



Universidad de las Ciencias Informáticas

Grupo de Soporte a la Gestión de la Producción de la Dirección Técnica  
de la Infraestructura Productiva

# Sistemas de Lógica para el Videojuego de Carreras de Autos Rápido y Curioso

Informe de Aporte personal al proyecto Juegos de Consola

Presentado en opción del título de  
Máster en Informática Aplicada

Autor: Ing. Rolando Kindelan Nuñez.

Tutor: MS.c. Lidiexy Alonso Hernández.

Asesor: MS.c. Manuel Villanueva Betancourt.

La Habana

Junio 2010

## Declaración jurada de autoría

Yo, Rolando Kindelan Nuñez, con carné de identidad 84082633546, declaro que soy el autor principal del resultado que expongo en la presente memoria titulada “Sistemas de Lógica para el Videojuego de Carreras de Autos Rápido y Curioso”, para optar por el título de Máster en Informática Aplicada.

Este trabajo fue desarrollado durante el período 2005-2009 en colaboración con mis colegas de equipo, quienes me reconocen la autoría principal del resultado expuesto en esta memoria.

Autorizo a la Universidad de las Ciencias Informáticas a hacer el uso que estime pertinente de los resultados aquí presentados, como propietaria de los derechos legales de este proyecto.

Finalmente declaro que todo lo anteriormente expuesto se ajusta a la verdad, y asumo la responsabilidad moral y jurídica que se derive de este juramento profesional.

Y para que así conste, firmo la presente declaración jurada de autoría en Ciudad de la Habana a los \_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_.

---

Rolando Kindelan Nuñez

## Agradecimientos

A mi madre, por todo el amor, la comprensión y el apoyo brindado incondicionalmente desde el comienzo. A mi padre, por preocuparse, darme tanto apoyo, confianza y mantenerse fuerte en sus principios a pesar de los momentos tan duros por los que ha estado pasando.

A mi novia, por todo el amor, la comprensión y además por convertirse en ese pie de apoyo tan necesario en estos últimos momentos, que tan difíciles me han resultado, por creer en mí y decirme en algún momento que el stress da cáncer.

A toda mi familia. A mis amigos de la UCI. A mis compañeros de la Dirección Técnica de la IP, que supieron darme apoyo y colaboraron para que pudiera dedicar parte de mi jornada laboral al desarrollo de este trabajo. A Lidiexy, mi tutor, por toda la paciencia que ha tenido conmigo, por tener esa visión tan necesaria y por estar ahí para mí cada vez que acudí en su ayuda. Al profesor Villanueva, por ayudarme incondicionalmente a darle forma a la investigación que arrojó como resultado fundamental, los elementos en los que se basa este trabajo.

A Fidel y Raúl, por ser los artífices de esta Revolución que me ha dado la oportunidad de seguir madurando como profesional. A la UCI por ser la cuna de tantos profesionales y por brindarme los medios necesarios para llegar hasta hoy.

A mis profesores Lago e Igr, por ser los motores impulsores del proyecto “Juegos de Consola”, y cuyo trabajo fue uno de los pilares fundamentales de mi desarrollo como profesional. Al equipo de desarrollo de “Rápido y Curioso”, en especial a Carlos, Yordanis, Yury, Erislandis, Hassan y Osvaldo por ayudarnos a terminar la memoria colectiva. A Marvyn por ser mi compañero desde mis primeros días en la UCI, y ser el amigo con el que compartí la mitad del peso del desarrollo de la Maestría.

En fin a todas aquellas personas que de una forma u otra han contribuido al desarrollo de este trabajo.

Muchas Gracias por todo.

## Síntesis

La industria de los videojuegos a nivel mundial, es una fuente de ingresos económicos importante para cualquier país. Teniendo en cuenta esta razón y ante la necesidad de modernizar los parques de diversiones cubanos, surge en la Facultad 5 de la Universidad de las Ciencias Informáticas, el proyecto “Juegos de Consola” con el objetivo de proveer productos competitivos y al mismo tiempo fomentar el desarrollo de la industria cubana de videojuegos.

Desde sus inicios en el 2004, este proyecto fue encargado de la realización del producto “Rápido y Curioso”, un videojuego de conducción de automóviles que simula un recorrido por las calles de Ciudad de La Habana. El juego cuenta con un entorno gráfico tridimensional (3D), sonido realista 3D, interacción con el usuario, además de la posibilidad de ser jugado en red.

En este trabajo específicamente se abordan los sistemas de lógica elaborados para el mismo, como respuesta a la pregunta fundamental con que siempre comienza el desarrollo de estos tipos de videojuegos: ¿Qué es lo que debe hacer el jugador?

La lógica del videojuego como se mencionaba anteriormente, se elaboró en forma de varios sistemas muy relacionados entre sí: un editor de pistas como sistema de adquisición de datos de la pista y el entorno, que además describe los circuitos de la carrera; e interfaces lógicas que interpretan la información obtenida a través del editor, respetando una arquitectura base con dos niveles de abstracción (a nivel de auto y de juego) que además puede ser extendida para soportar varios modos de juego.

Junto a cada decisión tomada durante el desarrollo, se expone además una valoración donde se destacan sus ventajas y resultados.

**Palabras Claves:** juegos de consola, rápido y curioso, lógica de videojuegos, sistemas de lógica, editor de pistas, interfaz de lógica, modos de juego.

# Índice de Contenidos

<b>1. Introducción.....</b>	<b>1</b>
<b>2. Desarrollo .....</b>	<b>4</b>
<b>2.1. Fundamentación Teórica .....</b>	<b>4</b>
2.1.1. <i>Diseño de videojuegos .....</i>	<i>4</i>
2.1.2. <i>Lógica de videojuegos.....</i>	<i>7</i>
<b>2.2. Contribución Personal.....</b>	<b>11</b>
2.2.1. <i>Arquitectura del Módulo de Lógica.....</i>	<i>11</i>
2.2.2. <i>Informaciones que intercambia el módulo con otros módulos del videojuego. ....</i>	<i>16</i>
2.2.3. <i>Aportes Fundamentales .....</i>	<i>18</i>
<b>3. Conclusiones .....</b>	<b>25</b>
<b>4. Recomendaciones .....</b>	<b>26</b>
<b>5. Glosario de Términos .....</b>	<b>27</b>
<b>6. Referencias .....</b>	<b>30</b>

# 1. Introducción

Los videojuegos han experimentado un crecimiento considerable en cuanto a escala y complejidad desde sus modestos comienzos en la década de los 60. En los días actuales los videojuegos se caracterizan por sus elevados niveles de realismo, especialmente en áreas como los gráficos, la simulación física, la inteligencia artificial, entre otros aspectos. [1]

Entre la amplia gama de videojuegos existentes, los de conducción de automóviles<sup>1</sup> han alcanzado una amplia demanda dentro de la industria del entretenimiento. Algunos de los mismos son de entretenimiento ligero, diseñados para entretener al jugador durante varios minutos, sin que necesiten en su elaboración una filosofía del diseño en detalle [2]. En cambio, otros videojuegos más complejos, persiguen como objetivo elevar al jugador a un lugar maravilloso y que jugarlo sea una experiencia única cada vez, debido a la alta interactividad que posee el usuario [3].

Las razones anteriores constituyeron un elemento de peso para que nuestro país, decidiera modernizar los parques de diversiones introduciendo videojuegos que reflejen la realidad nacional y eleven a otros niveles el entretenimiento que estas instituciones proveen, brindándoles a los niños cubanos una recreación sana.

Desde el 2003 en la Facultad 5 de la Universidad de las Ciencias Informáticas (UCI), se venían desarrollando aplicaciones de Realidad Virtual, como simuladores de conducción de automóviles, simuladores de entrenamiento de tiro, paseos virtuales, maquetas tridimensionales (3D) y demás. Estos hechos permitieron que en el 2004 se le asignara a esta facultad, la misión de elaborar un videojuego de conducción de automóviles para niños de entre 8 y 12 años de edad y que se llamara Rápido y Curioso. Este videojuego debía ser implantado inicialmente en un laboratorio preparado para este fin en el Parque Lenin de Ciudad de la Habana.

Para el cumplimiento de esta misión se creó el proyecto “Juegos de Consola”, con un equipo conformado por estudiantes y profesores de la Facultad. El producto resultante de los esfuerzos de este equipo fue el videojuego “Rápido y Curioso”, que cuenta con un entorno gráfico tridimensional (3D), sonido realista 3D, interacción con el usuario, además de la posibilidad de ser jugado en red.

El proceso de creación de “Rápido y Curioso” fue un proceso exigente porque este producto fue el primer intento en el país de desarrollar un videojuego de estas características, por lo que inicialmente no se contaba con la experiencia necesaria para emprender este tipo de proyecto.

---

<sup>1</sup> **Videjuegos de conducción de automóviles:** También conocidos como videojuegos de carreras de autos.

Debido a lo anterior se emprendió un trabajo intenso de investigación, dirigido por profesores con una alta preparación, pero que tampoco tenían experiencia en el área.

Entre los retos a que el equipo tuvo que enfrentarse, estaba el darle respuesta a: ¿Qué es lo que debe hacer el jugador? Que constituye la pregunta fundamental con que comienza el diseño de estos videojuegos [4].

Debido a que entre otras cosas, se necesitaba definir cuál era el rol que iba a desempeñar un jugador del videojuego, cuáles eran sus objetivos, limitaciones y las acciones que podría realizar en cada momento; el problema a resolver para responder las interrogantes anteriores fue el de: *¿Cómo incorporar roles, reglas, acciones y objetivos de los jugadores que den sentido al videojuego Rápido y Curioso?*

Persiguiendo como objetivo general el de *desarrollar la lógica del videojuego Rápido y Curioso*. Enmarcando la investigación en la *lógica de videojuegos* como objeto de estudio, y centrándose en la *lógica de videojuegos de carreras de autos* como campo de acción.

