



http://www.bitfellas.org/e107_plugins/content/content.php?content.732

Page 1/8

12 96k Game Round Table

Magic, Tue 01 Apr 2008

96k Game Round Table

Written by Magic of Nah-Kolor

Hugi focused on the 96k games phenomenon. Games, 96k small in size, which mostly get released at Breakpoint every year.

Featuring some the best and most known 96k game coders on the demoscene:

Archee, coder of the BP'07 winner 'Sumotori Dreams';

KeyJ of Kaki-Arts, coder of the BP'07 3rd place 'kakiarts 8-ball';

Ryg of Farbrausch, one of the main coders of the BP'04 winner 'Kkrieger';

Stingray of Scarab, coder of the BP'05 Amiga winner 'Lotus Turbo Challenge'.

WVL of Xenon, coder of the BP'06 winning 'Pinball Dreams C64'.

Hugi: What is it about making 96K games that is so appealing to you?

Archee: I don't have too much time to work on scene products. In the small size categories, it's great, original ideas may win without too much time invested in art development.

KeyJ: To me, 96k games are the game equivalent to 64k intros: You have plenty of space to write something cool, but you have no space to waste either. Also, 96k games and 64k intros are usually quite code-centric, and I just love sizecoding

Ryg: Nothing in particular. I worked on kkrieger and don't plan on doing another one anytime soon.



http://www.bitfellas.org/e107_plugins/content/content.php?content.732

Page 2/8

Stingray: Just the challenge to squeeze a complete game into 96k.

WVL: Actually I didnt plan on making a 96K game when I started. I just wanted to prove it was possible to port Pinball Dreams to the C64 and try to make it as close a port as possible.

Hugi: How did the planning and making of your game go?

Archee: I started to engineer characters that can balance, move like humans. I was ROTFLING a lot at home when I came up with these clumsy guys, I thought it would be a good idea to share this with other people. The idea of making it a sumo just came at the end.

KeyJ: It started back in 2003 and it wasn't a demoscene project at all. At that time, I had a university assignment to write a virtual billiards game, that is, you play it on an empty table with a 3D tracking device instead of a cue. Parallel to the development of that project (which used some weird special VR API), I wrote a PC version using OpenGL. Frustrated with the bugs it had, I abandoned the project in late 2003, if I remember correctly. However, I rediscovered the game in late 2006, because it is an ideal candidate for a 96k game: The main game code was only 20k back then, but it already had decent gameplay features. So I essentially took the old code and ported it from the platform-independent, but large GLUT library to plain Win32. I also fixed the bugs thanks to intense play-testing conducted by my brother, added some new graphics and of course sound and music. At that point, the game was something like 60k in size -- plenty of space to add the really cool title screen, which bumped it up to 80k. The final version added two new music tracks and is 82k. In addition to that, I also created a stripped-down version without title screen, sound, music and a few other less important features, which is just 13k.

Ryg: Well, we first wanted to do a couple of Quake1-style levels with multiplayer in 96k. Going for perpixel lighting was a pretty obvious step at the time, because it's mostly about the graphics hardware you have; it doesn't really make the content creation much more difficult (you need to have normal maps, but that's pretty easy if you use generated content anyway). So we were getting this Doom 3-ish look, which was pretty cool but had the problem that there was now a huge discrepancy between the visuals (which were pretty state-of-the-art) and the gameplay (which was slightly more sophisticated than Doom 1/2). We developed the thing over a pretty long time of intensive work (most of our evenings and weekends for three quarters of a year), and later met for 2 weeks to work on it fulltime. It was pretty complicated because we needed to start creating levels and art well before the generation, engine and gameplay code was even near completion. So there was a lot of experimenting on all sides, a lot of content and code that later ended up being unusable, and it only started coming together very, VERY late into the project. It ended with a pretty awful crunch phase - we had all those seperate rooms, sometimes connected with a tunnel between them, but mostly very small segments. And then we were putting things together and it pretty much blew up in our face - RAM consumption was over 1.5GB in the first standalone versions, pretty much



