

# Universidad de las Ciencias Informáticas

Facultad 5 Realidad Virtual



## TÍTULO: ALGORITMO DE DISTRIBUCIÓN INTELIGENTE DE OBJETOS EN ESCENAS 2D.

Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas

**Autores:** *Faimaris González Ramírez*

*Mileiny de Armas Viera*

**Tutores:** *Tamara Martínez Labaut*

*Lázaro Campoalegre Viera*

Ciudad Habana

Junio 2009

### Datos de Contacto

#### **Tutora:**

Nombre y Apellidos: Ing. Tamara Martínez Labaut

Edad: 24

Ciudadanía: Cubana

Instituto: Universidad de las Ciencias Informáticas (UCI)

Título: Ingeniero Informático

Categoría Docente: Instructor Recién Graduado

E-mail: [tmartinez@uci.cu](mailto:tmartinez@uci.cu)

#### **Co-Tutor:**

Nombre y Apellidos: Ing. Lazaro Campoalegre Vera

Edad: 25

Ciudadanía: Cubana

Instituto: Universidad de las Ciencias Informáticas (UCI)

Título: Ingeniero Informático

Categoría Docente: Instructor Recién Graduado

E-mail: [lcampoalegre@uci.cu](mailto:lcampoalegre@uci.cu)

## Dedicatoria

*Quisiera encontrar las palabras que perdí o unir las que me quedan con las que quiero expresar, las más lindas, todas para mi abuelo, pero solo logre escribir las que encontré en un viejo baúl.*

*Dedicado a mi abuelo, mi lucero, mi orgullo, mi más grande inspiración en la vida, pero sobre todo por ser la persona que no supo que me dio los mejores ejemplo que podría darle a una nieta, gracias a ti aprendí a detestar el alcohol, a tener siempre una sonrisa y amar más que a nada, mi propia vida, porque sin ella no podría idolatrarte o adorar a mi familia, la que construiste para mí.*

*A mi mamá por ser mi más preciado tesoro.*

*Mileivys*

*A mi familia por hacer todo lo posible para que llegue este momento.*

*A mi novio por todo el amor y el apoyo que me ha dado.*

*A la Revolución por darme esta oportunidad.*

*Faimaris*

## Agradecimientos

*A Dios en primer lugar.*

*A la Revolución y al Comandante Fidel por darme la oportunidad de llegar a esta Universidad.*

*A la Facultad 5 porque siempre me sentí orgullosa de pertenecer a ella.*

*A mi mamá, por no dejarme flaquear en las tantas veces que lo intenté, por ser la fuerza que me inspira a ser alguien en la vida y porque por ella aprendí a levantarme ante los tropiezos que me ha impuesto el más bello de los caminos pedregosos, la vida.*

*A mi papá por ser mi sostén, por entenderme y extender su mano cuando lo necesite.*

*A mi hermano, por quererme y ser mi rayito de sol, para que tome de mí el mejor de los ejemplos.*

*A mi tía, por ser mi tía, casi una amiga de las que quedan pocas y le podría contar todo, gracias por existir y por ser casi mi segunda mamá.*

*A mi abuela Olga que adoro por encima de todo, y que porque la mencione en este lugar no deja de ser un eslabón más que importante en mi vida.*

*A mis tíos, a quien le debo por igual a todos, a tío Coribio porque gracias a él cumplí un sueño realidad.*

*A mis amigos, aquellos que conocí en la UCI y estuvieron en las malas y en las buenas (Alberto Pacheco, Wilker, Marcos, Yoandris, Fatina, Dayanis, Shirama y Mile).*

*A mis compañeros los que llegaron más tarde, un montón y más (Odalys, Familia, Daillet, Teudny, Joe) a ustedes a todos les debo algo, un consejo, un apoyo etc. También siéntanse incluidos los que siempre fueron mis eternos compañeros de aula, de Facultad o Universidad.*

*A mis profesores, aquellos de los que me llevo una enseñanza, un recuerdo o una magnífica lección.*

*A Millet, a quien me atrevo a tutear solo en este papel, por no apoyarme en las tantas veces que le pedí consejo para irme de la Universidad y por creer en el fondo que si podía llegar aquí, gracias por su apoyo.*

*A Brígida, por acogerme cuando pertenezco a su segunda generación de estudiantes en la UCF.*

*A mis tutores Tamara y Campo, a Coca y Omar por las tantas veces que los molestamos,*

*A mi compañera de tesis, por compartir esta tarea conmigo, a Ariagnis por quien tuve este tema de tesis.*

*A Alberto Abreu, mi director de preuniversitario, por guiar mis pasos hasta aquí cuando dude de mi futuro.*

*Y a quien su lugar hubiese sido de los primeros en un día como hoy. Gracias a ti por siempre, por siempre tenderme tu mano, tu abrigo y tus oídos.*

*Y en general, gracias, muchas gracias a todos aquellos que dudaron de mí, gracias a ustedes tuve fuerzas para vencer.*

*Mileirys*

*A la Revolución y al Comandante Fidel por haberme dado la oportunidad de superarme.*

*A mis padres que han hecho todo lo posible por que llegue a ser una profesional.*

*A mi hermano que me ha impulsado a convertirme en un ejemplo a seguir.*

*A mis abuelas por todo el amor que me han dado.*

*A mi novio que siempre ha estado a mi lado, apoyándome, aconsejándome, dándome mucho amor y ayudándome durante todo este tiempo.*

*A mis suegros que me han acogido como una hija y han soportado mis malcriadeces.*

*A Mila y Longo que me han hecho más alegres mis días.*

*A los profesores Coca y Omar por guiarnos y aconsejarnos durante el desarrollo de este trabajo.*

*A los tutores Tamara y Campo por su preocupación.*

*A los amigos que han estado dispuestos para lo que necesite.*

*Faimaris*

## Resumen

En el presente trabajo se realiza una guía de observación de los videojuegos del género objetos escondidos de mayor popularidad en la sociedad cubana, los cuales son conocidos como "Detectives". El propósito de dicha guía es detectar las carencias y dificultades que estos videojuegos presentan; los problemas detectados son similares a los del reciente videojuego desarrollado en la Universidad de las Ciencias Informáticas (UCI), en coordinación con el ICAIC titulado "Fernanda", el cual entra dentro de este género, pero con ciertas particularidades debido a que está orientado a niños. A partir de los problemas detectados se implementa un algoritmo para la solución de estos, el cual está explicado detalladamente en este trabajo, como también se muestran los resultados deseados que se obtienen al aplicarlo sobre un nivel del juego, para lo cual se realizó un pequeño demo.

El algoritmo finalmente resuelve problemas como la distribución estática y caótica de los objetos y la presencia de fondos con poca variedad de colores presentes en los juegos de este tipo, incluido "Fernanda".

**Palabras Claves: Algoritmo, videojuego.**

## Tabla de contenido

Dedicatoria.....	II
Agradecimientos.....	III
Resumen.....	V
Introducción .....	1
Capítulo 1: Fundamentación Teórica .....	5
Introducción .....	5
1.1.-Videojuegos.....	5
1.2.- Evolución de los videojuegos .....	5
1.3.-Principales géneros .....	6
1.4.-Desarrollo de videojuegos.....	14
1.5.-Estructura básica de un videojuego. ....	17
1.6.-Inteligencia Artificial en los videojuegos .....	20
1.6.1.-Técnicas y Algoritmos de IA en Videojuegos.....	21
Capítulo 2: Propuesta de Solución .....	29
Introducción .....	29
2.1.- Teoría de Colores .....	29
2.2. Problemas detectados .....	30
2.3.- Resultados de la encuesta aplicada a los estudiantes de la Facultad 5 en la Universidad de las Ciencias Informáticas.....	34
2.4.-Propuesta de solución.....	39
2.5.- El archivo de configuración.....	39
2.6.-Algoritmo de asignación .....	41
2.7.- Patrón de Arquitectura seleccionado .....	42
Capítulo 3: Análisis de resultados .....	44

Introducción .....	44
3.1.- Pseudocódigo de la solución propuesta .....	44
3.2.- Resultados.....	54
CONCLUSIONES.....	57
RECOMENDACIONES.....	58
REFERENCIAS.....	59
BIBLIOGRAFÍA.....	60
APÉNDICE.....	62
Glosario de términos .....	62
ANEXOS .....	64

## Índice de figuras

Fig. 1.1 Super Mario Bros para Game & Watch.....	9
Fig. 1.2 Final Fantasy V en una Game Boy Micro.....	10
Fig 1.3 Tabla Wii Fit, ayuda a ponerse en forma haciendo ejercicio con Wii.....	11
Fig 1.4 Primer nivel de “Cassandra’ Journey”, publicado por “Big Fish Games”, ejemplo de juegos de objetos escondidos.....	12
Fig 1.5 Estructura b’asica de los videojuegos.....	19
Fig. 1.6 FSM para un cambio de luz.....	22
Fig. 1.7 Ejemplo de un Juego que aplica Árbol de Decisión.....	25
Fig. 2.1 Imagen del videojuego “BigCityAdventureSF”.....	30
Fig. 2.2 Imágenes del mismo nivel cargado dos veces diferentes.....	31
Fig. 2.3 Imagen del videojuego “MysteryPI”.....	32
Fig. 2.4 Primer nivel del videojuego “Fernanda”.....	32
Fig. 2.5 Tabla de indicadores para las Preguntas.....	37
Fig. 2.6 Tabla de Frecuencias para Rol.....	38
Fig. 2.7 Ejemplo de archivo de configuración.....	40
Fig. 2.8 Ejemplo de asignación para cada categoría.....	41
Fig. 2.9 Técnica del Alpha Blending actuando en la creación de la transparencia.....	42
Fig. 2.10 Vista clásica de una arquitectura de tres capas.....	43
Fig. 3.1 Ejemplo de los objetos que deben posicionarse en lugares que parezcan estar colgados...50	50
Fig. 3.2 Ejemplo de posicionamiento lógico.....	51
Fig. 3.3 Ejemplo de solapamiento de imágenes.....	52
Fig. 3.4 Ejemplo de superposicionamiento parcial.....	53

## Introducción

La Inteligencia Artificial (IA) incluye varios campos de desarrollo tales como la robótica, usada principalmente en el campo industrial; comprensión de lenguajes y traducción; visión en máquinas que distinguen formas y que se usan en líneas de ensamblaje; reconocimiento de palabras y aprendizaje de máquinas; sistemas computacionales expertos y se podrían mencionar otro montón de aplicaciones. Pero en esta ocasión el tema que nos interesa es IA aplicada a los juegos.

Esta ciencia es una herramienta muy útil que ayuda a crear opciones de respuesta para con los personajes y las acciones del jugador en los videojuegos. Está aplicada a los videojuegos, es un área de programación en la que los usuarios pueden llegar a realizar diversos intentos para hacer que la computadora actúe de una manera similar a como lo hace la inteligencia humana. Hay un número de principios subyacentes detrás del videojuego. Lo más importante es tener un sistema basado en reglas. La información y las reglas deben estar incorporadas en una base de datos, para que cuando el videojuego se enfrente a una situación, encuentre la información apropiada, y actúe en ella según las reglas que se aplican de acuerdo a dicha situación.

La IA como rama de la ciencia, pretende reproducir un ambiente coincidente a la realidad física, que nos traslade de un mundo físico a un mundo computarizado, logrado por los efectos reales que seamos capaces de llevar a los entornos virtuales, que a su vez algunos son producidos con la ayuda de comportamientos inteligente vinculados a la IA.

Dicha especialidad se podría ver estrechamente vinculada al desarrollo de agentes racionales no vivos en el mundo de la simulación, entendiendo por agentes, cualquier cosa capaz de percibir su entorno (recibir entradas), procesar tales percepciones y actuar en su entorno (proporcionar salidas). Se estudian los comportamientos inteligentes con el objetivo de ofrecer credibilidad y similitud con la realidad que nos rodea. (1)

Según Marvin Lee Minsky, científico estadounidense, considerado uno de los padres de la computación y fundador del laboratorio de IA del Instituto Tecnológico de Massachussets, plantea que la IA es una ciencia dedicada a la contribución de máquinas, que implementan tareas propias de humanos dotados de Inteligencia. Este criterio establece una unión entre las máquinas, la inteligencia y la actividad del ser humano.

Los términos IA, provienen de orígenes muy antiguos, aunque su concepto es un poco más joven.

De manera general esta especialidad se encarga de construir procesos que cuando se ejecutan sobre una arquitectura física producen acciones o resultados, basándose en la secuencia de entradas percibidas y en el conocimiento almacenado en tal arquitectura.

Es por el afán de lograr similitud con la realidad que surge la necesidad de crear comportamientos inteligentes que vayan más allá de un objeto primario.

Para lograr el comportamiento que se desea se hace necesario estudiar algoritmos inteligentes que sean capaces de ofrecer el resultado anhelado. Se habla de IA porque dentro de esta es posible encontrar una solución para el problema que en este trabajo se procura resolver.

Los estudios de animación del Instituto Cubano de Arte e Industria Cinematográfica tienen dentro de sus proyectos de trabajo el desarrollo de una serie de animados para niños, cuyo nombre es "Fernanda", la cual ha tenido gran aceptación por parte de los niños cubanos. Aún así los estudios de animación se interesaron en realizar un juego que relacione el quehacer de la detective Fernanda apoyado por el jugador o usuario, con el fin de lograr que los niños se identifiquen más con los dibujos animados cubanos.

Debido a que una de las especialidades de la Facultad 5 en la Universidad de las Ciencias Informáticas es Entornos Virtuales, se decide aceptar la producción de este juego "Fernanda".

"Fernanda" no tiene un comportamiento inteligente para lograr que los objetos de su escena tengan una posición distinta a la original después de haber cambiado de nivel o de recomenzar otra vez la partida.

El mismo no cuenta con un dinamismo que brinde posibilidades de cambio de posición para los objetos y hacia ahí se dirige esta investigación.

En aras de su perfección surge este proyecto de tesis cuyo **problema a resolver** se define en:

¿Cómo propiciar un comportamiento inteligente en objetos de una escena bidimensional?

Para materializar la solución del problema se tendrá en cuenta el siguiente **Objeto de estudio**: Proceso de desarrollo de técnicas de Inteligencia Artificial para videojuegos.

El **Objetivo General** que se traza esta tesis es: Modelar un algoritmo que controle la distribución inteligente de objetos en escenas 2D.

El **Campo de Acción** responde a: El proceso de distribución inteligente de objetos en escenas 2D mediante un algoritmo.

Para darle solución a este problema, se desarrollan las siguientes tareas.

## Tareas investigativas:

1. Estudio de los momentos evolutivos de la IA en los videojuegos.
2. Análisis de las técnicas básicas y avanzadas del desarrollo de Juegos 2D y sus relaciones con la Inteligencia Artificial.
3. Determinación de las características y requerimientos de los objetos en entornos 2D a partir de la experiencia en el desarrollo de Video Juegos en la esfera global.
4. Implementación, de un algoritmo inteligente capaz de controlar y distribuir objetos en escenas para el juego Fernanda.
5. Evaluación de los resultados obtenidos.

**Resultados esperados:** Se espera la elaboración de un Algoritmo que logre la distribución de objetos en escenas bidimensionales, de modo tal que denote la existencia de un comportamiento inteligente en el Juego Fernanda.

Los métodos utilizados para realizar esta investigación son los siguientes:

## Métodos Teóricos:

- El **Análisis Histórico-Lógico**, permitió el estudio de la evolución de las técnicas de IA y los algoritmos utilizados para el desarrollo de los videojuegos bidimensionales que posibilitaron la identificación de la actual tendencia rectora de este proceso.
- El **Inductivo-Deductivo** posibilitó determinar las características y requerimientos de los objetos bidimensionales en los videojuegos.

