

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA UNAN-MANAGUA



Facultad De Ciencias E Ingeniería

Departamento de Computación

Tema: Programación de videojuegos

Subtema: Programación del video juego Nica-Aventure en 3D para dispositivos móviles bajo la plataforma Androide en el primer semestre de 2013.

Seminario de graduación para la obtención del título de Lic. En Ciencias De la Computación.

Tutor (a): Msc.Juan de Dios Bonilla Anduray

Integrantes:

- Br. José Luis Álvarez
- Br. Marcela Eugenia Chávez Guillén

Managua, Nicaragua

Fecha de entrega: 08 de noviembre de 2013

DEDICATORIAS

Este seminario lo dedico a mi Dios quien supo guiarme por el buen camino, darme fuerzas para seguir adelante y no desmayar en los problemas que se presentaban, enseñándome a encarar las adversidades sin perder nunca la dignidad ni desfallecer en el intento.

A mi familia, quienes por ellos soy lo que soy. Para mis padres por su apoyo, consejos, comprensión, amor y ayuda en los momentos más difíciles, a mis hermanos por ser partícipes de cada evento en mi vida, a José Luis Álvarez por confiar siempre en mí y recordarme que trabajando juntos como un equipo y con fé en DIOS todo es posible.

Marcela Eugenia Chávez Guillén

DEDICATORIAS

Dedico este seminario a Dios primeramente por brindarme la oportunidad y la dicha de la vida, al brindarme los medios necesarios para continuar mi formación, a mi madre quien me ha apoyado y guiado a lo largo de mi vida. A mi novia Marcela quien me apoyo y alentó para continuar, cuando parecía que me iba a rendir. Sin ellas no hubiese podido concluir este ciclo de mi vida y tampoco podría convertirme en un verdadero profesional.

José Luis Álvarez

AGRADECIMIENTOS

Este trabajo está dedicado a toda mi familia, por su cariño y porque siempre me han apoyado. Gracias principalmente a mis padres Cándida Guillen y Eugenio Chávez: quiero que sepan que para mí son los mejores ya que me enseñaron que puedo llegar hasta donde me lo proponga, a mis hermanos por su apoyo incondicional, a José Luis Álvarez por estar todo el tiempo apoyándome pensando siempre en cómo materializar cada idea que pasa por mi mente, gracias por la ayuda que de ustedes maestros y compañeros he recibido; éste trabajo es solo una pequeña muestra de lo mucho que les agradezco por la educación que me han brindado.

Gracias a DIOS, por ser la fuerza que me impulsa a seguir adelante.

Marcela Eugenia Chávez Guillén

AGRADECIMIENTOS

Gracias a Dios.

A mi madre, Maritza Álvarez, que siempre me han dado su apoyo incondicional. A mis maestros quienes me guiaron en este camino de estudios y aprendizaje, por todo su trabajo y dedicación para enseñarme una profesión. De todos ustedes es este triunfo y para ustedes es todo mi agradecimiento.

José Luis Álvarez

Contenido

INDICE DE FIGURAS.....	10
INDICE DE TABLAS.....	12
I. RESUMEN.....	13
II. INTRODUCCION.....	14
III. ANTECEDENTES.....	15
IV. JUSTIFICACIÓN.....	16
V. OBJETIVOS.....	17
5.1. Objetivo General.....	17
5.2. Objetivos Específicos.....	17
VI. MARCO CONCEPTUAL.....	18
6.1. Programación.....	18
6.2. Lenguaje de programación.....	18
6.2.1. Programación orientada a objeto (POO).....	18
6.2.2. Software para desarrollar.....	20
6.2.2.1. Las herramientas de desarrollo y modelado necesarias en el proyecto.....	21
6.2.2.1.1. UNITY 3D 2014.....	21
6.2.2.1.2. 3Ds MAX 2014.....	21
6.2.2.1.3. ADOBE PHOTOSHOP CS6.....	23
6.2.2.1.4. MOVIE MAKER.....	25
6.3. Videjuego.....	26
6.4. Programación de videojuegos.....	26
6.5. Géneros de juegos.....	26
6.6. Clasificación de Juegos según el número de jugadores.....	29
6.7. Plataformas para los videojuegos:.....	30
6.7.1. Historia de la telefónica móvil.....	31
6.7.1.1. Tendencias móviles.....	32
6.7.1.2. Sistemas Operativos móviles.....	32
6.7.2. Dispositivos Móviles.....	35
6.7.3. Android.....	36
6.7.3.1. El origen de Android.....	36
6.7.3.2. Las actualizaciones.....	37

6.7.3.3.	Versiones de Android.....	37
6.7.3.4.	Ventajas.....	39
6.7.3.5.	Características.....	39
6.7.3.6.	Arquitectura de Android.	42
6.8.	Áreas involucradas en el desarrollo de un videojuego.....	49
6.9.	El efecto de inmersión	52
6.9.1.	Modelo de experiencia en el Gameplay: inmersión en los videojuegos.	53
6.9.2.	Implicancia del diseño gráfico en el efecto de inmersión en el videojuego:.....	55
6.10.	Transmitiendo identidad a través de los videojuegos:	57
6.11.	Ambientes Gráficos	57
6.11.1.	Graficas 2d de computadoras	58
6.11.2.	Gráficos de ordenador en 3D.....	59
6.11.2.1.	Historia.....	60
6.11.2.2.	Panorama general.....	60
6.11.2.3.	Modelado	60
6.11.2.4.	Animación	61
6.11.2.5.	Rendering	61
6.11.2.6.	Comunidades	62
6.11.2.7.	Distinción de gráficos 2D fotorrealistas.....	62
6.12.	Metodología SUM para Videojuegos	62
VII.	DISEÑO METODOLOGICO	66
7.1.	Tipo de estudio	66
7.2.	Metodología	66
7.2.1.	Fases de Pre-producción.....	66
7.2.2.	Fase de Producción.....	66
7.2.2.1.	Diseño del Juego	66
7.2.2.1.1.	Diseño artístico	67
7.2.2.1.2.	Diseño Mecánico.....	67
7.2.2.1.3.	Motor del juego	67
7.2.2.2.	Diseño Técnico	67
7.2.2.3.	Implementación	67
7.2.2.4.	Pruebas Alpha	68
7.2.2.5.	Pruebas Beta.....	68

7.2.3.	Fase de Post Producción	69
7.3.	Estudio de factibilidad	70
7.3.1.	Factibilidad Técnica	70
7.3.2.	Factibilidad Económica	70
7.3.3.	Factibilidad Operativa	71
VIII.	DESARROLLO	72
8.1.	RESULTADO 1: FASE DE PRE – PRODUCCION.....	72
8.2.	RESULTADO 1: FASE DE PRODUCCION.....	73
8.2.1.	DISEÑO DEL JUEGO	73
8.2.1.1.	Diseño Artístico - Guion del juego.....	73
8.2.1.2.	Diseño Mecanico - Guión del juego para el documento de diseño	77
8.2.1.3.	Motor del videojuego.....	78
8.2.2.	DISEÑO TÉCNICO	79
8.2.2.1.	Diagrama de casos de uso generalizado	79
8.2.2.2.	Detalle de casos de usos	80
8.2.2.3.	Diagrama de clases simplificado.....	83
8.2.2.4.	Detalle de clases	84
8.2.2.5.	Descripción de las clases.....	87
8.2.2.6.	Diagramas de secuencia	89
8.2.2.7.	Cronograma de Trabajo.....	92
8.2.3.	IMPLEMENTACIÓN	100
8.2.3.1.	Primer Incremento	100
8.2.3.1.1.	Librería jPCT-AE	100
8.2.3.1.2.	Cámara	100
8.2.3.1.3.	Escenario	101
8.2.3.1.4.	Modelado	102
8.2.3.1.5.	Personajes	102
8.2.3.1.6.	Objetos.....	105
8.2.3.2.	Segundo Incremento.....	108
8.2.3.2.1.	Cámara	108
8.2.3.2.2.	Objetos.....	108
8.2.3.2.3.	Texturas	114
8.2.3.2.4.	Iluminación.....	114

8.2.3.2.5. Gravedad	114
8.2.3.2.6. Colisiones	115
8.2.3.2.7. Sonido.....	115
8.3. RESULTADO 2: ESCENARIOS DINÁMICOS E INTERACTIVOS CON EL USUARIO.	116
8.4. RESULTADO 3: TÉCNICA DE PROGRAMACIÓN UTILIZADA.	120
8.5. RESULTADO 4: PROMOVER LA CULTURA NICARAGÜENSE POR MEDIO DE ESCENARIOS QUE PERMITAN DISFRUTAR LA OFERTA TURÍSTICA DE LA RUTA DE LOS TESOROS.....	123
IX. CONCLUSIONES	125
X. RECOMENDACIONES.....	126
XI. BIBLIOGRAFÍA.....	127

INDICE DE FIGURAS

<i>Figura. 1 Unity 3D</i>	21
<i>Figura. 2 Motorola DynaTAC 8000X.</i>	31
<i>Figura. 3 Dispositivos Móviles con sistema Operativo Android (web, 2011)</i>	35
<i>Figura. 4 Versiones de Android. (web, 2011)</i>	39
<i>Figura. 5 Kernel de Linux. (web, 2011)</i>	43
<i>Figura. 6 Librerías y ejecución. (web, 2011)</i>	43
<i>Figura. 7 Estructura de Aplicaciones. (web, 2011)</i>	44
<i>Figura. 8 Aplicaciones. (web, 2011)</i>	44
<i>Figura. 9 Componentes Fundamentales que cumplen un rol en la experiencia del Gameplay (web, 2011)</i>	55
<i>Figura. 10 Renderizado 3D con trazado de rayos y oclusión ambiental (licuadora y YafaRay).</i>	61
<i>Figura. 11 Modelo 3d de un Dunkerque clase acorazado representa con sombreado plano.</i>	62
<i>Figura. 12 Fases del proceso de la metodología SUM.</i>	65
<i>Figura. 13 Nicaragua turística (CRECER12, 2013)</i>	73
<i>Figura. 14 El empresario Enrique Zamora Llanes, el expelotero de Grandes Ligas, Denis Martinez, junto a representantes de American Airlines y de la Camará Nacional de Turismo, en el lanzamiento de la Ruta de los Tesoros de Nicaragua (Ricardo., 2011)</i>	74
<i>Figura. 15 Imágenes de la ruta de los tesoros (Ricardo., 2011)</i>	75
<i>Figura. 16 El empresario Enrique Zamora Llanes propietario de Nicarao Lake Resort (Ricardo., 2011)</i>	75
<i>Figura. 17 Pasaporte de visitante de la ruta de los tesoros (Ricardo., 2011)</i>	76
<i>Figura. 18 Imágenes de la ruta de los tesoros (Ricardo., 2011)</i>	77
<i>Figura. 19 Diagrama de casos de usos generalizado</i>	79
<i>Figura. 20 Diagrama de clases generalizado.</i>	83
<i>Figura. 21 Clases S_CARPICKUP, CASAS, TEMPORIZADOR, ARBOL, CHECKPOINT</i>	84
<i>Figura. 22 Clases OBSTACULO_BARRIL, VEHICULO, REDUCTOR_VELOCIDAD, DIRECTIONAL_LIGHT.</i>	85
<i>Figura. 23 Clases MANAGUA_GRANADA, TERRENO.</i>	86
<i>Figura. 24 Diagrama de secuencia 1</i>	89
<i>Figura. 25 Diagrama de secuencia 2</i>	89
<i>Figura. 26 Diagrama de secuencia 3</i>	90
<i>Figura. 27 Diagrama de secuencia 4</i>	90
<i>Figura. 28 Diagrama de secuencia 5</i>	91
<i>Figura. 29 Diagrama de secuencia 6</i>	91
<i>Figura. 30 Portada del juego.</i>	102
<i>Figura. 31 Vistas frontal, lateral, superior de Nicaboy sin textura</i>	104
<i>Figura. 32 Vistas frontal y lateral de Nicaboy con textura</i>	104

<i>Figura. 33</i>	<i>Vistas en perspectiva de Nicaboy.</i>	<i>105</i>
<i>Figura. 34</i>	<i>Vistas del terreno.</i>	<i>105</i>
<i>Figura. 35</i>	<i>Modelado de la carretera Managua-Granada.</i>	<i>106</i>
<i>Figura. 36</i>	<i>Vistas en perspectiva de la carretera.</i>	<i>107</i>
<i>Figura. 37</i>	<i>Cámara en la parte trasera del personaje.</i>	<i>108</i>
<i>Figura. 38</i>	<i>Objeto árbol y los materiales aplicados.</i>	<i>109</i>
<i>Figura. 39</i>	<i>Objeto árbol caído una variante de la clase árbol utilizado como obstáculo.</i>	<i>109</i>
<i>Figura. 40</i>	<i>Objeto Checkpoint en dos distintas capturas.</i>	<i>110</i>
<i>Figura. 41</i>	<i>Objeto Checkpoint en dos distintas capturas</i>	<i>111</i>
<i>Figura. 42</i>	<i>Objeto casas ubicado sobre el terreno con texturas.</i>	<i>111</i>
<i>Figura. 43</i>	<i>Texturas disponibles para el objeto casa y sus dos variantes.</i>	<i>112</i>
<i>Figura. 44</i>	<i>Texturas disponibles para el objeto casa y sus dos variantes.</i>	<i>112</i>
<i>Figura. 45</i>	<i>Objeto reductor de velocidad y su modelado.</i>	<i>113</i>
<i>Figura. 46</i>	<i>Objeto camioneta descompuesta y su modelado.</i>	<i>113</i>
<i>Figura. 47</i>	<i>NicaBoy en cuenta Regresiva antes de iniciar la carrera.</i>	<i>116</i>
<i>Figura. 48</i>	<i>Modelado del ambiente ubicando objetos casas para causar un efecto inmersivo en el jugador.</i>	<i>117</i>
<i>Figura. 49</i>	<i>NicaBoy en Marcha con velocímetro en pantalla.</i>	<i>117</i>
<i>Figura. 50</i>	<i>NicaBoy atravesando un reductor de velocidad.</i>	<i>118</i>
<i>Figura. 51</i>	<i>NicaBoy acabando de pasar por un CHECKPOINT.</i>	<i>118</i>
<i>Figura. 52</i>	<i>NicaBoy perdido en la montaña navegando el ambiente.</i>	<i>119</i>
<i>Figura. 53</i>	<i>NicaBoy corriendo en dos ruedas luego de impactar un obstáculo.</i>	<i>119</i>
<i>Figura. 54</i>	<i>Código de la clase playcar programado en java script.</i>	<i>120</i>
<i>Figura. 55</i>	<i>Código de la clase Cronometro programado en java script.</i>	<i>121</i>
<i>Figura. 56</i>	<i>Código de la clase Checkpoints programado en java script.</i>	<i>121</i>
<i>Figura. 57</i>	<i>Código de la clase Cuenta regresiva programado en java script.</i>	<i>122</i>
<i>Figura. 58</i>	<i>Breve Reseña Histórica de Granada.</i>	<i>123</i>
<i>Figura. 59</i>	<i>Video en formato MP4 con la oferta turística de granada.</i>	<i>124</i>
<i>Figura. 60</i>	<i>Pasaporte de visitante otorgado al jugador que ha realizado el recorrido Managua-granada.</i>	<i>124</i>

INDICE DE TABLAS

<i>Tabla 1. Factibilidad económica de software/licencias.</i>	<i>70</i>
<i>Tabla 2 Factibilidad económica de dispositivos y servicios.</i>	<i>71</i>
<i>Tabla 3 Caso de uso SELECCIONAR OPCION MENU.</i>	<i>80</i>
<i>Tabla 4 Caso de uso ARRANCAR AUTO.</i>	<i>80</i>
<i>Tabla 5 Caso de uso ACELERAR AUTO.</i>	<i>81</i>
<i>Tabla 6 Caso de uso ESQUIVAR OBSTACULOS.</i>	<i>81</i>
<i>Tabla 7 Caso de uso PAUSA.</i>	<i>82</i>
<i>Tabla 8 Caso de uso REGRESAR AL MENU.</i>	<i>82</i>

I. RESUMEN

El mundo de los videojuegos hoy en día es un campo en constante desarrollo. La posibilidad de aplicar los videojuegos para potencializar nuestra cultura de una forma amena es uno de los motivos que nos ha llevado a seleccionar este tema.

En la introducción de esta investigación se brinda una idea general del tema que se profundizará en el estudio.

Los antecedentes de esta investigación le mostrarán información de estudios previos a nuestra investigación.

La ruta de los tesoros de Nicaragua es la base de la historia que forma el guion de nuestro juego. Esta se describe en la sección de preproducción en este documento.

La metodología implementada en esta investigación y la factibilidad del videojuego se describen en la sección de Diseño Metodológico.

En la sección de producción del videojuego se muestran diagramas y gráficos de resultados producto de la aplicación de las etapas y herramientas de la metodología de desarrollo de videojuegos seleccionada para el presente estudio.

Para terminar, se presentan las conclusiones de nuestra investigación las cuales muestran el cumplimiento de los objetivos y los resultados obtenidos. Cabe señalar que brindamos recomendaciones para ejecutar el videojuego, así como la bibliografía que consultamos para el desarrollo de nuestra investigación.

II. INTRODUCCION

En la actualidad la creciente demanda de información a llevado al desarrollo de las tecnologías en dispositivos móviles, especialmente los teléfonos celulares, haciendo que estos pasen de ser sólo utilizados para hacer o recibir llamadas, a aparatos en los cuales se puede realizar infinidad de tareas, como: leer el diario, mandar un mail, llevar una agenda, jugar, y muchas más funcionalidades, que van en aumento junto con la tecnología que ofrecen, como pantalla táctil, reconocimiento por voz, captura de video, etc.

Lo anteriormente mencionado nos lleva a concluir que las funcionalidades ofrecidas por los dispositivos móviles se asemejan cada día más a las que ofrece un computador moderno, permitiendo incluso la instalación de programas externos. Es en este punto donde se amplía el público al que apunta el desarrollo de software, esto es debido principalmente a la portabilidad y comodidad que ofrecen estos dispositivos, permitiendo el acceso a la información y entretenimiento en cualquier momento y lugar.

Aprovechando el auge y capacidades de los dispositivos móviles se ha comenzado a desarrollar un amplio mercado de aplicaciones para estos, entre ellos los videojuegos, que demuestran de forma gráfica e interactiva las posibilidades que ofrecen.

El desarrollo de un videojuego implica una combinación armónica entre el arte y la tecnología, siendo necesario para su creación amplios conocimientos computacionales y de diseño gráfico, llegando a considerarse hoy en día por algunos entendidos como un medio para el arte, ya que en ellos se expone música, videos, diseños, tanto de personajes como escenarios.

El proyecto presentado constituye una indagación en el desarrollo de videojuegos y su aplicación a las nuevas tecnologías, como lo son las plataformas móviles, en este caso, el popular sistema operativo de Google, Android; aprovechando el aumento en las capacidades de estos equipos, especialmente el procesamiento de gráficos en tres dimensiones, las características propias de los equipos modernos, como las pantallas táctiles y la demanda que existe en el campo de videojuegos.

III. ANTECEDENTES

En Nicaragua el desarrollo de videojuegos es un área de la computación muy pero muy poco explorada, se ha sabido de talleres y pequeños proyectos en los cuales se aborda este tema tal es el caso del “Taller: Diseño colectivo de un videojuego con herramientas Open Source” en agosto del 2012 organizado por el centro cultural de España en Nicaragua, el cual consistió en el diseño de un videojuego hasta la realización de un prototipo jugable.

Pero más allá de esto no se cuenta con empresas u organizaciones dedicadas al desarrollo de videojuegos, debido tal vez a los requerimientos de tiempo y dinero que esto amerita.

IV. JUSTIFICACIÓN

Nicaragua al igual que muchos es un país con un alto índice de demanda en juegos de video ya que estos los podemos encontrar en consola (play station I,II,III, wii, Xbox 360 etc.), juegos para computadoras, para celulares, en internet o en línea, existiendo varios sitios de distribución masiva legal e ilegal de estos, pero si nos damos un tiempo para analizar somos un país inmerso en el consumismo de tecnología y sobre todo de videojuegos, es por esta razón que hemos decidido iniciarnos en este mundo de producción de videojuegos con el fin de promover el desarrollo de esta rama de la informática que ha sido poco explorada.

V. OBJETIVOS

5.1. Objetivo General

Desarrollar el videojuego nica-aventure en 3D para móvil en el primer semestre de 2013 en el curso de seminario de graduación de la UNAN-Managua.

5.2. Objetivos Específicos

- 5.2.1.** Elaborar el Documento de diseño del Videojuego Nica-Adventure.
- 5.2.2.** Elaborar escenarios dinámicos e interactivos con el usuario.
- 5.2.3.** Utilizar la POO en el desarrollo del Videojuego Nica-Adventure.
- 5.2.4.** Promover la cultura nicaragüense por medio de escenarios que permitan disfrutar la oferta turística de la ruta de los tesoros.

VI. MARCO CONCEPTUAL

El presente se trata de proporcionar la información más relevante de los temas abarcados en este trabajo.

6.1. Programación

Es el proceso de diseñar, codificar, depurar y mantener el código fuente de programas computacionales. El código fuente es escrito en un lenguaje de programación. El propósito de la programación es crear programas que exhiban un comportamiento deseado.

En un plano más técnico, la programación se realiza mediante el uso de algoritmos, que son secuencias finitas, ordenadas y no ambiguas de instrucciones que deben seguirse para resolver un problema. (Wikipedia., 2013)

6.2. Lenguaje de programación

Es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones.

Un lenguaje de programación permite al programador especificar de manera precisa sobre qué datos debe operar una computadora, como estos datos deben ser almacenados o transmitidos y que acciones debe tomar bajo una variada gama de circunstancias. (Wikipedia., 2013)

6.2.1. Programación orientada a objeto (POO)

La programación orientada a objetos o POO (OOP según sus siglas en inglés) es un paradigma de programación que usa los objetos en sus interacciones, para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento. (Wikipedia., 2013)

La programación orientada a objeto introduce nuevos conceptos, entre ellos tenemos los siguientes:

- Clase: Definiciones de las propiedades y comportamiento de un tipo de objeto concreto. La instanciación es la lectura de estas definiciones y la creación de un objeto a partir de ellas.
- Herencia: Las clases no están aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen. La herencia organiza y facilita el polimorfismo y el encapsulamiento, permitiendo a los objetos ser definidos y creados como tipos especializados de objetos preexistentes. Estos pueden compartir (y extender) su comportamiento sin tener que volver a implementarlo. Esto suele hacerse habitualmente agrupando los objetos en clases y estas en árboles o enrejados que reflejan un comportamiento común. Cuando un objeto hereda de más de una clase se dice que hay herencia múltiple.
(Por ejemplo, herencia de la clase C a la clase D) es la facilidad mediante la cual la clase D hereda en ella cada uno de los atributos y operaciones de C, como si esos atributos y operaciones hubiesen sido definidos por la misma D. Por lo tanto, puede usar los mismos métodos y variables públicas declaradas en C. Los componentes registrados como "privados" (private) también se heredan, pero como no pertenecen a la clase, se mantienen escondidos al programador y sólo pueden ser accedidos a través de otros métodos públicos. Esto es así para mantener hegemónico el ideal de POO.
- Objeto: Instancia de una clase. Entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos), los mismos que consecuentemente reaccionan a eventos.
- Método: Algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un "mensaje". Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio en las propiedades del objeto, o la generación de un "evento" con un nuevo mensaje para otro objeto del sistema.
- Evento: Es un suceso en el sistema (tal como una interacción del usuario con la máquina, o un mensaje enviado por un objeto). El sistema maneja el evento enviando el mensaje adecuado al objeto pertinente. También se puede definir como

evento la reacción que puede desencadenar un objeto; es decir, la acción que genera.

- Atributos: Características que tienen las clases.
- Mensaje: Una comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos con ciertos parámetros asociados al evento que lo generó.
- Propiedad o atributo: Contenedor de un tipo de datos asociados a un objeto (o a una clase de objetos), que hace los datos visibles desde fuera del objeto y esto se define como sus características predeterminadas, y cuyo valor puede ser alterado por la ejecución de algún método.
- Estado interno: Es una variable que se declara privada, que puede ser únicamente accedida y alterada por un método del objeto, y que se utiliza para indicar distintas situaciones posibles para el objeto (o clase de objetos). No es visible al programador que maneja una instancia de la clase.
- Componentes de un objeto: Atributos, identidad, relaciones y métodos.
- Identificación de un objeto: Un objeto se representa por medio de una tabla o entidad que esté compuesta por sus atributos y funciones correspondientes. (Libre, 2013)

6.2.2. Software para desarrollar

- Java <http://www.java.com/es/download/>
- Android SDK <http://developer.android.com/sdk/index.html>
- Eclipse IDE <http://www.eclipse.org/> versión clásica
- Unity 3D <http://spanish.unity3d.com/>
- Android ISO <http://www.android-x86.org/download>
 - <http://code.google.com/p/android-x86/> (López, 2012)

6.2.2.1. Las herramientas de desarrollo y modelado necesarias en el proyecto

6.2.2.1.1. UNITY 3D 2014

Como se comentó anteriormente los videojuegos en su mayoría son desarrollados en los llamados “motores gráficos” y aprovechando que cada vez más se liberan motores gratuitos, es necesario aprender a usar una de las alternativas que nos ofrece el mercado ya que nos facilita la tarea de generar un prototipo inmersivo y dispondremos de muchas herramientas. En este sentido se eligió Unity3D, principalmente por su versatilidad al contar con una comunidad detrás que siempre aporta con cosas nuevas para el programa, su capacidad para exportar a múltiples plataformas y por ser gratuita (con sus versiones comerciales correspondientes). Además, cuenta con la ventaja de ser una plataforma más fácil de usar respecto a las otras 3D, ya que está pensada principalmente para el desarrollo de juegos casuales en vez de juegos AAA.

Esta herramienta se complementa con aquellas más clásicas de diseño como Photoshop, Illustrator, Cinema4D, Malla, entre otras.



Figura. 1 Unity 3D

6.2.2.1.2. 3Ds MAX 2014

Es un software de modelado y animación 3D que proporciona una solución completa de modelado, animación, renderización y composición en 3D a los creadores de juegos, cine y gráficos de movimiento. 3ds Max 2014 aporta nuevas herramientas para generar

multitudes, animar partículas e igualar perspectivas, además de compatibilidad con los sombreadores Microsoft® DirectX 11®. (Autodesk, 2013)

Características de 3ds Max

- Mapas vectoriales
- Animación con generación de multitudes
- Sistema de flujo de partículas ampliado
- Más rendimiento en la ventana gráfica
- Renderización en la ventana gráfica con DX 11
- Igualación de perspectivas
- Zoom y encuadre 2D

Ventajas

- Se puede crear visualizaciones y modelos 3D de alta calidad para poder repetir rápidamente diseños arquitectónicos, modelos de ingeniería civil y diseños de productos.
- Permite crear multitudes en movimiento o estáticas con la función de relleno de multitudes, que ofrece un alto nivel de control de un movimiento humano casi verosímil.
- Permite cargar gráficos vectoriales como mapas de texturas y renderizarlos con resoluciones dinámicas. La compatibilidad con mapas vectoriales le permite ampliar la imagen sin mermar su nitidez.
- Es posible componer de forma más rápida y eficaz con la herramienta de ajuste de perspectivas, que ajusta con gran parecido la vista de cámara de una escena a la perspectiva de un fondo fotográfico.
- Se puede crear y editar activos de mayor calidad con una API que muestra capacidades DirectX 11 en 3ds Max Design con soporte para lenguaje de sombreado de alto nivel (HLSL, High Level Shading Language).
- Permite disfrutar de un mayor control con funciones que le permiten aplicar zoom y encuadrar la ventana gráfica como si fuera una imagen en 2D, sin afectar a la posición real de la cámara. (Autodesk, 2013)

Requerimientos de sistema

- Sistema operativo Windows® 8 o Windows® 7 de 64 bits
- Procesador multinúcleo Intel o AMD de 64 bits
- 4 GB de RAM (se recomienda 8 GB), pero funciona con 2GB, sin afectar el rendimiento del equipo.
- 4,5 GB de espacio libre en el disco para la instalación
- Monitor VGA de 1024 x 768 con 4 MB de VRAM (se recomienda 32 MB)
- Ratón de 3 botones
- Explorador web Microsoft® Internet Explorer®, Apple® Safari® o Mozilla® Firefox®
- Consulte el asistente de Hardware recomendado para obtener una lista completa de las tarjetas gráficas y los sistemas recomendados.

6.2.2.1.3. ADOBE PHOTOSHOP CS6

Requisitos del sistema

Windows

- Procesador Intel® Pentium® 4 o AMD Athlon® de 64 bits. (de 2 GHz o superior).
- Microsoft® Windows® 7 con Service Pack 1, Windows 8 o Windows 8.1
- 1 GB de RAM.
- 2,5 GB de espacio disponible en el disco duro para la instalación; se requiere espacio libre adicional durante la instalación (no se puede instalar en dispositivos de almacenamiento flash extraíbles).
- Resolución de 1024 x 768 (se recomienda 1280 x 800) con tarjeta gráfica OpenGL® 2.0, color de 16 bits y 512 MB de VRAM (se recomienda 1 GB)*
- Se requiere disponer de una conexión a Internet y registrarse para la activación del software y la validación de los abonos obligatorias, y para el acceso a los servicios online†.

Mac OS

- Procesador Intel multinúcleo con compatibilidad de 64 bits.
- Mac OS X v10.7, v10.8, o v10.9

- 1 GB de RAM.
- 3,2 GB de espacio disponible en el disco duro para la instalación; se requiere espacio libre adicional durante la instalación (no se puede instalar en un volumen que utilice un sistema de distinción entre mayúsculas y minúsculas en archivos, ni en dispositivos de almacenamiento flash extraíbles).
- Resolución de 1024 x 768 (se recomienda 1280 x 800) con tarjeta gráfica OpenGL 2.0, color de 16 bits y 512 MB de VRAM (se recomienda 1 GB)*.
- Se requiere disponer de una conexión a Internet y registrarse para la activación del software y la validación de los abonos obligatorias, y para el acceso a los servicios online.

Funciones

- Enfoque inteligente completamente nuevo
- Funciones de Photoshop Extended incluidas
- Remuestreo inteligente
- Reducción del temblor de la cámara
- Adobe Camera Raw como filtro
- Adobe Camera Raw 8
- Rectángulos redondeados y otras formas editables
- Selección de múltiples formas y rutas
- Pintura en 3D mejorada
- Generación de activos de imagen en tiempo real y conectividad a Adobe Edge Reflow CC con tecnología Adobe Generator
- Compatibilidad con Objeto inteligente para Galería de desenfoques y Licuar
- Motor Mercury Graphics Engine
- Parche según el contenido y Movimiento con detección de contenido
- Controles 3D sobre el lienzo al alcance de su mano (ADOBE, 2013)

6.2.2.1.4. MOVIE MAKER

Windows Movie Maker (formalmente conocido como Windows Live Movie Maker, y Sundance como nombre clave para Windows 7) es un software de edición de vídeo que es parte de la suite de software Windows Essentials.

Características Del Programa

EL Windows Movie Maker se caracteriza por ser un programa de edición de vídeos creado por Microsoft y tiene diferentes funciones como:

- Transiciones de vídeo
- Efectos de vídeo
- Importación de imágenes, vídeo y audio
- Creación de títulos o créditos
- Una herramienta de automovie que sirve para organizar imágenes en la línea de tiempo.

Versiones:

- Versión 1.1: Se incluyo en el sistemas operativo Windows XP
- Versión 2.0: Esta fue una actualización gratuita de la versión anterior con algunas nuevas funciones y extensiones.
- Versión 2.1: Esta fue una actualización de las versiones anteriores que se incluyo en las Versiones del sistemas operativo Windows XP.
- Versión 2.5 y 2.6: esta fue una actualización de la versión 2.1, fue un gran éxito ya que poseía mas transiciones de vídeo y efectos ademas se lanzo en el año 2010.

La versión más actual de Windows Movie Maker es Windows Live Movie Maker, esta versión es totalmente diferente que el Windows Movie Maker 2.6 ya que es-

ta versión tiene una línea de tiempo diferente y con mejoras en efectos de transición y vídeo esta versión fue lanzada en el año 2008. (MICROSOFT, 2010)

6.3. Videojuego

Es un dispositivo electrónico que, a través de ciertos mandos o controles, permite simular juegos en la pantalla de un televisor, una computadora, consolas, celulares u otro dispositivo electrónico.

Gracias al videojuego, el usuario puede interactuar con el aparato en cuestión.

Los videojuegos se han convertido en una industria internacional surgiendo una gran variedad de “tipos” de juegos que albergan una gran variedad de mecánicas con las que se construyen las reglas para cada uno. Con el tiempo los distintos medios especializados junto a múltiples investigaciones relacionadas con la estructura y desarrollo de videojuegos han construido categorías para las mecánicas de juego, obviamente no son absolutas ya que hay una constante búsqueda de métodos más óptimos e innovadores y esto trae como resultados la mezcla de más de alguna de las categorías establecidas. (Wikipedia, 2008)

6.4. Programación de videojuegos

Es la actividad por la cual se diseña y crea un videojuego, desde el concepto inicial hasta el videojuego en su versión final. Ésta es una actividad multidisciplinaria, que involucra profesionales de la informática, el diseño, el sonido, la actuación entre otros. (Wikipedia, 2010)

6.5. Géneros de juegos

- **Juegos de Acción:** Los juegos de acción son aquellos que requieren decisiones rápidas y que proponen desafíos simples, como llegar a una cierta meta, pero que en su transcurso debe enfrentar múltiples enemigos. Por lo general el jugador controla solo un personaje principal.

