

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, nº 1

Madrid, 28049

Spain

Javier Gómez

Herramienta de diseño de juegos tipo mazmorra para Gamemaker Studio 2

Javier Gómez

C\ Francisco Tomás y Valiente Nº 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, \LaTeX 2_ε, 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

1	Introducciónintroduccion/introduccion	1
1.1	Motivación	1
1.2	Objetivos	1
1.3	Estructura	1
2	Estado del arte	3
2.1	Diseño Gráfico	3
2.2	Diseño Lógico	3
3	Requisitos	5
3.1	Requisitos funcionales	5
3.1.1	Requisitos de persistencia	5
3.1.2	Requisitos de contexto	5
3.1.3	Requisitos del canvas	5
3.1.4	Requisitos de llaves	7
3.1.5	Requisitos de comprobación de la completitud	7
3.1.6	Requisitos de conexión con Game Maker Studio 2	7
3.2	Requisitos no funcionales	7
4	Diseño	9
5	Desarrollo	11
6	Integración, pruebas y resultados	13
7	Conclusiones y trabajo futuro	15
7.1	Conclusiones	15
7.2	Trabajo futuro	15
	Bibliografía	17
	Definiciones	19
	Acrónimos	21

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

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-1.– **cambio de contexto**

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

3.1.3. Requisitos del canvas

RF-1.– **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-1.1.– **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-1.2.– Indexación de espacios jugables

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

RF-1.3.– Vértices de un espacio jugable

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

RF-1.4.– Grid de coordenadas Los vértices de todo *espacio jugable* deben pertenecer a un *grid* de coordenadas.

RF-2.– Modificación de los espacios jugables

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

RF-3.– Eliminación de espacios jugables

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

RF-4.– Creación de vértices

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

RF-5.– Eliminación de vértices

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

RF-6.– Crear un elemento bloqueador

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

RF-6.1.– 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-6.2.– 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-6.3.– 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-7.– Seleccionar un elemento bloqueador

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

RF-8.– Eliminar un elemento bloqueador

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

RF-9.– Crear un elemento llave

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

RF-10.– Seleccionar un elemento llave

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

RF-11.– Desplazar un elemento llave

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

RF-12.– Eliminar un elemento llave

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

RF-13.– Zoom

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

RF-14.– Desplazar vista

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

3.1.4. Requisitos de llaves

RF-1.– Crear tipos de llaves

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

RF-2.– Eliminar tipos de llaves

El usuario debe poder eliminar tipos de llaves previamente creados.

RF-3.– 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-4.– 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-5.– 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 comprobación de la completitud

RF-1.– nombre

descripcion

3.1.6. Requisitos de conexión con Game Maker Studio 2

RF-1.– nombre

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.

opción de estilo Son los valores que modifican el funcionamiento del estilo. Se ponen entre corchetes y separadas por comas en el comando `\documentclass` y antes de el nombre del estilo que irá entre llaves.

ACRÓNIMOS

IEEE Institute of Electrical and Electronics Engineers.

WYSIWYG What You See Is What You Get.

WYTIWYG What You Think Is What You Get.

ÍNDICE TERMINOLÓGICO

budgettitle, 25

colores, 2

 predefinidos, 2

eigenvalue, 40

opciones, 47

