Key Features of CGI Studio
With the advances in graphics content for display-enabled embedded systems, efficient and easy-to-use HMI/GUI (Human Machine Interface / Graphics User Interface) tools are becoming increasingly important.

The CGI Studio from Fujitsu is one such tool specifically designed for automotive applications, especially automotive clusters and infotainment systems.

CGI Studio enhances the system developer’s ability to input a design, develop and verify the application, and rapidly deploy it to the target system. This HMI tool helps throughout the entire process—from rapid prototyping of user interfaces all the way to serial development of the embedded system—saving time and money. Almost as important, CGI Studio completes these steps easily and cost-efficiently, with a process flow that is as seamless as possible.

This white paper will examine the most significant characteristics of Fujitsu CGI Studio—specifically its cost effectiveness, seamless operation, separation of code and user-interface data, and system benchmarking. The paper will then discuss each of the Fujitsu CGI Studio modules.

**Notable Features of CGI Studio**

**Cost Factors**

CGI Studio allows early evaluation of HMI development. The tool is structured to support design reviews, even in the early stage of the project. It also conveniently supports changes to the project implementation (such as graphics content) in the later stages of the project. These factors help make the CGI Studio cost effective.

For example, CGI Studio gives a realistic preview of the graphics and scenes as soon as they are imported. This is implemented within the IDE of the tool and enables the designer to have a WYSIWYG view of the graphics. The tool also can run and immediately verify widgets or business logic imported into the tool. This is only possible because CGI Studio can simulate graphics applications on the host PC.

This HMI tool also allows the look and feel of the graphics to be changed without needing to recompile or rebuild the project (re-skinning). This is achieved by separating code and graphics data at the very beginning of the project. For example, at the outset of the project, the graphics designer can churn out 2D and 3D graphics using industry-standard tools such as 3D Studio Max or Maya. The software engineer can work independently on business logic to assign behavior to the graphic elements. The two elements can then be imported into the tool and verified together. If needed, the look and feel can be changed later simply by re-importing new graphics assets without affecting the business logic.

**Figure 1: Different Roles in Graphics Application Development Using CGI Studio**

**Artist:**
- Graphic design concept and content definition

**Tools:**
- Digital Content Creation Tools (DCC), HMI Tool

**Technical Artist:**
- Bridge between Artist and Engineer, 3D Model Optimization, scene composition and application dynamics

**Tools:**
- DCC, HMI Tool

**Software Engineer:**
- Widget and application development, testing, integration

**Tools:**
- HMI Tool, Software Development Tools
These tools help throughout the entire process—from rapid prototyping of user interfaces to serial development of the embedded system.

**Seamlessness**

CGI Studio can be easily integrated into the existing tool chain environment in each stage of the application development. The resulting seamless process flow reduces costs, simplifies tool management, and provides a common knowledge basis.

The project team—the studio artists, technical artists, and embedded-software engineers—can focus on their individual roles. The studio artist can design the 2D or 3D graphics content using industry-standard studio tools and preview the design with the HMI tool. The technical artist can import that graphics content into the HMI tool and use the graphics to create scenes with appropriate lighting and composition. (The technical artist also focuses on the graphics, but his work is more hardware-aware than that of the studio artist.) Finally, the software engineer can add the business logic, which will assign appropriate behavior to the graphics elements.

The tool chain supports a flexible licensing model for use in a variety of situations, such as tool evaluation, rapid prototyping of a reference application, and product development. The user’s cost is determined by the resources actually needed.

**Separation of Code and UI Data**

Like all HMI tools, CGI Studio uses both the UI data (e.g., graphics assets such as bitmap files, 2D/3D models, and asset properties) and code that defines the application behavior. The code embodies business logic and can be generated by a state machine tool or hand coded. In the latter case, CGI Studio may be programmed with common software languages.

Advanced HMI tools enhance the developer’s ability to input a design, develop and verify the application, and rapidly deploy it to the target system.

CGI Studio is structured so that the UI data is strictly separated from the code portion. This means the look and feel of the graphics can be changed without touching or rebuilding the code. The new data simply replaces the old UI data in the tool output.

**System Benchmarking**

Another benefit is that CGI Studio can determine hardware requirements for the intended graphics application early in the design process. For example, the tool can help determine the minimum number of MIPS needed for graphics. It can provide guidance about the required power of the GPU as well as the internal bus throughputs and external graphics memory bandwidth requirements. The possible reduction in hardware iterations and project time provides immense benefits.

