

UNIVERSIDAD AUTÓNOMA METROPOLITANA  
UNIDAD AZCAPOTZALCO  
DIVISION DE CIENCIAS BASICAS E INGENIERIA

Licenciatura en Ingeniería en Computación

Proyecto Tecnológico

Desarrollo de un videojuego de Rol para dispositivos  
móviles con Android

Alumno: De la Cerda Bacab Adán Israel

Matrícula: 205305342

Trimestre 15P

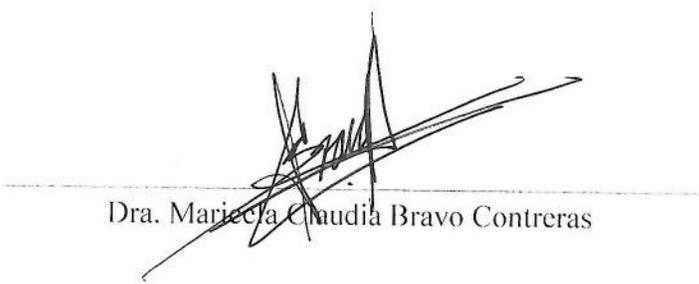
Julio 2015

Asesora:

Dra. Maricela Claudia Bravo Contreras  
Profesor Titular. Departamento de Sistemas

---

Yo, Maricela Claudia Bravo Contreras, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Dra. Maricela Claudia Bravo Contreras

Yo, Adán Israel de la Cerda Bacab, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco



Adán Israel de la Cerda Bacab

## Tabla de Contenido

RESUMEN .....	5
INTRODUCCIÓN .....	6
ANTECEDENTES .....	6
• Referencias Internas .....	6
• Referencias Externas .....	6
JUSTIFICACIÓN .....	7
OBJETIVO GENERAL .....	8
OBJETIVOS ESPECÌFICOS .....	8
DESARROLLO DEL PROYECTO .....	10
DESCRIPCIÓN TÉCNICA .....	10
RESULTADOS Y CONCLUSIONES .....	16
ENTREGABLES .....	17
Bibliografía .....	18

Figura 1 Escena de carga.....	8
Figura 2 Escena de juego al inicio del juego.....	9
Figura 3 Escena de menú principal .....	9
Figura 4 Menu de acciones de personaje jugador .....	10
Figura 5 Representación básica del mapa de juego.....	11
Figura 6 Maquina de estados finitos para el comportamiento del NPC .....	13
Figura 7 Ejecucion algoritmo A* (antes).....	14
Figura 8 Ejecucion Algoritmo A* (ruta generada).....	15
Figura 9 Ejecucion algoritmo A* (despues).....	15
Figura 10 Detección de colisiones .....	16

## RESUMEN

El presente proyecto se llevo a cabo satisfactoriamente utilizando *AndEngine*<sup>1</sup> el cual es una implementación de un motor de juego 2D de OpenGL<sup>2</sup> para dispositivos con sistema operativo *Android*<sup>3</sup>, el cual utiliza el lenguaje de programación Java y fue utilizado dentro del entorno de desarrollo integrado (IDE<sup>4</sup>) Eclipse

También fueron utilizados elementos gráficos (*sprites*<sup>5</sup>) para la representación del movimiento de los personajes controlados por el jugador o por los algoritmos de inteligencia artificial.

El mapa dentro del cual se desplazaran los personajes fue desarrollado utilizando *Tiled*<sup>6</sup> en formato *tmx*<sup>7</sup>

Se utilizó el algoritmo A\* para determinar el movimiento de los personajes (controlados por el jugador o de los *npc*'s<sup>8</sup>) dentro del juego detectando colisiones.

Una maquina de estados finitos fue utilizada para las estrategias de inteligencia artificial y así determinar su comportamiento dependiendo de las condiciones.

En base a lo desarrollado, el proyecto cumplió su objetivo, los diferentes módulos que se propusieron se desarrollaron correctamente.

Existe la posibilidad de llevar a cabo actualizaciones como pueden ser, aumento de mapas, añadir nuevos tipos de personajes, agregar tipos de ataque (magia ofensiva, magia curativa, etc.), entre otros.

---

<sup>1</sup> *Andengine* es un motor de juegos en 2D gratuito para la plataforma *Android*.

<sup>2</sup> Las bibliotecas gráficas OpenGL facilitan la comunicación con el hardware gráfico de nuestra máquina [8].

