CASE STUDY: Simulation

In the Southern part of Finland, the Archipelago with its many small islands means that boats are the main means of transport. Navigating the area is extremely complex and can be dangerous and challenging due to traffic density and inclement weather conditions. Further, when planning your boat trip, regular shipping traffic (such as large transport and container ships and ferry routes) also has to be taken into consideration.

This case study focuses on Finnish APL programmer Tomas Gustafsson and how he – and a tiny team – developed the incredible *Stormwind* navigation simulator, which won the Open Data Innovation Contest Apps4Finland in 2013.

In many ways, the Stormwind Simulator represents an almost 'lunatic' approach to a bunch of challenges that traditionally belong to a variety of separate domains. It is a realtime 3D simulation, and could almost be considered to be a game. Stormwind manages a vast amount of geodata and visualises the real world by combining artwork with complex maths and physics. You could compare it to an entire manufacturing chain, from processing billions of measurements to running highly optimised code in a loopy manner. It's about creating a working solution while simultaneously modeling it, without knowing in advance if it's even possible.

The team behind the solution is tiny, yet the outcome is hugely impressive – and Tomas states emphatically that "only APL can cater for this". The toolset behind Stormwind is carefully defined and mastered, and tightly kept in hand. All of the source code is written in-house, both that for managing and refining the huge amount of data and that for executing the application in the end user's computer; hence Stormwind is written in one single programming language, Dyalog! The only external software distributed with the Stormwind Simulator (the package, with all assets, etc.) is the third-party 3D engine. This middleware connects to Windows DirectX over the .NET interface.

"APL gives us what I think can be accurately named 'data at your fingertips'. Using APL, we are able to tweak, alter, cut and glue the data, and then use it, experience it, with only a few handgrips. We do this interactively; we try and test, but as it also looks good visually, the production code sits there in front of us. We do this ad hoc, as true visionaires. There is no-one to direct us with boring statements like 'you must first dimension it', or 'you must first set up that loop', or 'compile it first'. We like to think of APL as our secret weapon. And yes, we know it's a bit unfair to the competition." Tomas says with a smile.

The Concept

The original idea was to create a simulation of the real world using only raw data. This is a procedural process.

According to Tomas "At no point do we modify the original meaning of the data – it wanders straight from a first-time entry into a database (somewhere out there), straight to a colored pixel on the computer screen. We do, however, change its shape multiple times on its long journey from database to screen, but the meaning and the subject that a numerical value represents is always preserved. Eventually, it ends up in a visualisation identical to its origin, namely, the item it was originally measured from. This could, for example, be a ground height, or a description of a real-world item, for example, the blinking sequence of a navigational light attached to a buoy.

"Over time, mankind measured the surroundings and recorded the data in multiple formats – such as navigational maps, weather predictions, tidal movements, landscaping, etc. Units were created as needed.

