

Escuela Politécnica Superior

18  
19

# Trabajo fin de grado

Herramienta de diseño de juegos tipo mazmorra para Gamemaker Studio 2



Javier Gómez

Escuela Politécnica Superior  
Universidad Autónoma de Madrid  
C/ Francisco Tomás y Valiente nº 11



**UNIVERSIDAD AUTÓNOMA DE MADRID  
ESCUELA POLITÉCNICA SUPERIOR**



**Grado en Ingeniería de Textos**

**TRABAJO FIN DE GRADO**

**Herramienta de diseño de juegos tipo mazmorra  
para Gamemaker Studio 2**

**Autor: Javier Gómez**

**Tutor: Carlos Aguirre**

**marzo 2019**

**Todos los derechos reservados.**

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución comunicación pública y transformación de esta obra sin contar con la autorización de los titulares de la propiedad intelectual.

La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual (*arts. 270 y sgts. del Código Penal*).

**DERECHOS RESERVADOS**

© 3 de Noviembre de 2017 por UNIVERSIDAD AUTÓNOMA DE MADRID

Francisco Tomás y Valiente, n.º 1

Madrid, 28049

Spain

**Javier Gómez**

*Herramienta de diseño de juegos tipo mazmorra para Gamemaker Studio 2*

**Javier Gómez**

C\ Francisco Tomás y Valiente N.º 11

IMPRESO EN ESPAÑA – PRINTED IN SPAIN

*Lo peor es cuando has terminado un capítulo  
y la máquina de escribir no aplaude.*

*Orson Welles*



# RESUMEN

---

El objeto de este trabajo de fin de grado es el desarrollo de una aplicación de escritorio que facilite a los desarrolladores de videojuegos la tarea del diseño de niveles.

# PALABRAS CLAVE

---

Diseño de documento,  $\text{\LaTeX}$  2<sub>ε</sub>, thesis, trabajo fin de grado, trabajo fin de master





# ABSTRACT

---

In our School a considerable number of documents are produced, as many aducational as research. Our students also contribute to this production through his final degree, master and thesis projects. The objective of this material is to facilitate the editing of all these documents and at the same time to promote our corporate image, facilitating the visibility and recognition of our center.

In this sense we have tried to design a style of  $\text{\LaTeX}2_{\epsilon}$  that maintains a corporate image and with simple commands that allow to maintain the corporate image with the necessary quality without forgetting the needs of the author. For this, a set of simple commands have been created around complex packages. These commands allow you to perform most of the operations that a document of this type may need.

Likewise, you can control a little the design of the document through the options of the style but always maintaining the institutional image.

# KEYWORDS

---

Document design,  $\text{\LaTeX}2_{\epsilon}$ , thesis, final degree project, final master project



# ÍNDICE

---

<b>1</b>	<b>Introducciónintroduccion/introduccion</b>	<b>1</b>
1.1	Motivación .....	1
1.2	Objetivos .....	1
1.3	Estructura .....	1
1.4	Glosario, acrónimos y definiciones .....	1
1.4.1	Glosario .....	1
1.4.2	Acronimos .....	2
1.4.3	Definiciones .....	2
<b>2</b>	<b>Estado del arte</b>	<b>5</b>
2.1	Diseño Gráfico .....	5
2.2	Diseño Lógico .....	5
<b>3</b>	<b>Requisitos</b>	<b>7</b>
3.1	Requisitos funcionales .....	7
3.1.1	Requisitos de persistencia .....	7
3.1.2	Requisitos de contexto .....	7
3.1.3	Requisitos del canvas .....	8
3.1.4	Requisitos de llaves .....	9
3.1.5	Requisitos de condiciones de apertura .....	9
3.1.6	Requisitos de comprobación de la completitud .....	10
3.1.7	Requisitos de conexión con Game Maker Studio 2 .....	10
3.2	Requisitos no funcionales .....	10
<b>4</b>	<b>Diseño</b>	<b>11</b>
<b>5</b>	<b>Desarrollo</b>	<b>13</b>
<b>6</b>	<b>Integración, pruebas y resultados</b>	<b>15</b>
<b>7</b>	<b>Conclusiones y trabajo futuro</b>	<b>17</b>
7.1	Conclusiones .....	17
7.2	Trabajo futuro .....	17
	<b>Bibliografía</b>	<b>19</b>
	<b>Definiciones</b>	<b>21</b>
	<b>Acrónimos</b>	<b>23</b>