<sup>3</sup> *Android* es un sistema operativo móvil basado en Linux que permite desarrollar aplicaciones y juegos para dispositivos móviles en lenguaje Java. [1]

<sup>4</sup> *Integrated Development Environment*.

<sup>5</sup> Elemento visual que puede ser una única imagen o componerse de varias que se van alternando hasta crear una animación [9].

<sup>6</sup> Editor de mapas de propósito general que permite especificar cosas tan abstractas como, áreas de colisión o posiciones de objetos de mejora, etc. Toda la información la almacena en un formato estandarizado *tmx*[10]

<sup>7</sup> *TMX(Tile Map XM)* formato utilizado por *Tiled* para describir un mapa basado en baldosas. Puede describir el mapa con cualquier tamaño de baldosa. Cualquier cantidad de capas así como propiedades personalizadas para la mayoría de los elementos [11]

<sup>8</sup> *Npc* (Non Player Character) Se refiere a un personaje en un videojuego que es controlado por la inteligencia artificial.

## INTRODUCCIÓN

En este proyecto se desarrolló un videojuego de rol para dispositivos móviles con sistema operativo *Android*. El objetivo del juego es poner a prueba las capacidades del jugador para crear y aplicar una estrategia con la cual logre derrotar al enemigo.

La dinámica del videojuego está diseñada por turnos, en cada uno de estos el jugador puede mover a uno o a todos sus personajes, así como atacar a alguno de los personajes controlados por las estrategias de inteligencia artificial (IA) programadas en el juego. Las estrategias de IA pueden, en cada uno de sus turnos, realizar las mismas acciones que el jugador o realizar acciones que contra ataquen al jugador para vencerlo.

Cuando el jugador y la IA terminan sus turnos se repite el ciclo del turno. Si el jugador o la IA no les quedan ningún personaje para seguir combatiendo el combate se dará por terminado.

## ANTECEDENTES

- **Referencias Internas**

Actualmente existen los siguientes proyectos terminales concluidos:

1. Implementación de motor de juego para juegos tipo *Tower Defense* [2], el cual es un videojuego, aunque se diferencia en que dicho proyecto utiliza el lenguaje de programación C# y no está diseñado para dispositivos móviles.
2. Plataforma de juego programable para *Quoridor* [3], el cual es un videojuego, aunque en este caso no compite un jugador humano, sino, con un agente inteligente creado por el jugador en lenguaje C++.
3. *Programación de un war-game 2d para plataformas móviles y touchscreen con J2ME* [4], el cual se relaciona con el proyecto actual en que es también un videojuego para dispositivos móviles, aunque se diferencia en que dicho proyecto utiliza el lenguaje de programación J2ME<sup>9</sup>.

- **Referencias Externas**

En la actualidad existen diversas compañías que desarrollan videojuegos para la plataforma *Android* como lo son *Electronic Arts*, *GameHouse*, *Zeptolab*, *HalfBrick Studios*, *Rovio* (Creador del famoso juego *Angry Birds*), entre otros.

---

<sup>9</sup>*Java Platform, Micro Edition* (Plataforma Java edición micro) ofrece un entorno flexible y sólido para aplicaciones que se ejecutan en dispositivos móviles.

A diferencia de la mayoría de las aplicaciones que se distribuyen en *Google Play Store*<sup>10</sup> este proyecto es distribuido de forma gratuita en el sitio de *aptoide*<sup>11</sup> debido al costo de distribución en *Google Play Store*.

También existen videojuegos de rol similares como lo son:

- *Final Fantasy Tactics* de la compañía de videojuegos *SquareSoft*
- *Tactics Ogre* de la compañía de videojuegos *Atlus*
- *Disgaea* de la compañía de videojuegos *Nippon Ichi Software*

## JUSTIFICACIÓN

Un ingeniero en computación debe de tener en cuenta que su campo de trabajo no se limita a los sistemas computacionales.

En la actualidad existe una amplia gama de dispositivos en los que un ingeniero en computación puede aplicar los conocimientos adquiridos en el transcurso de sus estudios.

Entre los dispositivos que actualmente tienen un gran auge se encuentran los móviles, principalmente teléfonos inteligentes y tabletas electrónicas (*tablets*).

Mucho se ha hablado acerca de los efectos negativos de los videojuegos: muy pocos creen que los efectos positivos son más que los negativos.