**CGI Studio Scene Composer**

- 2D and 3D scene creation
  - Hybrid 2D/3D scene composition
  - Distinct layer support for 2D and 3D content
- Artifact import
  - Clearly defined import workflow
  - Artifacts imported from DCC tools
  - Models and geometry in FBX format
  - 2D TrueType fonts
  - 3D fonts
  - Shader programs
  - Widgets
- Screen and scene composition
  - Drag and drop of imported 2D/3D graphical artifacts and widgets
  - 2D and 3D scene composition
  - Light and camera configuration
  - Widget configurations including assignment to 2D/3D objects
  - Animation creation
  - Real-time scene visualization
  - Multi-layer and display support
- Asset export
  - Export resources for host and target
  - Verification and testing
    - Active scene rendered in WYSIWYG window
    - Early visual inspection of imported content
    - Dynamic changing of object properties in the active scene for immediate results
This HMI tool is based on 3D and 2D Candera engines. The application allows platform-independent abstraction and fully supports the OpenGL ES 2.0 standard to give system designers the greatest possible design flexibility.

The CGI Studio HMI tool chain from Fujitsu has been specifically designed for automotive applications such as automotive clusters and infotainment systems.

The licensing model covers a variety of usage scenarios, such as tool evaluation, prototyping, reference design and product development for one or more vehicle platforms. The tool chain is hardware and OS agnostic.

CGI Studio consists of different blocks or modules, the primary ones being Scene Composer, Candera Engine and Player.

**Scene Composer**

The CGI Studio Scene Composer enables creation, composition, verification and testing of 2D and 3D scenes and screens. Artifacts can be imported and dropped into the scenes. Objects, lights, transitions and animation can be created and configured, and assets exported for the host target system.

Scene Composer allows combining 2D and 3D content, enabling a seamless process flow. Within a scene, 2D and 3D content can be separated into distinct layers to address each respective need. The application can have any combination of graphics: 2D only, 3D only, or both.

One input to this tool is the graphics assets, such as models and geometry in FBX format, textures, 2D TrueType fonts, and 3D fonts. These assets could be generated using industry-standard 3D and 2D authoring tools.

Another key input to the Scene Composer is the widgets. As mentioned earlier, these blocks of code assign behavior to the graphics elements. For example, a widget could control the movement of a needle for a cluster application. Other inputs to the tools are animation and a shader program for the GPU.

Scene Composer allows imported 2D/3D graphics assets and widgets to be dragged and dropped. This enables easy 2D/3D scene composition and object configuration. Controls for light and camera are also available. Animations may also be created with the tool, which can be configured with the assigned 2D/3D widget properties. Scene Composer is “display controller layer and multi-display aware.” It can export its output to be run on either the host or the target.

Because the active scene is always rendered using a WYSIWYG approach, it is possible to visually inspect the imported graphics assets early in the development. Object properties can be dynamically changed in the active scene to get immediate results and timely evaluation.
**Candera 3D Engine Feature Set**

- Full support of the OpenGL ES 2.0 feature set
- OS, application and hardware independent
- Robust scene-management features
  - Scene-based screen composition
  - Multiple scenes and screens (including camera inputs)
  - Multiple render targets
  - Multi-pass rendering
  - Anti-aliasing, including FSAA
  - Dithering
- 3D objects
  - Advanced 3D object handling and processing capabilities
  - 3D surfaces (mesh)
  - Hierarchical object groups
  - 2D objects in 3D space
    - Billboards
    - Point sprites
  - 2D and 3D text support
- Enhanced 2D TrueType font rendering: LTR, RTL, Bidi (e.g., Arabic)
- 3D processing
  - Dynamic lighting
  - Static and dynamic shadowing
  - Materials
  - Various render modes (e.g., winding, culling, blending, shading)
  - Multi-pass rendering
  - Morphing
  - Wireframe-rendered models
  - Environment mapping
  - Textures, including multi-texturing
  - Rendering order, layering
  - Level of detail for 3D objects
- Animation framework
  - Modification of 3D object attributes over time
- Key frame-based animations
- Animation playback with adjustable speed
- Built-in interpolation strategies
- Application-controlled world time
- Asset management
  - Scene Composer resources (e.g., scenes, trees, 3D models, textures, text and fonts, shaders, animations, widgets)
  - Different VRAM-upload strategies
  - Support of “on-the-fly” updates
- Optimization
  - Optimized render order and state management
- Platform-rendered abstraction
  - Platform-independent engine
  - Tiny platform integration I/F
  - Hardware-layer configuration support
Candera Engine
This module runs the output of the Scene Composer. The Candera engine can run either on the host or on the actual target, and can support development of both 2D and 3D graphics.

