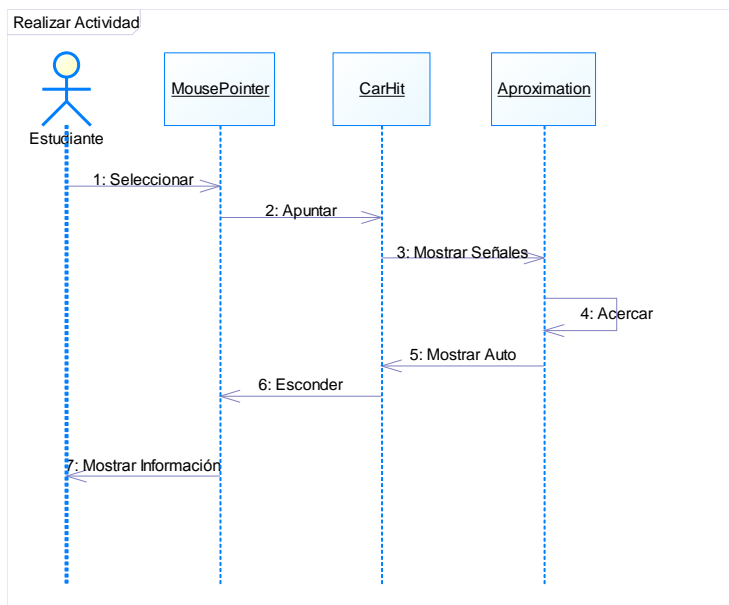


Figura 27. Diagrama de Secuencia: Realizar la actividad: Rol espectador.

3.3.8 Ingresar Nombre del Estudiante

En la Figura 28 se indica el diagrama de secuencia en caso de que el estudiante haya alcanzado la meta y puede almacenar su puntaje.

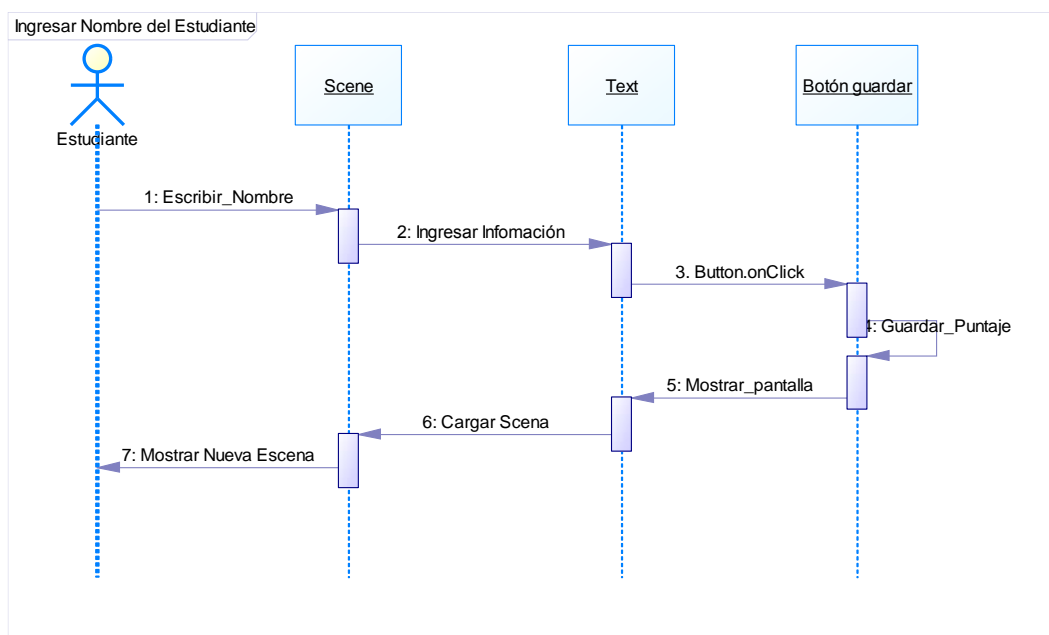


Figura 28. Diagrama de Secuencia: Ingresar Nombre del Estudiante.

3.4 Diagramas de Clases y Componentes

3.4.1 Diagramas de Clases

Los diagramas de clases son uno de los tipos de diagramas más útiles en UML, ya que trazan claramente la estructura de un sistema concreto al modelar sus clases, atributos, operaciones y relaciones entre objetos (Calleros, 2019).

Rol Peatón

En la Figura 29 se presenta el diagrama de clases del rol de Peatón en el cual se encuentran las clases *ThirdPersonUser* y *ThirdPersonCharacter* que son las que le permiten al personaje desplazarse por la escena. Estas clases se obtienen como paquetes importados desde *Unity* y que se los conoce como *Standard Assets*, además de una clase *AudiolImage* que se coloca en cada señal para que se reproduzca la información de la misma y *Minimap* que permite mostrar un mapa para visualizar el entorno cercano al personaje.

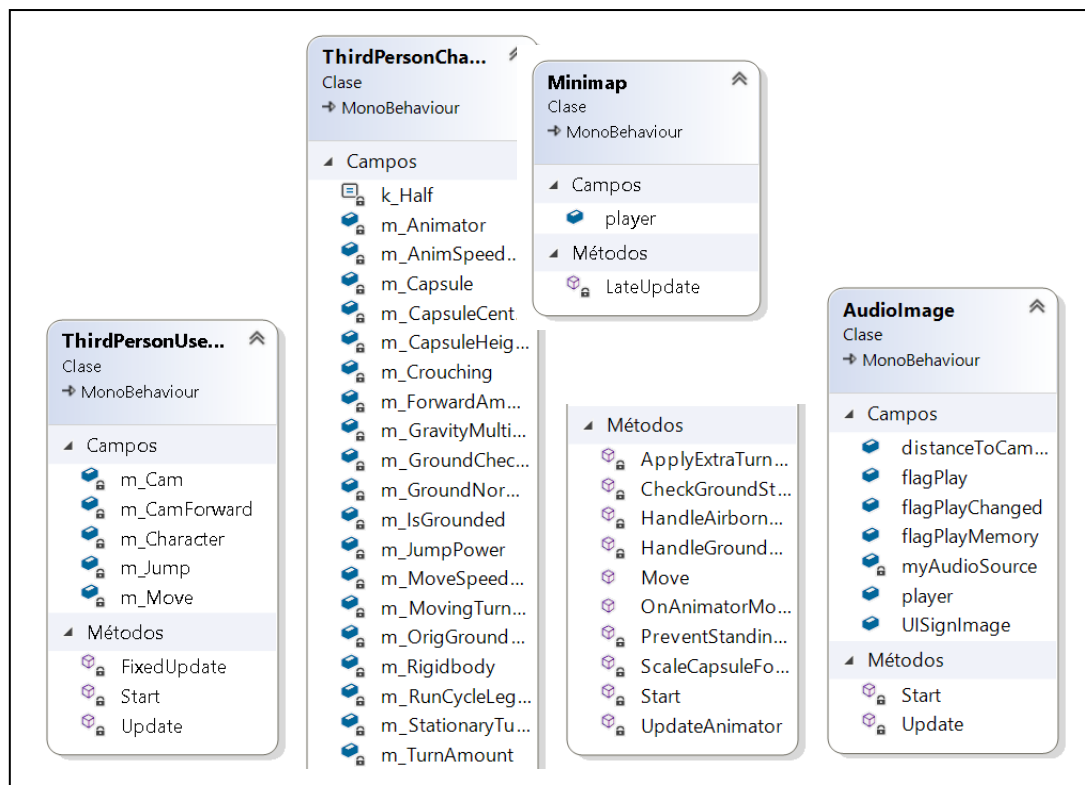


Figura 29. Diagrama de Clases del Rol de Peatón.

Rol Conductor

En la Figura 30 se detalla el diagrama de clases del Rol de Conductor formado por las clases: *CarController* para la conducción del coche; *InfractionsMonitor* para contabilizar las infracciones que comete, *LookSigns* que permite visualizar la imagen y reproducir el audio de la señal de tránsito por la que atraviesa el coche y *Minimap* que permite mostrar un mapa para visualizar el entorno cercano al coche.

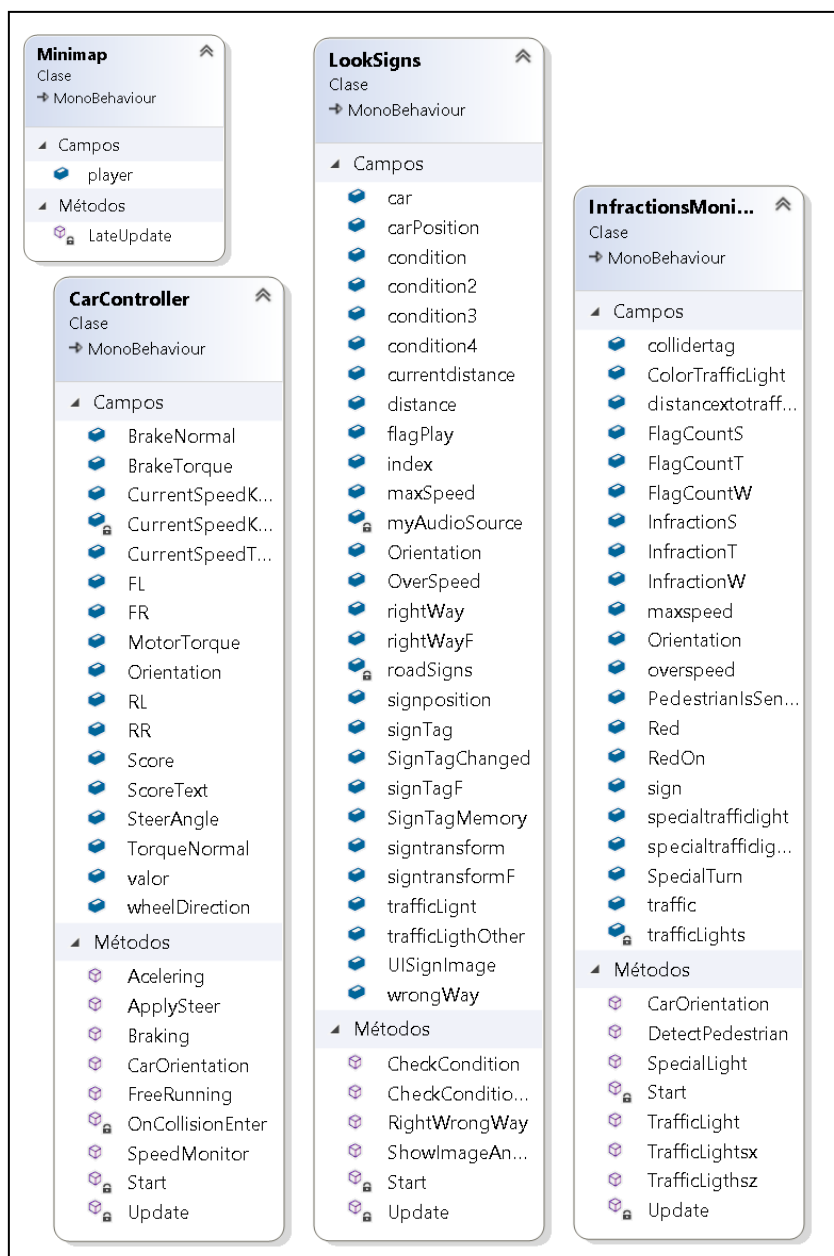


Figura 30. Diagrama de Clases del Rol de Conductor.

Autos Automáticos

En la figura 31 se muestra la clase *CarAi* la cual posee todos los mecanismos para la simulación de los vehículos automáticos.

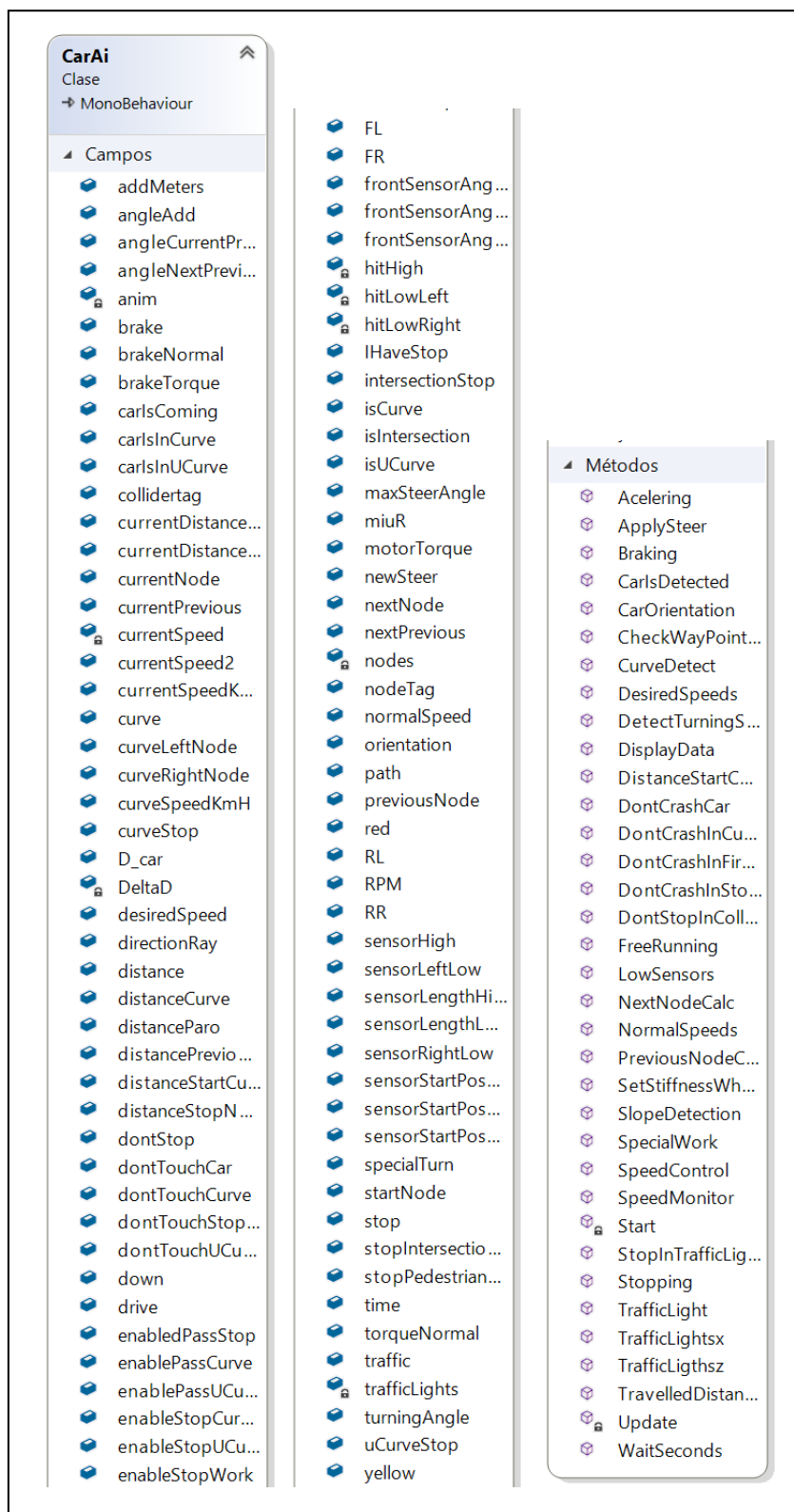


Figura 31. Diagrama de Clases CarAi.

En la figura 32 se muestra las clases *LookSpeedSings* la cual brinda al carro automático las velocidades a las que debe circular, y las clases del *Animator* las cuales controlan las acciones del vehículo.

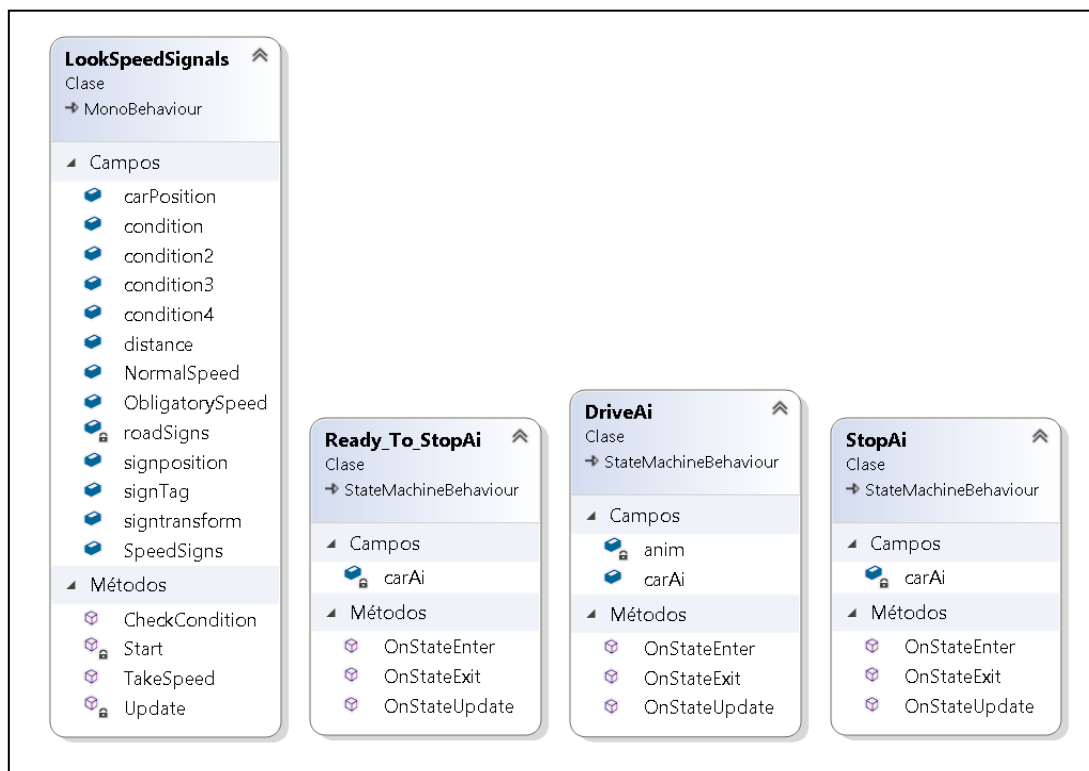


Figura 32. Diagrama de Clases de Autos Automáticos.