- Juegos de Aventura: Los juegos de aventura se diferencian por tener una estructura bastante rígida, con secuencias de movimiento y videos que van revelando parte de la historia en la que el jugador se involucra. Este tipo de juegos por lo general poseen un guión interesante, complejo, que va siendo descubierto a medida que se va jugando.
- Juegos de Rol (en inglés RPG): Los juegos de rol permiten al jugador asumir el rol del personaje principal, controlándolo directamente en el mundo del juego. Uno de los elementos claves de los juegos de rol es la idea de progreso del personaje, donde a través del curso del juego las acciones del jugador son premiadas con nuevas características permitiendo tomar ventaja y volverse mejor para ciertas actividades. Los desafíos que plantean los RPG varían según la historia, la configuración del mundo y el tipo de personaje que elija, por lo que suelen estar acompañados con un profundo trasfondo en la historia del mundo como en el de cada personaje.
- Juegos de estrategia: Este tipo de juegos ponen al jugador desde una perspectiva mayor al de un personaje, más bien como si se tratase de un ser omnisciente. Sus mecánicas de juego se caracterizan porque requieren de un cuidadoso y hábil pensamiento a la hora de planear y distribuir recursos para lograr el objetivo. En prácticamente todos los casos, tienes que escalar fuerzas y crear unidades según los recursos que vayas adquiriendo por unidades especializadas. El componente clave en todos estos juegos es la distribución estratégica de los recursos y el uso de unidades con variadas habilidades bajo el control del jugador.
- Juegos de puzzles: Centrado en la solución de problemas, estos juegos requieren que el jugador resuelva puzzles de manera lógica o recorrer lugares complejos a modo de laberintos. Las mecánicas de juego suelen ser muy básicas y muchas veces este tipo de juegos suelen ser una parte de un juego mayor, comúnmente de aventuras.
- Hay que agregar también que con el surgimiento de los juegos online y de las plataformas móviles se han agregado un par de términos muy comunes pero que en su

naturaleza corresponden más bien a una subcategoría dentro de estas grandes ya mencionadas, como es el caso para los juegos casuales (que en su mayoría se encuentran dentro de los juegos de puzzles o aventura) o los MMORPG (que son un tipo de juego de rol pero online, lo que lo hace masivo con mucha gente jugando al mismo tiempo).

- Juegos de Simulación: Juegos basados en la reproducción, generalmente de forma realista, del funcionamiento de alguna actividad. Ejemplos: “Microsoft Flight Simulator” (simulador de vuelo), “Los Sims” (simulador social) o Nintendogs (simulador de mascotas).
- Juegos Educativos y Entrenamiento Mental: Juegos cuyo objetivo es transmitir al jugador algún tipo de conocimiento o mejorar habilidades mentales. Su mecánica puede abarcar cualquiera de los otros géneros. Habitualmente están dirigidos a niños de menos de 10 años aunque se ha puesto de moda el desarrollo de este tipo de videojuegos para cualquier edad. Referentes en este campo son la familia de juegos de “Aprende con Pipo” o la famosa saga de Nintendo “Brain Training”.
- Juegos de Carreras: Estos juegos se caracterizan por desarrollar carreras de vehículos (motos, coches, etc.). El reto es que el jugador demuestre su pericia al volante de su vehículo y no cómo de realista es ésta, si no podrían considerarse de simulación. En esta familia destacan: “Formula 1”, “Gran Turismo” o “Need For Speed”.
- Juegos de Lucha: Estos juegos retan al jugador a ir ganando combates de lucha para conseguir un campeonato o un premio. La mayor dificultad que presentan este tipo de juegos es la combinación de teclas para realizar los movimientos especiales característicos de cada luchador. Clásicos de este género son: “Street Fighter” o “Mortal Kombat”.
- Juegos Musicales: Su desarrollo gira en torno a la música, ya sea de tipo "karaoke" o en la que se debe bailar o tocar algún instrumento. Ejemplos: “Sing Star”, “Dance Dance Revolution!”, o “Guitar Hero”.
- Juegos de Deportes: Se consideran juegos deportivos aquellos que simulan los deportes tradicionales como el fútbol, baloncesto, golf, etc. Podemos destacar “Pro Evolution Soccer” o “Wii Sport”.

- Juegos de Plataformas: Juegos en los que el protagonista ha de avanzar a través de un mapa con distintas plataformas u objetos situados a diferentes alturas para llegar al objetivo. Claros exponentes son: “Sonic the Hedgehog” y “Super Mario Bros”.
- Juegos de Disparo: También conocidos como "Shooters", “FPS” o "Shoot'em up". En estos juegos el protagonista ha de abrirse camino a base de disparos. Clásicos de este género son: “Doom” y “Quake”.
- Juegos MMOG: Se les llama también juegos masivos multijugador online. Se caracterizan porque los jugadores pueden interactuar con otros de forma competitiva o colaborativa. Los jugadores pueden agruparse en grupos llamados “clanes” con otros jugadores que posean rasgos en común, y complementarse entre ellos gracias a sus propias habilidades para poder “vencer” a otros clanes en determinados retos. Encabeza este grupo el juego “World of WarCraft”.
- Juegos Mixtos: Se caracterizan por poseer aspectos propios de cada uno de los apartados anteriores y ser un reto para los diseñadores, pues obligan a dominar diferentes conceptos de juego. Entre los grandes títulos de este tipo de juegos podemos encontrar los que mezclan la aventura, acción, plataformas y conceptos RPG como “The Legend of Zelda”, “Castlevania”, “Uncharted”, “Metal Gear”, o la saga “GTA”.

6.6. Clasificación de Juegos según el número de jugadores

Lo primero que tenemos que pensar es si nuestro juego va a ser jugado por un jugador o por varios. De acuerdo con Zagal (Zagal, Rick, & Hsi, 2006) y (Pepe, 2004) cuando jugamos en grupo podemos hacerlo competitivamente, colaborativamente y cooperativamente. Esta clasificación se basa en la forma en la que se asocian los jugadores para realizar sus objetivos o cómo consiguen sus metas dentro del juego, y no en cómo realizan las tareas para lograrlo. Creemos que es mejor este criterio, porque una de las características que debe cumplir un juego está basada en la libertad del jugador para realizar las acciones dentro del juego, limitadas solamente por los objetivos a cumplir. La clasificación de juegos según el número de jugadores puede ser:

- Videojuegos Individuales: Un juego individual es el que recoge un solo jugador. Este se enfrenta a la Inteligencia Artificial del juego para conseguir una serie de objetivos dentro de la trama.
- Videojuegos Competitivos: Un juego competitivo es aquel juego donde un jugador juega para conseguir su propio éxito personal o meta. Este objetivo final puede ser totalmente opuesto al de otros jugadores, por lo que generalmente al ganar un jugador, pierden el resto de jugadores.
- Videojuegos Colaborativos: En el otro lado de la balanza tenemos los juegos colaborativos donde el éxito individual se transforma en que el grupo consiga la meta deseada, es aquí donde aparece el concepto “equipo”. Resumiendo, un individuo gana cuando gana el equipo y con ello todos sus miembros. La meta es común, global y única para todos los individuos del juego.
- Videojuegos Cooperativos: Basándose en las ideas de los puntos anteriores aparecen los juegos cooperativos, donde cada uno de los individuos puede tener sus propios objetivos individuales, aunque se agrupan en equipos para buscar un beneficio conjunto para la consecución de sus propias metas, que pueden obtener gracias a los demás. Es decir, el equipo no es primordial para conseguir la meta del jugador, sino algo circunstancial. Se pueden tener metas a nivel de equipo, pero deberán ser algo secundario. (Wikipedia, 2009)

6.7. Plataformas para los videojuegos:

Como ya se mencionó, el desarrollo de la tecnología dio paso al nacimiento de los videojuegos. Esta estrecha relación permite una evolución tanto en hardware como en software de los videojuegos y da cuenta del soporte que permite su reproducción, elemento que hay que tener en cuenta a la hora de diseñar ya que cada uno es distinto y posee sus ventajas y limitante. Estos soportes han ido evolucionando y en este momento nos encontramos con una revolución: El mercado de los dispositivos móviles.

Hasta la fecha el soporte principal de los videojuegos han sido las “consolas” de juegos, hechas específicamente para hacer correr distintos títulos y se han desarrollado a lo largo del tiempo con nuevas versiones mejoradas, a cada cambio se le llama “nueva generación”. Actualmente las consolas más populares son Xbox-360 de la empresa Mi-

crosoft, PlayStation 3 de Sony y Nintendo Wii de Nintendo Entertainment que corresponden a la séptima generación. Hay que tener pendiente también que estamos a punto de pasar a la próxima, Nintendo WiiU es una de las consolas que serán parte de esta nueva generación. Hay que tener en consideración también el mercado de consolas portátiles, las principales son PSVita y Nintendo 3Ds a las que ya se les atribuye pertenecer a la octava generación de consolas. Sin embargo las consolas no han sido el único soporte, también se ha ganado su lugar la computadora personal, que si bien no están hechas específicamente para este propósito son un medio más para reproducir videojuegos que muchas empresas tienen en cuenta. Actualmente Microsoft Windows, Mac de Apple y Linux cuentan con juegos que pueden ser jugados desde sus sistemas operativos. Por último mencionar lo que para varios expertos es una revolución: Los dispositivos móviles. Su desarrollo ha sido tan rápido que han abierto una ventana a nuevos desarrolladores ya que sus costos son mucho menores y en internet se han ofrecido varias herramientas gratuitas que permiten programar videojuegos para estas plataformas. Este año destacan los teléfonos y tablets con Android y su plataforma de distribución Google Play, Iphone e Ipad con IOS que dispone de la App Store y Windows Phone con su Marketplace.

6.7.1. Historia de la telefónica móvil.

En muy pocos años los teléfonos móviles han llegado a convertirse en un dispositivo esencial en nuestras vidas, actualmente hay miles de modelos y los diferentes fabricantes siguen innovando intentando destacar sobre los de la competencia.



La tecnología inalámbrica tuvo gran aceptación, por lo que a los pocos años de implantarse se empezó a saturar el servicio, por lo que se presentó la imperiosa necesidad de desarrollar e implementar otras formas de acceso múltiple al canal y transformar los sistemas analógicos a digitales para darle cabida a más usuarios.

Para separar una etapa de la otra, a la telefonía celular se ha categorizado por generaciones.

Figura. 2 Motorola DynaTAC 8000X.

6.7.1.1. Tendencias móviles

En la actualidad el desarrollo de aplicaciones móviles está enfocado a las diferentes necesidades de los humanos, para que sean de manera eficiente y eficaz la realización de las actividades que día con día se llevan a cabo. Existe una gama de tendencias móviles que van desde la domótica, que más que nada es controlar la casa u habitación mediante un dispositivo móvil, también existen tendencias que están enfocadas al comercio, a la industria, bancarias, empresariales, de información, juegos, medicina, educación, entre otras, que como se menciona anteriormente hace que los usuarios puedan llevar a cabo desde el dispositivo móvil las diferentes tareas que tiene día con día, además de poder realizarlas desde cualquier lugar y en cualquier instante.

6.7.1.2. Sistemas Operativos móviles

La descripción de los sistemas operativos es breve ya que el objetivo de este trabajo es enfocado al sistema operativo Android. Sólo se dar un panorama general de los sistemas operativos móviles existentes actualmente.

6.7.1.2.1. Palm OS

Palm OS es un sistema operativo propietario destinado a dispositivos móviles, más específicamente a PDA (Personal Digital Assistant). Palm OS comenzó su desarrollo en 1996 y Palm Inc. comenzó a licenciarlo en diciembre de 1997 con sus novedosos aparatos PalmPilot. A partir de ese momento el soporte y el desarrollo de Palm OS se disparó, llegando en enero del 2001 a tener 100.000 personas registradas en su red de desarrolladores trabajando en proyectos para Palm OS.

Palm OS fue uno de los pioneros en el mercado de los dispositivos móviles y por varios años se mantuvo como uno de los mejores sistemas operativos, sobre todas las cosas por ser muy usable y simple.

Las primeras versiones de este sistema operativo estuvieron basadas en un SO multitareas creado por Motorola. Las principales características de la plataforma Palm eran:

- Hardware altamente integrado con el SO, basado en un procesador de 68k.
- Usaba un display monocromático; preferible antes que implementar los colores de manera pobre.
- Pocas funciones del SO, se centraba sobretodo en la usabilidad.

- Estaba diseñado para ser una herramienta práctica, no un sistema orientado a personas con conocimiento informático.

6.7.1.2.2. Symbian

Es el resultado de una alianza entre varias empresas multinacionales de renombre en el mercado tales como Nokia, Sony Ericsson, Samsung, Siemens, Motorola y otras.

Sus orígenes provienen del EPOC32, otro sistema operativo para dispositivos móviles, el cual pertenece a una familia de sistemas operativos que tiene sus orígenes a finales de 1980 y principios de 1990 con el EPOC16.

Luego de unos años, más precisamente en 1997, apareció la primera versión del denominado EPOC32, que luego pasaría a llamarse Symbian OS.

El objetivo de Symbian fue crear un sistema operativo para terminales móviles que pudiera competir con el de Palm o el Windows Phone de Microsoft y ahora Android de Google Inc. Y iOS de Apple Inc.

6.7.1.2.3. Bada

Bada, océano o mar en coreano, es un sistema operativo para teléfonos móviles desarrollado por Samsung Electronics. Está diseñado para cubrir tanto los teléfonos smartphones de gama alta como los de gama baja. Samsung afirma que Bada rápidamente reemplazará la plataforma de teléfonos más básicos, convirtiendo futuros teléfonos en smartphones. Se basa en el sistema operativo propiedad de Samsung SHP OS, utilizado en muchos de sus teléfonos como el Samsung Wave.

6.7.1.2.4. MeeGo

Fue la unión de los sistemas operativos Maemo y Moblin, con el que Intel y Nokia pretenden competir con el sistema Android de Google. El proyecto del nuevo sistema a diferencia de Android está supervisado por la Linux Foundation. MeeGo se presentó como un sistema preparado para funcionar en netbooks, dispositivos portátiles, sistemas en vehículos, televisiones y teléfonos multimedia. Básicamente se trata de una distribución Linux con soporte para ARM e Intel/Atom que usa Qt para su interfaz. El software MeeGo se pueden clasificar en dos categorías: el sistema operativo (SO) y la experiencia del usuario (UX). El software del sistema operativo tiene, en una gran mayoría, li-

encia copyleft para garantizar la apertura del sistema subyacente, mientras que en el software UX es predominantemente el uso licencias BSD.

6.7.1.2.5. BlackBerry

El BlackBerry OS es un sistema operativo móvil desarrollado por Research In Motion para sus dispositivos BlackBerry. El sistema permite multitarea y tiene soporte para diferentes métodos de entrada adoptados por RIM para su uso en computadoras de mano, particularmente la trackwheel, trackball, touchpad y pantallas táctiles.

Su desarrollo se remonta la aparición de los primeros handheld en 1999. Estos dispositivos permiten el acceso a correo electrónico, navegación web y sincronización con programas como Microsoft Exchange o Lotus Notes aparte de poder hacer las funciones usuales de un teléfono móvil.

6.7.1.2.6. Windows Phone

Windows Phone, anteriormente llamado Windows Mobile es un sistema operativo móvil compacto desarrollado por Microsoft, y diseñado para su uso en teléfonos inteligentes (Smartphones) y otros dispositivos móviles. Windows Phone hace parte de los sistemas operativos con interfaz natural de usuario.

Se basa en el núcleo del sistema operativo Windows CE y cuenta con un conjunto de aplicaciones básicas utilizando las API de Microsoft Windows. Está diseñado para ser similar a las versiones de escritorio de Windows estéticamente.

6.7.1.2.7. iOS

Como todos sabemos iOS es conocido actualmente como el sistema operativo diseñado por Apple en California destinado para sus dispositivos de última generación (iPod, iPhone e iPad).

El desarrollo originalmente se basa del núcleo sistema operativo Mac OS X, (Darwin-BSD), pero a diferencia del sistema original, el ambiente gráfico del mismo es rediseñado para adaptarse a la plataforma móvil, permitiendo a su vez una mayor comodidad de uso y un sistema mucho más estable, a este nuevo entorno gráfico se le conoce como “SpringBoard” y este constantemente ha sido mejorado desde sus primeras versiones, las cuales aparecen con la integración del iPhone. En esta época el sistema únicamente para el diseño del iPhone toma el nombre de iPhone OS, con el avance de las versio-

nes del mismo y con la aparición de diferentes dispositivos, el sistema se renombra con el Acrónimo iOS.

Como parte interesante podemos destacar que en cada versión del sistema iOS ha enfocado gran parte de la programación en su ambiente gráfica, y gracias a esto para beneficio de los usuarios (y un problema para los programadores), es que cada versión nueva liberada por Apple requiere que el software diseñado este mayormente enfocado en explotar las nuevas características del sistema operativo y las alternativas que éste ofrece en sus distintas mejoras.

6.7.1.2.8. Android

Android es un sistema operativo para dispositivos móviles como teléfonos inteligentes y tabletas. Es desarrollado por la Open Handset Alliance, la cual es liderada por Google. Fue desarrollado inicialmente por Android Inc., una firma comprada por Google en 2005. Es el principal producto de la Open Handset Alliance, un conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio.

Ahora bien en este sistema operativo es el que se basa este trabajo, ya que es uno de los mayores SO comercializados y es el de mayor interés para el desarrollo de aplicaciones móviles con software abierto.

6.7.2. Dispositivos Móviles



Figura. 3 Dispositivos Móviles con sistema Operativo Android (web, 2011)

Existe una gran cantidad de dispositivos con diferentes resoluciones, hay que tener esto en cuenta al desarrollar un juego o aplicación.

6.7.3. Android

6.7.3.1. El origen de Android

En octubre del año 2003 Andy Rubín, Rich Miner, Nick Sears y Chris White empezaron a formar a Android Inc., una compañía con ubicación en Palo Alto (California) que comenzó a “fabricar” las piezas con las que se iría montando, poco a poco, este gigantesco robot verde que hoy disfrutamos. En sus inicios, únicamente trascendió que la actividad de la empresa se centraba en “el desarrollo de software para teléfonos móviles”.

Android Inc. comenzó a trabajar sin darse a conocer a fondo, hasta que Google vio que podría ser una compañía con un futuro prometedor del sector móvil, con la clara intención de replicar su éxito de la Web en el futuro de las telecomunicaciones inalámbricas. La fecha clave para llegar a entender mejor el éxito de Android es el 5 de noviembre de 2007. Ese día se fundaba la OHA (Open Handset Alliance), una alianza comercial de 35 componentes iniciales liderada por Google, que contaba con fabricantes de terminales móviles, operadores de telecomunicaciones, fabricantes de chips y desarrolladores de software. El mismo día se dio a conocer por vez primera lo que hoy conocemos como Android, una plataforma de código abierto para móviles y actualmente también para Tablets, que se presentaba con la garantía de estar basada en el sistema operativo Linux. Pocos días después, en concreto el 12 del mismo mes de noviembre, se “liberaba” el primer “kit” de desarrollo de aplicaciones para que los programadores comenzasen a desarrollar aplicaciones móviles y así nacían las Apps del sistema.

Cuando escuchamos que Android es un sistema operativo de código abierto quiere decir que cualquier desarrollador puede crear aplicaciones en lenguaje C u otros lenguajes y compilarlas a código nativo de ARM (API de Android), el sistema operativo está compuesto por 12 millones de líneas de código, incluyendo 3 millones de líneas de XML, 2,8 millones de líneas de lenguaje C, 2,1 millones de líneas de Java y 1,75 millones de líneas de C++.

6.7.3.2. Las actualizaciones

Uno de los elementos diferenciales de Android respecto al resto de sistemas operativos móviles es la cantidad de actualizaciones que recibe. Los usuarios, al final acaban valorando más a aquel que se preocupa por mejorar las prestaciones de su dispositivo sin necesidad de cambiarlo continuamente y es innegable que en ese terreno Android es el número uno indiscutible. (web, 2011)

6.7.3.3. Versiones de Android

El 23 de septiembre del 2008 salió la Primera versión oficial de S.O. Android, que cuenta con la integración de los servicios de Google, navegador Web con capacidad para hacer Zoom y cargar múltiples páginas, descarga de aplicaciones desde Android Market, multitarea, Wifi y Bluetooth.

El 30 de Abril del 2009, salió la versión **Cupcake (1.5)** Basada en el kernel de Linux 2.6.27 añade la compatibilidad con grabación de video, más rápido de cámara puesta en marcha y la captura de imágenes, transiciones de pantallas animadas, nuevos Widgets y carpetas para colocar en la pantalla de inicio, GPS del dispositivo más rápidos, teclado virtual y subida directa de vídeos a YouTube. El 15 de septiembre del 2009, la versión **Donut (1.6)** basado en el kernel de Linux 2.6.29 mejora las aplicaciones de búsqueda y por voz, indicador gráfico de batería y la integración de las funciones de la cámara con la galería de imágenes, alterna entre los módulos de captura de video.

El 26 de octubre del 2009, Con la versión **Eclair (2.0/2.1)** llegaron mejoras sustanciales, entre las que destacan la gestión de múltiples cuentas de correo y sincronización de contactos simultáneamente además de soporte para Microsoft Exchange y la compatibilidad con HTML5. (web, 2011)

El 20 de mayo del 2010, la actualización a **Froyo (2.2)** quedará como la primera en incorporar la función de compartir internet (convirtiendo nuestro móvil en un Hotspot wifi), aunque la implementación del Flash de Adobe en su versión 10.1 supuso un gran avance, sobre todo si lo observamos desde la perspectiva que nos da el tiempo.

El 06 de diciembre del 2010, finalmente, hasta la llegada del reciente Ice Cream Sandwich, **Gingerbread (2.3)** nos trajo mejoras generales en velocidad y optimización global

del sistema, además de la compatibilidad para el estándar del futuro en medios de pago, el NFC (Near Field Communication).

El 10 de mayo del 2011, **Honeycomb (3.0/3.1/3.2)** Optimizado especialmente para Tablets y equipos con pantallas grandes, escritorio 3D con Widgets rediseñados, soporte para video-chat mediante Google Talk, mejor soporte para redes wifi, mejorado el trabajo multitarea, mejoradas las notificaciones y las personalizaciones de la pantalla principal, mejoras en el navegador web, añade soporte para una gran variedad de periféricos y accesorios USB. (web, 2011)

El 19 de Octubre del 2011, **Ice Cream Sandwich (4.0)** Versión que unifica el uso en cualquier dispositivo, tanto en teléfonos, tabletas, televisores, netbooks, etc. Interfaz limpia y moderna llamada “Holo” con una nueva fuente llamada “Roboto”, muy al estilo de Honeycomb. Opción de utilizar los botones virtuales en la interfaz de usuario, en lugar de los botones táctiles capacitivos. Llega la aceleración por hardware, lo que significa que la interfaz podrá ser manejada y dibujada por la GPU y aumentando notablemente su rapidez, su respuesta y evidentemente, la experiencia de usuario. Multitarea mejorada, estilo Honeycomb. Añadiendo la posibilidad de finalizar una tarea simplemente desplazándola fuera de la lista. Añadido un gestor del tráfico de datos de internet. El entorno le permite establecer alertas cuando llegue a una cierta cantidad de uso y desactivación de los datos cuando se pasa de su límite. (web, 2011)

En el 2012 **Jelly Bean (4.1, 4.2)** es la versión más rápida y suave de Android. Jelly Bean mejora en la simplicidad y la belleza de Android 4.0, y presenta una nueva experiencia de búsqueda de Google en Android.

Todo en Jelly Bean se siente rápido, fluido y sin problemas. Desplazamiento entre pantallas de inicio y cambiar entre aplicaciones sin esfuerzo, como si pasara las páginas de un libro. Jelly Bean cuenta con un mejor rendimiento en todo el sistema, incluyendo los cambios de orientación más rápidos, respuestas más rápidas al cambiar entre aplicaciones recientes y la representación más suave y más consistente a través de todo el sistema de amortiguación vsync y triples.

Jelly Bean tiene respuestas táctiles más reactivas y uniforme, y hace que su dispositivo aún más sensible al aumentar la CPU de su dispositivo instantáneamente al tocar la

pantalla, y se vuelve hacia abajo cuando no lo necesita para mejorar la vida de la batería. (Página oficial de Android 2012).



Figura. 4 Versiones de Android. (web, 2011)

6.7.3.4. Ventajas

Se explican algunas de las ventajas que tiene Android:

- Código abierto
- Libertad
- Sin fronteras
- Comunidad
- Costos y gustos
- Batería
- Personalizable
- Multitarea
- No solo móviles (Android, 2013)

6.7.3.5. Características

➤ Diseño de dispositivo

La plataforma es adaptable a pantallas más grandes, VGA, biblioteca de gráficos 2D, biblioteca de gráficos 3D basada en las especificaciones de la OpenGL ES 2.0 y diseño de teléfonos tradicionales. Con los nuevos modelos es más cómoda la visualización, ya que la pantalla es más grande que las primeras versiones de

teléfonos celulares. Existen una gran variedad de dispositivos al gusto del usuario, y con las funcionalidades requeridas o necesarias para cada uno de ellos.

➤ **Almacenamiento**

En lo que concierne al almacenamiento de información en el dispositivo se puede utilizar SQLite, que es una base de dato muy liviana. Y que es una de las que utiliza el sistema operativo Android.

➤ **Conectividad Android**

Soporta las siguientes tecnologías de conectividad: GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wifi, LTE y WIMAX.

GSM/EDGE (Tasa de Datos Mejoradas para la evolución de GSM) Es una tecnología de la telefonía móvil celular, que actúa como puente entre las redes 2G y 3G.

IDEN: es una tecnología de telecomunicaciones móviles, desarrollado por Motorola, que proporciona a sus usuarios los beneficios de un radio troncal y un teléfono celular.

➤ **Mensajería**

Las formas de mensajería pueden ser SMS y MMS, incluyendo mensajería de texto y ahora la Android Cloud to Device Messaging Framework (C2DM) es parte del servicio de Push Messaging de Android.

➤ **Navegador web**

El navegador web incluido en Android está basado en el motor de renderizado de código abierto WebKit, emparejado con el motor JavaScript V8 de Google Chrome. El navegador obtiene una puntuación de 93/100 en el test Acid3.

➤ **Soporte de Java**

Aunque la mayoría de las aplicaciones están escritas en Java, no hay una máquina virtual Java en la plataforma. El bytecode Java no es ejecutado, sino que primero se compila en un ejecutable Dalvik y corre en la Máquina Virtual Dalvik. Dalvik es una máquina virtual especializada, diseñada específicamente para Android y optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesador limitados. El soporte para J2ME puede ser agregado mediante aplicaciones de terceros como el J2ME MIDP Runner.

➤ **Soporte multimedia**

Android soporta los siguientes formatos multimedia: WebM, H.263, H.264 (en 3GP o MP4), MPEG-4 SP, AMR, AMR-WB (en un contenedor 3GP), AAC, HE-AAC (en contenedores MP4 o 3GP), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF y BMP.

➤ **Soporte para Streaming**

Streaming RTP/RTSP (3GPP PSS, ISMA), descarga progresiva de HTML (HTML5 <video> tag). Adobe Flash Streaming (RTMP) es soportado mediante el Adobe Flash Player. Se planea el soporte de Microsoft Smooth Streaming con el port de Silverlight a Android. Adobe Flash HTTP Dynamic Streaming estará disponible mediante una actualización de Adobe Flash Player.

➤ **Soporte para hardware adicional**

Android soporta cámaras de fotos, de vídeo, pantallas táctiles, GPS, acelerómetros, giroscopios, magnetómetros, sensores de proximidad y de presión, termómetro, aceleración 2D y 3D, etc.

➤ **Market (Play store)**

El Android Market es un catálogo de aplicaciones gratuitas o de pago en el que se pueden ser descargadas e instaladas en el dispositivos Android sin la necesidad de un PC.

➤ **Multi-táctil**

Android tiene soporte nativo para pantallas multi-táctil que inicialmente hizo su aparición en dispositivos como el HTC Hero. La funcionalidad fue originalmente desactivada a nivel de kernel (posiblemente para evitar infringir patentes de otras compañías). Más tarde, Google publicó una actualización para el Nexus One y el Motorola Droid que activa el soporte para pantallas multi-táctiles de forma nativa.

➤ **Bluetooth**

El soporte para A2DP y AVRCP fue agregado en la versión 1.5 el envío de archivos (OPP) y la exploración del directorio telefónico fueron agregados en la versión 2.0;49 y el marcado por voz junto con el envío de contactos entre teléfonos lo fueron en la versión 2.2.54

➤ **Video llamada**

Android soporta video llamada a través de Google Talk desde su versión Honey-Comb. Además de que existen una gran variedad de aplicaciones con lo que se pueden hacer este tipo de comunicación.

➤ **Multitarea**

Multitarea real de aplicaciones está disponible, es decir, las aplicaciones que no estén ejecutándose en primer plano reciben ciclos de reloj, a diferencia de otros sistemas de la competencia en la que la multitarea es congelada.

➤ **Características basadas en voz**

La búsqueda en Google a través de voz está disponible como "Entrada de Búsqueda" desde la versión inicial del sistema.

➤ **Tethering (Anclaje de red).**

Android soporta tethering, que permite al teléfono ser usado como un punto de acceso alámbrico o inalámbrico (todos los teléfonos desde la versión 2.2, no oficial en teléfonos con versión 1.6 o superiores mediante aplicaciones disponibles en el Android Market, por ejemplo PdaNet). Para permitir a un PC usar la conexión 3G del móvil Android se podría requerir la instalación de software adicional.

6.7.3.6. Arquitectura de Android.

De acuerdo a catalán (2011), menciona en su guía de Android como es que es la arquitectura de Android.

6.7.3.6.1. Kernel del Linux

En la base tenemos el kernel 2.6 de Linux, Android lo utiliza por su robustez demostrada y por la implementación de funciones básicas para cualquier sistema operativo, por ejemplo: seguridad, administración de memoria y procesos, implementación de conectividad de red (network stack) y varios intérpretes (drivers) para comunicación con los dispositivos físicos (hardware).

Android utiliza como base el kernel de Linux pero los dos sistemas no son lo mismo, Android no cuenta con un sistema nativo de ventanas de Linux ni tiene soporte para glibc (librería estándar de C) ni tampoco es posible utilizar la mayoría de aplicaciones de GNU de Linux.

Además de todo lo ya implementado en el kernel de Linux, Android agrega algunas cosas específicas para plataformas móviles como la comunicación entre procesos (lograda a través del binder), la forma de manejar la memoria compartida (ashmem) y la administración de energía (con wakelocks). De las características únicas del kernel utilizado por Android encuentran más información en Android Kernel Features.



Figura. 5 Kernel de Linux. (web, 2011)

6.7.3.6.2. Librerías y ejecución.

Sobre el kernel, tenemos un conjunto de librerías de C y C++ utilizadas por el sistema para varios fines como el manejo de la pantalla (surface manager), mapas de bits y tipos de letra (Free Type), gráficas en 2D y 3D (SGL y OpenGL), manejo de multimedia (Media Framework), almacenamiento de datos (SQLite) y un motor para las vistas web y el navegador (WebKit).

Junto a estas librerías, encontramos lo necesario para la ejecución de las aplicaciones a través de la máquina virtual Dalvik. Cada aplicación utiliza una instancia de la máquina virtual ejecutando un archivo DEX (Dalvik Executable) y el sistema está optimizado para que se ejecuten múltiples instancias de la máquina virtual. Se desarrolla en Java pero no se utiliza una máquina virtual de Sun para su ejecución ni tampoco archivos CLASS.

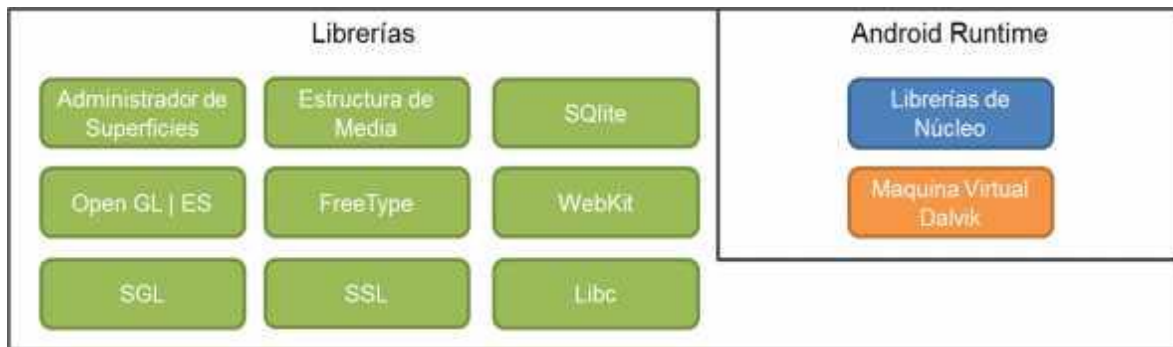


Figura. 6 Librerías y ejecución. (web, 2011)

6.7.3.6.3. Estructura de Aplicaciones

Sobre las librerías encontramos una estructura que nos brinda un contexto para desarrollar, este framework permite a los desarrolladores aprovechar un sistema de vistas ya construido, administrar notificaciones y acceso a datos a través de proveedores de contenido entre otras cosas.