Además se plantearon un grupo de tareas para guiar la investigación y dar cumplimiento al objetivo planteado, que se pueden resumir en las siguientes:

- Elaborar un guión de juego<sup>22</sup> que defina las reglas a seguir, los roles, objetivos, situaciones y demás elementos filosóficos que darían forma al juego.
- Analizar la lógica de videojuegos existentes de corte similar para detectar elementos que pudieran aportar al desarrollo de la lógica del Rápido y Curioso.
- Asimilar la biblioteca gráfica G3D, para su utilización durante el desarrollo del videojuego.
- Elaborar un Editor de Pistas de Carrera con una interfaz gráfica de usuario amigable que permita:
  1. Editar gráficamente el circuito de carrera sobre una base de geometría tridimensional.
  2. Importar entornos geométricos en formato BSP que sirvan de base para el circuito.
  3. Imponer y probar un conjunto de reglas de tráfico.
  4. Exportar los circuitos a un formato binario TRK.
  5. Importar los circuitos desde un formato binario TRK.
- Desarrollar los elementos necesarios para procesar la información obtenida de la pista, de manera que sea útil a los módulos del videojuego.

---

<sup>22</sup> **Guión de Juego:** Conocido también como Documento de Diseño del Juego y describe el concepto general del juego, el público a quien está dirigido el juego, qué se hace, entorno en que se desarrolla, reglas, objetivos, personajes, roles de los personajes, etc [4].

Con el cumplimiento de estas tareas se esperaba implementar estructuras y modelos que permitieran gestionar la información de la pista, el entorno; así como las funcionalidades necesarias para procesarla coherentemente de manera que se satisfaga el guión del videojuego.

Como resultado se creó un módulo con estructuras bien definidas para la edición de las pistas de carreras, extracción de información de la pista y del entorno; y que contara además con las interfaces necesarias para procesar toda la información convenientemente a niveles diferentes de abstracción (auto y juego), estableciendo comunicación con el resto de los módulos del videojuego.

En el Informe Final del Proyecto [\[5\]](#), se dedica el capítulo nueve a la descripción del módulo de lógica elaborado. En el capítulo ocho se abordan los mapas utilizados en el videojuego así como su proceso de creación, optimización y uso.

## 2. Desarrollo

En esta sección se abordarán los temas que como contribución intelectual el autor del presente trabajo aportó al proyecto Juegos de Consola y que se encuentran descritos en el capítulo nueve del Informe Final del Proyecto [5].

Inicialmente se presenta un pequeño análisis teórico sobre diseño y lógica de videojuegos. Posteriormente se muestra a grandes rasgos, la solución técnica propuesta por el autor para desarrollar la lógica del Rápido y Curioso. Se explicarán decisiones que se tomaron frente a cada reto que se fue presentando.

Más adelante se hace un análisis de la información que intercambia el módulo con el resto de los componentes del videojuego y por último se mostrarán los aportes principales del trabajo.

### 2.1. Fundamentación Teórica

#### 2.1.1. Diseño de videojuegos

Contrario a lo que mucha gente piensa, el primer paso en el desarrollo de cualquier tipo de videojuego -incluso antes de pensar cómo debe ser el sistema- es el diseño del videojuego<sup>3</sup> realizado por un diseñador de videojuegos<sup>4</sup>. El diseñador del juego puede desempeñar una

---

<sup>3</sup> **Diseño del juego:** Es lo que determina la forma del juego, las elecciones que los jugadores serán capaces de tomar en el juego y las consecuencias que esas decisiones tendrán en el resto del juego [4]. Además el diseño del juego establece cuáles serán los criterios de ganar o perder que el juego incluirá, cómo el usuario será capaz de controlar el juego, qué informaciones el juego deberá intercambiar con él y además establece cuán difícil será el juego. En resumen, el diseño del juego determina todos los detalles de cómo el juego funcionará [6].

<sup>4</sup> **Diseñador de juegos:** Determina la naturaleza del juego creando el diseño del juego [4].

amplia variedad de tareas en un proyecto, pero su preocupación fundamental debe ser siempre el diseño del juego y la jugabilidad<sup>5</sup>.

### Proceso de Diseño de videojuegos

Entre los procesos de diseño de videojuegos más comunes se encuentra el proceso de diseño incremental [6]. Este proceso establece que el diseño, prueba y evaluación de los resultados deben realizarse una y otra vez durante todo el desarrollo del videojuego, cada vez que haya una mejora en la jugabilidad o en las características, hasta que se satisfagan las expectativas del diseñador. Ver Figura 1.



Figura 1. Proceso de Diseño Iterativo.

### Documento de Diseño

El documento de diseño es el artefacto principal del diseño del juego, es realizado por los diseñadores de juegos y su cliente objetivo es el equipo de desarrollo del videojuego [2]. Este documento es una referencia textual utilizada en el desarrollo de un juego, que trata de describir con detalle todos los aspectos importantes del diseño del juego. No existe un formato

<sup>5</sup> **Jugabilidad:** Es un término empleado en el diseño y análisis de juegos que describe la calidad del juego en términos de sus reglas de funcionamiento y de su diseño como juego.

estándar para este documento, el formato siempre es a elección de los diseñadores de juegos [7].

Entre sus principales secciones se pueden contar las siguientes:

- **Mecánica de Juego:** Es la parte más importante del documento, también se le llama sección del juego porque describe lo que a los jugadores se les permite hacer y cómo se juega el juego [3].
- **Inteligencia Artificial o Mundo:** Describe cómo el mundo va a reaccionar a las acciones de los jugadores, incluso cuando estos no están haciendo nada [4]. ¿Cómo se comportan los opositores a los que se enfrentan los jugadores en el mundo del juego? ¿Qué van a hacer en cada situación? [6]
- **Elementos de Juego o Caracteres:** Estos elementos son las diferentes partes del juego, que se reunirán en los niveles para crear una experiencia atractiva para los jugadores [4]. Los diseñadores podrán tomar estos elementos y, mediante la combinación de ellos de una forma única e interesante [8], crear una variedad de niveles que mantengan a los jugadores interesados durante horas [3]. En el caso de que el juego no contenga niveles, se puede asumir que tiene un único nivel y ubicar todos los elementos del juego en el mismo [6].
- **Resumen de la Historia:** En esta sección se realiza la reseña de la historia que se cuenta en el juego [8]. Es una forma rápida para que todos en el equipo comprendan la historia a grandes rasgos y cómo debe ser mostrada en el videojuego [7]. Además esta sección debe escribirse de una manera que sea agradable a la lectura [2].
- **Progresión del Juego:** Dependiendo de la naturaleza del juego, esta sección puede resultar la más larga del documento. Aquí es donde se profundiza en las experiencias por las que puede atravesar el jugador [7], y cómo cambian con el progreso del tiempo [3]. En esta sección se proporciona una guía para el equipo de arte y los diseñadores de niveles en cuanto a qué tipos de ambientes se tendrán que crear para el juego [4]
- **Sistemas de Menús:** Aquí se debe especificar el menú principal y cualquiera de las otras pantallas de opciones que se presentarán al usuario, en varios puntos fuera del juego en sí [6]. La realización de estos menús no tiene un impacto significativo en la jugabilidad, y como resultado se debe separar en una sección independiente. Se deben incluir descripciones de cómo los jugadores guardarán sus juegos y cómo los van a cargar más tarde [4]. Además, se deben describir los tipos de interfaz en que los jugadores accederán a estos menús: si van a usar el puntero del ratón y hacer clic, o se utilizarán del teclado las teclas Enter, las flechas, o ambas [2].

Una vez diseñado el juego, el documento de diseño es utilizado constantemente por el resto del equipo de desarrollo, para auxiliarse y saber cómo tienen que hacerse las cosas. El documento de diseño marca las pautas por las que se rige la lógica del juego.

## 2.1.2. Lógica de videojuegos

Lamentablemente, existe muy poca documentación acerca de lógica de juegos. No se sabe si es producto de que la mayoría de los videojuegos actuales, son desarrollados con licencias propietarias y no publican los resultados de sus investigaciones, o si es debido a que nuestro país por lo general no tiene acceso a muchas de las fuentes donde estos documentos son publicados.

Aun así, se logró comprobar que el concepto de lógica de juego es muy polémico, encontrándose muchas divergencias entre los autores, siendo la siguiente la definición más cercana al interés de este trabajo:

**Lógica del Juego:** En un principio, se tiende a pensar que la Lógica del Juego se simplifica en implementar la inteligencia de los personajes del videojuego. No sólo implica esto, también se tienen que programar todos los comportamientos de todas las escenas (por ejemplo, "si el jugador pasa cerca de la puerta, que se abra"). En la Lógica del Juego se diseña y se programan todos los algoritmos necesarios para definir e implementar las reglas del juego, la interacción entre los elementos y los comportamientos de todos los avatares. Es el gran núcleo del videojuego y donde más programadores son necesarios [9].

Es decir, que la Lógica del Juego es una interpretación a nivel de sistema de software del documento de diseño del videojuego. A menudo se considera como el módulo más importante del videojuego, porque es el que transforma una agrupación de código fuente, en un producto "jugable" de valor observable para los usuarios.

