



Menu Master

The ultimate UI authoring toolchain

Menu Master is a complete User Interface (UI) authoring toolchain. It helps video game developers to dramatically reduce the time it takes to author any sophisticated User Interface: Front end, Head-up display (HUD), in-game menu.

It seemed to be harmless.

Do not underestimate the complexity of the User Interface...

With Menu Master, your UI production has never been so safe.



100% Compatible

UI authored with Menu Master can be quickly deployed on any C++ platform.



Artist Driven

No need for programmers during authoring, instant WYSIWYG real-time preview.



Rapid UI Concept Design & Iterative Authoring

Basic UI can be tested at any time, and gradually upgraded until obtaining a cutting-edge UI.



No Creativity Constraints

Use any kind of multimedia objects: 2D bitmaps, 2D vector graphics, 3D meshes, videos ...



Easy Localization

Texts are saved in an editable file, allowing addition of new languages, Unicode is supported.



Developer Friendly

Open architecture, small footprints, easy to integrate, no memory allocations, flexible data ...



Fully Scalable

Supports any third-party technologies, upgradeable to follow the evolution of the industry.

:: Menu Master > Bundle



Menu Master Studio

An intuitive visual tool that allows UI Designers to easily and quickly create sophisticated UIs without any need for programmers' assistance.

Menu Master Data Generator

A smart module that optimizes all UI data according to each device constraints and capabilities.

Menu Master Development Kit

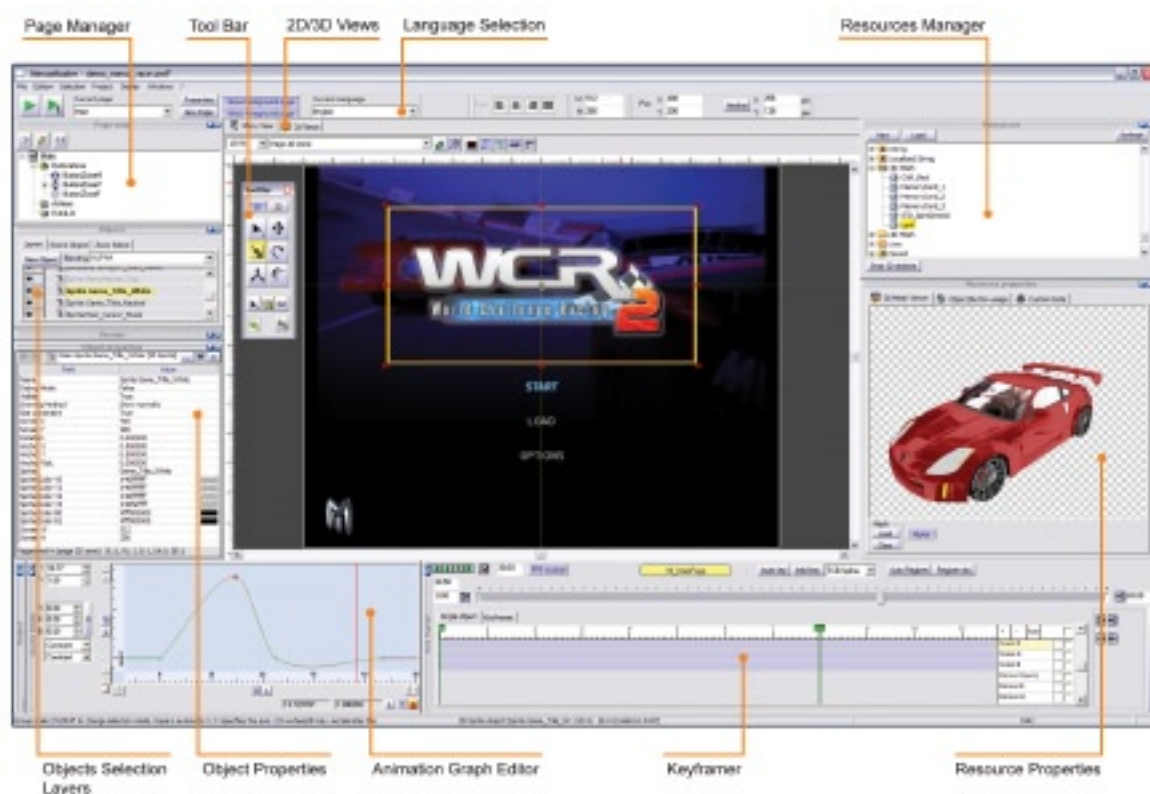
The flexible and clean SDK that allows Engineers to easily integrate the UI data into the device.

