

.werkzeug for mobile

.tekknology for mobile

.theprodukt GmbH
Rellinger Straße 7
D-20257 Hamburg
www.theprodukt.com

july.2005 (revised June 2006)

personal contact: [oliver.joey.waechter](mailto:oliver.joey.waechter@theprodukt.com)
waechter@theprodukt.com

the problem / the current situation

At the moment, Mobile Gaming is regarded as one of the sectors with the highest potential in the mobile industry. However, there are some technical limitations to fully utilize this potential.

Limitations, not just in displaying complex graphics, but more importantly in the amount of data that is needed for proper in-game graphics. This data needs to be transferred to the mobile device and to be stored there. So the bottleneck is bandwidth and storage capacity.

Standard compression technologies for textures used at the moment, like JPEG compression, are 2nd grade solution, since JPEG images are still far too big for high quality game environments on mobile devices and higher compression rates are always in correlation to reduced image quality due to increased artefacts.

the effect

The number of high quality games, that may use the upcoming graphics power of next generation mobile devices is very limited, because developers hesitate to work with the above mentioned limitations.

Existing mobile games hardly meet the gamer's expectation and thus the number of sold games is currently rather low, which in turn does put developers on hold.

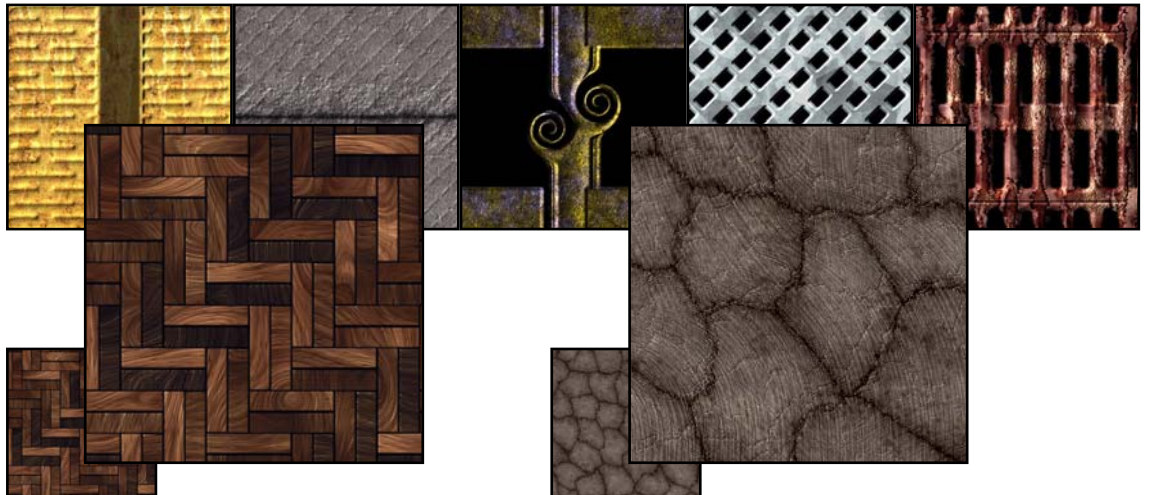
Furthermore this problem is not limited to the sector of gaming but spreads out to other sectors. Imagine the possibilities of 3d interfaces and menus on mobile devices that are hardly implemented at the moment, but should be a standard on next generation's devices.

A vicious circle that needs to be broken!

the solution

Rather than waiting for larger bandwidth and storage capacity to be widely available, the problem can be solved right now by using **generative computer graphics**.

Graphic elements (textures) are not transferred to, but generated on the mobile device. Only the relevant code with much smaller amount of data needs to be transferred and results in High Quality Textures.



the example .kkrieger

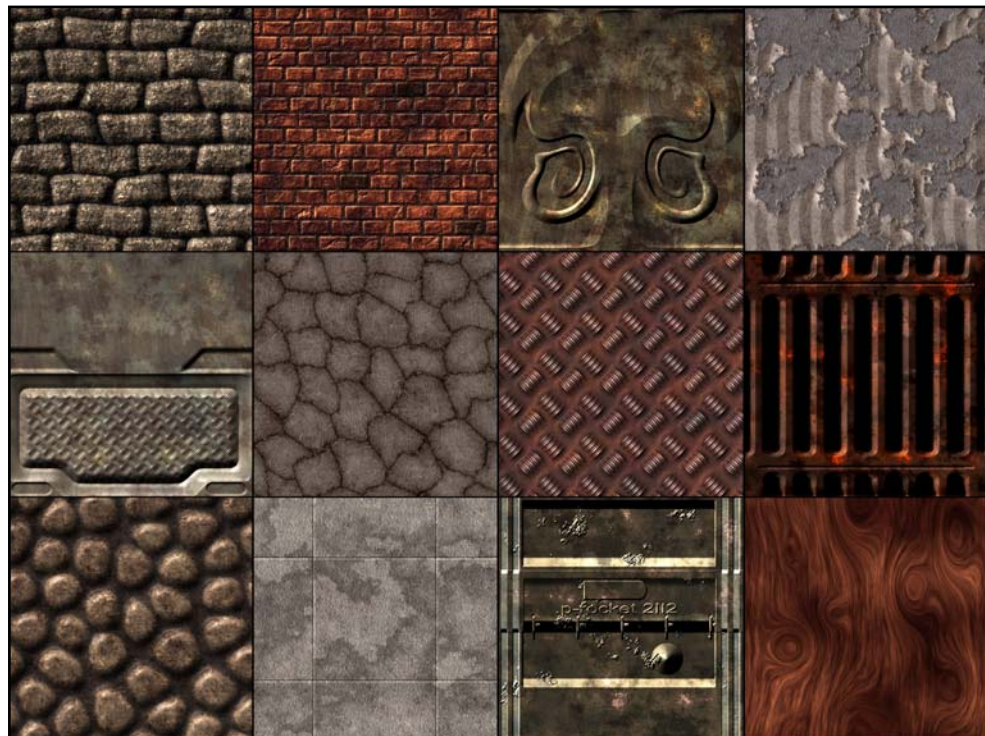
.kkrieger is more than just a proof-of-concept for generative computer graphics. It's a fully functional 3d first person shooter, done in 96KB!