En la mayoría de los videojuegos de auto, este módulo se encarga de:

- Inicializar el contenido del juego y crear el mundo donde se desarrollará la acción.
- Determinar cuándo empieza y cuándo termina una partida.
- Implementar las reglas que controlan las acciones de los jugadores y controlar que estas reglas se cumplen durante el desarrollo de la partida.
- Permitir que los jugadores interactúen con el entorno sin romper estas reglas.
- Controlar el estado de todos los elementos que están participando en la partida, desde las posiciones de los avatares hasta las variables globales de la partida: tiempo de juego, puntuación, resultados y demás elementos.

El proceso de ejecución de estas tareas generalmente se subdivide en dos fases:

– **Control de física y de colisiones:** se calcula si las posiciones y acciones que se piden se pueden llevar a cabo o si existen interacciones entre los elementos. En este caso el sistema de física decide cuál es el movimiento real de los elementos. La salida de esta fase indica las posiciones finales de los objetos en el mundo. Los cálculos de esta fase siempre se realizan en un tiempo máximo definido que permita la sensación de fluidez al usuario.

– **Control de la Lógica del Juego:** se vela por el cumplimiento de las reglas del juego y que el comportamiento de los elementos esté dentro de los márgenes [10]. También se detecta si el juego ha finalizado en este último paso. Finalmente, se actualizan todas las variables globales de la partida: tiempos, puntuaciones, y demás elementos.

Un enfoque muy utilizado para el control y ejecución de estas tareas se basa en la utilización de máquinas de estados [11], dividiendo diferentes situaciones en que puede encontrarse el juego o un jugador específico en estados, cada uno de los cuales tiene un conjunto definido de acciones a ejecutar [9]. Esto es frecuentemente implementado utilizando el patrón de diseño Estado/State, que permite a un objeto determinado alterar su comportamiento cuando su estado interno cambia [12].

Otros autores prefieren abordar la lógica de videojuegos desde una perspectiva basada en reglas, como el que se plantea en [13].

### Ejemplos de la lógica de otros videojuegos

Ya que la documentación actual del diseño de la arquitectura en los juegos existentes es virtualmente inexistente, no ha quedado otra salida que inferir los guiones y por ende, la lógica de algunos de los videojuegos de conducción de autos más conocidos como: Need for Speed, Grand Theft Auto, Mario Kart y Reevolt.

**Need for Speed (NFS):** Es una saga de videojuegos de carreras publicado por Electronic Arts. Es un juego de carreras arcade/simulador, que ubica a los jugadores en las calles, carreteras y autopistas reales; además les brinda la posibilidad de conducir automóviles reales como deportivos, todoterrenos, entre otros (Ver Figura 1). Algunas de sus modalidades son: contrarreloj, derrape, persecución, carrera simple, torneo y otra en la que se va eliminando el último corredor hasta que sólo queden dos (este modo de juego es conocido como knock-out). Los entornos del juego pueden contener un diseño abierto<sup>6</sup> o cerrado<sup>7</sup> [14].

---

<sup>6</sup> **Diseño abierto:** Permite que el auto salga del asfalto y tome atajos.

<sup>7</sup> **Diseño cerrado:** Basado solamente en un argumento determinado o misiones a ejecutar.



Figura 2. Escena del NFS.

**Grand Theft Auto (GTA):** Es una saga de videojuegos de género acción/aventura del tipo "sandbox" creada por David Jones y luego por Sam Houser y Dan Houser. Originalmente fue creada por DMA Design, que posteriormente se llamó Rockstar North, de la compañía Rockstar Games. GTA cuenta la historia de distintos criminales, que se relacionan y se envuelven en problemas con otros personajes conforme va pasando el tiempo (Ver Figura 2). Lo revolucionario del género, fue la libertad de acciones que podía tomar el jugador, aparte de la polémica que creó por su alto contenido en violencia. Tenía una vista superior y un potencial limitado, pero sentó las bases de la saga [15].



Figura 3. Escena del GTA.

**Mario Kart:** Es un videojuego de carreras diseñado por Nintendo EAD. El juego innovador inauguró el género de carreras alcohólicas. Como indica su nombre, los vehículos son carros de carrera o de motor (kart en lo adelante), aunque el juego no fue creado por su realismo, el kart posiblemente fue el vehículo más apropiado para crear un juego desenfadado, en donde no hay reglas y en el que cada kart se vale de todo tipo de objetos mágicos repotenciadores o usados como arma o para estorbar e impedir que el rival llegue antes a la meta [16].



Figura 4. Escena del Mario Kart 64.

**Revolt:** Es un juego cuyos vehículos son de control remoto que se mueven por el interior de varias edificaciones. Cada auto posee un arsenal de armas y demás artefactos para obstaculizar la carrera de los contrarios, sean dirigidos o inteligentes. Muy bien logrado y hace al usuario sentirse realmente pequeño como el RC<sup>8</sup> que está manejando [17].



Figura 5. Escena del Revolt.

<sup>8</sup> RC: Se refiere a automodelismo, una de las tres modalidades que existen de radio control, las otras dos variantes son: Aeromodelismo y Modelismo naval.

## 2.2. Contribución Personal

La principal contribución del autor de este trabajo al videojuego Rápido y Curioso, se centra en la lógica del mismo. La lógica del videojuego no fue diseñada desde su comienzo, sino aproximadamente a poco más de la mitad del proceso de desarrollo, cuando ya existían varios módulos desarrollados.

Fue entonces que el autor, que en ese entonces se desempeñaba entre otras cosas como uno de los integradores del proyecto, se percató de la necesidad de un módulo que unificara al resto de los componentes y le diera un sentido a todo el caudal de código existente.

Quizás por esa razón se buscó desarrollar la lógica del videojuego de forma desacoplada al resto de los componentes del mismo, formando una especie de módulo de servicio que se ocupara de realizar los cálculos concernientes a los detalles lógicos del videojuego, que pudieran ser utilizados por el resto de los módulos y les permitiera enfocarse estrictamente en sus funciones [5].

Debido a que lo que se necesitaba era que el módulo brindara servicios a los demás componentes del videojuego, se optó por un *Estilo Arquitectónico Centrado en los Datos*<sup>9</sup> para establecer la integración entre todos los módulos desde el punto de vista lógico [1]. Donde el módulo de lógica garantizaría la información a consumir por el resto de los componentes del juego.

Esto fue todo un reto desde el punto de vista de diseño de software, pero finalmente el resultado fue un módulo formado por varios sistemas o partes que interactúan entre sí [1], manteniendo la modularidad conseguida en el proyecto hasta el momento y garantizando una mejor interacción entre los componentes del videojuego.

### 2.2.1. Arquitectura del Módulo de Lógica

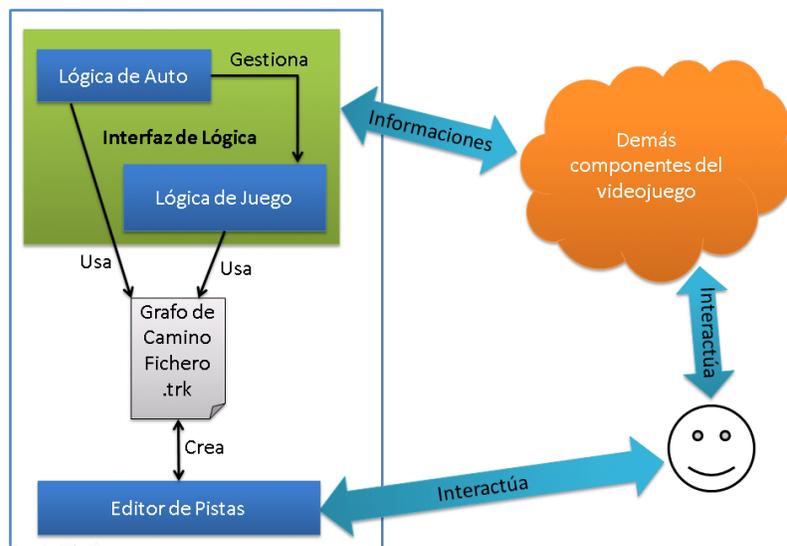
Para la confección del módulo de lógica del videojuego Rápido y Curioso, fue necesario desarrollar un Editor de Pistas para crear los circuitos de carrera, mediante los cuales se podría adquirir la información necesaria de la pista, de la carrera y del entorno.

---

<sup>9</sup> **Estilo Arquitectónico Centrado en los Datos:** Es utilizado para crear una integrabilidad a nivel de datos [1].

Estos circuitos se modelan con grafos de camino dirigidos, cuyos nodos se ubican convenientemente en el inicio de la carrera, cuando existen bifurcaciones y en el final de la carrera. Los nodos están enlazados por segmentos dirigidos que describen la pista en ese tramo e indican además el sentido de la trayectoria. Esta información es almacenada en un fichero \*.trk. Este proceso se realiza en tiempo muerto, es decir, antes de efectuar la carrera.

Estos grafos de camino son interpretados por componentes lógicos que brindan una interfaz de comunicación con el resto de los módulos del videojuego. Estos componentes lógicos se crearon con una arquitectura base que puede ser extendida para soportar otros modos de juego, y está formada por dos niveles diferentes de abstracción: un componente que engloba los detalles lógicos del auto y otro que encapsula la Lógica del Juego. Ambos componentes utilizan el grafo de camino correspondiente al circuito que está siendo recorrido. La Figura 5 ilustra lo explicado anteriormente.



**Figura 6.** Arquitectura del Módulo de Lógica.

## Editor de Pistas

Como se comentaba anteriormente, un Editor de Pistas de Carreras es la herramienta utilizada para realizar el trazado de los circuitos de las pistas de carreras del videojuego en cuestión, mediante la cual el diseñador de pistas de carreras define el recorrido construyendo de forma manual un grafo de caminos con la trayectoria a seguir por los competidores, también puede definir tipos de terrenos en partes de la pista, variándole parámetros tales como fuerza de rozamiento a los tramos de la pista; de esta forma se consigue simular hielo, arena, entre

otros, todo es cuestión de la imaginación del usuario y de las posibilidades que brinde el Editor de Pistas utilizado.

