**Proyecto**

Lab21

**Grupo**

Dire Wolf Games

FORMATO DE DEFINICIÓN DE NIVELES

Hito:

Fecha entrega: 23-12-2016

Versión: 1.0

Componentes:

* Aarón Colston Avellà Hiles
* Sergio Huertas Ferrández
* Eduardo Ibáñez Frutos
* Marina López Menárguez
* Rubén Moreno Mora
* Rafael Soler Follana

Contenido

[Contenido 1](#_Toc470160229)

[1. Introducción 2](#_Toc470160230)

[1.1. Propósito 2](#_Toc470160231)

[2. Formato 2](#_Toc470160232)

[2.1. Descripción. 2](#_Toc470160233)

[2.2. Especificación. 3](#_Toc470160234)

# Introducción

## Propósito

*Definir un formato propio para el diseño de niveles que en primera instancia será un archivo JSON para facilitar la depuración y modificación “a mano” de los niveles durante el desarrollo del Videojuego.*

*Se pretende crear este archivo desde Blender, mediante un script en Python que exporte los elementos de la Escena a un archivo de texto con formato JSON.*

*Para la implementación final se creará un archivo binario para la carga.*

# Formato

## Descripción.

*El archivo puede contener uno o más niveles <levels> que contienen un identificador, un nombre y uno o más elementos o modelos <elements>. Estos elements son los distintos modelos creados en Blender de manera que ninguno se repita.*

*La posición de los elements en la escena se define a continuación y se dividen en:*

* *<static-element>: Elementos estáticos del escenario tales como el suelo, paredes, mesas no movibles, etc.*
* *<initial-enemies>: La posición inicial de los enemigos en el nivel. Define también la rotación inicial del mismo. Se pueden añadir elementos personalizados a cada enemigo definiendo “Custom Properties” de Blender para el objeto a exportar, tales como el estado, la vida, etc. Este punto queda a desarrollar conforme el juego avance.*
* *<initial-consumable>: La posición inicial de los elementos “consumibles”. Los que se pueden coger del escenario por el player, tales como munición o botiquines. También aceptan las “Custom Properties”.*
* *<initial-entity>: Las entidades como puertas, generadores… Aceptan “Custom Properties”.*

*En todos los casos el <element-id> de cada elemento debe ir relacionado con el tipo de objeto creado en el Videojuego.*

## Especificación.

{"levels": [

{

"level-id": int,

"level-name": string,

"elements": [

{ "element-id": int, "mesh":"path/to/mesh.ext" },

...

],

"static-elements": [

{ "element-id": int,

"position": [ "x":float, "y":float, "z":float ],

"rotation": [ "x":float, "y":float, "z":float ] },

...

],

"initial-enemies": [

{ "element-id": int,

"position": [ "x":float, "y":float, "z":float ],

"rotation": [ "x":float, "y":float, "z":float ],

"state": int },

...

],

"initial-consumable": [

{ "element-id": int,

"position": [ "x":float, "y":float, "z":float ],

"rotation": [ "x":float, "y":float, "z":float ],

"state": int },

...

],

"initial-entity": [

{ "element-id": int,

"position": [ "x":float, "y":float, "z":float ],

"rotation": [ "x":float, "y":float, "z":float ],

"state": int },

...

],

]}