Figura. 7 Estructura de Aplicaciones. (web, 2011)

6.7.3.6.4. Aplicaciones

Las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y otros. Todas las aplicaciones están escritas en lenguaje de programación Java.

Marco de trabajo de aplicaciones:

Los desarrolladores tienen acceso completo a los mismos APIs del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Este mismo mecanismo permite que los componentes sean reemplazados por el usuario.



Figura. 8 Aplicaciones. (web, 2011)

6.7.3.6.5. Aplicaciones Móviles

La tecnología móvil ha ido cambiando considerablemente en los últimos años llevando a cabo el desarrollo de aplicaciones móviles ayudando a los usuarios a realizar activida-

des diarias con los dispositivos móviles, ya, que la mayoría de las personas cuenta con un teléfono inteligente. EL benefician de los desarrolladores y programadores es de que se tiene una mayor oportunidad de trabajo para la elaboración de aplicaciones móviles. Ahora bien para el desarrollo de aplicaciones para teléfonos inteligentes, hay diferentes opciones de aplicaciones que se deben ser comprendidas para poder elaborar la aplicación como son:

- Aplicaciones web móviles
- Aplicaciones móviles nativas
- Aplicaciones híbridas

A continuación se da una explicación de cada una de ellas y algunas de sus características.

6.7.3.6.6. Aplicaciones Web Móviles.

El desarrollo móvil web es el desarrollo de aplicaciones web regulares, pero optimizadas para ser visualizadas desde un dispositivo móvil o Tablet. Por definición, estas aplicaciones serán accedidas utilizando el navegador que viene por defecto dentro de los dispositivos. Que esto viene siendo como consultar una página web, que dependiendo del tamaño se adecue la estructura de la página como es el caso de la tecnología responsive web design (Diseño web sensible o adaptativo).

6.7.3.6.7. Características de aplicaciones móviles web

1. Serán accedidas desde un navegador desde el teléfono.
2. Para todos los dispositivos se desplegará casi que de la misma forma.
3. Es prácticamente lo mismo que desarrollar cualquier aplicación o sitio web: ya que se utiliza HTML, CSS y JavaScript.

➤ Ventajas

1. Compatibilidad: Se reutiliza casi en un 100% el mismo código fuente para todos los sistemas operativos orientados a móviles, así que para hacer alguna modificación, solo se tendrá que modificar un código fuente.
2. Rango de usuarios: Se puede llegar a más tipos de sistemas operativos, ya que no se basa en uno en específico.

3. Fácil uso: No hace falta descargar nada, sólo con acceder a una URL los usuarios podrán utilizar la aplicación.

4. Actualizado siempre: Las actualizaciones son de manera inmediata sin necesidad de pedir permiso al usuario o el tener que esperar la autorización de los cambios que se realicen.

5. Más sencillo y rápido de llevar a cabo: muchas veces, es mucho más fácil desarrollar y diseñar dentro de un browser utilizando HTML, CSS y JavaScript que hacer lo mismo en Java, Objective-C o JavaScript (Appcelerator, Sencha). Posicionar elementos, estilizarlos y ciertas flexibilidades son posibles en minutos con tecnologías móviles.

➤ Desventajas

1. No se podrán utilizar los componentes del teléfono como la cámara, el acelerómetro, GPS etc. Ya que estas aplicaciones al ser accedidas desde el navegador, no tienes disponibilidad de ciertos privilegios o permisos para poder manipular estos componentes.

2. Debe adaptarse a las medidas de un dispositivo móvil, además, de que no existen medidas estándares para diferenciar los tamaños entre dispositivos y esto hace que la adaptación de la aplicación web móvil sea imprecisa.

3. Posibilidad de usarla offline (en algunos): Tienes esta aplicación en mente que necesitas que se puedan utilizar en cualquier momento. Olvídate del “offline mode” cuando estés desarrollando con web. Esto no aplica para los y otros, ya que Safari implementa Web Storage de HTML5 y Offline Application Cache.

Ejemplos de aplicaciones móviles web: Facebook Móvil, Linked In Touch, Panagamers — Visto desde un móvil, entre otros.

6.7.3.6.8. Aplicaciones Nativas.

Por otra parte, el **desarrollo móvil nativo** es el desarrollo de aplicaciones que serán instaladas en el sistema de archivos de cada dispositivo y serán distribuidas por los mercados de aplicaciones móviles, como el **AppStore** (iOS) o el **Play Store** (Android).

6.7.3.6.9. Características de las aplicaciones móviles nativas

Serán descargadas desde el mercado de aplicaciones para el sistema operativo: Play Market o el AppStore.

Es una aplicación distinta por sistema operativo: una para Android y otra para iOS (y en muchos casos una para iPhone y otra para iPad).

Se desarrollan con lenguajes como Java (Android) u Objective-C (iOS), pero hay alternativas como Appcelerator que permiten desarrollarlas utilizando JavaScript puro, para ambas plataformas.

Ejemplos de aplicaciones nativas: Foursquare, Universiada Nacioanl 2012, Magicab, Instagram, Difusión FCA, entre otras.

➤ Ventajas

- Posibilidad de utilizar funcionalidades nativas: siempre se puede utilizar funcionalidades nativas, como la cámara, el acelerómetro, etc.
- Nuevas alternativas para desarrollar multiplataforma: Aun cuando lo regular es desarrollar estas aplicaciones utilizando las herramientas oficiales (como el Android SDK y Java en Android), han surgido varias nuevas tecnologías para llegar a varios dispositivos con casi el mismo código fuente, como Appcelerator.
- Performance: Típicamente llegan a desempeñarse mejor que las aplicaciones web.
- Hay un “market” para estas aplicaciones: Las personas pueden encontrar tu aplicación utilizando el AppStore o el Play Market. En cualquiera de los casos, estas aplicaciones pueden ser puestas a la venta.
- Modo offline: Tu aplicación puede utilizarse sin necesidad de una conexión a internet.

➤ Desventajas

- Costosas: Por lo general, son más costosas de llevar a cabo, ya que se necesita invertir mucho más tiempo.
- Mayores costos de mantenimiento: De igual forma, mantener varias versiones de la aplicación es costoso.
- Actualizaciones y aprobaciones: Poner una aplicación en el AppStore es tedioso, puede llevar hasta más de una semana. De igual forma, si tienes una nueva funcionalidad o se detectó un error que se quiere arreglar, se debe de esperar otra semana para que Apple apruebe los cambios.

- No se llega a todos los sistemas operativos, por lo cual hay que desarrollar una aplicación para cada uno de ellos.

6.7.3.6.10. Aplicaciones Híbridas.

Luego de hablar de las dos opciones de aplicaciones por qué no hacer una mezcla de ambas opciones, a esto es lo que se le conoce como aplicaciones móviles híbridas, que se han tenido mayor relevancia gracias a PhoneGap.

Este tipo de aplicaciones permite el uso de tecnologías multiplataforma como HTML, JavaScript y CSS pero permite acceder a buena parte de los dispositivos y sensores del teléfono. Buena parte de la infraestructura es tipo web y la comunicación con los elementos del teléfono se hace mediante comunicadores tales como PhoneGap (<http://phonegap.com>). Un buen ejemplo de aplicaciones híbridas es Facebook.

La idea de utilizar aplicaciones híbridas es poder llegar no solo a una plataforma si no que se pueda extender a varias sin necesidad de estar sacando nuevas versiones de la aplicación cada cierto tiempo, para lo cual el uso de lo nativo es para utilizar los beneficios que ofrece el dispositivo como la cámara, acelerómetro, alertas, etc. Y por parte utilizar la tecnología web y el desarrollo web para realizar envío y recepción de información que este en cambio continuo.

6.7.3.6.11. Características

- Todas las ventanas son nativas, estas funcionan como contenedores de vistas web.
- Todas las páginas que son detalles con información son JavaScript y HTML, por su fácil desarrollo.
- Recibe información al instante desde una página web o sistema web.
- La información es gestionada y almacenado por un sistema web en la red.

Cuando las vistas son vistas web, se utiliza una ventana (un Web View) para mostrar el contenido o la página web. Así, nativo sólo es el contenedor o wrapper de la aplicación.

Por el otro lado, todas las ventanas que necesiten funcionalidad nativa del teléfono, como la cámara o detalles que queremos guardar en la aplicación, utilizamos componentes nativos.

Algunas de las herramientas más utilizadas hoy en día para crear aplicaciones híbridas son:

- PhoneGap: Quien permite realizar aplicaciones con HTML5, CSS3 y JavaScript, pero que serán empaquetadas como aplicaciones nativas.
- Trigger.io: Básicamente, casi igual que PhoneGap. Pero con distinta estructura semántica en las propiedades de los elementos.
- Titanium Appcelerator: Utilizando Web Views para embeber un navegador web dentro de alguna ventana nativa.
- Java (Android SDK) u Objective-C (Xcode), utilizando vistas web embebidas en la aplicación.

6.8. Áreas involucradas en el desarrollo de un videojuego

Algo que todos saben en la industria de videojuegos es que el desarrollo de uno de ellos exige el trabajo multidisciplinar de distintos profesionales, esto debido a que su proceso es muy complejo y los mejores títulos están hecho por personas especialistas en su cargo dirigidos por un diseñador de juegos, o el director de proyecto quien debe tener la capacidad de comunicarse con su equipo y tener los conocimientos básicos y lógica que permiten que los juegos funcionen.

Para el caso de los juegos casuales se suele trabajar con equipos pequeños ya que no requieren tanta profundidad en ciertas áreas, lo que no quiere decir que no estén presentes, pero estamos hablando de un medio más acotado e inmediato.

Según el autor Scott Rogers en su libro *LevelUP*, las áreas principales en el desarrollo de un juego son:

- **Programación:** Usando lenguajes de programación como C++, java u otros, un programador escribe líneas de código que permiten que texto y gráficos sean desplegados, desarrollar sistemas de control que permiten al jugador interactuar con el juego, crear sistemas de cámara que permiten ver el mundo del juego,

trabajar en las físicas que lo alteran, programar la inteligencia artificial de los enemigos y el script de cada objeto.

- **Arte:** Inicialmente no habían artistas en los videojuegos y esto estaba a cargo de los programadores o “programadores de arte” debido a que los recursos para representar formas eran demasiado limitados. Sin embargo con el tiempo y la evolución tecnológica surgen herramientas de representación digital donde se hacen necesaria la presencia de los artistas (que perfectamente pueden ser diseñadores gráficos) Éstos se encargan de representar visualmente de mejor manera a personajes, escenarios y estética general de cada juego según su concepto central. Con el tiempo se han subdividido en distintas áreas especializadas, siendo las más comunes las del Artista conceptual, encargado de traducir conceptos en imágenes que dan vida a personajes, mundos y enemigos usando técnicas tradicionales y digitales. Modeladores 3D (cuando el juego lo requiere) quienes están a cargo de tomar los conceptos iniciales de los personajes y cada elemento del juego y transformarlos en modelos de tres dimensiones que serán usados en el juego. Así mismo están los animadores, los diseñadores de interfaz, artistas de escenarios y de efectos visuales, todos ellos a cargo de un Director de Arte quien supervisa el trabajo y se encarga de que todas las piezas mantengan una sola línea estética de acuerdo al proyecto.
- **Diseño de juegos:** Conocidos como director, productor, diseñador líder, o diseñador senior de juegos. Todos cumplen el mismo rol: crear las ideas y reglas que componen el videojuego. Un diseñador de juego necesita manejar muchas habilidades, tener un vasto conocimiento de juegos y saber comunicar por qué un juego es malo y otro es bueno. Dentro de esta área también existen especialidades como el diseñador de niveles, diseñador de sistemas, diseñador de combates, encriptadores, según lo requiera el juego. Todos ellos bajo la supervisión del Director creativo quien mantiene la visión de juego del proyecto y ofrece sugerencias para mejorar el trabajo de los otros diseñadores.
- **Tester:** De la palabra en inglés “Test” podría traducirse como un “probador” y es que es el encargado de probar el juego una y otra y otra vez con la misión de en-

contrar cualquier falla en él (conocidos como “Bugs”), de decir si algún sistema funciona o no, de dar su punto de vista como jugador sobre temas de usabilidad, visualización de datos y otros elementos que puedan interferir en la experiencia del juego. El tester debe tener una gran habilidad de comunicación, paciencia y persistencia a la hora de describir estos problemas para que el equipo pueda trabajar en ellos. En el mercado se usa la sigla QA de Quality assurance para expresar que cuentan con el riguroso estándar de calidad que garantiza que ese juego está (casi) libre de problemas (bugs). Teniendo esto el juego recién está listo para ser lanzado al mercado.

- **Composición:** La música es un factor muy importante para la experiencia del juego y un compositor se encarga de eso. Actualmente la tecnología permite simular instrumentos y grabar canciones directamente desde un teclado conectado al computador y un sintetizador. También muchos de ellos son capaces de crear piezas de orquesta completa, sin embargo los software de audio actuales son suficientes para mezclar sonidos profesionales en una muestra. Este trabajo queda a cargo del diseñador quien debe analizar cada muestra y seleccionar las que van de la mano con la visión de su juego.
- **Diseño sonoro:** Es el área encargada de crear todos los efectos de sonido que serán usados en el juego, un trabajo no menor ya que hay mucha información que se le da al jugador por medio del sonido. El profesional debe ser partícipe del diseño del juego mismo ya que debe saber como comunicar sonidos positivos y negativos dependiendo de las acciones del jugador, respetando los tiempos y niveles de sonido según las mecánicas del juego en cuestión.
- **Guión:** Si bien la idea principal del videojuego reside en el diseñador de las mecánicas de juego, el papel del escritor entra en juego cuando se quiere dar una coherencia a la historia de trasfondo, para escribir diálogos entre sus personajes, cuando se requieren elementos en el juego que sean fáciles de entender como puntos explicativos o tutoriales y cuando queremos añadir cosas como manuales, biografías de cada personaje o información extra que se quiera agregar dentro y fuera del juego mismo. Por supuesto el encargado debe saber escribir, usar

una gramática apropiada y saber cómo escribir en formatos digitales, pero por sobre todo saber escribir para videojuegos, lo que es algo distinto a escribir un libro o una película.

Aparte de los ya mencionados pueden haber muchas otras áreas que un proyecto requiere para ser llevado a cabo: Producción, Dirección técnica, Publisher, Product Manager, Bussines Development por mencionar algunas otras. Todo depende de la envergadura del proyecto y los requerimientos del equipo de trabajo.

6.9. El efecto de inmersión

Dijimos que una de las mejores características de los videojuegos es el efecto de inmersión que generan en sus jugadores, pero para entender mejor de qué trata este fenómeno debemos entender unos conceptos antes.

Cuando hablamos de juegos, a diferencia de otro medio de entretenimiento, hay que tener claro que la esencia de ellos está en su naturaleza interactiva, no existe juego sin un jugador, a esta acción de interacción, de someterse a las reglas en la estructura del juego y las experiencias que se generan mientras lo juega es lo que se denomina Gameplay o jugabilidad. La “experiencia de juego” se puede definir como un ensamblaje hecho por las sensaciones, sentimientos, pensamientos, acciones y construcciones del jugador en las opciones que ofrece la jugabilidad del videojuego.

Actualmente en la cultura digital, el término Gameplay es utilizado para evaluar la calidad de un juego al jugarlo, los retos que presenta, la innovación en su diseño, por cómo se comporta siendo un medio de entretenimiento. Sin embargo, en este contexto el término adquiere una cuota de subjetividad, ya que dentro de la “experiencia de juego” de uno lo influyen otros parámetros externos al producto como el contexto en el que es jugado, las experiencias anteriores del usuario, el desafío presente en relación al rango de habilidades que tiene el jugador, etc.

Como vemos este parámetro no es completamente definible pero se refiere siempre a la relación entre jugador-videojuego y sus variables. Desde este punto, se dice que un juego posee una buena jugabilidad cuando esta relación logra satisfacer las expectati-

vas del jugador, que dependen de las habilidades, del reto que se propone, de la evolución de tal, etc. pero a pesar de que varían según cada persona, el fin último es la diversión.

Un juego se considerará divertido cuando en el proceso de jugarlo se presentan constantemente desafíos que ponen a prueba las habilidades cognitivas, de velocidad, los reflejos, etc. Cuando su jugabilidad es nueva para el usuario, éste se somete a un estado de aprendizaje y reaprendizaje constantemente, esta estética de la repetición es la que caracteriza el placer de jugar el videojuego. Sin embargo, luego de un esfuerzo suficiente y de su repetición, el jugador llega a un punto donde domina el juego y eventualmente el acto de jugarlo llega a un punto de automatización en el que ya no se hace tan divertido. Por lo tanto no es suficiente con que el usuario se siente y observe y posiblemente participe en algún punto, éste debe volverse un participante activo en todo el proceso de jugar, por lo que el juego debe ir evolucionando a medida que evolucionan también las habilidades del jugador. Este tipo de participación genera fuertes experiencias de jugabilidad y captan fuertemente la atención del usuario llevándolo a un estado de inmersión, que puede ser definido como “La sensación de estar rodeado por otra realidad que toma toda nuestra atención, nuestro aparato perceptual completo”.

6.9.1. Modelo de experiencia en el Gameplay: inmersión en los videojuegos.

En base a los estudios realizados por Laura Ermi y Frans Mayra tomamos el modelo de experiencia en el Gameplay, denominado por ellos como modelo SCI que trabajaron en base a la observación de niños video-jugadores con un especial interés en el fenómeno de la inmersión, una representación heurística diseñada para guiar la atención a las dinámicas complejas que están envueltas en la interacción entre el juego y su jugador. A partir del modelo, ellas separan la experiencia en el Gameplay en 3 dimensiones basadas en la inmersión:

- **Inmersión sensorial:** Referido a la experiencia relacionada a la ejecución audiovisual del juego, elemento que es percibido incluso por los menos cercanos a ellos. La evolución tecnológica en los juegos digitales ha evolucionado de manera impresionante, tanto gráfica como sonora, las que fácilmente superan la infor-

mación sensorial que viene del mundo real, estimulando a los jugadores a concentrarse en el juego. Según las áreas involucradas en el desarrollo de un videojuego, lógicamente esta área está a cargo del trabajo en arte, la composición y el diseño sonoro.

- **Inmersión basada en desafíos:** Fundamentalmente basada en la interacción, es la sensación de inmersión más poderosa cuando se logra equilibrar habilidades del jugador con los desafíos del juego, elemento central en los juegos. Los desafíos pueden sugerir habilidades motoras o mentales, aunque en general tienen un poco de ambos. El área encargada del desarrollo de ésta es la del diseño de juegos.
- **Inmersión imaginativa:** Dimensión de la experiencia relacionada a los elementos de historia, personajes, mundos creados para el videojuego, donde uno es absorbido por el interés que generan, o porque se sienten representados con el personaje, área que ofrece al jugador la posibilidad de usar su imaginación enfatizada por los personajes, o simplemente disfrutar con la fantasía del juego. Esta dimensión está directamente relacionada con el área de guión, pero representada también por el resultado final al que llegan desde el arte conceptual y la composición.

A partir de la relación de estas 3 dimensiones, sumado a una serie de otros componentes fundamentales que cumplen un rol en la experiencia del Gameplay, Ermi y Frans nos proponen el siguiente diagrama:

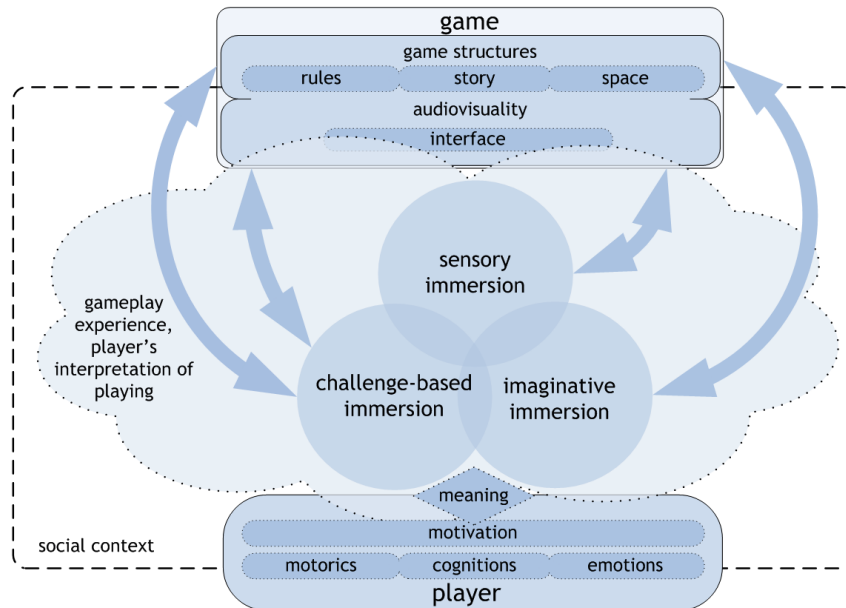


Figura. 9 Componentes Fundamentales que cumplen un rol en la experiencia del Gameplay (web, 2011)

Hay que entender que en todo juego estas 3 dimensiones se relacionan y se superponen constantemente, hay casos en los que se potencia una sobre otra pero eso no quiere decir no estén presentes, también es importante mencionar que la dimensión de inmersión basada en desafíos tiene un rol esencial en los juegos ya que requiere que el usuario tenga una participación activa con el medio (a diferencia de otros medios de entretenimiento como películas o libros, los que sólo pueden otorgarnos las otras dos dimensiones).

6.9.2. Implicancia del diseño gráfico en el efecto de inmersión en el videojuego:

Basados en el modelo visto previamente podemos sacar algunas conclusiones cuando nos preguntamos sobre el papel del diseño gráfico en el área de los videojuegos y en el efecto de inmersión que generan.

Observando el diagrama anterior inmediatamente podemos relacionar el diseño gráfico con la inmersión sensorial, debido a la naturaleza visual de la disciplina, podemos incluso acortarlo más a la inmersión sensoria relacionada a interfaces, estilo visual y arte conceptual con la que debe contar un videojuego, elementos que exigen gran calidad y

que crece a medida que la tecnología avanza. Sin embargo el diseño gráfico ha evolucionado con el tiempo y ahora extrapola estos límites y se relaciona más con el área perceptual de los proyectos, en este sentido la disciplina se puede ligar a áreas como: estructura de juego, interactividad, audiovisual, comunicación, representación, elementos que están presentes en las tres dimensiones del diagrama.

Al hacer una búsqueda en internet por juegos inmersivos, podemos encontrar más de un artículo que hace una lista de los juegos más inmersivos. Al leerlas encontraremos que más de un título se repite entre ellas como Bioshock, Amnesia, Mass Effects series, son comunes entre ellas, y si nos ponemos a analizar a cada uno podremos notar fácilmente que todos comparten un trabajo profundo en su guión, arte, música y efectos especiales que hacen que los juegos se vean realistas en entornos totalmente fantásticos. Estos trabajos están realizados por grandes compañías compuestas por más de 400 trabajadores. Es lógico pensar que en Nicaragua aún no hay profesionales suficientes para este tipo de desarrollo por lo que no pueden ser un referente aún a tener en cuenta si queremos crear videojuegos inmersivos, sin embargo nos dan una idea de qué elementos tener presentes y tomando algunos consejos de Scott Rogers podemos mencionar algunos:

Cuidar el desarrollo de la historia, entre más elementos hayan que no se relacionen a ella más se pierde la inmersión

Cada personaje debe tener su personalidad única con la que el jugador puede representarse o no, cuando hay varios personajes das la posibilidad de elegir el perfil que a él más le acomode.

Entre menos elementos de interfaz hayan en la escena, más inmersiva es la experiencia de juego, por lo que debe estar muy bien trabajada con los elementos precisos y gran simplicidad.

Si en el diseño de escenarios se ven lugares a los que el jugador podría acceder, entonces debiese ser así, nada rompe la inmersión más que espacios que están bloqueados por “muros invisibles”.

La música es un elemento fundamental en el proceso de inmersión, debe acompañar lo que se ve y se juega de manera coherente y a un volumen apropiado.

No hay nada más inmersivo que un trabajo de desafíos exponencial que vaya creciendo a medida que la habilidad del jugador crezca, se puede atraer a jugador solo por las mecánicas y es esta característica lo que hace a un juego divertido o no.

6.10. Transmitiendo identidad a través de los videojuegos:

Las características que hemos analizado de los videojuegos no solo sirven para divertirse sino también para entregar conocimiento. Como sabemos cuándo una experiencia de juego es grata repercute de manera fuerte en el consciente del jugador, recordándola y asociándola como una experiencia pasada que tendrá de referente para futuros juegos, dando como resultado una identidad que se posiciona y diferencia de entre otros juegos y otras experiencias de juego que haya tenido anteriormente. Habiendo sucedido el fenómeno de inmersión, el usuario no solo es capaz de recordar ciertos contenidos necesarios para jugar el juego, sino también relaciona hechos e información que se le entregó con su entorno y sus propias experiencias como persona, este punto es crucial para la transferencia de identidad, cuando el jugador le da sentido a lo que experimentó y lo asocia a conceptos en su vida diaria.

Los videojuegos son parte de la cultura popular actual y se espera que su influencia crezca con los años, lo que los transforma en un nuevo medio para transmitir contenido. Si bien están condicionado a la diversión, hemos analizado distintas aristas que pueden ser usadas para entregar información respecto a lo que se juega. Actualmente en Nicaragua, el desarrollo de videojuegos requiere iniciar una búsqueda en nuestra propia cultura para crear productos originales y valorados a nivel internacional.

6.11. Ambientes Gráficos

Un ambiente gráfico es la contraparte del ambiente carácter (MS-DOS por ejemplo). Una de las características principales es que cuentas con una gama de colores y resoluciones mucho mayor que el ambiente carácter. Puedes pintar sectores mucho pequeños en la pantalla (llamados pixel) lo que permite crear objetos muy bien definidos en la

pantalla (como la ventanas de Windows por ejemplo). Al poder utilizar más recursos visuales, éstos se han hecho muy populares entre los usuarios porque son atractivos y mucho más fáciles de utilizar que las feas pantallas carácter.

6.11.1. Graficas 2d de computadoras

El primer avance en la computación gráfica fue la utilización del tubo de rayos catódicos. Hay dos acercamientos a la gráfica 2d: vector y gráficos raster. La gráfica de vector almacena datos geométricos precisos, topología y estilo como posiciones de coordenada de puntos, las uniones entre puntos (para formar líneas o trayectos) y el color, el grosor y posible relleno de las formas. La mayor parte de los sistemas de vectores gráficos también pueden usar primitivas geométricas de forma estándar como círculos y rectángulos etc. En la mayor parte de casos una imagen de vectores tiene que ser convertida a una imagen de trama o raster para ser vista.

Los gráficos de tramas o raster (llamados comúnmente Mapa de bits) es una rejilla bidimensional uniforme de pixeles. Cada pixel tiene un valor específico como por ejemplo brillo, transparencia en color o una combinación de tales valores. Una imagen de trama tiene una resolución finita de un número específico de filas y columnas. Las demostraciones de computadora estándares muestran una imagen de trama de resoluciones como 1280 (columnas) x 1024 (filas) pixeles. Hoy uno a menudo combina la trama y lo gráficos vectorizados en formatos de archivo compuestos (pdf, swf, svg).

Con el nacimiento de las estaciones de trabajo (como las máquinas LISP, Paintbox computers y estaciones de trabajo Silicón Graphics) llegaron los gráficos 3D, basados en la gráfica de vectores. En vez de que la computadora almacene la información sobre puntos, líneas y curvas en un plano bidimensionales, la computadora almacena la posición de puntos, líneas y típicas caras (para construir un polígono) en un Espacio de tres dimensiones.

Los polígonos tridimensionales son la sangre de prácticamente todos los gráficos 3d realizados en computadora. Como consiguiente, la mayoría de los motores de gráficos de 3D están basados en el almacenaje de puntos (por medio de 3 simples coordenadas Dimensionales X,Y,Z), líneas que conectan aquellos grupos de puntos, las caras son

definidas por las líneas, y luego una secuencia de caras crean los polígonos tridimensionales.

El software actual para generación de gráficos va más lejos de sólo el almacenaje de polígonos en la memoria de computadora. Las gráficas de hoy no son el producto de colecciones masivas de polígonos en formas reconocibles, ellas también resultan de técnicas en el empleo de Shading(Sombreadores), texturing(Texturizado o mapeado) y la rasterización (En referencia a mapas de bits). (Ingsoft2011, 2012)

6.11.2. Gráficos de ordenador en 3D

en contraste con los gráficos por ordenador 2D, son gráficos que utilizan una representación tridimensional de datos geométricos (a menudo cartesiana) que se almacena en el ordenador a efectos de la realización de cálculos y la representación de imágenes en 2D. Estas imágenes se pueden almacenar para su posterior visualización o se muestra en tiempo real.

Gráficos por ordenador en 3D se basan en muchos de los mismos algoritmos como ordenador en 2D gráficos vectoriales en el modelo de alambre y 2D informáticos gráficos por puntos en la pantalla final renderizada. En el software de gráficos de ordenador, la distinción entre 2D y 3D en ocasiones es borrosa; aplicaciones 2D pueden utilizar técnicas 3D para lograr efectos como iluminación y 3D pueden usar técnicas de representación 2D.

Gráficos por ordenador en 3D se refieren a menudo como modelos en 3D . Aparte de la gráfica mostrada, el modelo está contenida dentro del archivo de datos gráficos. Sin embargo, hay diferencias. Un modelo 3D es la matemática representación de cualquier tridimensional del objeto. Un modelo no es técnicamente un gráfico hasta que aparezca. Debido a la impresión en 3D , los modelos 3D no se limitan al espacio virtual. Un modelo puede mostrar visualmente como una imagen en dos dimensiones a través de un proceso llamado *3D* o utilizar en no gráficos simulaciones y cálculos.

6.11.2.1. Historia

William Fetter se le atribuye haber acuñado el término *infografía* en 1961 para describir su trabajo en Boeing . Una de las primeras manifestaciones de la animación por ordenador fue *Futureworld* (1976), que incluyó una animación de un rostro humano y una mano que había aparecido originalmente en el 1971 corto experimental *A Mano animada por computadora* , creado por la Universidad de Utah, los estudiantes Edwin Catmull y Fred Parke. (Wayne, 2010)

6.11.2.2. Panorama general

Equipo de creación de gráficos 3D se divide en tres fases básicas:

- Modelado en 3D - el proceso de formación de un modelo de ordenador de la forma de un objeto
- Diseño y animación - el movimiento y la colocación de los objetos dentro de una escena
- 3D - los cálculos de la computadora que consiste en la colocación de luz, tipos de superficie, y otras cualidades, generan la imagen.

6.11.2.3. Modelado

El modelo describe el proceso de formación de la forma de un objeto. Las dos fuentes más comunes de los modelos 3D son las que un artista o ingeniero se origina en el equipo con algún tipo de herramienta de modelado 3D y modelos escaneados a un ordenador de objetos del mundo real. Los modelos también pueden ser producidos procesalmente o por medio de simulación física . Básicamente, un modelo 3D se forma a partir de puntos llamados *vértices* (o *vértices*) que definen la forma y la forma de *polígonos*. Un polígono es una superficie formada a partir de al menos tres vértices (un triángulo). Un polígono de cuatro puntos es un *quad* y un polígono de más de cuatro puntos es un *n-gon*. La integridad global del modelo y su idoneidad para el uso en la animación depende de la estructura de los polígonos.

6.11.2.4. Animación

Antes de pronunciar en una imagen, los objetos deben ser colocados (trazado) en una escena. Esto define las relaciones espaciales entre objetos, incluyendo ubicación y tamaño . Animación se refiere a la descripción *temporal* de un objeto, es decir, cómo se mueve y deforma con el tiempo. Métodos más populares son fotogramas clave , cinemática inversa y captura de movimiento . Estas técnicas se utilizan a menudo en combinación. Al igual que con el modelado, simulación física también especifica movimiento.

6.11.2.5. Rendering

Rendering convierte un modelo en una imagen ya sea mediante la simulación de transporte de la luz para obtener imágenes foto-realistas, o mediante la aplicación de algún tipo de estilo que en la representación no fotorrealista. Las dos operaciones básicas en la representación realista son el transporte (cantidad de luz que recibe de un lugar a otro) y dispersión (cómo las superficies interactúan con la luz). Este paso se realiza normalmente utilizando software de gráficos 3D o una API de gráficos 3D . La alteración de la escena en una forma adecuada para la representación también implica la proyección en 3D , que muestra una imagen tridimensional en dos dimensiones.

Ejemplos de renderizado 3D



Figura. 10 Renderizado 3D con trazado de rayos y oclusión ambiental (licuadora y YafaRay).



Figura. 11 Modelo 3d de un Dunkerque clase acorazado representa con sombreado plano.

Durante la representación 3D paso, el número de reflexiones "rayos de luz" pueden tomar, así como varios otros atributos, se puede adaptar para conseguir un efecto visual deseado. Dictada con cobalto.

6.11.2.6. Comunidades

Hay una multitud de sitios web diseñados para ayudar a educar y apoyar a los artistas gráficos 3D. Algunos son administrados por los desarrolladores de software y proveedores de contenido, pero hay sitios independientes también. Estas comunidades permiten que los usuarios pueden buscar consejo, publicar tutoriales, proporcione comentario o enviar ejemplos de su propio trabajo.