:: Menu Master Studio > Overview

Menu Master Studio puts artists in control; they can use any kind of multimedia component, and make user interface changes directly, without having to involve programmers, there are absolutely no creativity constraints.

Menu Master Studio allows iterative prototyping and reuse of concepts, resulting in significant time saving.

Your display engine can be integrated into Menu Master Studio, in order to get a rendering perfectly identical to what you will have in your game. All the effects of your display engine can be controlled.



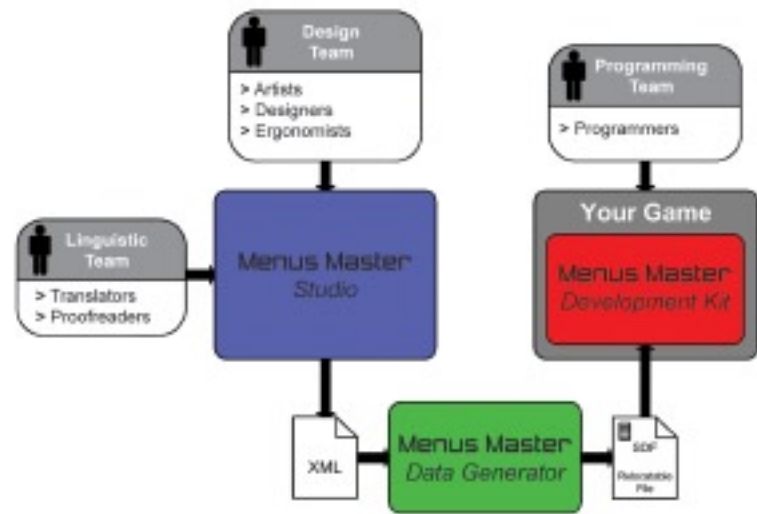
:: Menus Master > Workflow

Menus Master put artists in control and dramatically speeds up the entire an open architecture, based on drivers. It supports any homemade or third-party technologies. It is upgradable to follow the evolution of technologies.

Step 1: The Design section creates the User Interface with **Menus Master Studio**. The Linguistic section also comes into play to check if the texts are correct, and to possibly add new languages.

Step 2: **Menus Master Data Generator** builds, from this generic UI, intended specific UIs, optimized for each device, according to their specificities.

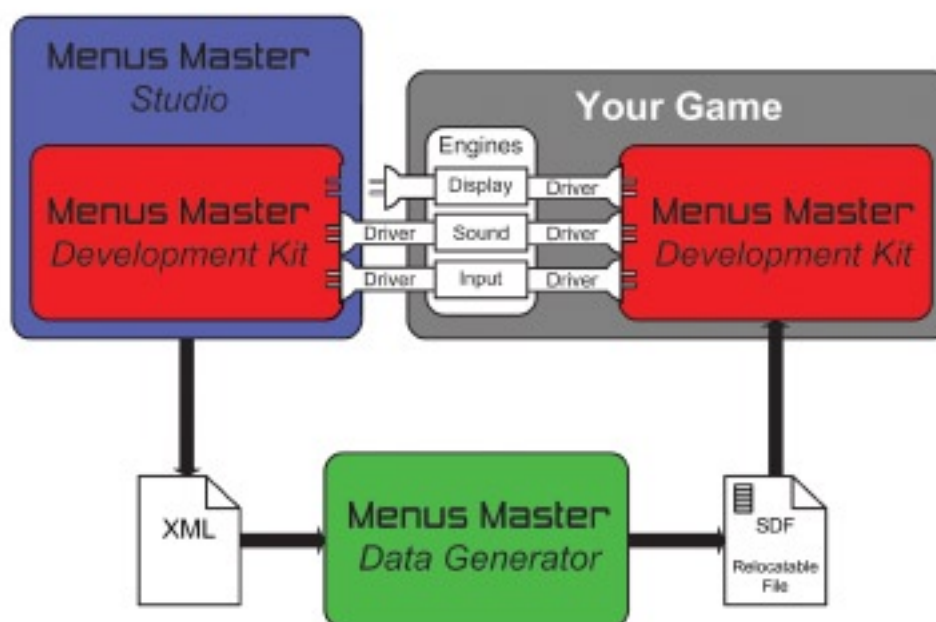
Step 3: The Technical section creates drivers for each device. The UI is then integrated with **Menus Master Development Kit**.