Algunos de los aspectos positivos de los videojuegos en general son entre otros:

- Mayor coordinación ojo-mano y orientación espacial
- Mejoran el rendimiento intelectual, la percepción y la comprensión.
- La frecuencia de uso de videojuegos puede reducir la intensidad de otras conductas problemáticas [6].

El desarrollo de videojuegos en México es una industria incipiente. La industria de los videojuegos es de las pocas que no se doblegan ante la crisis económica mundial.

Actualmente, el desarrollo de juegos se realiza para dispositivos móviles y tanto la industria en nuestro país como el programador de videojuegos tienen un campo de desarrollo enorme. Andro Miralrio, Productor de Sabarasa México<sup>12</sup>, sugirió detonar la carrera de diseñador de videojuegos en las universidades del país, ya que conjuga el arte con la creatividad tecnológica [7].

Como consecuencia de los beneficios que se pueden obtener al jugar videojuegos y al gran crecimiento que ha tenido últimamente su desarrollo en plataformas móviles, se propone programar un videojuego de combate-estrategia por turnos.

Este proyecto debe de ser realizado por un ingeniero en computación ya que la complejidad y los conocimientos necesarios para el desarrollo del mismo se adquieren a través de los

---

<sup>10</sup> Tienda electrónica para adquirir contenido para dispositivos móviles con sistema operativo *android* (antes conocido como *android market*).

<sup>11</sup> Tienda electrónica independiente de aplicaciones para sistema operativo *Android*, que permite a los usuarios configurar y administrar su propia tienda de aplicaciones para *Android* de manera gratuita [5].

<sup>12</sup> Empresa pionera en el desarrollo de videojuegos de consola y computadora en Latinoamérica.

estudios de licenciatura, como son los proporcionados por la Inteligencia Artificial, el Diseño de algoritmos y los conocimientos matemáticos.

## OBJETIVO GENERAL

Programar un videojuego de rol por turnos también llamado RPG<sup>13</sup> en 2D para dispositivos móviles con Sistema Operativo *Android 2.3.3* o superior utilizando el motor de juego *andengine*.

## OBJETIVOS ESPECÍFICOS

Se crearon los módulos necesarios para el manejo de los recursos del sistema así como para la interacción entre ellos.

- Manejador de Escenas
  - Escena de Carga



Figura 1 Escena de carga

---

<sup>13</sup>*Role-PlayingGame* (juego de rol) es un género de videojuegos que usa elementos de los juegos de rol tradicionales. Estos últimos no son juegos electrónicos, se juega a ellos con dados, lápices y hojas de papel.

- Escena de Juego



Figura 2 Escena de juego al inicio del juego

- Escena de menú Principal



Figura 3 Escena de menú principal

- Escena *Splash*<sup>14</sup>

<sup>14</sup> Escena inicial de un videojuego en la cual, por lo general se muestra la empresa desarrolladora del videojuego.

- Escena menú de acciones



Figura 4 Menu de acciones de personaje jugador

- Manejador de Recursos
- Objetos
  - Personaje
    - Jugador
    - Enemigo
  - Matrices
- Algoritmo A\*(A estrella)

## DESARROLLO DEL PROYECTO

Para la realización de este videojuego se utilizó la más reciente versión del motor de juego *Andengine* Open GL ES 2

## DESCRIPCIÓN TÉCNICA

Para la realización de este videojuego de estrategia por turnos, se utilizó *andengine*, el cual es una implementación de un motor de juego 2D de OPENGL, para la plataforma *Android*.

Debido a las limitaciones de tiempo en la realización de este proyecto, las características gráficas de este videojuego están limitadas. Debido a que *andengine* está programado en Java, se utilizó este mismo lenguaje en el entorno de desarrollo (IDE) Eclipse.

Para iniciar el diseño del juego se tuvo que recopilar una serie de imágenes también llamadas *sprites* que nos sirvieron para representar cada uno de los elementos del mapa.

En la Figura 5 se muestra la representación básica del mapa del juego con algunos elementos del entorno utilizando *Tiled* y almacenado en formato *tmx*



Figura 5 Representación básica del mapa de juego

Como se mencionó anteriormente, se distribuyó el juego en módulos para facilitar el manejo de los recursos y elementos del mismo.