# LISTAS

---

**Lista de algoritmos**

**Lista de códigos**

**Lista de cuadros**

**Lista de ecuaciones**

**Lista de figuras**

**Lista de tablas**

**Lista de cuadros**



# INTRODUCCIÓNINTRODUCCION/INTRODUCCION

---

## 1.1. Motivación

Esta memoria de tfg...

## 1.2. Objetivos

## 1.3. Estructura

## 1.4. Glosario, acrónimos y definiciones

Tres elementos que también están disponibles para los autores son el glosario, la lista de acrónimos y la lista de definiciones.

### 1.4.1. Glosario

Para realizar el glosario y simplificar su creación se han diseñado seis comandos. En todos ellos el primer parámetro es opcional (por tanto si se indica debe hacerse entre corchetes) y representa el elemento referenciado al estilo *see also*. Los comandos que empiezan por Ph (phantom), introducen la palabra en el índice pero sin escribirla en el texto mientras que los que no empiezan por Ph también escriben la palabra en el texto. La combinación de ambos comandos es imprescindible porque el diseño del glosario es muy crítico con la tipología y la misma expresión o palabra con un espacio de más o de menos o una letra en mayúscula o sin mayúscula hacen que haya entradas distintas en el glosario.

**\Index[]{}**  El primer parámetro es opcional y se corresponde con el *see also*. El segundo parámetro es la palabra a indexar.

**\Subindex[]{}{}**  El primer parámetro es opcional y se corresponde con el *see also*. El segundo parámetro es la palabra sobre la que se indexa y el tercero la palabra a indexar.

**\Subsubindex[]{}{}** El primer parámetro es opcional y se corresponde con el *see also*. El segundo y tercer parámetro es el punto de indexación y el cuarto la palabra a indexar.

**\PhIndex[]{}**  El primer parámetro es opcional y se corresponde con el *see also*. El segundo parámetro es la palabra a indexar.

**\PhSubindex[{}]{}** El primer parámetro es opcional y se corresponde con el *see also*. El segundo parámetro es la palabra sobre la que se indexa y el tercero la palabra a indexar.

**\PhSubsubindex[]{}{}** El primer parámetro es opcional y se corresponde con el *see also*.  
El segundo y tercer parámetro es el punto de indexación y el cuarto la palabra a indexar.

### 1.4.2. Acrónimos

Para definir un nuevo acrónimo se puede hacer en cualquier lugar del texto. Así, lo normal, es realizar la definición del acrónimo donde se use por primera vez, dicha definición será añadida a la sección de definiciones al final del texto. Para realizar la definición hay que utilizar el comando **\newacronym{label}{acron}{extended}** donde label es la etiqueta para hacer referencia al acrónimo, acron es el acrónimo en si mismo y extended es lo que significa el acrónimo.

Para hacer referencia a los acrónimos se pueden utilizar las siguientes funciones:

**\ac{label}** La primera vez que se use el acrónimo en el texto aparecerá en su forma extendida y entre paréntesis el acrónimo.

**\acs{label}** Se presenta el acrónimo.

**\acl{label}** Se presenta la forma extendida del acrónimo.

Un ejemplo es por ejemplo la definición de Institute of Electrical and Electronics Engineers (IEEE), que si repito aparece sólo como IEEE o puedo utilizar la forma extendida de esta forma: Institute of Electrical and Electronics Engineers. El formato corto siempre se presentará así: IEEE. Para ver ejemplos de su uso lo mejor es ver los fuentes de este manual justo en este mismo punto.