First presented on the breakpoint in January 2004.

.kkrieger features the full range of first person shooter's features, like scoring (life and ammunition, monsters, music and sound fx; not to forget dynamic lighting, and 100% procedural textures!

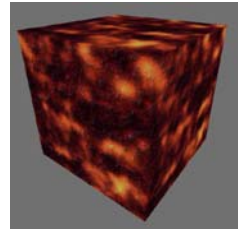
Still available on www.theprodukkt.com and widely popular, not just among the demo scene.



.werkzeug and .tekknology for mobile – background

Currently the work on .theprodukt's mobile toolset, consisting of .werkzeug and .tekknology, is being finished. Both proven tools, used for .kkrieger and various award-winning farbrausch demos. All algorithms used are highly optimised for filesize, while at the same time .werkzeug offers a most comfortable way to create the textures.

Thus our toolset covers the complete workflow for textures: Creating and optimizing on the PC, plus preparing (generating) for playback for the mobile device, so other applications, like games or menus can access these textures.



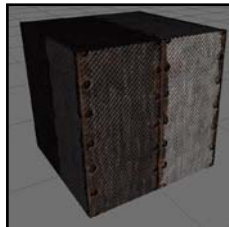
In contrast to hitherto existing approaches for texture generation, which are usually relying on a few simple perlin noise operations, .werkzeug can easily handle hundreds of operators (ops) leading to High Quality Textures as in .kkrieger.

The crew of .theprodukt is the longest active group (since 2000), using generative textures. As a result of this outstanding experience, all implemented ops (meaning effects/filters/modules/generators) and their parameters are well proven, tested and balanced. As a result, the user does create textures that will look exactly like he wants them to look. No trial and error, no randomness – always right on target.

.tekknology

All textures, created with .werkzeug, are a set of operators.

.tekknology takes these operators on the final device and generates the finished textures, ready to be used in whatever way. Textures can be stored in RAM or any storage device accessible.

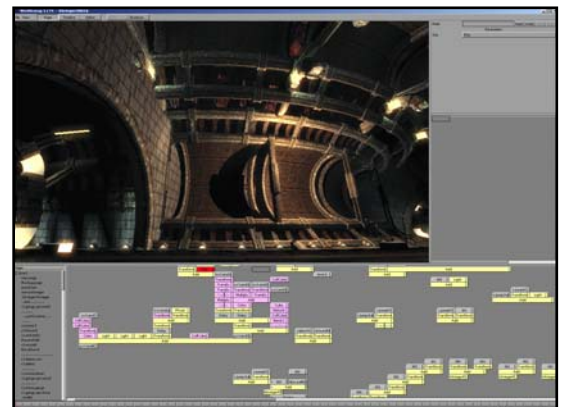


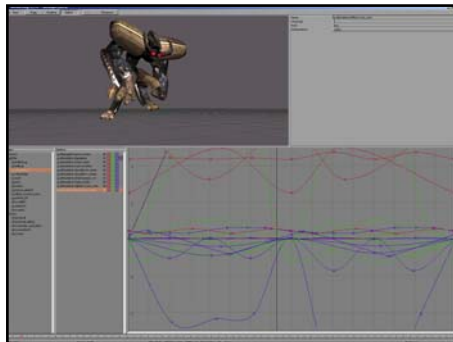
.tekknology is mainly C++ code that works best if implemented as middleware, which also allows for specific optimizations on given hardware to optimize speed.

Since .tekknology does not rely on proprietary hard- or software, any implementation of it is future proof and can be upgraded to future version of OS or hardware. Generated textures are not lost, when targeting different markets and devices, because they are 100% reproducible on any given hardware.

.werkzeug

To manage the large number of ops needed for good looking textures, we built .werkzeug. From the first version and the public release of .werkzeug 1.201 to today's .werkzeug 3.174 and .werkzeug for mobile, we have optimized its interface and workflow in real production for five years. It is far beyond the possibilities and options of normal tools. Seamless textures, multiple resolutions and libraries, unlimited undos and internal 16bit support go with out saying.