## Métodos Empíricos:

- La observación científica posibilitó obtener información sobre las técnicas de IA y los algoritmos utilizados en los videojuegos de objetos escondidos.
- La encuesta permitió el intercambio con los jugadores de videojuegos de objetos escondidos, en este caso específico, estudiantes de la Facultad 5 de la Universidad de las Ciencias Informáticas.

Finalmente el contenido de este documento está estructurado en tres capítulos, así como las correspondientes secciones de bibliografía, anexos y glosario de términos, organizados de la siguiente forma:

**Capítulo 1:** Se expone en este la fundamentación teórica y el estado del arte acerca de los videojuegos y su relación con la IA; describe conceptos claves que introducen al desarrollo de los posteriores capítulos. Se hace un estudio profundizando en los videojuegos bidimensionales, haciendo énfasis en los algoritmos que puedan ser utilizados para el desarrollo de estos videojuegos, como también se mencionan técnicas y herramientas de IA aplicadas a estos.

**Capítulo 2:** Observación de los videojuegos de objetos escondidos más populares, como son los llamados “detectives” identificando los problemas que estos presentan en cuanto a la distribución de los objetos, entre otros. Se hace una propuesta de solución para resolver dichos problemas en “Fernanda” y en otros videojuegos de este tipo.

**Capítulo 3:** Se muestra más detallada la descripción de la propuesta de solución y se presentan los resultados que se obtienen al aplicar dicha solución en un nivel del videojuego “Fernanda”.

# Capítulo 1: Fundamentación Teórica

## Introducción

La creciente industria de Software ha llegado a grandes extremos en la actualidad, así como se conoce, de sus orígenes y sus primeros pasos, el fin de su desarrollo es inimaginable. El hombre desde que comienza a pensar siente la necesidad de simular su realidad, quizás en sus inicios no conocía el concepto de simular o igualar, pero fue modificando su vida y desarrollando nuevas técnicas, hasta que así comprendió el concepto de mundo virtual. Una vez aquí quiso más y hasta la actualidad no se ha conformado, ha creado desde insignificantes juegos sencillos hasta enormes simuladores capaces de representar un mundo virtual muy similar a la realidad.

Para entender un poco más el tema es que se desarrolla este capítulo, aportando a los posteriores, conceptos que servirán de base para un mejor entendimiento de los interesados en este tema, así como una breve descripción de algunos aspectos que se tratarán en la investigación.

### 1.1.-Videojuegos

Un videojuego (del inglés video game) es un programa de computación, creado para el entretenimiento, basado en la interacción entre una o varias personas y un aparato electrónico (ya sea una computadora, un sistema arcade, una videoconsola, un dispositivo handheld ó actualmente un teléfono celular), el cual ejecuta dicho videojuego. En muchos casos, estos recrean entornos y situaciones virtuales en los cuales el jugador puede controlar a uno o varios personajes (o cualquier otro elemento de dicho entorno), para conseguir uno o varios objetivos por medio de reglas determinadas.

La industria del videojuego surgió con fuerza en Estados Unidos, Europa y Australia con la llegada del tenis de mesa y otros juegos muy sencillos a finales de la década de 1970 y se extendió rápidamente por todo el mundo. Tras conocer un crecimiento espectacular a lo largo de la década de 1980, la industria japonesa (especialmente la Nintendo Company Ltd.) se lanzó de lleno a perfeccionar y desarrollar la tecnología del juego, introduciendo juegos tan populares como el Super Mario Bros. (2)

### 1.2.- Evolución de los videojuegos

Los años de formación de los videojuegos están dados por la creación de juegos básicos que hicieron uso de dispositivos electrónicos interactivos con diferentes formatos de pantalla. El primer ejemplo fue en 1947, donde la idea de un “Dispositivo de Entretenimiento con Tubos de Rayos Catódicos”, fue concebida por Thomas T. Goldsmith Jr. y Estle Ray Mann. Inspirado en la pantalla de un radar este

dispositivo consistió en un equipo analógico que permitía a un usuario controlar un vector dibujado a puntos para simular un misil disparado a objetivos representados por dibujos pegados a la pantalla.

Otros ejemplos son: la computadora NIMROD en el Festival de Gran Bretaña en 1951, el juego “OXO” de Alexander S. Douglas para la computadora EDSAC (el primer calculador electrónico de la historia) en 1952, el juego interactivo llamado “Tennis for Two” de William Higinbotham en 1958 y el “Spacewar!” de los estudiantes Martin Graetz, Steve Russell, y Wayne Wiitanen para la computadora DEC PDP-1 en 1961. (3)

Los videojuegos se dividen en distintas categorías: educativos, de aventuras y deportivos. Los más populares emplean sonidos reales y colores, además de rápidos efectos visuales. Los juegos deportivos, como el fútbol, el baloncesto o el hockey sobre hielo, adquirieron especial popularidad a finales de la década de 1980, cuando determinados equipos profesionales prestaron su nombre a estas versiones en vídeo de su deporte.

Los establecimientos dedicados a la venta exclusiva de videojuegos son ya clásicos en la industria del ocio del mundo entero. Los sistemas de 16 bits, introducidos por las empresas japonesas a comienzos de la década de 1990, mejoraron notablemente la calidad de los gráficos de los videojuegos destinados al uso doméstico. El CD-ROM (un sistema de memoria con enorme capacidad de almacenamiento) ofrece hoy una amplia variedad de juegos a precios muy competitivos. En el futuro, el uso de la televisión de alta definición y de las líneas de telecomunicaciones para la transmisión de este tipo de juegos contribuirá a aumentar aún más su realismo.

## **1.3.-Principales géneros**

### **1.3.1.-Aventura**

Juegos en los que el protagonista debe avanzar en la trama interactuando con diversos personajes y objetos. También suelen ser incluidos en esta sección los videojuegos conocidos como Hit n' Run, mencionados más abajo. Si bien es cierto que en el pasado el género aventura comprendía una gama de subgéneros y estilos más amplia, con el paso del tiempo y hasta la actualidad se ha ido consolidando. Se pudieran haber incluir juegos de lo más variados, muchos de los cuales hoy se consideran pertenecientes a otros estilos como plataformas, rompecabezas e incluso acción, sirviendo en cierto modo como género abanico, para una época en la que apenas se habían empezado a desarrollar algunas fórmulas, sin que estas estuvieran aún lo suficientemente desarrolladas como para reclamar nuevos géneros a los que pertenecer por derecho.

Algunos ejemplos de cada uno de estos subgéneros podrían ser:

- Aventuras de acción.
- Aventuras
- Aventuras conversacionales

### **1.3.2.-Shooter**

En estos juegos el protagonista ha de abrirse camino a base de disparos. Según su temática y desarrollo pueden clasificarse en diferentes sub-géneros como la acción en primera persona o "FPS", acción en tercera persona o "TPS", shooter multijugador masivo o "MMFPS", acción táctica, los Shooter de scrolling lateral o "Run-And-Gun", los mata marcianos o los juegos de pistola. Algunos ejemplos son:

- Acción Táctica
- Mata Marciano
- Sobre raíles

### **1.3.3.-Educativos**

Juegos cuyo objetivo es dar a conocer al usuario algún tipo de conocimiento. Su mecánica puede abarcar cualquiera de los otros géneros. Están dirigidos para todas las edades, por lo que se suelen considerar como juegos casuales. English Training ayuda a mejorar el inglés, al igual que Mi Experto en Francés, esta lengua. Brain Age, Brain Training y Big Brain Academy activarán tu mente a golpe de juegos y matemáticas.

### **1.3.4.-Estrategia**

Se caracterizan por la necesidad de manipular a un numeroso grupo de personajes, objetos o datos para lograr los objetivos. Según su temática los hay de gestión (ya sea esta económica o social) y bélicos.

Mientras, por su mecánica, pueden ser clasificados en tiempo real (también llamados "RTS" (Real Time Strategy)), o por turnos (TBS, o Turn Based Strategy). Por ejemplo:

- Estrategia en Tiempo Real

- Estrategia por Turnos, La mayoría de estos son juegos históricos de batalla.

## 1.3.5.-Lucha

Juegos basados en el combate cuerpo a cuerpo. Se dividen en juegos de uno contra uno o "versus", juegos de avanzar y pegar o "beat'em up" y el híbrido de ambos "ree-For-All" o todos contra todos.

Este género es uno de los más aclamados del mundo del videojuego, de aquí la prolífera lista de juegos de lucha que podemos encontrar en el mercado.

## 1.3.6.-Survival horror

En ellos los protagonistas viven aventuras dónde deben salir airosos de situaciones típicas de una película de terror (escapar de una casa llena de zombis, huir de un asesino, resolver misterios para aplacar a los fantasmas, etc.). Un factor importante es el terror psicológico, ayudado de una buena ambientación y apartado sonoro. Algunos ejemplos son:

- Project Zero/Fatal Frame, Dino Crisis, Parasite Eve, Resident Evil, Clock Tower, Silent Hill o Alone in the Dark.

## 1.3.7.-Plataformas



Fig. 1.1 Super Mario Bros para Game & Watch

Los juegos de plataformas o simplemente plataformas, son un género de videojuegos que se caracterizan por tener que andar, saltar o escalar sobre una serie de plataformas y acantilados, con enemigos, mientras se recogen ítems para poder completar el juego. Suele usar scroll horizontal hacia izquierda o hacia la derecha. Es uno de los géneros más veteranos de este mundo.

Aunque dentro de este género se encuentran también juegos como Donkey Kong, Adventure Island, Castlevania, Prince of Persia o Earthworm Jim. También son juegos de plataformas Super Mario 64 o Crash Bandicoot, que adaptan a las 3D el estilo de juego de este género, presentando extensos escenarios independientes. Además, también nos encontramos con las 2.5D, que se trata de la combinación de las 3D y las 2D, como ocurre en, por ejemplo, New Super Mario Bros.

### 1.3.8.-Videojuegos de rol



Fig. 1.2 Final Fantasy V en una Game Boy Micro.

Se inspiran en los juegos de rol clásicos, donde el protagonista interpreta un papel y ha de mejorar sus habilidades mientras interactúa con el entorno y otros personajes. Son diversos los subgéneros que engloba esta categoría: los roguelike, juegos de rol multijugador masivos en línea, MUD o los tácticos, a medio camino entre el género de estrategia y el rol, conocidos también como Strategic RolePlaying Games. También otro subgénero no tan popular son los Videojuegos de rol de batallas por turno, que se caracterizan por ser un videojuego de rol normal, pero al enfrentarse con los enemigos no es en tiempo real sino por turno.

La duración de un sólo juego de éstos, ahora demanda mucho tiempo; puede variar mucho para su terminación o fin por parte del usuario promedio. Si se juega a un ritmo sano, detallista y a fondo, no es difícil observar un periodo de varios meses o más de un año (dependerá de la veneración que le dedique cada jugador a determinado título o si lo abandona por periodos).

### 1.3.9.-Musicales

Su desarrollo gira en torno a la música. Ejemplos:

- Karaoke: Singstar (PlayStation), Lips (XBox 360), Boogie (NDS, Wii, PS2) o Karaoke Revolution (PS2, PS3, Wii, Gamecube, XBox, XBox 360).

- Instrumentos musicales: Guitar Hero, Rock Band, Donkey Konga, Wii Music o Jam Sessions.  
Baile: Dance Dance Revolution, Pump It Up, Stepmania.
- Otros: Bust a Groove, beatmania, Electroplankton, Space Channel 5, Elite Beat Agents.

### 1.3.10.-Party Games

En este género los jugadores habrán de ir avanzando por turnos por un tablero virtual e ir superando diversas pruebas de tipos muy diversos en los que compiten entre sí por llegar lo antes posible a la meta, o conseguir la máxima cantidad posible de puntos. Algunos ejemplos son:

- Sonic Shuffle
- Mario Party
- Rayman Raving Rabbids
- Viva Piñata: Party Animals
- Wii Play

### 1.3.11.-Simulación

Involucran al jugador en una situación simulada determinada, ya sea de gestor de un zoo, una ciudad o una vida propia virtual. Ejemplo hay varios, pero solo mencionaremos algunos.

- Simuladores de baile: Dance Dance Revolution.
- Simuladores de vuelo: Ace Combat o Microsoft Flight Simulator.
- Simuladores ferroviarios: Microsoft Train simulator,
- Simuladores de submarinos: Silent Hunter o Forever Blue.
- Simuladores de relaciones sociales: Los Sims o Animal Crossing.

### 1.3.12.-Deportivo



Fig. 1.3 Tabla Wii Fit, ayuda a ponerse en forma haciendo ejercicio con Wii.

Se basan en deportes, reales o ficticios, y pueden subdividirse en simuladores y en "arcade" (menos realistas que los primeros).

- Fútbol: Pro Evolution Soccer, FIFA, PC Fútbol, Mario Strikers: Charged Football
- Baloncesto: NBA Live
- Fútbol Americano: Madden NFL
- Ciclismo: Pro Cycling
- Béisbol: MLB 2K, MLB Power Pros
- Skate: Tony Hawk's Pro Skater, Skate
- Golf: Tiger Woods PGA Tour, Everybody's Golf, Pangya
- Varios: Mario & Sonic en los Juegos Olímpicos, serie oficial de los JJ.OO., Wii Sports

Suelen tener una revisión anual de cada uno de los deportes, aparte de las posibles ediciones especiales de varios eventos.

### 1.3.13.-Carreras

Son juegos 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, arcade y simuladores.

- Arcade: Crash Team Racing y Mario Kart.
- Simuladores: Gran Turismo, Top Gear, Project Gotham Racing, Need for Speed y Forza Motorsport.

### 1.3.14.-Sandbox

También conocidos como acción-aventura, se caracterizan por ser juegos en los que el jugador puede hacer lo que él quiera, como viajar libremente por el mapa del juego, e interactuar con casi todo lo que este a su disposición. Estos juegos son una mezcla de disparos, luchas y carreras. Son muy conocidos por su temática "antihéroe" ya que la mayoría tienen como protagonistas a criminales. Algunos ejemplos son:

- La saga Grand Theft Auto, Driver, El Padrino, Just Cause, Los Simpson Hit and Run, True Crime, Scarface, Fallout, No More Heroes, Pepsiman o S.T.A.L.K.E.R.: Shadow of Chernobyl.

### 1.3.15.-Géneros mixtos y nuevos géneros

La creación continua de videojuegos ha dado a menudo como resultado obras que podrían bien englobarse en más de un género incluso no pertenecer a ninguno de los ahora establecidos. Algunos ejemplos serían:

- Half-Life (Shoot'em up/Aventura de acción), Tomb Raider (Aventura de acción/plataformas/puzzle), Koudelka (videojuego de rol), Flashback: The Quest for Identity (videoplataformas/aventura de acción), Bishi-Bashi (habilidad), Dragon's Lair (aventura animada), Portal (Shoot'em up/Puzzle/Plataformas), etc.

Ejemplos del surgimiento de nuevos géneros a partir de experimentos iniciales podrían ser los "Videojuegos Musicales", como Bust a Groove/Bust a Move o Parapa the Rapper, o a comienzos/mediados de los ochenta el género juego de plataformas, por fases avanzables, que posteriormente hemos conocido, respecto del plataformas primigenio de pantalla fija (tipo Donkey Kong o Mario Bros.), que pasaría poco a poco a ser englobado dentro del género, más amplio, rompecabezas, siendo a su vez muchos de estos incluidos junto con otros, al principio, dentro del género entonces muy diverso y heterogéneo de juego de aventura. La especialización que hoy conocemos, la cual convierte a muchas de las clasificaciones originales en poco menos que arbitrarias o inviables, no ha sido sino fruto de una paulatina profundización y desarrollo, acorde con los adelantos tecnológicos, de las distintas fórmulas iniciales que con el tiempo se fueron dando.