Este grafo de caminos es guardado en un fichero binario que almacenará el recorrido de la pista, así como otros datos de interés y será encuestado por el sistema en todo momento de la carrera para cada competidor, para comprobar que no se produzcan violaciones del sentido del circuito previamente definido, para llevar el control de la carrera y de la dirección de cada conductor; y también para simular las condiciones físicas de la superficie del tramo actual.

La edición del circuito de carrera se realiza antes de utilizar la pista en el videojuego, por esa razón los Editores de Pistas tienen la opción de explorar dicho circuito una vez que se crea. La herramienta implementada en el proyecto para editar los mapas del videojuego, presenta dos modos de utilización, un modo de edición donde el usuario edita el circuito de carrera y otro de prueba o exploración que permite al usuario recorrer la pista para comprobar si obtuvo el resultado deseado en el proceso de edición. Para más información ver los capítulos ocho y nueve del informe final del proyecto [5].

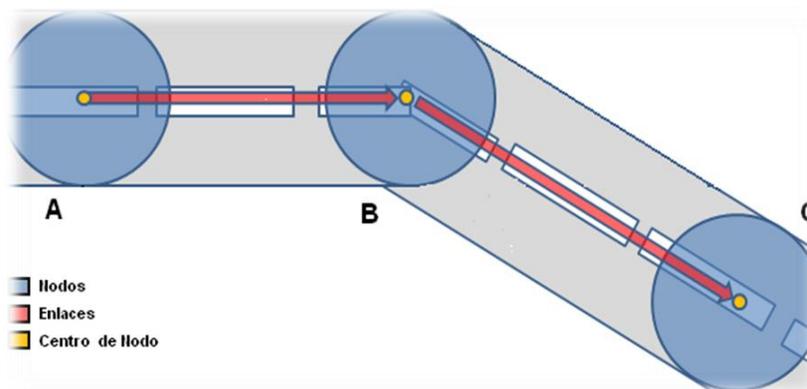
## Lógica del Auto

La Lógica del Auto, es el elemento que encapsula los detalles lógicos elementales de cada entidad del videojuego. Brinda al resto de los módulos una interfaz de lógica a nivel de auto que es encuestada frecuentemente para obtener informaciones esenciales que le facilitan el trabajo.

Entre las principales funcionalidades de la Lógica del Auto están las siguientes:

1. **Saber si se puede jugar.** El vehículo puede jugar siempre y cuando el auto no esté en pausa y no se haya acabado el juego. Esta funcionalidad debe ser encuestada para poder realizar cualquier acción que implique el movimiento del auto, en cualquier componente del videojuego, ya sea simulación física, inteligencia artificial (IA), la misma actualización de nodos, entre otros. Esto garantiza que cuando no sea posible jugar, no exista ningún elemento funcional hasta que llegue la señal lógica que lo permita.
2. **Indicación de hacia dónde debe ir.** Se representa mediante una flecha que indica la dirección del siguiente tramo (Ver Figura 8).
3. **Cantidad de vueltas recorridas.** Representa la cantidad de veces que se llega al nodo de tipo parcial, es decir que cuando el nodo actual es de tipo parcial quiere decir que el auto ha recorrido una vuelta completa.
4. **Lugar que ocupa en la carrera.**
5. **Tiempo que lleva de carrera.**

6. **Saber si está en sentido correcto.** Se verifica la proyección de la dirección actual del auto, con la dirección establecida por el circuito en ese tramo, si da positivo está en sentido correcto.
7. **Pausar el auto.**
8. **Última Dirección correcta.** Cuando ocurre algún incidente que provoque que el vehículo se vuelque o se salga de la pista, es deseable poder ubicarlo en la última posición correcta que tuvo durante la trayectoria. Esto se logra hallando el vector de desplazamiento erróneo del auto y calculando su proyección en la trayectoria correcta.
9. **Gestión de Puntuaciones.** Gestiona los 10 mejores tiempos, si el usuario entra en el ranking se muestra un panel con las puntuaciones y la puntuación del usuario seleccionada.
10. **Saber si acabó el juego.** Para un vehículo el juego se acaba, cuando el número de vueltas recorridas es mayor o igual al máximo de vueltas a recorrer o cuando el cronómetro indique un tiempo mayor que el tiempo límite de carrera (definido en ocho minutos: 00:08:00) y pierde el juego solamente cuando el cronómetro o tiempo de carrera exceda el tiempo límite.
11. **Mensajes de estado lógico y del final del juego.** Estos mensajes se utilizan para la comunicación entre las entidades del juego que están del lado del servidor (donde se ejecuta la física, la lógica y demás cálculos pesados) con las entidades del lado del cliente (que sólo muestran información). El mensaje de estado lógico es como una instantánea que se hace en todo momento de la lógica del auto. Hay otro tipo de mensaje que se envía al finalizar el juego y almacena la información de si el jugador actual es o no ganador, un arreglo de puntuaciones y el índice que representa la posición del jugador en la lista de puntuaciones. Estos datos son enviados a través de la red y utilizados en los clientes para mostrar la información y actuar en consecuencia.



**Figura 7.** Grafo de Caminos en un tramo de la pista.



**Figura 8.** Captura de pantalla que muestra la flecha de dirección.

En la Figura 7 se muestra un esquema de los nodos con su centro y los enlaces entre ellos, que describen un tramo de la pista. En la Figura 8, se muestra la flecha de dirección que indica al auto hacia dónde tiene que dirigirse (suponiendo que el auto está actualmente en el nodo **A** de la Figura 7, la flecha de dirección describe la dirección del enlace que une el nodo **C** al **B**).

## Lógica del Juego

La Lógica del Juego como su nombre indica, encapsula la lógica a nivel de juego y para esto realiza diversas funciones entre las que se encuentran las siguientes:

1. **Gestionar la lógica de los autos.** Aquí se hace coincidir cada entidad del juego con un componente de Lógica de Auto, se encuestan cada una de las funcionalidades de la lógica a nivel de auto para tomar decisiones a nivel de juego.
2. **Cuándo inicia el Juego.** El juego inicia una vez que hayan pasado tres segundos de que todos los clientes se hayan conectado.
3. **Posición de los autos en la carrera.** Para esto se halla el nodo más lejano visitado en la carrera por los autos, se toman las distancias de cada auto al siguiente de este nodo y luego se ordenan. Una vez se ha ejecutado este proceso la lógica de cada auto es actualizada para que pueda informar a cada entidad su posición en la pista.
4. **Saber si se puede jugar.** Es posible jugar si ya el juego empezó, no está en pause y además no se ha acabado. Esta funcionalidad es encuestada a menudo para saber si es posible ejecutar simulaciones y otros procesos a nivel de juego.

5. **Pausar el Juego.** Para poner la pausa en el juego basta que al menos un vehículo esté en pausa.
6. **Posición inicial de los autos.** La posición inicial de los autos está dada por un leve procesamiento de la posición que marca el nodo de inicio del circuito.
7. **Cuándo termina el Juego.** Para que el juego se termine hay que esperar a que todos los jugadores completen la carrera.

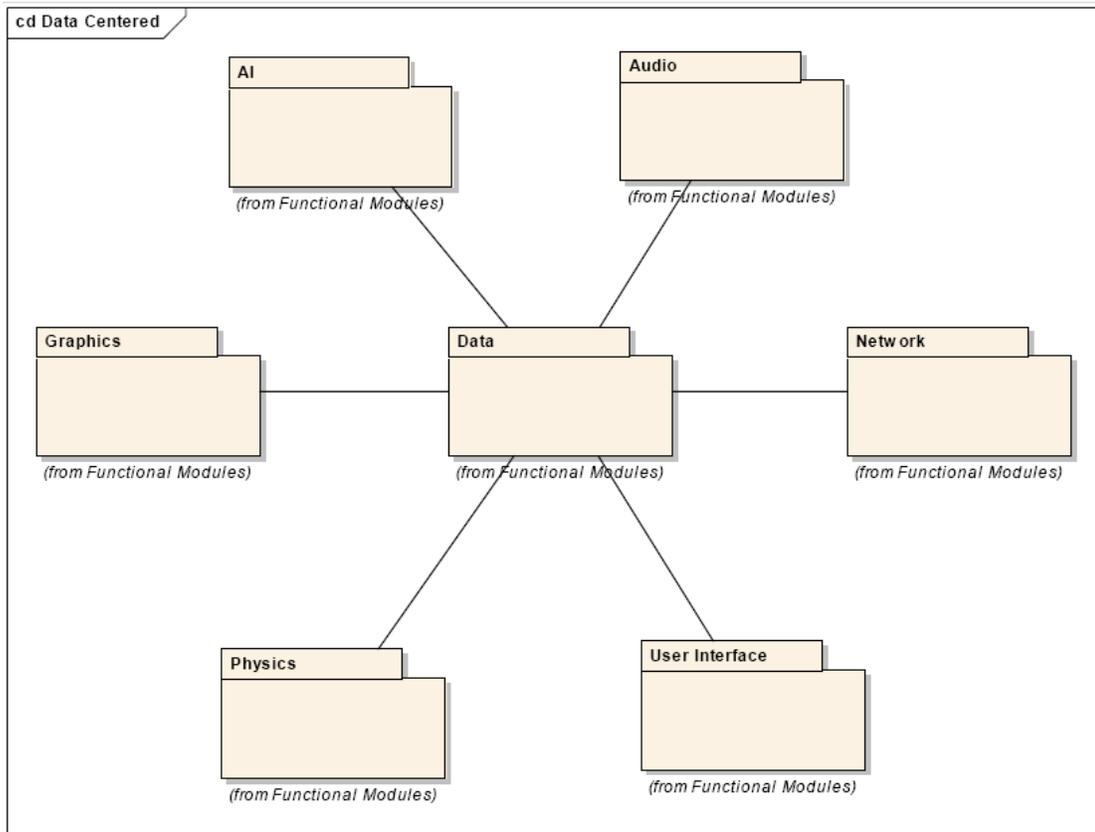
Otros datos importantes que brinda la Lógica del Juego son: el máximo de vueltas necesarias para terminar la carrera, el tiempo máximo que se necesita para completar las vueltas, el tiempo inicial para el conteo del principio del juego y la cantidad de entidades que están en juego.

