Style Sheet

General considerations

We follow Google style guide to create the source code of Space Invaders. Please, visit http://google-styleguide.googlecode.com/svn/trunk/cppguide.xml

The language used in the project is C++.

Default language for writting code is English. This is specially sensible for comments, name of variables, attributes, methods,...

Headers

Header files

Header file has the "*.h" extension in its name.

Constants

Always name them in capital letters.

Separate words using the underline character "_"

Enumerators

When enumerating, start the first enumerator as prephix_NO_type if neccesary. This value has no meaning and it is for management purpouses only. For instance,

```
typedef enum {
	CHAR_NO_RENDER,//No render for this character: camera, collision objects,...
	CHAR_2D,
	CHAR_LINES3D,
	CHAR_3D,
	CHAR_MAX_RENDER_MODE //For management purpouses only. Always at the end of the
enumeration
} CHAR_RENDER_MODE;
```

The last enumerator has to be named as prephix_MAX_type. This is important for loops control.

As an example

```
typedef enum {
	CP_LEFT_LASER,		///<Code for left laser
	CP_RIGHT_LASER,		///<Code for right laser
	CP_MAX_PLAYER_LASERS	///<Amount of extra lasers the player has
} CP_PLAYER_LASER;
```

```
for (unsigned int i=CP_LEFT_LASER;i<CP_MAX_LASERS;i++) {…}
```

Syntax

All constant, types,... has the general rule prephyx_constant. In case of using multiple words, separate all by "_" character. All the names are in capital letters.

Header inside a file

All the files have a header inside the code at the very beginning of the source code. Each header is provided as a comment of several lines in a doxygenated fashion. It has to indicate a general description of the content of the file. If neccesary, a prefix to all methods, attributes and so on that belong to or are defined in that file and may be reached from outside.

Include the original author and the last time it was updated.

As an example

```
/**	Definition common algebra types
*	Prefix: ALG_
*	@Author: Ramón Mollá
*	@version: 2011-10
*/
```

Always include some resilient code at the beginning of the header in order to avoid loading a header serveral times like this one

```
#ifndef PREPHIX_NAME
#define  PREPHIX_NAME
…
#endif		//End of the header
```

This improves also compiling performance

Source Code

Blocks of code associated to if-then-else, switch, for loops,... are indented and have a single "{" symbol at the line just before and another one } at the line just after.

As an example

```
if (Player[CurrentPlayer].Laser[CP_LEFT_LASER].Alive)
{
	P.v[X3D]	= Position.v[X3D] - .9f;
	P.v[Y3D]	= Position.v[Y3D] + .8f;
	P.v[Z3D]	= CHAR_GenericMesh[CHAR_PLAYER].pos.v[Z3D];
}
```

Literal values are forbidden at any position in the code. Use always typefied values, enums, #define,...

All the variables start with a capital letter. In case of using multiple words, start every word with a capital letter and do not use "_" characters.

Indent blocks of code and do not finish an statement with a curly backet "{". Break the statement and let the curly start the next line. It helps to separate blocks of code.

Classes

All the classes start by the capital character “C”.

Do not use structs if it is not strictly neccesary Use always classes instead.

Use separate modules for every class. Every module has to hold at least a file with the name module.cpp and another header one with the name module.h

Documentation

All the code inserted has to be doxygenated. http://www.doxygen.org/

Netiquette

Please, do not upgrade anything if it does not compile and executes properly. Be polite with everybody working in the same project as you.

Do not synchronize temporal files or complete folders like debug, release, “*.ilk”, “*.manifest”, “*.ncb”,... or personal files like “*.user”.