### 1.3.16 Puzzle

Los videojuegos puzzle, e igual llamados videojuegos de lógica o inteligencia, son una variedad de videojuegos que se caracterizan por ser de agilidad mental. Estos pueden involucrar problemas de lógica, estrategia, completar palabras o hasta simple azar. Se dice que los puzzles suelen ser adictivos. Entre los puzzles más comunes se encuentra el Tetris, el cual tiene como objetivo acomodar piezas que caen para organizarlas de la manera correcta. Otros pueden ser organizar objetos en un escenario establecido, como en Sokoban. También pueden ser alcanzar una meta eliminando enemigos o activando botones, como en Bomberman.

Tetris es considerado el juego característico de este género.

Muchos videojuegos que no son clasificados como puzzles, contienen acertijos que hay que resolver para conseguir avanzar, como si fueran puzzles. Esto sucede en juegos como The Legend of Zelda y Resident Evil.

El **juego de objetos escondidos** (a veces llamado “fotos escondidas”) es un género de los videojuegos puzzle, en el cual el jugador debe encontrar los objetos que se piden en una lista que están escondidos dentro de una imagen. Los juegos de objetos escondidos tienen una tendencia popular entre los juegos casuales y apenas se tiene que pagar para comprarlos.

Incluidos en este género se encuentran los videojuegos de “Big Fish” y de los “Spin Top” (como el “Mystery PI: The Lottery Ticket”). Además de estos también se encuentra incluido el juego “Fernanda”, dado que las características que este tiene son similares a las de dichos juegos de objetos escondidos existentes.



Fig. 1.4 Primer nivel de “Cassandra’ Journey”, publicado por “Big Fish Games”, ejemplo de juegos de objetos escondidos.

## 1.4.-Desarrollo de videojuegos

El desarrollo de videojuegos es la actividad por la cual se diseña y produce un videojuego, desde el concepto inicial hasta el juego en su versión final, el producto terminado.

Ésta es una actividad multidisciplinaria, que involucra profesionales de la informática, el diseño, el sonido, la actuación, entre otros.

El desarrollo de un videojuego generalmente sigue el siguiente proceso:

- Concepción de la idea del juego
- Diseño
- Planificación
- Producción
- Pruebas
- Mantenimiento

El proceso es similar a la creación de software en general, aunque difiere en la gran cantidad de aportes creativos (música, historia, diseño de personajes, niveles, entre otros) necesarios. El desarrollo también varía en función de la plataforma objetivo (PC, celulares, consolas), el género (estrategia en

tiempo real, RPG (Rol Playing Game) ó juegos de rol, aventura gráfica, plataformas, entre otros) y la forma de visualización (2d, 2.5d y 3d).

## 1.4.1.-Concepción

En esta etapa es necesario definir los aspectos fundamentales que conformarán el videojuego, entre los que se encuentran:

- Género: Dentro de que género o géneros se va a desarrollar el juego. De no corresponder a un género conocido, se deben especificar las características
- Game play: Lo que generará diversión a la hora de jugarlo.
- Story board: Algunas ideas sueltas acerca de cómo debe lucir el juego en cuanto a personajes, ambientación, música, entre otros.

## 1.4.2.-Diseño

En esta fase se detallan todos los elementos que compondrán el juego, dando una idea clara a todos los miembros del grupo desarrollador acerca de como son. Entre estos elementos tenemos:

## 1.4.3.-Diseño de Arte

Abarca los elementos de:

- Historia: Forma en que se desenvolverán los personajes del juego y la historia del mundo representado. No todos los juegos tienen historia.
- Sonido: Detallada descripción de todos los elementos sonoros que el juego necesita para su realización. Voces, sonidos ambientales, efectos sonoros y música.
- Interfaz: Es la forma en que se verán los elementos GUI y HUD, mediante los cuales el usuario interactuará con el juego.
- Gráficos: Dependiendo de si el juego es 2d, 2.5d o 3d, en este apartado se deben especificar los sprites, tiles y modelos 3d a utilizar. Generalmente esta fase incluye un desarrollo conceptual y una especificación de las características de los modelos.

## **1.4.4.-Diseño de mecánicas**

Es la especificación del funcionamiento general del juego. Es dependiente del género y señala la forma en que los diferentes agentes virtuales interactuarán dentro del juego, es decir, las reglas que rigen éste.

## **1.4.5.-Diseño de programación**

Describe la manera en que el videojuego será implementado en una máquina real (PC, consola, celular, entre otros) mediante un cierto lenguaje de programación y siguiendo una determinada metodología. Generalmente en esta fase se generan diagramas de UML que describen el funcionamiento estático y dinámico, la interacción con los usuarios y los diferentes estados que atravesará el videojuego como software.

De toda la fase de diseño es necesario generar un documento llamado Documento de Diseño, que contiene todas las especificaciones de arte, mecánicas y programación.

## **1.4.6.-Planificación**

En esta fase se identifican las tareas necesarias para la ejecución del videojuego y se reparten entre los distintos componentes del equipo desarrollador. También se fijan plazos para la ejecución de dichas tareas y reuniones clave, con la ayuda de herramientas de diagramación de actividades como GANTT y PERT.

## **1.4.7.-Producción**

Se llevan a cabo todas las tareas especificadas en la fase de planificación, teniendo como guía fundamental el documento de diseño. Esto incluye entre otras cosas la codificación del programa, la creación de sprites, modelos 3d, la grabación de sonidos, voces y música, la creación de herramientas para acelerar el proceso de desarrollo, entre otras.

## **1.4.8.-Pruebas**

Al igual que los software convencionales, los videojuegos deben pasar por una etapa donde se corrigen los errores inherentes al proceso de programación y a diferencia de este, los videojuegos requieren un refinamiento de su característica fundamental, la de producir diversión de manera interactiva (jugabilidad). Generalmente esta etapa se lleva a cabo en dos fases:

- Pruebas Alpha: Se llevan a cabo por un pequeño grupo de personas, que con anterioridad estén involucradas en el desarrollo, lo que puede incluir artistas, programadores, coordinadores, etc. El propósito es corregir los defectos más graves y mejorar características de jugabilidad no contempladas en el documento de diseño.
- Pruebas Beta: Estas pruebas se llevan a cabo por un equipo externo de jugadores, que sean contratados para la ocasión o un grupo miembro del proyecto (grupo QA). De estas pruebas el videojuego debe salir con la menor cantidad posible de defectos menores y ningún defecto medio o crítico.

## 1.5.-Estructura básica de un videojuego.

Conocer cómo funciona un juego no ayudara a comprender básicamente el comportamiento de los algoritmos dentro del mismo, por esta razón brevemente damos esta explicación:

Por lo general un videojuego es un programa diferente a los que comúnmente se desarrollan. Estos deben funcionar en tiempo real mientras se ejecuta, él deberá estar realizando algunas tareas como: actualizar coordenadas, calcular colisiones, todo esto en dependencia del juego. También estará esperando la ocurrencia de algún evento, en el que actuará en consecuencia de lo que se le haya ordenado a través de la acción de un usuario. Todo esto ocurrirá dentro de un ciclo o loop como se le suele llamar comúnmente. (4)

Básicamente la estructura de un videojuego está dada por las siguientes partes:

1. Inicialización
2. Ciclo de videojuego
  1. Entrada
  2. Procesamiento
  3. Salida
3. Finalización.

En qué consiste cada una de estas partes:

**1.5.1.-Inicialización:** Inicializa todo lo que será usado luego en el ciclo del videojuego. Por ejemplo: la biblioteca gráfica, un modo gráfico, el sistema de sonido/música, de texto y cualquier otro tipo de sistema necesario. Además se reserva memoria para los objetos que intervienen en el juego, creación de estructuras de datos, carga de sonidos, de imágenes y de recursos en general. También en este proceso se inicializa las posiciones iniciales de los personajes y carga de puntajes desde un archivo, entre otras.

**1.5.2.-Ciclo del videojuego:** el ciclo del videojuego es un loop que se estará repitiendo una y otra vez. Aquí es donde ocurre toda la acción del juego, y la única forma para poder salir de este ciclo es cuando el jugador pierde, llega al final del juego o sale del videojuego con alguna combinación de teclas o presionando algún botón del mouse. El ciclo del juego consta básicamente de tres partes:

**1.5.2.1.-Entrada:** en esta parte se obtiene desde algún dispositivo de entrada (teclado, mouse, joystick, etc.) todo lo que realiza el jugador, por ejemplo que tecla presionó/soltó del teclado, que botón del mouse presionó/soltó, si movió el mouse en alguna dirección, etc.

**1.5.2.2.-Procesamiento:** es donde se procesa toda la información que se recibió en el punto anterior, se toman decisiones a partir de los datos de entrada y se encuentra presente toda la lógica del juego. Se procesa la física, inteligencia artificial, comunicación de datos en red, etc.

**1.5.2.3.-Salida:** muestra toda la información ya procesada en el punto anterior, además de los gráficos en pantalla y reproducción de sonido.

**1.5.3.-Finalización:** por último en esta parte se hace básicamente lo opuesto a lo hecho en la inicialización, es decir, eliminar de la memoria todos los recursos almacenados, ya sea imágenes, sonidos o música. Cierra todos los sistemas que se abrieron en la inicialización y guardar datos de puntajes en un archivo.

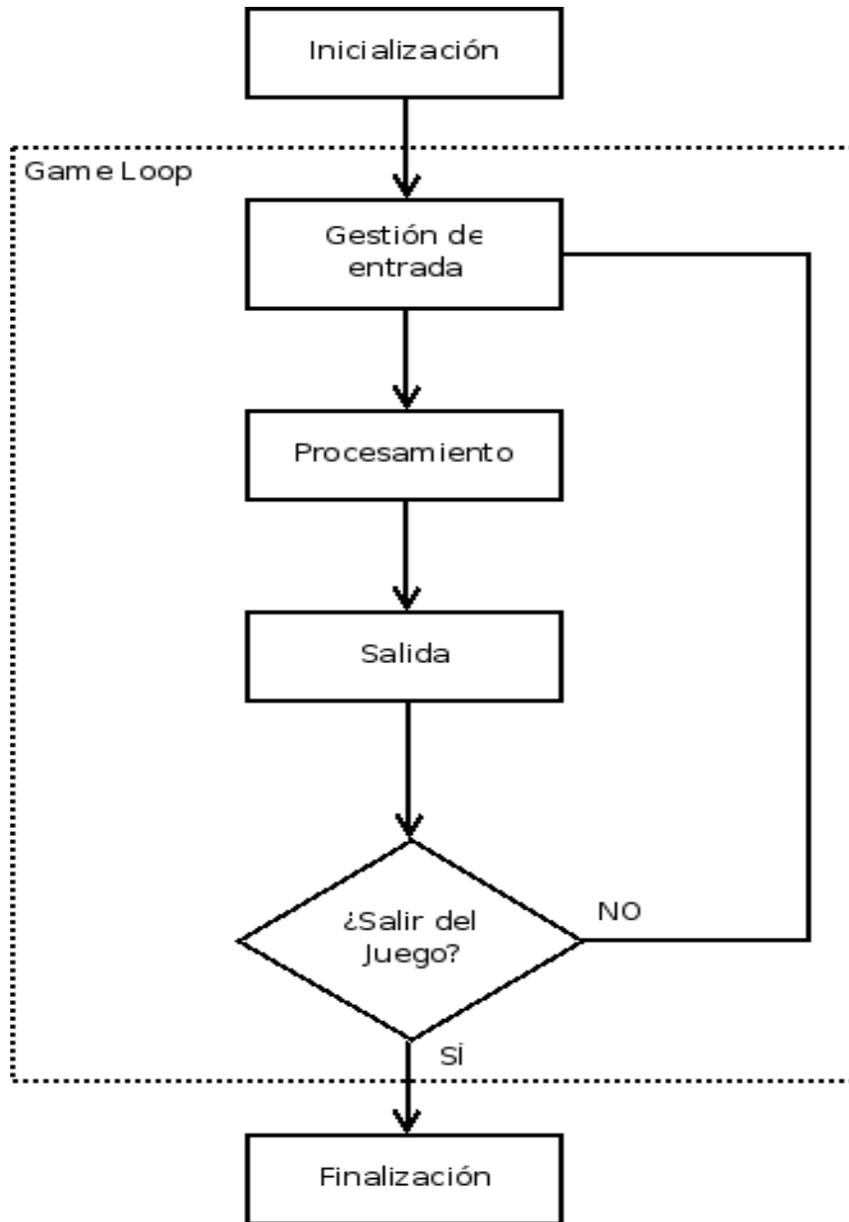


Fig. 1.5 Diagrama de flujo de la Estructura básica de los videojuegos.

## 1.6.-Inteligencia Artificial en los videojuegos

La inteligencia artificial podría definirse (en el mundo de los videojuegos) como el conjunto de algoritmos diseñados para simular un ser inteligente, con todo lo que eso conlleva, bajo un mundo virtual.

Existen muchos tipos de programas que hacen un gran uso de elementos de Inteligencia Artificial, como pueden ser los planificadores económicos, sistemas de reglas de negocio, sistemas de automática y control, entre otros. Estos elementos incluyen localización, búsqueda en árboles, resolución de problemas, toma de decisiones y aprendizaje. Pero uno de los campos del desarrollo de software que más ha ido tomando de la IA ha sido, sin duda, el desarrollo de videojuegos.

La ley de Moore sobre el incremento en la velocidad de los procesadores a lo largo del tiempo parece también aplicarse al desarrollo de los videojuegos, que parecen duplicar su complejidad cada 18 meses. Este aumento de complejidad, hace que los juegos aumenten su interés y su capacidad de adicción, lo que ha logrado que los juegos pasen de ser una mera distracción de unos minutos a una forma de expresión artística y un hobby serio para los jugadores. (5)

La historia de la Inteligencia Artificial aplicada a los juegos se remonta a finales de los años 50, cuando Arthur L. Samuel, trabajador de IBM, desarrolló un programa para la IBM 701, la primera máquina comercial de IBM, capaz de jugar a las damas mediante aprendizaje (6). Este programa representa el primer programa desarrollado en el mundo capaz de aprender y supuso una demostración de fuerza tanto a nivel de desarrollo software como a nivel de hardware que marcó un gran incremento en las ventas de máquinas de IBM. Después de este acercamiento, la aplicación de técnicas de IA en los videojuegos se fue extendiendo, en gran medida debido a que los juegos empezaron a dejar de ser mayoritariamente juegos de jugador contra jugador, lo cual motivaba la aparición de técnicas que permitieran a los usuarios sentir que estaban jugando contra otros oponentes humanos o, al menos, con un nivel de dificultad similar para que el juego supusiera un reto y fuera “adictivo”.

Más de 50 años después, la IA se ha convertido en un pilar fundamental de los videojuegos, tal y como puede verse en la existencia de módulos formativos específicos en los Máster de Videojuegos, así como en la aparición de una gran multitud de libros acerca de este tema tan específico o la gran demanda de especialistas en IA por parte de empresas desarrolladoras de videojuegos.

¿En qué aspectos concretos ayuda la IA a los Videojuegos? Ante todo a modelar el comportamiento de los jugadores no humanos (NPC en inglés), y para ello aporta múltiples ayudas, entre estas se profundiza en las que pueden ser aplicadas al tipo de juego al que está dirigido este trabajo.

## 1.6.1.-Técnicas y Algoritmos de IA en Videojuegos

Para lograr como producto final un videojuego bidimensional y que además aporte inteligencia en su programación, hay que tener en cuenta algoritmos y técnicas que hacen posible su funcionamiento, sean estos conformados con técnicas inteligentes o no, aunque el objetivo final, en el caso de estudio de esta investigación es lograr un comportamiento inteligente es la escena del juego. Por lo tanto algunos de los algoritmos mencionados a continuación, en su mayoría están dirigidos a su aplicación en los videojuegos.