Peatones Automáticos

En la Figura 33 se indica el Diagrama de Clases para los peatones automáticos PNJ (Personajes no jugadores).

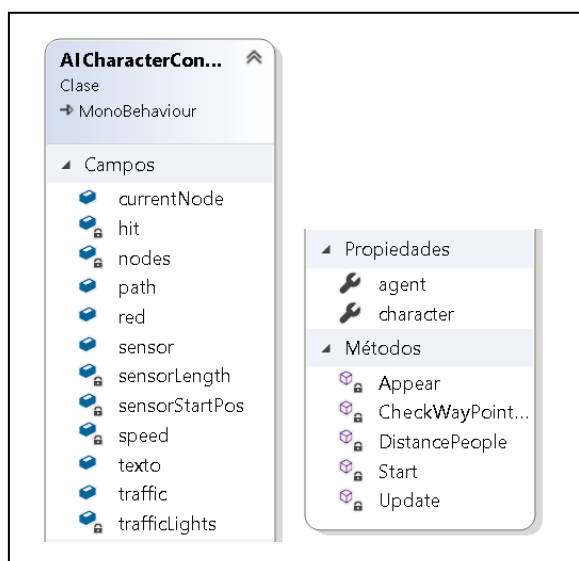


Figura 33. Diagrama de Clases de Peatones Automáticos.

Semáforos

En la Figura 34 se indica el Diagrama de Clases para los tres tipos de Semáforos utilizados en la aplicación.

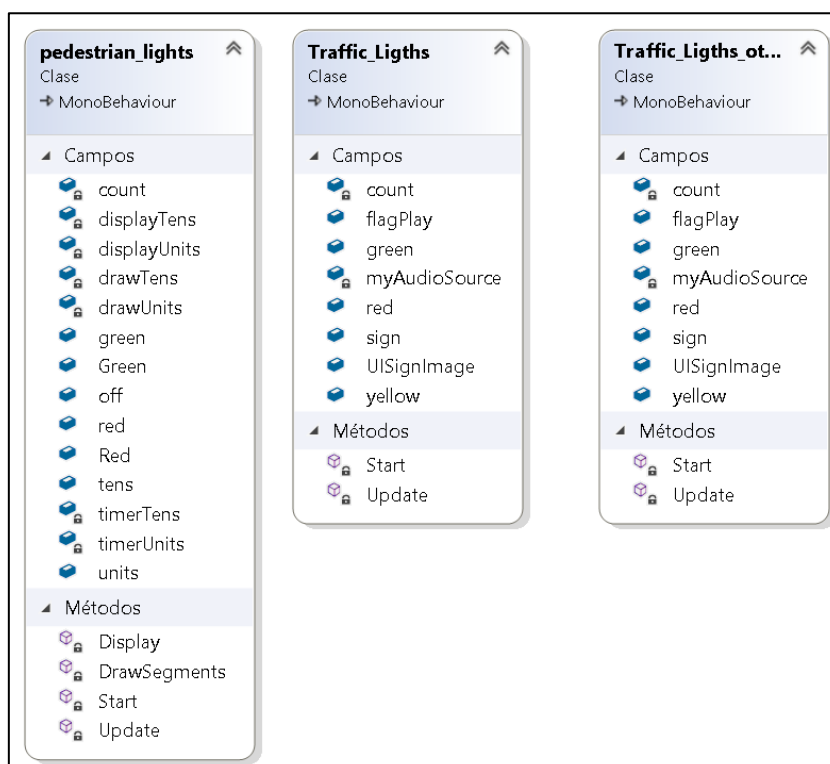


Figura 34. Diagrama de Clases Semáforos.

Cámara Espectador

En la Figura 35 se indica el Diagrama de Clases para el control de la Cámara en el rol de Espectador y la selección del vehículo a visualizar.

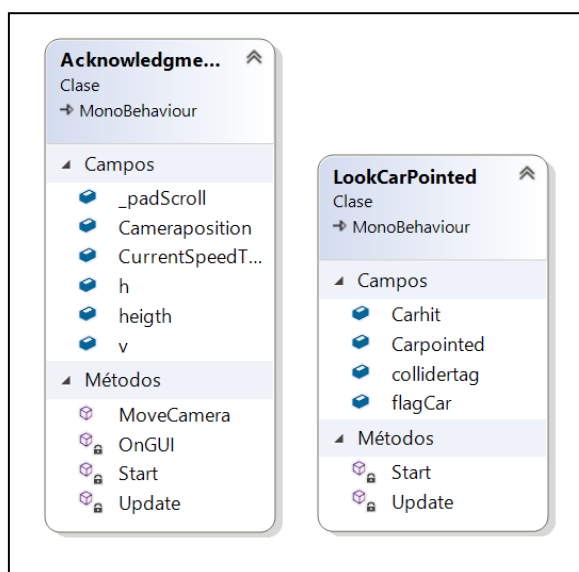


Figura 35. Diagrama de clases del Rol de Espectador.

Guardar Puntaje

En la Figura 36 se muestran las clases para el guardado de los puntajes, su almacenamiento y la visualización de los mejores puntajes.

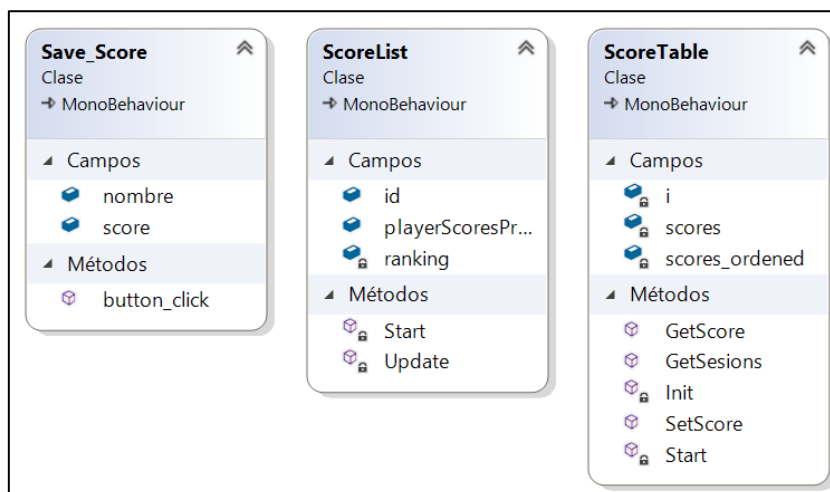


Figura 36. Diagrama de Clases para Guardar el Puntaje.

Navegación

En la Figura 37 se encuentran las clases para la navegación entre pantallas de la aplicación.

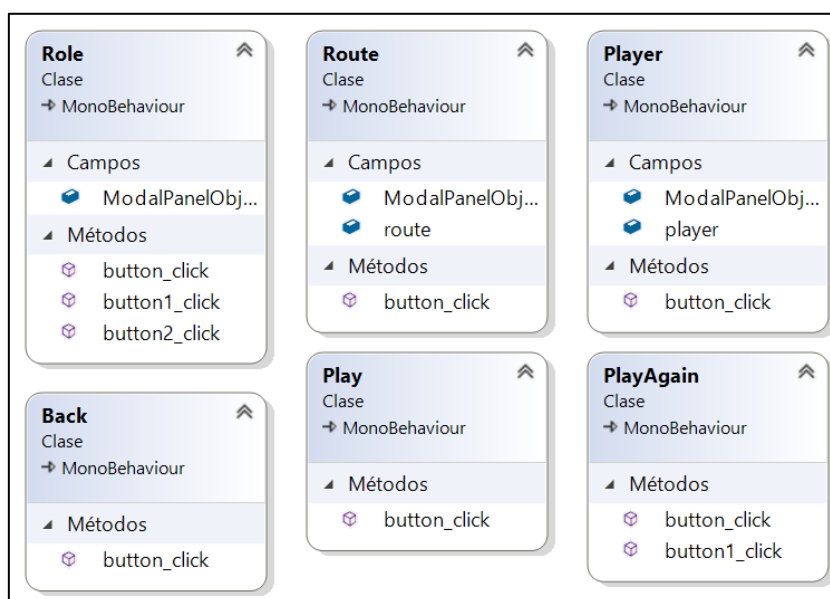


Figura 37. Diagramas de clases de Navegación.

3.4.2 Diagrama de Componentes

El diagrama de componentes proporciona una visión física de la construcción del sistema de información. Muestra la organización de los componentes software, sus interfaces y las dependencias entre ellos. En la Figura 38 se indica el diagrama de Componentes de la aplicación desarrollada.

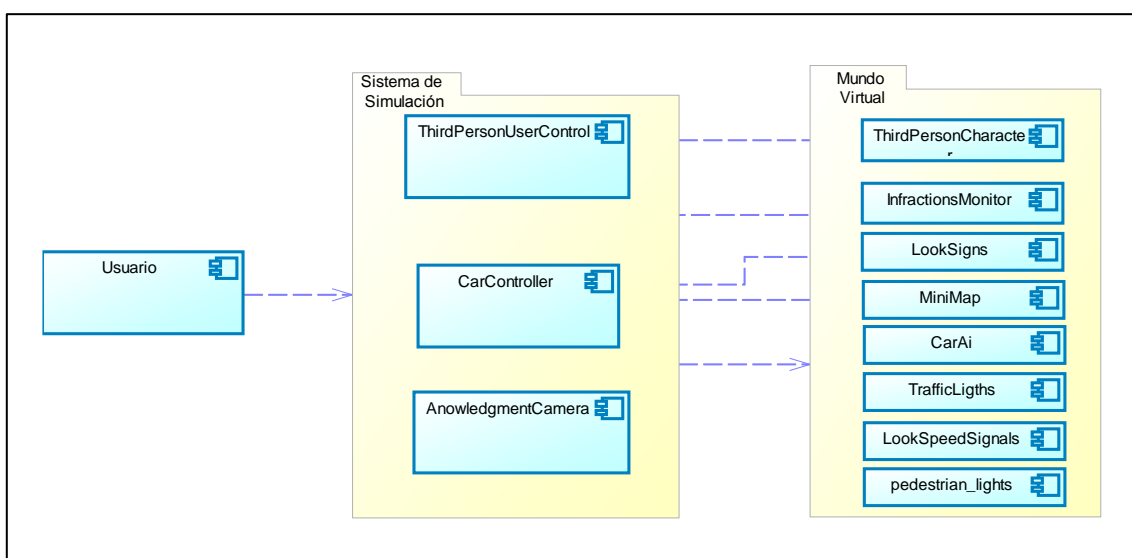


Figura 38. Diagrama de Componentes de la Aplicación.

CAPÍTULO IV

DESARROLLO E IMPLEMENTACIÓN

4.1 El entorno del sistema de simulación

Para que la aplicación simule lo mejor posible las situaciones que se presentan en la realidad es necesario disponer de un entorno que reproduzca las características que tiene un sistema de tránsito. Se ha considerado los siguientes elementos para conformar el entorno: la ciudad, los coches automáticos, los peatones, los semáforos y las señales de tránsito.

4.1.1 La ciudad

Para el entorno de la aplicación se utilizó el asset “SimplePolyCity” el cual brinda, los modelos utilizados para cada lugar de la ciudad con sus materiales.

La escena inicial se presentó sin ningún tipo de material como se muestra a continuación en la Figura 39.

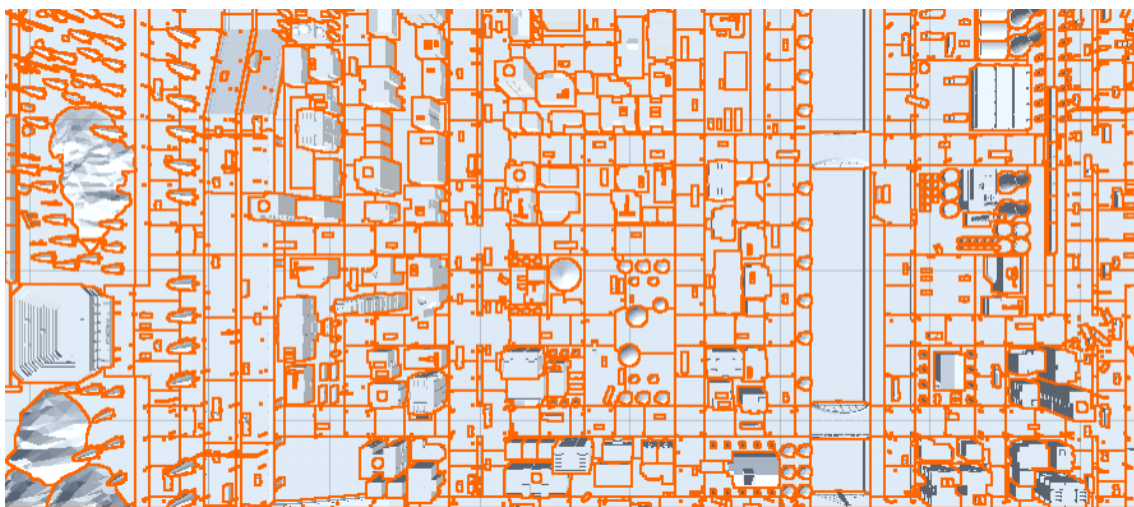


Figura 39. Ciudad Despintada.

A continuación se extrajo todos los materiales del modelo de la siguiente forma, como se muestra en la Figura 40.

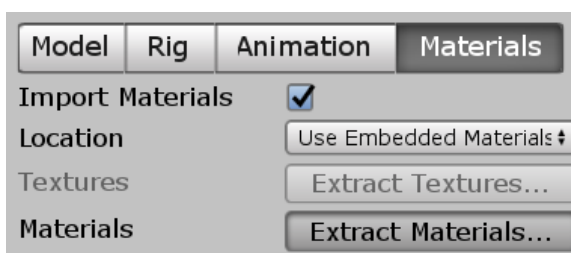


Figura 40. Extracción Materiales Ciudad.

Una vez extraído todos los materiales se procedió a colocar la textura a cada uno de ellos como se muestra en la Figura 41.

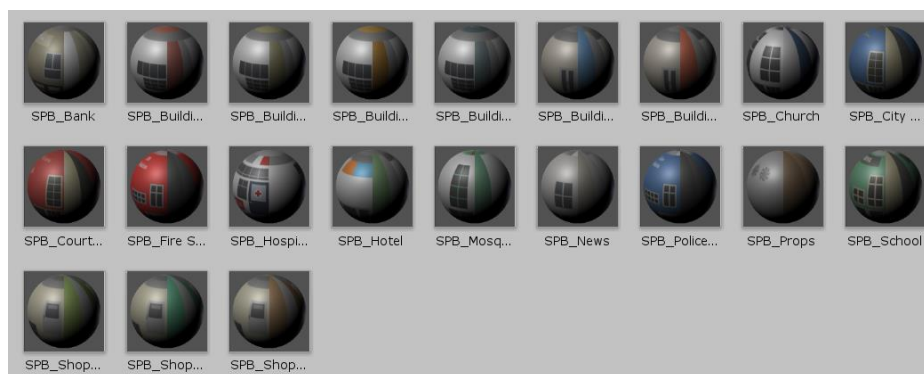


Figura 41. Materiales Ciudad Pintados.

Una vez colocados los materiales, la ciudad tomó color como se muestra en la Figura 42.



Figura 42. Ciudad Pintada.

Para proveer un efecto de familiaridad con la aplicación, se colocó nombres a las edificaciones con denominaciones similares a las de los sitios que se encuentran en la parroquia de Tumbaco, en el Cantón Quito donde se encuentra localizada la escuela. Los edificios escogidos para denominarlos fueron la estación de bomberos, el departamento de policía, el hospital, el juzgado y la escuela. Allí se colocó los nombres de Cuerpo de Bomberos del Distrito Metropolitano de Quito, UPC, Clínica de Especialidades Tumbaco, Consejo de la Judicatura y la escuela Víctor Manuel Peñaherrera. Para realizar esta tarea se cambió las texturas de los objetos. A continuación se muestra las texturas originales, como se puede ver en la Figura 43.

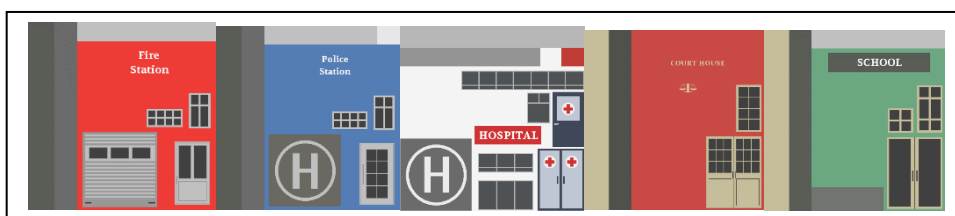


Figura 43. Texturas edificios originales.

Los textos de las imágenes originales eran extraños porque se encontraban en el idioma inglés. Las nuevas texturas muestran los nuevos diseños.

