carmenta
geospatial innovations

Carmenta Engine 5.2

For Windows and Linux

General

- Software Development Kit (SDK) for rapid development of interactive geographic applications.
- High performance visualization and processing of geographic data.
- 32-bit and 64-bit versions. Supports development of native 64-bit applications on 64-bit Windows.
- Fast kernel developed in optimized C++.
- .Net API [1] that conforms with .Net guidelines, C++ API and Java API.
- All APIs have fast direct access to the kernel, i.e. no middle layer such as COM is used.
- Optimized internally for multi core CPUs.
- Hardware accelerated map visualization that offloads CPU.
- Parallel asynchronous processing that keeps the application's user interface responsive. Map layers may update in separate threads.
- Map controls for WPF [1], WinForms [1], Qt and QML, Win32 [1], MFC [1] and X11 [2].
- Optimized for moving large numbers of objects on top of map.
- The kernel has a built-in Profiler for measuring and tuning the performance of map and application layers.
- Built-in tiled map image file cache, suitable for optimizing vector layers on low-end hardware.
- Presentation of map layers and application layers is defined in map configuration files, and can be changed at runtime using the APIs.
- Map configurations are visually defined using the SDK tool Carmenta Studio.
- Component based architecture built around intuitive data flow paradigm.
- Easy deployment using runtime merge modules (.msm) [1] or XCopy deployment.
- Supports over 70 geographic data formats natively.
- Different layers can have different coordinate systems.
- Vector and raster data reprojection is done on the fly.
- Geographically correct spatial calculations such as projection, distance, great circle, scale factors, azimuth / angle calculations etc available for all reference systems.
- Maps can use both locally stored GIS data and layers that are accessed via remote servers (OGC WMS and WCS).
- Multiple windows and views can present the same data with different visualizations simultaneously.

Visualization

- Visualizes points, lines and polygons. Handles different kinds of dashed lines, pattern fills and textures.
- True transparency on all drawing operations, including raster data.
- Anti-aliasing of text, symbols, lines and polygons removes jaggedness in presentation.
- Raster filtering, bilinear or bicubic, improves visualization of scanned maps etc.
- Off screen drawing to bitmaps, files or memory that can be further processed (PNG, GIF, JPEG and BMP).
- Raster symbols from file (TIFF, PNG, GIF, JPEG and BMP).
- Vector symbols (SVG) from file or string.
- Resizable, rotatable vector and font symbols. Halo and outline effects.
- Multiple visualizations on objects such as multiple texts and symbols at a point or line.
- Complex line styles with auto-placement of symbols or labels along lines or in nodes, texts or symbols that clip lines etc.
- Level of detail in 2D with automatic switching on/off of map layers based on scale and/or geographic area.
- Flexible layer handling controls the drawing order of layers in run-time. Ability to mix raster and vector layers in arbitrary order.
- Attribute data controlled visualization, selection and discrimination.
- Animation support with double/triple buffering for smooth window updating when displaying dynamic data and editing objects.
- Automatic label placement of text and symbols prevents overlapping.

Extending with Custom Components

- Developers can ”plug-in” their own data reading, processing and visualization code as components that fit seamlessly into the Carmenta Engine data-flow model.
- Custom processing can be developed using the .Net, C++ or Java API.
- Custom visualization can be developed either using native GDI or OpenGL or by using Carmenta Engine high-level graphics API.
- Custom processing written in Python script can be embedded into map configurations.

Software Development Kit Contents

- Rapid prototyping development tool.
- Carmenta Studio - a visual editor for map configurations.
- Carmenta Explorer - a map configuration viewer.
- Comprehensive documentation, including tutorials, technical articles and API documentation.
- Sample applications with source code for .Net C# WinForms and WPF, C++ for Qt, MFC, Win32 and X11.
- Sample map configurations explaining how to do typical world, topographic and street maps.

[1] Not available for Linux.

[2] Not available for Windows.