### **2.2.2. Informaciones que intercambia el módulo con otros módulos del videojuego.**

Como se expresaba anteriormente, la relación que se establece entre el módulo de lógica y los demás componentes del videojuego se rige por un enfoque centrado en los datos.

Este estilo arquitectónico es utilizado para crear una integrabilidad a nivel de datos. Los módulos funcionales tienen un bajo acoplamiento entre sí, hecho que con frecuencia repercute en el rendimiento. La estrategia centrada en los datos minimiza muchos de los riesgos identificados en la arquitectura en capas. Primero, los módulos no tienen una interacción directa con ninguno de los otros, mitigando el asunto de la repercusión de cambios [1]. La Figura 9 explica gráficamente lo anterior.

Como se ha dicho previamente, los datos o informaciones que brinda este módulo al resto, son calculados a partir de las informaciones adquiridas de la pista de carrera y la complejidad de los cálculos varía en dependencia de lo que se quiera obtener. Algunos datos se consiguen fácilmente, como por ejemplo: la información de la posición actual, la velocidad y la dirección del vehículo. Otros como: la última posición correcta del auto, la dirección del tramo siguiente por donde debe transitar el vehículo y el lugar que ocupa el auto en la carrera, requieren de un análisis más complejo. Pero en cualquiera de los casos, estas complicaciones son transparentes al resto de los elementos del juego.



**Figura 9.** *Arquitectura Centrada en los Datos.*

La Figura 10 muestra el esquema de intercambio de datos entre componentes del videojuego y el módulo de lógica.



**Figura 10.** *Esquema de intercambio de información entre módulos.*

Entre la información que se intercambia se encuentran las siguientes:

1. Las *Entidades del Servidor* operan a nivel de auto y necesitan intercambiar las siguientes informaciones lógicas: saber si se puede jugar, si acabó el juego, además actualiza la velocidad, dirección, traslación para que la Lógica del Auto pueda hacer sus operaciones.

2. Los *Agentes Inteligentes* como tipo especial de entidades del servidor, reciben las mismas informaciones que estos, pero además necesitan saber: la distancia al centro de la pista, el ancho de la misma, cuándo hay una curva y el sentido de la trayectoria.

3. La *Física* como elemento fundamental de un auto, recibe informaciones lógicas a nivel de auto como son: la última posición correcta del auto, su dirección, pero además necesita informaciones físicas de la pista, que permitan simulaciones más realistas.

4. El *Servidor* utiliza informaciones de lógica a nivel de juego como las siguientes: si se puede jugar el juego, si se inició del juego, la posición inicial de juego y si finalizó el juego.

5. Las *Entidades del Cliente* utilizan información lógica a nivel de auto también, lo que se utiliza aquí es el estado lógico del juego, que involucra gran cantidad de informaciones como: si el jugador está en el sentido correcto de la trayectoria, si está en pausa, la flecha de dirección, la cantidad de vueltas recorridas, la posición que ocupa en la carrera y las puntuaciones.

6. Los *Modelos* por su parte reciben la posición dónde deben dibujarse.

### **2.2.3. Aportes Fundamentales**

Dentro de los aportes fundamentales del presente trabajo a la comunidad de desarrollo de videojuegos, se encuentra la creación de un módulo que controla la lógica del videojuego Rápido y Curioso; desarrollado con una especie de arquitectura tipo sistema de sistemas (según definiciones de [1]) y con un enfoque centrado en los datos para la interacción con el resto de los componentes.

Dando como resultado un Editor de Pistas, como sistema aparte al juego en sí, pero con funciones bien definidas para la generación de informaciones de la pista, la partida y el entorno.

Durante el desarrollo del Editor de Pistas se tomaron varias decisiones, que influyeron en el desempeño del videojuego en su totalidad y aunque algunas parezcan insignificantes y obvias, ahorraron gran cantidad de quebraderos de cabeza. A continuación se detalla una pequeña selección de ellas, con una breve descripción del por qué de su elección:

- La forma esférica de los nodos del grafo de camino, responde a que la esfera<sup>10</sup> es el sólido por excelencia para modelar un punto crítico de la pista. Debido a que los autos en su trayectoria siempre tienen que tener actualizado el último lugar significativo de la pista por donde pasaron, otras formas geométricas como los cubos o pirámides no garantizan una completa exactitud y dan cobertura a que el conductor se salte nodos, hecho que incide negativamente en los cálculos realizados por las interfaces lógicas, un ejemplo de esto se aprecia en las curvas.
- Los nodos son enlazados mediante segmentos dirigidos, debido a que los segmentos son los que definen los tramos de la pista, y por ende, del circuito de carrera; almacenando informaciones que pueden representar mejoras desde el punto de vista físico, visual e incluso de comportamiento. La forma recta permite una buena aproximación para hallar la curvatura de la pista, información requerida para saber dónde hay una curva, el ángulo de la misma, y hacer más exacto el cálculo de los errores de trayectoria, de dirección del vehículo, última posición correcta, entre otros.
- El hecho de que los circuitos de carrera son creados manual y no automáticamente, se debe a que los diseñadores del juego son quienes saben cuáles serán los circuitos de la carrera. Un enfoque automatizado generaría muchos circuitos sin sentido para el juego y sería en extremo complejo su depuración, alejándose demasiado de los límites de tiempo, alcance y costo del proyecto.
- La utilización de splines<sup>11</sup> para la exploración automática de los circuitos. Es producto a la simplicidad de la representación y la facilidad de cómputo que brinda la interpolación o ajuste de curvas mediante splines. Esto permite de una manera sencilla realizar el recorrido de un auto por el circuito evitando giros bruscos en las curvas de la trayectoria, producto a que las curvas de la pista son discretizadas por sucesiones de segmentos.
- El formato del fichero del grafo de camino no es estándar, porque las particularidades del grafo de camino, la forma de los nodos y aristas, el pre procesamiento necesario de los datos exigían una solución adecuada y dentro de los límites de alcance y tiempo con que contaba el proyecto, cosa que no hubiera sido posible con la utilización de otros formatos de fichero como 3DS, ASE, COLLADA, GRAPHML, entre otros.

---

<sup>10</sup> **Esfera:** Cuerpo sólido limitado por una superficie curva cuyos puntos equidistan de otro interior llamado centro de la esfera.

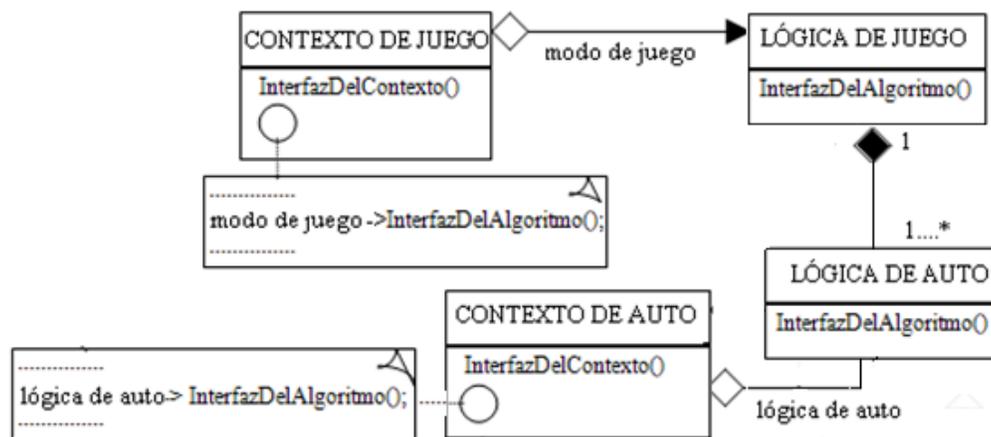
<sup>11</sup> **Spline:** es una curva definida en porciones mediante polinomios, en este caso los polinomios son lineales porque están definidos a partir de la ecuación de la recta que denota cada enlace.

Otro resultado importante fue la desagregación de la lógica en dos niveles de abstracción, para garantizar el bajo acoplamiento y alta cohesión de la solución. Para más detalles ver Capítulo 9 de [5].

Además se logró obtener una interfaz de lógica genérica para cualquier videojuego de carrera de autos. Amparada por una arquitectura base que puede ser extendida para soportar varios modos de juego.

### Ejemplo de extensión de la Arquitectura Base del Módulo de Lógica para soportar varios Modos de Juego.

La Figura 11 muestra la arquitectura base de los componentes de lógica, que como se dijo anteriormente está formada por dos niveles de abstracción, cada uno de los cuales ofrece interfaces que son utilizadas en el resto de los módulos según su contexto.



**Figura 11.** Arquitectura Base de las Interfaces de Lógica del módulo.

Extender la arquitectura base para que soporte varios modos de juego es posible utilizando el patrón Estrategia<sup>12</sup> (ver Figura 12), sobre los dos niveles de abstracción del módulo actual. Es decir, convertir la Lógica del Auto en una jerarquía de Lógicas de Auto especializadas y la Lógica del Juego en otra jerarquía de modos de juego especializados (ver Figura 13).