- **Manejador de Escenas:** Es el encargado de crear, establecer y desechar las diferentes escenas que utilizará el juego así como la interacción entre ellas (escena de carga, escena de juego, escena de menú principal, escena *splash* y escena menú de acciones)
- **Manejador de Recursos:** Es el encargado de administrar los recursos que utilizarán las diferentes escenas como son cámara, *HUD*<sup>15</sup>, texturas, fuentes, audio, música, mapa, matrices y algoritmos.
- **Objetos:** Este paquete contiene los siguientes módulos:
  - **Personaje:** Encargado de crear y administrar (mover, animar *sprite*, etc.) los diferentes personajes, ya sean, Jugador (controlados por el usuario) o Enemigo (controlados por la Inteligencia artificial).
  - **Matrices:** Encargado de administrar las diferentes matrices de control que utilizará el juego
    - **Matriz Entorno:** Contiene las posiciones dentro del mapa donde se encuentra algún obstáculo que impida el movimiento del personaje.
    - **Matriz Personajes:** Contiene las posiciones dentro del mapa donde se encuentra cada uno de los personajes.
    - **Matriz Colisión:** Contiene todas las posiciones ocupadas dentro del mapa, ya sea por objetos o por personajes.

<sup>15</sup> *HUD (Head up Display)*.- Área en la que se muestra las estadísticas vitales del personaje(en este caso se mostrará los HP)

- Algoritmo A\*(A estrella): Algoritmo encargado de encontrar una ruta aceptable entre dos puntos del mapa evitando los obstáculos (objetos o personajes).

## Algoritmos de Inteligencia Artificial

### Comportamiento de *npc*'s

Para definir el comportamiento de cada uno de los *npc*'s se utilizó una máquina de estados finitos la cual cuenta con los siguientes estados: avanzar, retroceder, ataque físico, defender y un estado inicial. Para poder cambiar de un estado a otro el *npc* tiene que evaluar diferentes variables que pueden ser, la distancia a la que se encuentra un personaje del jugador, HP<sup>16</sup> tanto de un personaje del jugador como del *npc*, la diferencia entre el número de *npc*'s disponibles y el número de personajes del jugador, etc. La máquina de estados finitos se encuentra representada en la siguiente figura

