OPQA Vs. QAOP

The Final Battle

Making of

Original idea

Earlier this year, I planned to make a very different game for the CPCRetroDev. It was about a small robot that had to escape from a factory overcoming screens with different obstacles. I wanted to do it in mode 1 with smooth animations at 50Hz and after a week I realized that it was quite ambitious for the knowledge I had about the CPC at the time, so I ended up leaving it in the drawer.

At the end of summer I tried to retake the project again, but I decided it was a good time to think another idea that was simpler and feasible in the time I had until the deadline. As I have always find amusing the (absurd) debate between OPQA and QAOP, I thought it was a perfect time to use it in a game, and since I like puzzle games, doing something similar to Puyo Puyo was interesting and viable.

Prototype

Once the new game idea was clear, I developed a simple prototype. I worked with the random generation of the blocks at first, then I made them to fall on the board, and perhaps the most complicated thing to develop was to check the combinations of pieces of the same color. As it’s not necessary that they were in line, I had to write an optimized recursive function (and trying to not abuse the stack) in which I would check the four directions until there were no more pieces of the same color.

After having this functionality implemented the game started to take shape. Although if I had stayed there, the game for one player would have been somewhat boring...

AI

Another fundamental pillar of the game was to create an artificial intelligence smart enough for each game to be a challenge, but not too much so the player wouldn't feel an unfair defeat.

At the end, I developed a simple state machine where the computer finds the best position to leave the current block (aiming to make columns of the same colour), then the block is moved and then it finally drops it in place. For each stage there are some parameters changed in order to make the game progressively more difficult.

After designing the AI, I realized that I could assign it to both players, so I created a simple demo mode that automatically appears when the menu music ends.

Music and game modes

To make the game more enjoyable, I created some music themes for the menu, stage clearing,, etc. I didn't create one for the main game because I play sound effects in stereo (using channels A and C), and I also didn't want an annoying looped music playing in the background.

I had never composed anything before, but taking a look at the sample themes and some Amiga MODs I could see some interesting patterns and I could compose the music of the game in less than a week.

As for the game modes, from the beginning I wanted to include two main modes: one player against the machine and two players facing each other. As I made the game speed up as the player made combos, I also included an endless mode in order to see players' high scores.

Later, I played the original versions of Puyo Puyo (for Famicom Disk System and MSX2) and I saw that they included an intriguing mission mode. I took inspiration to make some missions, although later on I was thinking of others based on score so the player should take advantage of some techniques to make more points.

Optimizations and final touches

After inserting the graphics of character avatars, different types of pieces, the background, musics, game mechanics,... I was running out of memory. The first optimizations consisted of cleaning the code and refactoring some functions.

Later, I compressed the tile map of the background and it went down from occupying a thousand bytes to about eighty.

To do the compression easily, I used version 1.5 of CPCtelera, which at the time of writing these lines requires a checkout of the development branch of the repository. As additional advantages, it was easier to create a CDT with a custom loading screen, and the creation of SNA files in each compilation made testing easier and faster.

A couple of tips for aspiring CPC programmers: do not underestimate the power of this computer and don't pretend to make a complex game as your first project.