### 1.4.3. Definiciones

Las definiciones se realizar de forma similar a los acrónimos. La diferencia está en los comandos utilizados. En este caso el comando tiene un parámetro más en el que se introduce el elemento definido en plural y el comando a utilizar es `\newdefinition{label}{defined}{plural}{extended}`. Los comandos para referenciar la definición serán:

**\dfn{label}** Se pone la palabra definida.

**\dfnpl{label}** Se pone la palabra definida en plural.



**\Dfn{label}** Se pone la palabra definida en mayúscula.

**\Dfnpl{label}** Se pone la palabra definida en plural y mayúscula.

Es importante no poner un punto al final de la definición dado que se añade automáticamente al final de las definiciones.

espacio jugable Al igual que con los acrónimos voy a realizar lo mismo con las definiciones incluida la definición de acrónimo. Para ver cómo se usan es importante editar los fuentes de este documento.



## ESTADO DEL ARTE

---

2.1. Diseño Gráfico

2.2. Diseño Lógico



# REQUISITOS

---

## 3.1. Requisitos funcionales

**RF-1.**— Un ejemplo anidado.

**RF-1.1.**— Este es el primer punto anidado.

**RF-1.2.**— Y este el segundo.

**RF-2.**— Se sale de la anidación.

### 3.1.1. Requisitos de persistencia

**RF-1.**— **Guardar el diseño**

El usuario debe poder guardar los datos del diseño sobre el que está trabajando, especificando el nombre del fichero. Dicho fichero se guardará con la extensión. *.maze*

**RF-2.**— **Cargar el diseño**

El usuario debe poder cargar los datos de un diseño previamente guardado.

**RF-3.**— **Crear nuevo diseño**

El usuario debe poder crear un nuevo diseño vacío sobre el que trabajar.

### 3.1.2. Requisitos de contexto

**RF-4.**— **cambio de contexto**

El usuario podrá cambiar el contexto actual de la aplicación para realizar tareas diferentes

**RF-4.1.**— **Contexto de edición de *espacios jugables***

En este contexto el usuario podrá realizar todas las operaciones que tengan que ver con la creación, edición, selección y eliminación de los *espacios jugables*.

**RF-4.2.**— **Contexto de edición de elementos llave y de elementos bloqueadores**

En este contexto el usuario podrá realizar todas las operaciones que tengan que ver con la creación, edición, selección y eliminación tanto de elementos llave como de elementos bloqueadores.

**RF-4.3.**— **Contexto de edición de pantallas**

En este contexto el usuario podrá realizar todas las operaciones que tengan que ver con la creación, edición, selección y eliminación de las regiones denominadas pantallas.

### 3.1.3. Requisitos del canvas

#### RF-5.— Dibujar *espacios jugables*

El usuario, en el contexto de *diseño* debe poder dibujar *espacios jugables* en el canvas haciendo click en un punto del canvas y soltando el click en otro.

##### RF-5.1.— Anexión de *espacios jugables*

Si el conjunto **A** de los *espacios jugables* que intersecan con el recién creado es no vacío, entonces se eliminarán los *espacios jugables* pertenecientes a **A** y se creará un nuevo *espacio jugable* resultante de la unión de todos los contenidos en **A** junto con el recién creado.

##### RF-5.2.— Indexación de *espacios jugables*

Todo *espacio jugable* debe ser indexado por una estructura para su futura transformación en grafo.

##### RF-5.3.— Vértices de un *espacio jugable*

Para todo *espacio jugable* deben definirse sus vértices en el canvas.

##### RF-5.4.— Grid de coordenadas

Los vértices de todo *espacio jugable* deben pertenecer a un *grid* de coordenadas.

#### RF-6.— Modificación de los *espacios jugables*

El usuario debe poder modificar la posición de las aristas y los vértices de un *espacio jugable* preexistente.

#### RF-7.— Eliminación de *espacios jugables*

El usuario debe poder extraer areas rectangulares de un *espacio jugable*, de manera que el resultado respete la definición de *espacio jugable*.