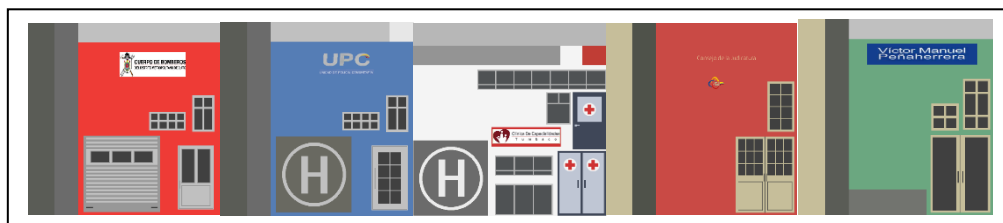


Figura 44. Texturas edificios cambiadas.

Se colocó también nombres sobre los edificios creados para reconocerlos desde una vista superior. Para esto se creó las siguientes texturas, como se puede ver en la Figura 45.



Figura 45. Texturas Techo Edificios.

Un paso importante en el desarrollo de la aplicación fue la creación de rutas por las cuales circularán los autos automáticos. Las rutas permiten implementar la capacidad de autonomía de los coches automáticos para dirigirse por las calles. (Ver Figura 46)

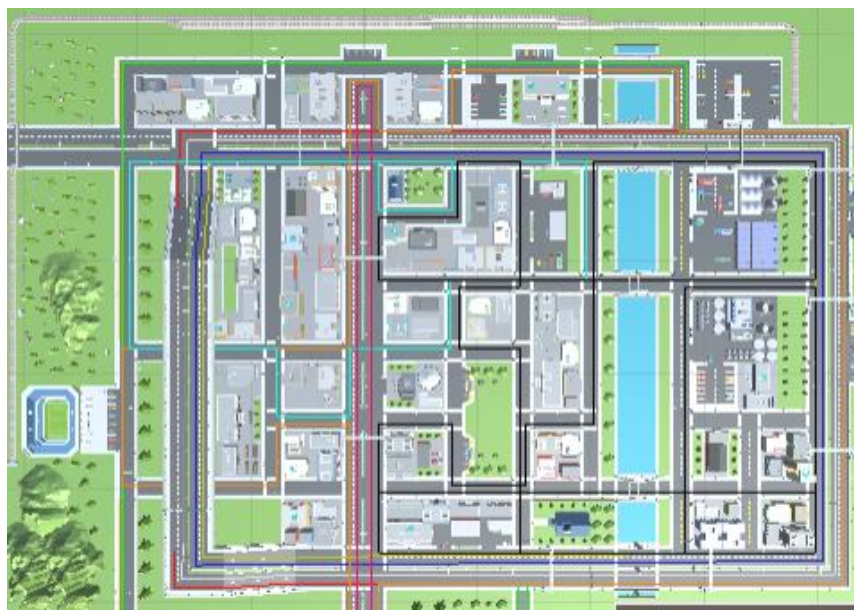


Figura 46. Ciudad con Paths.

El programa de seguimiento de trayectoria itera en cada punto (nodo) al siguiente punto (nodo) en la lista cuando detecta que el coche controlado por el juego está dentro de una distancia especificada desde su posición inicial hasta el nodo que determina su dirección actual. Las esferas (rojas) a lo largo del camino indican los puntos (nodos), como se puede ver en la Figura 47.

Para girar el coche hacia el nodo que debe dirigirse, el programa realiza cálculo de vectores, que determinan la cantidad de dirección que el sistema tiene que

proporcionar al coche. Estos cálculos se basan en un vector inicial creado tanto por la posición del punto en el camino actual y la posición actual del propio vehículo. La dirección en el plano XZ se obtiene mediante un vector relativo desde el coche al siguiente punto (nodo) en el camino. La función *Transform.InverseTransformPoint* de Unity permite obtener ese vector.

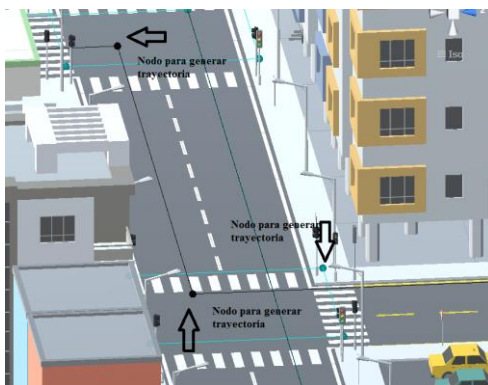


Figura 47. Nodos y Paths.

4.1.2 Coches automáticos

En cada una de las rutas se colocó grupos de vehículos automáticos orientados de acuerdo a las direcciones de las mismas, como se puede ver en la Figura 48.

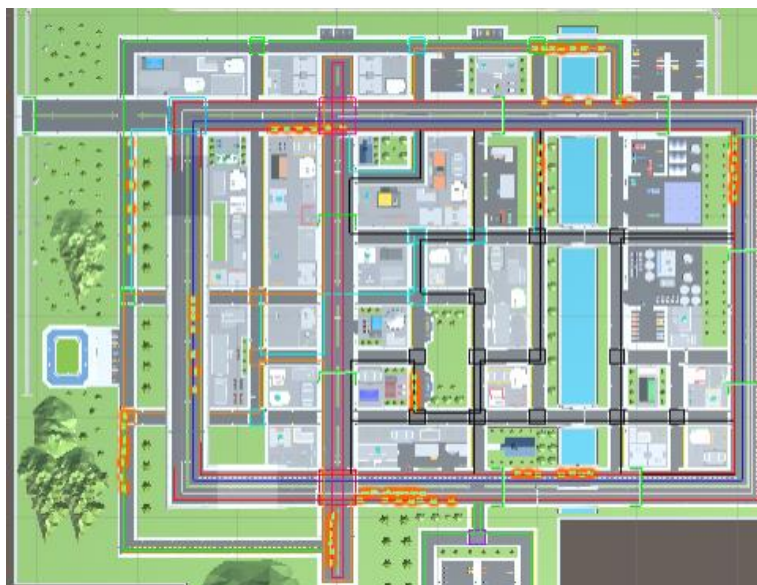


Figura 48. Autos en Ciudad.

Para la creación de los vehículos se utilizaron los modelos que se encontraban dentro del asset "SimplePolyCity", como se muestra en la figura 49.



Figura 49. Modelos de Autos.

Se aplicó textura y material para tener varios modelos de vehículos con llantas separadas como se muestra en la Figura 50.



Figura 50. Modelos de Autos Pintados.

Una vez obtenido cada tipo de vehículo se procedió a colocar colisionadores de llantas (*wheelcolliders*) en el objeto del vehículo. Se dotó a los coches de tres dispositivos sensores uno en un punto alto sobre el coche y los otros dos en los extremos laterales frontales del coche. Con estos sensores se evita las colisiones y se simula el efecto referente al respeto a las señales de tránsito, como se puede ver en la Figura 51.



Figura 51. Características Autos.

Para la colocación de los colisionadores de llantas (*wheelcolliders*) de los vehículos se llenó el cuadro con las características requeridas para la aplicación, como se puede ver en la Figura 52.

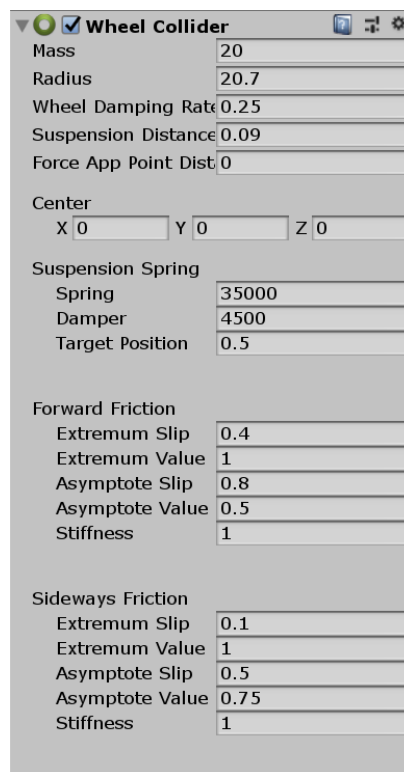


Figura 52. Componentes de los WheelColliders.

En el cuerpo (*body*) de cada tipo de vehículo se colocó una malla de colisión (*Mesh Collider*) para detectar cuando el vehículo choca contra cualquier otro objeto de la escena. El cuerpo de cada coche tiene una etiqueta (*tag*) característica que permite diferenciar cuando el rayo que se emite golpea contra el coche, como se puede ver en la Figura 53.

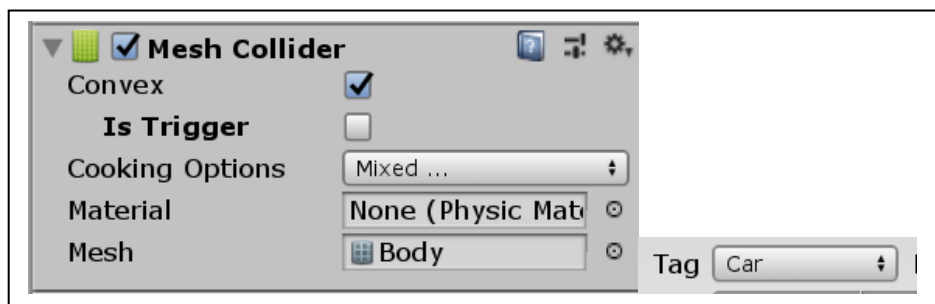


Figura 53. Componentes del body.

También se creó un objeto de juego (*gameObject*) llamado *WheelsMoving* que contiene cuatro círculos que se colocaron en la parte frontal en forma de textura de las llantas para simular el movimiento de rotación. A este objeto se comunica la

velocidad angular en RPM de los *colisionadores* de las llantas del vehículo. A continuación en la Figura 54, se muestra el vehículo con dichas llantas:

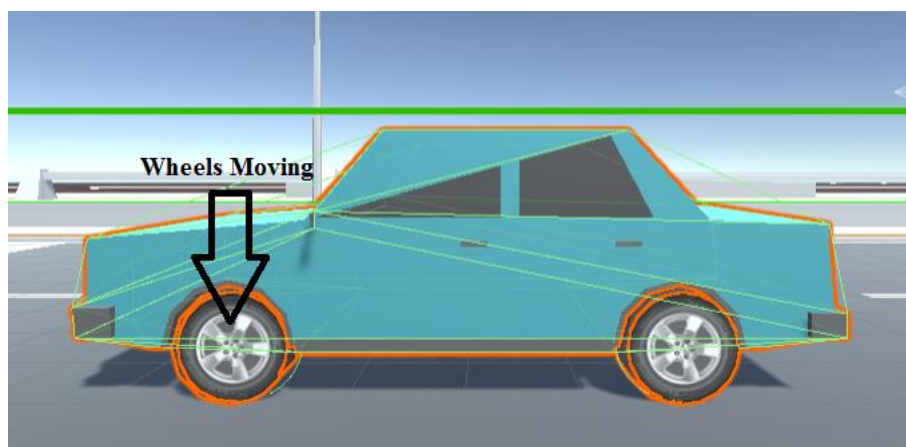


Figura 54. Wheels Moving.

En cada tipo de vehículo se llenó el cuadro con las características requeridas para la aplicación, como se puede ver en la Figura 55.



Figura 55. Características CarAi.

Scripts de coches automáticos

Para producir el efecto correspondiente al respeto a los semáforos se utilizó Máquinas de Estados Finitos (FSM). Con esta herramienta se produce el funcionamiento del coche automático en tres estados: a) Conducción (*Drive*), b) Listo para parar (*ReadyToStop*) y c) Parar (*Stop*); donde El paso de un estado de funcionamiento a otro está definido por el cumplimiento de condiciones. Si la luz roja está encendida el coche debe pasar del estado “Conducción” (*Drive*) al estado de “Listo para parar” (*ReadyToStop*). El coche pasa del estado “Listo para parar” (*ReadyToStop*) al estado de “Parar” (*Stop*) únicamente si detecta la presencia de un colisionador que indica la presencia de la intersección.

Para salir del estado “Parar” será necesario que la luz del semáforo indique el color verde. Del estado “Listo para parar” se regresa al estado de “Conducción” si la luz del semáforo vuelve a verde. El diagrama de estados utilizado se muestra en la Figura 56.

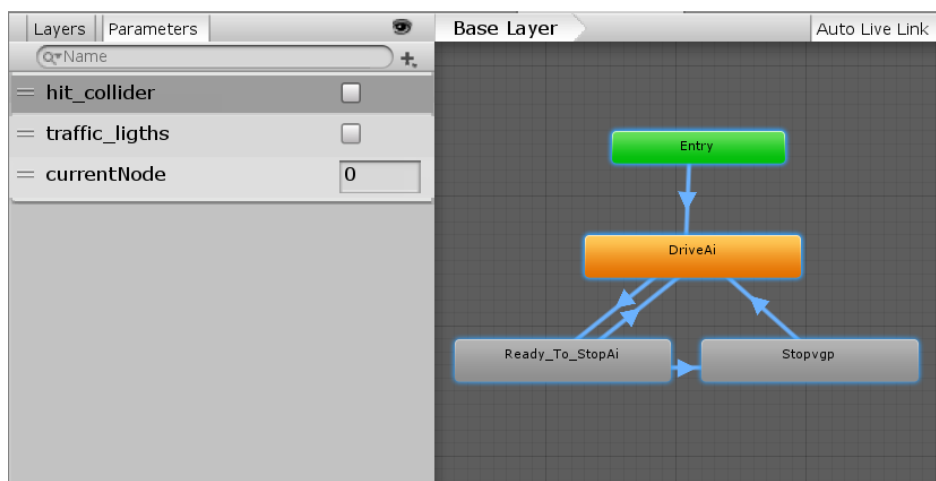


Figura 56. Máquina de Estados.

Para los vehículos automáticos se colocaron las siguientes condiciones:

- A) Control de coches mediante semáforos:** Para este propósito se utiliza la herramienta de emisión y detección de rayos que provee Unity. Se emite un rayo desde un punto que marca la posición del coche, hacia el piso con el objeto de encontrar las señales de intersección que se identifican mediante las etiquetas (tags) de colisionadores (*colliders*) en el piso. El coche debe detectar la luz roja del semáforo que le corresponde. Cuando se acerca a la intersección y la luz está en rojo debe detenerse justo antes de topar el paso

peatonal. Para lograr este objetivo es necesario aplicar el freno a una distancia que asegure que el coche se logre detener. Para ello se consideró la expresión que relaciona la distancia de paro con la velocidad del coche. (Ver Ecuación 3)

$$d_f = \frac{1}{2} \frac{v_o^2}{\mu g} \quad (3)$$

En esta expresión se reconoce la velocidad del coche (v_o), la gravedad (g) y el coeficiente de rozamiento (μ). (Ver Figura 57)



Figura 57. Semáforo.

- B) Control de Velocidad:** La velocidad de los coches está determinada por un controlador que actúa para producir tres modos de funcionamiento sobre el coche. El aumento de velocidad (*Accelerating*) se produce cuando la velocidad real es menor que la velocidad deseada y consiste en aplicar torque al motor. El modo de funcionamiento “Libre” (*FreeRunning*) se produce cuando la velocidad es la velocidad deseada y consiste en no aplicar ni torque de motor ni torque de freno, con lo cual el coche mantiene su velocidad. Cuando la velocidad disminuye a un 90% de la velocidad deseada, el torque vuelve a aplicarse al motor. Si la velocidad alcanza un 110% de la velocidad deseada se aplica el modo de funcionamiento “Frenar” (*Breaking*) que consiste en aplica torque de freno al coche.

C) Pare: Para la señal de Pare se ha considerado una distancia desde el nodo de intersección, por lo que si el coche alcanza esa distancia, se detiene y permanece detenido durante tres segundos. Al finalizar este tiempo se emite un rayo que sale desde el nodo de intersección en dirección al lado izquierdo del coche para determinar la presencia de coches hasta una distancia de 120m. Si no hay coches acercándose, el coche procede a avanzar y si algún coche es alcanzado por el rayo, el coche se detiene y permanece en ese estado hasta que no vengan coches, como se puede ver en la Figura 58.



Figura 58. Señal Pare

D) Limitar velocidad: Para esto se crea un objeto de juego “*SpeedSigns*” que contiene todas las señales que limitan la velocidad en el ambiente de la ciudad y que tienen la ubicación: *Speed30*, *Speed50*, *Speed90*. Luego mediante un lazo se recorre uno a uno los objetos hasta encontrar el que cumple la condición de estar activo para la posición actual del coche. Esta condición de activo significa que debe estar a la derecha a máximo 10m, que debe estar adelante dentro de un rango de 30m del coche en la dirección del coche y que la señal debe estar orientada a 90 grados del coche como para ser observada.