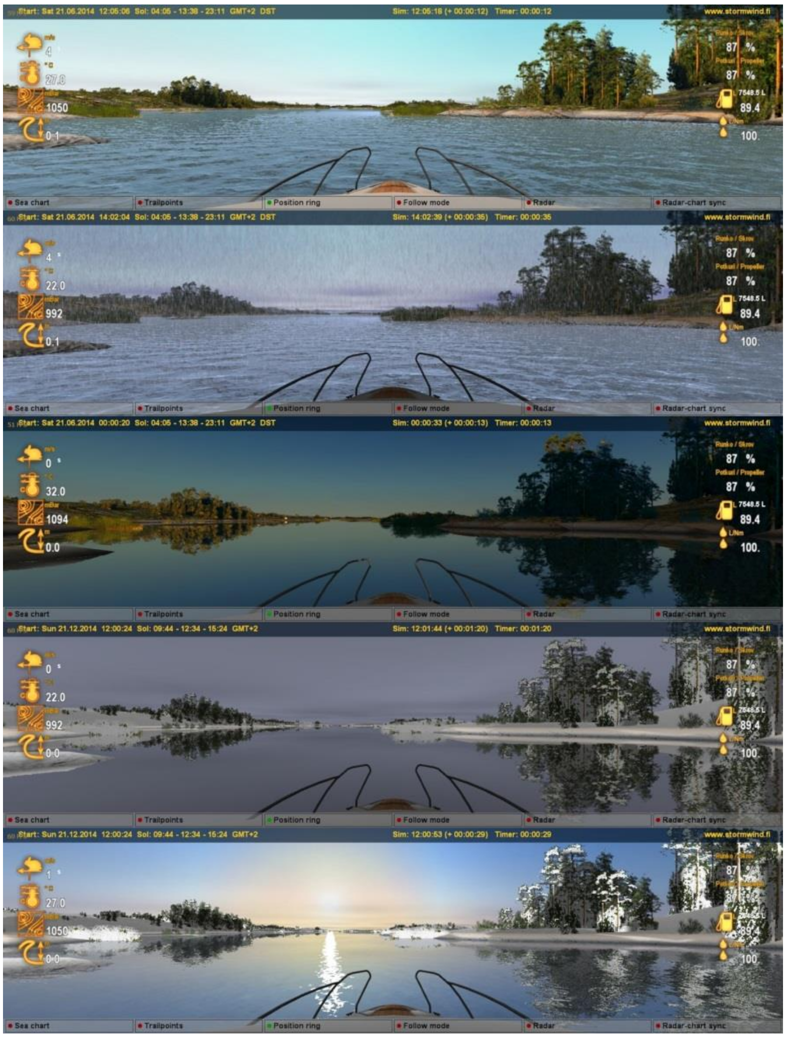
"Stormwind takes advantage of these measurements and we use metres, latitude, longitude, seconds, angle degrees and temperature degrees, kilogrammes and newtons to describe distance, size, place, time, view, energy, burden and movement. When, for example, a house was built, someone, somewhere, documented its dimensions and corner co-ordinates. Those figures are persistent, as the community maintains them.

"The cornerstone of Stormwind is quite simply the existance of data! Easy access to all sorts of interesting data is facilitated via the internet, and when the European states joined forces, we suddenly had something really good to work with. The EU *Inspire directive* stipulates that member states should – at a minimum – show the public what data is available, and, if they really want to take the big step, make its data available for free. This is actually what Finland did! We now have a vast amount of public data available to download. The golden thought process behind this step is that the availability of free data generates new technology and business and, as a result, new income for the community. This is a much more healthy way to allow and inspire than the old way of putting a price tag on everything."

Getting into Gear

Tomas continues "On the one hand, we had data. These data were probably sufficient in terms of quantity but – unfortunately – not in the formats we wanted, nor even with the isolated logical meaning we needed. But we were convinced APL would do the necessary job with the data. And we succeeded! Stormwind won the Open Data Innovation Contest *Apps4Finland* in 2013 and winning an Apps4Finland award is no small feat – given Finland is the country that also gave the world the game of Angry Birds! I cannot over-emphasise the strength of Dyalog for all this. Nothing stresses a computer system like a game or simulation, both in terms of data management and performance demand. With APL, we get both; we have the data at our fingertips while we get an enormous code line throughput.

"On the other hand, we had a vision. We wanted to draw an experience to the computer screen and strengthen that experience by using sound effects. While computer game makers start by creating a storyline and a world, probably a fantasy one at that, we had a real world already. Our world is the archipelago, as experienced from a boat. One would imagine there isn't much to do on that front; water is water, and it goes on until you meet land. But this is absolutely not true in the Finnish archipelago, considered to be the world's most complex waterway. There is an entire traffic system embedded out there, just as airplanes have one in the air, although 'air is just air'. The archipelago gives slightly less freedom to move around, since we also need to consider islands and water depth. Everyone knows that being outdoors under varying conditions implies something; you experience, learn and get feelings. Together this forms a part of your instinctive life. For example, being out in a forest after a shower of rain projects a certain feeling, as does spooky moonlight at midnight. We already knew what it feels like to drive a boat in the archipelago, and we wanted to let those feelings flood out from the computer screen together with a technological correctness, carried by wave physics, outdoors light management, boat movement, navigation instruments, etc. The eventual purpose is to let the user train, because driving a potentially fast boat is not at all an easy task, and each year there are dozens of hazardous or deadly boating accidents."



