MultiTouch Cornerstone

Software for multitouch interaction and object tracking

High-performance tracking engine and SDK

MultiTouch Cornerstone Engine

- Computer Vision Through Screen tracks unlimited numbers of fingers, hands and objects
- Uses rear diffuse infrared technology
- Scalability up to 14 Full HD displays
- Fast tracking, up to 120fps
- Supports Windows, Mac and Linux operating systems
- Supports multiple programming languages and protocols
- Runtime libraries included with MultiTouch display products
- Clustered tracking for massive installations

MultiTouch Cornerstone SDK

- Tool for creating interaction-rich multitouch applications
- High performance C++/OpenGL development environment
- Flash/ActionScript 3 development environment
- Multitouch emulator for developer use
- Supports wide range of development environments
- Gesture engine
- Integration with multiple tracking engines
- Hardware independent

Unlimited

You can simultaneously track as many fingers, hands and objects on display as you need.

Scalable

You can drive up to 14 displays concurrently.

Flexible

MultiTouch Cornerstone supports a range of operating systems and development environments.

MultiTouch Ltd, Kiviaidankatu 2 B, 00210 Helsinki, Finland // www.multitouch.fi // sales@multitouch.fi // tel. +358 45 630 8580
MultiTouch Cornerstone supports multiple development environments and methods.

**Core Features**

- **Extendable**-based C++ interaction engine guarantees smooth performance with easy scalability.
- **Integrated** animation environment allows the extension of widgets in a flexible way.
- **CSS** support allows applications, or their individual components to be controlled via CSS files. This allows for easy control of specific properties as desired, using a familiar file format. Cornerstone uses a simplified version of CSS, but components can be used to adjust not only the look of the widgets, but also the behavior of the widgets as needed.
- **XML** file format for widgets makes it possible to load/stored widgets from the hard-disc. This can be used to add application development so that application blocks is defined in the XML file, rather than specifying them in code.

**Core Technologies**

The following technologies are used throughout the Cornerstone SDK:

- **Multi-display rendering** provides support both LCD displays, and projector systems. For non-projector systems, the rendering engine includes keystoned connection, and projector edge blending features. With these options one can achieve smooth transitions between adjacent video projectors, without the need to scale to a high-end projection. The image warping is performed in the graphics hardware, with minimal overhead.

**Multi-channel audio processing** framework enables the use of spatial audio, as an integrated part of the applications. Different audio streams can be targeted to either specific loudspeaker/headphoniness, or stereo or multi-channel audio for delivering the sound where the content is.

**Background threading** of complex tasks frees the application to run in real-time even when consuming operations need to be performed. Use cases for the technology are data base access and media streaming.

**Plugin Components**

While the most important widgets are included in the Cornerstone SDK, the SDK allows for easy extension of the applications by adding additional widgets and capabilities. This can be done in a number of ways, including:

- **WebBrowser** - a minimal multitouch application
- **VirtualKeyboard** - shows how to use threaded rendering
- **ItemFlow** - demonstrates the use of the ItemFlow widget
- **InputFlags** - how to use the basic widget class
- **Layer** - another widget customization example
- **Marker** - how to use threaded rendering
- **MarkerView** - demonstrates the use of the ItemFlow widget
- **Events** - shows how to use the internal event manager

**Upcoming Components**

- **Example plugins** include:
  - **PluginSample** - a very basic plugin widget
  - **CSSSample** - how to use CSS to control the colors (fonts, colors) etc. of the widgets
  - **Browsers** - shows a bookmark list, and can also switch between different browsers
  - **MarkerView** - shows how to access marker data from the applications
  - **InputFlags** - how to use a simple function call.

**VirtualKeyboard** - allows for easy extension of the applications by adding additional widgets and capabilities. This can be done in a number of ways, including:

- **GigaImageWidget** - with multitouch support
- **High-speed sprite rendering** which is already used in a number of applications

**Layer widget** adds multitouch awareness to any application. Available effects include:

- **Lightphases** - light-moving things that move around using a finger's gestures.
- **Champagne** - simulation of champagne bubbles
- **Graffiti** - simulated graffiti painting
- **Lighting** - dimming lines of electricity between the fingers on the screen.
- **Cut** - generates snow-flakes and other organic shapes when interacting with the screen.
- **Layer** - another widget customization example
- **Marker** - how to use threaded rendering
- **ItemFlow** - demonstrates the use of the ItemFlow widget
- **Events** - shows how to use the internal event manager
MultiTouch Cornerstone

1. 1 PC, 1 Cell
   - 1 PC running 1 Cell. PC can be running Window, Linux, or OS X. Application can be written with C++ or Flash or any other language using TUIO, XML or Windows 7 Touch.

2. 1 PC, 2 Cells
   - 1 PC running 2 Cells. PC can be running Window, Linux, or OS X. Application is written with C++ or other suitable language using TUIO or XML outputs.

3. 1 PC, 4 Cells
   - 1 PC running 4 Cells. PC can be running Window or Linux. Application is written with C++ or other high performance language using TUIO or XML outputs.

4. 1 PC “Tracking”, 1 PC “Application”, 6 Cells
   - 1 tracking PC and 1 application PC running 6 Cells. The tracking PC is running Linux, while the application PC can be running Windows or Linux. Application is written with C++.

5. 3 PCs “Tracking Cluster”, 1 PC “Application”, 14 Cells
   - 3 tracking PCs and 1 application PC running 14 Cells. The tracking PCs are running Linux, while the application PC is running Windows or Linux. Application is written with C++.

6. 6 PCs “Tracking Cluster”, 1 PC “Application”, 24 Cells
   - 5 tracking PCs and 1 application PC running 24 Cells. The tracking PCs are running Linux, while the application PC is running Windows. Application is written with C++.

7. 1 PC, Projector box with 4 projectors in portrait mode (3 meters wide)
   - 1 PC running a projection box with 4 projectors in portrait mode. PC can be running Window or Linux. Application is written with C++. Projector keystone correction and edge-blending are handled by Cornerstone.

8. 3 PCs “Tracking Cluster”, 1 PC “Application”, Projector box with 8 projectors in portrait mode (10 meters wide)
   - 3 tracking PCs and 1 application PC running 8 projectors in portrait mode. The tracking PCs are running Linux, while the application PC is running Windows or Linux. Application is written with C++. Projector keystone correction and edge-blending are handled by Cornerstone.