**A\*:** El algoritmo de búsqueda A\* (A Estrella) se clasifica dentro de los algoritmos de búsqueda en grafos, este encuentra, siempre y cuando se cumplan unas determinadas condiciones, el camino de menor coste entre un nodo origen y uno objetivo.

Como todo algoritmo de búsqueda en anchura, A\* es un algoritmo completo: en caso de existir una solución, siempre dará con ella.

Este algoritmo no solo es aplicado a juegos puzzle, es también aplicado a otro tipo de juegos, independientemente que sean juegos 2D o 3D.

### Propiedades de la búsqueda A\*

- La búsqueda A\* es óptima
- Su búsqueda es completa
- Almacena todo los nodos en memoria

### **Máquinas de Estado:**

Una máquina de estados finitos (FSM) es un modelo de los comportamientos de un sistema o un objeto complejo, con un número limitado de condiciones definidas o modos que cambian con las circunstancias.

Las máquinas de estados finitos cuentan con 4 elementos fundamentales:

# Capítulo 1: Fundamentación Teórica

---

- 1- Estados que definen comportamientos y pueden producir acciones.
- 2- Transición de estados que son el movimiento de un estado a otro.
- 3- Reglas o condiciones que deben cumplirse para llevar a cabo una transición de estados.
- 4- Eventos de entrada que pueden ser generados externa o internamente, que pueden disparar reglas y llevar a transición de estados.

Una máquina de estados finitos debe tener un estado inicial que provee el punto de partida y un estado actual que recuerde el producto de la última transición de estado. Los eventos de entradas recibidos actúan como gatillos que causan la evaluación de un conjunto de reglas que controlan la transición del estado actual a otros estados.

Las máquinas de estados finitos se usan típicamente para controlar sistemas donde el conocimiento se representa en estados y las acciones son controladas por reglas.

Esta técnica ha sido adoptada por el campo de la IA debido a que se originó en las matemáticas y se uso inicialmente para la representación del lenguaje.

Las máquinas de estados son una de las técnicas más básicas utilizadas dentro de la IA para videojuegos. Al aplicarse a estos permiten establecer distintos tipos de comportamientos en los NPC (Non-Player Character ó caracteres no-jugadores) que van cambiando a lo largo del tiempo en función de una serie de parámetros. Ejemplificando el concepto de máquina de estados, si entramos en una cueva llena de enemigos, éstos pueden estar programados para matarnos si estamos cerca de ellos, pero si conseguimos “robar” el cofre del tesoro que están vigilando, todos los enemigos nos perseguirán. Este comportamiento “simula” una especie de señal de alerta y hace creer al humano que los NPC son más inteligentes. Hay algunas variantes de las máquinas de estados que tienen más interés para la IA, como son las máquinas de estados difusas, dónde las transiciones entre los estados ocurren de forma difusa, es decir, en función de una serie de probabilidades, lo cual hace todavía más interesante el comportamiento de los NPC.

## Ventajas de las Máquinas de Estados Finitos.

- Su simplicidad la hace sencilla de implementar para desarrolladores sin experiencia con poco o ningún conocimiento extra.

## Capítulo 1: Fundamentación Teórica

---

- Dado un número de entradas y un estado actual conocido el estado de transición puede ser predecible facilitando las pruebas.
- Son rápidas de diseñar e implementar al igual que en su ejecución.
- Es una técnica vieja, muy usada, bien probada y con muchos ejemplos de los que aprender.
- Son relativamente flexibles. Hay un número de formas de implementar un sistema basado en Máquina de Estados Finitos en cuanto a su topología y es fácil incorporar otras técnicas.
- Fácil de llevar de una representación abstracta sin significado a una implementación codificada.
- Pocos requerimientos de microprocesador. Dominios bien definidos donde el tiempo de ejecución se comparte entre módulos o subsistemas. Solo el código del estado actual debe ser ejecutado y quizás una pequeña cantidad de lógica para determinar el estado actual.
- Fácil determinación de la posibilidad de alcanzar un estado, cuando se representa de forma abstracta se hace obvio cuando se puede alcanzar un estado desde otro y que se requiere para hacerlo.
- Las condiciones para la transición de estados son rígidas.

### Desventajas de las Máquinas de Estados Finitos.

- Su naturaleza predecible puede no ser deseada en algunos dominios como juegos de computadoras.
- Sistemas grandes implementados con máquinas de estados finitos pueden ser difíciles de manejar y mantener sin un diseño bien pensado. El estado de transición puede causar cierto grado de “factor espagueti” al tratar de seguir la línea de ejecución.
- Al no estar preparado para todo el dominio de problemas, solo debe ser usado cuando el comportamiento del sistema pueda ser descompuesto en estados separados con condiciones bien definidas para la transición de estados. Esto significa que todo: estados, transición y condiciones deben ser conocidos y bien definidos. (7)

Un ejemplo de máquinas de estados finitos es el cambio de luz el cual puede estar encendido o apagado.

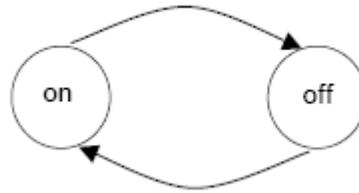


Fig. 1.6 FSM para un cambio de luz

La **Lógica Difusa/Borrosa** es aplicada con frecuencia en los videojuegos. Esta se basa en lo relativo de lo observado, reconoce más que simples valores verdaderos y falsos: grados de veracidad o falsedad. Para cada conjunto difuso, existe asociada una función de pertenencia de sus elementos que indica en qué medida el elemento forma parte de ese conjunto.

La Lógica Difusa o Borrosa se basa en la teoría que para un instante dado, no es posible precisar el valor de una variable  $x$ , sino tan solo reconocer el grado de pertenencia a cada uno de los conjuntos en que se ha participado el rango de variación de la variable. (8)

Los sistemas de lógica se ven aplicados en la tecnología cotidiana. Imitan la forma en que toman decisiones los humanos, con la ventaja de ser mucho más rápidos. Estos sistemas son generalmente robustos y tolerantes a imprecisiones y ruidos en los datos de entrada.

### Ventaja al Aplicar la Lógica Difusa

Una de las ventajas al aplicar la Lógica Difusa está dada porque ofrece salidas de una forma veloz y precisa, disminuyendo así las transiciones de estados fundamentales en el entorno físico que controle.

Entre las múltiples aplicaciones que tiene se podría mencionar su presencia en el juego "THE SIMS", el cual funciona como un simulador de vida, aquí es donde se construyen viviendas por el jugador, para de esta forma tratar de aportar una mejor forma de vida a las personas.

Dentro de estas viviendas estarán varios NPCs que actuarán de acuerdo a su perfil psicológico (definido al inicio del juego mediante algunos atributos); estos NPCs son llamados Sims.

Mediante las Máquinas de Estados Finitos, basadas en sistemas de Lógica Difusa es que se le aporta inteligencia a la programación de este juego. (8)

**“Pathfinding” o búsqueda de caminos:** los algoritmos de pathfinding o búsqueda de caminos son aquellos que encuentran un camino válido desde un punto A a otro B en un terreno con obstáculos. Es otro de los algoritmos que la mayoría de los videojuegos 2D utilizan, estos cuentan con pantallas confeccionadas a modo de mapas, dónde se mueven tanto los jugadores como los NPC. Por regla general, algunos (o todos) NPC, tienen que buscar al jugador para evitar que este termine la fase o continúe su objetivo y, para esta tarea, se pueden utilizar los algoritmos de “pathfinding” muy habituales y técnicas básicas dentro de la IA. (8)

### Aplicaciones del “Pathfinding” o búsqueda de caminos

En el mundo de los videojuegos existen muchas aplicaciones para este tipo de algoritmos. Son necesarios siempre que se tenga un escenario o nivel con objetos sólidos o infranqueables en los que se quiere que el computador mueva un objeto de un punto a otro del escenario. Esto ocurre en los juegos de estrategia y de rol, donde se les dice a las tropas donde deben moverse y ellas se las apañan para alcanzar ese punto a través del terreno. Pero también cuando el ordenador necesita de un pathfinding para mover un enemigo de un punto a otro del escenario.

**Inteligencia Emergente,** este es un concepto que ha empezado a implementarse en los Videojuegos hace relativamente poco tiempo. Estas técnicas de IA permiten que en los juegos se muestre una serie de criaturas (un buen ejemplo sería un Tamahotchi), que son capaces de ir aprendiendo en función de las decisiones del jugador y también evolucionan de forma acorde a estas acciones y a su aprendizaje. Estas técnicas están muy inspiradas en el funcionamiento de la naturaleza, por lo que el jugador tiene la sensación de que la evolución de estas criaturas es “real” y no simulada.

El añadir este tipo de comportamientos emergentes a los juegos, aporta a que mejoren en cuanto a variedad y también permite volver a jugar los juegos sin sentir que se está haciendo exactamente lo mismo que la primera vez.

**“Inteligencia Estafadora”:** Algunos juegos utilizan técnicas para detectar cuándo el jugador está en una situación muy ventajosa con respecto a la computadora y, entonces, favorecen el comportamiento de la IA para conseguir una situación más compleja para el jugador y aumentar el grado de dificultad. Este tipo de técnicas son muy habituales en videojuegos de deportes como el EA Sport’s Madden NFL o el Mario Kart. (8)

### Árbol de decisión:

Un árbol de decisión es un diagrama que representa en forma secuencial: condiciones y acciones; condiciones que se consideran en primer lugar, en segundo lugar y así sucesivamente. Este método permite mostrar la relación que existe entre cada condición y el grupo de acciones permisibles asociado con ella. Cada vez que se ejecuta un árbol de decisión, sólo un camino será seguido dependiendo del valor actual de la variable evaluada. Pueden ser aplicados a problemas generales de decisión y a juegos de estrategia, con una o más decisiones.

En este caso particular nos interesa lo tenga que ver con árboles de juego aplicados a juegos, valga la redundancia. Este tipo de juego a la hora de implementarlos, los objetivos y la forma de conseguirlos suelen ser muy claros (conseguir todos los trozos del pastel de colores, matar al rey, comerse todas las damas del contrario, etc.). Lo bueno de este tipo de juegos es que su tiempo de vida es potencialmente infinito, ya que, con muy pocos recursos, cada partida será distinta y el jugador puede estar toda la vida jugándolo sin aburrirse.

Un ejemplo de los Árboles de Decisión aplicado a juegos, es el llamado Puzzle o Rompecabezas, que posee fichas ordenadas en forma de matriz en un tablero.



Fig. 1.7 Ejemplo de un Juego que aplica Árbol de Decisión

### Ventajas de Los Árboles de Decisión

Son un método muy efectivo porque:

- Dan claridad al problema y las opciones pueden ser cambiadas.
- Permiten analizar completamente todas las posibles consecuencias de las decisiones.

## Capítulo 1: Fundamentación Teórica

---

- Permiten un marco de trabajo para cuantificar los valores de los resultados y las probabilidades de alcanzarlos.
- Ayudan a hacer las mejores decisiones sobre la base de información existente y mejores beneficios.
- Pueden ser usados en conjunto con el sentido común. (Jaramillo A, 2008)

Existen más técnicas y algoritmos aplicados a videojuegos, estas no son las únicas existentes, pero al centrar esta investigación en los videojuegos bidimensionales y categorías puzzles, los resultados fueron más limitados, además que se tuvo en cuenta que pudieran aportar algo al resultado final.

Después de estudiar las técnicas de inteligencia artificial y los juegos de objetos escondidos más exitosos en el mercado, los autores de esta tesis llegaron a la conclusión de que no es necesario utilizar ninguna técnica que realice el posicionamiento automáticamente por las siguientes razones:

La comparación de píxeles consiste en buscar el lugar donde el objeto se asemeje más al fondo, esta posición se encuentra restando cada pixel del objeto a los del fondo y calculando un valor que represente el resultado de todas estas restas, mientras menor sea ese resultado es mayor la semejanza del objeto y el fondo. Este coeficiente de semejanza se calculará de la misma forma en cada carga del nivel y por tanto será seleccionada la misma posición lo cual haría al juego estático.

En caso de que se escogiera un valor aleatorio dentro de las X posiciones más semejantes (de menor coeficiente), quedaría la posibilidad de que todas estas se encuentren en una misma región corridas 2 o 3 píxeles de la de menor coeficiente de semejanza, este corrimiento el usuario no lo nota.

La única manera de resolver este problema es con un algoritmo que detecte regiones, que es extremadamente complicado y que ningún juego de este tipo usa. Aun así, siempre estos introducen errores que pueden atentar contra la estética del juego y que un diseñador no cometería, es por eso que este no se puede separar de la realización de los niveles y la solución que propone esta tesis lo incluye.

Aún si se pudiese implementar una herramienta que distribuya los objetos sobre el fondo sin errores, perdería el juego "Fernanda" toda lógica en el posicionamiento que consideramos importante debido a que este está orientado a niños.

## Capítulo 1: Fundamentación Teórica

---

La solución que propone esta tesis resuelve todos estos problemas de manera eficiente e incluye a un diseñador que será el que le otorgue la estética y el nivel de dificultad al juego. Los juegos de este tipo que hay en el mercado no han renunciado tampoco al trabajo de los diseñadores y aun así sus niveles son muy estáticos lo cual demuestra que nuestra solución es superior.

### Capítulo 2: Propuesta de Solución

#### Introducción

Este capítulo contiene una guía de observación de los juegos de objetos escondidos, haciendo énfasis en los conocidos popularmente como “Detective”, ya que “Fernanda” es muy similar a estos, pero con otros objetivos y otra orientación. También se realiza una encuesta con el objetivo de reafirmar los posibles problemas que afecten a “Fernanda”. Además, de la guía y la encuesta, el capítulo contiene una propuesta de solución para eliminar los problemas que presenta “Fernanda”, así como un pequeño epígrafe con algunas particularidades de los colores, ya que estos son tomados en cuenta para brindar una solución.

#### 2.1.- Teoría de Colores

*Quando el ojo ve un color se excita inmediatamente, y ésta es su naturaleza, espontánea y de necesidad, producir otra en la que el color original comprende la escala cromática entera. Un único color excita, mediante una sensación específica, la tendencia a la universalidad. En esto reside la ley fundamental de toda armonía de los colores...*

*Goethe, Teoría de los colores, p. 317*

Por características antes expuestas respecto a “Fernanda” de que es un juego orientado a niños, con ciertas concesiones lógicas ya explicadas y con características particulares tomadas de los juegos de objetos escondidos, a la hora de elaborar el algoritmo se tuvo en cuenta los colores de los objetos y los del fondo donde estos puedan ser colocados, con el objetivo de confundirlos un poco con el fondo donde se ubique, aunque en ocasiones un objeto que no tenga nada que ver con el fondo puede estar ubicado en otro lugar debido a su posición lógica, anteriormente declarada, en este caso por los creadores del algoritmo.

Por esta razón es bueno aclarar ciertas particularidades de los colores:

El color es una sensación que es percibida por los órganos visuales; está producida por los rayos luminosos y depende de su longitud de onda y de las características del órgano receptor.

El ojo humano sólo percibe el color cuando la iluminación es abundante.

Con poca luz vemos en blanco y negro. El color blanco resulta de la superposición de todos los colores, mientras que el negro es la ausencia de color.

La luz blanca puede ser descompuesta en todos los colores (espectro) por medio de un prisma. En la naturaleza esta descomposición da lugar al arco iris.

Es bien sabida por los psicólogos la influencia emocional que desencadenan los colores en el espíritu humano. Las respuestas emocionales varían enormemente dependiendo del color y de la intensidad de éste, así como de las diferentes combinaciones de colores que se pueden dar.

Normalmente cada color individual lleva asociado un conjunto de emociones y asociaciones de ideas que le son propios.