#### RF-8.— Creación de vértices

El usuario debe poder crear un nuevo vértice en una arista.

#### RF-9.— Eliminación de vértices

El usuario debe poder eliminar un vértice siempre y cuando sus dos vértices vecinos estén alineados.

#### RF-10.— Crear un elemento bloqueador

El usuario debe poder crear un elemento bloqueador dibujando una línea en el canvas.

##### RF-10.1.— Especificación del tipo de elemento bloqueador

El elemento bloqueador creado por defecto será de tipo bidireccional, pero el usuario debe poder especificar que sea de tipo unidireccional, en cuyo caso debe especificar en qué dirección tendrá efecto el bloqueo.

##### RF-10.2.— Cálculo de vecindad al crear un elemento bloqueador

En el caso de que la creación de un elemento bloqueador divida un *espacio jugable* en dos, se han de indexar estos dos nuevos *espacios jugables* y se ha de definir su vecindad teniendo en cuenta el tipo de elemento bloqueador y sus condiciones de apertura.

##### RF-10.3.— Eliminación de elementos bloqueadores inválidos

Si el elemento bloqueador que se desea crear no cumple con la definición de elemento bloqueador, este será eliminado del diseño.

#### RF-11.— Seleccionar un elemento bloqueador

El usuario debe poder seleccionar un elemento bloqueador preexistente del canvas.

#### RF-12.— Eliminar un elemento bloqueador

El usuario debe poder eliminar un elemento bloqueador preexistente del canvas.

#### RF-13.— Crear un elemento llave

El usuario debe poder crear un elemento llave, instanciando un tipo de llave preexistente, en el canvas.

#### RF-14.— Seleccionar un elemento llave

El usuario debe poder seleccionar un elemento llave preexistente del canvas.

**RF-15.— Desplazar un elemento llave**

El usuario debe poder desplazar un elemento llave preexistente dentro del propio canvas.

**RF-16.— Eliminar un elemento llave**

El usuario debe poder eliminar un elemento llave preexistente del canvas.

**RF-17.— Zoom**

El usuario debe poder acercar y alejar la vista que tiene del canvas.

**RF-18.— Desplazar vista**

El usuario debe poder desplazar la vista que tiene del canvas.

### 3.1.4. Requisitos de llaves

**RF-19.— Crear tipos de llaves**

El usuario debe poder crear tipos de llaves especificando su nombre y su clase.

**RF-20.— Eliminar tipos de llaves**

El usuario debe poder eliminar tipos de llaves previamente creados.

**RF-21.— Modificar el nombre de un tipo de llave**

El usuario debe poder modificar el nombre que se le hubiera asignado a un tipo de llave.

**RF-22.— Modificar la clase de un tipo de llave**

El usuario debe poder modificar la clase que se le hubiera asignado a un tipo de llave.

**RF-23.— Seleccionar un tipo de llave**

El usuario debe poder seleccionar un tipo de llave de entre la lista de todos los tipos de llaves.

### 3.1.5. Requisitos de condiciones de apertura

**RF-24.— Edición de condiciones de apertura**

Cuando el usuario haya seleccionado un elemento bloqueador, debe ser capaz de describir las condiciones bajo las cuales este se abrirá. Dicha descripción debe ser escrita siguiendo la sintaxis de un lenguaje.

**RF-24.1.— Condición vacía**

El lenguaje para la descripción de condiciones debe admitir a condición vacía.

**RF-24.2.— Condición simple**

El lenguaje para la descripción de condiciones debe admitir condiciones simples, compuestas por un único requisito para la apertura

**RF-24.3.— Operadores lógicos**

El lenguaje para la descripción de condiciones debe reconocer conectores formados por operadores lógicos

**RF-24.4.— Condiciones compuestas**

El lenguaje para la descripción de condiciones debe admitir condiciones compuestas, formadas conectando condiciones (simples o compuestas) con operadores lógicos

**RF-24.5.— Orden de precedencia**

El lenguaje para la descripción de condiciones debe respetar el orden de precedencia habitual: resolución del contenido de los paréntesis primero y conexión de izquierda a derecha.

