

# Pekka Kana 2 Map file format

## File Format

Data type	Bytes	Name	Description
char[]	5	Version	Version number, should be 1.3.
char[]	13	Tileset	Tileset image file name
char[]	13	Background	Background image file name
char[]	13	Music	Music file name
char[]	40	Map name	Name of the map (Not filename)
char[]	40	Author	Name of the map author
char[]	8	Level number	Number of the level, in the episode
char[]	8	Weather	Weather effect (List below)
char[]	8	Switch 1 time	Switch 1 time, hard coded to be 2000
char[]	8	Switch 2 time	Switch 2 time, hard coded to be 2000
char[]	8	Switch 3 time	Switch 3 time, hard coded to be 2000
char[]	8	Time	Time limit of the level
char[]	8	Extra	Scrolling type (List below)
char[]	8	Background	Not used
char[]	8	Player sprite	The player sprite, this is an index to the <i>prototypes</i> array
char[]	8	X	X position of the icon, on the level selection map
char[]	8	Y	Y position of the icon, on the level selection map
char[]	8	Icon	Icon on the level selection map
int	4	Prototypes	Amount of sprites, in the level
char[]	13 * prototypes	Sprite name	File name of sprite

**Note:** After this the fore- and background tiles as well as the sprites are stored, this is continued below.

### Notes:

- Every char[] entry with the size of 8 bytes have to be converted into their respective data types. (Look at *Class* to find the respective data types for each variable.)
- Every char[] is of course null terminated.

## Loading fore-/background tiles as well as sprites

All three are stored the same way in the file.

Data type	Bytes	Name	Description
char[]	8	StartX	The X position of the <b>first</b> tile/sprite
char[]	8	StartY	The Y position of the <b>first</b> tile/sprite
char[]	8	Width	The X position of the <b>last</b> tile/sprite
char[]	8	Height	The Y position of the <b>last</b> tile/sprite

**Note:** Replace *layer* with each respective layer. (Fore-/background, sprites)

### Example:

Pseudocode
<pre>int MAP_WIDTH = 256;  for (int y = startY; y &lt;= startY + height; y++) {     for (int x = startX; x &lt;= startX + width; x++) {         layer[MAP_WIDTH * x + y] = read a byte;     } }</pre>

Before saving a level, the level editor calculates the first and last positions of the tiles/sprites and only stores the used tiles. This saves space.

See method `RECT LaskeTallennusAlue(UCHAR *alue)` for more details.

Methods dealing with map data. (All found in file: „PK2Map.cpp“)

Method	Line	Description
<code>int LataaVersio13(char *filename)</code>	984	Load map file version 1.3
<code>RECT LaskeTallennusAlue(UCHAR *alue)</code>	362	Calculate used area
<code>int Tallenna(char *filename)</code>	444	Save map file

**Note:** Scrolling only affects the background image.

Weather	
Normal	0
Rain	1
Falling Leaves	2
Rain & Leaves	3
Snow	4

Scrolling	
Static/None	0
Vertical	1
Horizontal	2
Horizontal &	3
Vertical	

## Class (File „PK2Map.h“, line: 64; Translated directly from finnish)

### C++ Class

```
class PK2Map {  
  
    public:  
    char version[5];  
    char tileset_image[13];  
    char background_image[13];  
    char music_file[13];  
    char name[40];  
    char author[40];  
  
    int level_number;  
    int weather;  
    int time;  
  
    byte extra;  
    byte background;  
  
    DWORD switch1_time;  
    DWORD switch2_time;  
    DWORD switch3_time;  
  
    int player_sprite;  
  
    byte background_tiles[MAP_SIZE];  
    byte foreground_tiles[MAP_SIZE];  
    byte sprites[MAP_SIZE];  
    char prototypes[MAP_MAX_PROTOTYPES][13];  
  
    int x,y;  
    int icon;  
};
```

Constants	Value	File	Line
MAP_SIZE	256 * 224	PK2Map.h	14
MAP_MAX_PROTOTYPES	100	PK2Map.h	20

### Notes:

- Unimportant variables of the class are not represented here, to see the full class look at the class file. (See above: *Class*)
- Variables not used in game, but stored in file (with default values):
  - `byte background = 0;`
  - `DWORD switch1_time = 2000;`
  - `DWORD switch2_time = 2000;`
  - `DWORD switch3_time = 2000;`