http://www.bitfellas.org/e107_plugins/content/content.php?content.732

Page 3/8

of a death sentence really, especially since we were about 10k bigger than the size limit. Gameplay was awfully buggy (it still is in the released version) - it worked in the small segments, but finally putting things together caused a real avalanche of bugs to appear, when enemies would suddenly get stuck in doors, walk up walls or end up in the middle of geometry, with the game code trying (and failing) to move them away from the collision and taking huge amounts of CPU time to do so. Meanwhile, werkzeug3 was crashing constantly because it kept running out of virtual memory, obviously a very frustrating situation for fiver2 and giZMo. At that point, Chaos and me divided the coding work; I worked on the memory consumption in the standalone versions (and overall memory management in werkzeug3) while he was patching up the game code. I got most of the major problems worked out in the next few days, so the artists could continue working, and after that size was up. I got it down to 100k in 2 days, but after that we pretty much hit a wall; the obvious places to size optimize were done, and what was left was tons of places where I could gain maybe 20 bytes each - but a lot of that was in the game code that was still very much in flux, and doing this kind of tweaking is basically a "point of no return" where you can quite easily screw up your code beyond repair (except returning to an earlier version) - we just couldn't afford to do it with the time we had left. We ended up writing a small tool that was able to very rudimentarily parse C++ (that is, enough to determine statement and block boundaries correctly, for the code we tested it on anyway). We then used that to output modified versions of the program that tracked which functions, which branches in if statements, etc. got actually executed, and used that information to automatically remove the unused code in a second pass. It was kind of messy because you had to tag all the regions of code that shouldn't be thrown out no matter what (error handling, for instance) and make sure all of the relevant untagged code got executed. It did work well enough to gain us the 4k we needed though, quite a good result for something that took 2 days of work. This was very good news because it meant we wouldn't have to throw a lot of content out, as we had feared we'd have to. We still had a lot of work to do up to the deadline, and in fact quite a few major gameplay problems didn't get fixed until very short before the compo (which is the reason that other major problems like enemies getting stuck etc. are still in the released version). The whole thing was an incredibly exhausting and frustrating experience, and after it was done we were all basically spent and needed nearly 8 months to recover before we seriously started working on scene projects again. It certainly got us a ton of recognition (especially outside the demo scene), but to this day I'm not certain whether it was actually worth it. We all have pretty bad memories associated with the whole process - the last few months before BP04 were a pretty hefty low in all of our lives that drained us for the rest of the year. Which is the main reason we still haven't done a final version, I guess - most of the individual problems (and all of the critical ones) have been fixed now, and it would take maybe a week of getting together and working on it to make a nice final version, but we all don't really want to spend any more time on the project.

Stingray: There wasn't any planning at all, started as some kind of "can it be done?" experiment and grew from there.

WVL: I started somewhere in 2002 I think, at that moment the idea was to use an extra cartridge (Ram Expansion Unit) to help push the game along. After coding for a while I was so annoyed that the current emulators did not emulate the REU correctly that I decided to try and make a version that would work without any expansions. That was in september 2003 .. After some years things really started to work and I decided to try and make a release at BP06 Too bad it was still a preview and I didnt have time to finish it yet



http://www.bitfellas.org/e107_plugins/content/content.php?content.732

Page 4/8

Hugi: What tools have you used for the making of your game?

Archee: C++, and usual scene stuff.

KeyJ: Mainly a C compiler, but a host of other custom-made tools were used as well: All bitmap graphics and the sounds in the game use their own specially tailored compression algorithms, so I had written some compressors in Delphi, C and Python for it. The music is made possible by kb's excellent V2 synthesizer, and the final executables were compressed by kkrunchy and Crinkler.

Ryg: werkzeug3

Stingray: Mainly asm-pro and a lot of self-coded converters for the graphics. The main tool was asmpo though.

WVL: I actually use very little tools, most tools we have are custom made for this game. Some tools that we DO use are : photoshop (for the backgrounds, but we're now switching to Timanthes), 64tass (6502 assembler), a text editor (ConTEXT)-VICE (c64 emulator) and the small tools that come with it. Then there's a long list of custom made tools which I won't bother you with

Hugi: Where do the physics come into your game. Please explain why they are so important.

Archee: The 96k limit shows to the audience that these character animations are procedural, not stored. Physics is an essential element of the character AI's input environment.

KeyJ: Well, it's a billiards game, so it's effectively all about physics. However, my implementation does only the very basic stuff required to keep the balls rolling. The ball collision formula can be found in every highschool-level textbook, the amounts of friction and attenuation are obtained by experimentation (i.e. tweaking parameters until it looks somewhat realistic). That's why the game clearly states that it's an *arcade* game, not a simulation.



http://www.bitfellas.org/e107_plugins/content/content.php?content.732

Page 5/8

Ryg: Not at all, and they aren't.

Stingray: There's no physics in the game, consider that the original was released back in 1990 for a 7.09Mhz A500. Just a very simple AI, that's all. The graphics are all precalculated (this happens while the crack intro is running), of course there are no complete graphics stored inside the executable for obvious reasons.

WVL: The Physics engine is the part of the code that decides how the ball will move. There are 2 routines that are the most difficult to do fast, accurate and still fit in the memory. The most important one is the routine that checks for collisions. I finally decided on a system that treats the ball as being a single pixel large and adds the radius of the ball to the outside of the walls. That way I only have to check a single pixel while the ball is traveling. The second important routine is the one that does the calculations when the ball hits a wall. Hard to do accurately if the processor can only add and subtract

Hugi: Can you tell anything about any future 96K game you are making or considering to make?

