

Chuck Sommerville about C64 Summer Games.

What do you generally remember about the process of porting the game to the Apple 2?

"We started with the C64 code, which was still under development. Kevin, the other programmer, and I each took some of the events, I also took the MCP, which was the part that dispatches to the individual games. Periodically we would get code updates, and had to integrate the changes into our code. I wrote a high speed sprite simulator and we converted the art to 7 copies of each so we could avoid shifting on the fly. Before each sprite was masked into the screen, a copy of the screen background was saved to restore later. Since the display was double buffered, which means we page flipped, we had to save a copy for each screen."

Scott Nelson already told me that you wrote almost everything alone?

"I worked with another programmer named Kevin McClard, I think."

Was this port the first thing you did when joining Epyx?

"Yes."

Are there any specialties in the Apple 2 version?

"You have probably already heard about this, but if you booted from the flip side on the disk, the graphics were rendered upside down, as a joke."

How would you compare your conversion to the original C64 one?

"I think overall it was fine, It ran a little slow, because it was missing the hardware sprite and sound support that the C64 had. The C64 was really a much better game machine than the Apple II."

Was it fun or just work to port somebody else's brainchild? Did you like the game?

"This was just to fill out the SKUs for Epyx to get an Apple version. The game was OK, but not really the type of game I play."

Any other stories or trivia you can tell about the Apple version?

"This was the first time I had ever had to write the disk boot code for a game as well. I created what I call a "bare bones" DOS. I used only the RWTS, (read/write track sector) code that Apple puts on track zero. Somehow the disk duplicator replaced the standard track zero with their own copy protected version. Using just the RWTS, I was able to make a system that would read sections of the disk and execute them. The files were just numbered, and the track/sector starting position, and file lengths for all of the files fit into one 256 byte table which took the place of the entire directory. As far as I could tell, Epyx had never had an apple product that just booted and ran in this manner. All the other publishers seemed to have solved this problem years before. My "Bare Bones" DOS continued to be used for a couple years after that on other Epyx products."

The hardest event to convert was Gymnastics, because the sprite data was compressed on the C64 version, and decompressed on demand. I had to make a decompressor that would write the data in a format that could be rendered with my sprite engine."

Thank you for the interview!

Peter Engelbrite about Atari 2600 VCS Summer Games.

In response to my mail asking for an interview, Peter already provided a heap of information:

"The VCS versions of Summer and Winter games were done as part of the same project. The VCS had died completely, but then the cost of the base unit dramatically dropped in price, creating a window of opportunity for cartridges. Programming the VCS is unlike programming for any other game system. There is very nearly nothing there to work with. Your game must fit in a 4K ROM space, you have a total of 128 bytes of ram, and there is no display screen. You get 3 8-bit shift registers, 3 1-bit registers, and 20 bits of background. You have to jam data into these registers on the fly as the TV electron beam is scanning down the screen! This means that the graphics you see are not drawn, but algorithmically generated. The section of code that does the display is called the kernel. In many kernels, you have to program in such a way that it takes exactly the same amount of time, regardless of the conditional operations. The ROMs had 16K, in 4 banks. To switch the bank, you read from a specific address. I had a tough bug, where it kept switching banks for no reason. It turned out that a certain operation caused a spurious read, flipping the bank. Craig Nelson was the project leader, and I really enjoyed working for him. One of the programmers was the same person who programmed the VCS Pacman, and another was in the process of starting up a supercomputer company. It was one of the first projects after Epyx decided to stop giving the programmers any royalties, so while previous programmers got substantial royalties on the VCS version due to their contracts for the original versions (on C64, etc.), we got our salaries and nothing more. I think that this pretty much represents the point (and part of the reason) where Epyx started going down (sigh). Epyx did one other VCS game: California Games. It was developed in 1-1/2 months by another programmer, a musician, and me!"

I then started to interview him and my first question was: What aspects made you choose the events the were done for the VCS version? Did you grab the one's which were thought to provide most fun, or were the most doable chosen?

"I think the choice of events was a combination of fun and practicality (mostly practicality). Keep in mind we were often putting multiple events into a 4K area. This is 4K of text:

[illegible]

0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef
0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef
0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef
0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef
0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef
0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef
0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef
0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef
0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef

Now, imagine fitting both ski jump and hot dog skiing into this space. It's amazing that it can do anything. In BMX bicycle, I created a bicyclist in 8 pixels by 10 pixels by 1 color, and you could see his head nod, the pedals go around, and do tricks. I think VCS programming is some of the most fun, hardest, and most microscopic programming there is, and I've worked with microcontrollers (such as the Fairchild F8)."

So all in all - was Epyx' VCS experiment a succesfull one?

"Although the finances of Epyx were not shared with the employees, it was my understanding that the VCS games were quite profitable."

In the mid 80's a lot of professional VCS coders joind Epyx when they merged with 'Arcadia'. Did some of Epyx' VCS knowledge come from there? Where these people still in the company when 'Summer Games' and the others were coded?

"Craig Nelson (and his brother) were the primary sources of VCS experience at that time."

There's many rumours in the internet about unreleased VCS titles from Epyx, especially 'Super Cycle' is rumoured to exist. Do you know more about this issue?

"I never heard of a Super Cycle for the VCS at Epyx. There is some possibility that an outside developer worked on one."

How did the original 'Summer Games' coders react/comment on your ports? Did you get any feedback from them?

"The original programmers (the term "coder" would refer to someone with no creativity) seemed pleased. Of course, if you can get a VCS to do anything, you're a hero."