6.11.2.7. Distinción de gráficos 2D fotorrealistas

No todos los gráficos de ordenador que aparecen 3D se basan en un modelo de alambre. Gráficos 2D con 3D fotorrealistas efectos se logran a menudo sin modelos de alambre y son a veces indistinguibles en la forma final. Algunos software de arte gráfico incluye filtros que se pueden aplicar a los gráficos vectoriales 2D o gráficos raster 2D en capas transparentes. Artistas visuales también pueden copiar ni visualizar los efectos 3D y manualmente rendir efectos fotorrealistas sin el uso de filtros. (Wikipedia, 2013)

6.12. Metodología SUM para Videojuegos

La metodología SUM para videojuegos tiene 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 eficientemen-

te 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.

6.12.1. Roles

La metodología define cuatro roles: equipo de desarrollo, productor interno, cliente y verificador beta. El productor interno y el cliente se corresponden en forma directa con los roles de Scrum Master y Product Owner de Scrum respectivamente.

El equipo de desarrollo tiene las características del Scrum team, pero a diferencia de Scrum se definen sub roles dentro del equipo. _Estos se corresponden con los que se utilizan habitualmente en la industria local y son los de programador, artista grafico, artista sonoro y diseñador de juego. Es necesaria esta definición ya que se requiere una alta especialización para satisfacer las distintas disciplinas que involucra del desarrollo de videojuegos, aspecto no contemplado en Scrum.

El rol de verificador beta no está presente en Scrum pero si se detecta su existencia en el relevamiento de la realidad local y en la industria del videojuego en general. Su responsabilidad es la de realizar la verificación funcional del videojuego y comunicar su resultado.

6.12.2. Ciclo de Vida

El ciclo de vida se divide en fases iterativas e incrementales que se ejecutan en forma secuencial con excepción de la fase de gestión de riesgos que se realiza durante todo el proyecto. Las cinco fases secuenciales son: concepto, planificación, elaboración, beta y cierre, como se aprecia en la Figura12, las fases de concepto, planificación y cierre se realizan en una única iteración, mientras que elaboración y beta constan de múltiples iteraciones.

Los objetivos principales de cada fase son los siguientes:

Concepto: Tiene como objetivo principal definir el concepto del videojuego lo que implica definir aspectos de negocio (publico objetivo, modelo de negocio), de elementos de juego (principales características, gameplay, personajes e historia entre otros) y técnicos (lenguajes y herramientas para el desarrollo). El concepto del videojuego se construye a partir de ideas y propuestas de cada rol involucrado sobre los aspectos a definir. Las propuestas se rellenan a través de reuniones y se analiza su factibilidad con pruebas de concepto. Esta fase finaliza cuando se tiene el concepto validado entre todas las partes involucradas.

Planificación: La fase tiene como objetivo principal planificar las restantes fases del proyecto. Para ello es necesario definir el cronograma del proyecto junto con sus principales hitos, conformar el equipo para la fase de elaboración de acuerdo a las necesidades técnicas del proyecto, determinar y tercerizar las tareas que el equipo no pueda cumplir, definir el presupuesto y especificar las características. Esto _ultimo consiste en describir, estimar y priorizar cada una de las características funcionales y no funcionales que definen el videojuego. Una característica funcional representa una funcionalidad del videojuego desde el punto de vista del usuario final, mientras que, una característica no funcional representa una propiedad o cualidad que el videojuego debe presentar. La planificación que se obtiene en esta fase es flexible ya que en cada iteración de la fase de elaboración se puede modificar para adaptarse a los cambios y reflejar la situación actual del proyecto.

Elaboración: El objetivo de esta fase es implementar el videojuego. Para ello se trabaja en forma iterativa e incremental para lograr una versión ejecutable del videojuego al finalizar cada iteración. _Estas se dividen en tres etapas, en la primera se planifican los objetivos a cumplir, las métricas a utilizar en el seguimiento, las características a implementar y las tareas necesarias para ello. En la segunda se desarrollan las características planificadas a través de la ejecución de las tareas que la componen. Al mismo tiempo se realiza el seguimiento para mantener la visión y el control de la iteración en base a los objetivos planteados. La tercera y _ultima implica la evaluación del estado del videojuego y de lo ocurrido en el transcurso de la iteración para actualizar el plan de proyecto respecto a la situación actual. Con esta forma de trabajo se puede evaluar el

avance del proyecto, lo cual permite realizar cambios a tiempo y tomar decisiones para cumplir con los plazos planificados. Además, la experiencia adquirida permite mejorar la forma de trabajo en cada iteración y aumentar la productividad.

Beta: La fase tiene como objetivos evaluar y ajustar distintos aspectos del videojuego como por ejemplo gameplay, diversión, curva de aprendizaje y curva de dificultad, además de eliminar la mayor cantidad de errores detectados.

Cierre: Esta fase tiene como objetivos entregar la versión final del videojuego al cliente según las formas establecidas y evaluar el desarrollo del proyecto. Para la evaluación se estudian los problemas ocurridos, los éxitos conseguidos, las soluciones halladas, el cumplimiento de objetivos y la certeza de las estimaciones.

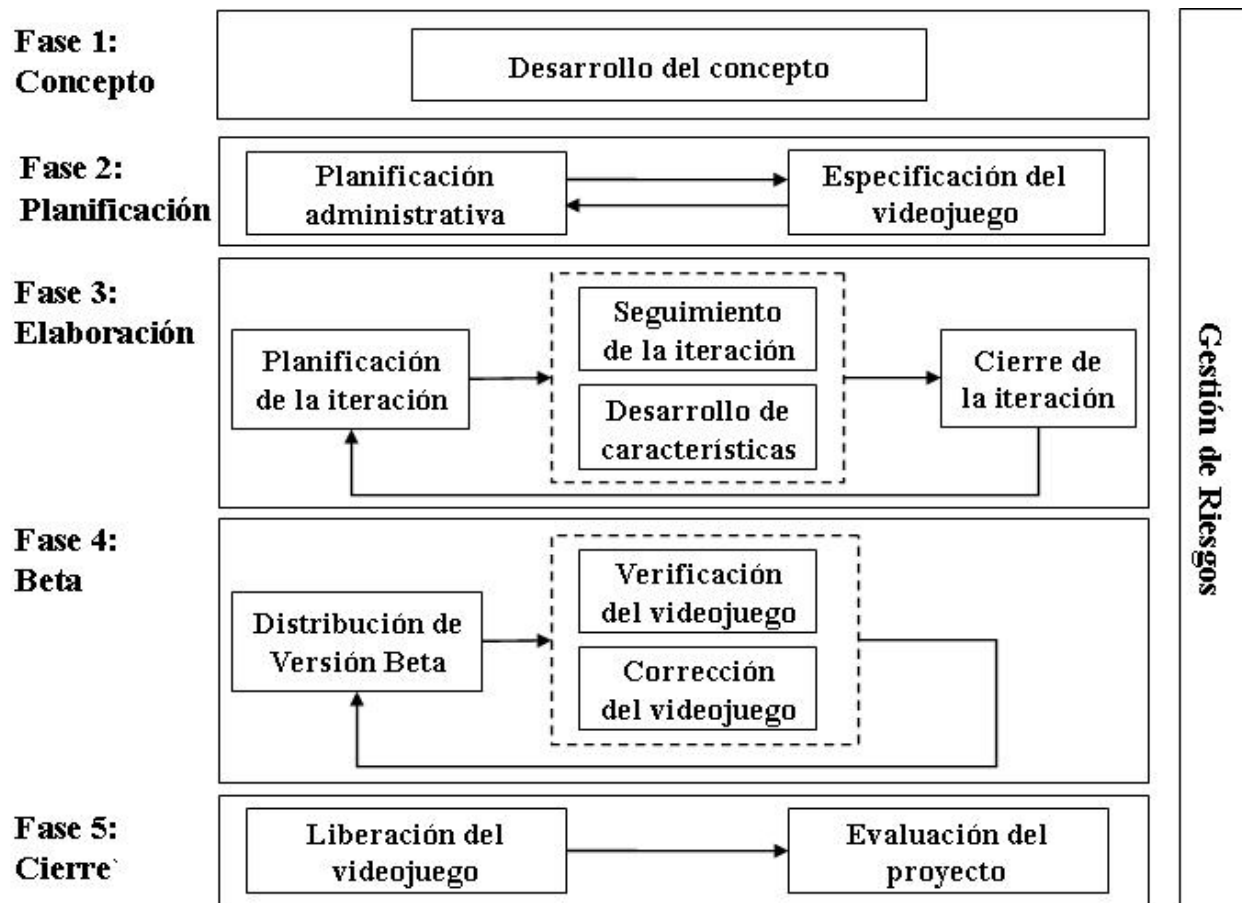


Figura. 12 Fases del proceso de la metodología SUM.

VII. DISEÑO METODOLOGICO

7.1. Tipo de estudio

El tipo de investigación es **Exploratoria** ya que el desarrollo de videojuegos es un tema poco estudiado en Nicaragua y lo que se quiere con este trabajo es conocer ampliamente el campo de desarrollo y las técnicas necesarias para producir videojuegos, es al mismo tiempo una investigación **aplicada** con un periodo de tiempo determinado en el cual se plantea tener como producto el videojuego con todas sus funcionalidades. Este es un estudio transversal ya que se realizó el análisis de la información recolectada durante un tiempo limitado que corresponde al primer semestre de 2013.

7.2. Metodología

La metodología a utilizar será **Iterativa Incremental** ajustándonos a las fases generales de desarrollo de un videojuego, las cuales son pre-producción, producción y post-producción, con el enfoque orientado a objetos y el lenguaje de modelado UML.

7.2.1. Fases de Pre-producción

En esta fase se realizará la concepción y delimitación de **la idea del videojuego**, se definirá **el género** del juego, la manera en la **que interactuará el jugador con el juego** y con qué **mecanismos** contará para ello y se hará **una breve descripción de la historia del juego**, se abordará de **manera general los ambientes y el personaje del videojuego**.

7.2.2. Fase de Producción

7.2.2.1. Diseño del Juego

En esta etapa se detallarán todos los elementos que compondrán el juego, dejando claramente visible los detalles de cómo será el juego en sus distintas interacciones.

7.2.2.1.1. Diseño artístico

Es este punto se describirá la **Historia del juego** caracterizando al personaje y el ambiente del juego, en si se describirá el mundo donde sucede el juego, creando el hilo argumental completo, con todos los detalles.

7.2.2.1.2. Diseño Mecánico

En este punto se describe cómo se va a interactuar en el juego y las reglas que rigen a éste, se diseñará el comportamiento del personaje principal Nicaboy en función de las interacciones y colisiones con el mundo que le rodea.

7.2.2.1.3. Motor del juego

En este punto se dará a conocer la forma en que se integraran las imágenes, sonido y otras herramientas multimedia dentro del motor de juego Unity 3D.

7.2.2.2. Diseño Técnico

En esta etapa se diseñan los diagramas con el funcionamiento estático y dinámico describiendo la interacción con los usuarios a través de diagramas de casos de uso y diagramas de clases en el primer incremento y detalle de casos de usos y los diferentes estados que atravesará el videojuego descritos en diagramas de secuencia en el segundo incremento.

De igual manera se incluye la planificación del juego y en ella se identifican las tareas necesarias para desarrollarlo, repartiéndolas entre los distintos componentes del equipo desarrollador. También se fijan plazos para la ejecución de dichas tareas con la ayuda de Microsoft Project.

7.2.2.3. Implementación

En esta etapa se realizara el montaje de todas las piezas del juego en el Motor del Juego. Se finalizan todos los contenidos del juego: scripts, efectos, modelado de personajes, sonido, objetos, escenario etc. Esta etapa se desarrolla en dos incrementos los cuales describimos a continuación:

- Primer Incremento
 - Librería
 - Cámara
 - Escenario
 - Modelado
 - Personajes
 - Objetos
 - Control
- Segundo incremento
 - Cámara
 - Objetos
 - Texturas
 - Iluminación
 - Gravedad
 - Colisiones
 - Sonido

7.2.2.4. Pruebas Alpha

En esta etapa ya tendremos una versión funcional del videojuego como resultado del primer incremento a la que se le aplicaran las pruebas alpha, que serán realizadas por el equipo de desarrollo del videojuego, por el tutor de este seminario Msc. Juan de dios Bonilla que ha estado monitoreando cada fase de desarrollo del videojuego, en busca de errores para su refinamiento. Uno de los aspectos a probar es la Jugabilidad.

7.2.2.5. Pruebas Beta

En esta etapa tendremos una versión del videojuego terminada después de efectuar el segundo incremento en el cual se adhieren texturas, iluminación, gravedad, colisiones y sonido. Estas pruebas se realizarán por un equipo externo al equipo de desarrollo, que es integrado por amigos parte de la comunidad de game players de

Nicaragua, el resultado de esta etapa constituye el **GOLD MASTER que es el juego listo para publicar** con un video adjunto que es el manual de usuario, este será un video que explica cómo se juega el videojuego.

7.2.3. Fase de Post Producción

Esta fase está contemplada a desarrollar y ejecutar fuera de este proyecto de seminario.

7.3. Estudio de factibilidad

7.3.1. Factibilidad Técnica

Desde el punto de vista técnico para el desarrollo del videojuego, son necesarios algunos recursos tecnológicos que no corresponden precisamente a un proceso de desarrollo pero que son de vital importancia para lograr nuestro objetivo, el cual consiste en desarrollar el videojuego Nica Adventure en entorno 3d para lo cual se necesita lo siguiente:

- a) Un motor de desarrollo de videojuegos 3D
- b) Programas de diseño gráfico 2D y 3D
- c) Lenguajes de programación
- d) Dos equipos de cómputo para el desarrollo del videojuego
- e) Un teléfono celular para probar el videojuego
- f) Servicio de Internet
- g) Impresora Laser

7.3.2. Factibilidad Económica

En correspondencia a lo planteado anteriormente, describimos los costos tanto de hardware como de software necesarios para el desarrollo del videojuego:

Software/Licencias	cantidad	Precio	Subtotal
Unity 3D 4.0 Pro	1	\$1,200	\$1,200
Autodesk 3Ds Max 2014	1	\$2,000	\$2,000
Photoshop CS6	1	\$ 940	\$ 940
Total Licencias/ Software			\$4140

Tabla 1. Factibilidad económica de software/licencias.

Dispositivos y Servicios	cantidad	Precio	Subtotal
Laptop Toshiba	2	\$740	\$1,480
Celular Samsung Galaxy Note 1	1	\$840	\$ 840
Impresora Laser	1	\$250	\$ 250
Servicio de Internet	1	\$468	\$ 468
Total Dispositivos y Servicios			\$3038

Tabla 2 **Factibilidad económica de dispositivos y servicios.**

Total Factibilidad económica \$7.178

7.3.3. Factibilidad Operativa

A continuación describimos los requisitos óptimos con los que deben contar los equipos de cómputo, el equipo celular y el servicio de internet

➤ **Equipos de Cómputo**

Procesador: Intel Core i3 2.4 Ghz

RAM: 4Gb

Video: 1Gb

Disco Duro: 250Gb

Sistema Operativo: Windows 7

➤ **Equipo Celular:**

Procesador: Dual Core de 1.4 GHz

RAM: 1Gb

Sistema Operativo: Android JellyBean 4.0 en adelante

➤ **Servicio de Internet:**

Velocidad de descarga: 3mpbs

Velocidad de salida: 3mps

Navegacion internacional: 3mpbs

Tipo de Servicio: Cable coaxial no adsl

VIII. DESARROLLO

Como resultado de la metodología Iterativa Incremental se evidencian dos resultados que se desarrollan en dos incrementos:

8.1. RESULTADO 1: FASE DE PRE – PRODUCCION

El videojuego Nica Adventure será estilo carrera y consistirá en un recorrido a contra reloj realizado por Nicaboy nuestro personaje principal; este recorrido se hará siguiendo la ruta de los tesoros de Nicaragua, la cual comprende varios departamentos del país.

El objetivo del videojuego es promover el turismo nicaragüense por eso se diseñaran paisajes y carreteras alusivas a las existentes en la ruta de los tesoros, de igual forma se mostrara información turística e histórica de los lugares que va visitando nuestro personaje, se incluirán contenido multimedia con información sobre la cultura, tradición y gastronomía de los sitios o lugares visitados y al final del recorrido se cargara una escena que simula el sello del pasaporte de visitante garantizando que el jugador conoce de la cultura Nicaragüense.

Dentro del juego el personaje principal tiene el control de su auto con los movimientos del teléfono, realizando desplazamientos hacia atrás, adelante o a los lados, acelerando y desacelerando el vehículo, durante la carrera el jugador se encontrara con muchos obstáculos como barriles de basura, árboles caídos, reductores de velocidad u otros vehículos en la vía.

Durante el juego el jugador podrá observar en el centro de la pantalla la cuenta regresiva, es decir el tiempo que le resta para llegar a la meta, conforme va avanzando en el juego habrán puntos de control en los cuales el jugador puede ir recargando tiempo, ya que si su tiempo llega a 0 el jugador pierde.

El proyecto no contempla que el juego tenga conexión a internet, base de datos, ni interacción con otro dispositivo.

8.2. RESULTADO 1: FASE DE PRODUCCION

8.2.1. DISEÑO DEL JUEGO

En esta sección se expone el entorno argumental del juego, el diseño que se creó en la primera iteración mediante dos guiones, el guion del juego y el guion para el documento de diseño. Además se explica la forma en que se integraran los componentes del videojuego. Teniendo como punto de partida que la idea principal del desarrollo del videojuego es potencializar el turismo Nicaragüense fusionando la adrenalina que contiene los juegos de autos de carrera y la geografía y cultura Nicaragüense.

8.2.1.1. Diseño Artístico - Guion del juego



Figura. 13 Nicaragua turística (CRECER12, 2013)

Nicaragua, ubicada en el corazón de América Central es el destino ideal para los que buscan disfrutar y conocer bellos paisajes turísticos llenos de historia y tradición. Sus dos mares, decenas de volcanes, lagos y ríos hacen el mérito para llamarla Tierra de lagos y volcanes. (CRECER12, 2013)

Sus inmensos atractivos turísticos permiten desarrollar una variedad de actividades como Surfear en las Playas del pacífico, realizar caminatas por volcanes y mágicas lagunas donde se pueden observar cientos de especies de animales y plantas excelentes para la práctica de turismo ecológico. Practicar Kayak entre las calmas aguas de un inmenso lago y paseos por centenares de isletas, fabulosos ríos, visitar ciudades coloniales, tesoros arqueológicos, bucear en las cristalinas aguas del Caribe. Esta es una tierra con una gran riqueza cultural, multiétnica y multilingüe que abre un nuevo camino en donde la solidaridad y el optimismo florecen, hacen de Nicaragua una tierra Única, Original. El arte culinario nicaragüense se remonta a tiempos precolombinos, a como lo

atestiguan los nombres de algunos de sus más conocidos platos. Luego, durante la conquista y colonia española, la unión de dos razas y dos culturas dio como resultado un menú criollo peculiar, creativo y variado, en el que se usan ingredientes interesantes y del que forman parte desde sopas y carnes, hasta dulces de diversos tipos. Los platos típicos tradicionales de Nicaragua son muchísimos. Cada región, según las particularidades culturales y geografía, produjo varios platos, bebidas o dulces que con el pasar de los años pasaron a conocerse en todo el país. Veamos ahora cuales son los platos más conocidos (El gallo Pinto, Nacatamal, Vigoron, Indio Viejo, Quesillo, Sopa de Mondongo, El rondón, etc.). (CRECER12, 2013)

Además de sus muchas frutas frescas, que se esperaría en los trópicos, cervezas nacionales y el mundialmente famoso y suave ron nicaragüense. En los últimos años el turismo se ha potencializado ya que empresarios y el gobierno han evidenciado que es este el responsable de la entrada de divisas al país lo que se traduce en su desarrollo y crecimiento a corto, mediano y largo plazo por lo que a finales de 2011 un grupo de empresarios de hoteles y prestadores de servicios ubicados en la ruta de los Tesoros de Nicaragua, han unido esfuerzos para presentar una variada oferta turística nacional, que incluye alojamiento, restaurantes, actividades al aire libre, riquezas culturales, históricas y naturales, todo esto con el objetivo de promover un turismo ecológico, sustentable, para cuidar las bellezas naturales y proteger nuestro patrimonio histórico. (CRECER12, 2013)



Figura. 14 El empresario Enrique Zamora Llanes, el expelotero de Grandes Ligas, Denis Martinez, junto a representantes de American Airlines y de la Camará Nacional de Turismo, en el lanzamiento de la Ruta de los Tesoros de Nicaragua (Ricardo., 2011)

La ruta comprende el eje geográfico que va desde Ocotal, Somoto, Estelí, Matagalpa, León; Granada, Las Isletas, Rivas, San Juan del Sur y Río San Juan. (Ricardo., 2011)

Podemos ubicar la ruta partiendo de Managua, de la siguiente manera: Managua – Granada – Rivas – Río San Juan – Matagalpa – Jinotega – Estelí – Madriz – León.



Figura. 15 Imágenes de la ruta de los tesoros (Ricardo., 2011)

Entre los empresarios que forman parte de la alianza encontramos a Sandra Mejía presidenta de HOPEN, Lorena Zamora, gerente de Nicarao Tours & Travel, Enrique Zamora, propietario de Nicarao Lake Resort, y al expelotero nicaragüense de Grandes Ligas, Denis Martínez, quien manifestó que el nombre de esta ruta se escogió como una forma de destacar la riqueza natural y cultural de Nicaragua, y que en la misma sobresale su pasado colonial, historia, bosques, gastronomía, tradiciones y la conservación y desarrollo económico de las comunidades ubicadas en la zona. (Ricardo., 2011)



Figura. 16 El empresario Enrique Zamora Llanes propietario de Nicarao Lake Resort (Ricardo., 2011)

“Con esta ruta pretendemos promover los variados y atractivos destinos turísticos del país, y motivar al turista nacional e internacional a que disfrute de esta alianza”, explicó Sylvia Levi, Presidente de la Cámara Nacional de Turismo, Canatur. Dentro de esta ruta se encuentran los hoteles La Ceiba, Hotel La Frontera, Los Balcones; Hotel La Mar, Ritzo, Hotel Café de Jinotega, Hotel Selva Negra, Hotel Montecristo, Los Robles y Punta Teonoste. (Ricardo., 2011)

“Cada uno de estos establecimientos podrá establecer alianzas locales con restaurantes, productores de artesanías y propietarios de atractivos turísticos que no tengan infraestructura de hospedaje”.



Figura. 17 Pasaporte de visitante de la ruta de los tesoros (Ricardo., 2011)

Actualmente cuentan con 250 habitaciones y más de 600 cama, la que aumentará con la incorporación de nuevos empresarios, con el objetivo de motivar al turista nacional a que disfrute de la más amplia y variada oferta turística. (Ricardo., 2011)



Figura. 18 Imágenes de la ruta de los tesoros (Ricardo., 2011)

8.2.1.2. Diseño Mecánico - Guión del juego para el documento de diseño

El comienzo para el desarrollo del juego fue el describir el mundo en el que se insertaría al jugador. El crear un guión fue una necesidad para crear una base al diseño de los elementos que contiene el mundo.

La idea es que nuestro auto recorra en una carrera a contra reloj la ruta de los tesoros de Nicaragua, que comprende las siguientes ciudades turísticas: Ocotral-Somoto-Estelí-Jinotega-Matagalpa-León-Managua-Granada-Las Isletas-Ometepe-Rivas-San Juan del Sur-Río San Juan, en el transcurso de esta carrera el jugador deberá llegar a puntos de control “Checkpoint” antes de que se le acabe el tiempo, cada nivel en la carrera será un viaje de una ciudad a otra, en las calles habrán obstáculos, vueltas muy cerradas que dificultaran el desplazamiento del auto y por tanto el control del jugador.

Cuando el auto llega a la ciudad destino se le mostrara una breve reseña histórica y característica del departamento al que ha llegado, se clasificara el tiempo de llegada y según el tiempo recorrido se le otorgara una medalla ya sea ésta oro, plata o bronce, al terminar el recorrido de cada nivel se le otorgara una certificación de visitante que haga

constar que ha estado en dicha ciudad y conoce sobre su historia y principales atractivos turísticos.

8.2.1.3. Motor del videojuego

Para lograr la implementación satisfactoria del videojuego Nica Adventure se ha utilizado como motor de desarrollo UNITY 3D, el cual reconoce diferentes lenguajes de programación, en este caso utilizamos JavaScript y C#, lo cual se hace de la siguiente manera:

- Se diseñaron los objetos 3D ya sean escenarios, personajes etc. en 3Ds Max
- Se importa el objeto en UNITY 3D.
- Este objeto importado trae propiedades o animaciones conocidas como prefabs las que son reconocidas en UNITY 3D.
- En cuanto al sonido se integraron pistas de audios en formato MP3 y se asocio al personaje principal mediante un componente de audio de igual manera se asocio sonido a los CHECKPOINT, de tal forma que cuando el sistema detecta que Nicaboy colisiona se reproduce el sonido asociado.
- Las texturas, son imágenes 2D asociadas a un componente que nos permitieron aportar realismo al ambiente grafico.
- Se agrego iluminación a travez del objeto Directional Light la cual es uniforme para todo el escenario.
- Este motor permite publicar el videojuego a varias plataformas entre ellas: web, flash, para PC, para Android, IOS.

8.2.2. DISEÑO TÉCNICO

8.2.2.1. Diagrama de casos de uso generalizado

El siguiente diagrama muestra gráficamente los casos de uso asociados a este proyecto y además muestra la interacción de los actores con estos. En este proyecto sólo existe un actor que interactúa con el juego, el jugador.

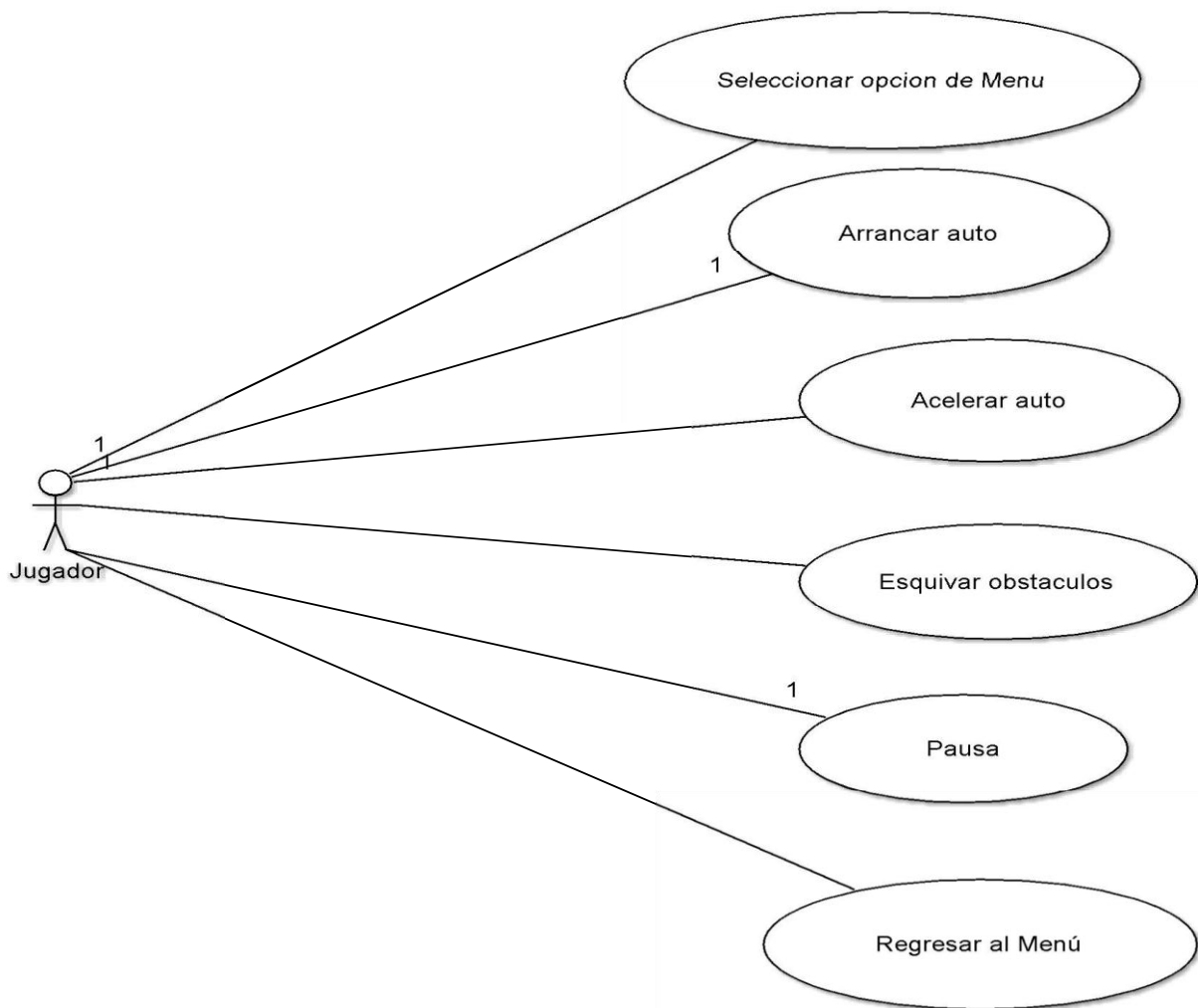


Figura. 19 Diagrama de casos de usos generalizado

8.2.2.2. Detalle de casos de usos

CU-1 Seleccionar opción de Menú.	
Usuarios:	Jugador
Propósito:	Ingresar a cualquiera de las opciones que se muestran en el menú
Precondiciones:	
PostCondiciones:	Se ha seleccionado una de las 3 opciones del menú
DESCRIPCIÓN DETALLADA, SECUENCIA EXITOSA	
1- Este caso de uso inicia cuando el jugador toca alguno de los botones de opción que se muestran en el menú del juego	
	2- El sistema inicia el nivel 1 del videojuego y así termina este caso de uso
	3- El sistema muestra las instrucciones del videojuego y pasa a otra escena y termina el caso de uso
	4- El sistema finaliza el juego

Tabla 3 Caso de uso SELECCIONAR OPCION MENU.

CU-2Arrancar auto.	
Usuarios:	Jugador
Propósito:	Hacer que el auto inicie la carrera
Precondiciones:	Ya debe haber aparecido la palabra GO en la pantalla
PostCondiciones:	El auto ha iniciado a correr
DESCRIPCIÓN DETALLADA, SECUENCIA EXITOSA	
1- Este caso de uso inicia cuando el jugador toca la pantalla del celular, esto causa que el auto empiece a moverse sobre la carretera.	
	2- El sistema mueve el auto hacia adelante.
	3- El sistema mueve la cámara en la misma dirección del auto.

Tabla 4 Caso de uso ARRANCAR AUTO.

CU-3 Acelerar auto	
Usuarios:	Jugador
Propósito:	Hacer que el auto se desplace hacia adelante conforme se acelera
Precondiciones:	El auto ya debe estar en movimiento
PostCondiciones:	El auto acelera y se desplaza hacia adelante incrementando la velocidad
DESCRIPCIÓN DETALLADA, SECUENCIA EXITOSA	
1- Este caso de uso inicia cuando el jugador mueve el celular hacia adelante	
	2- El sistema mueve el auto y va incrementando la velocidad

Tabla 5 Caso de uso ACELERAR AUTO.

CU-1 Esquivar obstáculos	
Usuarios:	Jugador
Propósito:	Hacer que el auto esquive los obstáculos presentes en la carretera
Precondiciones:	El auto debe estar en movimiento
PostCondiciones:	El auto ha esquivado un obstáculo
DESCRIPCIÓN DETALLADA, SECUENCIA EXITOSA	
1- Este caso de uso inicia cuando el jugador inclina el celular hacia la derecha o izquierda para esquivar un obstáculo	
	2- El sistema hace que el auto cambie su dirección.
	3- El sistema hace que el auto se desplace en la dirección según lo inclinación del celular

Tabla 6 Caso de uso ESQUIVAR OBSTACULOS.

CU-1 Pausa	
Usuarios:	Jugador
Propósito:	Pausar el juego
Precondiciones:	El juego debe estar iniciado
PostCondiciones:	El juego se puso en pausa
DESCRIPCIÓN DETALLADA, SECUENCIA EXITOSA	
1- Este caso de uso inicia cuando el jugador toca el botón de pausa que está en la pantalla del juego	
	2- El sistema pone el juego en pausa

Tabla 7 Caso de uso PAUSA.

CU-1 Regresar al Menú.	
Usuarios:	Jugador
Propósito:	Regresar al menú principal para leer instrucciones, reiniciar el nivel o salir del juego
Precondiciones:	El juego debe estar iniciado
PostCondiciones:	Se ha regresado al menú principal
DESCRIPCIÓN DETALLADA, SECUENCIA EXITOSA	
1- Este caso de uso inicia cuando el jugador toca el botón menú que se encuentra en la pantalla del juego	
	2- El sistema abandona el nivel en que se encuentra
	3- El sistema muestra la pantalla de menú principal

Tabla 8 Caso de uso REGRESAR AL MENU.

8.2.2.3. Diagrama de clases simplificado

El diagrama de clases describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos.