Automated fn(location, date, time, weather). Same place, same view angle, different times and conditions. Light and colour management is one of the trickiest things to code. A scientifically accurate computer representation of the world will not work; instead one has to submit a feeling and let the human mind complete it. Note the third image, which shows the incredible amount of light at Scandinavian midnight.

Grabbing the Data

"We were able to download a lot of relevant data from the National Land Survey's web pages, in addition to some other governmental sources. This is a rather mechanical job and clearly the data providers aren't quite 'there' yet in terms of customer service. The downloads require a lots of manual work and are generally rather clumsy. The acquired data sets are raw data and need translation and modification by custom APL code. But these are one-time steps and not overwhelmingly exhausting to perform. The most relevant data sets are land heights, terrain data, building data and forestry data, plus traffic system data. Suddenly, we sat there with a rather large number of gigabytes on the hard disks."

Refining the Data

"At this point, we had a concept for the data storage to be used by the end application. A run-time simulation cannot do lots of ad hoc data altering work, so everything must be pre-optimised. The data must be in a format that is almost ready to be thrown out to the rendering pipeline. Another thing to consider is the data granularity. Under no circumstances can we load multiple gigabytes into computer memory while filtering out only part of it for rendering – that would be too time consuming. And under no circumstances can we render multiple gigabytes on-screen and expect that to happen 60 times per second. There must be a paging mechanism that is able to live with whatever data is present at any location. Hence all data was split into small pieces, refined numerically, and output into a common co-ordinate system. The refining process relies heavily on pixel processing, since that is much faster than CPU-looping horizontally and vertically through tables. Doing part of the mathematics on the GPU instead of the CPU saved us lots of waiting time, although the entire refinement process for 30,000 square kilometers, from start to end, takes over 24 hours to complete. This is truly a matter of APL forcing the hardware into hard work; the total data throughput is counted in terabytes when APL moves the data segments back and forth while calculating things like radar shadows and visibility lanes."

The Application

As already noted, this was an iterative process. There are two major constituent parts to this; visual quality and rendering performance.

Tomas explains "We have a frame rate to deal with and we cannot allow it to look like a Charlie Chaplin movie with 20 frames per second. On the other hand, one could draw just about anything if there was time available. During the construction of the software we were forced to learn about today's hardware. We aimed at using GPUs that were a few years old and squeezing a lot out from those. This was found to be a good choice. The product is aimed at customers with either desktops or laptops, so we have not aimed for cutting edge and extremely expensive hardware to run the applications – although we're not too far from that either. A majority of the mainstream hardware is not aimed at gaming and thus not suitable for a realtime 3D simulation. We made a choice that was good enough, one that perhaps encourages people to upgrade their hardware, and we managed to take care of every interested customer in a successful way."

The Stormwind Simulator uses APL code connecting over Microsoft .NET to a third-party 3D engine, TrueVision3D, that in turn drives the Microsoft 3D platform, DirectX. The program runs in a loop, and there is at the most c0.03 seconds available for a cycle. It was a surprise to the team how fast APL is at processing simple data – a common belief has traditionally been that APL is at its best when dealing with huge data. That is indeed true, but it seems Dyalog developers have put in lots of effort in dealing with small data as well. The code throughput of APL is rather impressive, one loop cycle may invoke thousands of very complex APL code lines, and knowing what goes on behind the scenes with even one single APL symbol makes it all rather stunning.