Las sensaciones que producen los colores dependen de factores culturales y ambientales, y muchas veces de los propios prejuicios del usuario. Además hay que sumar a esto que no todas las personas ven los colores de la misma forma, ya que hay personas que sólo pueden ver bien la gama azul / naranja, otras la roja / verde y otras degeneran a la gama blanco / negro. Incluso se perciben los colores de forma diferente con el ojo derecho que con el izquierdo. (9)

Es por esto que los diseñadores de “Fernanda” tuvieron en cuenta el amplio espectro de colores, dando una tonalidad clara al juego, ya sea en sus fondos u objetos, aunque se ven objetos oscuros como por ejemplo un televisor que por lo general siempre es negro. A partir de ahí una vez dado los colores a los objetos, entonces entra a jugar la ubicación de los mismos, inicialmente ya vienen ubicados por la decisiones que fueron tomadas para una primera versión.

Como parte de mejorar la calidad y entretenimiento del juego llega la hora de aportar algo más de dinamismo en el juego, siendo aquí cuando se propone el algoritmo de esta tesis.

### **2.2. Problemas detectados**

Tras un análisis exhaustivo de la mayoría de los videojuegos “Detective” que existen, se han detectado una serie de posibles problemas que afectan a “Fernanda”, puesto que el objetivo de este es diferente al objetivo que pudieran tener otros “Detectives” ya existentes, por ejemplo, Fernanda está orientado a niños y esto trae consigo que no deba tener el mismo rigor que tienen los juegos “Detective” más populares, como aquellos que muestran objetos con nombres poco comunes que quizás un niño no reconocería, o los objetos perfectamente solapados que retan a los mayores y hasta a los jóvenes a descubrir el misterio de los objetos ocultos, pero para un niño esto resultaría algo difícil. Problemas como los de lograr una posición distinta de los objetos o una posición lógica de los mismos son los que se pretende resolver a lo largo del desarrollo de la tesis, con la elaboración de un algoritmo inteligente

## Capítulo 2: Propuesta de Solución

que se aplique a “Fernanda” o a cualquier otro juego de objetos escondidos. Los problemas que se observaron son:

- Los fondos son por lo general de colores semejantes en todas sus regiones (de un color parduzco en la mayoría de los casos) lo que hace que todos los objetos se confundan con el fondo por doquiera que estén, dado que estos son también de este color por lo general.



Fig. 2.1. Imagen del videojuego “BigCityAdventureSF”, nótese como el fondo es de un color pardo en toda la imagen y que todos los objetos son por lo general de ese mismo color.

- La distribución de los objetos sobre el fondo en la mayoría de los juegos analizados es estática lo que provoca que cada vez que se juegue el mismo nivel los objetos aparecen en el mismo lugar, en algunos son muy pocos los objetos que cambian, por lo general son los objetos a seleccionar los que varían de una vez a otra.



Fig. 2.2. Imágenes del mismo nivel cargado dos veces diferentes, nótese como los objetos no cambian de lugar, lo que cambia es la lista de los objetos a seleccionar.

- La distribución de los objetos es caótica y no sigue ninguna lógica, lo cual no supone ningún problema, pero como el juego “Fernanda” es orientado a niños esto debería tenerse en cuenta dado que a ellos les podría causar algunas dudas y predisposiciones.



Fig. 2.3. Imagen del videojuego “MysteryPI”, nótese como los objetos no siguen lógica alguna lo cual provoca que se encuentre una botella de bebida alcohólica, comida y un bolo de juegos en un camión de bomberos.

En el caso de “Fernanda” existen los mismos problemas de ubicación estática y caótica de los objetos, al igual que en los tipos de juegos mostrados en las imágenes anteriores, pero a diferencia de estos se debe tener en cuenta que “Fernanda” está orientado a niños y por tanto posee menos objetos y los colores de fondo son más variados, en fin, es un juego de menor dificultad como se muestra en la fig 2.4.



Fig. 2.4 Primer nivel del videojuego "Fernanda".

### 2.3.- Resultados de la encuesta aplicada a los estudiantes de la Facultad 5 en la Universidad de las Ciencias Informáticas.

La encuesta es una técnica de recogida de información donde, por medio de preguntas escritas organizadas en un formulario impreso, (**Anexo 1**) se obtienen respuestas que reflejan los conocimientos, opiniones, intereses, necesidades, actitudes o intenciones.

Es esencial tener en cuenta que el objetivo de la encuesta es buscar información a través de preguntas directas e indirectas, las cuales se organizan con determinados requisitos metodológicos en un cuestionario. O sea, que la encuesta es la técnica, y el cuestionario es el instrumento a través del cual encuestamos a la población.

#### Selección del tipo de encuesta

Según la vía de obtención de la información en este caso se aplica una entrevista directa puesto que es hecha personalmente a los sujetos encuestados.

Según el **tipo de pregunta** estas se clasifican según su función en:

- de contenido
- de control
- de filtro

- colchón

**Preguntas de contenido:** Son las que se relacionan directamente con los objetivos de la investigación y la encuesta, ofreciendo información significativa directa para la evaluación de las variables e indicadores.

**Preguntas de control:** Su función es comprobar la consistencia y veracidad de las respuestas del sujeto. Para lograrlo, se redacta una pregunta de contenido con su correspondiente pregunta de control, que trata exactamente la misma cuestión, pero expresada en forma diferente. Ambas deben aparecer en lugares diferentes al realizar el ordenamiento del cuestionario. Además, hay que considerar la pertinencia de las preguntas de control, incluyéndolas sólo cuando son indispensables, para evitar el recargamiento del instrumento.

**Preguntas de filtro** Permiten al investigador saber si el sujeto conoce o no acerca de la temática tratada, si está en condiciones de brindar algún tipo de información significativa al respecto, lo que determina la pertinencia de incluir preguntas de contenido sobre ese aspecto. Habiendo utilizado para esta entrevista preguntas de contenido y de control.

También se utilizó el recurso de las preguntas Dicotómicas así como las Politómicas.

- **Dicotómicas:** Las respuestas se refieren a variables dicotomizadas o polarizadas, por lo que sólo existen dos posibilidades: SI o NO, VERDADERO o FALSO, DE ACUERDO o EN DESACUERDO, etc.
- **Politómicas:** Son preguntas de selección múltiple, donde se establecen varias posibilidades de respuesta.

Una vez realizada la encuesta se llena una tabla (**Anexo2**) con valores de veracidad, dándole valor uno a las respuestas positivas y valor dos a las respuestas negativas, en el caso de las preguntas dicotómicas, y para el caso de preguntas politómicas dándole valores de uno a cinco o de uno a la cantidad determinada de posibles respuestas.

Después de esto se hace un análisis de los indicadores medidos que más puedan aportar al objetivo de la encuesta que es obtener información respecto al comportamiento de los objetos en cuanto a su posición

## Capítulo 2: Propuesta de Solución

lógica y posición por colores, para esto se miden varios indicadores y se elaboran varias preguntas dirigidas a esclarecer criterios sobre los objetos, lo cual queda explicado es la siguiente tabla.

<b>Variable:</b> <b>Comportamiento inteligente en objetos</b>	<b>Indicadores</b>	<b>Contenido de las preguntas:</b> <ul style="list-style-type: none"> <li>• <b>Dirigido a esclarecer si los objetos:</b></li> </ul>
<b>Posición Lógica</b>	<ul style="list-style-type: none"> <li>• <b>Lugar en que se encuentra el Objeto(Piso, Mesa, Estante)</b></li> <li>• <b>Estructura del objeto (Traje ninja, Raqueta, pelota etc.).</b></li> </ul>	<ul style="list-style-type: none"> <li>• <b>Tienen correspondencia en cuanto a su ubicación en la escena.</b></li> <li>• <b>Tienen correspondencia lógica en cuanto a su forma.</b></li> </ul>
<b>Posición por colores</b>	<ul style="list-style-type: none"> <li>• <b>Solapamiento del objeto Parcialmente</b></li> <li>• <b>Solapamiento del objeto Completamente</b></li> <li>• <b>Equivalencia de colores</b></li> </ul>	<ul style="list-style-type: none"> <li>• <b>Está posicionado parcialmente encima de otro.</b></li> <li>• <b>Está posicionado totalmente encima de otro.</b></li> <li>• <b>Tienen colores semejantes en cuanto a su ubicación parcial o total.</b></li> <li>• <b>Presentan colores confusos.</b></li> <li>• <b>Presentan colores llamativos.</b></li> </ul>



--	--	--

Fig. 2.5 Tabla de indicadores para las Preguntas

Como resultado de ese análisis se obtiene en cuanto a la **“Tabla de Frecuencia para Rol”** que la misma muestra el número de veces que ha ocurrido cada valor de Rol, así como los porcentajes y estadísticas acumuladas, obteniendo en 5 filas del fichero de datos, Rol es igual a Analista. Esto representa 25,0% de los 20 valores del fichero. Sucesivamente se hacen análisis como este para cada una de las preguntas.

**“Tabla de Frecuencias para Colores de Fondo”** para esta tabla se muestra el número de veces que ha ocurrido cada valor de Colores de Fondo, así como los porcentajes y estadísticas acumuladas. Esto representa 85,0% de los 20 valores del fichero. **(Anexo 3)**

**“Tabla de Frecuencias para Correspondencia de los colores”** Esta analiza el número de veces que ha ocurrido cada valor de correspondencia de los colores, así como los porcentajes y estadísticas acumuladas. Esto representa 60,0% de los 20 valores del fichero. **(Anexo 4)**

**“Tabla de Frecuencias para Fondos con tonalidades confusas”** Mientras que aquí se muestra el número de veces que ha ocurrido cada valor de fondos con tonalidades confusas, así como los porcentajes y estadísticas acumuladas. Esto representa 80,0% de los 20 valores del fichero.

**“Tabla de Frecuencias para Fondos Oscuros”** Esta tabla muestra el número de veces que ha ocurrido cada valor de fondos oscuros, así como los porcentajes y estadísticas acumuladas. Esto representa 75,0% de los 20 valores del fichero. **(Anexo 5)**

**“Tabla de Frecuencias para Presencia de Posición Lógica”** Muestra el número de veces que ha ocurrido cada valor de presencia de posición lógica, así como los porcentajes y estadísticas acumuladas. Esto representa 75,0% de los 20 valores del fichero. **(Anexo 6)**

**“Tabla de Frecuencias para Semejanza de colores”** Esta tabla muestra el número de veces que ha ocurrido cada valor de semejanza de colores, así como los porcentajes y estadísticas acumuladas. Esto representa 100,0% de los 20 valores del fichero. **(Anexo 7)**

“**Tabla de Frecuencias para Tipo de solapamientos**” Se muestra el número de veces que ha ocurrido cada valor de tipo de solapamientos, así como los porcentajes y estadísticas acumuladas. Esto representa 15,0% de los 20 valores del fichero. **(Anexo 8)**

“**Tabla de Frecuencias para Variación de Posición**” Esta tabla muestra el número de veces que ha ocurrido cada valor de variación de posición, así como los porcentajes y estadísticas acumuladas. Esto representa 5,0% de los 20 valores del fichero. (10) **(Anexo 9)**

Ejemplo de análisis de tablas:

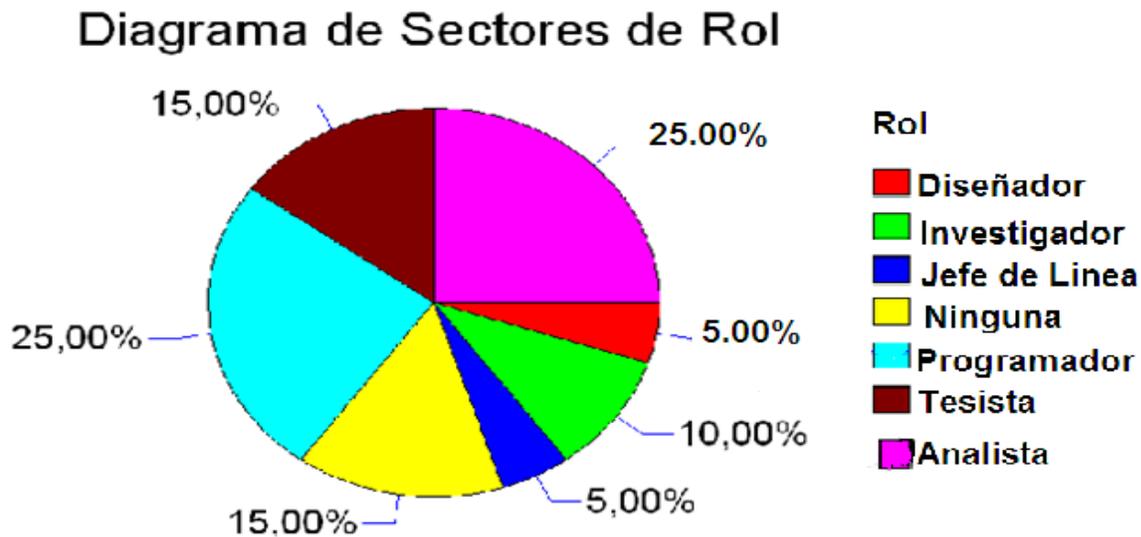


Fig. 2.6 Tabla Frecuencia para Rol.

Cada uno de estos análisis es realizado con el Statgraphics, software que permite hacer este tipo de análisis estadístico una vez que se le introducen los datos pertinentes.

Una vez obtenido todo este análisis, se toma la decisión de elaborar un algoritmo que resuelva en el juego de “Fernanda”, como también en los demás videojuegos de objetos escondidos que deseen aplicarlo, las deficiencias señaladas por los encuestados y que son aplicadas a “Fernanda” en materia de mejoras.

### 2.4.-Propuesta de solución

A raíz de estos problemas se decidió elaborar una solución que varíe la localización de los objetos para cada carga del nivel sin que se deje de confundir en el fondo y que permita la utilización de fondos y objetos con más variedad de colores además de dar la posibilidad de otorgarles alguna lógica al montaje.

La propuesta de solución establece algunas reglas como:

- La creación de un archivo de configuración por cada fondo donde el diseñador de niveles organiza los nombres de los objetos y las ubicaciones por categorías.
- Que el fondo tenga el mismo nombre de este archivo de configuración.
- Que las imágenes de los objetos, la imagen del fondo y el archivo de configuración se encuentren en el mismo directorio.

También se puede dividir en tres procesos fundamentales:

- Obtener la información necesaria del archivo de configuración.
- Realizar la asignación de coordenadas sobre el fondo a cada objeto.
- Mostrar el resultado, es decir, los objetos ubicados sobre el fondo.

### 2.5.- El archivo de configuración

El archivo de configuración posee una estructura que permite agrupar los objetos a mostrar y los posibles puntos donde hacerlo por categoría, esta categoría puede ser un color específico, lógica o ambos.

La estructura es la siguiente:

pasos	# de línea	datos
1	1	Cantidad de categorías definidas en el archivo.
2	2	Nombre de la primera categoría.
3	3	Cantidad de posibles ubicaciones (coordenadas sobre el

## Capítulo 2: Propuesta de Solución

		fondo) (n).
4	4..4+n	Coordenadas separadas por “;” cada una en una línea.
5	4+n+1	Cantidad de objetos pertenecientes a esta categoría (m).
6	4+n+1..4+n+1+m	Nombre de los objetos con su extensión.
Los pasos del 2 al 6 se repiten según la cantidad de categorías que se establezcan.		

En este archivo siempre deben existir mayor o igual cantidad de posibles ubicaciones que de objetos por cada categoría.