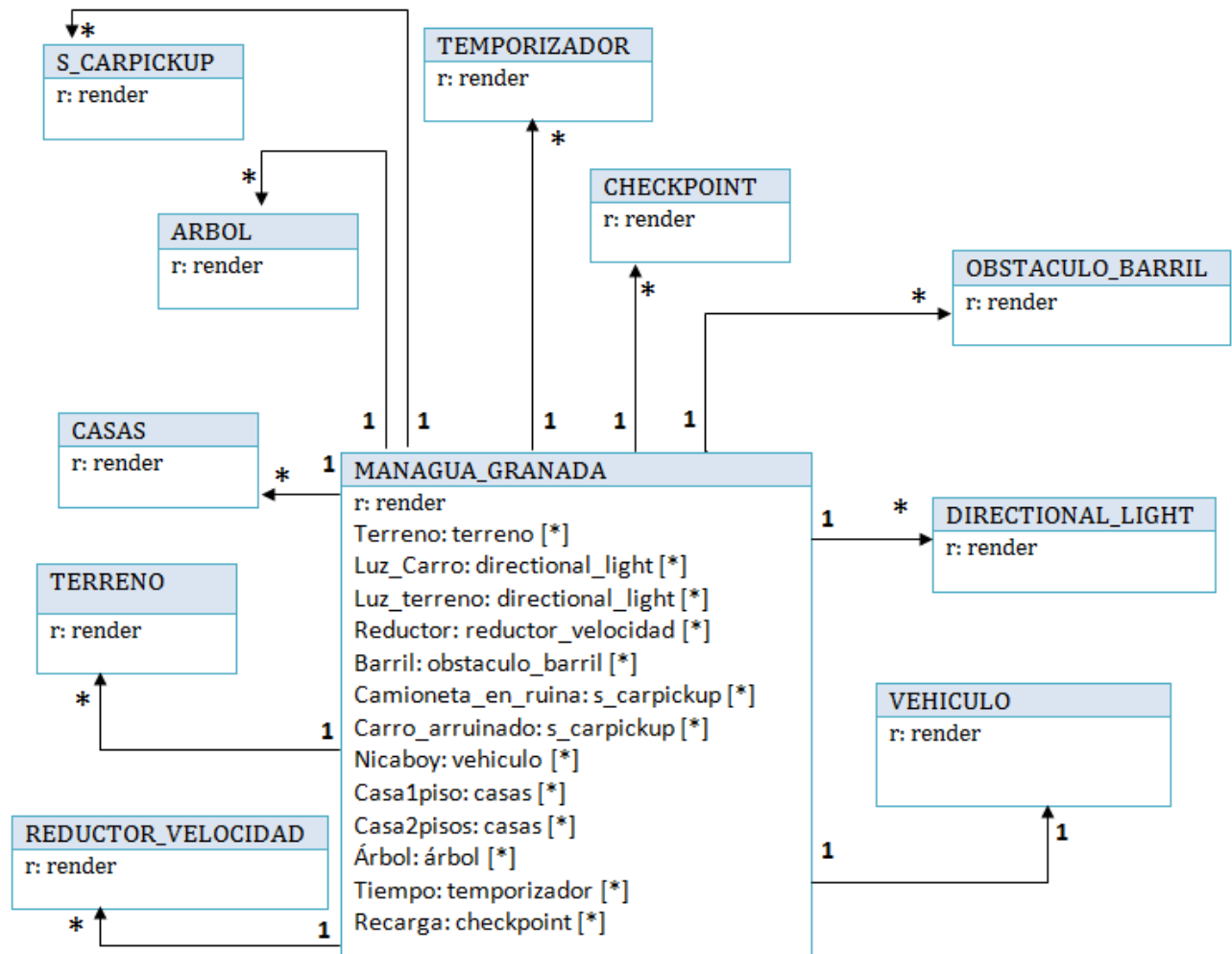


Figura. 20 Diagrama de clases generalizado.

8.2.2.4. Detalle de clases

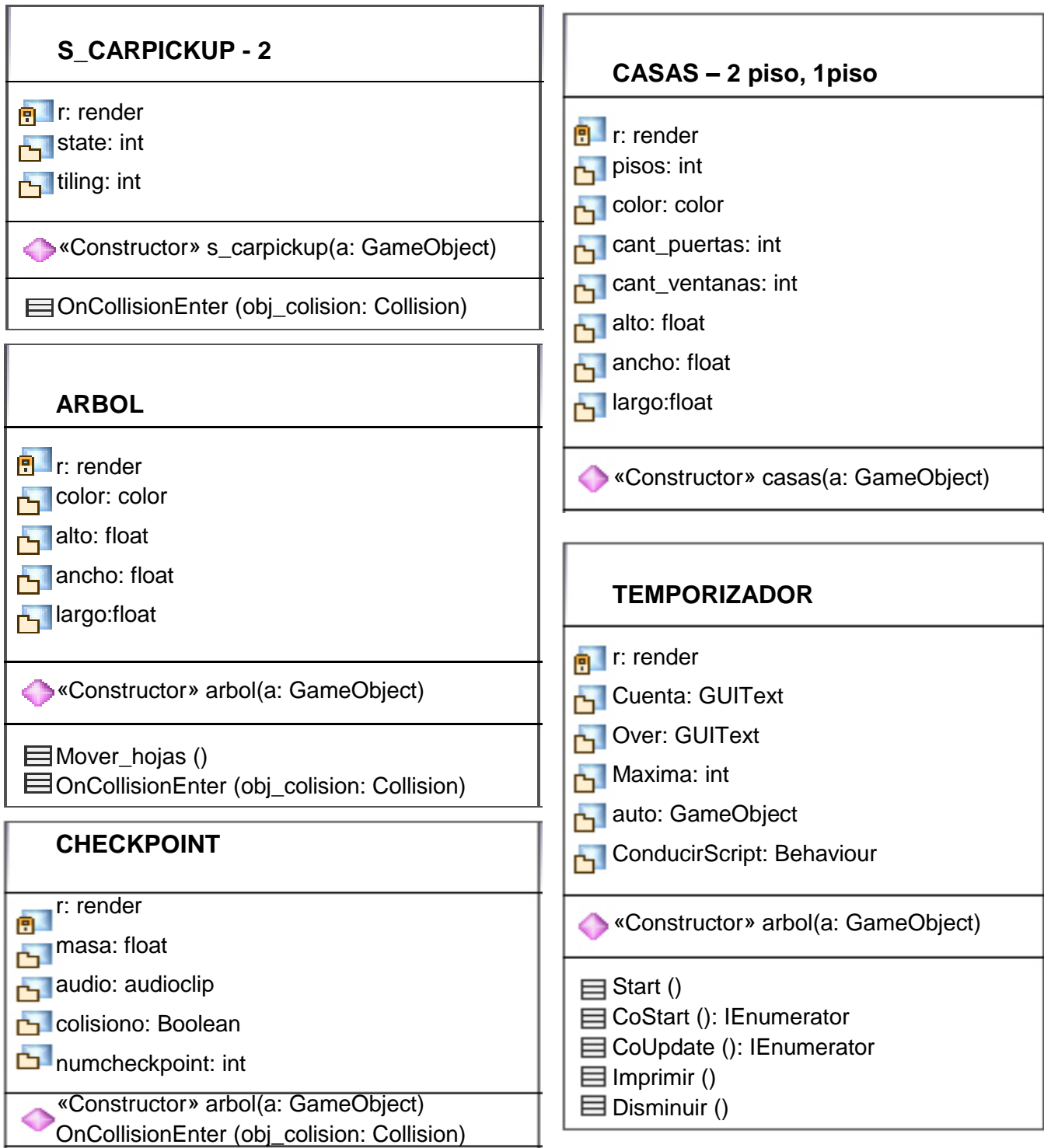


Figura. 21 Clases S_CARPICKUP, CASAS, TEMPORIZADOR, ARBOL, CHECKPOINT

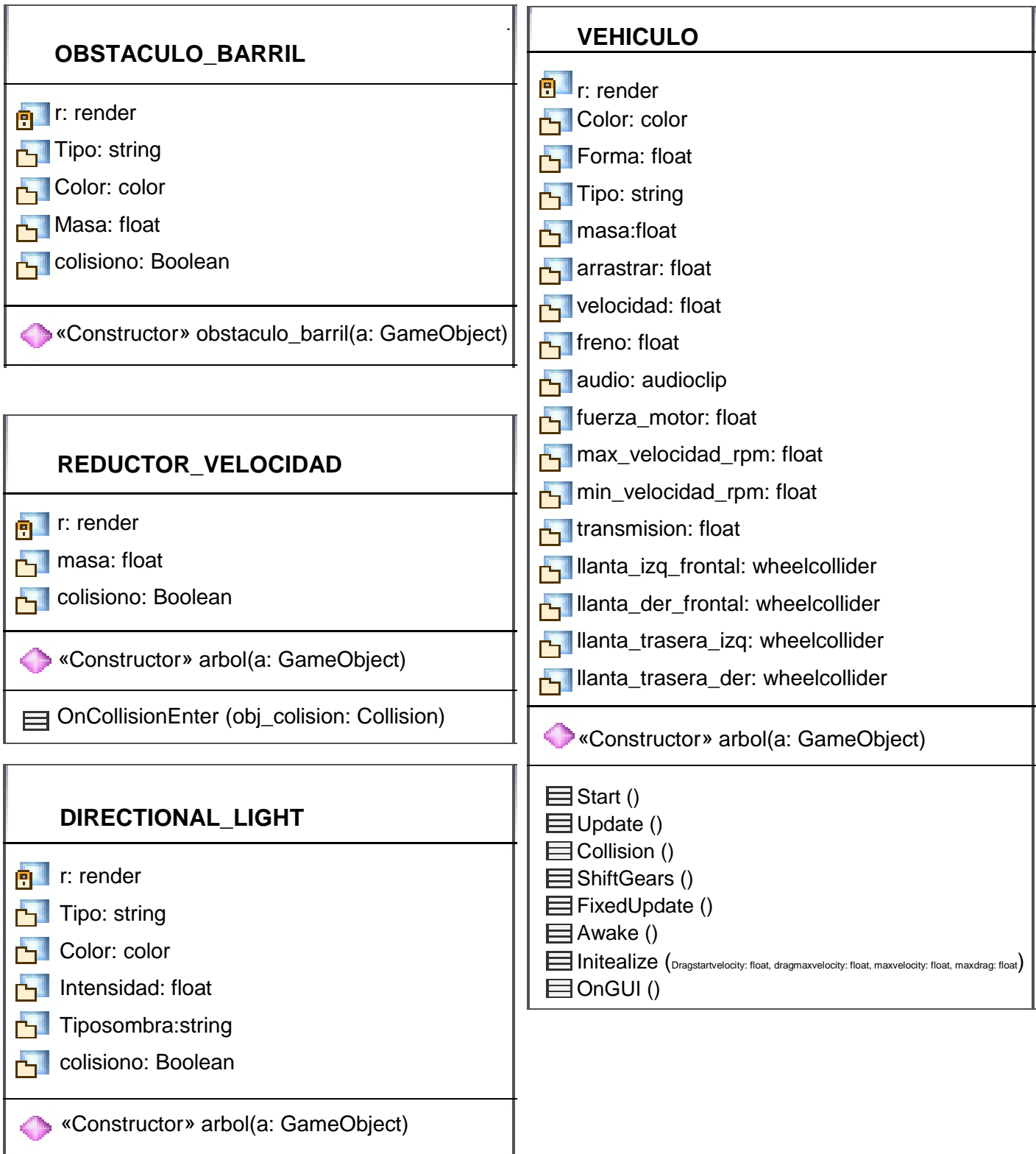


Figura. 22 Clases OBSTACULO_BARRIL, VEHICULO, REDUCTOR_VELOCIDAD, DIRECTIONAL_LIGHT.

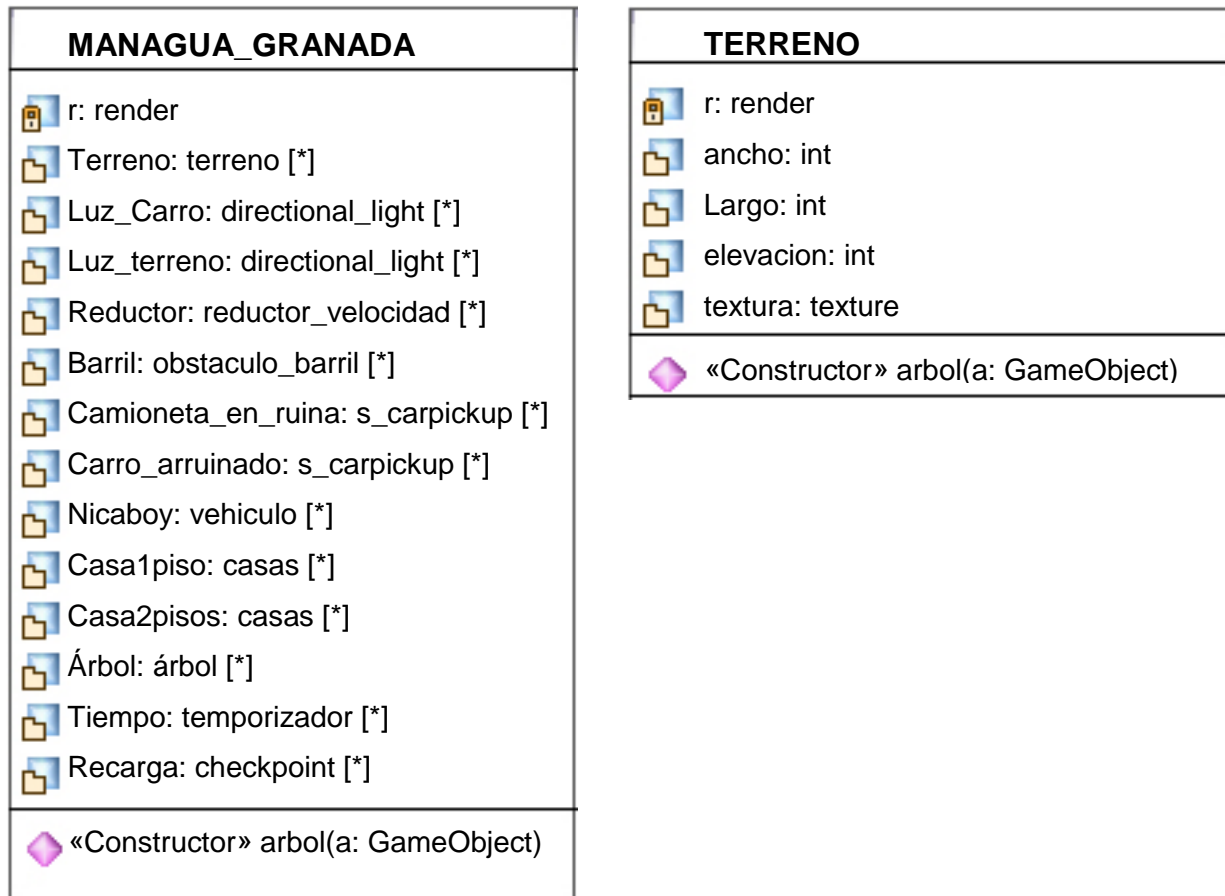


Figura. 23 Clases MANAGUA_GRANADA, TERRENO.

8.2.2.5. Descripción de las clases

S_CARPICKUP:

Representa a una camioneta en ruinas, la cual está ubicada en la carretera y sirve de obstáculo, esta contiene un *meshcollider* que tiene la propiedad de colisión, lo cual permite que cuando nuestro auto colisiona con la camioneta ocurre una acción determinada.

TEMPORIZADOR:

Representa el tiempo que viene disminuyendo y el cual si llega a 0 hace terminar el juego, el tiempo se recarga en cada checkpoint.

ARBOL:

Los objetos de esta clase representan a los arboles de la vida real, plasmados en este mundo virtual de tal forma que den vistosidad a nuestro escenario.

CHECKPOINT:

Son los puntos de control por los cuales debe pasar nuestro personaje para obtener más tiempo y seguir en la carrera.

OBSTACULO_BARRIL:

Representa un barril de metal situado en medio de la carretera con el objetivo de obstaculizar el paso a nuestro personaje y así disminuirle la velocidad, cuenta con colisiones de rebote para que nuestro personaje pierda bastante tiempo.

CASAS:

Representan casas tales como las de la vida real, están situadas a la orilla de la carretera, hay casas de 1 y 2 pisos.

TERRENO:

El terreno representa el plano de tierra en el cual está ubicada la carretera y las montañas que se ven el juego.

MANAGUA_GRANADA:

Representa el tramo de carretera en el cual nuestro personaje hace su recorrido, este tramo ha sido modelado de tal forma que la forma de la carretera sea parecido a la carretera real de Managua a Granda.

DIRECTIONAL_LIGHT:

Representa la luz que ilumina nuestro escenario, una de las luces se le asignó a la main camera que sigue a nuestro personaje.

VEHICULO:

Es el auto que representa a nuestro personaje, cuenta con varias propiedades físicas como un auto real (velocidad, masa, frenos, colisiones entre otras).

REDUCTOR_VELOCIDAD:

Representa policías acostados a como le llamamos aquí en Nicaragua, los cuales tienen el efecto en su colisión de impulsar a nuestro auto según la velocidad con la que va hasta el punto de cambiar su marcha y hacerlo correr en sentido contrario.

8.2.2.6. Diagramas de secuencia

DS: CU – 1 SELECCIONAR OPCION DE MENU

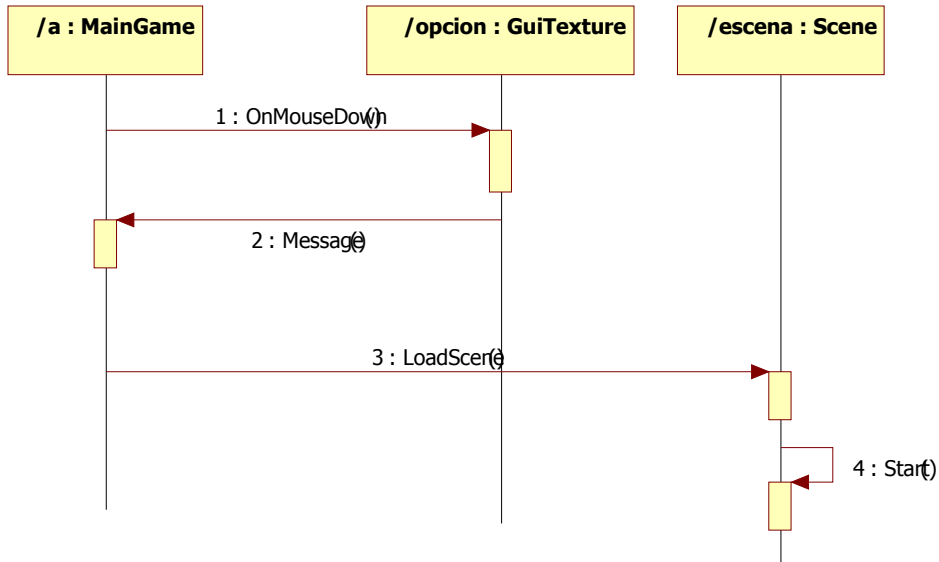


Figura. 24 Diagrama de secuencia 1

DS: CU – 2 ARRANCAR AUTO

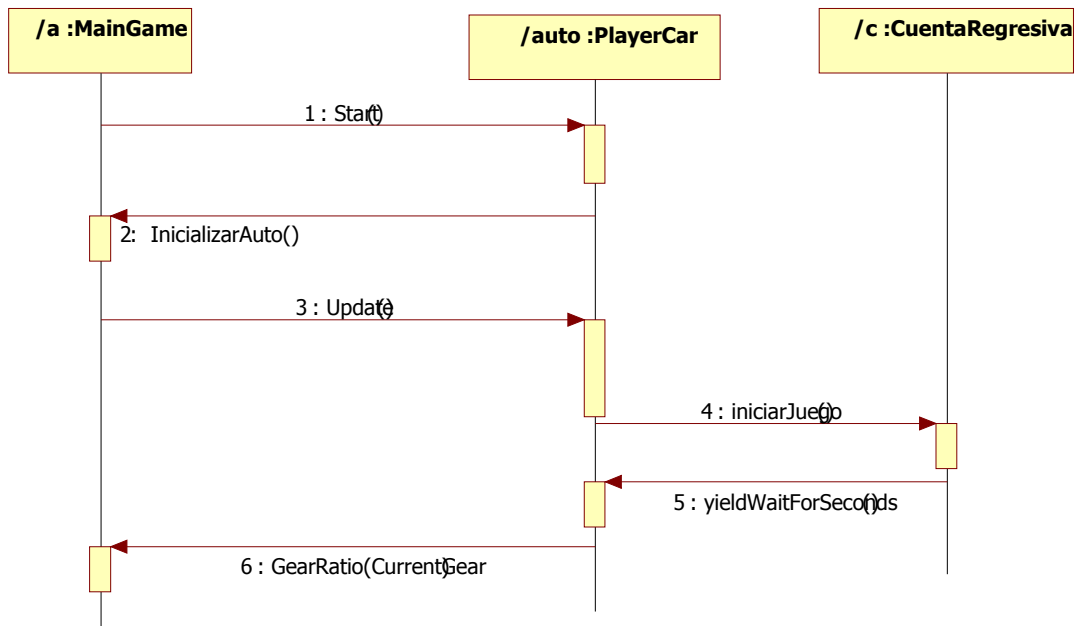


Figura. 25 Diagrama de secuencia 2

DS: CU – 3 ACELERAR AUTO

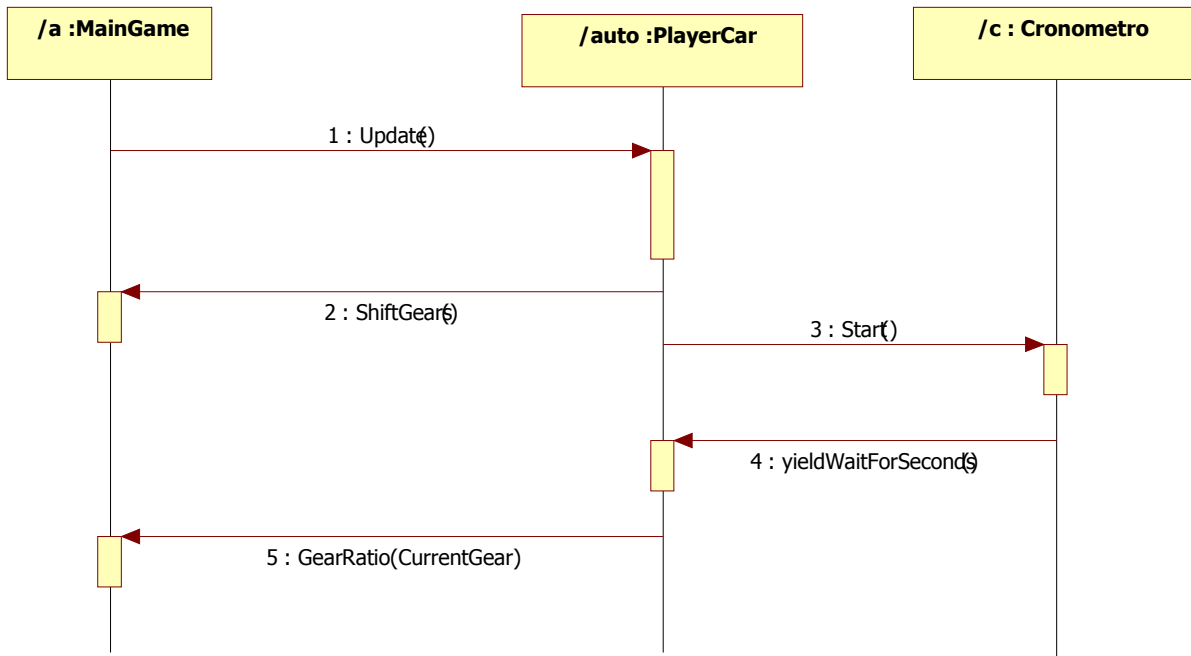


Figura. 26 Diagrama de secuencia 3

DS: CU – 4 ESQUIVAR OBSTACULOS

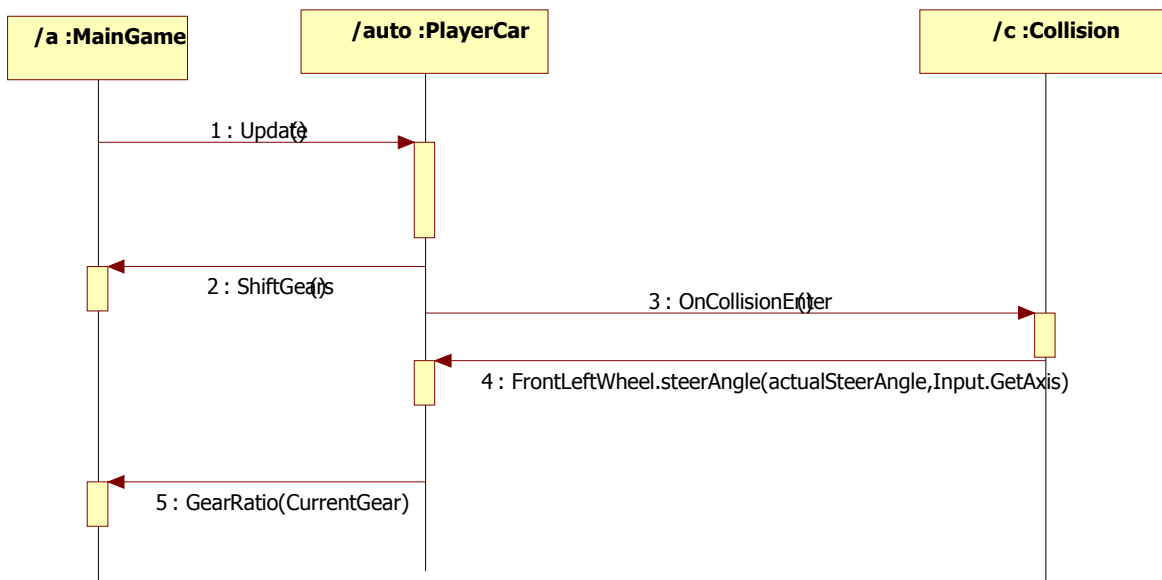


Figura. 27 Diagrama de secuencia 4

DS: CU – 5 PAUSA

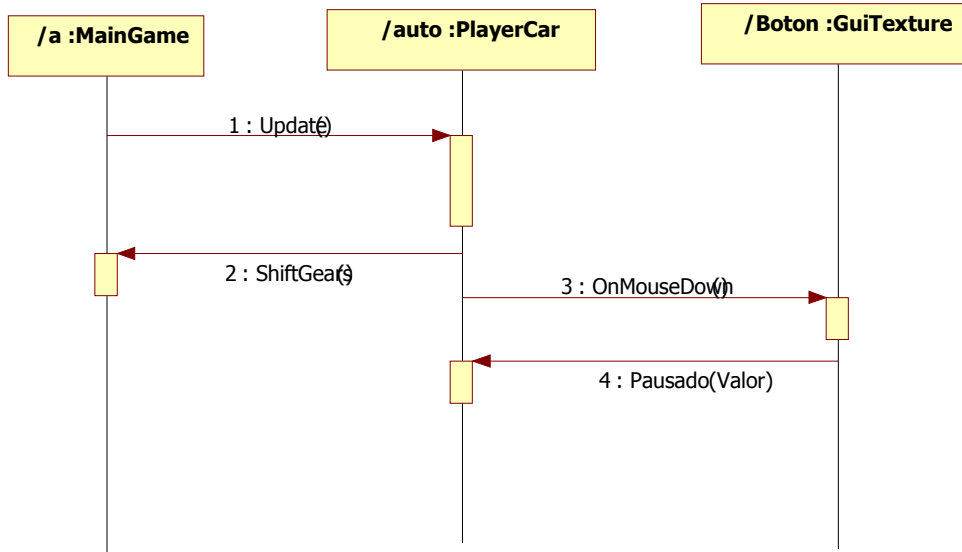


Figura. 28 Diagrama de secuencia 5

DS: CU – 6 REGRESAR AL MENU

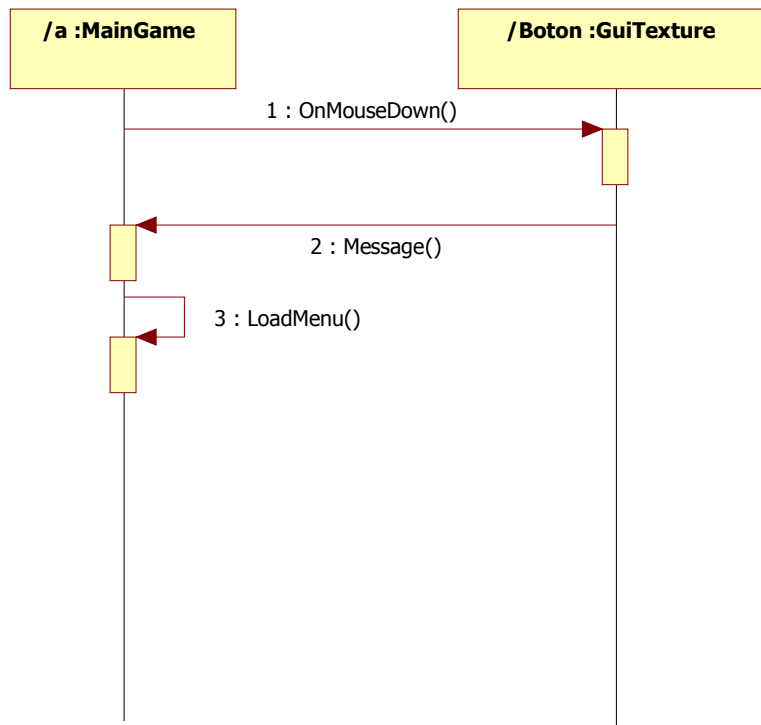


Figura. 29 Diagrama de secuencia 6

8.2.2.7. Cronograma de Trabajo

Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Prede	Nombres de los recursos	abril 2013 31 marzo D L M X J V S
1		Recopilar información para la propuesta	15 días	lun 01/04/13	vie 19/04/13		Jose Luis Alvarez, Marcela	
2		Delimitar el tema de investigación	6 días	mar 23/04/13	mar 30/04/13	1	Jose Luis Alvarez, Marcela	
3		Elaborar Objetivos y justificación	8 días	mié 01/05/13	vie 10/05/13	2	Marcela Chavez	
4		Buscar antecedentes teóricos del tema	15 días	lun 06/05/13	vie 24/05/13		Jose Luis Alvarez	
5		Construir el Marco Teórico y conceptual	47 días	lun 27/05/13	mar 30/07/13	4	Marcela Chavez	
6		Definir el tipo de estudio	3 días	mar 11/06/13	jue 13/06/13		Marcela Chavez	
7		Diseñar y aplicar la metodología de	10 días	vie 14/06/13	jue 27/06/13	6	Marcela Chavez	
8		Determinar la factibilidad del videojuego	1 día	jue 07/11/13	jue 07/11/13		Jose Luis Alvarez	
9		Implementar metodología de desarrollo	66 días	vie 28/06/13	vie 27/09/13	7	Marcela Chavez	
10		Ejecutar la fase de pre – producción	5 días	vie 28/06/13	jue 04/07/13		Marcela Chavez	
11		Ejecutar la fase de producción	61 días	vie 05/07/13	vie 27/09/13	10	Marcela Chavez	
12		Ejecutar la etapa de diseño del juego	10 días	vie 05/07/13	jue 18/07/13		Marcela Chavez	
13		Elaborar el guion del juego	3 días	vie 05/07/13	mar 09/07/13		Marcela Chavez	
14		Elaborar el guion del juego para el	3 días	mié 10/07/13	vie 12/07/13	13	Marcela Chavez	
15		Definir la manera en que	4 días	lun 15/07/13	jue 18/07/13	14	Jose Luis Alvarez	
16		Diseñar diagramas UML	6 días	vie 19/07/13	vie 26/07/13	15	Marcela Chavez	
17		Implementar el videojuego	45 días	lun 29/07/13	vie 27/09/13	16	Jose Luis Alvarez	
18		Realizar primer incremento	15 días	lun 29/07/13	vie 16/08/13		Jose Luis Alvarez, Marcela Ch	
19		Realizar segundo incremento	30 días	lun 19/08/13	vie 27/09/13	18	Jose Luis Alvarez, Marcela	
20		Capturar los escenarios del videojuego	5 días	lun 30/09/13	vie 04/10/13	11	Marcela Chavez	
21		Documentar la aplicación de la tecnica de	4 días	lun 07/10/13	jue 10/10/13	20	Jose Luis Alvarez	
22		Elaborar conclusiones y	2 días	vie 11/10/13	lun 14/10/13	21	Jose Luis Alvarez, Marcela	