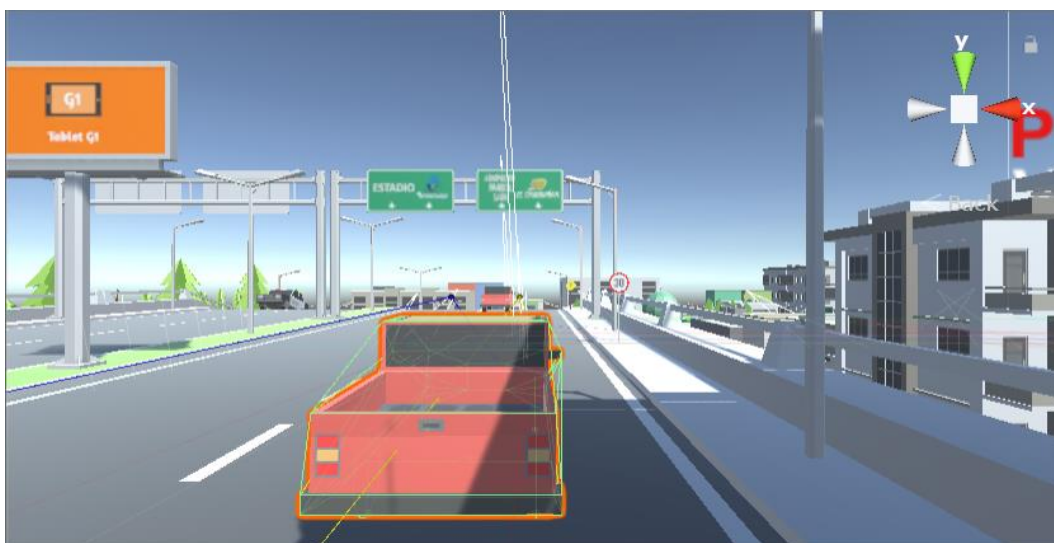


Figura 59. Límite de Velocidad

E) Manejo de giros: Los giros deben realizarse a una baja velocidad para asegurar que el vehículo no se desvíe de la trayectoria. Se considera que una adecuada velocidad para realizar el giro con un fiel seguimiento de la trayectoria es a 10Km/h. Para lograr que la velocidad alcance ese valor es necesario considerar que si la velocidad actual es mayor se debe calcular la distancia desde el nodo de giro a la que se producirá con el valor deseado de la velocidad. Esta distancia resulta de restar las distancias de parada que se producen con la velocidad actual y la velocidad deseada 10Km/h. Las distancias de parada a partir de determinada velocidad pueden obtenerse considerando que la energía cinética del coche se transforma en energía de rozamiento según la Ecuación 4.

$$E_c = E_r \quad (4)$$

La energía cinética depende de la masa del coche y del cuadrado de la velocidad, según la Ecuación 5.

$$\frac{1}{2} m v o^2 = m g d_f \mu \quad (5)$$

En la energía de rozamiento intervienen la masa del coche la gravedad, la distancia de paro y el coeficiente de rozamiento del piso. Utilizando esta expresión se puede deducir que para alcanzar una velocidad distinta de cero se puede considerar la misma deducción para obtener la distancia recorrida por el coche hasta alcanzar la velocidad final cuando se aplica el freno, según la Ecuación 6.

$$\frac{1}{2}mvo^2 - \frac{1}{2}mvf^2 = mgd_f\mu \quad (6)$$

$$\text{De donde: } d_f = \frac{1}{2}(v_0^2 - v_f^2)/(g\mu). \quad (7)$$

Con la Ecuación 7 se puede conocer a que distancia debe aplicarse el freno para lograr llegar a la velocidad final de 10km/h para iniciar la curva.

4.1.3 Semáforos

La aplicación utiliza dos tipos de semáforos de acuerdo a la orientación de la calle. El un tipo de semáforo actúa para las calles de sentido Norte-Sur o Sur-Norte. El otro tipo actúa para controlar el paso en el sentido Este-Oeste u Oeste-Este. (Ver Figura 60)

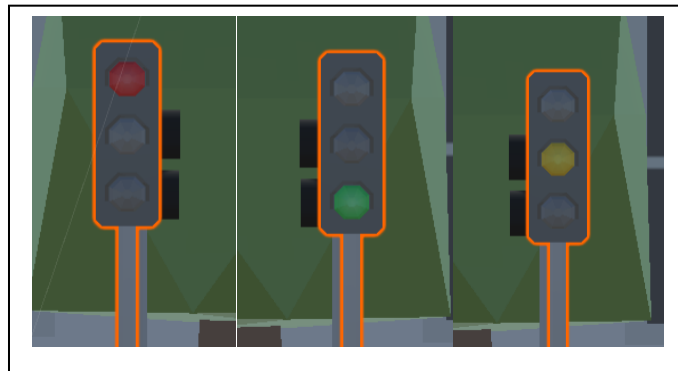


Figura 60. Modos de funcionamiento del semáforo.

También se diseñó semáforos peatonales que indican en qué momento pueden pasar el paso cebra los peatones que estén caminado por la vía. Cuando se encuentra en verde se muestra los segundos restantes para poder cruzar la vía. El tiempo que transcurre por cada ciclo total de los tres tipos de semáforo es de 100 segundos de los cuales 40 segundos son para la orientación horizontal, 40 segundos para la orientación vertical y 20 segundos para los peatones. Los semáforos de coches duran en rojo 60 segundos, en verde 35 segundos y amarillo 5 segundos. (Ver Figura 61)

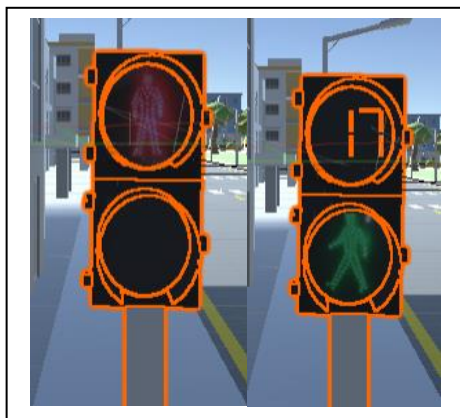


Figura 61. Semáforos Peatonales.

Los semáforos funcionan de la siguiente manera, mientras el semáforo peatonal se encuentre en rojo, los dos semáforos para los conductores se encuentran en verde, los dos tipos de semáforos para los conductores son horizontal y vertical, mientras el semáforo vertical esté en verde o en amarillo los otros dos se encontrarán en rojo, y así respectivamente cumpliendo los tiempos antes establecidos.

4.1.4 Peatones

Se crearon peatones en cada uno de las intersecciones con pasos cebra para que caminen cuando el semáforo peatonal se encuentre en verde.

Los peatones utilizados son los siguientes: a) Hombre (Ver Figura 62), b) Mujer (Ver figura 63) y c) Niños (Ver Figura 64).



Figura 62. Tipos de hombres.



Figura 63. Tipos de Mujeres.



Figura 64. Niño y Niña.

Para el desarrollo de los humanos se utilizó la aplicación Makehuman en donde se partió de una imagen básica, como se puede ver en la Figura 65.



Figura 65. Modelo básico humano.

De la imagen básica se procedió a dar forma según las características de las personas que se necesitaba, en la Figura 66 se muestra las características físicas que se pueden cambiar en el personaje.

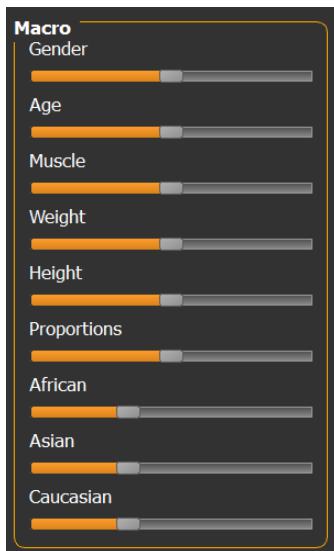


Figura 66. Características físicas.

Además para colocar la ropa al personaje Makehuman permite agregar los elementos desde la página web: <http://www.makehumancommunity.org/>, perteneciente a la comunidad de diseño, desde donde se copia la información necesaria para agregarla en el programa. (Ver Figura 67)

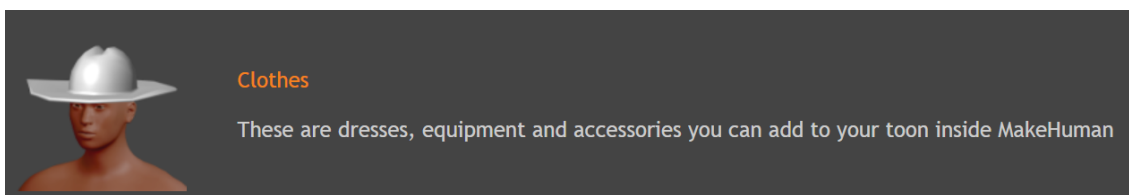


Figura 67. Ropa.

Para colocar la ropa en el personaje basta con arrastrar esta desde el programa al modelo y la ropa se carga en el personaje que ha sido creado. Los tipos de ropa se muestran en la Figura 68.

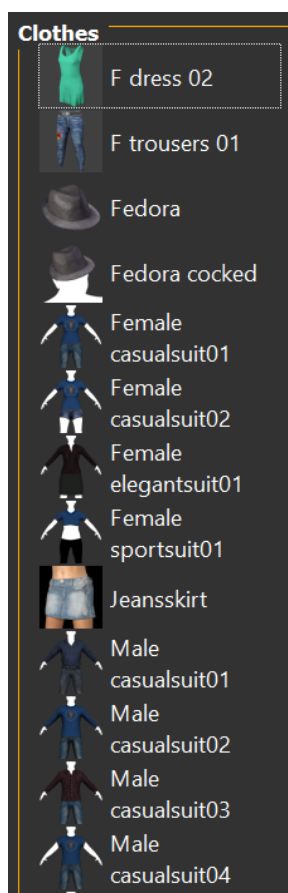


Figura 68. Tipos de Ropa.

Para que el personaje funcione en Unity es necesario que sea exportado en una pose específica conocida como pose T. A continuación en la Figura 69 se muestran las poses que permite el programa.

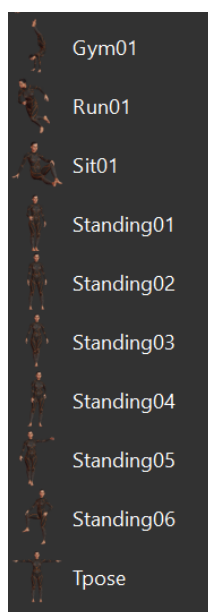


Figura 69. Poses.

Para el movimiento de las personas en la escena se utilizó rutas para las mismas y mallas de navegación (*Navmesh*). Las mallas de navegación se definieron en todas las intersecciones de la ciudad. (Ver Figura 70)

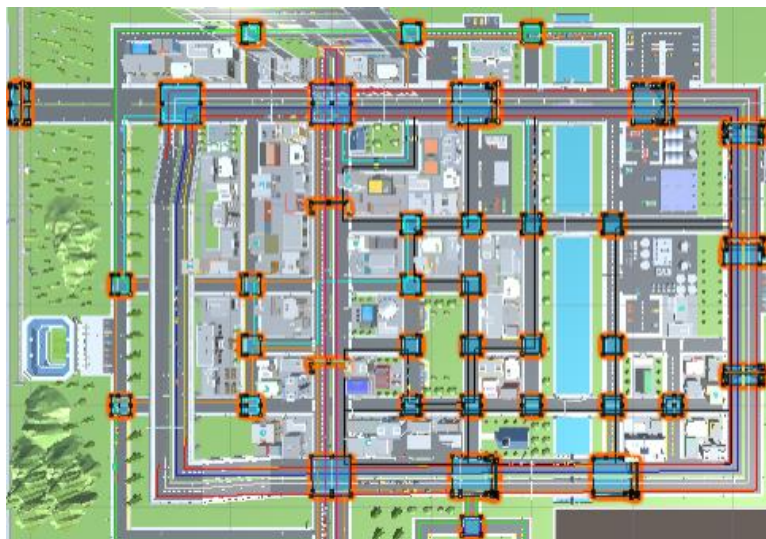


Figura 70. Navmesh en Ciudad.

En cada una de las mallas de navegación se colocó un grupo de personas según el tamaño de la trayectoria por la cual caminan (*path*). (Ver Figura 71)

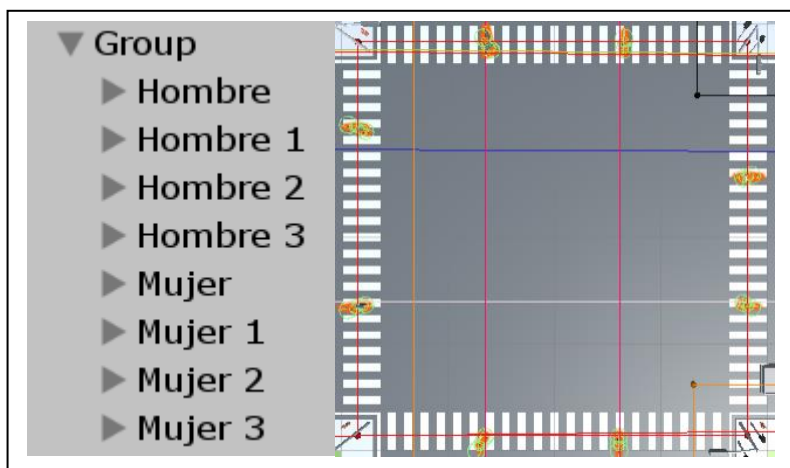


Figura 71. Personas en path.

Para el movimiento de las personas se utilizó el *AIThirdPersonController* de los *Standard Assets* de la aplicación *Unity*. Allí se modificó el *Avatar* y se copió el modelo a utilizar en la pose T. (Ver Figura 72)

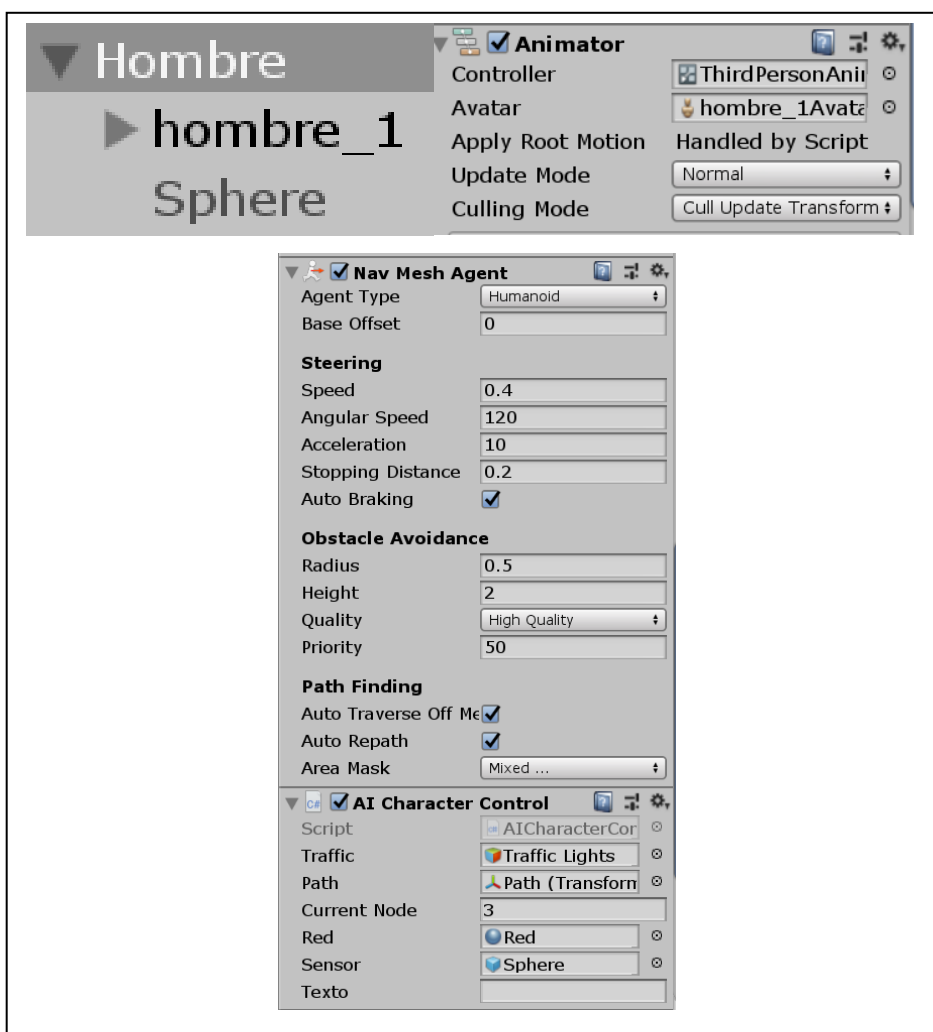


Figura 72. Personajes en Unity.

Tanto los coches como los peatones de la aplicación disponen de IA para dotar a estos objetos de funcionamiento autónomo en las escenas lo cual involucra realizar acciones inteligentes como la toma de decisiones y el seguimiento de trayectorias.

4.2 Desarrollo de la interfaz gráfica del rol de peatón

Para el desarrollo del rol de peatón se utilizó el *Third Person Controller* de los *Standard Assets* de la aplicación *Unity*. Se utilizó *Avatars* de niño y de niña para representar a los dos tipos de jugadores. (Ver Figura 72)



Figura 73. ThirdPersonController.

Se creó la ruta por la cual se debe cumplir la actividad para lograr llegar al punto destino como se muestra en la Figura 73.



Figura 74. Ruta peatón

En el camino se creó audios e imágenes de las señales para los peatones. (Ver Figura 75)



Figura 75. Señales Peatonas.

4.3 Desarrollo de la interfaz gráfica del rol de conductor

Para el desarrollo del rol de conductor se utilizó los tres tipos de vehículos mostrados anteriormente en tres rutas distintas según el tipo de vehículo las rutas son las siguientes: a) Bomberos – Escuela (Ver Figura 76), b) Estadio – Aeropuerto (Ver Figura 77) y c) Clínica – Parqueadero Chaquiñán (Ver Figura 78)



Figura 76. Ruta Bomberos-Escuela.