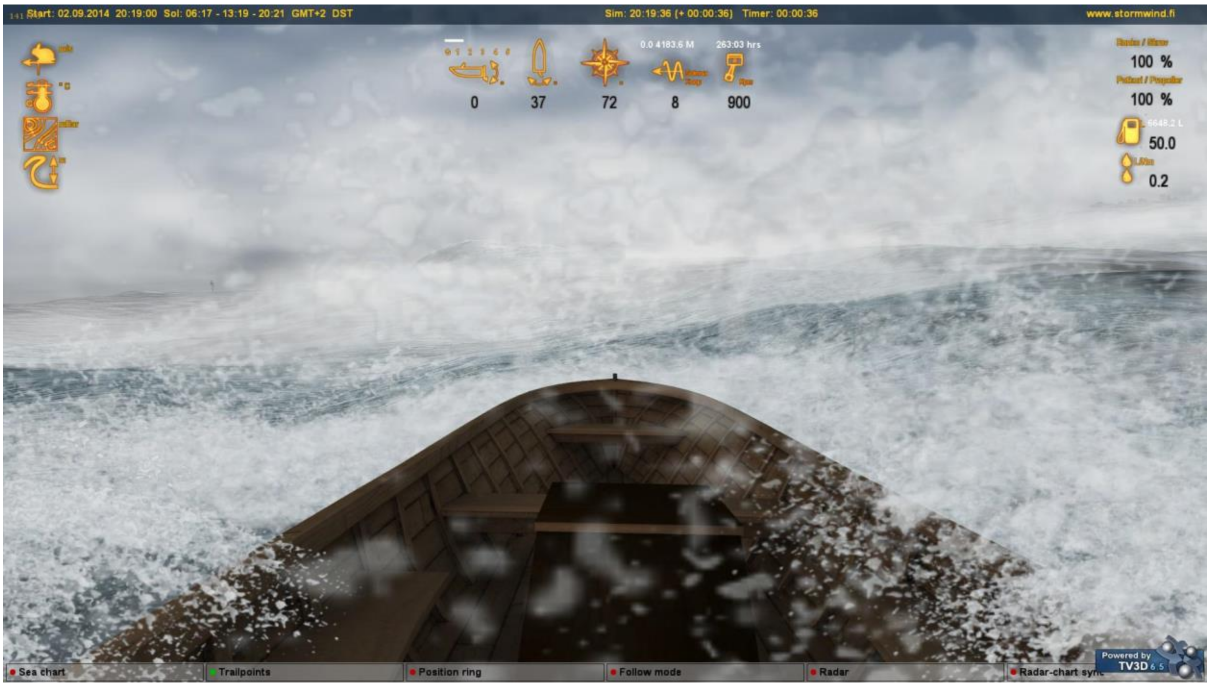
Tomas continues "In a loop, as mentioned above! Most Windows coders are used to having a callback mechanism, where Windows runs a message loop internally and invokes an appcode callback function when something happens – probably a user action in the user interface. APL coders are used to have a Wait command somewhere in the code. We know that APL suspends code execution at that point and waits for user actions, timers, data transmissions, etc. to occur. We have nothing of this sort, but nevertheless we do use Windows GUI elements in the application. A callback may appear at any point in time and it may originate from another thread. This potentially invites a real mess, but APL handles this gracefully. When needed, we are able to inject various out-of-sync events into a queue system that the main thread handles, and that way we restore order while the application continues to run smoothly."

The program creates a visualisation of 30,000 square kilometers of Finnish archipelago. It handles space and time, light amount, weather simulation, wave simulation, sound effects, boat physics and manouvering and other traffic together with AI (Artificial Intelligence) and pathfinding, radar simulation, sea charts viewing and much more. It utilises a number of 3D techniques for near photorealistic outdoor rendering. It supports up to 3 monitors that are re-configurable at runtime. It supports game devices like steering wheels and joysticks. It is able to control motion platforms and can send out NMEA (National Marine Electronics Association) data over a USB port. A multi-player version is under development.

Evolution and User Base

The Stormwind Simulator has attracted a lot of attention. There have been a number of stories in Finnish boating magazines, as well as in newspapers. The simulator has appeared in radio programmes. And as mentioned above, Stormwind won the apps4finland contest 2013; the competition for usage of national data. During the past 5 years, Stormwind has been a central visiting point in Northern Europe's biggest boating exhibition, the Helsinki Boat Show.

"Among the professional users we work with are the Finnish Naval Academy, a number of rescue departments, the Finnish Lifeboat Institution, navigation schools, two science centres and, as the largest user group, private consumers. Unfortunately, our country is small and the product rather specialised. Had we taken the gaming path, the user base would probably have been larger by now. But we have decided to wait and see for a bit longer. There are great plans for the future, perhaps we'll have a chance to review those in an upcoming study." Tomas concludes.



Prepare for a rollercoaster ride – review the Stormwind Simulator at:

- https://www.youtube.com/watch?v=MzrMjpJouDs Tech demo 4
- https://www.youtube.com/watch?v=v1rdS1sRX18 Tech demo 5
- http://video.dyalog.com/Dyalog13/?v=O63M3u0sDAg Tomas' presentation at Dyalog '13