Proyecto: cronograma final
 Fecha: vie 08/11/13

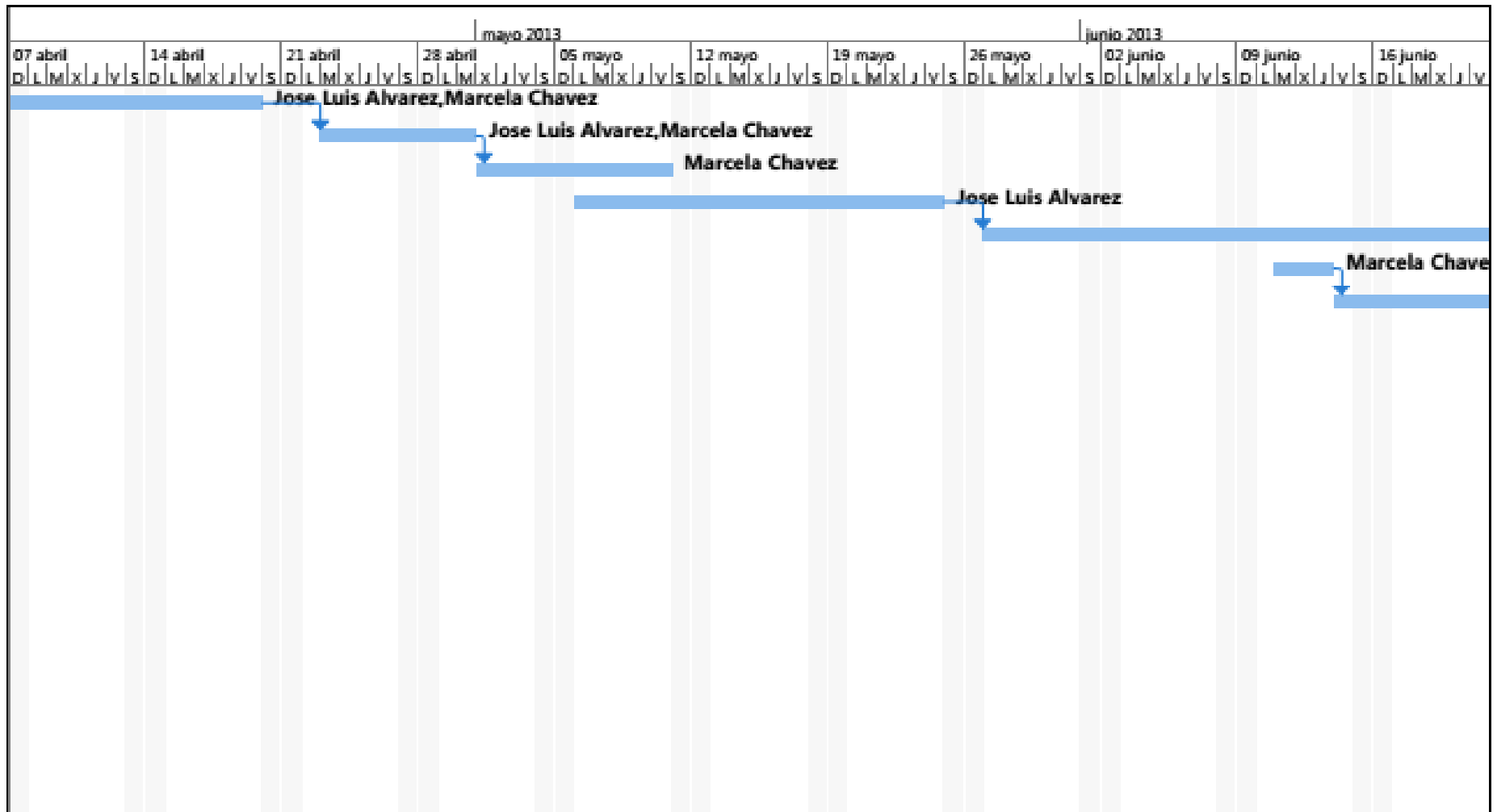
Tarea		Resumen inactivo		Tareas externas	
División		Tarea manual		Hito externo	
Hito		solo duración		Fecha límite	
Resumen		Informe de resumen manual		Progreso	
Resumen del proyecto		Resumen manual		Progreso manual	
Tarea inactiva		solo el comienzo			
Hito inactivo		solo fin			

Nica-Adventure “Descubriendo la Ruta de los Tesoros de Nicaragua”

Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Prede	Nombres de los recursos	abril 2013						
								D	L	M	X	J	V	S
23		Redacción del borrador del documento	5 días	mar 15/10/13	lun 21/10/13	22	Jose Luis Alvarez,Marcela							
24		Revisión y corrección del borrador	4 días	mar 22/10/13	vie 25/10/13	23	Msc. Juan de Dios Bonilla							
25		Presentación del informe	1 día	sáb 26/10/13	sáb 26/10/13	24	Jose Luis Alvarez,Marcela Ch							

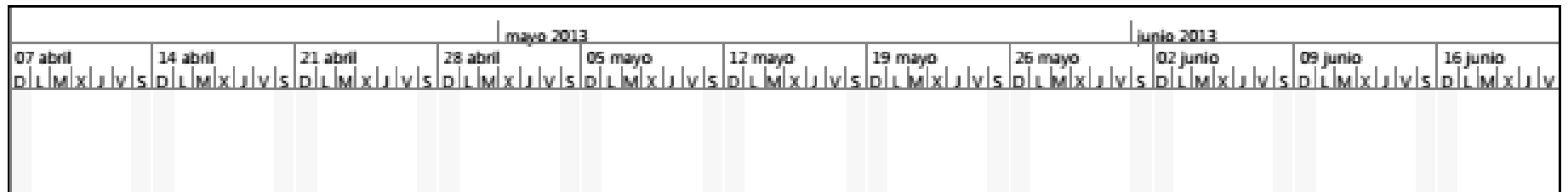
Proyecto: cronograma final Fecha: vie 08/11/13	Tarea		Resumen inactivo		Tareas externas	
	División		Tarea manual		Hito externo	
	Hito		solo duración		Fecha limite	
	Resumen		Informe de resumen manual		Progreso	
	Resumen del proyecto		Resumen manual		Progreso manual	
	Tarea inactiva		solo el comienzo			
	Hito inactivo		solo fin			

Nica-Adventure “Descubriendo la Ruta de los Tesoros de Nicaragua”



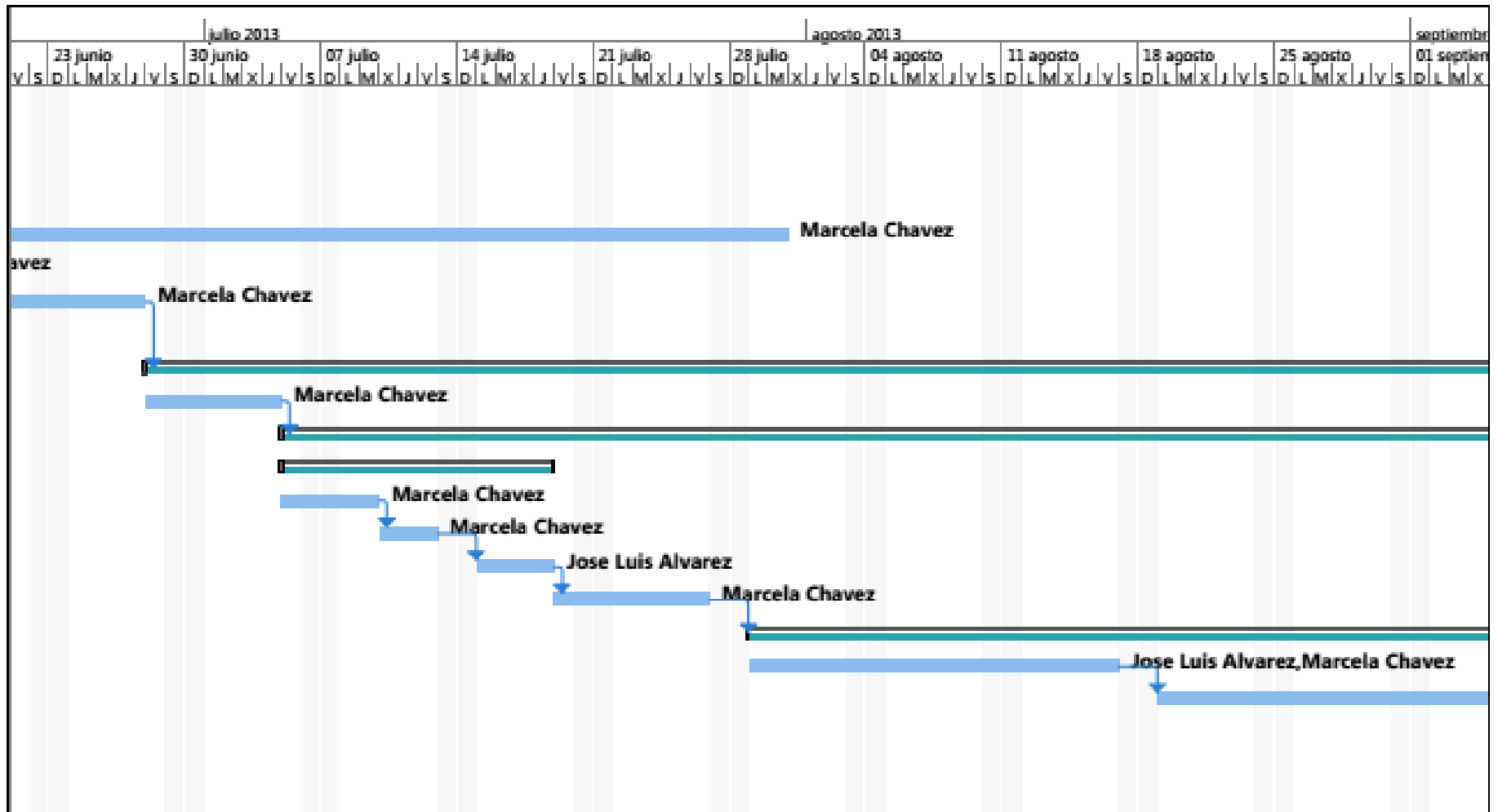
Proyecto: cronograma final Fecha: vie 08/11/13	Tarea		Resumen inactivo		Tareas externas	
	División		Tarea manual		Hito externo	
	Hito		solo duración		Fecha limite	
	Resumen		Informe de resumen manual		Progreso	
	Resumen del proyecto		Resumen manual		Progreso manual	
	Tarea inactiva		solo el comienzo			
	Hito inactivo		solo fin			

Nica-Adventure “Descubriendo la Ruta de los Tesoros de Nicaragua”



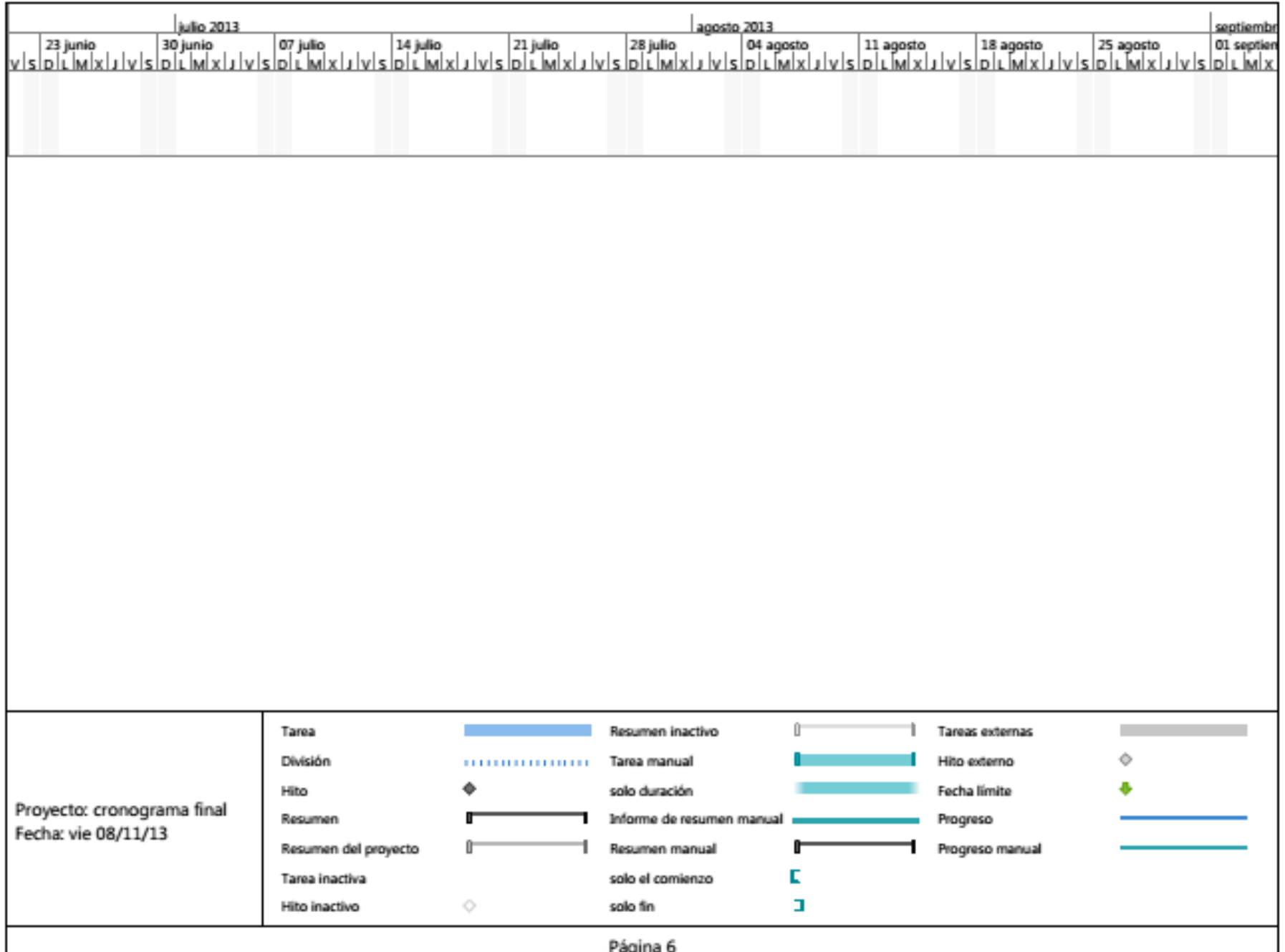
Proyecto: cronograma final Fecha: vie 08/11/13	Tarea		Resumen inactivo		Tareas externas	
	División		Tarea manual		Hito externo	
	Hito		solo duración		Fecha límite	
	Resumen		Informe de resumen manual		Progreso	
	Resumen del proyecto		Resumen manual		Progreso manual	
	Tarea inactiva		solo el comienzo			
	Hito inactivo		solo fin			

Nica-Adventure “Descubriendo la Ruta de los Tesoros de Nicaragua”

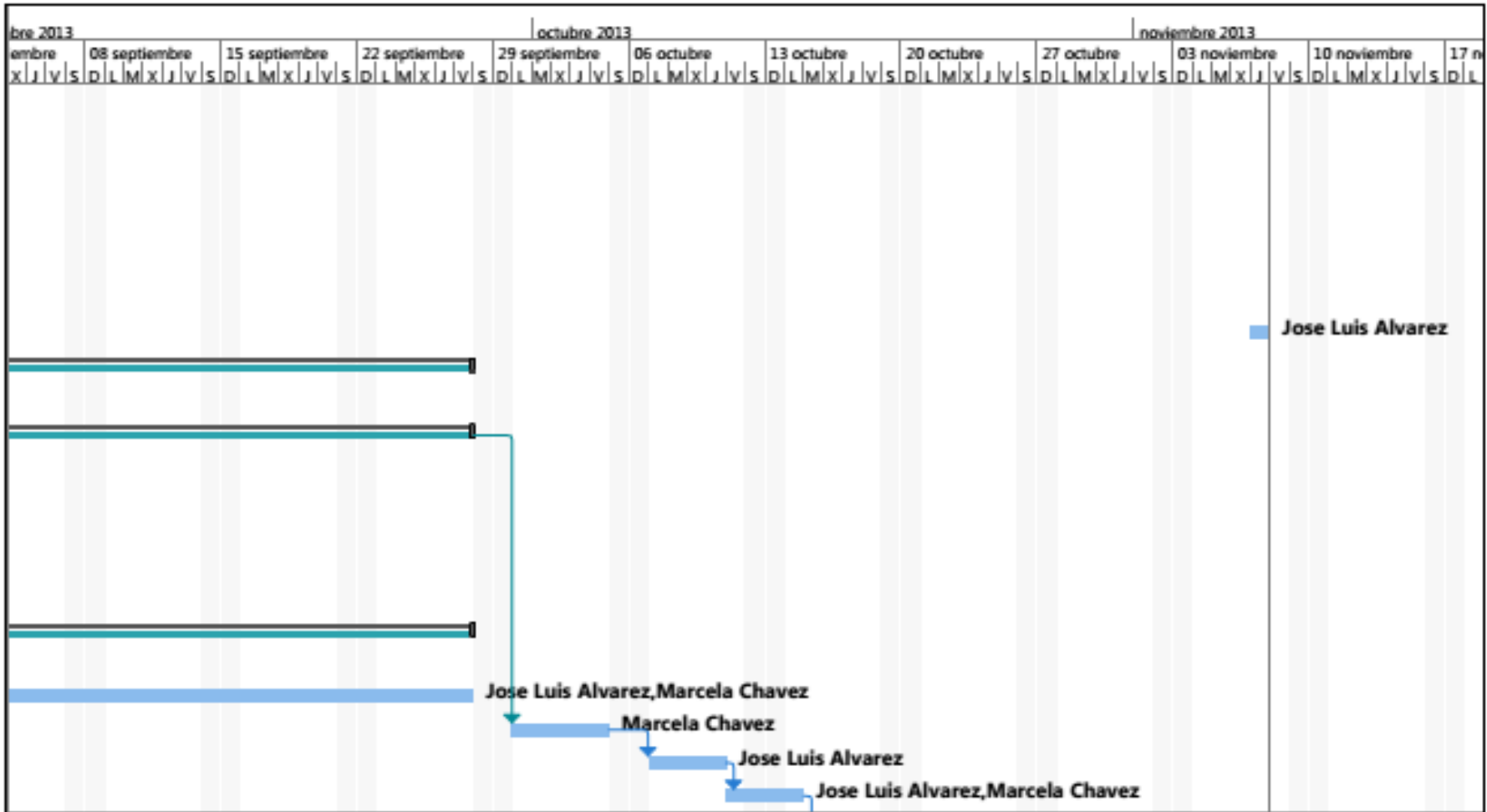


Proyecto: cronograma final Fecha: vie 08/11/13	Tarea		Resumen inactivo		Tareas externas	
	División		Tarea manual		Hito externo	
	Hito		solo duración		Fecha límite	
	Resumen		Informe de resumen manual		Progreso	
	Resumen del proyecto		Resumen manual		Progreso manual	
	Tarea inactiva		solo el comienzo			
	Hito inactivo		solo fin			

Nica-Adventure “Descubriendo la Ruta de los Tesoros de Nicaragua”



Nica-Adventure “Descubriendo la Ruta de los Tesoros de Nicaragua”



Proyecto: cronograma final Fecha: vie 08/11/13	Tarea		Resumen inactivo		Tareas externas	
	División		Tarea manual		Hito externo	
	Hito		solo duración		Fecha límite	
	Resumen		Informe de resumen manual		Progreso	
	Resumen del proyecto		Resumen manual		Progreso manual	
	Tarea inactiva		solo el comienzo			
	Hito inactivo		solo fin			

Nica-Adventure “Descubriendo la Ruta de los Tesoros de Nicaragua”



8.2.3. IMPLEMENTACIÓN

8.2.3.1. Primer Incremento

8.2.3.1.1. Librería jPCT-AE

La implementación del juego fue llevada a cabo con la librería gratuita jPCT-AE, que es una versión para Android de la librería jPCT. jPCT fue creada en base a OpenGL ES, que es una versión para portátiles del popular motor 3D OpenGL. Esta librería fue pensada especialmente para el desarrollo de juegos, con herramientas acorde para el control de la cámara, importación de modelos 3D, detección de colisiones y otros.

jPCT-AE es una versión más compacta de jPCT, esto por las obvias diferencias en procesamiento entre un teléfono móvil y un computador. jPCT-AE excluye algunas clases de jPCT, pero mantiene las esenciales para desarrollar una aplicación 3D.

La librería posee una página web en donde se encuentran las versiones de esta para su descarga, además de toda su documentación, un foro y ejemplos creados por usuarios de jPCT. La página es www.jpct.net.

8.2.3.1.2. Cámara

La implementación de la cámara se llevó a cabo mediante la clase *Camera* de la librería jPCT – AE. Esta clase representa la posición y la dirección del observador o cámara en la escena actual. Esta también contiene información sobre el campo de vista actual, en inglés **FOV Field Of View**. Un aspecto relevante de esta clase es su matriz de rotación la cual se aplica a todos los elementos del mundo. Esto es importante cuando se escoge el ángulo de rotación de la cámara: una rotación (virtual) con un ángulo w en un determinado eje sobre la cámara, es lo mismo que rotar el mundo alrededor del eje con un ángulo w .

El constructor de la clase *Camera* es el siguiente: **public Camera()**

Este constructor crea una nueva *Camera* con una vista por defecto: dirección a lo largo del eje z , posición en el origen y un FOV de 1.25 unidades.

El juego contiene una cámara, representada por la siguiente constante:

public static final int *MODE_NORMAL* = 0;

El **modo normal** es utilizado en todo el juego. Este posee una cámara situada atrás del personaje. La cámara dentro del juego se crea mediante el siguiente código:

```
//añadir cámara
1. cam = world.getCamera();
2. var moveCam: vector3 = vector3(0,0,0);
3. cam.moveCamera(moveCam, 0);
4. cam.moveCamera(Camera.CAMERA_MOVEOUT, Constants.CAM_DIST);
```

La primera línea instancia la cámara obteniendo la cámara actual que contiene el mundo. La segunda línea utiliza la clase SimpleVector de la librería jPCT-AE, la cual representa un vector tridimensional básico con coordenadas x,y y z. Este vector es utilizado para mover la cámara al origen. En la línea tres se mueve la cámara utilizando el método moveCamera, el cual traslada la cámara según la dirección dada por el vector y las unidades indicadas, en este caso cero. La cuarta línea mueve la cámara utilizando el modo CAMERA_MOVEOUT el cual mueve la cámara hacia a tras respecto a la posición actual según la constante CAM_DIST que indica un movimiento de 15 unidades.

8.2.3.1.3. Escenario

El escenario en el código es representado por el objeto *world*, el cual es una instancia de la clase *World* de la librería jPCT-AE. Esta clase es la más importante dentro de esta librería orientada a juegos, ya que contiene todos los elementos del mundo, luces, modelos, cámaras, etc.

El constructor de esta clase es el siguiente:

public World() Este constructor crea un nuevo mundo, como una sala vacía que debe ser llenada con objetos y luces.

Dentro de los métodos utilizados se encuentran los siguientes:

void addObject(Object3D obj) Añade un objeto a la colección de objetos del mundo.

Parámetros: *obj*, representa al objeto a añadir al mundo

Camera getCamera() Obtiene la cámara actual del mundo

void setCameraTo(Camera cam) Cambia la cámara actual del mundo por la cámara dada

void buildAllObjects() Llama la función *build()* para cada objeto del mundo:

8.2.3.1.4. Modelado

El juego se desarrolla por completo en las distintas carreteras que están dentro de la ruta de los tesoros, debido a esto el diseño del escenario se centró en el modelado de las carreteras y algunos paisajes característicos de los departamentos dentro de la ruta descrita.

Para el modelado de las carreteras y casas se utilizó el software de diseño 3D, 3Ds MAX 2014 (ver Figura 26) y como motor de desarrollo del videojuego UNITY 3D 4.0 (ver Figura 21).



Figura. 30 Portada del juego.

8.2.3.1.5. Personajes

- Programación de personajes

En el código del video juego cada personaje y objeto es representado por la una clase que extiende de Object3D.

La clase `Object3D` es usada para representar objetos en tres dimensiones. Estos objetos son usualmente añadidos a una instancia de `World` para ser renderizados. Un objeto de esta clase sólo puede ser añadido a una instancia de `World` a la vez. Estos objetos pueden ser definidos como hijos o padres, para así obtener un comportamiento jerárquico. La clase `Object3D` posee cuatro constructores, donde el utilizado en este proyecto es el siguiente:

Public `Object3D(Object3D obj)`

Este constructor trabaja de forma similar al método `cloneObject()` pero permite extender `Object3D` y continuar usando los métodos estáticos para cargar o usar primitivas utilizando la sentencia `super(Object3D)` en el constructor de la clase. Parámetros: `obj`, el objeto desde el cual se construye el objeto.

Algunos métodos usados de esta clase son:

`void animate(float index)`

Calcula una nueva forma para este objeto basada en los frames de la animación en la secuencia dada, donde `index` es un valor entre cero y uno.

`void translate(SimpleVector trans)`

Mueve el objeto en el mundo modificando la matriz de translación del objeto 3D. Parámetros: `trans`, vector de translación.

Para animar cada personaje se utilizó un hilo (`Thread`), el cual se encarga de revisar cada determinado tiempo el estado del personaje, para así actualizar su animación. Por otro lado, estos hilos contienen sus propias variables de estado, que se sincronizan con los estados del juego, para así poder terminarlos cuando el juego lo requiera. Como se muestra en el siguiente código de ejemplo, mientras el hilo tenga el estado `RUNNING`,

este seguirá actualizando el estado de animación y gravedad de la cámara que se explicará en el segundo incremento.

➤ Modelado de personajes

Para el diseño 3D del personaje se utilizó el software BLENDER VERSION 2.66, para modificar textura se utilizó PHOTOSHOP CS6 y para la animación UNITY 3D.

Nicaboy: Se pensó en un auto en manos de un Nica intrépido que gusta de viajar y conocer los diversos destinos turísticos que ofrece Nicaragua.

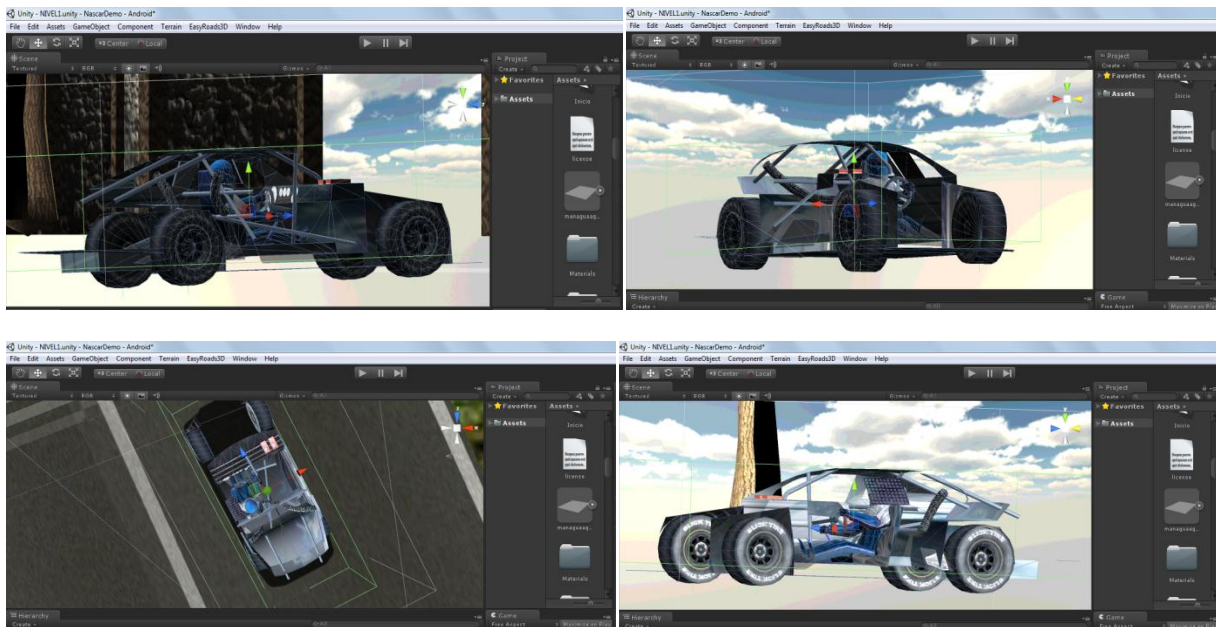


Figura. 31 Vistas frontal, lateral, superior de Nicaboy sin textura

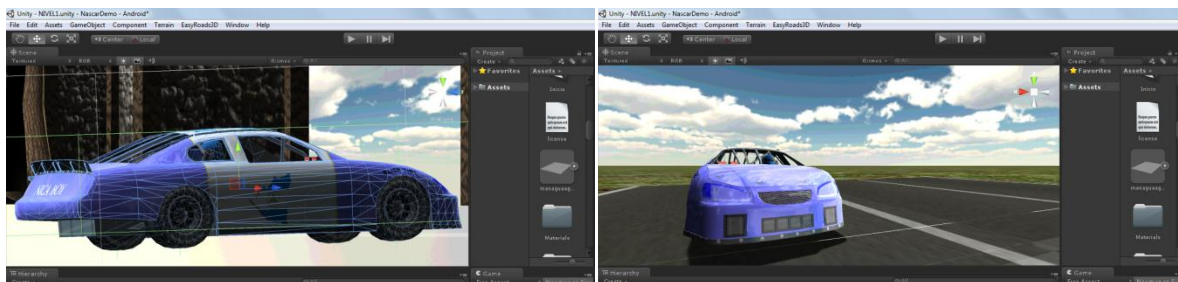


Figura. 32 Vistas frontal y lateral de Nicaboy con textura

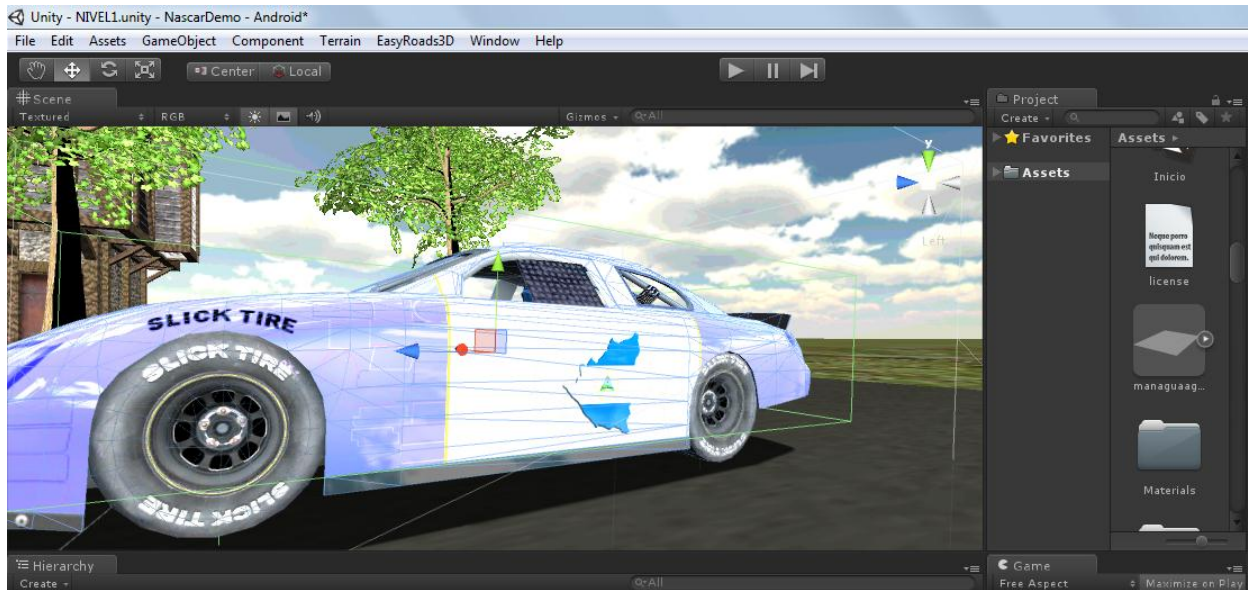


Figura. 33 Vistas en perspectiva de Nicaboy.

8.2.3.1.6. Objetos

TERRENO:

Este objeto es el plano de tierra en el cual está ubicada la carretera y las montañas que se ven el juego, sobre el terreno se aplica el objeto DIRECTIONAL_LIGHT.

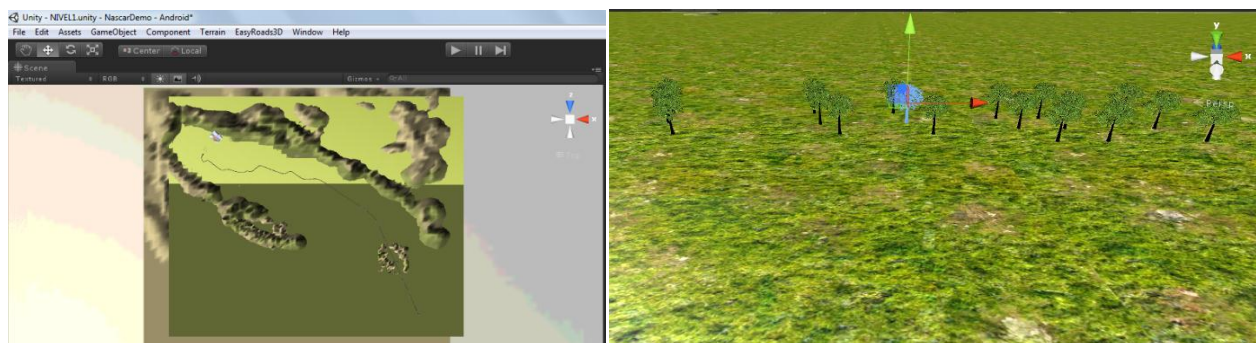


Figura. 34 Vistas del terreno.

CARRETERA: Este objeto representa el camino que debe recorrer Nicaboy para llegar a la meta y así poder ser certificado como visitante, el modelado de este objeto se corresponde con la carretera real de Managua a granada.