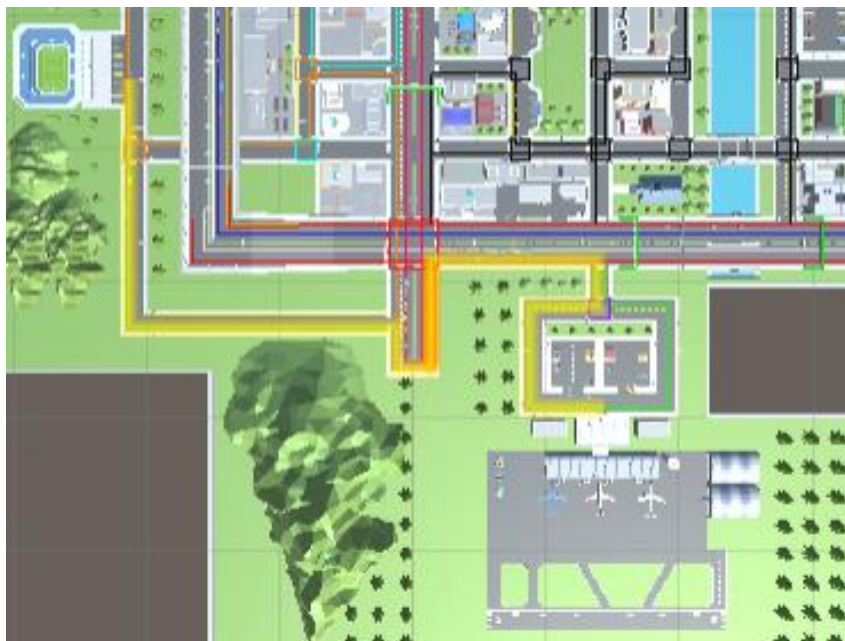


Figura 77. Ruta Estadio-Aeropuerto.

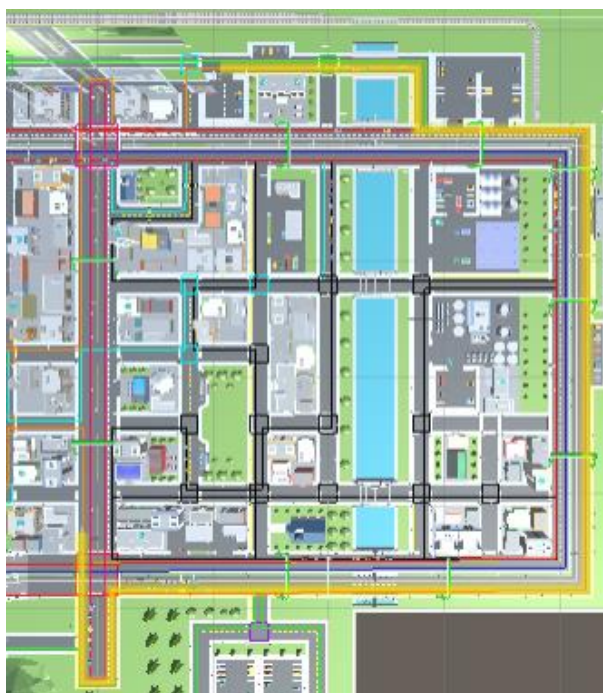


Figura 78. Ruta Clínica - Parqueadero Chaquiñán.

El vehículo es ubicado en el inicio de la ruta para que se maneje por medio del volante hasta llegar al punto de destino. Para cambiar el control al volante se modificó el input manager de la aplicación en los axis horizontal, vertical y jump que son los que le

permiten acelerar, girar el vehículo y frenar como se muestra a continuación en la Figura 79

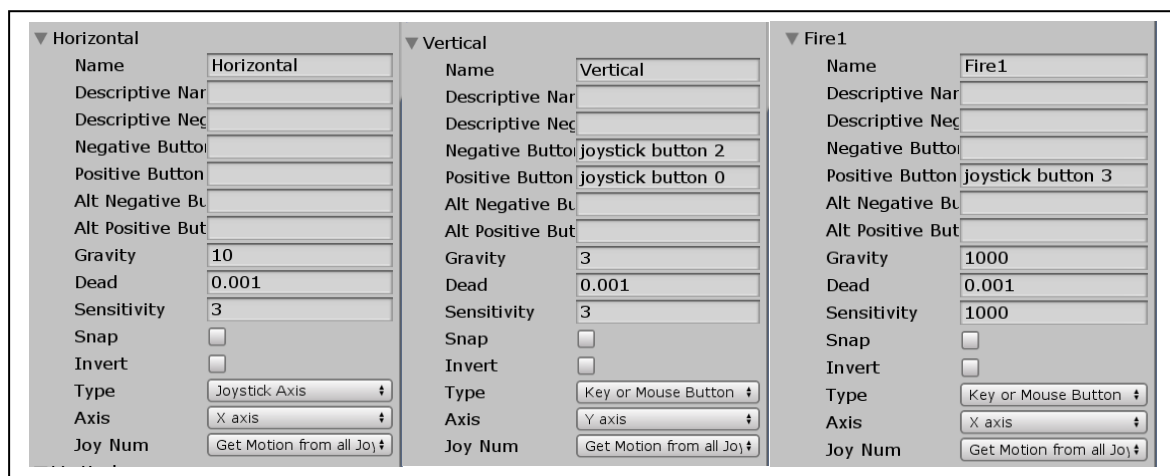







Figura 79. Inputs Volante






Para el rol del conductor se mostrarán las señales de tránsito necesarias para el buen funcionamiento de la aplicación igualmente con su audio respectivo.

Una muestra representativa de los audios que se crearon en la aplicación se describe en la Tabla 14.

Tabla 14

Audios Señales de tránsito.

Señal de tránsito	Audio
	Semáforo. Señal de tránsito luminosa. Si la luz es roja debe detenerse. Si es verde puede avanzar. Si es amarilla previene que pronto cambiará a luz roja y si es posible debe detenerse.
	Señal regulatoria. La velocidad máxima es 90 kilómetros por hora.
	Señal regulatoria. Pare. Es obligatorio detener el vehículo para que pasen los peatones. Luego puede seguir a la intersección para ver si no vienen vehículos por la vía principal. Entonces puede avanzar.
	Señal regulatoria. Vía de un solo sentido. Usted está en el sentido contrario.
	Señal preventiva. Se aproxima a un giro a la izquierda.

	Señal preventiva. Se aproxima a un semáforo
	Señal preventiva. Se aproxima a una vía de circulación en los dos sentidos
Señal de tránsito	Audio
	Señal preventiva. A su derecha se aproxima a una salida de vehículos.
	Señal informativa. Usted está en el camino del aeropuerto.
	Señal informativa. Usted se aproxima a un parqueadero.

4.4 Desarrollo de la interfaz gráfica del rol de espectador

Para el desarrollo del rol del espectador se utilizó una cámara la cual se la colocó en la parte alta de la escena para permitir observar todo el entorno de la ciudad. (Ver Figura 80)

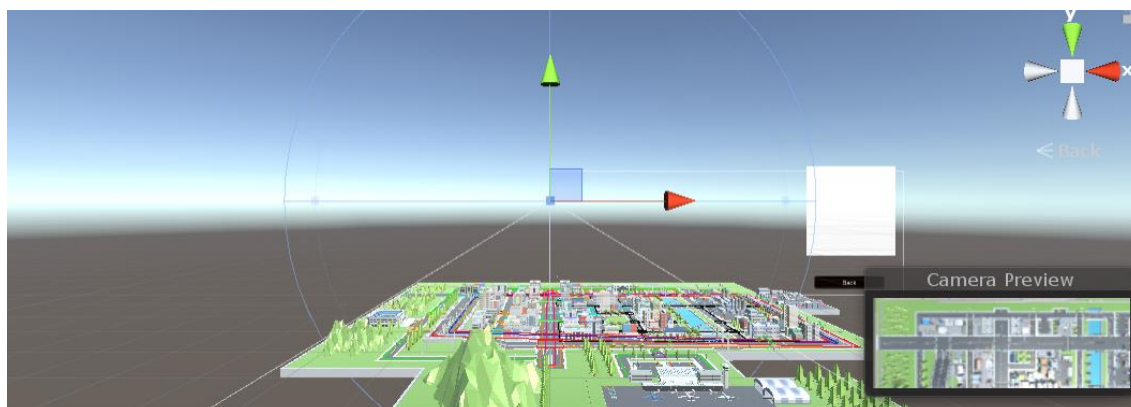


Figura 80. Rol de Espectador.

Mediante esta vista se observa la ciudad desde una perspectiva diferente que permite reconocer los detalles de la ciudad, además permite seleccionar un automóvil al cual se lo muestra en perspectiva y se muestran las señales del mismo.

4.5 Pruebas de caja blanca y caja negra

4.5.1 Pruebas de caja blanca

Para las pruebas de caja blanca se analizarán los if más importantes de la aplicación, los que permitieron que los autos automáticos circulen con normalidad los cuales son los siguientes: a) Control para evitar colisión en las curvas (Ver Figura 81), b) Control para evitar colisión con otros coches (Ver Figura 82) , c) Detección de Contravía (Ver Figura 83)

```

public void DontCrashInCurves()
{
    float ResetDistance;
    float StopToNodeDistance;
    StopToNodeDistance = 8.5f;
    ResetDistance = 20f;
    nodeTag = nodes[anim.GetInteger("currentNode")].tag;
    if (nodeTag == "NodeRightCurve" || nodeTag == "NodeLeftCurve" || nodeTag == "SpecialNode")
        enableStopCurveWork = true;// para tener una bandera que luego se resetea muy bien

    if (enableStopCurveWork)
    {
        currentDistanceToCurveNode = Vector3.Distance(transform.position,
nodes[anim.GetInteger("currentNode")].position);
        if (currentDistanceToCurveNode < StopToNodeDistance)
            isCurve = true;
        curveStop = isCurve && !enablePassCurve;
        if (curveStop)
        {
            dontTouchCurve = false;
            Braking();
        }
        else
            dontTouchCurve = true;

        if (curveStop && currentSpeedKmh < 0.1)
            { enablePassCurve = WaitSeconds(.1f); }

        // Sensar si viene carro a la izquierda
        if (enablePassCurve && currentSpeedKmh < 0.1)
        {
            float Lengthc = 5f;
            carIsInCurve = CarIsDetected(Lengthc, nodeTag);
        }
        if (carIsInCurve)
            Braking();
    }

    // Reseteo de nodo a condicion inicial
    if (nodes[previousNode].tag == "NodeRightCurve" || nodes[previousNode].tag ==
"NodeLeftCurve" || nodes[previousNode].tag == "SpecialNode")
    {
        distancePreviousNode = Vector3.Distance(transform.position,
nodes[previousNode].position);
        if (distancePreviousNode > ResetDistance && distancePreviousNode < 1.1 * ResetDistance)
        {
            isCurve = false;
            enablePassCurve = false;
            dontTouchCurve = true;
            curveStop = false;
            enableStopCurveWork = false;
        }
    }
}

```

Figura 81. Código de control de Manejo en Curvas.

A) Control para evitar colisión en las curvas

El funcionamiento es el siguiente: el coche automático recibe la información que se acerca a un nodo con etiqueta “Curva Izquierda” o “Curva Derecha”, que le advierte la proximidad de una curva. El programa activa una bandera “Habilita trabajo de

parada en Curvas” (*EnableStopCurveWork*) con la cual el coche automático conoce que está en el proceso de no chocar en la curva.

Cuando se habilita este proceso se procede a medir la distancia desde el coche al nodo y compararlo con un valor de umbral (*StopToNodeDistance*) que al ser alcanzado activa la bandera de “Es Curva” (*IsCurve*). Esta bandera es una de las dos condiciones que produce la activación del estado Parada en Curva, que es señalado con la bandera (*CurveStop*). La otra condición es que haya transcurrido dos segundos en el estado. Para lograr esta funcionalidad se utiliza una bandera de habilitación de avanzar del estado de Parada (*EnablePassStop*). Esta bandera está inicialmente desactivada y es activada al finalizar el tiempo de un temporizador de dos segundos “*wait2seconds()*”. A partir de que se encuentra en el estado de Parada en Curva, se desactiva la bandera de indicación de que el coche No Topa Curva (*DontTouchCurve*) que es uno de los habilitadores del proceso *SpeedControl()* que produce el movimiento del coche, además se ejecuta el comando de frenado *breaking()* por lo que como resultado el coche debe detenerse durante un tiempo de dos segundos.

Cuando finaliza el estado de Parada en Curva (\neg *CurveStop*), el coche indica mediante la habilitación de señal No Topa Curva (*DontTouchCurve*) para que el coche siga avanzando. El estado de Parada en Curva (*CurveStop*) es una de dos condiciones para habilitar el estado de detección de coches en la curva y la otra condición es que la velocidad sea cercana a cero. En el Estado de Detección de Coches en la curva se activa la bandera de Coche está en Curva (*CarIsInCurve*) que también deshabilita *SpeedControl()* y ejecuta el comando freno. Si no hay coches en la curva se deshabilita bandera (*CarIsInCurve*) con lo que el método *SpeedControl* retoma el control de la velocidad.

```

public void DontCrashCar()
{
    // Sensor Alto
    frontSensorAngleHigh = (float)Math.Atan((sensorStartPosHigh.y - transform.position.y) /
(distanceParo+.5f*currentSpeedKmH)) * 180f / (float)Math.PI;
    dontTouchCar = true; // inicia con DontTouchCar en true para verificar todos los sen
    if (Physics.Raycast(sensorStartPosHigh, Quaternion.AngleAxis(frontSensorAngleHigh,
transform.right) * transform.forward,
    out hitHigh, sensorLengthHigh))
    {
        collidertag = hitHigh.collider.tag;
        if (hitHigh.collider.CompareTag("Car"))
        {
            Braking();

            dontTouchCar = false;
        }
        dontTouchCar = dontTouchCar && true;
        Debug.DrawLine(sensorStartPosHigh, hitHigh.point);
    }

    //Sensores bajos
    if (down)
    {
        angleAdd = 14f;
        frontSensorAngleLow = 0 - angleAdd;
        LowSensors();
    }
    frontSensorAngleLow = 0;
    LowSensors();
}

```

Figura 82. Código para evitar colisión entre coches.

B) Control para evitar la colisión con otros coches.

Existe una regla muy conocida entre los conductores respecto de la distancia que deben guardar respecto del vehículo que va delante. Esta regla consiste en fijar un objeto en la carretera por ejemplo un poste lateral por donde está pasando el vehículo de adelante y contar desde 1001 hasta 1003. Solo en ese momento debe el coche pasar por el objeto mencionado. Para el caso de los coches con inteligencia artificial se dispone de la expresión de cálculo de la distancia de parada que ha sido ya indicada que permite conocer exactamente la distancia que requiere mantener el coche en cuestión con el coche delantero. El mecanismo mediante el cual se mantiene esa distancia es emitir un rayo que mantiene la distancia de parada hacia el coche de

adelante. Cuando el rayo golpea al coche se ingresa al modo de funcionamiento de frenado. Si el rayo no golpea a un coche el modo de funcionamiento es el aumento de velocidad.

```
public void RightWrongWay()
{
    if (rightWayF)
    {
        UISignImage.sprite = Resources.Load<Sprite>("RightWaySign");
        if (flagPlay)
        {
            myAudioSource = GameObject.Find("RightWaySign").GetComponent<AudioSource>();
            myAudioSource.Play();
            flagPlay = false;
        }
    }
    if (!rightWayF)
    {
        UISignImage.sprite = Resources.Load<Sprite>("WrongWaySign");
        if (flagPlay)
        {
            myAudioSource = GameObject.Find("WrongWaySign").GetComponent<AudioSource>();
            myAudioSource.Play();
            flagPlay = false;
        }
    }
}
```

Figura 83. Código de detección de contravía.

C) Detección de contravía

Para detectar que un coche está en contravía se implementó un sistema de orientación del coche, basado en el ángulo de rotación. Para este propósito se consideró que el mapa tiene las coordenadas definidas: el Norte a la derecha, el Sur a la izquierda, el Este hacia abajo y el Oeste hacia arriba. Para el coche, si el ángulo es 0 grados el coche está orientado hacia el Oeste, si el ángulo es 90 grados el coche está orientado hacia el Norte, si el ángulo es 180 grados está orientado hacia el Este y finalmente si el ángulo es 270 grados está orientado hacia el Sur. En Unity para determinar el ángulo del coche se utiliza el comando ***transform.eulerAngles.y***.

Las vías de un solo sentido han sido marcadas con una señal de tránsito de una sola vía que tiene una etiqueta con la dirección en la que está apuntando la señal.

Es decir, se tiene señales de una vía etiquetadas con las distintas direcciones Norte, Sur, Este y Oeste.

La orientación del coche se compara con el sentido de la vía y se determina si el coche está circulando en el sentido correcto o está en contravía.

4.5.2 Pruebas de caja negra

La aplicación desarrollada cuenta con 13 escenas respectivamente distribuidas, cada escena se encuentra conectada con la siguiente, mediante cargas en los respectivos objetos y la aplicación tiene el nombre de “SaveYourLife”, donde la primera escena es la siguiente. (Ver Figura 84)

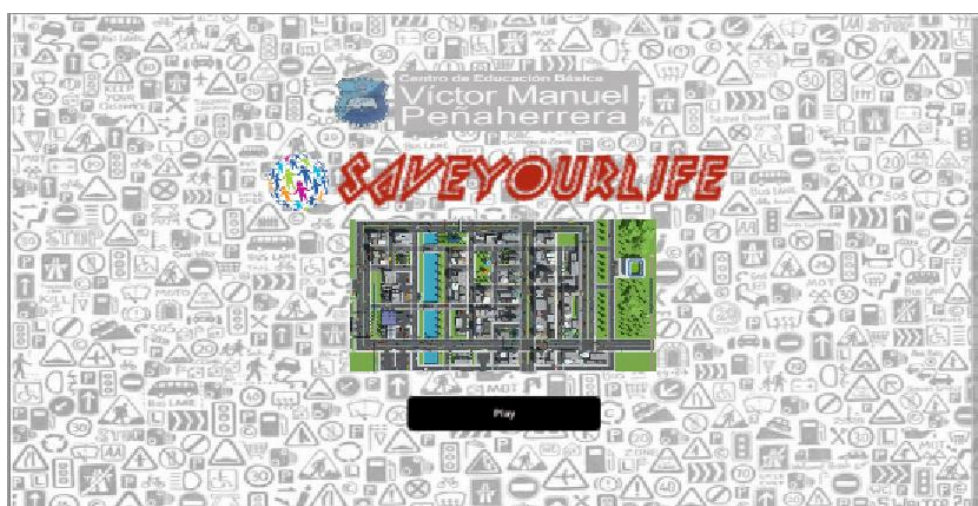


Figura 84. Pantalla Principal.