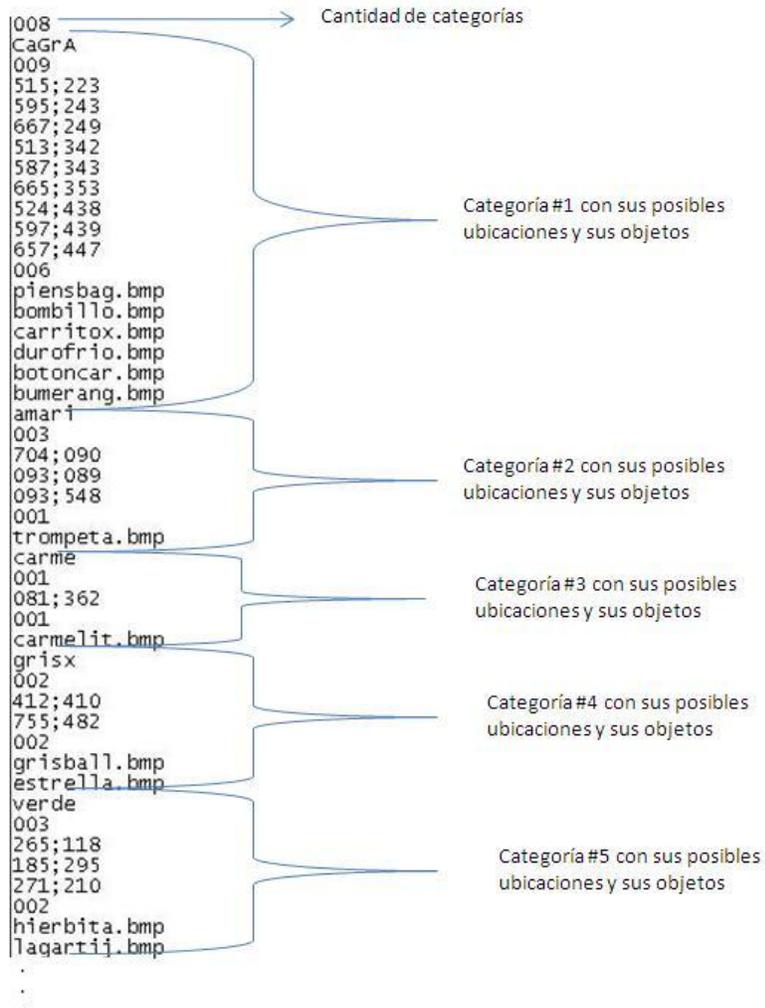


Fig. 2.7. Ejemplo de archivo de configuración.

### 2.6.-Algoritmo de asignación

A partir de la información que se extrae del archivo crea una lista de categorías y cada una de ellas tiene una lista de objetos y una de posibles ubicaciones.

A continuación en cada una de las categorías se le asigna a cada objeto un par de coordenadas que se escoge de forma aleatoria según su posición en la lista y cuidando que no se repita para otro objeto.

X1;Y1	X2;Y2	X3;Y3
Dir\hasta\imagen\objetoA	Dir\hasta\imagen\objetoB	Dir\hasta\imagen\objetoC

Fig. 2.8 Ejemplo de asignación para cada categoría.

Se crea con las listas de asignación de cada categoría una lista global de objetos y coordenadas donde se reúnen al fin todos los objetos de la imagen (**Anexo 10**), dicha lista global se organiza de forma descendente usando cualquier método de ordenamiento por tamaño del objeto para prepararlos para dibujarlos sobre el fondo, esto se hace para que en caso de que exista superposición de objetos los más grandes se dibujen primero y no oculten a los más pequeños.

Además de asignarle una coordenada a cada objeto, también se le asigna un valor “alfa” para un mejor ocultamiento de estos y con ello un aumento de complejidad en los niveles de “Fernanda”.

**Alpha blending** es una combinación convexa de dos colores que permite efectos de transparencia en gráficos de computadoras. El valor de alpha en el código del color va desde 0.0 a 1.0, donde 0.0 representa un color completamente transparente y 1.0 representa un color completamente opaco.

El valor del color resultante cuando el color “Valor1” con un valor alpha “ $\alpha$ ” es dibujado sobre un fondo opaco de color “Valor0” está dado por:

$$\text{Valor} = (1 - \alpha) \text{Valor0} + \alpha \text{Valor1}$$

El componente alpha puede ser usado para mezclar los componentes rojo, verde y azul por igual, como en el formato 32bit RGBA, o, alternativamente pueden existir tres valores alpha especificados que correspondan a cada uno de los colores primarios para el filtrado espectral del color.

Los objetos pueden aparecer como el vidrio, como el agua, o puede permitir ver a través de explosiones. (11)

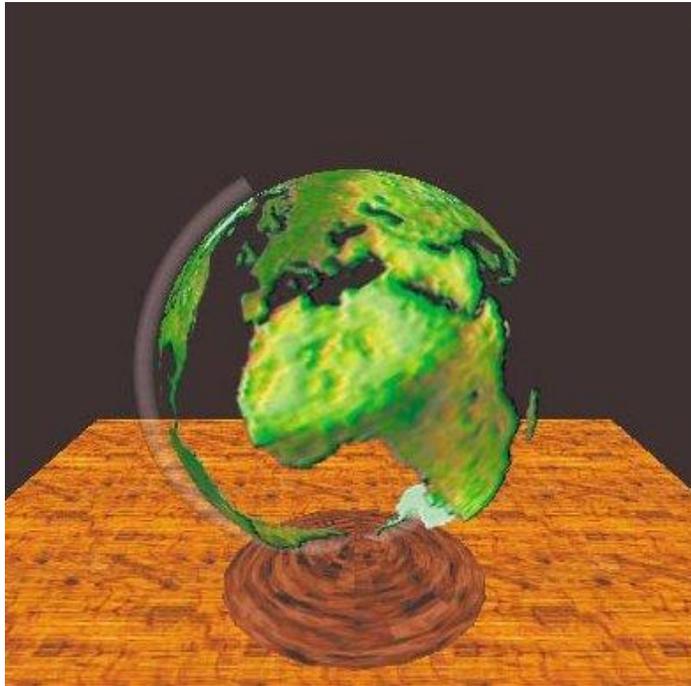


Fig. 2.9 Imagen donde se observa el Alpha Blending actuando en la creación de la transparencia.

### 2.7.- Patrón de Arquitectura seleccionado

#### Arquitectura clásica de tres capas

Una arquitectura común de los sistemas de información que abarca una interfaz para el usuario y el almacenamiento persistente de datos se conoce con el nombre de **arquitectura de tres capas** [Gartner95]; podemos observarla en la figura 2.4. He aquí una descripción clásica de las capas verticales:

1. **Presentación:** ventanas, reportes, etcétera.
2. **Lógica de aplicaciones:** tareas y reglas que rigen el proceso.
3. **Almacenamiento:** mecanismo de almacenamiento persistente.

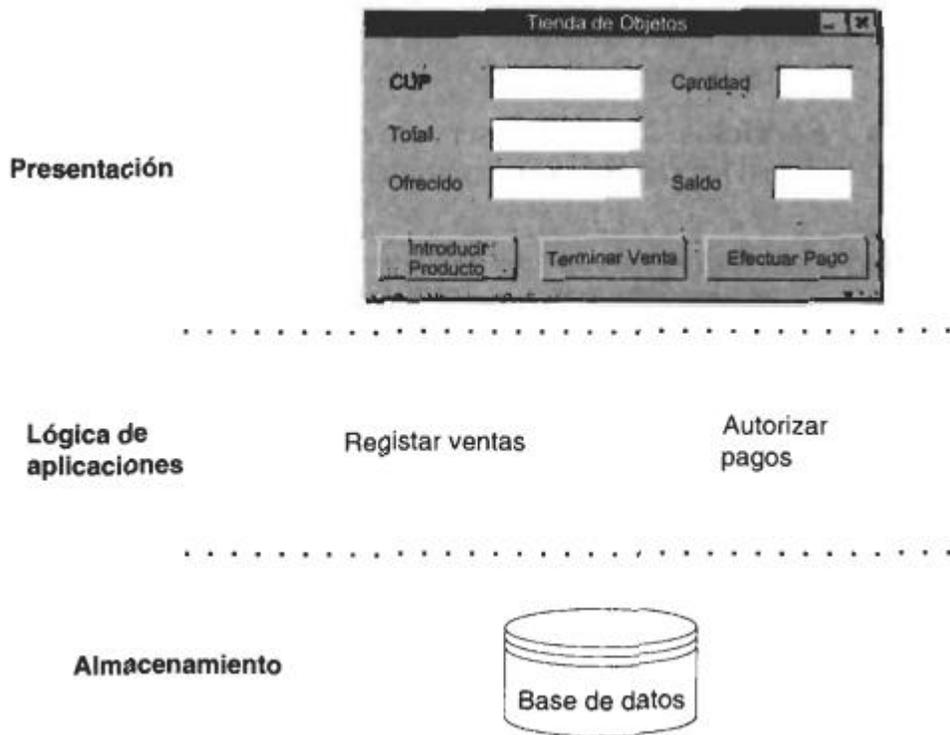


Fig. 2.10 Vista clásica de una arquitectura de tres capas.

La calidad tan especial de la arquitectura de tres capas consiste en aislar la lógica de la aplicación y en convertirla en una capa intermedia bien definida y lógica del software. En la capa de presentación se realiza relativamente poco procesamiento de la aplicación; las ventanas envían a la capa intermedia peticiones de trabajo. Y este se comunica con la capa de almacenamiento del extremo posterior. (12)

Este es el patrón que se utiliza en el demo la propuesta de solución, donde habrá una interfaz de presentación, en la capa de lógica de aplicaciones está dada por reglas y la de almacenamiento estará conformada por un archivo de configuración.

### Capítulo 3: Análisis de resultados

#### Introducción

Luego de haberse propuesto en el capítulo anterior una solución al problema planteado que resuelve satisfactoriamente la distribución de objetos que presenta “Fernanda” y la mayoría de los juegos de objetos escondidos, además de otros que se detectaron; se decide desarrollar su implementación, por lo que en este capítulo se expone el algoritmo bien detallado y luego se muestran los resultados que se obtienen de este.

#### 3.1.- Pseudocódigo de la solución propuesta

Para un mejor entendimiento de la forma de solucionar que se propone en este trabajo se explicará a continuación el pseudocódigo de dicha solución.

#### Función Leenúmero

Esta función es utilizada para leer números enteros, lo cual será necesario hacer dentro de otras funciones de mayor complejidad.

```
funcion leenumero (var fs:TFileStream):entero;  
var  
a:string;  
inicio  
  a="";  
  fs.Leer(a,3);  
//lee un numero del archivo, en este caso de 3 cifras y se lo asigna a la variable “a”  
  Resultado:=a;  
fin;
```

### Función leecoordenadas

Esta función lee los valores de las **x** y los de las **y** de cada coordenada que aparezcan en la lista de posiciones que posea una categoría. Para esto se utiliza la función antes mencionada como leenumero.

```
función leecoordenadas (var fs:TFileStream;cant_posiciones:entero):array de array de entero;  
var  
  a:string;  
  b:array de array de entero;  
  l:entero;  
inicio  
  SetLongitud(b,2,cant_posiciones);  
  //le da al arreglo dinámico bidimensional "b" las dimensiones "2xn" siendo "n" la  
  //cantidad de puntos a leer  
  desde l:=0 hasta cant_posiciones-1 hacer  
    inicio  
      b[0,l]:=leenumero(fs);  
      SaltaEspacio(fs);  
      b[1,l]:=leenumero(fs);  
      saltalineafs;  
    fin;  
  //en este ciclo se leen todos los valores "X" y "Y" de una categoría y se guardan en el  
  //arreglo "b".  
  Resultado:= b;  
fin;
```

### **Función leenombre**

Esta función se utiliza para leer nombres, la cual se llamará dentro de otras funciones de mayor complejidad.

```
función leenombre (var fs:TFileStream):string;  
var  
  a:string;  
inicio  
  a:="";  
  fs.Leer(a,12);  
//lee del archivo un nombre de un archivo de imagen y se lo asigna a la variable "a"  
  Resultado:=a;  
fin;
```

### **Función leeobjeto**

En esta función se leen todos los nombres de la lista de objetos de una categoría usando la función anterior de leer nombres y lo devuelve en una lista de "string".

```
funcion leeobjetos(var fs:TFileStream;cant_obj:entero):array de string;  
var  
  l:entero;  
  b:array de string;  
inicio  
  SetLongitud(b,cant_obj);  
//le da al arreglo dinámico "b" la dimensión "n" que es la cantidad de objetos a leer del archivo  
desde l:=0 hasta cant_obj-1 hacer  
  inicio  
    b[l]:=leenombre(fs);  
    saltalineafs;  
  fin;  
//en este ciclo se leen todos los nombres de objetos y se guardan en la variable "b"  
  Resultado:=b;  
fin;
```

### Estructura de datos

Se declaran las siguientes estructuras de datos para realizar las asignaciones de coordenadas a los objetos.

```
final=record
```

```
X:entero;
```

```
Y:entero;
```

```
dir:string;
```

```
fin;
```

```
//esta estructura de datos representa como tal una asignación de un objeto a un par de  
//coordenadas "XY" para su posterior ubicación
```

```
asignacion_cat=record
```

```
lista_asig:array de final;
```

```
cant_obj:entero;
```

```
fin;
```

```
//esta estructura de datos representa una categoría ya cuyos objetos ya han sido  
//asignados a un par de coordenadas "XY" por tanto posee una lista de asignaciones y  
//el tamaño de esta
```

### Función asignar

Esta función se encarga de asignarle a cada objeto de una categoría una posición aleatoria dentro de la misma categoría, cuidando que esta no se repita para otro objeto.

```
function asignar(cant_huecos:entero;lista_huecos:array de array de entero; cant_obj:entero;
```

```
lista_obj:array de string):asignacion_cat;
```

```
var
```

```
asig:asignacion_cat;
```

```
//para guardar una categoría asignada
```

```
l:entero;
```

```
Pos_a_probar:entero;
```

```
cont:entero;
```

```
inicio
```

```
setLongitud(asig.lista_asig,cant_obj);
```

```
//se le asigna al arreglo dinamico "asig.lista_asig" un tamaño igual a la cantidad de objetos que
//se asignarán.
cont:=0;
asig.cant_obj:=cant_obj;
repetir
  //ciclo que se encargara de recorrer la lista de objetos asignándole a cada uno un par de
  //coordenadas "XY" único.
  Pos_a_probar:=Random(cant_huecos);
  //genera un numero aleatorio entero de "0" a "cant_huecos" y lo guarda en "pos_a_probar".
  si cont=0 entonces
    inicio
      asig.lista_asig[cont].X:=lista_huecos[0,Pos_a_probar];
      asig.lista_asig[cont].Y:=lista_huecos[1,Pos_a_probar];
      asig.lista_asig[cont].dir:=lista_obj[cont];
      cont:=cont+1;
      //si es el primer objeto al que se le hace la asignación, asigno sin más e incremento
      //en 1 el contador de los objetos ya asignados
    fin
  sino
    inicio
      desde l:=0 hasta cont hacer
        //ciclo que prueba que el par de coordenadas "XY" no haya sido asignado
        //previamente
        inicio
          si ((lista_huecos[0,Pos_a_probar] = asig.lista_asig[l].X)and (lista_huecos[1,Pos_a_probar] =
asig.lista_asig[l].Y)) entonces
            break;
            //si fue asignado fin del ciclo
          sino
            inicio
              si (l=cont) entonces
                inicio
                  asig.lista_asig[cont].X:=lista_huecos[0,Pos_a_probar];
```

```
asig.lista_asig[cont].Y:=lista_huecos[1,Pos_a_probar];  
asig.lista_asig[cont].dir:=lista_obj[cont];  
cont:=cont+1;  
//si las coordenadas "XY" no ha sido asignado a ningún objeto anterior se  
//realiza la asignación  
fin;  
fin;  
fin;  
fin;  
  
hastaque cont=cant_obj;  
//hasta que a todos los objetos le hayan sido asignados a un par de coordenadas  
Resultado := asig;  
fin;
```

