SurvivorSurvivor Helicopter!

Level Data Format

Open Game Developers

[Version: 0.0.0.1] [Date: 05|08|2013]

SurvivorSurvivor Helicopter!

Change Log

Version	Author	Changes
0.0.0.1	El-Rico	A first-adjustment pass of the level file format

SurvivorSurvivor Helicopter!

Table of Contents

SurvivorSurvivor Helicopter!

Preface

This document exists to detail the particulars of the level format used for SurvivorSurvivor Helicopter! Level formations will not be described in this document, see the Level Designs Document for more information on the levels used in the game.

SurvivorSurvivor Helicopter!

Anatomy of a Level

Components

Tiles are assumed to be meshes in the ZED mesh file format.

Tile ID

An 8-bit value which provides more than enough variety in tiles for a single level.

Tile flags

An 8-bit value, which indicates whether this tile is a spawn point or if it is a helicopter landing zone.

File Layout

Type		Count	Name	Description
char		4	ID	Contains “SSHL” (SurvivorSurvivor Helicopter! Level)
char		256	Path	Path to the tile set
Unsigned 16-bit integer		2	Dimensions	The width and depth of the level, respectively
*for width*height*				
	byte	1	Tile ID	The tile to use (tile sets contain meshes with names starting at zero, zero and one are reserved for the spawn point tile and the helicopter landing zone tile, respectively)
	byte	1	Tile Flags	Eight OR-ed flags which determines the type of tile this is
end for				

Loading a Level

Pseudocode

```
read file_header ;((char*4)+(char*256)+(uint16*2))
if compare( file_header.id, “SSHL” ) not successful
        return;
if failed( check_tile0_and_tile1_are_present( file_header.path ) )
        return;
if failed( check_all_tiles_in_file_exist( file_header.path ) )
        return;
store_tile_ids_in_order_with_index( )    ;Takes the tile IDs from the previous check and
                                                ;stores them in a list so that the renderer
                                                ;doesn't need to keep switching meshes and
                                                ;textures as often
```

Rendering a Level

Pseudocode

```
for( x = 0; x < tile_id; ++x )
        itr =  tile_id_position_map[ x ].begin   ;tile_id_position_map = map< int, list > - The list contains
all                                              ;of the positions for the tile
        while( itr != tile_id_position_map[ x ].end )
                render_tile( x, itr.position )
```