En la pantalla principal se debe dar un clic al botón play para ingresar en las opciones del juego, donde se muestra el nombre de la escuela, de la aplicación y una vista panorámica de la ciudad. Además, en el fondo se encuentra una imagen con las señales de tránsito que existen, la siguiente escena se muestra a continuación. (Ver Figura 85)

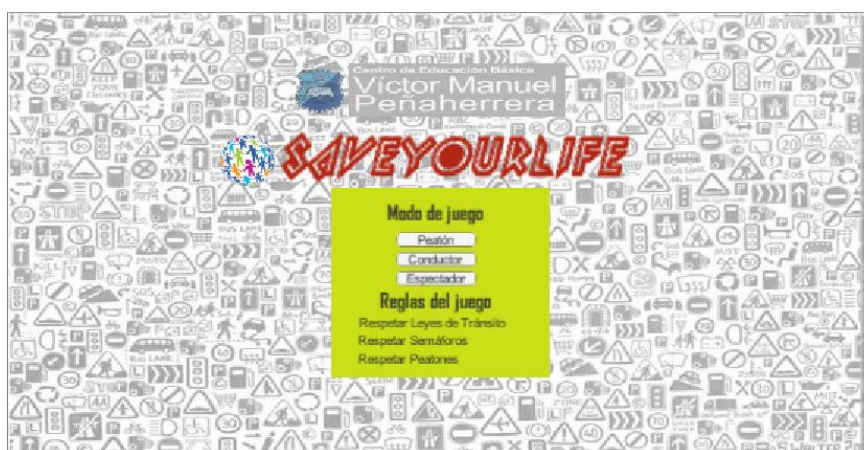


Figura 85. Modo de Juego.

En la pantalla “Modo de Juego”, el jugador debe seleccionar el rol en el cual desea participar, las opciones son peatón, conductor o espectador. Además se le muestra las reglas de juego que tiene que seguir el niño para cumplir con las actividades de las demás pantallas. En el caso de seleccionar la opción como peatón se le mostrará la siguiente pantalla. (Ver Figura 86)



Figura 86. Peatón.

En la pantalla de la Figura 86 se muestra la opción de escoger entre niño o niña, al seleccionar el tipo de personaje se mostrará la imagen del mismo, como se muestra a continuación en la Figura 87.



Figura 87. Personaje Seleccionado.

Para continuar a la siguiente pantalla se tiene que dar clic en la imagen seleccionada la cual se redirigirá a la simulación como se muestra a continuación: a) Niño (Ver Figura 89), b) Niña (Ver Figura 90), durante la carga se muestra las instrucciones de juego como se puede ver en la Figura 88.



Figura 88. Instrucciones Juego-Peaton.



Figura 89. Simulación peatón-Niño.



Figura 90. Simulación peatón-Niña.

En las pantallas de la Figura 90 y 91, el niño/a debe caminar por la ciudad conociéndola, evitando ser atropellado y mientras camina se despliega la información de las señales de tránsito mencionadas en la Figura 75, al dar clic en el botón Back regresará a la pantalla anterior.

En la pantalla de selección del juego en el caso de seleccionar el rol de conductor se desplegará la siguiente pantalla. (Ver Figura 91)

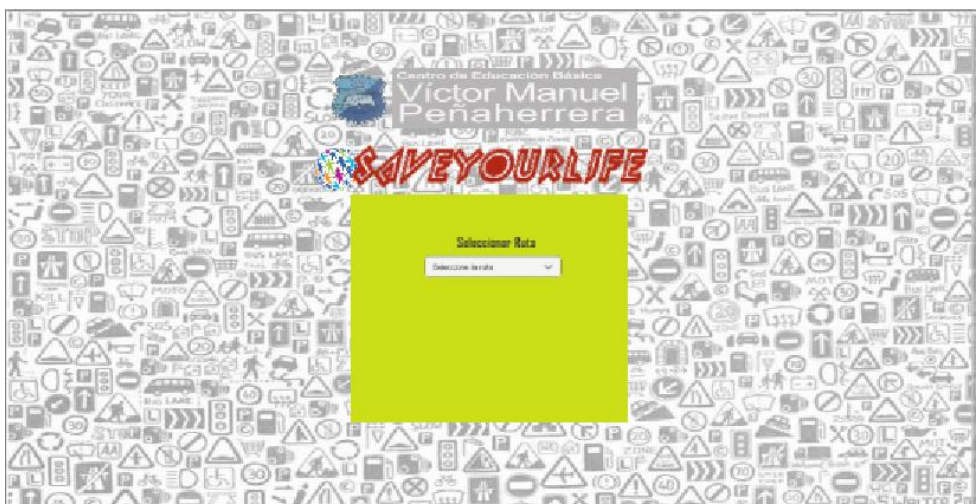


Figura 91. Pantalla Conductor.

En la pantalla de la Figura 91, se seleccionará la ruta y se mostrará el vehículo a utilizar como se muestra la Figura 92.

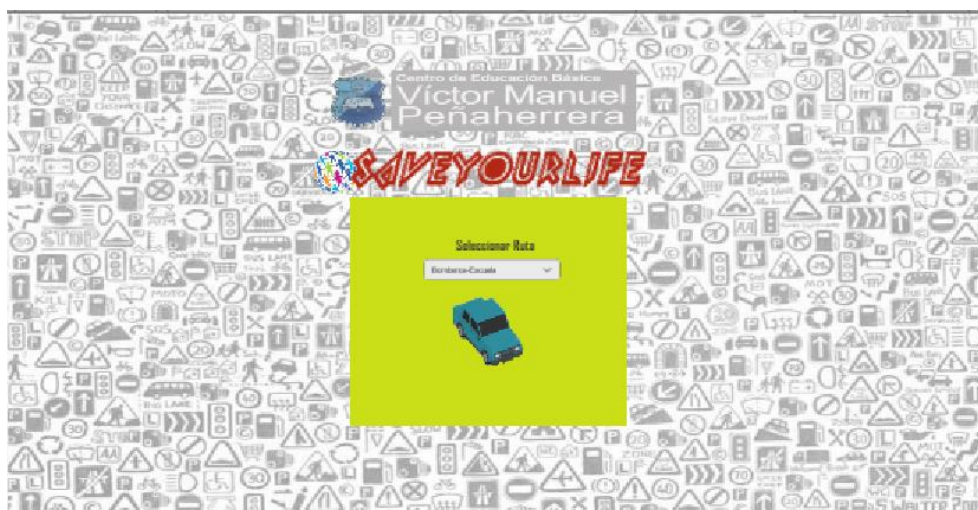


Figura 92. Ruta seleccionada.

Al dar clic en el vehículo de la Figura 92, se mostrará la pantalla de la aplicación según la ruta escogida y el auto mostrado como se muestra a continuación:
 a) Ruta Bomberos-Escuela (Ver Figura 94), b) Ruta Estadio-Aeropuerto (Ver Figura 95) y c) Ruta Clínica-Parqueadero Chaquiñan (Ver Figura 96), durante la carga se muestra las instrucciones de juego como se puede ver en la Figura 93.



Figura 93. Instrucciones de Juego-Conductor.



Figura 94. Simulación ruta bomberos-escuela.



Figura 95. Simulación ruta estadio-aeropuerto.



Figura 96. Simulación ruta clínica - parqueadero chaquiñán.

En estas escenas de las Figura 94, 95, 96, el jugador mediante las flechas conduce el coche a través de la ciudad. Debe respetar el semáforo, las señales de límite de velocidad y las de vías de un solo sentido, además debe evitar colisionar con los coches y no atropellar a las personas. Por cada infracción que comete se rebaja puntos y si el puntaje llega a 0 pierde el juego. Si se choca o atropella termina el juego, pero si cumple el recorrido podrá guardar su puntaje.

En el caso de perder por cualquiera de las situaciones anteriores se mostrará la siguiente pantalla. (Ver Figura 97)



Figura 97. Pantalla Game Over.

En el caso de cumplir con la actividad anterior se mostrará la siguiente pantalla donde el niño podrá ingresar su nombre. (Ver Figura 98)

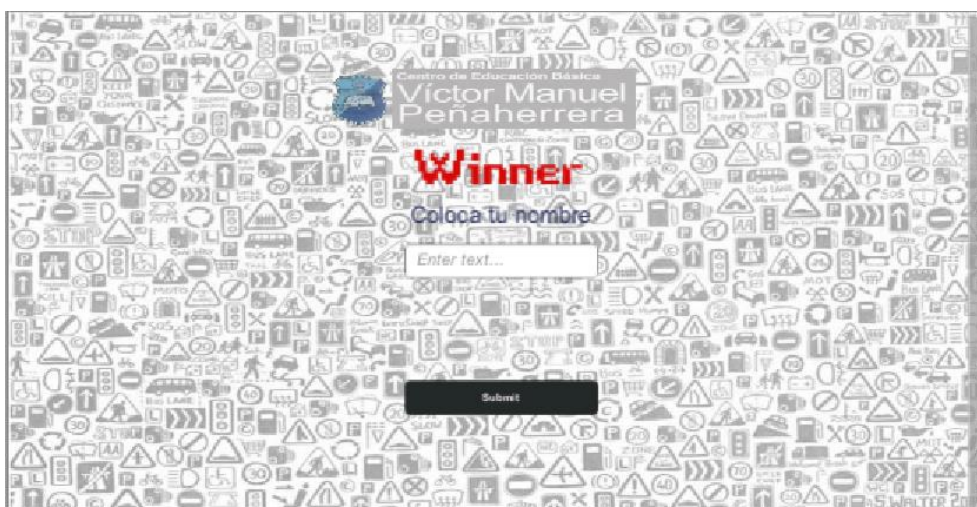


Figura 98. Pantalla Ganador.

Una vez ingresado el nombre en la pantalla de la Figura 98, se desplegará una pantalla con los mejores puntajes. (Ver Figura 99)

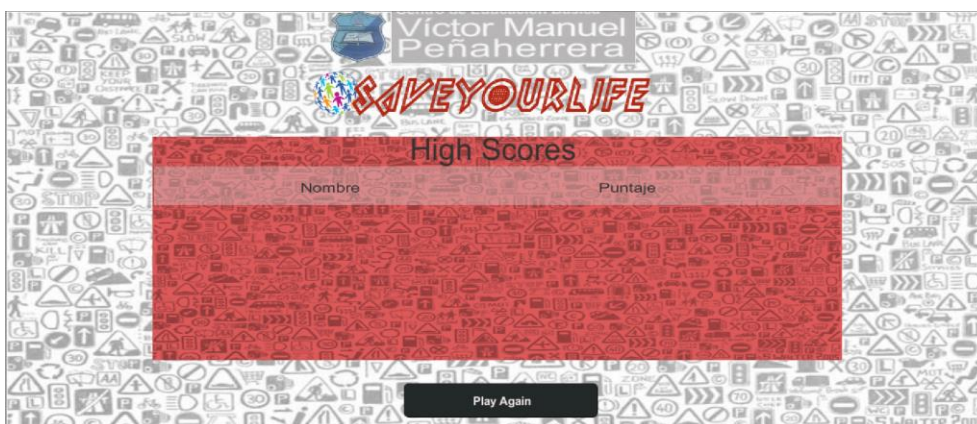


Figura 99. Tabla de Puntajes.

En el caso de que en la pantalla de modo se seleccione la opción de espectador se desplegará la escena de la Figura 101, durante la carga se muestra las instrucciones de juego como se puede ver en la Figura 100.



Figura 100. Instrucciones de Juego-Espectador.



Figura 101. Pantalla modo espectador.

Para seleccionar un vehículo en la pantalla de la Figura 101, se debe dar clic sobre el mismo y se desplegará la información que se está mostrando para dicho vehículo, para alejar o acercar se utiliza la rueda giratoria del mouse. Cuando se desee regresar se dará clic al botón Back. (Ver Figura 102)



Figura 102. Pantalla modo espectador – Auto Seleccionado.

CAPÍTULO V

RESULTADOS

5.1 Validación y Pruebas

La aplicación fue entregada en la escuela “Víctor Manuel Peñaherrera” con gran acogida por parte de la misma y se realizó una encuesta a los niños de 5to, 6to y 7mo de básica, la encuesta fue realizada a 30 niños de la escuela. Las preguntas de la encuesta se detallan en la Tabla 17.

Tabla 15

Preguntas de usabilidad escuela.

1. ¿Qué te pareció el manejo del coche?	Bueno	Normal	Lento	Malo
2. ¿Aprendiste las señales de tránsito?	Si	Más o Menos		Un poco
3. ¿Qué no te gustó del programa?	Bien	Lento	Voz	Animación
4. ¿A qué velocidad debería viajar en la ciudad?	Correcto		Incorrecto	
5. ¿Qué color debe tener el semáforo para avanzar?	Correcto		Incorrecto	
6. ¿Qué significa cuando el semáforo está en el color amarillo?	Correcto		Incorrecto	
7. ¿Qué significa la señal regulatoria PARE?	Correcto		Incorrecto	
8. ¿Para qué sirven las señales preventivas?	Correcto		Incorrecto	
9. ¿De qué color son las señales informativas?	Correcto		Incorrecto	
10. ¿Diga una señal informativa?	Correcto		Incorrecto	

En la escuela se utilizó únicamente una máquina en la cual se les fue pidiendo a los niños que la utilicen por un periodo de tiempo para probar la aplicación las características de dicha máquina se muestran en la Figura 103.

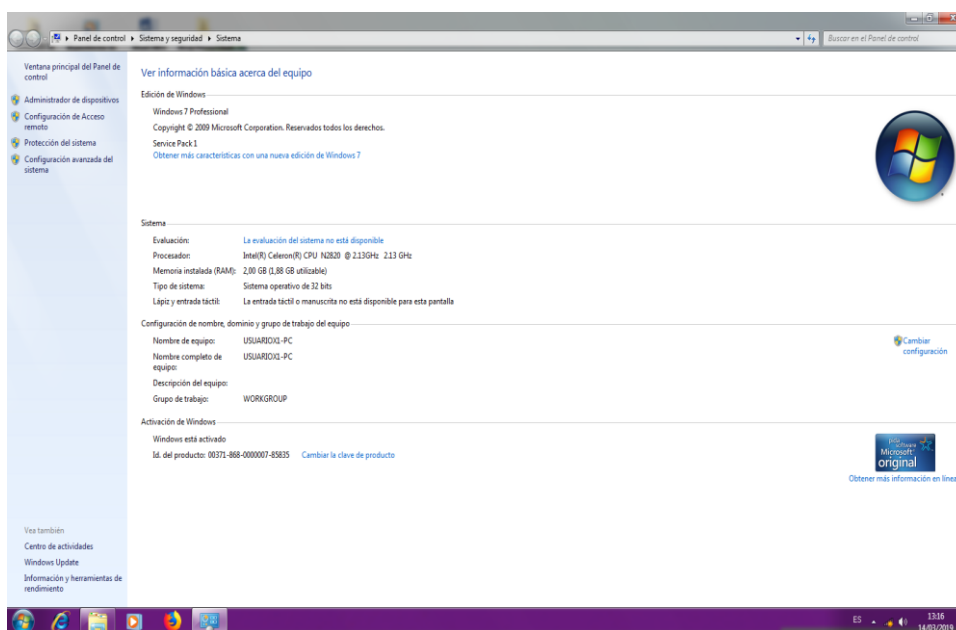


Figura 103. Características Computadora Escuela.

5.2 Resultados

Para la pregunta número 1 se obtuvieron los siguientes resultados. (Ver Figura 104)

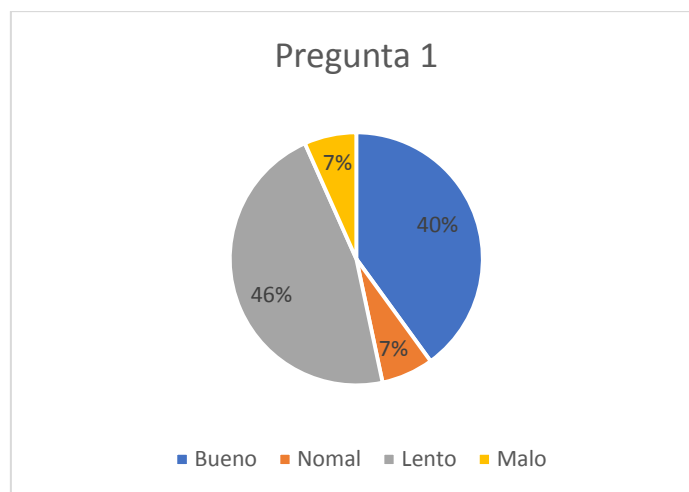


Figura 104. Pregunta 1.

De lo que se puede observar en la Figura 104, el sistema operativo que posee el equipo de la escuela cuyas características se mencionan en la Figura 103, hace que el 46% de los encuestados mencionen que la aplicación es lenta, mas, con un 40% la califican como buena, por lo que según esta pregunta la aplicación es acogida por los niños.

Para la pregunta número 2 se obtuvieron los siguientes resultados. (Ver Figura 105)

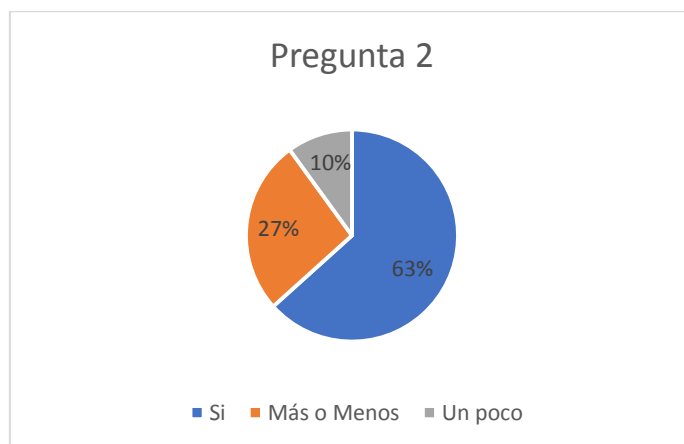


Figura 105. Pregunta 2.