### Función asigna\_cat

Esta función realiza todo el proceso para una categoría, lee sus datos usando funciones como “leenumero”, “leecoordenadas” y “leobjetos” para luego realizar la asignación usando la función “asignar” y devuelve un resultado de tipo “asignacion\_cat”.

```
funcion asigna_cat(var fs:TFileStream):asignacion_cat;  
var  
cant_huecos, cant_obj:entero;  
lista_huecos:array de array de entero;  
lista_obj:array de string;  
asig:asignacion_cat;  
inicio  
cant_huecos:=leenumero(fs);  
//lee numero desde archivo y lo guarda en "cant_huecos" que no es más que la  
//cantidad de coordenadas  
saltalineafs(fs);  
lista_huecos:=leecoordenadas(fs,cant_huecos);  
//lee todos los pares de coordenadas y las almacena en "lista_huecos"
```

```
cant_obj:=leenumero(fs);  
//lee un numero desde archivo y lo guarda en "cant_obj", este valor es la cantidad de  
//objetos de la categoría  
saltalineaf(fs);  
lista_obj:=leeobjetos(fs,cant_obj);  
//lee la lista de objetos y los guarda en "lista_obj"  
asig:=asignar(cant_huecos,lista_huecos,cant_obj,lista_obj);  
//realiza asignación a la categoría leída y la devuelve en "asig"  
Resultado:=asig;  
fin;
```

### **Función lee\_nombre\_cat**

Esta función lee el nombre de una categoría, la cual se usa dentro de la función que lee y asigna todas las categorías.

```
function lee_nombre_cat(var fs:TFileStream):string;  
var  
  a:string;  
inicio  
  a:"";  
  fs.Leer(a,5);  
//lee el nombre de una categoría y lo guarda en "a"  
  saltalineaf(fs);  
  Resultado:=a;  
fin;
```

### **Función imagedir**

Esta función se encarga de conformar la dirección hasta cada objeto a partir del nombre del mismo y la dirección hasta la carpeta que los contiene.

```
función imagedir(directorio:string;nombre_image:string):string;  
var  
  direccion:string;  
inicio  
  direccion:=directorio+nombre_image;  
  //con el nombre del objeto y la dirección de la carpeta que los contiene construye la  
  //cadena de caracteres que será la dirección hasta ese objeto y lo devuelve  
  Resultado:=direccion;  
fin;
```

### **Estructura de datos**

Estos tipos de datos se usarán en la próxima función.

```
categoria=record  
//record es lo mismo que un struct  
  lista_asignados:asignacion_cat;  
  nombre_cat:string;  
  end;  
//estructura de datos que representa una categoría ya asignada con su nombre  
  
lista_categorias= array of categoria  
//estructura de datos que representa una lista de categorías ya asignadas
```

### Función main

Esta función es la que va a leer del archivo los datos de todas las categorías y usando la función “leenumero” y “lee\_nombre\_cat” y realizara la asignación categoría por categoría usando la función “asigna\_cat”. A continuación construye la lista global con todas las asignaciones de las categorías para realizar el ordenamiento y lo devuelve en la variable “listafinal”.

```
funcion main (var fs:TFileStream):array de final;  
var  
  listafinal:array de final;  
  cant_cat:entero;  
  I,J:entero;  
  cat:lista_categorias;  
  punt:entero;  
inicio  
  SetLongitud(listafinal,0);  
  //se inicializa tamaño del arreglo “listafinal” en 0  
  punt:=0;  
  cant_cat:=leenumero(fs);  
  //lee la cantidad de categorías que contiene el archivo  
  SetLongitud(cat,cant_cat);  
  //le asigna un tamaño al arreglo dinamico de categorías igual a la cantidad de  
  //categorías del archivo  
  saltalineafs(fs);  
desde I:=0 hasta cant_cat-1 hacer  
  inicio  
    cat[I].nombre_cat:=lee_nombre_cat(fs);  
    cat[I].lista_asignados:=asigna_cat(fs);  
  fin;  
  //ciclo que recorre las todas las categorías realiza la asignación de sus objetos a sus  
  //coordenadas  
  desde I:=0 hasta cant_cat-1 hacer  
    inicio  
      SetLongitud(listafinal,length(listafinal)+cat[I].lista_asignados.cant_obj);
```

```
si l<>0 entonces
punt:=punt +cat[l-1].lista_asignados.cant_obj;

desde J:=punt hasta cat[l].lista_asignados.cant_obj-1+punt hacer
inicio
  listafinal[J].X:=cat[l].lista_asignados.lista_asig[J-punt].X;
  listafinal[J].Y:=cat[l].lista_asignados.lista_asig[J-punt].Y;
  listafinal[J].dir:=imagedir(directorio,cat[l].lista_asignados.lista_asig[J-punt].dir);
fin;
fin;
//ciclo que recorre la lista de categorías y va construyendo con los objetos y el par de
//coordenadas asignados a estos de cada categoría, una lista que agrupa todas las
//asignaciones de todas las categorías (lista global)
Resultado:=listafinal;
fin;
```

### 3.2.- Resultados

Al probar la funcionalidad del algoritmo antes propuesto en uno de los niveles de “Fernanda”, se obtienen los resultados esperados satisfactoriamente. **(Anexo 11)**

Uno de estos resultados es el **posicionamiento lógico** de ciertos objetos, los cuales debido a la forma en que están dibujados, deben aparecer en zonas donde concuerden lógicamente. Ejemplo de este corolario se muestra en la figura 3.1, donde se puede observar que los objetos tales como la raqueta, el traje de ninja y el gorro son objetos que por la forma en que se ven solo deben aparecer en lugares que parezcan estar colgados.



Fig. 3.1 Ejemplo de los objetos que deben posicionarse en lugares que parezcan estar colgados.

Además del posicionamiento lógico anteriormente mencionado se puede observar en la figura 3.2 otro resultado, donde aparece un gato y un perro, los cuales por lógica deben estar ubicados en el suelo, es por esta razón que los mencionados objetos solo varían de posición únicamente en dicha región.



Fig. 3.2 Ejemplo de posicionamiento lógico

Otro de los resultados es el superposicionamiento completo o solapamiento de imágenes, el cual se puede ver cuando un objeto queda totalmente encima de otro, siempre quedando encima el objeto más pequeño, ya que se ha previsto con el ordenamiento burbuja, lo cual se ha explicado antes; logrando que no aparezcan primero los objetos más pequeños, aunque el diseñador es quien decidirá cuales quedarán sobre otros. Nótese entonces en la figura 3.4 cómo el pajarito azul pequeño queda totalmente encima de la gorra azul-carmelita ubicada en el estante.



Fig. 3.3 Ejemplo de solapamiento de imágenes

También se observa el resultado de **superposicionamiento parcial**, el cual se puede notar en la figura 3.4, donde parte del bumerán queda sobre el carrito y el pico del tucurú queda encima del sombrerito azul.



Fig. 3.4 Ejemplo del superposicionamiento parcial.

### CONCLUSIONES

El presente trabajo titulado “Algoritmo de distribución inteligente de objetos en entornos bidimensionales”:

Aporta la base para la comprensión de los temas que se tratan en la investigación. Se analizan las principales particularidades de los juegos, así como algunos algoritmos que se utilizan en los géneros puzzle y otros juegos en general.

Los textos, documentos y sitios consultados han ofrecido los aspectos necesarios sobre los temas que estos abordan, considerándose la bibliografía satisfactoria.

Se enmarcan los aspectos claves del objeto de estudio, lo que conlleva a tener una visión sobre el camino a tomar para resolver la problemática existente. Se valoraron las tendencias actuales, seleccionando las herramientas más utilizadas y que pudieran aportar a la realización de la solución a proponer.

El análisis realizado permite entender aspectos importantes para la confección de un algoritmo inteligente para videojuegos 2D de objetos escondidos, llegando a la conclusión de que no es necesario utilizar ninguna de las técnicas o algoritmos de IA, sino que se da una nueva solución eficiente y óptima para arreglar los problemas existentes en este tipo de videojuego.

Se expone la solución técnica para resolver el problema planteado en dicho trabajo de tesis. Se define la estructura del algoritmo que se va a implementar y con esto quedan sentadas las bases para implementación.

Finalmente se muestra el algoritmo implementado.

### RECOMENDACIONES

A partir de las experiencias obtenidas en el desarrollo del trabajo y con vistas de lograr un aprovechamiento óptimo del resultado alcanzado se recomienda:

- Incluir este algoritmo en el juego "Fernanda" para mejorar la ubicación de los objetos.
- Realizar un editor del archivo de configuración lo cual posibilitaría la rápida realización de nuevos niveles.
- Agregar al módulo de visualización del juego "Fernanda" una funcionalidad que permita otorgarle a los objetos un nivel de transparencia con respecto al fondo, usando la técnica "Alpha Blending".
- Ordenar la lista global de asignaciones de forma descendente mediante un método de ordenamiento.
- Fomentar la investigación en aquellos proyectos en los podría ser aplicado el algoritmo.

### REFERENCIAS

#### Trabajos citados

1. **Fernández, Luis A García.** *Revista de Divulgación Científica y Tecnológica de la Universidad Veracruzana.* 3, México : s.n., 2004, Vol. XVII.
2. **Figueroa, Roberto Albornoz.** *¿Como empezar en el Desarrollo de Videojuegos?* 2006.
3. **Graells, Pere Marquéz.** Los Videojuegos. *UAB.* [En línea] 27 de 8 de 2008. [Citado el: 6 de 1 de 2009.] <http://www.pangea.org/peremarques/videojue.htm>..
4. **Wexler, James.** *Artificial Intelligence in Games: A look at the smart behind Lionhead Studio's "Black and White" and where it can and will go in the future.* Nueva York : s.n., 2002.
5. **Samuel, A.** *Some Studies in Machine Learning Using the Game of Checkers.* s.l. : IBM Journal, 1995. 3.
6. **Pérez, José Carlos Cortizo.** Weblogs: Sistemas Inteligentes. [En línea] 24 de 4 de 2008. [Citado el: 4 de 2 de 2009.] [http://weblogs.madrimasd.org/sistemas\\_inteligentes/archive/2008/04/24/89956.aspx](http://weblogs.madrimasd.org/sistemas_inteligentes/archive/2008/04/24/89956.aspx).
7. **Martínez, Tamara y Antunes, Yaima.** *Propuesta de Técnica para la Comunicación entre Agentes Virtuales.* Habana : s.n., 2008.
8. **Molina, Orlando Arboleda.** *Máquinas de Estado.* 2008.
9. **Moreno, L.** *Teoría de Colores.* 2004.
10. **Simons, Beatriz Castellanos.** *La encuesta y la entrevista en la investigación educativa.* La Habana : s.n., 1998.
11. **Cardona, A; Storti, M y Zuppa, C.** *TEXTURAS TRANSPARENTES PARA VISUALIZACION DE MALLAS.* San Luis : s.n., 2008, Vol. XXVII.
12. **Larman, Craig.** *UML y Patrones: Introducción al análisis y diseño orientado a objetos.* Mexico,D.F : Dawn Speth White, 2000. 970-1 7-0261-1.

---

---

## BIBLIOGRAFÍA

1. *AI Game Dev*. [Online] [Cited: 10 2, 2007.] <http://aigamedev.com/>.
2. AMANR, R Y YOUNG, R.M. *Artificial Intelligence and Interactive Entertainment*.
3. Árbol de decisión. [2009]. Disponible en: <http://ciberconta.unizar.es/Biblioteca/0007/arboles.html>
4. BÉJAR, J. *Árboles de Decisión. Curso 2003/2004*.
5. BUCKLAND, M y COLLINS, M. *AI Techniques for Game Programming*. s.l. : Thomson Course Technology, 2002.
6. CHAMPANDARD, A. *AI.game.development*. New Riders, 2003.
7. CHARLES, D. *Biologically Inspired Artificial Intelligence for Computer Games*. 2008.
8. FIGUEROA, R.A. *¿Como empezar en el Desarrollo de Videojuegos?* 2006.
9. FUNGE, D.J. *Artificial Intelligence for Computer Games*. A K Peters, 2004.
10. GRAELLS, P M. Los Videojuegos. UAB. [En línea] 27 de 8 de 2008. [Citado el: 6 de 1 de 2009.] <http://www.pangea.org/peremarques/videojue.htm>.
11. LARMAN, CRAIG. *UML y Patrones: Introducción al análisis y diseño orientado a objetos*. Mexico,D.F : Dawn Speth White, 2000. 970-1 7-0261-1.
12. MARTÍNEZ, T Y A, Y. *Propuesta de Técnica para la Comunicación entre Agentes Virtuales*. Habana : s.n., 2008.
13. MCGONIGAL, J. *All Game Play is Performance: The State of the Art Game*. 2005.
14. MOLINA, O.A. *Máquinas de Estado*. 2008.
15. MUNAKATA, T. *Fundamentals of the New Artificial Intelligence. Neural, Evolutionary, Fuzzy and More*. 2008.
16. O'REILLY. *AI for Game Developers*. 2004.
17. PAU, A.F. *IA-Algoritmos de Juegos*, 2008
18. PÉREZ, J.C.C. Weblogs: Sistemas Inteligentes. [En línea] 24 de 4 de 2008. [Citado el: 4 de 2 de 2009.] [http://weblogs.madrimasd.org/sistemas\\_inteligentes/archive/2008/04/24/89956.aspx](http://weblogs.madrimasd.org/sistemas_inteligentes/archive/2008/04/24/89956.aspx).
19. POTENVIA, P. #2-Pathfinding is not "A- start". CTO Kynogon.
20. FERNÁNDEZ, L.A G. *Revista de Divulgación Científica y Tecnológica de la Universidad Veracruzana*. 3, México : s.n., 2004, Vol. XVII.
21. SAMUEL, A. *Some Studies in Machine Learning Using the Game of Checkers*. s.l. : IBM Journal, 1995. 3.

- 22. SIMONS, BEATRIZ CASTELLANOS.** *La encuesta y la entrevista en la investigación educativa.* La Habana : s.n., 1998.
- 23. Sincronización por Frame Rate.** *VJuegos.*[Citado el : Mayo 02,2008.]  
[http://prog.vjuegos.org/index.php?option=com\\_content&task=view&id=124&Itemid=148](http://prog.vjuegos.org/index.php?option=com_content&task=view&id=124&Itemid=148)
- 24. SMED, J Y HAKONEN, H.** *Algorithms and Networking for Computer Games.* Finlandia : University of Turku, 2006.
- 25. MORENO, L.** *Teoría de Colores.* 2004.
- 26. ALBERTO, C; STORTI, M ; ZUPPA, C.** *TEXTURAS TRANSPARENTES PARA VISUALIZACION DE MALLAS.* San Luis : s.n., 2008, Vol. XXVII.
- 27. WEXLER, JAMES.** *Artificial Intelligence in Games: A look at the smart behind Lionhead Studio's "Black and White" and where it can and will go in the future.* Nueva York : s.n., 2002.
- 28.** *Los videojuegos.* Giga, 2006.

### APÉNDICE

#### Glosario de términos

**Inteligencia Artificial:** Se denomina inteligencia artificial a la rama de la ciencia informática dedicada al desarrollo de agentes racionales no vivos.

\*\*\*\*\*

**Escena:** Ambiente, conjunto de circunstancias espaciales y temporales en que tiene lugar una situación o un hecho.

\*\*\*\*\*

**Bidimensional:** Adjetivo que tiene dos dimensiones, altura y anchura: imagen bidimensional.

\*\*\*\*\*

**Ítems:** Hace referencia a datos.

\*\*\*\*\*

**Modelar:** Configurar o conformar algo no material. Ajustarse a un modelo.

\*\*\*\*\*

**Distribución:** Dar a algo el destino conveniente.

\*\*\*\*\*

**Dinamismo:** Modos del movimiento.

\*\*\*\*\*

**Algoritmo:** Conjunto de pasos o instrucciones finito que se deben seguir para realizar una determinada tarea.

\*\*\*\*\*

**Simular:** Sinónimo de representar.

\*\*\*\*\*

**Puzzle:** Juego consistente en formar una determinada figura con piezas sueltas (rompecabezas) o categoría de juegos de objetos escondidos.