Nica-Adventure “Descubriendo la Ruta de los Tesoros de Nicaragua”

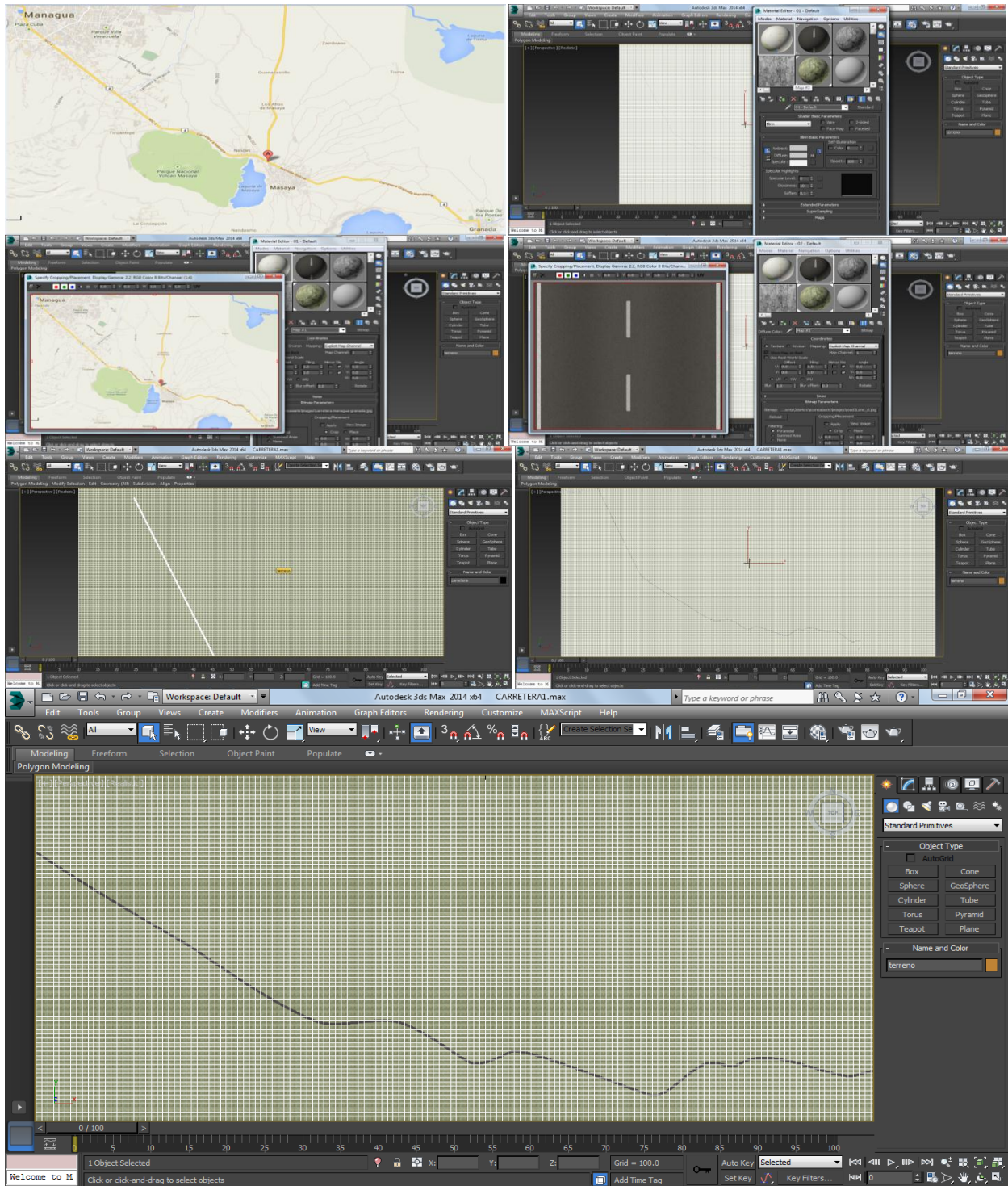


Figura. 35 Modelado de la carretera Managua-Granada.

Nica-Adventure “Descubriendo la Ruta de los Tesoros de Nicaragua”

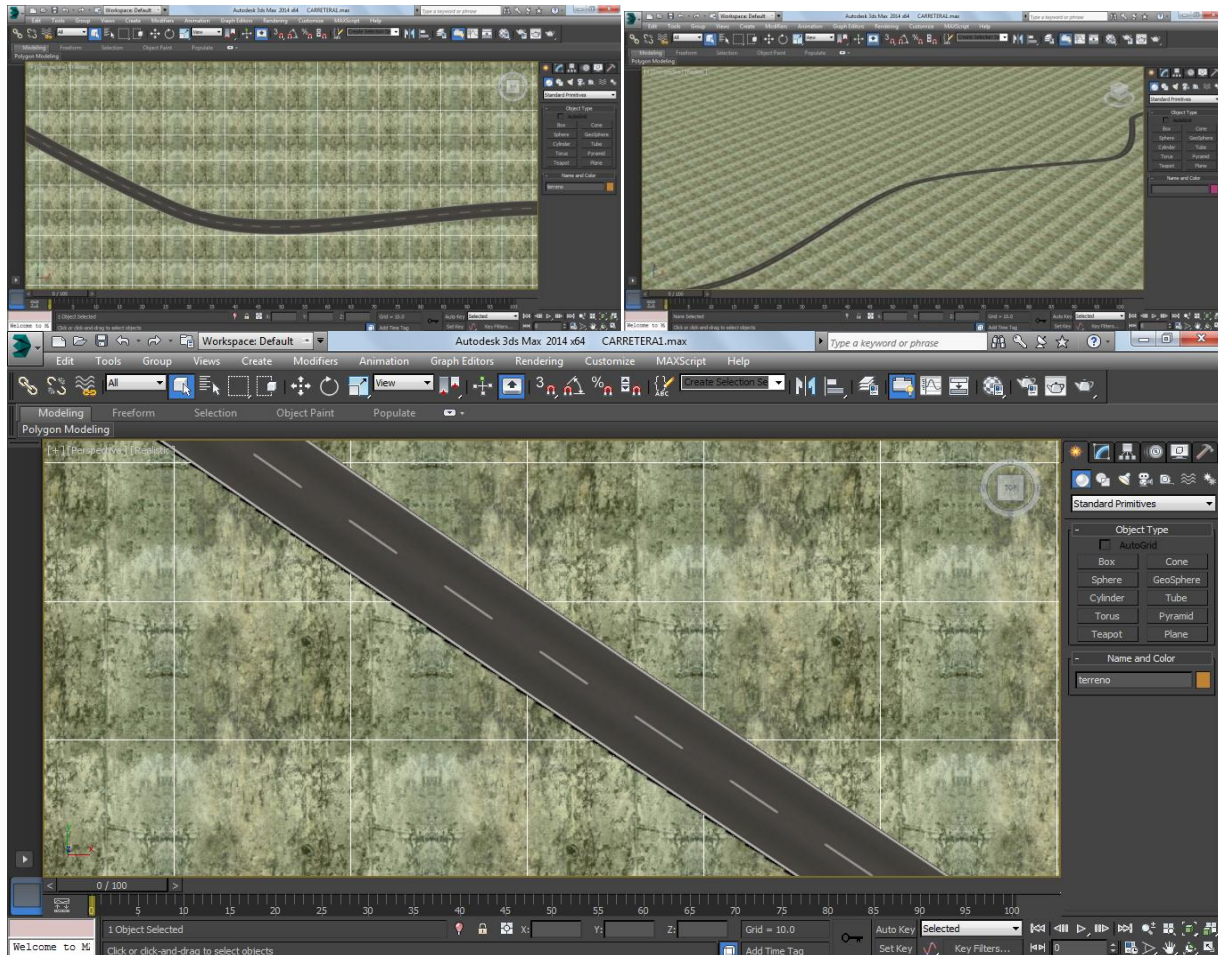


Figura. 36 Vistas en perspectiva de la carretera.

TEMPORIZADOR:

Representa el tiempo que viene disminuyendo y el cual si llega a 0 hace terminar el juego, el tiempo se recarga en cada checkpoint

CHECKPOINT:

Son los puntos de control por los cuales debe pasar nuestro personaje para obtener más tiempo y seguir en la carrera.

8.2.3.2. Segundo Incremento

8.2.3.2.1. Cámara

Siempre utilizando la librería jPCT – AE se realizó la implementación de la cámara ubicándola en la parte trasera del carro de modo que durante todo el recorrido se tendrá una vista de avanzada.

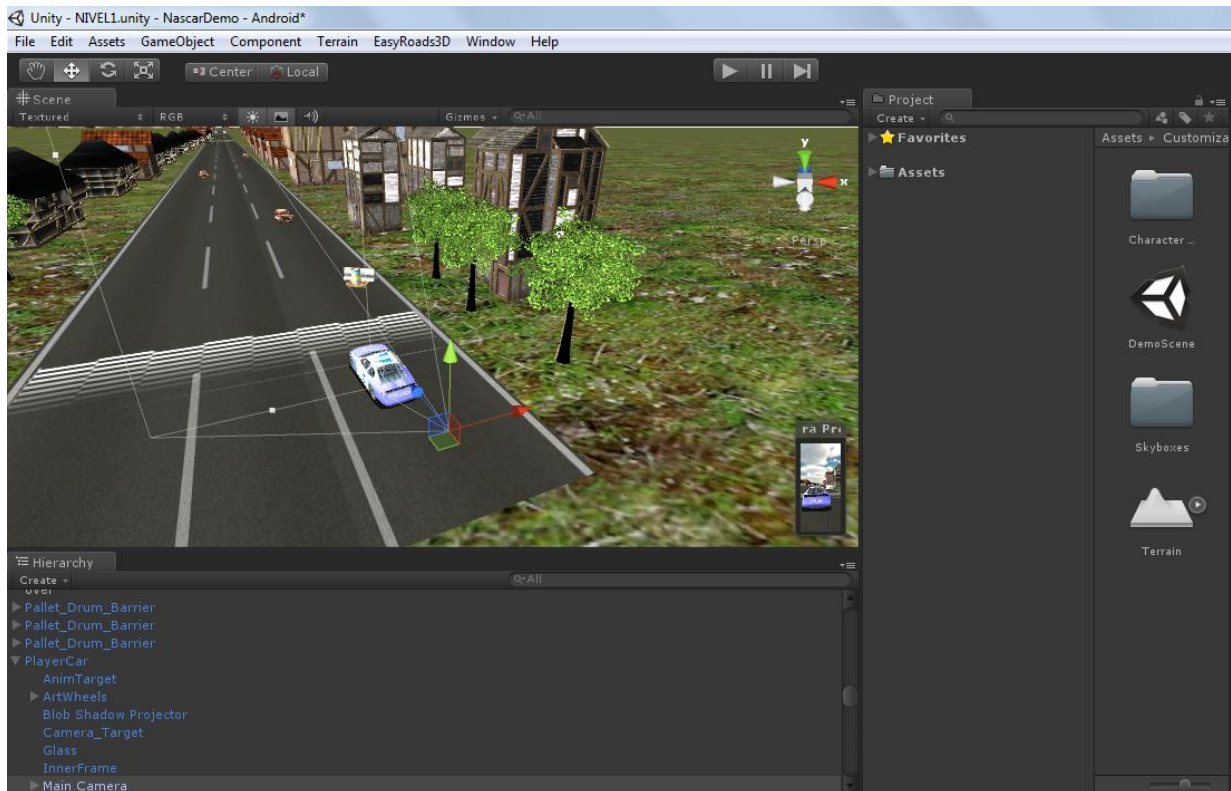


Figura. 37 Cámara en la parte trasera del personaje.

8.2.3.2.2. Objetos

ARBOL:

Los objetos de esta clase representan a los arboles de la vida real, plasmados en este mundo virtual de tal forma que den vistosidad a nuestro escenario.

Nica-Adventure “Descubriendo la Ruta de los Tesoros de Nicaragua”

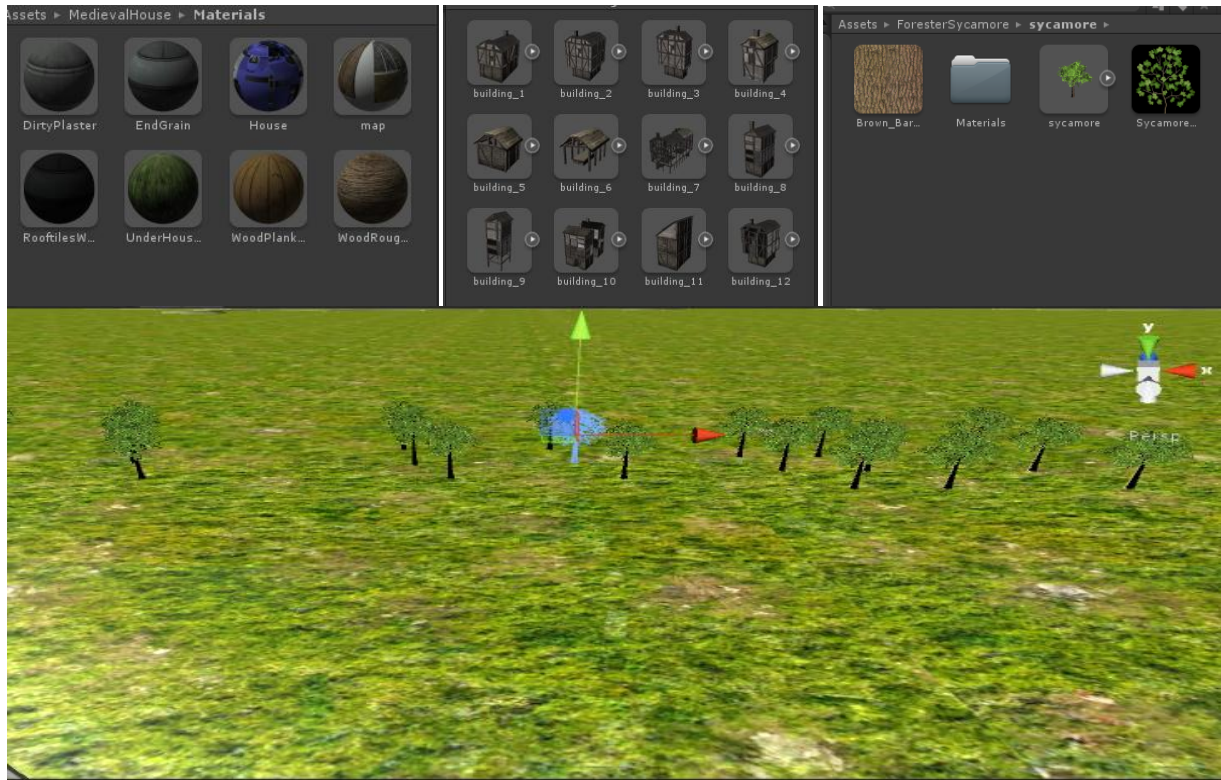


Figura. 38 Objeto árbol y los materiales aplicados.

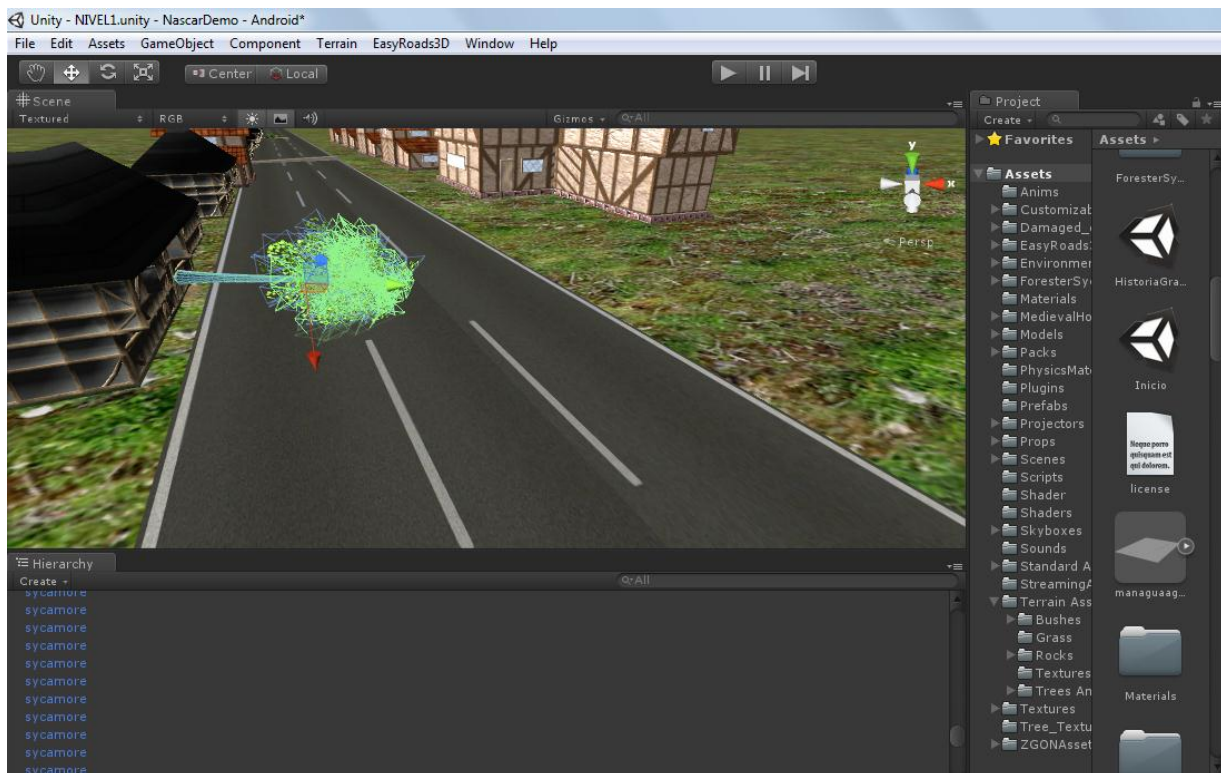


Figura. 39 Objeto árbol caído una variante de la clase árbol utilizado como obstáculo.

CHECKPOINT: Son los puntos de control por los cuales debe pasar nuestro personaje para obtener más tiempo y seguir en la carrera.

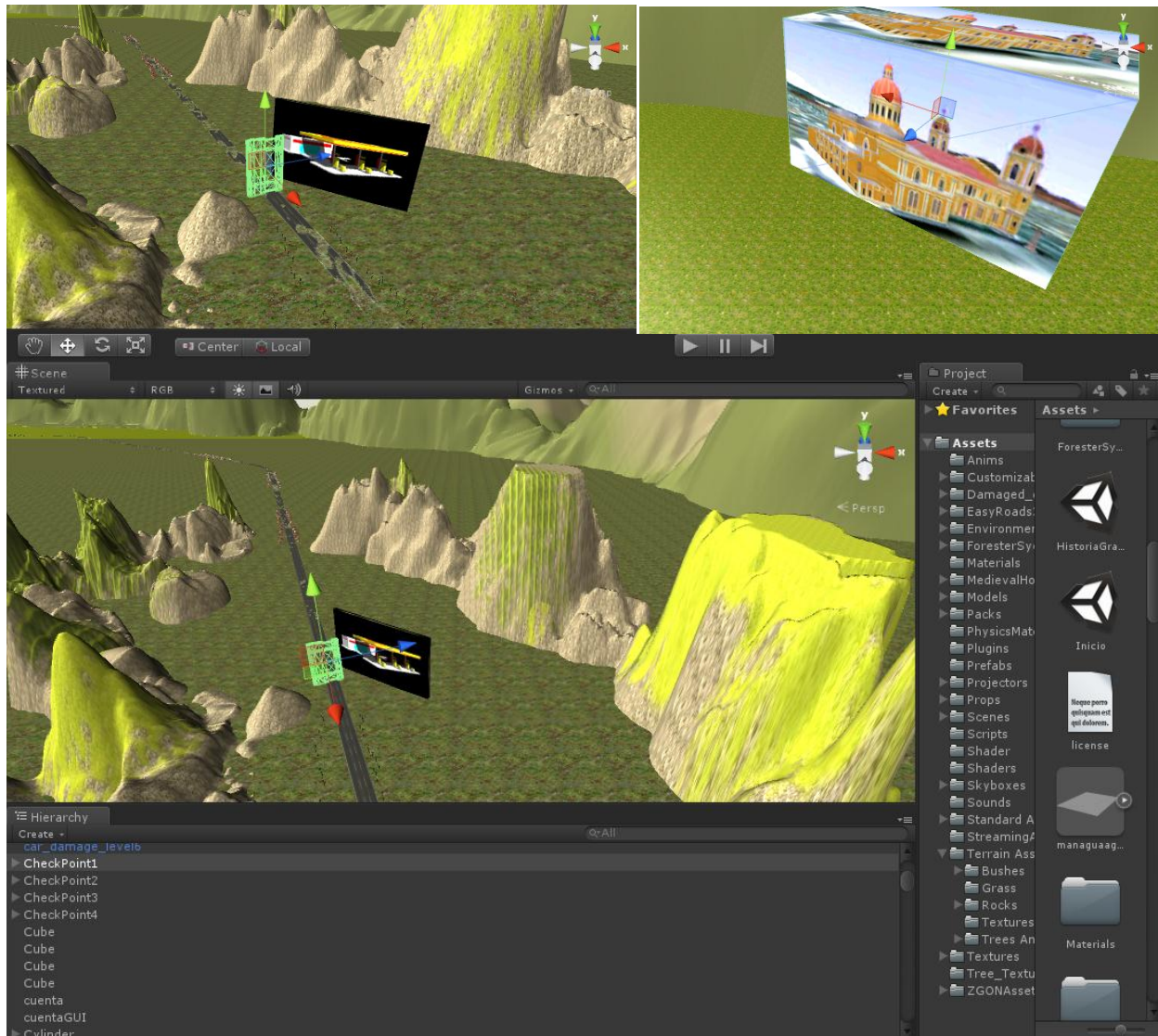


Figura. 40 Objeto Checkpoint en dos distintas capturas.

OBSTACULO_BARRIL: Este objeto se utiliza como obstáculo y se le aplico una colisión que consiste en hacer rebotar al vehículo en movimiento y así atrasar su avance.

Nica-Adventure “Descubriendo la Ruta de los Tesoros de Nicaragua”

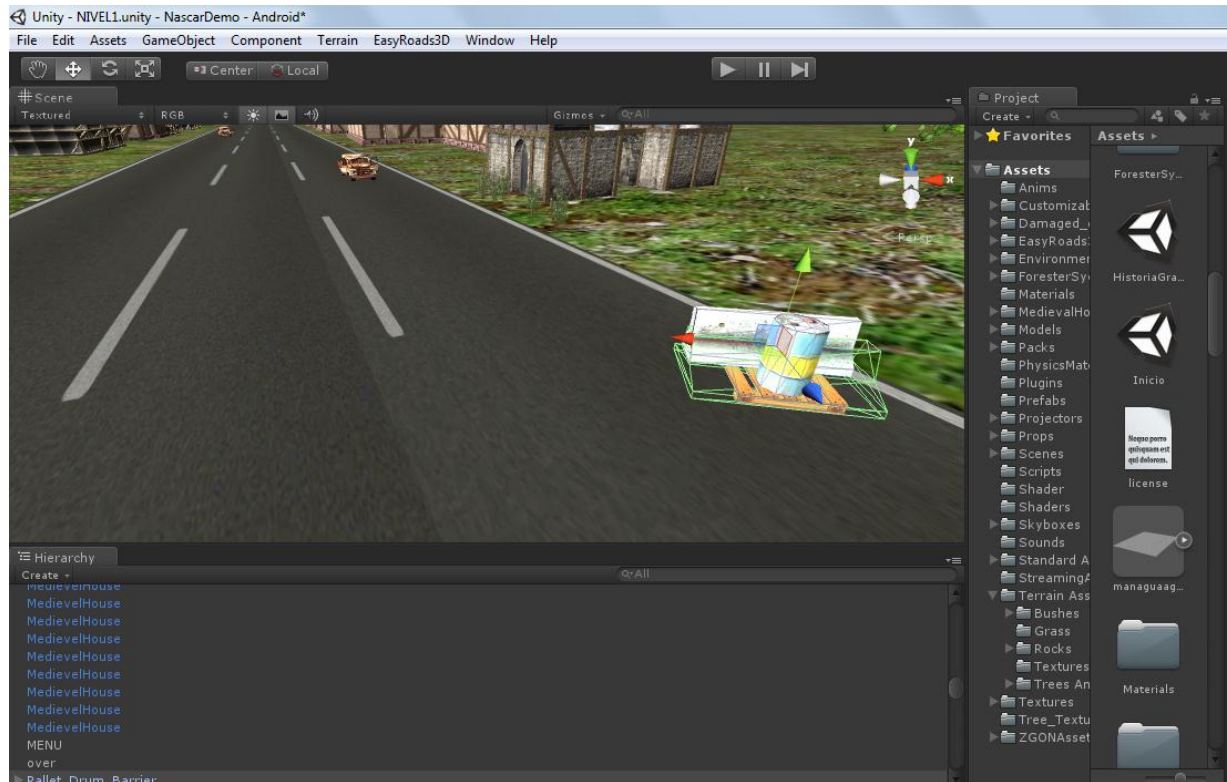


Figura. 41 Objeto Checkpoint en dos distintas capturas

CASAS: Este objeto es una instancia de la clase CASAS y hay dos tipos: casas de 1 y 2 pisos.



Figura. 42 Objeto casas ubicado sobre el terreno con texturas.

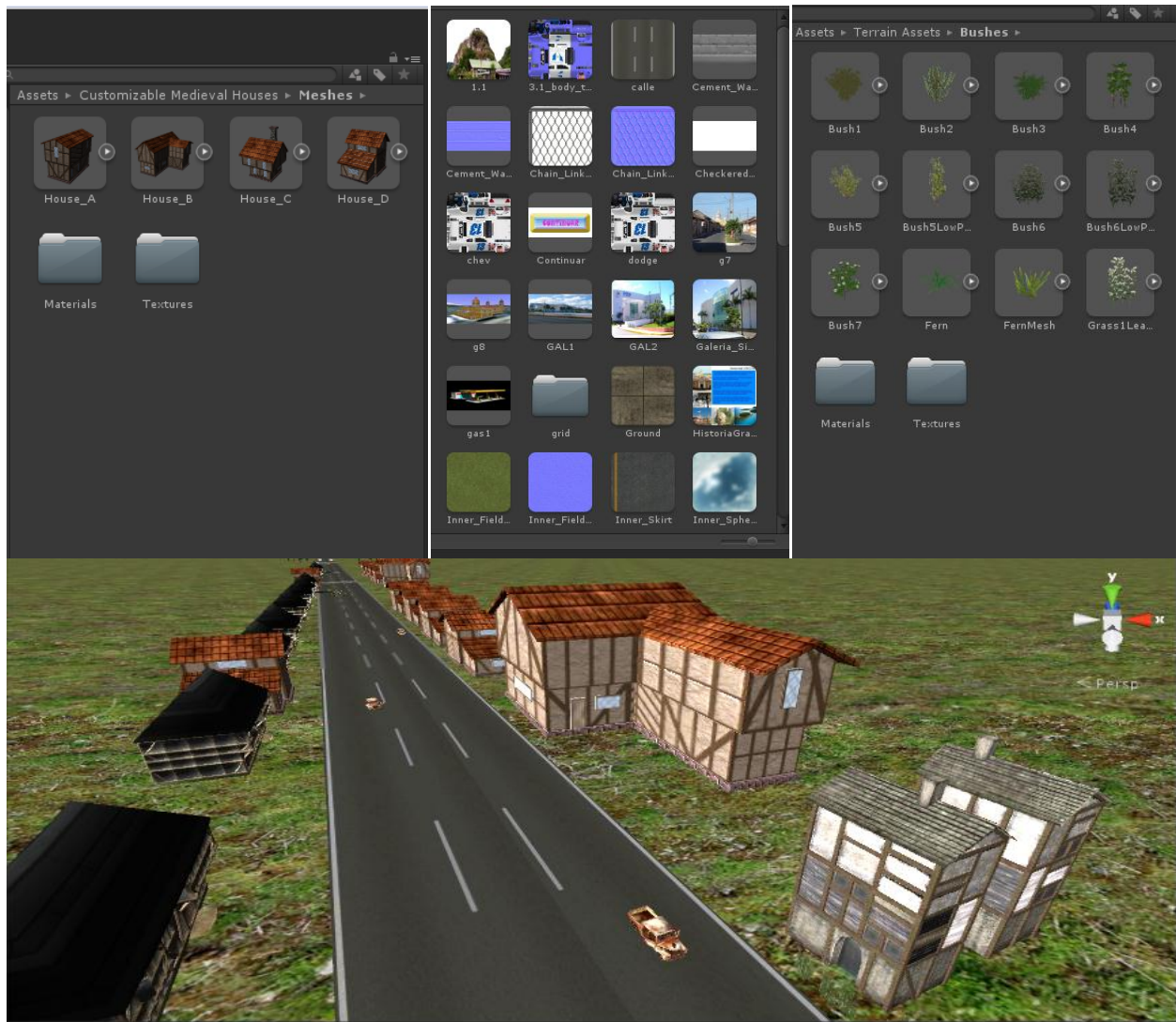


Figura. 43 Texturas disponibles para el objeto casa y sus dos variantes.

DIRECTIONAL_LIGHT: Objeto ubicado sobre el escenario para iluminar todo el ambiente gráfico.

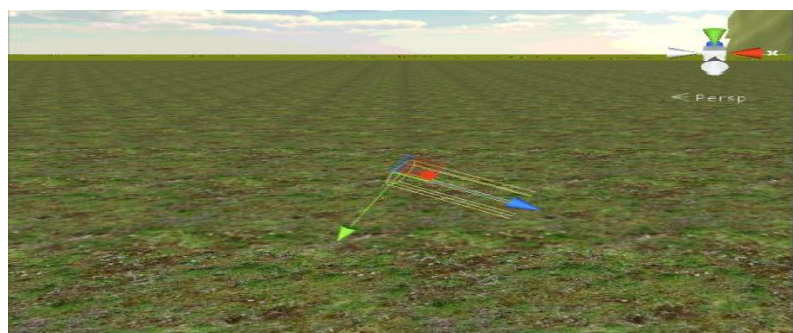


Figura. 44 Texturas disponibles para el objeto casa y sus dos variantes.

REDUCTOR_VELOCIDAD: Este objeto representa un obstáculo para el jugador ya que obliga al jugador a desarrollar destreza al manejar porque de no ser así pierde tiempo y no logra llegar a la meta.

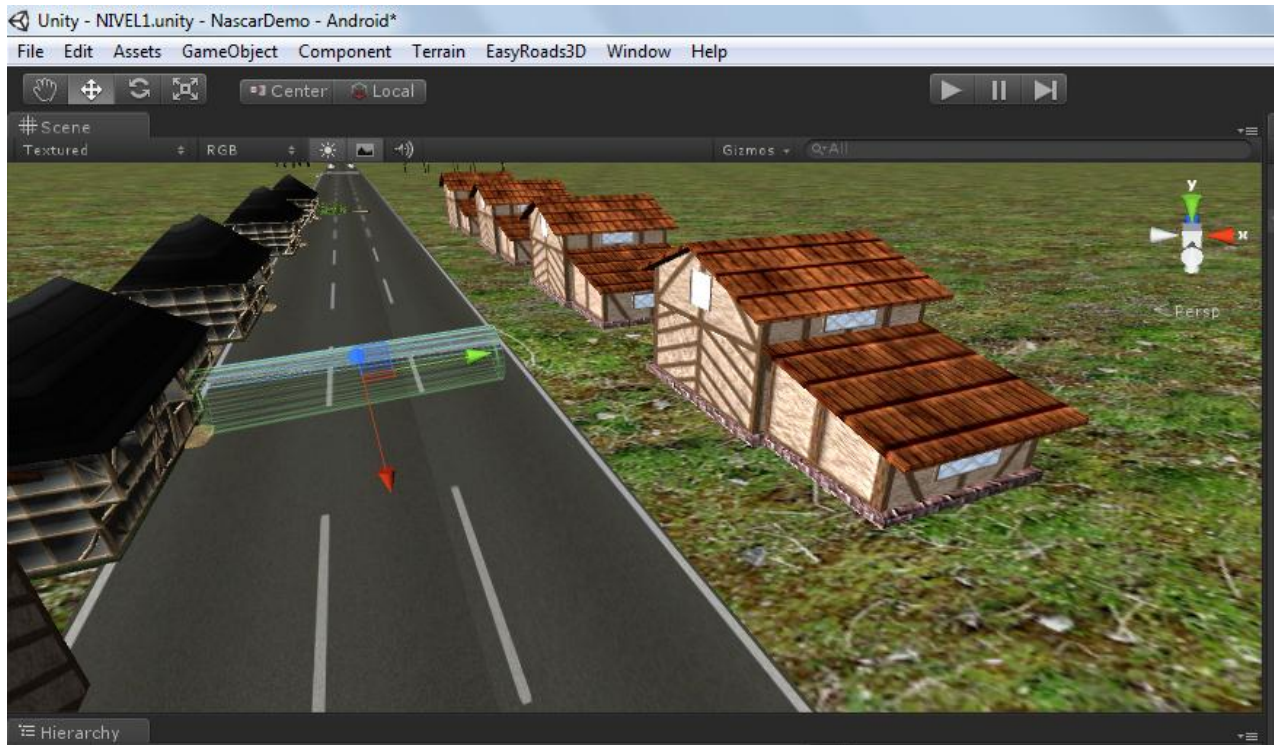


Figura. 45 Objeto reductor de velocidad y su modelado.

S_CARPICKUP:

Este objeto es una camioneta en ruinas, la cual está ubicada en la carretera y sirve de obstáculo



Figura. 46 Objeto camioneta descompuesta y su modelado.

8.2.3.2.3. Texturas

Para la generación de texturas se utilizó el software Adobe Photoshop CS6, cuidando por sobre todo el tamaño (ancho y alto deben ser potencias de dos) y peso de la imagen, ya que imágenes muy pesadas pueden llevar a sobrepasar la capacidad de la máquina virtual de Android (16mb aproximadamente).