Lo que nos indica los resultados de la pregunta 2 es que la aplicación está cumpliendo con su objetivo de enseñar a los niños las leyes de tránsito, ya que el 63% menciona que si aprendió las leyes de tránsito.

Para la pregunta número 3 se obtuvieron los siguientes resultados. (Ver Figura 106)

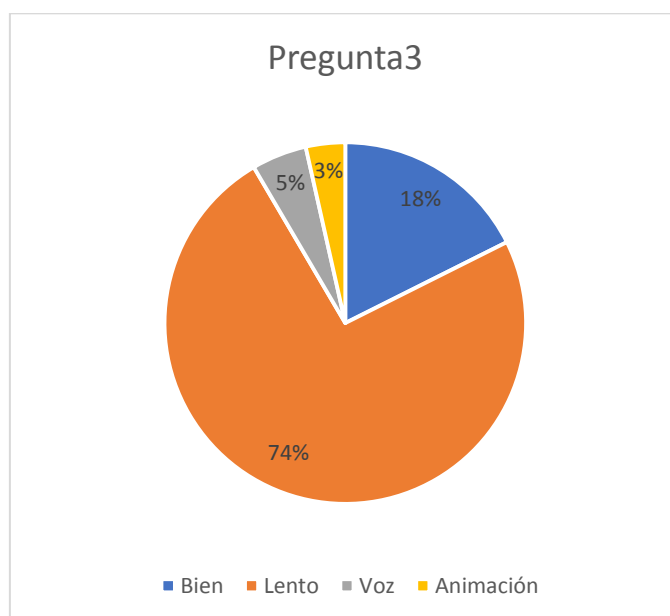


Figura 106. Pregunta 3.

En la pregunta se nota que lo que más disgustó de la aplicación es la lentitud del coche, pero este problema no es del sistema de simulación sino de las computadoras que poseen en la escuela.

A partir de la siguiente pregunta la encuesta se centra en el conocimiento que se obtuvo de la aplicación, es decir, se les hizo una especie de evaluación sobre que aprendieron de la aplicación y que les falta por aprender.

Para la pregunta número 4 se obtuvieron los siguientes resultados. (Ver Figura 107)

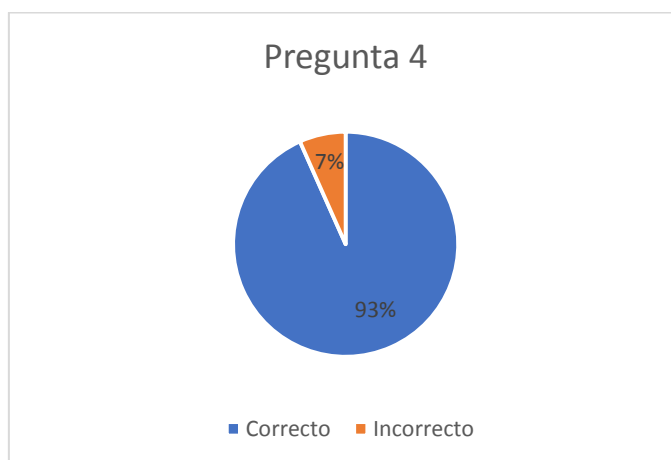


Figura 107. Pregunta 4.

El conocimiento sobre los límites de velocidad, mediante la aplicación fueron comprendidos por los niños, como se muestra en la gráfica de la figura ya que el 93% respondió correctamente esta pregunta, lo que demuestra que la aplicación si les permite a los niños conocer más sobre las leyes de tránsito.

Para la pregunta número 5 se obtuvieron los siguientes resultados. (Ver Figura 108)

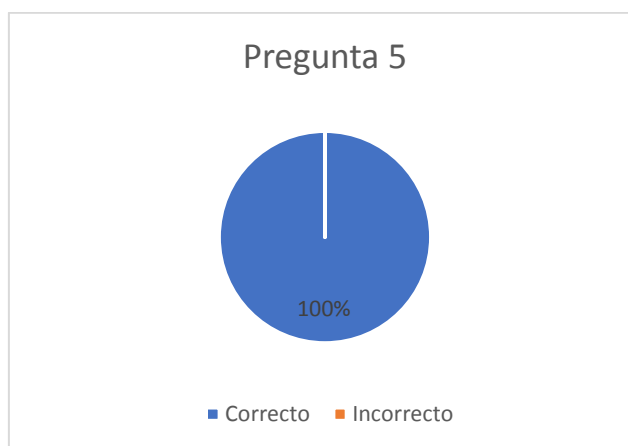


Figura 108. Pregunta 5.

En esta pregunta se mide la comprensión sobre los semáforos por parte de los niños de la escuela, como se puede ver en la Figura 108, conocen la luz de semáforo que les permite avanzar la cual es la luz verde y se observa que la aplicación informa sobre esto.

Para la pregunta número 6 se obtuvieron los siguientes resultados. (Ver Figura 109)

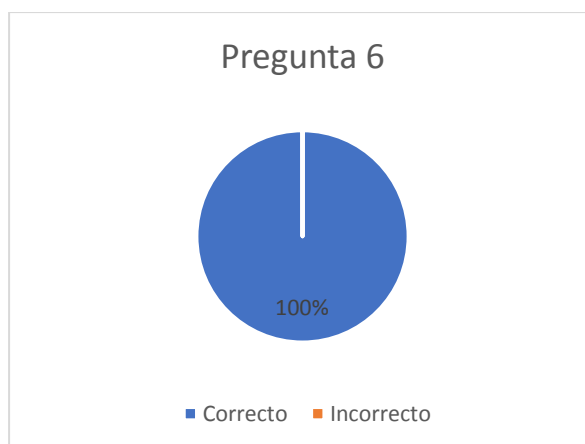


Figura 109. Pregunta 6.

En esta pregunta los niños para responderla utilizaron la información que les mencionaba la aplicación, frases como "que pronto va a cambiar a luz roja" o "si es posible debe detenerse", las cuales se mencionan cuando se encuentran en el camino con un semáforo en rojo.

Para la pregunta número 7 se obtuvieron los siguientes resultados. (Ver Figura 110)

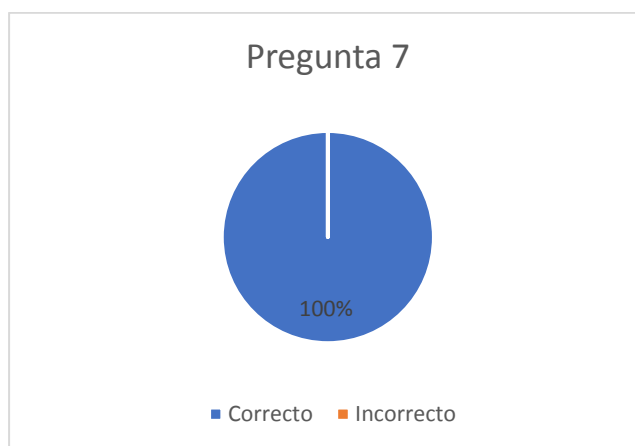


Figura 110. Pregunta 7.

Como se puede observar en la Figura 110, la respuesta de los niños a esta pregunta también cumple el 100%, ya que mediante la aplicación se les explica en audio la información de las señales respectivas por lo que pueden contestar a las preguntas al captar esta información y comprender más sobre los distintos tipos de señales.

Para la pregunta número 8 se obtuvieron los siguientes resultados. (Ver Figura 111)

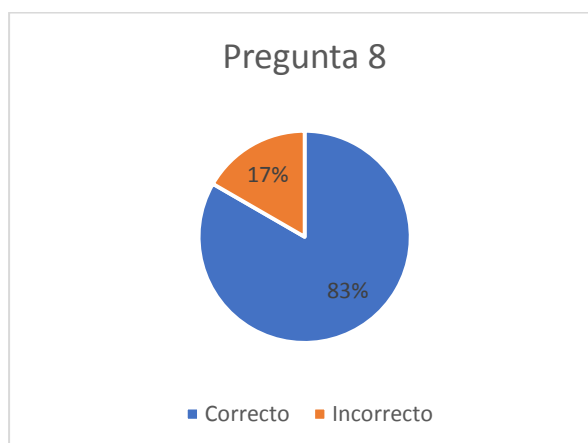


Figura 111. Pregunta 8.

Como se puede observar en la Figura 111, a esta pregunta el 83% contestó correctamente, ya que la aplicación cuenta con los detalles de las señales preventivas y los niños tienen que deducir para que sirven las mismas ya que no existe una

descripción general por tipo de señal de tránsito, pero como se puede observar no hubo problema con esta situación.

Para la pregunta número 9 se obtuvieron los siguientes resultados. (Ver Figura 112)

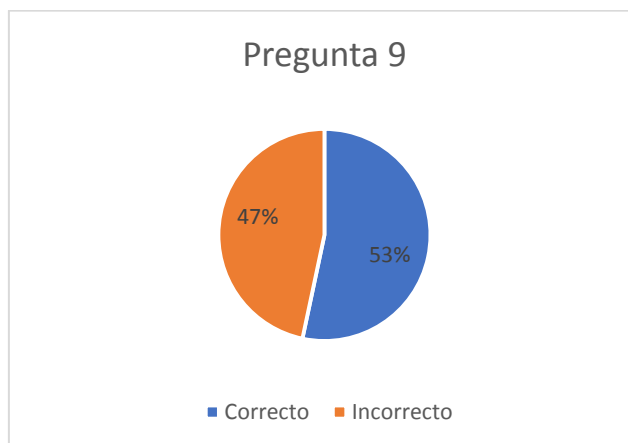


Figura 112. Pregunta 9.

Como se puede ver en la Figura 112, esta pregunta causó ciertos problemas, pero el número de aciertos superó al número de fallos. Una posible explicación puede ser que en la aplicación hay ciertos lugares donde se encuentran estas señales, pero los niños no transitaron por estas para poder responder esta pregunta correctamente.

Para la pregunta número 10 se obtuvieron los siguientes resultados. (Ver Figura 113)

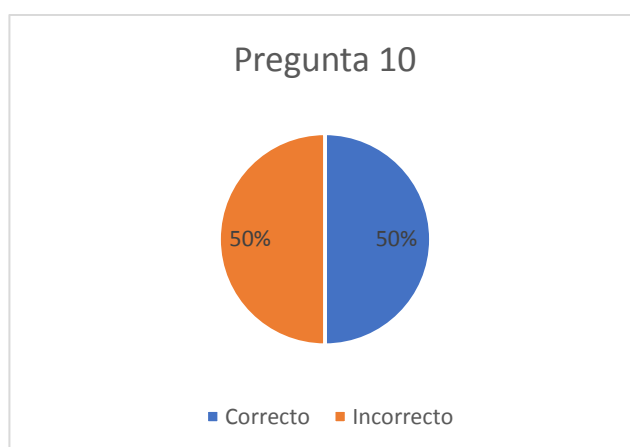


Figura 113. Pregunta 10

Al igual que la pregunta anterior esta pregunta se les dificultó a los niños ya que al no haber visto muchas de estas señales no pudieron elegir una para responder correctamente.

Además se realizó una encuesta a los profesores utilizando una ficha de catalogación y evaluación de programas educativos (Pere Marquès-UAB/2002), como se puede ver en las siguientes tablas: a) Información del Sistema (Ver Tabla 16), b) Aspectos Pedagógicos y Funcionales (Ver Tabla 17), c) Aspectos Técnicos y Estéticos (Ver Tabla 18), d) Recursos Didácticos (Ver Tabla 19) y e) Esfuerzo Cognitivo (Ver Tabla 20)

Tabla 16

Información de la Aplicación.

**FICHA DE SIMPLIFICADA CATALOGACIÓN Y EVALUACIÓN DE PROGRAMAS EDUCATIVOS
Pere Marquès-UAB/2002**

Título del material: Sistema de Simulación 3D sobre Educación Vial Utilizando Técnicas de Inteligencia Artificial.

Caso de Estudio: Escuela “Víctor Manuel Peñaherrera”.

Dirección URL: No aplica.

Autores/Productores: Carlos Proaño

Temática: Juegos Educativos (Educational Games)

Objetivos:

- Revisar el estado del arte de las aplicaciones de multimedia educativa enfocada a la educación vial.
- Realizar el análisis de requerimientos del sistema, aplicando la norma IEEE 830.
- Analizar y seleccionar la metodología para el diseño y desarrollo del sistema de simulación.
- Realizar la implementación y las pruebas del sistema con los niños de la escuela “Víctor Manuel Peñaherrera”.

Contenidos que se tratan:

- Desarrollo Cognitivo del Pensamiento de niños entre 9 y 12 años.
- Desarrollo Psicomotriz de niños entre 9 y 12 años.

Destinatarios: Niños entre 9 y 12 años.

Tipología: Video juego educativo.

Adaptaciones para Colectivos con Necesidades Educativas Especiales: No aplica.

Mapa de navegación y breve descripción de las actividades:

Este juego se compone de un menú en el cual el usuario escoge que rol desea empeñar en el juego, entre las opciones están conductor, peatón o espectador, en el caso de seleccionar la opción de conductor aparecerá un menú donde se debe seleccionar la ruta a conducir, en el caso del peatón se le permitirá seleccionar el personaje, y en el caso del espectador se le permite seleccionar el vehículo a observar, únicamente en el caso del conductor se colocará el puntaje y se podrá almacenar, en los otros dos casos, únicamente se podrá visualizar la ciudad. En la Figura 1, Figura 2 y Figura 3 se encuentran las pantallas principales de cada uno de los roles.



Figura 1. Rol de Conductor.



Figura 2. Rol de Peatón.

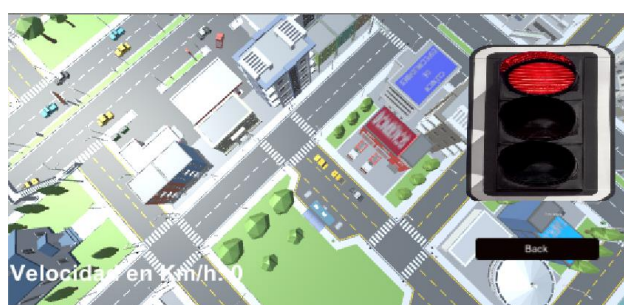


Figura 3. Rol de Espectador.

Documentación: Ninguno.

Servicios on-line: Ninguno.

Requisitos Técnicos: PC con sistema operativo Windows y/o Ubuntu, Sistema Operativo de 32 y/o 64 bits.

Otros: Se debe tener instalado el archivo ejecutable de la aplicación.

Tabla 17

Aspectos Pedagógicos y Funcionales de la Aplicación.

ASPECTOS PEDAGÓGICOS Y FUNCIONALES. (Marcar con una X, donde proceda, la valoración)

DESCRIPCIÓN	EXCELENTE	ALTA	CORRECTA	BAJA
Eficacia didáctica, puede facilitar el logro de sus objetivos.				
Facilidad de instalación y uso.				
Relevancia de los aprendizajes, contenidos.				
Versatilidad didáctica: modificable, niveles, ajustes, informes.				
Considera problemáticas de acceso.				
Capacidad de motivación, atractivo, interés.				
Adecuación a los destinatarios de los contenidos y actividades.				
Potencialidad de los recursos didácticos: síntesis, resumen.				
Tutorización, tratamiento diversidad, evaluación (preguntas de refuerzo).				
Enfoque aplicativo/creativo de las actividades.				
Fomento del auto-aprendizaje, la iniciativa, toma de decisiones.				