<sup>12</sup> **Patrón Estrategia/Strategy:** Patrón de comportamiento a nivel de objetos, tiene como propósito definir una familia de algoritmos encapsulando por separado cada uno de ellos y haciéndolos, por tanto, intercambiables. Esto permite a los algoritmos variar con independencia de los clientes que los usan [12].

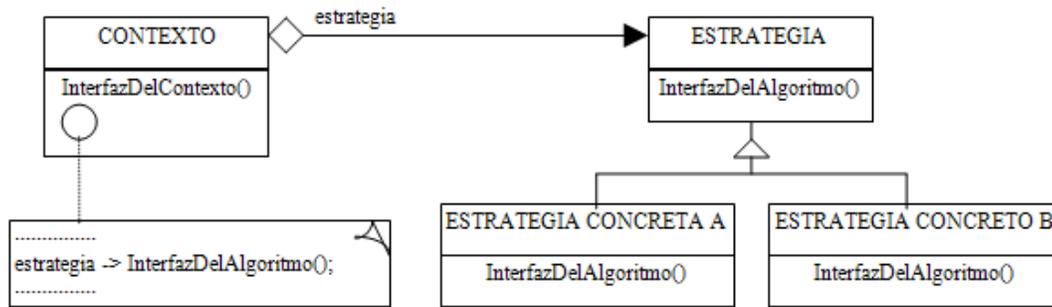


Figura 12. Patrón Estrategia. Tomado de (GoF, 1994).

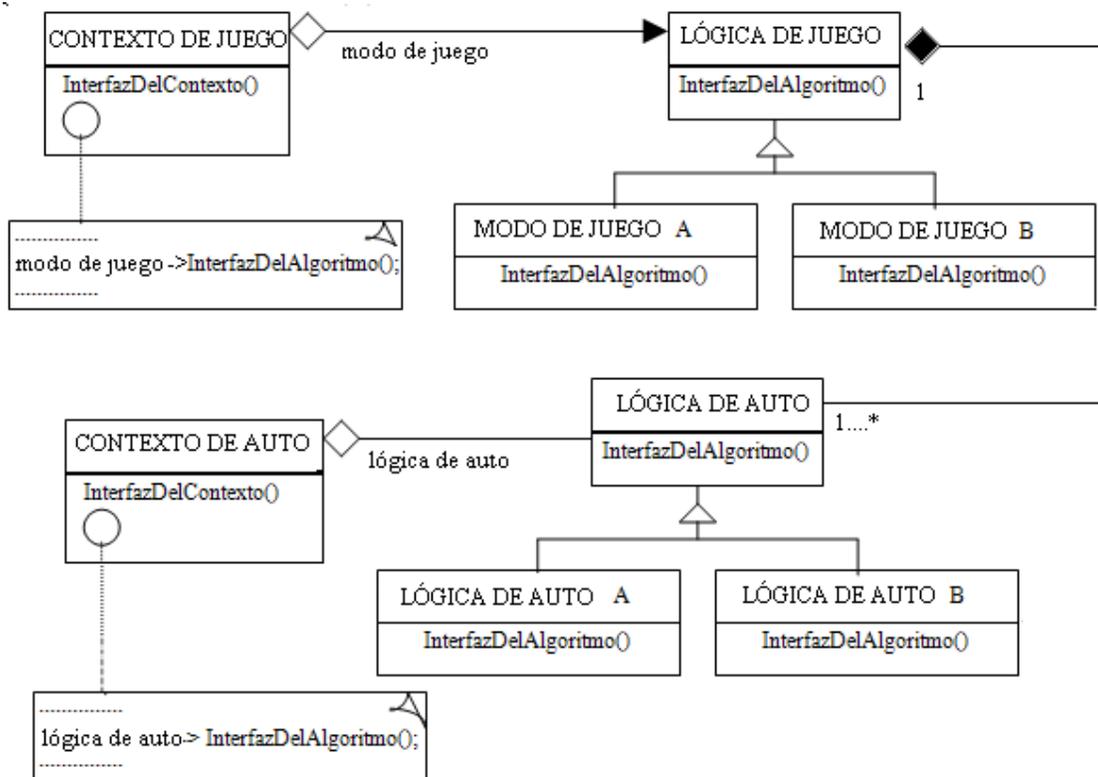
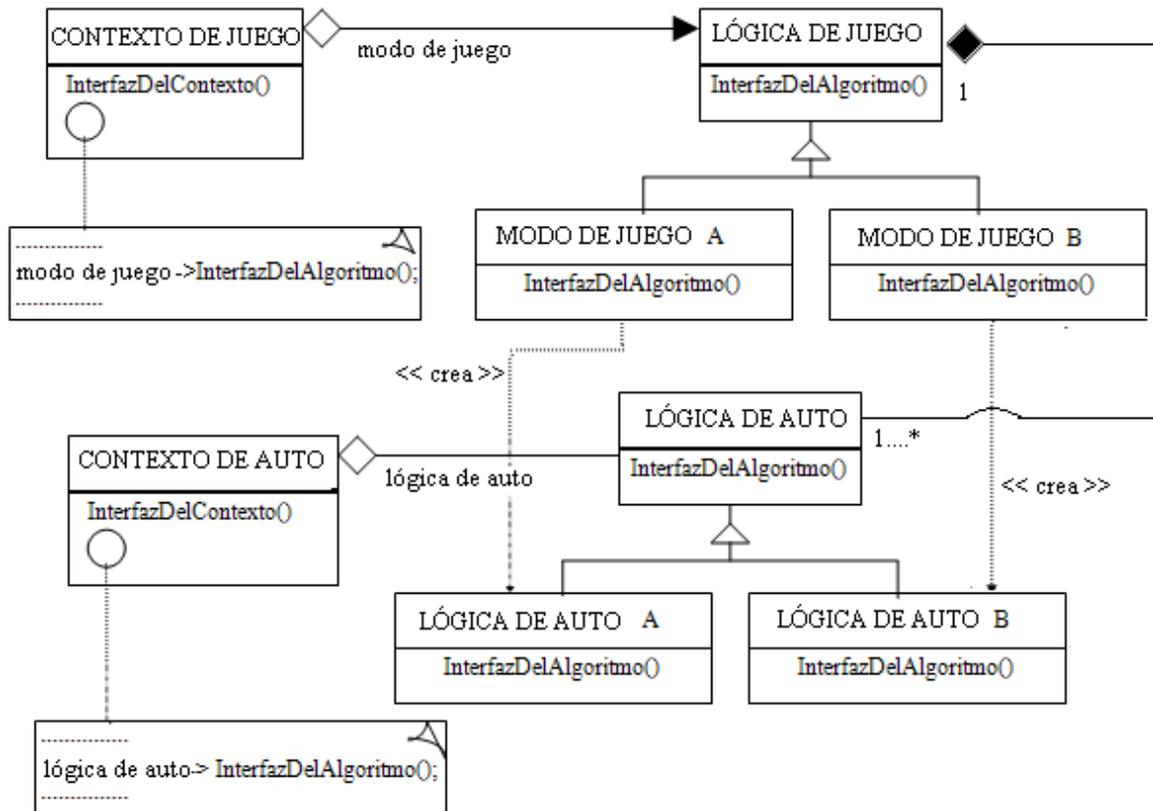


Figura 13. Aplicación del Patrón Estrategia. Jerarquías de Lógicas de Juego y de Auto.

El problema que surge con este enfoque, está en cómo conocer las instancias concretas de Lógica de Auto a crear por cada Modo de Juego. Para solucionar este inconveniente el patrón Constructor<sup>13</sup> ocupa el papel protagónico. Hay que implementar la Lógica de Juego abstracta como un constructor abstracto de Lógicas de Auto, y cada Modo de Juego será un constructor

<sup>13</sup> **Patrón Constructor/Builder:** Patrón de creación a nivel de objetos, que tiene como propósito separar la construcción y la representación de un objeto complejo, para así permitir que el mismo proceso de construcción sirva para crear diferentes representaciones [12].

concreto que construye las instancias concretas de Lógica de Auto que necesita. La Figura 14 ilustra la Arquitectura base extendida para soportar varios modos de juego.