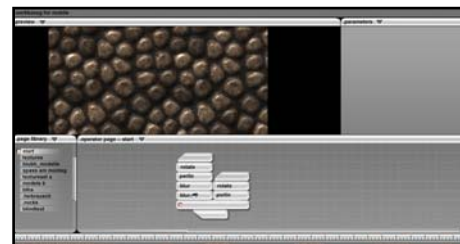


Since .werkzeug is absolutely focused on textures, we also implemented maximum support for creating bumpmaps, perfectly fitting the created texture.

.werkzeug can layer more than 100 ops with multiple different effects, while at the same time offering an intuitive user interface to work with these ops and stay in total control of the final result.

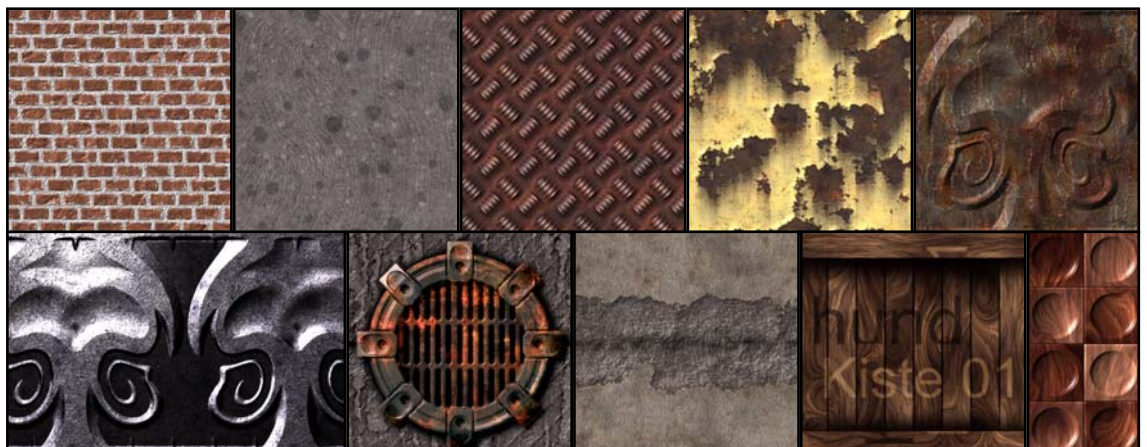
For the final commercial release of .werkzeug, we will offer a texture setup library (TSL), that enables the user to use as a basis for his own textures, rather than just using predefined textures (which is possible, too).

.werkzeug will be the designer's favourite texture generating tool.



the summary

- _ complete workflow
- _ extremely small filesize
(about 100 times smaller than comparable JPEG compression,
with absolutely no artefacts)
- _ 100% reproducible results
on any given hardware
- _ High Quality Textures, with
full control of the result
- _ internal 16 Bit support
- _ seamless tileable textures
- _ perfectly matching bumpmaps
- _ practice proven, easy to use toolset
- _ unlimited undo
- _ large material database



.theprodukt – company background

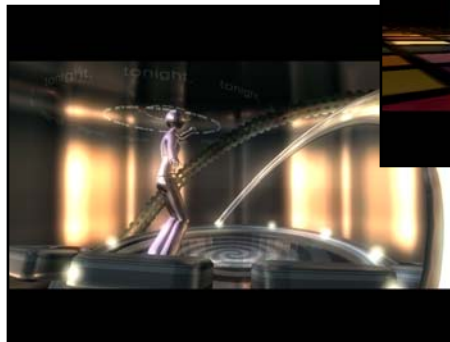
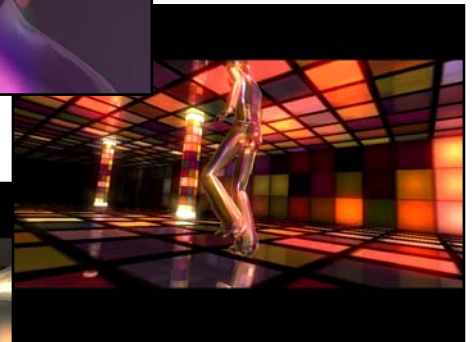
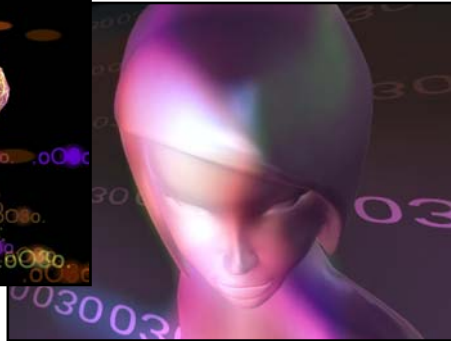
.theprodukt GmbH was founded in April 2004.

All founding partners have collaborated in the demo-group “farbrausch”.

Some examples of this collaboration:



fr-030: candytron
released 20-apr-2003



fr-025: the.popular.demo
released 20-apr-2003