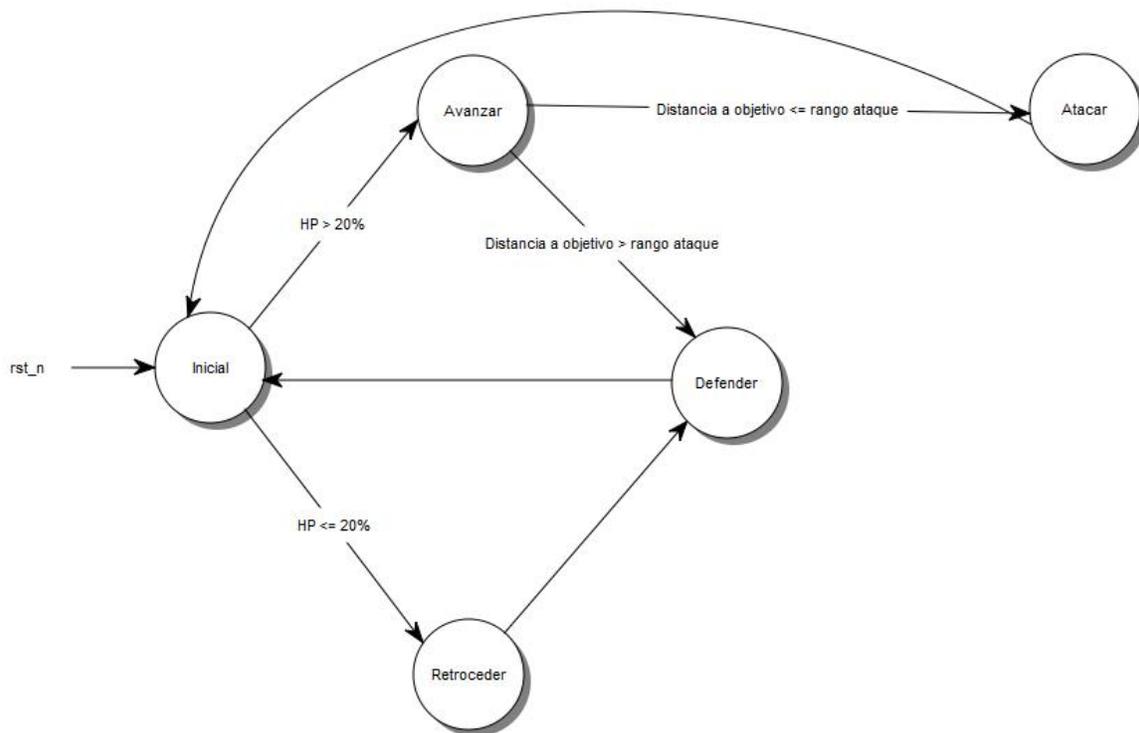


Figura 6 Máquina de estados finitos para el comportamiento del NPC

### Creación de rutas

Después de elegir el punto destino de desplazamiento de una unidad ya sea *npc* o algún personaje del jugador se utilizó el algoritmo A\* (A estrella), para determinar una ruta de desplazamiento aceptable evitando obstáculos. Cada una de las unidades tendrá un movimiento limitado dentro del mapa para evitar una exploración completa del mapa en un solo turno.

<sup>16</sup> *Hit Point*.- Representación de la cantidad de energía restante del personaje jugador, cuando este número llega a 0 se considera que el personaje ha sido eliminado.

La creación de la ruta de movimiento para las unidades depende de la ejecución del algoritmo A\* el cual necesita, los puntos de origen y destino en la matriz de mapa, para determinar la ruta de movimiento.

Cuando la posición de destino calculada por el algoritmo A\* está siendo ocupada (ya sea por un personaje, un enemigo o un obstáculo). El algoritmo busca la posición libre en las inmediaciones del destino la cual tenga la distancia Manhattan más corta desde el origen (busca la posición en sentido horario ARRIBA, DERECHA, ABAJO, IZQUIERDA).



Figura 7 Ejecucion algoritmo A\* (antes)

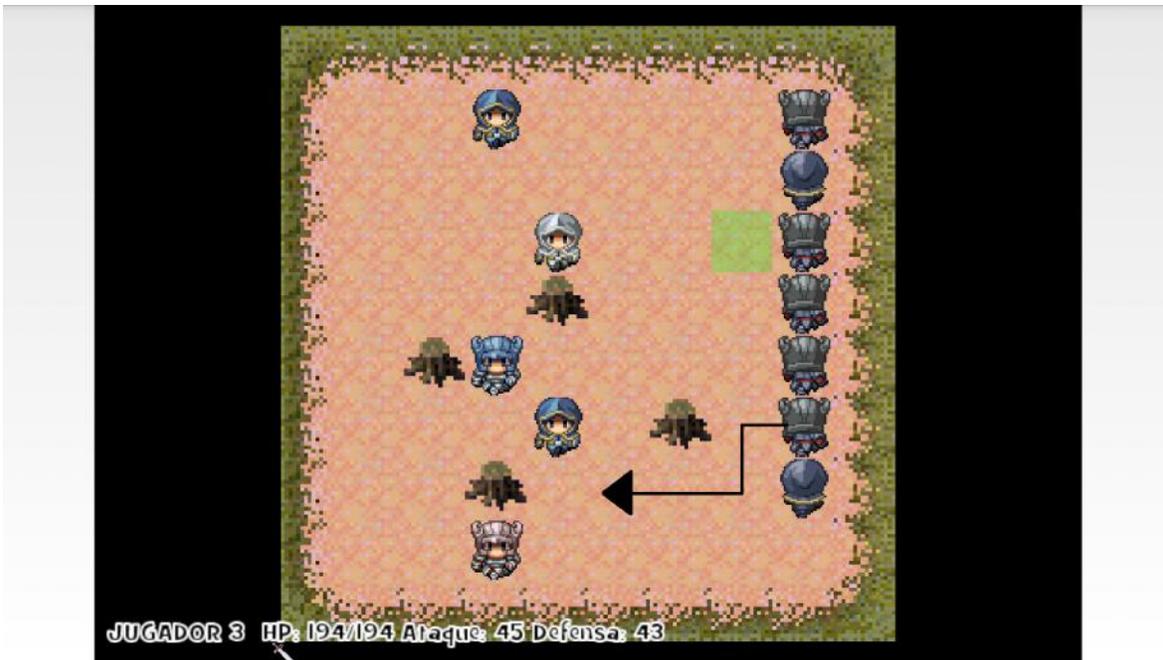


Figura 8 Ejecucion Algoritmo A\* (ruta generada)



Figura 9 Ejecucion algoritmo A\* (despues)

### Detección de colisiones

Debido a que todas las unidades están interactuando dentro del mismo mapa, es posible que, al elegir un punto destino de movimiento, este punto esté ocupado, ya sea por otro personaje (jugador o enemigo) o un elemento de entorno. Para evitar esto se creó la “matriz

de colisiones”, la cual indica las posiciones dentro del mapa hacia los que no es posible desplazarse.