Tabla 18*Aspectos Técnicos y Estéticos de la Aplicación.***ASPECTOS TÉCNICOS Y ESTÉTICOS.** (Marcar con una X, donde proceda, la valoración)

DESCRIPCIÓN	EXCELENTE	ALTA	CORRECTA	BAJA
Entorno audiovisual: presentación, pantallas, sonido, letra.				
Elementos multimedia: calidad, cantidad.				
Calidad y estructuración de los contenidos.				
Estructura y navegación por las actividades, metáforas.				
Hipertextos descriptivos y actualizados.				
Interacción con las actividades: diálogo, análisis respuestas.				
Ejecución fiable, velocidad de acceso adecuada.				
Originalidad y uso de tecnología avanzada				

Tabla 19*Recursos Didácticos que Utiliza la Aplicación.***RECURSOS DIDÁCTICOS QUE UTILIZA:** (Marcar uno o más)

<input type="checkbox"/> INTRODUCCIÓN	<input type="checkbox"/> EJERCICIOS DE APLICACIÓN
<input type="checkbox"/> ORGANIZADORES PREVIOS	<input type="checkbox"/> EJEMPLOS
<input type="checkbox"/> ESQUEMAS	<input type="checkbox"/> RESÚMENES/SÍNTESIS
<input type="checkbox"/> GRÁFICOS	<input type="checkbox"/> ACTIVIDADES DE AUTOEVALUACIÓN
<input type="checkbox"/> IMÁGENES	
<input type="checkbox"/> PREGUNTAS	

Tabla 20*Esfuerzo Cognitivo que Exige la Aplicación.***ESFUERZO COGNITIVO QUE EXIGEN LAS ACTIVIDADES DEL PROGRAMA:** (Marcar uno o más)

<input type="checkbox"/> CONTROL PSICOMOTRIZ	<input type="checkbox"/> RAZONAMIENTO (deductivo, inductivo, crítico)
<input type="checkbox"/> MEMORIZACIÓN/EVOCACIÓN	<input type="checkbox"/> PENSAMIENTO
<input type="checkbox"/> COMPRENSIÓN/INTERPRETACIÓN	DIVERGENTE/IMAGINACIÓN
<input type="checkbox"/> COMPARACIÓN/RELACIÓN	<input type="checkbox"/> PLANIFICAR/ORGANIZAR/EVALUAR
<input type="checkbox"/> ANÁLISIS/SÍNTESIS	<input type="checkbox"/> HACER HIPÓTESIS/RESOLVER PROBLEMAS
<input type="checkbox"/> CÁLCULO/PROCESO DE DATOS	<input type="checkbox"/> EXPLORACIÓN/EXPERIMENTACIÓN
<input type="checkbox"/> BUSCAR/VALORAR INFORMACIÓN	<input type="checkbox"/> EXPRESIÓN (verbal, escrita, gráfica, etc.)/CREAR

Se realizó la encuesta a 10 profesores de 5to, 6to y 7mo de básica obteniendo los siguientes resultados. (Ver Figura 114)

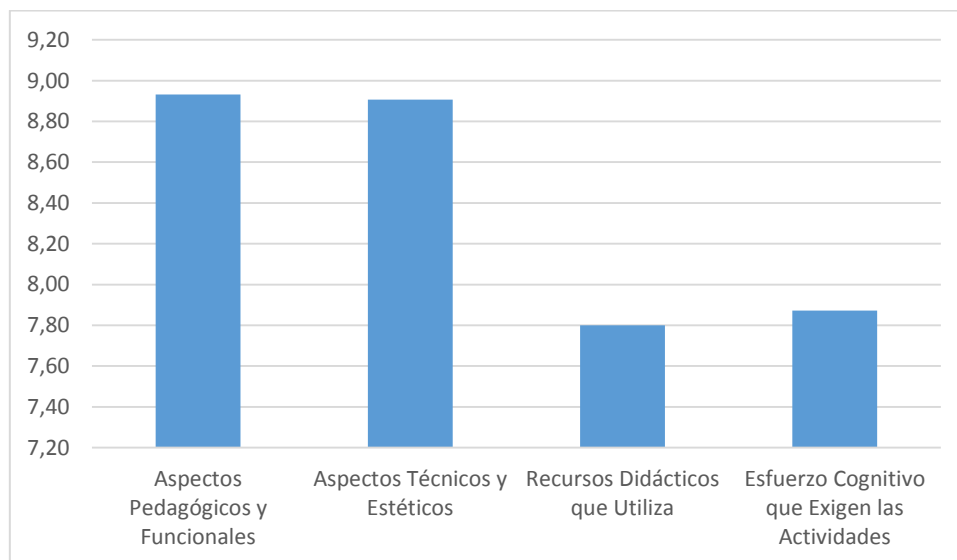


Figura 114. Resultados Encuesta Profesores

De los que se puede observar en la Figura 114 se puede concluir que:

- a) En cuanto a los aspectos pedagógicos y funcionales de la aplicación le falta mejorar la problemática de acceso, obteniéndose un valor de 8,93/10.
- b) En cuanto a los aspectos técnicos y estéticos de la aplicación le falta mejorar el entorno audiovisual, los elementos multimedia, calidad y estructuración de los contenidos, estructura y navegación por las actividades, la interacción con las actividades y la ejecución fiable, obteniéndose un valor de 8,91/10.
- c) En cuanto a los aspectos de recursos didácticos que utiliza la aplicación le falta incluir ejercicios de aplicación, actividades de autoevaluación, resúmenes y síntesis, obteniéndose un valor de 7,8/10.
- d) En cuanto a los aspectos de esfuerzo cognitivo que exigen las actividades de la aplicación le falta contemplar: cálculo/proceso de datos, actividades que fomenten procesos de planificación, organización y evaluación, actividades de resolución de problemas y planteamiento de hipótesis, obteniéndose un valor de 7,87/10.

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

- Se ha realizado una aplicación que permite simular un entorno para educación vial en 3D que integra técnicas de inteligencia artificial (pathfinding y búsqueda de ruta) y cuya utilidad ha sido comprobada en la escuela de niños Víctor Manuel Peñaherrera.
- La simulación de la realidad es demasiado compleja, ya que posee una gran cantidad de elementos (autos, personas, edificios, carreteras, puentes peatonales, subidas y bajadas) y en el desarrollo de la aplicación aparecieron muchos problemas para lograr que los coches actúen en forma inteligente. El funcionamiento de los coches de Unity utiliza el motor de física y se considera que la fuerza que mueve el coche está relacionada a la fricción con el piso. El simple movimiento de un coche sobre una calle representa un problema porque se debe ajustar demasiados parámetros para lograr un funcionamiento aceptable.
- La actuación adecuada de los coches en respuesta al color de la luz de los semáforos es una tarea que fue resuelta mediante el uso de la herramienta de lanzamiento y detección mediante rayos que golpean a los objetos. Pero el movimiento de los coches a pesar de ser simulados no es uniforme y el punto de detección variaba mucho por lo que en un principio no era posible controlar adecuadamente el punto de parada. Para resolver este inconveniente fue necesario elevar el punto de emisión del rayo a una altura que no sería alcanzable por un coche real.
- Como resultado de haber elevado la altura de emisión del rayo produjo como consecuencia algunos problemas al ejecutar el programa pues en algunos casos el rayo golpeaba con los rótulos elevados o con los semáforos o incluso con los puentes, con lo cual se perdía la detección adecuada del punto de parada. La solución a este problema fue eliminar en algunos semáforos la propiedad de *Meshcollider* con lo cual se vuelven transparentes a los rayos.
- Una característica importante que se deseaba lograr es que los coches circulen a una velocidad parecida a la de los coches en la realidad. En la vía perimetral se colocaron límites de velocidad de 90km/h y en las vías internas la velocidad

es de 50km/h o de 30km/h. Para lograr este objetivo era obligatorio realizar un escalamiento de las unidades del mundo real a las del mundo construido. Esta característica fue obtenida gracias a que en Unity las unidades de medida de los objetos están en metros.

- La complejidad del sistema simulado aumenta por la presencia de situaciones que no se consideran inicialmente como la presencia de ascensos y descensos pronunciados. Un problema que aparece con esto es que los coches no se detienen a pesar de que se coloque el máximo torque de frenado (*BrakeTorque*). Frente a esto la solución es aumentar el rozamiento de la rueda con el piso (*Stiffness*). Otro problema con el ascenso es que los coches disminuyen su velocidad y esto produjo que el coche posterior se frene y luego no tenga la inercia suficiente para subir la cuesta.
- En los descensos en cambio los coches aumentan la velocidad y por tanto se presenta la posibilidad de que colisionen con los coches de adelante. Para resolver este problema se colocaron señales de límite de velocidad a 30km/h justo antes de alcanzar los descensos. Estas señales son leídas por los coches autónomos para disminuir su velocidad.
- Las curvas generan otro problema por el poco espacio que puede existir en el carril al cual desea acceder el coche, por esta razón en la aplicación se generó un retardo de tiempo para detectar si existe el espacio suficiente antes de realizar la curva.
- También debe mencionarse que la aplicación ha sido realizada sobre un modelo de ciudad que tiene ciertas características que facilitan el desarrollo de la aplicación. A este respecto la principal característica que facilitó el desarrollo de la aplicación es que las calles son todas verticales u horizontales. Otra característica importante es que no hay redondeles. Para limitar la complejidad del sistema se utilizaron coches de tamaño similar.
- Para la simulación de los peatones se presentaba un problema de complejidad por el número de peatones que debían cobrar vida. Si se considera que lo que se simula es una ciudad, esto significa que debe tener muchos peatones en las vías. Debido a que el objetivo del proyecto era un sistema de simulación para educación vial se decidió que era conveniente simular los peatones únicamente

en los sitios donde existen semáforos y en los pasos peatonales. Los peatones automáticos aparecen únicamente el momento en que están habilitados para cruzar la calle.

- La aplicación ha sido probada por los niños de la Escuela “Víctor Manuel Peñaherrera” y el principal problema que se indica de la aplicación es que es lenta. Esta lentitud debe atribuirse a las computadoras utilizadas para realizar la prueba lo cual se podría solucionar añadiendo una tarjeta de video que no estaba incluido en el presupuesto de este proyecto.
- La utilidad de la aplicación se refleja en los resultados de las encuestas. Algo que debe destacarse es que la información audible y repetida se almacena con facilidad en la mente de los aprendices. En respuesta a la pregunta sobre la señal Pare algunos alumnos repitieron las palabras que se enuncian en la aplicación.
- Con respecto a la clasificación de las señales de tránsito en informativas, preventivas y regulatorias se observa que el estudiante da poca importancia a esta información pues a pesar de que todas las señales anuncian inicialmente el tipo de señal, así también la imagen con el color respectivo, esta información les aparece dispersa y los niños no logran relacionar el tipo de señal con la categoría de uso y de igual forma el tipo de señal con el color.
- Unity es una excelente herramienta para el desarrollo de sistemas de simulación y videojuegos educativos, por su flexibilidad y fácil uso.
- Entre las herramientas de Unity utilizadas tenemos la máquina de estados la cual nos permite en los vehículos automáticos cambiar de estado, mediante condiciones.
- Otra herramienta de Unity son los paths que permiten tener una visión de la ruta que deben seguir los carros automáticos para manejarse por la ciudad.
- La norma ISO 830 ayuda a documentar adecuadamente los requerimientos funcionales y no funcionales del sistema en base a los requerimientos de los usuarios.
- La Metodología Huddle posee una plantilla de documento la cual permite comprender la forma de diseñar e implementar un sistema de simulación (videojuego serio).

RECOMENDACIONES

- Se recomienda que se utilice al menos cuatro artefactos de los doce diagramas manejados por la versión 2.0 de UML, que son diagramas de casos de uso, diagramas de secuencia, diagramas de clases y diagramas de componentes.
- Se recomienda que en todo proyecto de desarrollo de sistemas de simulación (videojuegos serios), se utilice una metodología que permita documentar adecuadamente el sistema y obtener productos razonables en tiempos no extensos.
- Se recomienda que se analice la magnitud del proyecto en detalle antes de emprender con la realización. Debe realizarse una investigación previa sobre la factibilidad del proyecto. Es decir el tiempo, los recursos económicos, la dificultad del proyecto, para lograr cumplir con el objetivo.
- Se recomienda que se realice pruebas de la aplicación para revisar el funcionamiento adecuado de la misma para su entrega final.

REFERENCIAS BIBLIOGRÁFICAS

- Anuario De Estadística De Transporte 2016 Resumen ... (s. f.). Recuperado 24 de enero de 2019, de
https://www.librarybus.com/view?t=ANUARIO+DE+ESTAD%C3%8DSTICA+DE+TRANSPORTE+2016+Resumen+...&u=http%3A%2F%2Fwww.ecuadorencifras.gob.ec%2Fdocumentos%2Fweb-inec%2FEstadisticas_Economicas%2FEstadistica%2520de%2520Transporte%2F2016%2F2016_Anuario%2520de%2520Transporte_Resumen%2520Ejecutivo.pdf
- Arceo, F. D. B. (2003). Cognición situada y estrategias para el aprendizaje significativo. *Revista Electrónica de Investigación Educativa*, 5(2). Recuperado de
<https://redie.uabc.mx/redie/article/view/85>
- Arias Ramos, N. M. (2015). *Análisis y diseño de una página web interactiva orientada a niños (6 hasta 12 años) de educación básica, que permita el aprendizaje de los reglamentos de tránsito y leyes de educación en seguridad vial*. (Thesis, Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas. Carrera de Ingeniería en Sistemas Computacionales). Recuperado de
<http://repositorio.ug.edu.ec/handle/redug/9945>
- Aroca, Á. (2012, marzo 15). Unity 3D, desarrollo de videojuegos para iOS y Android, gratis hasta el 8 de Abril. Recuperado 27 de agosto de 2018, de Genbeta website:
<https://www.genbeta.com/desarrollo/unity-3d-desarrollo-de-videojuegos-para-ios-y-android-gratis-hasta-el-8-de-abril>
- Audacity. (s. f.). Recuperado 22 de enero de 2019, de Audacity website:
<http://audacity.org.es/>

- Audacity, programa para editar sonido digital. Software Libre. (2006, marzo 30). Recuperado 27 de agosto de 2018, de Aula Intercultural website:
<https://aulaintercultural.org/2006/03/30/audacity-programa-para-editar-sonido-digital-software-libre/>
- Calleros, J. (2019). *Manual UML*. Recuperado de
https://www.academia.edu/38220802/Manual_UML
- César Coll. (2010). Enseñar y aprender en el mundo actual: desafíos y encrucijadas. *Pensamiento Iberoamericano*. Recuperado de http://www.psyed.edu.es/prodGrintie/articulos/CC_2010_PensamientoIberoamericano.pdf
- Chan, M. T., Chan, C. W., & Gelowitz, C. (2016). Development of a Car Racing Simulator Game Using Artificial Intelligence Techniques. *Int. J. Comput. Games Technol.*, 2015, 12:12–12:12. <https://doi.org/10.1155/2015/839721>
- Ciencias Sociales – Ministerio de Educación. (s. f.). Recuperado 27 de agosto de 2018, de <https://educacion.gob.ec/curriculo-ciencias-sociales/>
- David Istance. (2012). *Crear entorns innovadors per millorar l'aprenentatge*. (Debats d'Educació). Universitat Oberta de Catalunya. Fundació Jaume Bofill. Museu d'art Contemporani de Catalunya.
- David Poole. (s. f.). *Computational Intelligence and Knowledge*. Recuperado de <https://www.cs.ubc.ca/~poole/ci/ch1.pdf>
- El desarrollo de juegos con la herramienta Unity 3D. (s. f.). Recuperado 22 de enero de 2019, de <https://www.yeeply.com/blog/desarrollo-de-juegos-con-unity-3d/>
- Eugene. Wieggers, K. (2003). *Software Requierements : Practical Techniques for Gathering and Managing Requirements Throughout the Product Development Cycle* K.E. Wieggers.

Francisco Mora. (2013). *Neuroeducación. Solo se puede aprender aquello que se ama.*

Editorial Alianza.

Gestor de citas bibliográficas Zotero - Universidad de Chile. (s. f.). Recuperado 22 de enero de 2019, de <http://www.uchile.cl/portal/informacion-y-bibliotecas/ayudas-y-tutoriales/95186/zotero>

Gutiérrez Utande, P. L. (2017). Aplicación de inteligencia artificial en videojuegos : uso de la variante del algoritmo Minimax poda alpha-beta para su desarrollo [Info:eu-repo/semantics/bachelorThesis]. Recuperado 25 de enero de 2019, de <http://oa.upm.es/48910/>

H. Antonio Jiménez R. (2011). Guia de Audacity. Recuperado 22 de enero de 2019, de <http://www.telmexeducacion.com/proyectos/AulaCiencia/Forms/DispForm.aspx?ID=2&Source=http%3A%2F%2Fwww%2Etelmexeducacion%2Ecom%2Fproyectos%2FAulaCiencia%2FForms%2FAllItems%2Easpx&RootFolder=%2Fproyectos%2FAulaCiencia>

Hendricks, C. C. (2001). Teaching Causal Reasoning Through Cognitive Apprenticeship: What are Results from Situated Learning? *Journal of Educational Research*, 94(5), 302-311.

Leañez, O. (2016, mayo 22). El arte de los videojuegos: ¿Que es Unity y para que sirve?

Recuperado 22 de enero de 2019, de El arte de los videojuegos website:

http://theunitydeveloper.blogspot.com/2016/05/que-es-unity-y-para-que-sirve_21.html

MakeHuman - EcuRed. (s. f.). Recuperado 27 de agosto de 2018, de

<https://www.ecured.cu/MakeHuman>

Mark Prensky. (2011). *Enseñar a nativos digitales*. Madrid, España: Ediciones SM.

Richard Gerver. (2012). *Crear hoy la escuela del mañana. La educación y el futuro de nuestros hijos*. Ediciones SM.

Spiritu3.com. (2015, agosto 26). Qué es y Para qué sirve Unity 3D. Recuperado 22 de enero de 2019, de Agencia de Publicidad 2.0: Spiritu3.com website:
<https://spiritu3.com/para-que-sirve-unity-3d/>

Stellman, A., & Greene, J. (2005). *Applied Software Project Management*.

Telégrafo, E. (2014, diciembre 3). Reforma creará cátedra de educación vial en colegios. Recuperado 27 de agosto de 2018, de El Telégrafo website:
<https://www.elselegrafo.com.ec/noticias/justicia/1/reforma-creara-catedra-de-educacion-vial-en-colegios>

Telégrafo, E. (2018, marzo 12). La seguridad vial es responsabilidad de todos. Recuperado 23 de agosto de 2018, de El Telégrafo website:
<https://www.elselegrafo.com.ec/noticias/septimo/1/la-seguridad-vial-es-responsabilidad-de-todos>

Torres, R. (s. f.). Guías temáticas: Gestores de referencias bibliográficas: Zotero. Recuperado 22 de enero de 2019, de
[//guiastematicas.bibliotecas.uc.cl/gestores_bibliograficos/zotero](http://guiastematicas.bibliotecas.uc.cl/gestores_bibliograficos/zotero)

Tutoriales gratis de Makehuman en español. (s. f.). Recuperado 22 de enero de 2019, de
<https://megatutos.com/html/megatutos/section/makehuman>

Ullastres, Á. M. (2009). Las Metas Educativas 2021. Un proyecto iberoamericano para transformar la educación en la década de los bicentenarios. *CTS: Revista iberoamericana de ciencia, tecnología y sociedad*, 4(12), 7.

Zotero: Guía básica. (s. f.). 6.