**RF-25.— Sintetización de condiciones**

El sistema debe ser capaz de sintetizar las condiciones escritas por el usuario, siguiendo la sintaxis del lenguaje de descripción de condiciones, en clases compatibles con las aristas de los grafos, para inducir en estos una

exploración condicionada.

### 3.1.6. Requisitos de comprobación de la completitud

#### RF-26.— Comprobación

El usuario debe poder comprobar que su diseño no contenga errores de tipo *dead lock*.

##### RF-26.1.— Conversión a grafo

El diseño se debe poder convertir a una estructura de tipo grafo para poder realizar la exploración del mismo

##### RF-26.2.— Notificación de errores

Si el diseño comprobado contiene un error de tipo *deadlock* se ha de notificar al usuario del camino escogido para encontrar dicho error.

### 3.1.7. Requisitos de conexión con Game Maker Studio 2

#### RF-27.— Creación de pantallas

El usuario debe poder seleccionar regiones rectangulares del canvas que representarán las pantallas del juego que está diseñando. Al crearse, las pantallas tendrán un nombre asignado por defecto.

#### RF-28.— Seleccionar una pantalla

El usuario debe poder seleccionar una pantalla.

#### RF-29.— Cambiar el nombre de una pantalla

El usuario debe poder cambiar el nombre de una pantalla.

#### RF-30.— Eliminar una pantalla

El usuario debe poder eliminar un región que representa una pantalla de su juego

#### RF-31.— Modificar dimensiones de una pantalla

El usuario debe poder modificar las dimensiones de una pantalla, desplazando uno de sus bordes.

#### RF-32.— Conversión a ficheros .yy.room

El sistema debe ser capaz de transformar las pantallas a ficheros .yy.room de *Game Maker Studio 2*, con las dimensiones y nombre determinados por el diseño del usuario. En estos ficheros figurarán solamente los bloques sólidos que cubrirán los *espacios no jugables* del diseño.

##### RF-32.1.— Construcción de un área con el mínimo número de rectángulos

Los *espacios no jugables*, que tendrán forma poligonal, han de cubrirse del menor número de rectángulos posible. Estos rectángulos, en la traducción a ficheros .yy.room, se representarán como bloques sólidos del juego. descripcion

## 3.2. Requisitos no funcionales

#### RNF-1.— Un ejemplo anidado.

RNF-1.1.— Este es el primer punto anidado.

RNF-1.2.— Y este el segundo.

#### RNF-2.— Se sale de la anidación.



## DISEÑO

---



## DESARROLLO

---



## **INTEGRACIÓN, PRUEBAS Y RESULTADOS**



## CONCLUSIONES Y TRABAJO FUTURO

---

### 7.1. Conclusiones

### 7.2. Trabajo futuro





# BIBLIOGRAFÍA

---

- [1] K.S.NARENDRA AND K.PARTHSARATHY, *Identification and control of dynamical system using neural networks*, IEENN, 1 (1990), pp. 4–27. [Download](#).
- [2] R. W. ZUREK AND L. J. MARTIN, *Interannual variability OF planet-encircling dust activity on Mars*, J. Geophys. Res., 98 (1993), pp. 3247–3259. [Descargar](#).



# DEFINICIONES

---

**acrónimo** Sigla cuya configuración permite su pronunciación como una palabra; por ejemplo, ovni: objeto volador no identificado; TIC, tecnologías de la información y la comunicación.

**definición** Proposición que expone con claridad y exactitud los caracteres genéricos y diferenciales de algo material o inmaterial.

**espacio jugable** regiones de un nivel por las que el jugador podrá mover a su avatar.



# ACRÓNIMOS

---

**IEEE** Institute of Electrical and Electronics Engineers.



# ÍNDICE TERMINOLÓGICO

---

budgettitle, 25

colores, 2

    predefinidos, 2

eigenvalue, 40

opciones, 47