En cuanto a la asignación texturas a modelos complejos como por ejemplo la carretera, se utilizó el software 3Ds Max, el cual provee una forma sencilla y flexible de administrar la posición y forma en que las texturas se despliegan sobre los modelos (coordenadas).

8.2.3.2.4. Iluminación

Para la iluminación del escenario se utilizó el objeto `DIRECTIONAL_LIGHT` que pertenece a la clase `GameObject` y tiene propiedades como el tipo de iluminación, color, intensidad de la luz, sombra, este es aplicado al escenario para lograr una mejor visión del ambiente gráfico.

8.2.3.2.5. Gravedad

La gravedad es fundamental en juego; sin esta no tenemos sensación de peso y por lo tanto poco realismo. Debido a lo anteriormente mencionado es crucial implementar algún mecanismo que simule esta propiedad.

Para implementar la gravedad se utilizó el siguiente método:

```
public void gravity(){
    if(state != Constants.JUMP){
        SimpleVector gravityVector = checkForCollisionEllipsoid( new SimpleVector(0,
        1, 0), Constants.ELLIPSOID, Constants.RECURSION);
        translate(gravityVector);
    }
}
```

8.2.3.2.6. Colisiones

Las colisiones son fundamentales en cualquier juego, ya que estas nos permiten interactuar entre objetos, simulando diversas acciones entre ellos, por ejemplo el típico choque en un juego de autos o dar el pie para que ocurra otra acción, como abrir una puerta.

Dentro de la librería utilizada existe un mecanismo muy intuitivo y a la vez eficaz de implementar colisiones; existe una propiedad dada a los *Object3D* que los clasifica según tres categorías:

static int COLLISION_CHECK_NONE: No se realiza ninguna clase de colisión con este objeto

static int COLLISION_CHECK_OTHERS: Otros objetos pueden colisionar con este objeto

static int COLLISION_CHECK_SELF: Este objeto puede colisionar con otros objetos

Así por ejemplo el protagonista es clasificado como un objeto que puede colisionar con otros y el castillo como un objeto que ocasiona que colisione otro.

8.2.3.2.7. Sonido

El sonido en el software fue implementado mediante las clases *AudioManager* y *SoundPool*, proporcionadas por el sistema operativo Android. La primera clase *AudioManager* provee un mecanismo para controlar el volumen y los sonidos. La segunda *SoundPool* es una colección de samples que pueden ser cargados en memoria desde un recurso de la aplicación o desde un archivo en el sistema. *SoundPool* usa el servicio *MediaPlayer* para decodificar el audio en un formato 16 bit PCM mono o estéreo. Esto permite a la aplicación pasar audios comprimidos sin tener que bajar el rendimiento al descomprimir y reproducir al mismo tiempo.

8.3. RESULTADO 2: ESCENARIOS DINÁMICOS E INTERACTIVOS CON EL USUARIO.

Como punto de partida para elaborar el diseño de los escenarios utilizamos las pautas que nos da nuestro Guion del juego.

Ya que nuestra historia es un recorrido a contra reloj de la ruta de los tesoros, nuestro personaje principal “NicaBoy”, tiene aceleración, retroceso, giros y una determinado efecto en respuesta a una colisión. Por ejemplo: al atravesar un reductor de velocidad el auto es impulsado hacia arriba y en dependencia de la velocidad que este lleve puede perder el control del auto o incluso volcarse a como sucede en la realidad, ya que la idea es que el jugador se sienta como que está manejando en el mundo real.

Ajustándonos a la historia al final del recorrido al jugador se le da a conocer acerca de la cultura e historia de la ciudad a la que está llegando y se le extiende un sello de visitante, de modo que el jugador se sienta un viajero real en la travesía del recorrido de la ruta de los tesoros y así promover nuestra cultura y atractivos a través del turismo Nicaragüense. A continuación se muestran algunos de los escenarios modelados y operativos dentro del videojuego:



Figura. 47 NicaBoy en cuenta Regresiva antes de iniciar la carrera.

Nica-Adventure “Descubriendo la Ruta de los Tesoros de Nicaragua”



Figura. 48 Modelado del ambiente ubicando objetos casas para causar un efecto inmersivo en el jugador.



Figura. 49 NicaBoy en Marcha con velocímetro en pantalla.

Nica-Adventure “Descubriendo la Ruta de los Tesoros de Nicaragua”

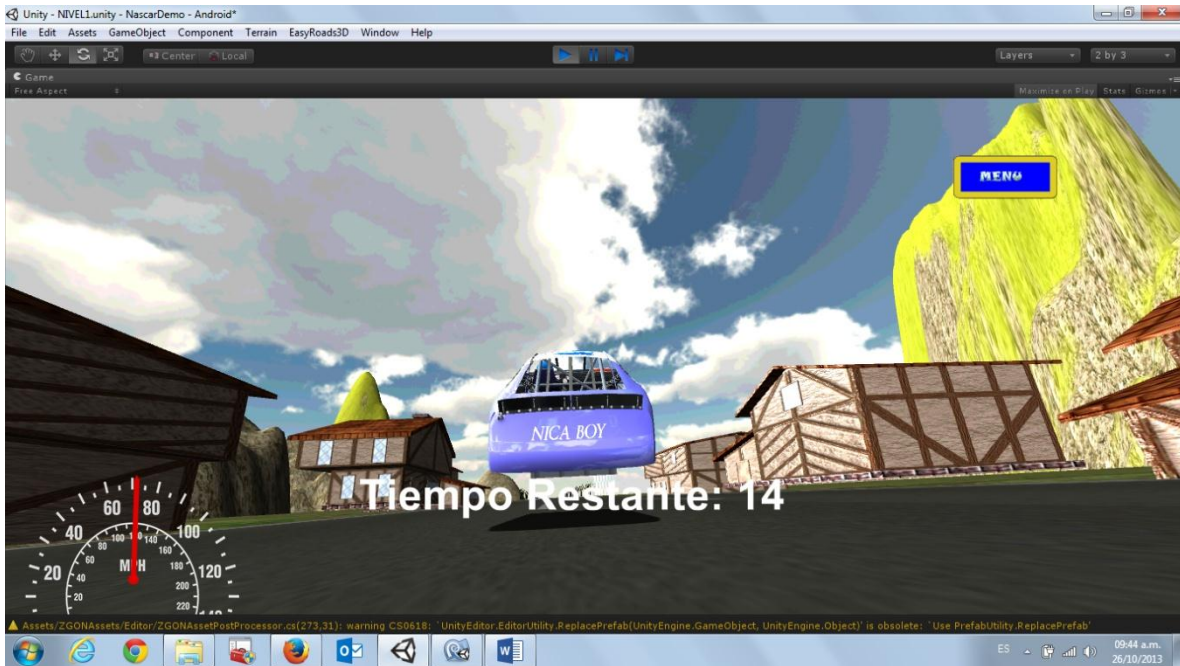


Figura. 50 NicaBoy atravesando un reductor de velocidad.



Figura. 51 NicaBoy acabando de pasar por un CHECKPOINT.

Nica-Adventure “Descubriendo la Ruta de los Tesoros de Nicaragua”



Figura. 52 NicaBoy perdido en la montaña navegando el ambiente.



Figura. 53 NicaBoy corriendo en dos ruedas luego de impactar un obstáculo.

8.4. RESULTADO 3: TÉCNICA DE PROGRAMACIÓN UTILIZADA.

A cómo podemos observar dentro de la sección 10.2 que es el desarrollo del videojuego mediante dos incrementos, se ha aplicado la programación orientada a objetos ya que nos permite manipular de manera sencilla ordenada y optima los objetos 3D.

Las ventajas de usar programación orientada a objetos, son varias y muy importantes a la hora de programar códigos de gran tamaño permitiendo ahorrar tiempo y como consecuencia dinero. La herencia de clases, las funciones virtuales y paso de mensajes son otras de las características de este modelo que hizo posible integrar todos los objetos presentes en el videojuego logrando la interacción entre estos.

A continuación se presentan capturas de las clases programadas:

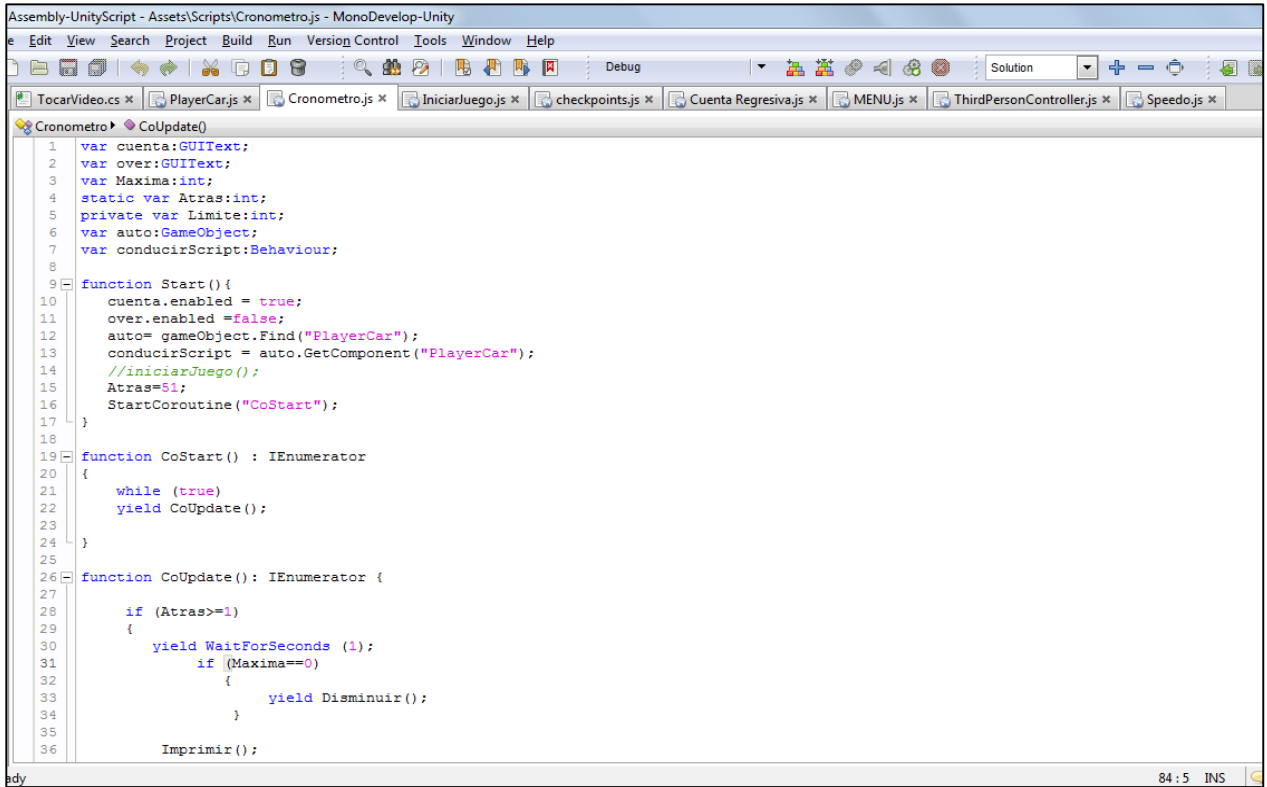
```

10 var highSpeedSteer = 10.0;
11 var lowSpeedSteer = 16.0;
12 var highSpeedLevel = 30;
13 var animatedTarget : Transform;
14
15 // These variables are for the gears, the array is the list of ratios. The script
16 // uses the defined gear ratios to determine how much torque to apply to the wheels.
17 var GearRatio : float[];
18 var CurrentGear : int = 0;
19
20 // These variables are just for applying torque to the wheels and shifting gears.
21 // using the defined Max and Min Engine RPM, the script can determine what gear the
22 // car needs to be in.
23 var EngineTorque : float = 60.0;
24 var MaxEngineRPM : float = 800.0;
25 var MinEngineRPM : float = 200.0;
26 private var EngineRPM : float = 0.0;
27
28 function Start () {
29     // I usually alter the center of mass to make the car more stable. I'ts less likely to flip this way.
30     rigidbody.centerOfMass += Vector3(0, -1.0, .15);
31     BrakePower = 30;
32     // Stop all animations
33     animatedTarget.animation.Stop("Camera_Bump");
34     animatedTarget.animation["Camera_Bump"].speed = 3;
35 }
36
37 function Update () {
38
39     // Compute the engine RPM based on the average RPM of the two wheels, then call the shift gear function
40     //EngineRPM = (FrontLeftWheel.rpm + FrontRightWheel.rpm)/4 * GearRatio[CurrentGear];
41     EngineRPM = (FrontLeftWheel.rpm)/4 * GearRatio[CurrentGear];
42     ShiftGears();
43 }

```

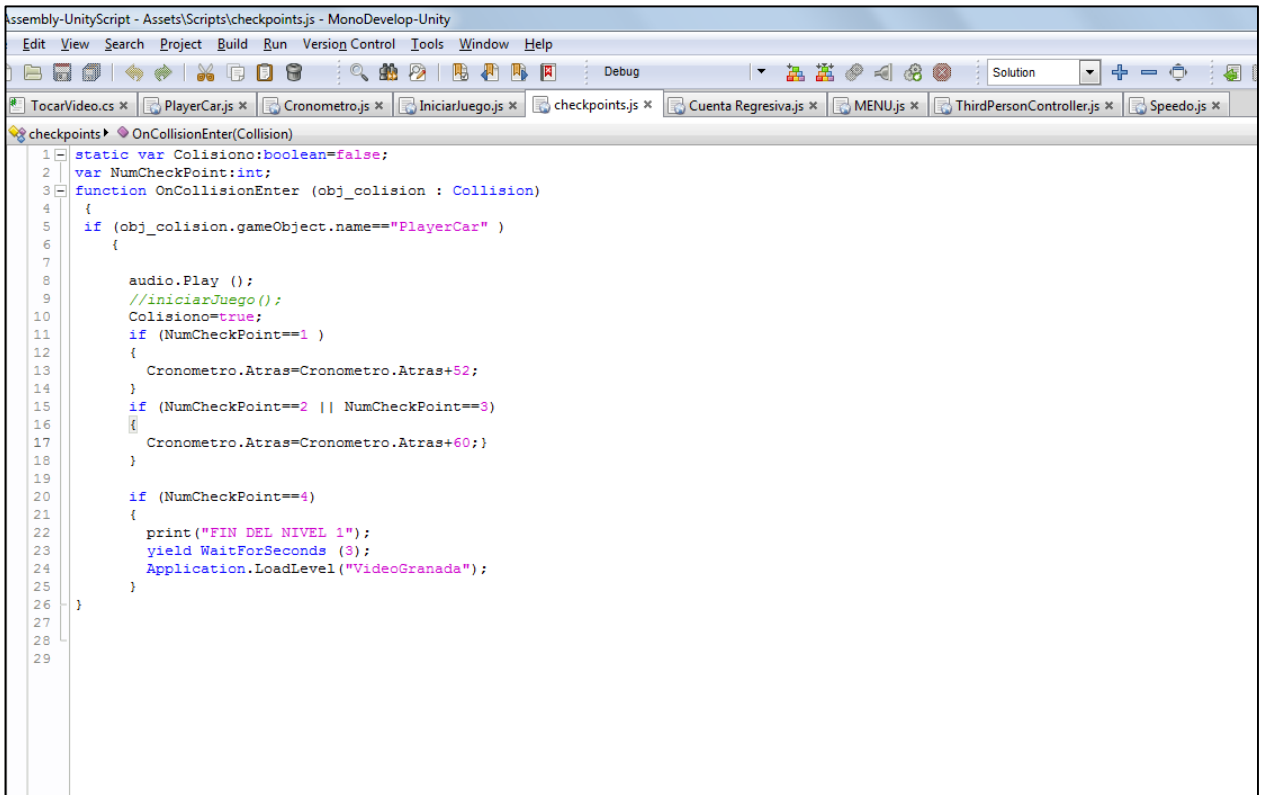
Figura. 54 Código de la clase playcar programado en java script.

Nica-Adventure “Descubriendo la Ruta de los Tesoros de Nicaragua”



```
Assembly-UnityScript - Assets/Scripts/Cronometro.js - MonoDevelop-Unity
e Edit View Search Project Build Run Version Control Tools Window Help
Debug
TocarVideo.cs x PlayerCar.js x Cronometro.js x IniciarJuego.js x checkpoints.js x Cuenta Regresiva.js x MENU.js x ThirdPersonController.js x Speedo.js x
Cronometro ▶ CoUpdate()
1 var cuenta:UIText;
2 var over:UIText;
3 var Maxima:int;
4 static var Atras:int;
5 private var Limite:int;
6 var auto:GameObject;
7 var conducirScript:Behaviour;
8
9 function Start(){
10     cuenta.enabled = true;
11     over.enabled =false;
12     auto= gameObject.Find("PlayerCar");
13     conducirScript = auto.GetComponent("PlayerCar");
14     //iniciarJuego();
15     Atras=51;
16     StartCoroutine("CoStart");
17 }
18
19 function CoStart() : IEnumerator
20 {
21     while (true)
22         yield CoUpdate();
23 }
24
25 function CoUpdate(): IEnumerator {
26
27     if (Atras>=1)
28     {
29         yield WaitForSeconds (1);
30         if (Maxima==0)
31             yield Disminuir();
32     }
33
34     Imprimir();
35
36
84 : 5 INS
```

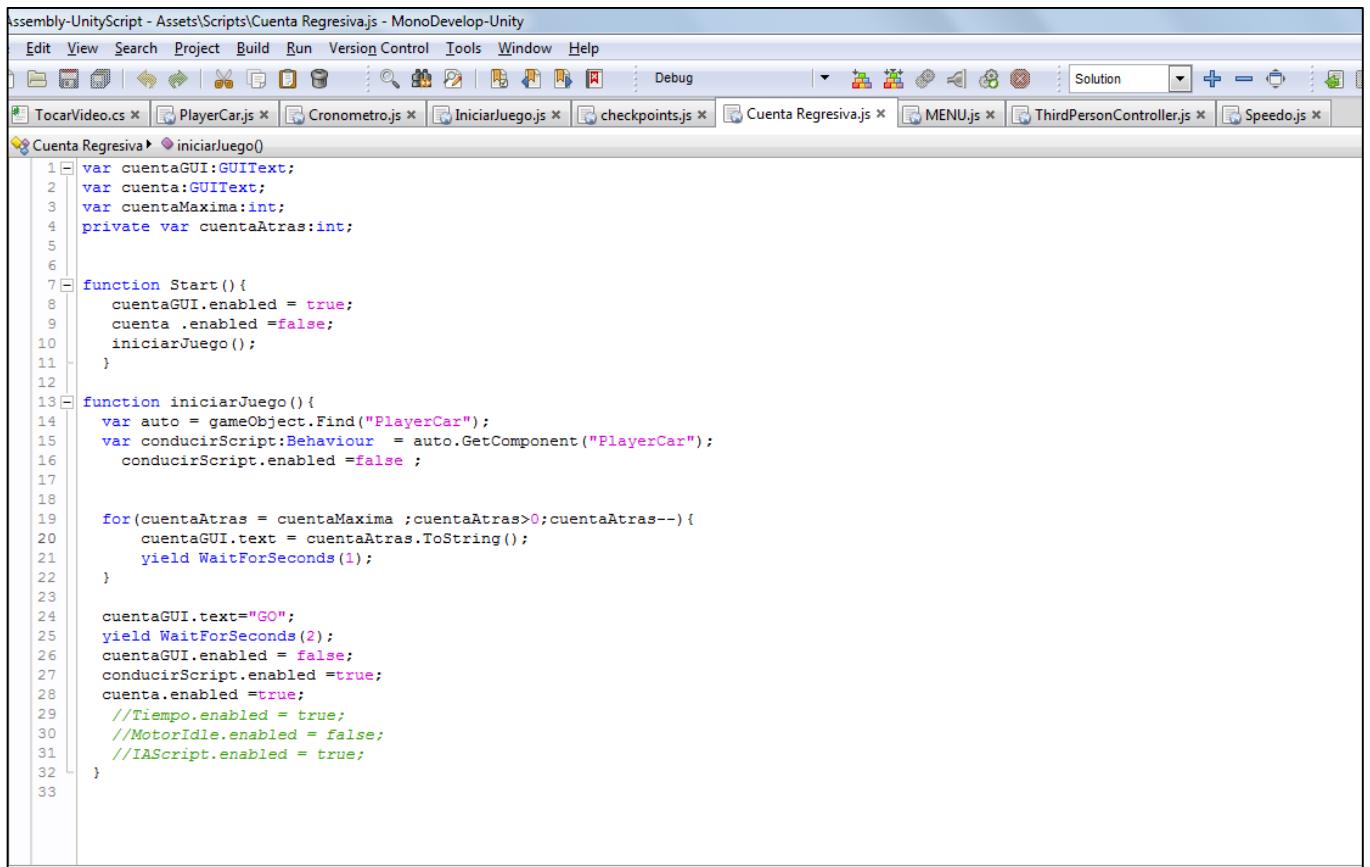
Figura. 55 Código de la clase Cronometro programado en java script.



```
Assembly-UnityScript - Assets/Scripts/checkpoints.js - MonoDevelop-Unity
Edit View Search Project Build Run Version Control Tools Window Help
Debug
TocarVideo.cs x PlayerCar.js x Cronometro.js x IniciarJuego.js x checkpoints.js x Cuenta Regresiva.js x MENU.js x ThirdPersonController.js x Speedo.js x
checkpoints ▶ OnCollisionEnter(Collision)
1 static var Colisiono:boolean=false;
2 var NumCheckPoint:int;
3 function OnCollisionEnter (obj_colision : Collision)
4 {
5     if (obj_colision.gameObject.name=="PlayerCar" )
6     {
7
8         audio.Play ();
9         //iniciarJuego();
10        Colisiono=true;
11        if (NumCheckPoint==1 )
12        {
13            Cronometro.Atras=Cronometro.Atras+52;
14        }
15        if (NumCheckPoint==2 || NumCheckPoint==3)
16        {
17            Cronometro.Atras=Cronometro.Atras+60;}
18        }
19
20        if (NumCheckPoint==4)
21        {
22            print("FIN DEL NIVEL 1");
23            yield WaitForSeconds (3);
24            Application.LoadLevel("VideoGranada");
25        }
26    }
27
28
29
```

Figura. 56 Código de la clase Checkpoints programado en java script.

Nica-Adventure “Descubriendo la Ruta de los Tesoros de Nicaragua”



```
1 var cuentaGUI:GUIText;
2 var cuenta:GUIText;
3 var cuentaMaxima:int;
4 private var cuentaAtras:int;
5
6
7 function Start(){
8     cuentaGUI.enabled = true;
9     cuenta.enabled =false;
10    iniciarJuego();
11 }
12
13 function iniciarJuego(){
14     var auto = gameObject.Find("PlayerCar");
15     var conducirScript:Behaviour = auto.GetComponent("PlayerCar");
16     conducirScript.enabled =false ;
17
18
19     for(cuentaAtras = cuentaMaxima ;cuentaAtras>0;cuentaAtras--){
20         cuentaGUI.text = cuentaAtras.ToString();
21         yield WaitForSeconds(1);
22     }
23
24     cuentaGUI.text="GO";
25     yield WaitForSeconds(2);
26     cuentaGUI.enabled = false;
27     conducirScript.enabled =true;
28     cuenta.enabled =true;
29     //Tiempo.enabled = true;
30     //MotorIdle.enabled = false;
31     //IAScript.enabled = true;
32 }
33
```

Figura. 57 Código de la clase Cuenta regresiva programado en java script.

8.5. RESULTADO 4: PROMOVER LA CULTURA NICARAGÜENSE POR MEDIO DE ESCENARIOS QUE PERMITAN DISFRUTAR LA OFERTA TURÍSTICA DE LA RUTA DE LOS TESOROS.

Como resultado del efecto de inmersión característica importante en un videojuego se han insertado elementos multimedia que promocionan y describen la cultura Nicaragüense evidenciando nuestra amplia propuesta gastronómica, nuestros bellos paisajes turísticos y enmarcando todo esto dentro de nuestra historia o guion del juego.

Estas son capturas de las pantallas que se muestran los contenidos multimedia insertados dentro del juego como elemento de inmersión en la cultura Nicaragüense:



Figura. 58 Breve Reseña Histórica de Granada.

Nica-Adventure “Descubriendo la Ruta de los Tesoros de Nicaragua”

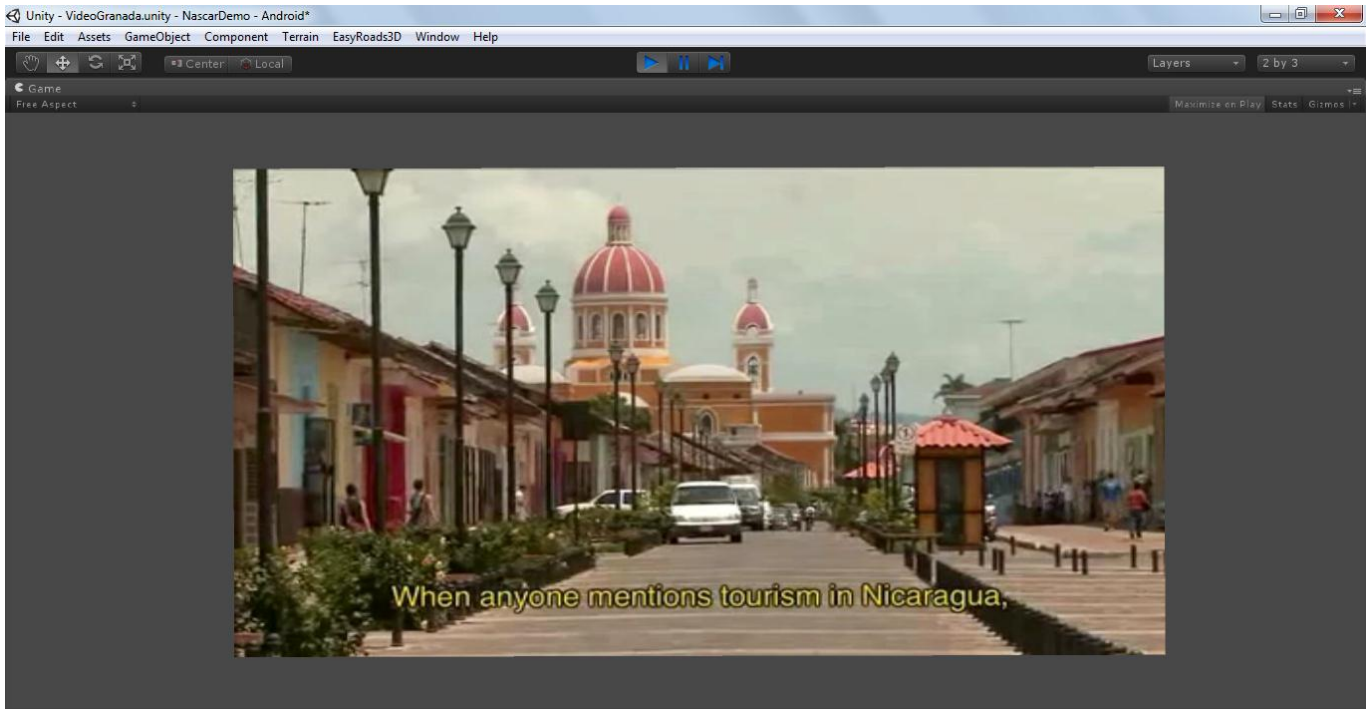


Figura. 59 Video en formato MP4 con la oferta turística de granada.



Figura. 60 Pasaporte de visitante otorgado al jugador que ha realizado el recorrido Managua-granada.

IX. CONCLUSIONES

A través de este trabajo se demostró en forma práctica la gran capacidad que poseen los dispositivos móviles actuales, capacidad tanto de procesamiento como de interacción con el usuario. También deja de manifiesto las posibilidades que se vislumbran en el ámbito de desarrollo de software para estos dispositivos, tanto en el área de computación gráfica como de software tradicional.

Mediante este trabajo se aplicó una gran gama de conocimientos adquiridos durante esta carrera, desde ciencias básicas, con cálculos trigonométricos bastante complejos, hasta ciencias de la computación con técnicas de programación avanzadas.

Además de los conocimientos entregados por la universidad, se debió adquirir conocimientos pertenecientes a otras áreas relacionadas con la elaboración de juegos, como son el diseño gráfico, modelado 3D y composición de música.

Durante la creación del juego surgieron muchos inconvenientes relacionados con el procesamiento, debido a la limitada capacidad de procesamiento de los teléfonos en comparación a los computadores tradicionales. Esto nos llevó a diseñar los objetos de tal forma que no cargaran la memoria del teléfono.

Fue difícil comenzar con el desarrollo de este proyecto, ya que el campo de programación de juegos 3D para dispositivos móviles está aún en sus etapas iniciales, por lo que encontrar engines destinados a ello y documentación fue bastante difícil.

Para finalizar, el objetivo general planteado fue: Desarrollar el videojuego nica-aventures en 3D para móvil en el primer semestre de 2013 en el curso de seminario de graduación de la UNAN-Managua, fue cumplido en su totalidad, logrando lo expuesto a través de este informe. En cuanto a los específicos, también todos fueron completados implementando las funcionalidades del juego.

X. RECOMENDACIONES

Se recomienda utilizar el programa 3Ds MAX para el diseño de personajes, objetos y texturas, para luego ser importadas y fácilmente manipulados en el motor que esté utilizando para la programación del juego.

Para facilitar el trabajo de inicio se debe seleccionar el motor a utilizar y documentarse del entorno de trabajo de él y con qué software de diseño y modelado 3D interactúa mejor.

Se recomienda que el juego sea ejecutado en plataforma android de 4.0 para arriba, con un procesador de 1.3GHz y memoria RAM de 1GB.

XI. BIBLIOGRAFÍA

ADOBE ADOBE [En línea] = ACERCA DE PHOTOSHOP. - 18 de 06 de 2013. - 16 de 08 de 2013. - <http://www.adobe.com/la/products/photoshop.html>.

Android Andriod [En línea]. - 2013. - 11 de 07 de 2013. - <http://www.android.com/>.

Autodesk Autodesk [En línea]. - 2013. - 14 de 05 de 2013. - <http://www.autodesk.es/>.

Autodesk Autodesk [En línea] = 3DS MAX. - 15 de 07 de 2013. - 12 de 08 de 2013. - <http://www.autodesk.es/products/autodesk-3ds-max/overview>.

CRECER12 CRECER12 [En línea] = NICARAGUA // CRECER12. - 2013. - 10 de 08 de 2013. - <http://crecer2012.org/>.

Ingsoft2011 [En línea] = Desarrollos de Ambientes Gráficos 2D y 3D. - 25 de 05 de 2012. - 15 de 07 de 2013. - <http://ingsoft2011.comeze.com/?p=8>.

Libre Androide El Androide Libre [En línea] = Primeros pasos en android. - 2013. - 25 de 06 de 2013. - <https://www.facebook.com/elandroidelibre>.

López José [En línea] = Android-Programación orientada a objetos.. - 23 de 07 de 2012. - 25 de 06 de 2013. - <http://tecsapiens.com/primeros-pasos-en-android/>.

MICROSOFT MICROSOFT [En línea] = Movie Maker. - 19 de 01 de 2010. - 26 de 08 de 2013. - <http://windows.microsoft.com/es-419/windows7/products/features/movie-maker>.

Ricardo. Guerrero LA PRENSA ONLINE [En línea] = Alianza empresarial para promover el turismo nacional // LA PRENSA ONLINE. - 23 de 12 de 2011. - 11 de 08 de 2013. - <http://www.elnuevodiario.com.ni/economia/236679>.

Wayne Carlson E. SCRIBD [En línea] = Una Cronología histórica de Computación Gráfica y Animación // SCRIBD. - 11 de 2010. - 30 de 07 de 2013. - <http://es.scribd.com/doc/54173433/historia-y-evolucion-del-dibujo-computacion-grafica-home>.

web Maestros del Maestros del web [En línea] // Maestros del web. - 2011. - 24 de 08 de 2013. - <http://www.maestrosdelweb.com/category/guias/guia-android/>.

Wikipedia [En línea] = Jugabilidad y experiencia del jugador. - 2009. - 02 de 07 de 2013. - <http://jugabilidad.wikispaces.com/Clasificaci%C3%B3n,+Tipo+y+G%C3%A9neros>.

Wikipedia Wikipedia [En línea] = Videojuego. - 2008. - 29 de 06 de 2013. - <http://es.wikipedia.org/wiki/Videojuego>.

Wikipedia Wikipedia [En línea] = Desarrollo de videojuegos. - 2010. - 29 de 06 de 2013. - http://es.wikipedia.org/wiki/Desarrollo_de_videojuegos.

Wikipedia Wikipedia [En línea] = Graficos 3D por computadora // Wikipedia. - 2013. - 30 de 07 de 2013. - http://es.wikipedia.org/wiki/Gr%C3%A1ficos_3D_por_computadora.

Wikipedia. Wikipedia. [En línea] = La Programación.. - 2013. - 11 de 06 de 2013. - <http://es.wikipedia.org/wiki/Programaci%C3%B3n>.

Wikipedia. Wikipedia. [En línea] = Lenguaje de programación.. - 2013. - 10 de 06 de 2013. - http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n.

Wikipedia. Wikipedia. [En línea] = Programación orientada a objetos. - 2013. - 25 de 06 de 2013. - http://es.wikibooks.org/wiki/Programaci%C3%B3n_Orientada_a_Objeto.