Geographic databases and formats

- Tile handling with the ability to create seamless map layer presentation from many individual files.
- Can generate low resolution variants (”pyramids”) of raster data to improve reading performance.
- Application developers can add support for additional formats with their own data plug-ins.
- Support for spatial database queries.
- File indexing for efficient reading of large datasets.
- Reads directly (conversion is not needed) from a large number of formats:

ADRG
ARINC [2,4]
ASRP
AutoCAD DXF [4]
AutoCAD DWG [4]
AUX
BIL
BSQ
BSP
BSB Nautical
BMP [1]
CADRG [4]
CEOS (Spot)
CIB [4]
CMRG (PCMap) [4]
DEM
DFAD [4]
DTED
ECW
ELAS
Envisat N1

ESRI Shape (.shp) [1,2]
ESRI Shape 3D [2]
ESRI Binary ADF
ESRI ASCII Grid
Erdas IMG
Erdas LAN/GIS
GeoSoft raster
GeoTIFF [1]
GIF [1]
GridASCII
HDR
Intergraph raster
Japanese DEM
C-MAP CM93 [3,4,5,6]
JPEG (.jpg) [1]
JPEG2000
MapInfo TAB
MapInfo MIF
MFF
MFF2
MrSID

MySQL [4]
NITF
NOAA
Oracle Spatial [4]
OGC WMS
OGC WCS PolGASP
PNG [1]
Raw
RPF [4]
S57 Nautical Charts [2,3,4]
SDTS DEM
SQL Server 2008 [4]
TIFF [1]
USGS ASCII
USGS DOQ
USRP
VTP BT elevation
VPF
SRTM HGT
WMO GRIB1
WMO GRIB2

[1] Reads and writes.

[2] Uses advanced spatial indexing technology for fast access of large files.

[3] With optional S52 nautical chart presentation.

[4] Functionality available as an additional Carmenta Engine Extension.

[5] Not available in Carmenta Engine Linux version.

[6] Not available in Carmenta Engine 64-bit version.

Interaction

- API calls for querying geographic objects at a screen position.
- High-level interaction tool interface that developers may use to ”plug in” their own interaction handling.
- Developers may also handle low level event and use API to control interaction.
- ”Flying carpet” tool for moving 3D camera.
- Tool for panning, zooming and rotating 3D worlds.
- Tool for panning, zooming and editing in 2D.
- Tool for creating 2D objects.
- Overview window functionality.

Data processing “on the fly”

- Add elevation (z) from elevation raster to 2D objects.
- Buffer zone generation for raster data and vector data.
- Clipping of geographic points, lines, polygons and meshes by geographic polygons or viewing area.
- Connection / desegmentation of lines and polygons.
- Data reduction through line and polygon ”thinning”.
- Filter objects based on their attribute values with the use of logical expressions.
- Geographic and UTM/MGRS grid generation.
- Generate circle and ellipse objects from point + radius.
- Hill shading with dynamic sun position.
- Isoline calculation, e.g. elevation contours from gridded data.
- Merging heterogenous rasterdata with different resolutions into homogenous data.
- Offset position of objects.
- Rasterizing 2D vector layers.
- Raster value convertions, e.g. floating point to integer, RGB to indexed color and vice versa.
- Reclassification of raster values and vector attributes.
- Reprojection of vector and raster data.
- Resampling of raster data.
- Size calculations - polygon area/perimeter and line length.
- Terrain accessibility analysis for terrain vehicles.
- Transformation of line objects to polygons and vice versa.
- Smoothing of line and polygon shapes.

Coordinate Systems and Projections

- Configurable reference systems, projections and geodetic datums, support for EPSG IDs.
- Handles embedded reference system information for the most commonly used geodata formats.
- Supports more than 15 types of projections, including Mercator, Transverse Mercator, UTM, Lambert, Albers and Stereographic.
- Polynomial projections for georeferencing using control points.

System Requirements

Operating Systems:
Windows XP SP3 32-bit, or Windows 7 and Windows Vista SP2 both 32-bit or 64-bit (supports native 64-bit applications).

Linux: Red Hat Enterprise 6, Ubuntu 10.04 LTS and SUSE Linux Enterprise Server 11 SP1.

Graphics devices:
Using OpenGL as renderer requires a device supporting OpenGL 1.1 and GLX 1.3 (Linux). Using DirectX renderer requires a device supporting DirectX 9.1 (Windows).

Development tools:
Visual Studio 2005, 2008 or 2010 (Windows).

Carmenta AB, Tel +46-31-775 57 00, www.carmenta.com