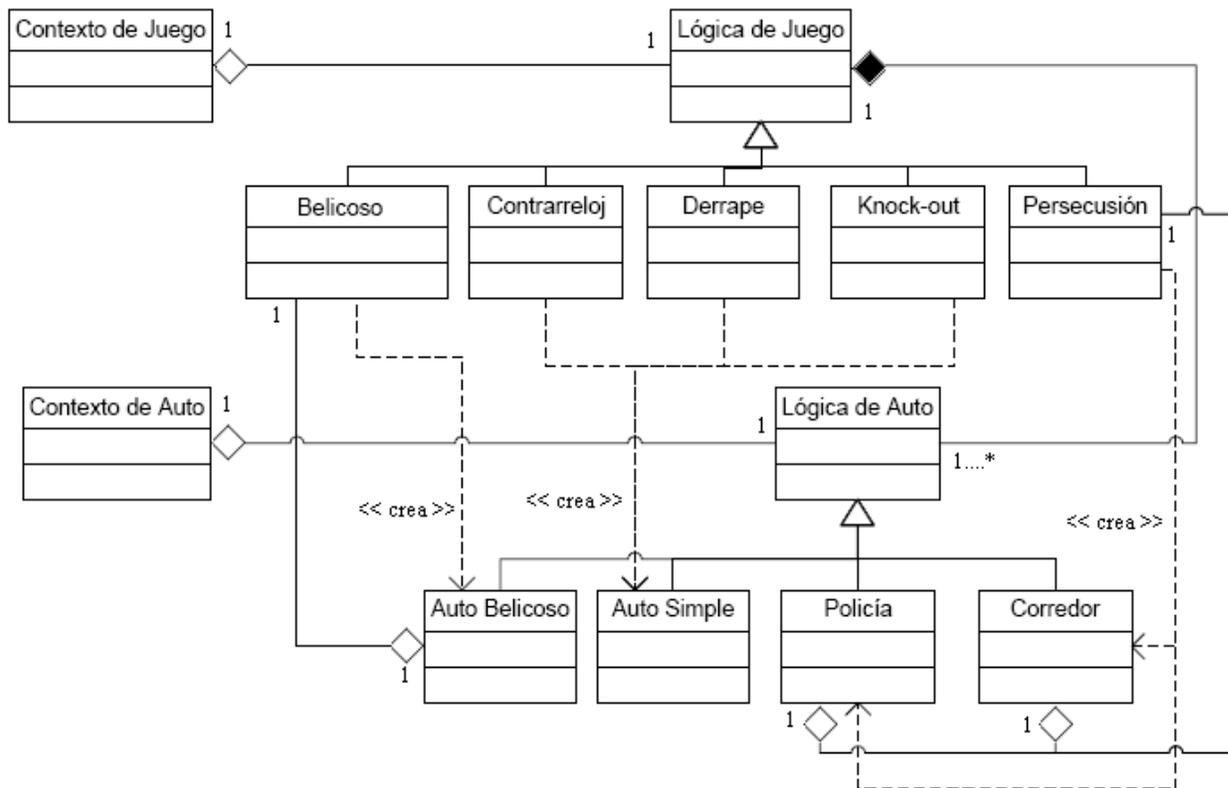


**Figura 14.** Arquitectura base extendida para soportar varios modos de juego.

### Aplicación de la Arquitectura Base Extendida a Videojuegos Conocidos.

A continuación se muestra cómo quedaría la arquitectura base extendida de los componentes lógicos del módulo de lógica de Rápido y Curioso, para lograr un comportamiento similar a videojuegos conocidos como: Need for Speed, Mario Kart y Revolt. Estos videojuegos cuentan con varias modalidades que se pueden resumir en las siguientes: contrarreloj, derrape, persecución, knock-out<sup>14</sup> y belicoso. Para modelarlos basta con extender la arquitectura base añadiendo una jerarquía de modos de juego, por cada uno de los anteriormente mencionados; y una jerarquía de lógicas de auto que represente las distintas maneras de jugar los modos de juego (Ver Figura 15).

<sup>14</sup> **Knock-out:** Se va eliminando el último corredor hasta que sólo queden dos.



**Figura 15.** Arquitectura base adaptada para soportar algunas modalidades del NFS, Mario Kart y Revolt.

En estos enfoques la idea es mantener la Lógica de Juego y de Auto sin modificar, sólo habría que poner los métodos como virtuales para que puedan ser redefinidos en las clases derivadas. En la Lógica de Juego, el método para adicionar las lógicas de auto se pondría virtual puro, para respetar lo planteado en el apartado anterior, de convertirla en un constructor abstracto de Lógicas de Auto. Recayendo la responsabilidad de creación de instancias concretas de lógicas de auto, en los modos de juego derivados.

La Lógica de Auto se debe mantener intacta en Auto Simple, para poder modelar el comportamiento actual que tienen los autos de Rápido y Curioso. Sin embargo, tanto en Policía, como en Corredor habría que redefinir el comportamiento, porque según el modo de juego, el policía debe perseguir a los corredores e intentar arrestarlos, y estos últimos evitar que eso ocurra. Aquí es necesaria una relación de agregación con el modo de juego Persecución porque ambos tipos de Lógica de Auto necesitan información adicional de la partida (ver Figura 15), como la posición del Policía o Corredor más cercano, según sea el caso, para informarle al jugador (mensaje en pantalla) y este obre en consecuencia.

En caso de que los vehículos sean autónomos, no es necesario desarrollar el comportamiento del policía (perseguir) y el de corredor (huir) en el Módulo de Inteligencia Artificial. Ambos comportamientos pueden ser modelados fácilmente sin necesidad de añadir

una programación profunda en el modo de juego Persecución. En el caso de perseguir, es suficiente con que el auto con el rol de policía intente sobrepasar a alguno de los corredores, en caso de que esto ocurra el juego termina, o se le pone una multa. En el caso del corredor, cuando note la presencia del policía debe evitar ser arrestado o que le pongan la multa, esto se logra evitando que lo alcancen. Ambos comportamientos se desprenden de la manera natural con que se comportan los agentes inteligentes, que siempre intentan sobrepasar a los contrarios, y evitar que los sobrepasen –están intentando ganar la carrera-, para más información en cuanto a los agentes inteligentes ver capítulo 10 de [5].

El modo de juego Knock-out utiliza los autos simples, porque sólo necesita ir eliminando el último en cada vuelta hasta que queden dos. Esto se logra mediante una funcionalidad implementada en la Lógica del Juego, que halla la localización de cada jugador y se le informa. De manera que se consigue crear un ambiente más emotivo porque el usuario al conocer su posición, sabe que será eliminado y se preocupa porque esto no suceda. Sólo habría que alterar en este modo de carrera el ciclo central de la lógica, añadiendo la llamada al método eliminar Lógica de Auto, antes de actualizar el número de vuelta o cada un intervalo de tiempo que se defina.

En el caso de juego Contrarreloj, se utilizaría autos simples, y se mantendría el comportamiento de sus clases padres, para que todo se comporte igual, porque este es el modo de juego que actualmente está implementado.

Para el Derrape sólo habría que alterar la información física de la pista mediante el Editor de Pistas, en las zonas que se definan. De manera que al enviarse el mensaje de estado de juego, el módulo de física actúe en consecuencia dando lugar al derrape. En este modo de carrera se controlarían informaciones como: quién tiene el mejor derrape, felicitaciones a los mismos, un score mostrando los puntos acumulados en cada derrape y otras por el estilo.

El Mario Kart y el Revolt se asemejan al modo de juego denominado como Belicoso. Este modo de juego no se aleja mucho del modo Contrarreloj, porque también se trata de una carrera contra el tiempo; la diferencia radica en que cada jugador es un tanto belicoso porque intenta por todos los medios de que sus adversarios no lo alcancen; aunque para lograrlo tenga que lanzarle objetos, dispararle, o regar aceite en la pista. En este modo de juego, el auto también debe obtener información sobre sus adversarios, como por ejemplo: si hay alguien cerca, entonces ejecuta una determinada acción, acción que se envía a través de los mensajes de estado por la red. En cualquier caso, cada objeto lanzado contra los adversarios será un modelo con textura y propiedades físicas determinadas, que lo harán funcionar de forma natural basándose en el módulo de física, incluyendo las colisiones con los demás vehículos.

Esto ha sido una breve demostración de las muchas variantes en las que puede ser extendida la arquitectura base de los componentes lógicos del videojuego Rápido y Curioso.

### 3. Conclusiones

La posibilidad de disponer de un módulo de lógica es vital para cualquier tipo de videojuego. Para los videojuegos de carreras de autos, es posible implementar la lógica de una manera desacoplada al resto de los módulos, brindándole a estos una interfaz que puede ser común para cualquier videojuego de carrera de autos.

Tanto es así, que es posible modelar el guión y argumento de muchos de los videojuegos de carreras de auto más conocidos como: Need for Speed, Revolt y Mario Kart; modificando el módulo de lógica propuesto, sin necesidad de alterar las interfaces que le brinda al resto de los componentes del videojuego, sin que estos cambios los afecten.

La información del entorno puede ser recuperada efectivamente mediante un Editor de Pistas. Los grafos de camino obtenidos ofrecen una abstracción bastante acertada de los circuitos de carrera. Al utilizar nodos esféricos se garantiza una exactitud elevada al obtener informaciones como el centro y ancho de la pista en cualquier parte de la misma, el tipo de los nodos permite conocer detalles importantes del sitio en que están ubicados. Los enlaces dirigidos con forma recta, discretizan correctamente los tramos de la pista y la trayectoria que debe ser recorrida en el tramo que representan.

Se quiere destacar lo novedoso de varias de las decisiones más importantes tomadas en este proyecto, como son: la división de la lógica en varios niveles de abstracción, la creación de una interfaz común a cualquier videojuego de conducción de automóviles.

El componente de lógica se diseñó como una arquitectura lógica base, pensada para ser ampliada mediante patrones de comportamiento y de construcción como el patrón estrategia y constructor. Esto posibilita que se puedan agregar nuevas funcionalidades en versiones posteriores sin hacer grandes modificaciones, en vistas a confeccionar otros videojuegos reutilizando el mayor número de componentes.

## 4. Recomendaciones

1. Añadir a los enlaces entre nodos del Editor de Pistas, soporte para almacenar más informaciones del tramo de la pista que representan.

Estas informaciones pueden ser:

- a) Físicas: el coeficiente de rozamiento, textura de la superficie, pendiente, ángulo de peralte, entre otros elementos. Estos datos serían procesados por el módulo de física.
  - b) Ambientales: niebla, lluvia, nieve. Estas informaciones pueden ir acompañadas de la intensidad del fenómeno. Estos datos se enviarían a los clientes incluidos en los mensajes de estado lógico, para invocar un módulo de efectos especiales en caso de existir, que modele dichos efectos naturales, con la intensidad especificada, esto modelaría efectos como lluvia tenue, media y abundante.
2. Extender el módulo de lógica para modelar varios modos de juego.