Archee: I have time for about one or two scene releases per year. I always put an emphasis on physics and code. It might also turn out to be funny. I never invest too much time in design.

KeyJ: I'm dreaming about a remake of a certain classic game from the DOS era ... but I'm not going to tell any details right now.

Ryg: As said, nothing in the works.

Stingray: One day I might find the time to finish the Lotus sequel. Other than that, I don't have actual plans to do another 96k game, though I have some ideas, I won't reveal them here, they are just ideas, nothing more, nothing less

WVL: I'm not making any more games, but once in a while I'm talking and helping Jackasser/Instinct who is working on a port of Eye Of The Beholder! The nice thing is that he is using the original data and converting those with special tools. It already looks very cool and it's possible to crawl through the sewers already!



http://www.bitfellas.org/e107_plugins/content/content.php?content.732

Page 6/8

Hugi: What 96K games from the past did you like most and why?

Archee: Kkrieger, amazingly well generated environment.

KeyJ: As a coder, I very much prefer the (size-)coding stunts some programmers were able to pull: Things like "Pinball Dreams C64", "Lotus Esprit Turbo Challenge", "Turrican 32k" and of course "kkrieger".

Ryg: I liked Sumotori Dreams this year. Can't say much about previous years, really.

Stingray: I didn't really follow the 96k games closely but one that I find very entertaining is "sumotori dreams", truely a great 96k game

WVL: Sorry, I really don't know that many One that I do play a lot is GreenRunner on the C64.

Hugi: Please tell us about the 96K compo you won. Which party it was and how you experienced this compo and win and picking up your price...

Archee: Breakpoint 2007. It was only my second big international demoparty. It has been my dream for a long time to get a decent product on the big screen. I rarely get the chance to meet so many enthusistic coders. The prize itself did not matter as much as seeing people having a good time.

KeyJ: Originally, the game was to be released at The Ultimate Meeting 2006. Unfortunately, it was the only entry in the gamedev competition, so the compo was cancelled and I moved on to Breakpoint 2007. The competition was very stiff there, and two games heavily based on ragdoll physics won, deservedly. "Kakiarts 8-ball" got the third place, which was the result we hoped for.

Ryg: Breakpoint 2004. I was leading the compo team, but don't remember much of the compo at all, except that the tape it was



http://www.bitfellas.org/e107_plugins/content/content.php?content.732

Page 7/8

recorded on turned out to be damaged. We had to improvise and call out the game submitters to do the compo in realtime. So it was mainly stressful; I didn't notice much of the audience reaction to any entry because I was at the other end of the hall, trying to make sure the compo worked out. And directly after it was finished, I had to move back to the recording area and continue working, because we'd had a lot of equipment problems and were way behind schedule. As for the prizegiving, I was next to the stage and handing out the prizes to Steeler and Scamp. I don't even know if I went up to the stage when the winner was announced; in any case, it was zero surprise, because I had prepared the results file, prizemoney envelopes and Steeler's cue cards earlier and already knew we had won what (you'd have to ask one of the other ones involved, basically).

Stingray: I made the game for the 96k game compo at Breakpoint 2005. It was nice to see it on the bigscreen, even though there was a bit of stress beforehand (one of the cia's in the compo amiga was broken/not working properly so the keyboard routine didn't work etc). All in all it was a great experience and winning the compo felt good

WVL: It was the 96K game compo during Breakpoint 2006! Very cool to see it running on the bigscreen. Actually it was a compo without much competition, during BP07 the competition was a lot harder! (good thing for me) Picking up a prize is always nice, and I could even thank DeeKay for his nice balls.

Hugi: There have been many nice 96K games in the past. Where will this competition go to in the future?

Archee: People never run out of ideas. Even small products can say a lot.

KeyJ: I personally don't see any real evolution in 96k games currently. .kkrieger is still by far the game with the highest "wow" factor. But again, there hasn't been much progress in 64k games either. The special thing about 96k games isn't graphics and sound anyway, it's always been about fun new ideas (see "Sumotori Dreams") or sizecoded re-implementations of other games.

Ryg: We'll find out soon enough.



http://www.bitfellas.org/e107_plugins/content/content.php?content.732

Page 8/8

Stingray: Hard to say, I bet we'll see some pretty outstanding games (like Pinball Dreams on C64) and a lot of average/bad games as well. Just like it always was. I don't think it will change in the near future.

WVL: There's no real 96K game compos on the C64.. But we do have 1K/2K etc game compos every year. Some of them are really very neat and they're becoming better every year I still want to make a game for the crapgame compo as well

Hugi: Thank you very much Archee, Ryg, KeyJ, Stingray and WVL for discussing the 96k game subject with us! Perhaps it's wishful thinking but Hugi does hope seeing a new 96k game from all of you in the future. Have a nice day!

Links related to this article

Sumotori Dreams

Kkrieger

Lotus Turbo Challenge

Magic of Nah-Kolor