\*\*\*\*\*

**Sprite:** Tipo de mapa de bits dibujados en la pantalla de ordenador por hardware gráfico especializado. Generalmente son utilizados para producir una animación, como un personaje corriendo, alguna expresión facial o un movimiento corporal.

\*\*\*\*\*

**GUI (Interfaz Gráfica de Usuario):** tipo de interfaz de usuario que utiliza un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.

\*\*\*\*\*

**HUD:** La información que se muestra en la pantalla durante un juego.

\*\*\*\*\*

## ANEXOS

### Anexo: 1 Entrevista a estudiantes de la Facultad 5 de la Universidad de las Ciencias Informáticas.

Estudiante: Se reconoce su afición por los videojuegos, a la vez que es identificado como jugador habitual, por lo que son muy valiosas sus opiniones acerca de los aspectos que indaga la presente encuesta. El propósito de la misma, es obtener información sobre las carencias y dificultades de los videojuegos del género: objetos escondidos.

#### Datos generales del entrevistado:

Proyecto al que pertenece: \_\_\_\_\_ Rol que desempeña: \_\_\_\_\_

#### Interrogantes:

1. ¿Al disfrutar de los videojuegos conocidos popularmente como “detectives” y volver intentar el mismo nivel, de acuerdo a la forma que describe un objeto, ¿cree usted que el mismo presente una posición lógica?

\_\_\_\_\_ SI  
\_\_\_\_\_ NO

2. ¿Está Ud. de acuerdo con que los objetos varían su posición dado un cambio de escena?

\_\_\_\_\_ Lo apruebo totalmente  
\_\_\_\_\_ Simplemente lo apruebo  
\_\_\_\_\_ estoy indeciso  
\_\_\_\_\_ Simplemente lo desapruebo  
\_\_\_\_\_ Lo desapruebo totalmente

3. \_\_\_\_\_ Los fondos presentados en las escenas son oscuros.

1    2    3    4    5

1. Totalmente de acuerdo
2. En parte de acuerdo
3. No sé qué opinar
4. En parte en desacuerdo
5. Totalmente en desacuerdo

4. Considera que los colores, no se corresponden con los colores reales de los objetos.

\_\_\_\_\_ SI  
\_\_\_\_\_ NO

5. Existen solapamientos parcial o total de los objetos.

- Parcial
- Total
- Parcial y Total

6. ¿Cree usted que los objetos solapados se ubican teniendo en cuenta colores semejantes?

- Lo apruebo totalmente
- Simplemente lo apruebo
- estoy indeciso
- Simplemente lo desapruebo
- Lo desapruebo totalmente

7. ¿Los fondos de escenas presentan tonalidades confusas?

- SI
- NO

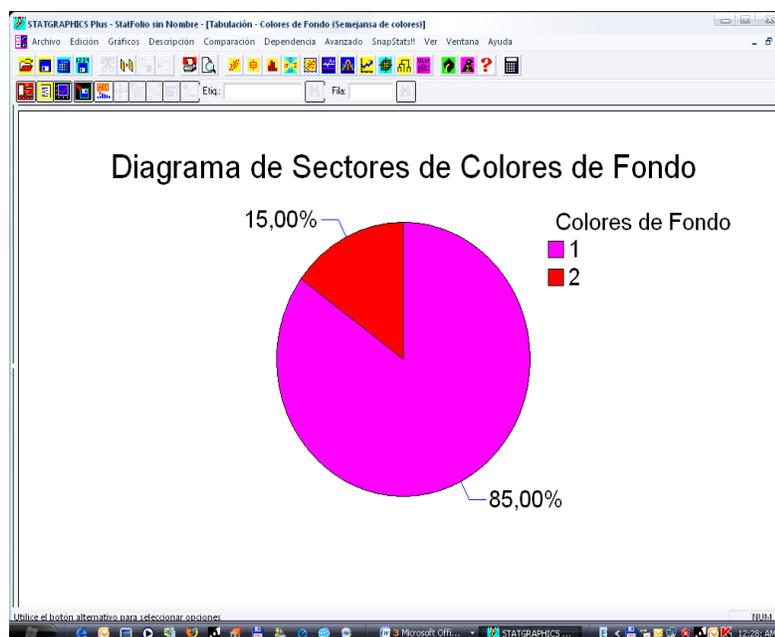
8. ¿Considera usted que el color del fondo que generalmente tienen estos juegos, hacen que usted esfuerce más su visión?

- SI
- NO

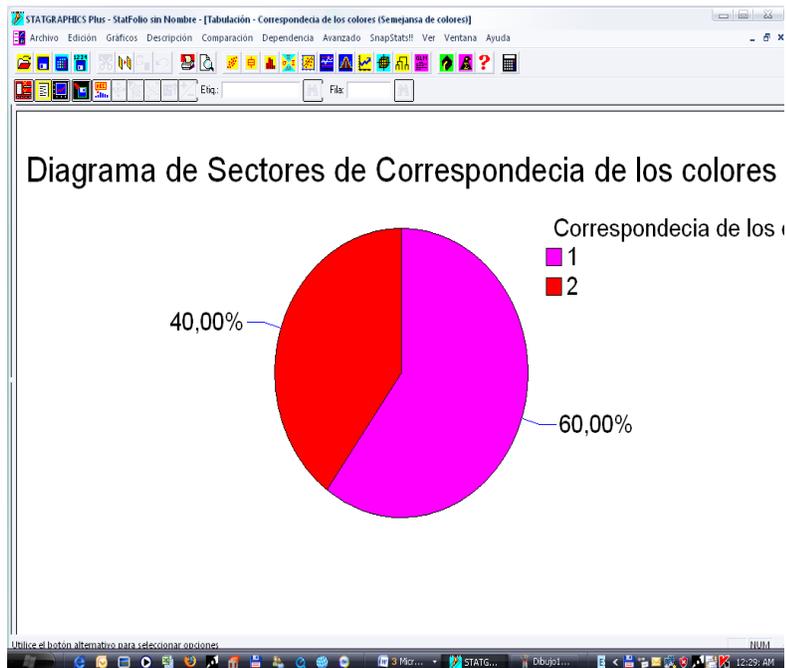
**Anexo: 2 Relación de estudiantes entrevistados en la Facultad 5 de la Universidad de las Ciencias Informáticas.**

1	Roll del estud	Presencia de	Variacion de	Fondos Oscu	Corresponde	Tipo de sola	Semejansa d	Fondos con t	Colores de Fc
2	Programador	1	1	1	1	3	1	1	1
3	Ninguno	2	3	1	1	3	1	2	1
4	Tesista	1	5	2	2	1	1	1	1
5	Analista	1	5	1	1	3	1	1	1
6	Programador	1	5	1	2	3	1	1	1
7	Jefe de linea	1	3	1	1	3	1	2	2
8	Diseñador	2	5	2	1	1	1	1	1
9	Analista	1	5	1	2	3	1	1	1
10	Tesista	2	3	1	2	3	1	2	2
11	Investigador	1	5	2	1	3	1	1	1
12	Investigador	1	5	1	2	3	1	1	1
13	Analista	1	3	1	1	1	1	1	1
14	Tesista	1	5	1	1	3	1	1	1
15	Programador	1	5	1	2	3	1	2	1
16	Ninguno	2	5	4	1	3	1	1	1
17	Ninguno	1	5	1	1	3	1	1	2
18	Analista	1	5	1	2	3	1	1	1
19	Analista	2	5	1	1	3	1	1	1
20	Programador	1	5	2	2	3	1	1	1
21	Programador	1	5	1	1	3	1	1	1

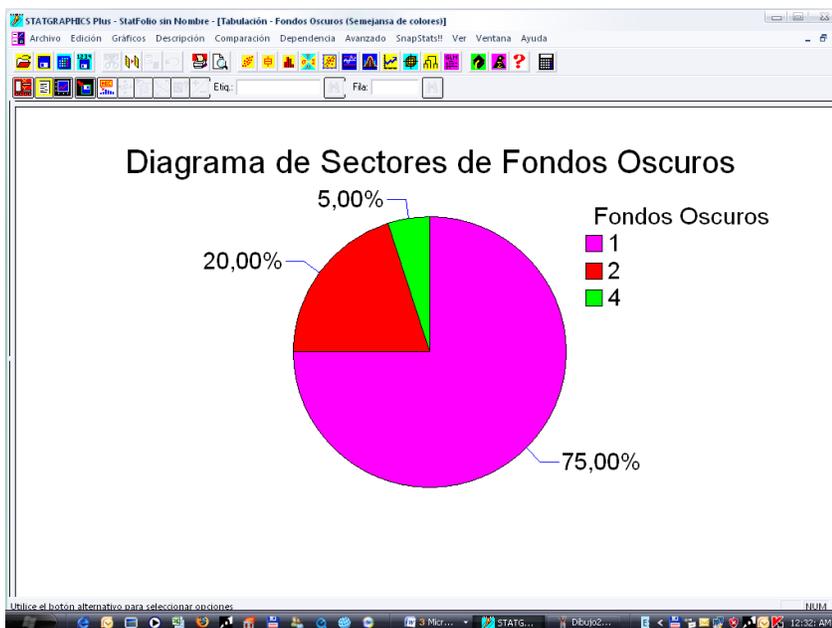
**Anexo: 3 Tabla de frecuencia de colores de fondo.**



**Anexo: 4 Tabla de frecuencia de correspondencia de colores.**



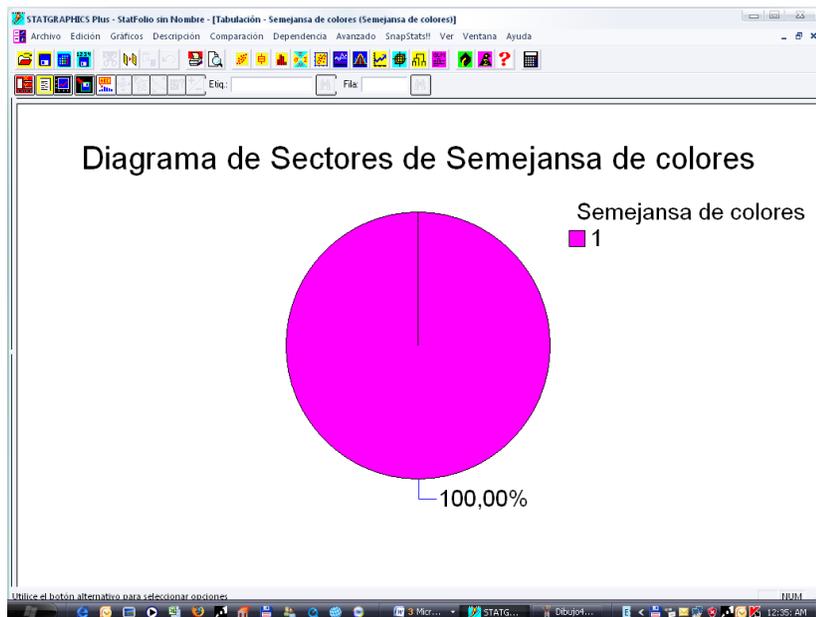
**Anexo: 5 Tabla de frecuencia para fondos oscuros.**



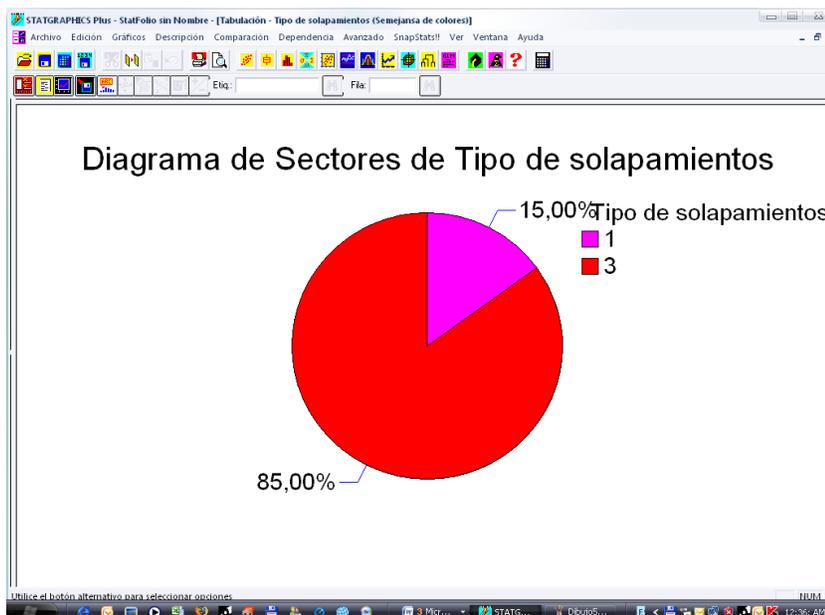
**Anexo: 6 Tabla de frecuencia para presencia de Posición Lógica**



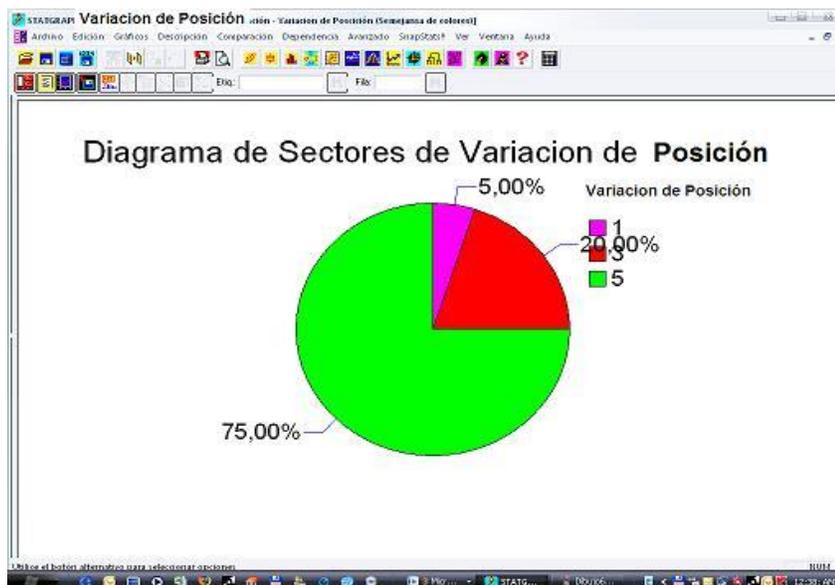
**Anexo: 7 Tabla de frecuencia para semejanza de colores.**



**Anexo: 8** Tabla de frecuencia para tipo de solapamiento.



**Anexo: 9** Tabla para variación de posiciones.



**Anexo: 10 Interfaz de distribución de los objetos con sus coordenadas asignadas.**

Form1

abrir

tabla imagen

CaGrA	665	353	piensbag.bmp
	657	447	carritox.bmp
	595	243	durofrío.bmp
	597	439	botoncar.bmp
	524	438	bumerang.br
amari	93	89	trompeta.bmp
carne	81	362	carmelit.bmp
grisx	755	482	grisball.bmp
	412	410	estrella.bmp
verde	189	295	hierbita.bmp
	265	138	lagartij.bmp
fondo	407	248	nijanero.bmp
	39	209	nirjazul.bmp
	325	375	raketaxy.bmp
overx	585	161	pajarote.bmp
	627	158	techopla.bmp
pisox	201	532	gatolazo.bmp
	545	548	perronij.bmp

Dist

Exit

