Bleank Technology

OUR TECHNOLOGY

For more than five years now Bleank has been developing its own technology to meet the highest real time rendering standards. We give the greatest attention to detail to deliver the best quality possible on personal computers equipped with mid-range graphic cards. Our technology was developed with C++ in order to deliver the fastest and most confortable interactions.

C++ OBJECT ARCHITECTURE

Our technology was entirely developped using C++ as we believe it is the language of excellence when you need the best performance / productivity ratio.

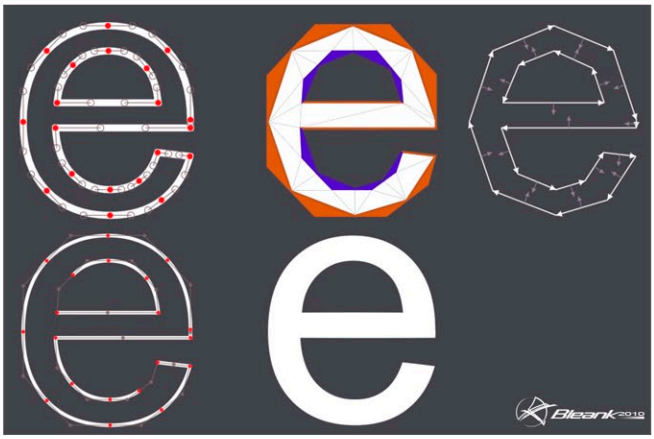
We developped an object-oriented architecture that can easily be improved whenever we need through the addition of independent components (plug-ins).

All these components still rely on the CORE module that provides the following C++ services :

- Runtime Reflection of an object's parameters
- A system capable of binding parameters across the module instances
- Multi-threaded handling of low level tasks
- Management of delegates and events (C#-like) between objects
- Memory allocation schemes through "objects pools"
- Operating System abstraction
- Basic mathematical library

2D HIGH QUALITY RENDERING

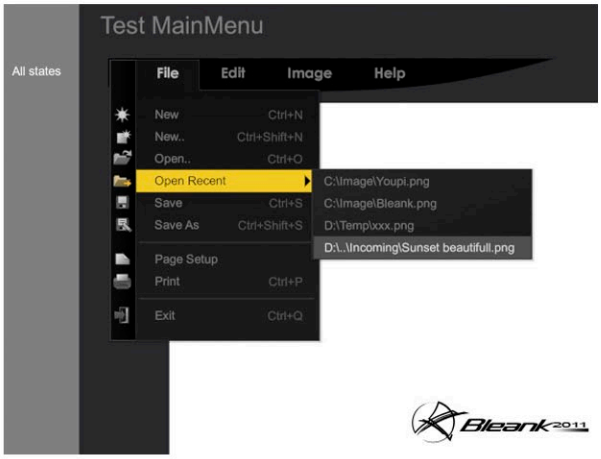
From early 2010, we focused our efforts on 2D rendering to rival and even surpass the services offered by technologies such as Flash or HTML5. Our technology is thus capable of rendering any type of 2D anti-aliased primitive in a single pass without the use of temporary buffers. We are now able to smoothly render animated vector graphics without any feeling of stuttering as is usually the case with non-subpixel primitives rendering.



Here is a non-exhaustive list of the features available in our 2D rendering engine :

- Brushes (Linear Gradient, Radial Gradient, Bitmap, ...)
- Primitives (Sphere, Rectangle, Shape, Curves, Lines, ...)
- Strokes
- Filters
- Specific font rendering system
- Linear-space anti-aliasing

Bleank Technology



USER INTERFACE

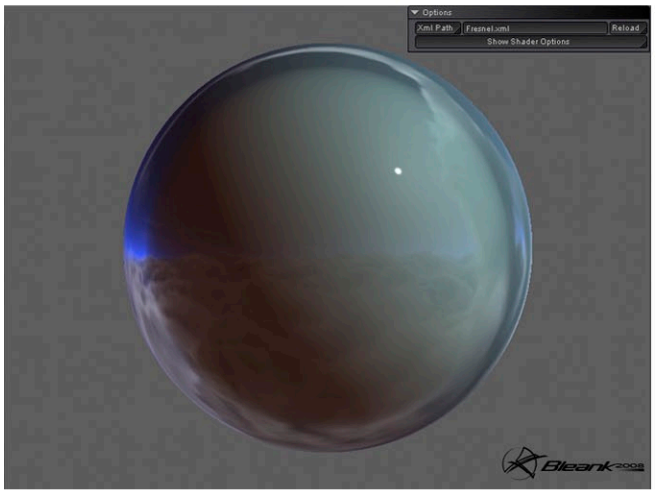
Our technology integrates a powerful user interface module containing a collection of objects and controls very similar to what the WPF API is providing like, for example, the panel classes (Grid, Stack, Dock, Canvas, ...) and controls like Sliders, ListBox, TextBox. Events propagate by following a dependency tree as well.

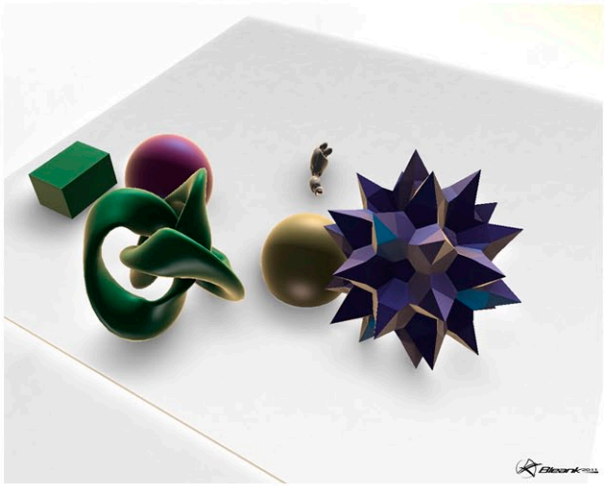
Here are a few features our GUI system is offering :

- Element state tracking that displays transitions between different states of an element via an animation. Very useful to move a button when the user clicks on it, for example.
- Templating for full description of the rendering of each element.
- Automatic interface generation from data through the use of "DataTemplates".

3D RENDERING

A 3D engine based on DirectX11 specifications that uses the full extent of GPU capabilities.
The engine can interface with the DirectX9, DirectX10 or OpenGL APIs through a powerful and generic abstraction layer.
It is also entirely based on a system of customizable shaders.





LIGHTING

Lighting in a 3D scene is undoubtedly the most important element to create stunning impressions of realism. This is why we have developed several techniques to achieve realistic rendering, adapted to every type of scenes.

- For static scenes we have integrated Spherical Harmonics rendering, a very useful technique that compresses radiance arriving at a point. This is a powerful way to obtain convincing light atmosphere.
- For dynamic scenes, we have implemented Screen Space Ambient Occlusion and Global Illumination techniques.

Bleank Technology

POST-PROCESS EFFETCS

Our technology includes a complete post processing effects set. These effects enhance the picture by adding components such as : glow, glare, motion blur and so on. This step is crucial to depict lighting, motion or depth of field effects.

Some features of our HDRI Tone Mapping system :

- HDRI Tone mapping
- Glares / Streaks generations
- Anti aliasing
- Depth Peeling
- Depth Of Field

VOLUMETRIC EFFECTS

We created a volumetric rendering component during the development of our projects. It can render complex 3D effects such as smoke, clouds or shadows projected through participating media.

It can also voxelize any traditional 3D object and compute pure volumetric shapes from noise functions.