Who was responsible for the PAL conversion of 'Summer Games'? I noticed some color problems here, for instance half the objects in the 'Skeet Shooting' are almost invisible?

"The PAL versions were done internally, but I don't remember who was responsible."

Weren't you worried about people destroying their joysticks when they played 'Summer Games'?

"We called those kind of games "joystick busters", when there was repeated, rapid clicking. Epyx did come out with it's own joystick at one time (I don't think there's a connection, though)."

Are there any Easter eggs in Summer Games?

"There were no easter eggs in the VCS versions: we were fighting for every byte. We would look for subroutine calls towards the end of a routine so that we replace the call/ ret to a jump (it is logically equivalent). In the surfing event in California Games, the random colors of the wave are generated by a pointer scrolling through the code. We would try to find ways for separate events to share code. The Games, Summer edition had lots of Easter Eggs."

Thank you for the interview!

Randy Glover about C64 Summer Games.

I already know you worked on the swimming events of Summer Games, did you work on other parts of the game too?

"Swimming is the only section I worked on."

Can you tell me more about the creation of this event? Who had the idea for it and how did it develop?

"All events were discussed but many were not included. The problem to overcome with swimming was the control. I do not recall if I came up with the control ideas or if I got swimming and then had to come up with the controls. Either way, I think the control system worked pretty well."

Was it difficult to program? I'm asking this, because Chuck Sommerville already told me that the swimming event was hardest to port to the Apple 2, since you were using packed data for the sprites?

"Somewhat difficult. I can't speak to all the details but as a company not Summer Games was the first Epyx program where a "graphic artist" was used. Until then it was all up to the programmers. I imagine the packed sprites came out of that. (as well as space)"

Did you work on the Atari 800 version too? Who programmed that?

"Only the C-64. I have no idea on the Atari."

Most internet sources credit you as musician for Summer Games. So you did all the music in the game, inclusive all the national anthems?

"Well you better e-mail them al and tell them they are wrong. I did not work on any of the Summer Games sound or music tracks."

If no, who did the music then?

"Unknown."

Any additional stories or trivia you can tell about "Summer Games"?

"It was a pretty big in-house project and one of the first to have multiple programmers working on the same project (partially due to time frames). Although I did not play the game much I think it came out pretty well."

Thank you for the interview!

Scott Nelson about C64 Summer Games.

How much of "Sweat!" actually went into "Summer Games"? How did "Summer Games" evolve?

"Actually, there's probably more of Party Mix in Summer Games than Sweat. Sweat never got much beyond the concept stage. When Starpath and Epyx merged, we needed a C64 game quickly. That meant a project that 5 or 6 people could work on at the same time, and still actually reduce the time to market. Decathlon was our first choice, because it seemed like it would be easy to break up into discrete tasks that each programmer could work on without needing to interact. Black Box programming at it's finest. Although we liked idea of a decathlon, we /didn't/ like some the events. In particular, we didn't see any way to make the marathon quick, and still feel like a marathon. So we scraped the idea of using any particular events, and instead went with a freer format. We just brain stormed some ideas for events, wrote up a basic outline of how each would work, and a programmer worked on one.

I was in charge of the MCP (master control program) - the thing that loads all the events, keeps track of the high scores, name entry. I also was working on the assembler and some other tools."

Both the Atari 800 & the C64 version seem to be programmed by the same team, is that right?

"No. The Atari version was contracted out. They worked from the original design and source, and so the original programmers got credit."

Did you work on any other port of "Summer Games"?

"No, at least, not directly. Chuck Sommerville did basically the entire Apple II port, and the IBM-PC port came much later, since we had to wait for the PC to exist first."

What was the major difference between working on the Atari and the C64 version?

"Uh... For the Atari version, I didn't have any source code to work from, which made changes difficult. I did fix a bug in the option keys read routine, but that's about all."

Where there any ideas/disciplines that didn't made it in the final product? (Was Summer Games 2 based on these? :-))

"Yes, we had a big list of possible events, and picked the ones we thought would be easiest to do. Some of the events that looked hard ended up in Summer Games II."

Any additional stories or trivia you can tell about "Summer Games?"

"While working on Summer Games, Epyx was approached by a design house that was trying to push their sprite editor. Their tool used big sprites with double sized pixels, four color pixels, which we had already rejected as too ugly. Summer Games used the high-res monochrome sprites overlaid to get multiple colors, with special interrupt technology to get more per screen. A lot more work, but we thought the end result was worth it. The tool designers basically said that we were foolish to shoot for something so difficult, and that rapid development was incredibly important, and when the next CES came around, "we'd be sorry."

At the CES where Summer Games was released, HES software was previewing what was later released as HES games, which obviously used the sprite editor we had already rejected. One of the Mucky-mucks from HES brought his team by our booth, and berated them for not having graphics that looked as good as ours. The first thought that came to my mind was 'I am so sorry for those guys'."

Which other Epyx games did work for besides Summer Games?

"My real claim to fame is writing and designing the fastload cartridge. I also developed the Vorpak disk technology, which almost every Epyx C64 game used. The development of Vorpak was kind of strange - we were working on the Barbie game, and it required speech. There were basically two choices - we could make a super fast disk loader, that read speech tracks from the disk, or use some sort of speech compression. The C64 only has a 4 bit DAC, and I believed that even the best possible recording would still sound unacceptable, so I pushed for the loader. In the end, we did both - Barbie used the disk loader, and 4 bit PCM recording, and Impossible Mission used speech compression."

Thank you for the interview!