## 5. Glosario de Términos

**Circuito de Carreras:** Es un camino cerrado utilizado en ciertas modalidades de automovilismo y motociclismo. Los circuitos se distinguen de los tramos de rally y de las rectas usadas en los arranques en que el camino se recorre en círculos varias veces. Generalmente, el objetivo de una competencia en circuito es dar la mayor cantidad de vueltas al circuito en el menor tiempo posible [18].

**Componente:** O módulo, es todo aquel recurso desarrollado para un fin concreto y que puede formar solo o junto con otros, un entorno funcional requerido por cualquier proceso predefinido. Son independientes entre ellos, y tienen su propia estructura e implementación.

**Diseño del juego:** Es lo que determina la forma del juego, las decisiones que los jugadores serán capaces de tomar en el juego y qué ramificaciones esas elecciones tendrán en el resto del juego [4]. Además el diseño del juego establece cuáles serán los criterios de ganar o perder, cómo el usuario será capaz de controlar el juego, qué informaciones el juego deberá intercambiar con él, y establece cuán difícil será el juego. En resumen, el diseño del juego determina todos los detalles de cómo el juego funcionará [6].

**Diseñador del juego:** Es la persona que diseña el juego, que establece la forma y la naturaleza del juego [4].

**Discretización:** Es la acción de organizar con mayor eficiencia la distribución de los objetos en el mundo, de asignar diferentes propiedades a los polígonos que componen el mundo discreto. Por ejemplo, en un juego de coches podemos asignar una propiedad a cada polígono para representar la fricción del suelo, se puede tener un control del sistema de carreteras mediante un grafo de caminos, etc.

**Documento de Diseño:** Describe el concepto general del juego, el público a quien está dirigido, qué se hace en el juego: entorno en que se desarrolla el juego, reglas, objetivos, personajes, roles de los personajes, etc. Además se definen las interfaces con las que se interactúa para jugar, los controles, caracteres, niveles. En resumen, todo lo que el equipo de desarrollo necesita saber sobre el diseño del juego. Los artistas lo utilizan para situar las interfaces que reflejan lo que fue diseñado, los programadores lo usan para definir los módulos de software para esas características, los editores de niveles para entender cómo sus niveles se ajustan a la historia, el productor para generar planificaciones financieras y un calendario más exactos, y los aseguradores de la calidad para desarrollar los planes de prueba [4].

**Editor de Niveles:** Se refiere a las herramientas que se usan para crear los entornos. Estas herramientas se conocen también como Editores de Mapas o de Escenarios [5].

**Editor de Pistas:** Es la herramienta utilizada para realizar el trazado de los circuitos de las pistas de carreras en los videojuegos, mediante la cual el diseñador de pistas de carreras define el recorrido construyendo de forma manual un grafo de caminos con la trayectoria a seguir por los competidores, también puede definir tipos de terrenos en partes de la pista, variándole parámetros tales como fuerza de rozamiento a los tramos de la pista; de esta forma se consigue simular hielo, arena, entre otros, todo es cuestión de la imaginación del usuario y de las posibilidades que brinde el Editor de Pistas utilizado [5].

**Enlace:** Segmento dirigido que relaciona cada par de nodos en el grafo de caminos.

**Guión de juego:** Ver Documento de Diseño.

**Grafos de camino dirigidos:** O simplemente Grafo de Caminos, se entiende en el contexto de este trabajo a los grafos que representan información de la trayectoria a seguir por los jugadores. Se dice dirigidos porque cada par de nodos del grafo están conectados mediante un enlace con un sentido bien definido, que permite en caso de que se violado mostrar un mensaje de error de Trayectoria Incorrecta.

**Lógica del videojuego:** Módulo del videojuego que se encarga de implementar el guión, es decir: el módulo que se encarga de garantizar que las reglas, objetivos y demás elementos del videojuego se respeten.

**Mundo:** También conocidos en realidad virtual como entornos virtuales, mundos virtuales o mapas y se refiere a la simulación de mundos o entornos, denominados virtuales, en los que el hombre interactúa con la máquina en entornos artificiales semejantes a la vida real.

**Modo de juego:** Una de las modalidades en que se puede presentar un juego.

**Nodo:** En informática, de forma muy general, un nodo es un punto de intersección o unión de varios elementos que confluyen en el mismo lugar. Junto con las aristas o enlaces constituye una parte fundamental de los grafos.

**Pista de Carrera:** Se refiere a la calle, camino o autopista donde se desarrollan competencias de carreras de cualquier tipo: de vehículos, personas o animales.

**Segmento dirigido:** Es un segmento cuyos puntos limitantes definen un sentido, que está determinado por la normalización de la resta del punto final y el inicial.

**Videojuego:** Es un programa informático creado para el entretenimiento. Se basa en la interacción entre una o varias personas y un aparato electrónico que ejecuta dicho programa. Estos dispositivos pueden ser un ordenador, una consola o simplemente un teléfono celular. En muchos casos, en los videojuegos se recrean entornos y situaciones virtuales, en los cuales el jugador puede, respetando determinadas reglas, controlar a los personajes o cualquier otro elemento de dicho entorno, para conseguir objetivos específicos.

**Videojuegos de Carreras de Autos:** También conocidos como Videojuegos de Conducción de Automóviles, son videojuegos en los que se pilotan diferentes vehículos, ya sean reales o ficticios, para ganar en diferentes carreras. Dentro de este apartado se pueden distinguir dos variantes, arcades y simuladores.

**Videojuegos de Simulación:** En este género se engloban todos los videojuegos que pretenden emular ante los jugadores vivencias y situaciones lo más realistas posible. Aunque esta definición es muy genérica, cabe remarcar que la diferencia entre un juego de simulación y otro radica en el nivel de detalle del entorno y del control que podemos ejercer sobre el mismo. Entre este género, podemos hablar de simulación de: coches, aviación, sociales, empresariales [9].

**Videojuegos de Arcade:** Se refiere a los videojuegos contenidos en máquinas recreativas que se pueden hallar en sitios como bares, locales comerciales y de recreo. Como curiosidad, en inglés, a las galerías comerciales se las llama "shopping arcade". Arcade proviene de la arquitectura de las galerías, formada en arcos, y en castellano se denominan arcadas o soportales. A las máquinas que estaban en estas galerías comerciales se les comenzó a llamar "Arcade machines". De estas máquinas recreativas se ha derivado el género "Arcade" para denominar a los videojuegos que popularizaron estas máquinas [9].

## 6. Referencias

1. **Plummer, J.**, *A Flexible and Expandable Architecture for Computer Games*. 2004, Arizona State University: Arizona. p. 412.
2. **Crooks, C.E.**, *Awesome 3D Game Development: No Programming Required*. 2004, Massachusetts: CHARLES RIVER MEDIA, INC.
3. **Freeman, D.**, *Creating Emotions in Games. The Craft and Art of Emotioneering*. 2003: New Riders Publishing. 576.
4. **Rouse, R.**, *Game Design Theory and Practice*. 2005, Texas: Wordware Publishing.
5. **Nuñez, R.K. and M.A.M. Rodríguez**, *Informe Final del Proyecto "Juegos de Consola"*. 2009, UCI: Habana, Cuba.
6. **Fullerton, T., C. Swain, and S. Hoffman**, *Game Design Workshop-Designing, Prototyping, and Playtesting Games*. 2004, San Francisco: CMP Books.
7. **Miller, C.H.**, *Digital Storytelling. A Creator Guide to Interactive Entertainment*. 2004, Burlingong: Elsevier, Inc.
8. **Sheldon, L.**, *Character Development and Storytelling for Games*. 2004, Boston: Thomson Course Technology PTR.
9. **Gavaldà, J.D.i. and H.T. Navarro**, *Fundamentos y Programación de Videojuegos*. 2008, Cataluña: Universidad Oberta de Cataluña.
10. **Rucker, R.**, *Software Engineering and Computer Games*. 2003, Dorchester: Adison Wesley.
11. **Flynt, J.P. and O. Salem**, *Software Engineering for Game Developers*. 2005, Boston: Thomson Course Technology PTR.
12. **GoF**, *Design Patterns: Elements of Reusable Object-Oriented Software*. 1994, Oregon.
13. **BinSubaih, A., S. Maddock, and D. Romano**. *Game Logic Portability*. In *ACM SIGCHI International Conference on Advances in computer entertainment technology*. 2005. Valencia, Spain: ACM.
14. **Wikipedia**. *Need for Speed*. 2006 [cited 2010 2010/02/14]; Available from: [http://es.wikipedia.org/wiki/Need\\_for\\_Speed](http://es.wikipedia.org/wiki/Need_for_Speed).

15. **Wikipedia**. *Grand Theft Auto*. 2006 [cited 2010 2010/02/14]; Available from: [http://es.wikipedia.org/wiki/Grand\\_Theft\\_Auto](http://es.wikipedia.org/wiki/Grand_Theft_Auto).

16. **Wikipedia**. *Mario Kart (Serie)*. 2006 [cited 2010 2010/05/29]; Available from: [http://es.wikipedia.org/wiki/Mario\\_Kart\\_%28saga%29](http://es.wikipedia.org/wiki/Mario_Kart_%28saga%29).

17. **Abandonia**. *Revolt*. 2006 [cited 2010 2010/05/29]; Available from: <http://www.abandonia.com/es/games/15259/Re-Volt.html>.

18. **Wikipedia**. *Circuitos de Carrera*. 2006 [cited 2010 2010/02/14]; Available from: [http://es.wikipedia.org/wiki/Circuito\\_de\\_carreras](http://es.wikipedia.org/wiki/Circuito_de_carreras).