Esta matriz es comprobada cada vez que se seleccione una posición dentro del mapa ocupada por algún obstáculo y cancela la ejecución del algoritmo A\* si es necesario.



Figura 10 Detección de colisiones

### Finalización del juego

El juego se dará por finalizado cuando se cumpla alguna de las siguientes condiciones

1. Todos los personajes enemigos son derrotados (Puntos de golpe reducidos a 0)
2. Todos los personajes jugadores son derrotados (Puntos de golpe reducidos a 0)

## RESULTADOS Y CONCLUSIONES

En base a lo desarrollado, el presente proyecto cumplió su objetivo, los diferentes módulos que se propusieron se desarrollaron correctamente.

El proyecto se encuentra libre de errores de compilación, sin embargo, como todo desarrollo de software es posible que presente algunos errores en la fase de ejecución,

dichos errores serán detectados y reportados por los usuarios que descarguen e instalen la aplicación en sus dispositivos para su futura corrección.

Existe la posibilidad de llevar a cabo actualizaciones como pueden ser, aumento de mapas, añadir nuevos tipos de personajes, agregar tipos de ataque (magia ofensiva, magia curativa, etc.), entre otros.

El juego será publicado en el sitio de *aptoide* (<http://www.aptoide.com>) y no en *play store* (<https://play.google.com>) como fue mencionado en la propuesta debido a que existe un cobro de 25.00 USD para poder publicar aplicaciones por parte de *Google*.

## ENTREGABLES

1. Código fuente comentado de la aplicación.
2. Archivos con los *sprites* y sonidos utilizados.
3. Reporte final.

Los entregables, en formato pdf, se entregarán en un CD.

## Bibliografía

1. Android Developers. [1 de Junio de 2015.] *Introduction to Android* [En línea]. [Disponible en:] <http://developer.android.com/guide/index.html>
2. Montaña A., O. X. "Implementación de motor de juego para juegos tipo Tower Defense". Proyecto Terminal de Ingeniería en Computación, Ciencias Básicas e Ingeniería, UAM Azcapotzalco, 2011.
3. Hernández H., F. A. y Hernández Piña, H. C. "Plataforma de juego programable para Quoridor". Proyecto Terminal de Ingeniería en Computación, Ciencias Básicas e Ingeniería, UAM Azcapotzalco, 2010.
4. Cabral G., L. M. "Programación de un war-game 2d para plataformas móviles y *touch screen* con J2ME". Proyecto Terminal de Ingeniería en Computación, Ciencias Básicas e Ingeniería, UAM Azcapotzalco, 2011.
5. Aptoide Own your Android market. [Citado el: 1 de Junio de 2015] *About us*[en línea] [Disponible en:] <http://www.aptoide.com/page/about>
6. Slideshare. [Citado el: 1 de Junio de 2015.] Coral V, M. *Videojuegos pros y contras*. [En línea] [Disponible en:] <http://www.slideshare.net/hsockmerlyn/videojuegos-pros-y-contras>.
7. El Universal. [Citado el: 1 de Junio de 2015.] *Impulsan videojuegos en México*. [En línea] [Disponible en:] <http://www.excelsior.com.mx/2011/12/04/dinero/791420>
8. División de Arquitectura de Computadores de la Universidad Rey Juan Carlos. [22 de Febrero de 2012.] Pastor, L., et. al. *Introducción a OPENGL* [En línea] [Disponible en:] <http://dac.escet.urjc.es/docencia/GV3D/DocGL1.pdf>.
9. Prieto M., M. J. *Desarrollo de juegos con J2ME*. AlfaOmega, 2005.
10. Tuts+. [Citado el: 1 de Junio de 2015] *Introduction to Tiled Map Editor: A Great, Platform-Agnostic Tool for Making Level Maps*[en línea] [Disponible en:] <http://gamedevelopment.tutsplus.com/tutorials/introduction-to-tiled-map-editor-a-great-platform-agnostic-tool-for-making-level-maps--gamedev-2838>
11. Tiled Documentation. [Citado el: 1 de Junio de 2015] *TMX Map Format* [en línea] [Disponible en:] <http://doc.mapeditor.org/reference/tmx-map-format/>