Specifically, the Candera 2D engine supports dynamic scene graph and 2D animations, including smooth rotation, scaling, and translation of bitmaps. In order to enable seamless integration, the 2D engine supports interaction with the 3D engine in several ways. For example, it supports render to texture, hardware layers, multiple displays, alpha blending, 2D widgets, and text rendering. It can also post-process 3D images and combine scenes with 3D.

The Candera 3D engine is based on OpenGL ES 2.0 and is fully compliant with the standard specification. It is OS, application, and hardware agnostic. Its key functions are screen and scene management, 3D object handling and processing, animation framework, and asset management in the embedded system's memory. It also optimizes render order and state management.

CGI Studio Player
The Player module helps in application development and verification, and in the design and verification of widgets.

Other Modules
Besides meeting other needs, the tool’s additional modules enhance its adaptability and globalization capabilities.

CGI Studio Courier is an interactive framework that handles data binding and messaging with the host system, allowing smooth integration of this tool. It can interface with Matlab, VisualState from IAR, and other proprietary state machine tools. Courier facilitates rapid application prototyping, easy simulation, changes and porting to any customer platform.

Analyzer helps assess application performance, detect bottlenecks and optimize the final design.

Translator provides context-based translation for different languages. This single, central, translation repository supports a highly parallel, iterative process.

Other modules, such as the Photoshop importer, HTML5 module, and functional safety module, serve unique purposes.

Summary of Key Benefits
- CGI Studio can be used for both automotive clusters and infotainment systems. This widely applicable solution has a scalable tool chain for both 2D and 3D UI, as well as for hybrid 2D and 3D scenes.

- The HMI tool enables early evaluation of graphics assets in the Scene Composer window using a WYSIWYG approach. Application simulation in CGI Player on the development system also gives a solid look and feel to the visual content.

- CGI Studio has many functions to help developers trace graphics assets, including asset-generation reports, asset verification, and version control. The tool can connect graphics assets with requirement management.

- The Scene Composer tool keeps the graphics assets and code separate, allowing re-skinning of graphics even in later project stages, and eliminating the need to rebuild the project as the graphics are modified.

- The same set of tools can be shared among the studio graphics designer, technical artist, and software engineer.

- Widget development can be done using industry-standard programming languages such as C, C++, or C#.

- The Analyzer module allows system benchmarking and performance analysis before the application is deployed to the target. This tool provides valuable feedback regarding the feasibility of the system hardware.

- On the target, CGI Studio also supports multi-threading, incremental updating and partitioning of asset libraries.

- The state-of-the-art feature set supports HTML5, OpenGL ES 2.0, interoperability with GenIVI, and functional safety (ISO26262).
Conclusion

In short, the CGI Studio HMI tool chain enables system developers to seamlessly implement a graphics application from concept to actual embedded implementation. This makes possible such key benefits as cost effectiveness, a seamless process, separation of code and user-interface data, and system benchmarking.

Fujitsu CGI Studio:
Major Features & Benefits

- Cost effectiveness
- Seamless operation
- Strict separation of code and user interface data
- Early system benchmarking

Advanced Fujitsu Graphics Display Controllers

An essential element of any graphical user interface is the graphics display controller (GDC) itself. Fujitsu is a leader in this marketplace.

The Fujitsu MB86R1x "Emerald" 2D/3D graphics display controllers are designed for high-end embedded graphical applications in the automotive market. The SoCs, which features an ARM® Cortex-A9 processor, support Fujitsu’s 360-degree Wrap-Around Video Imaging Technology.

The most recent member of the family, "Emerald-P," is the first SoC in the world with an integrated APIX®2 interface. The APIX technology, which was developed by Inova Semiconductor, is a standard high-speed link for video and control in various automotive applications.

The Fujitsu MB91590 "Sapphire" GDC series provides a solution for applications that require robust color display and video-input capabilities. The solution is particularly well-suited for automotive dashboard and HMI applications such as automotive instrument clusters and center consoles.

For more information about the Fujitsu GDCs, visit http://us.fujitsu.com/semi/gdc.

For more information about CGI Studio, visit http://us.fujitsu.com/semi/cgistudio.