:: Menus Master > Open Architecture

Menus Master has an open architecture, based on drivers. It supports any homemade or third-party technologies. It is upgradable to follow the evolution of the industry.

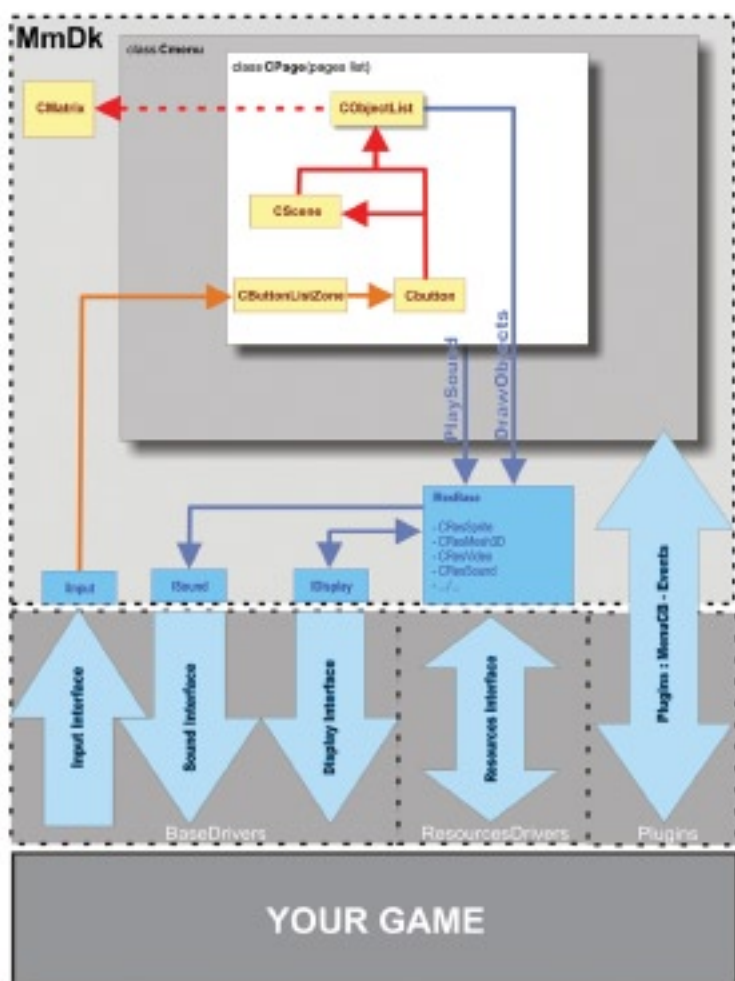
Thanks to this open architecture, a user interface authored with **Menus Master Studio** can be immediately deployed on any c++ platform.



"Menu design and user interface (UI) creation is often one of the most problematic tasks as it relies on a synergetic meshing of programming and art skills. And despite being normally outside the critical pipeline from asset creation to engine integration and game logic, it remains highly vulnerable to last-minute revisions and hence is usually one of the last pieces of work to be completed. Yet potentially menu design should be straightforward problem to solve, not least from a technological point of view".



:: Menus **Master** Development Kit

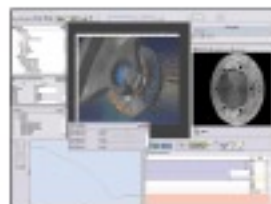
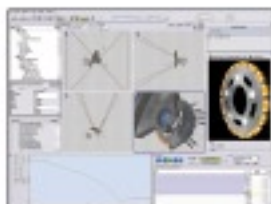


Menus **Master** Development Kit is easy to integrate into any game engine, thanks to its drivers based architecture.

The game engine loads UI data in one sole memory allocation, thanks to a smart data relocation system.

Several UIs can be stored in memory together, they can possibly be used at the same time.

The game engine may interact with the UI via C++ actions and events if needed.



FIND OUT MORE

To find out more information about Menus **Master** please contact:
Sébastien Kohn at contact@omegame.com,
or visit our website at www.omegame.com.

