

Cannon Duels

Making Of

David Asta - 2018

Motivation

I got my first computer when I was about 10. It was an Amstrad CPC464. Initially I started playing the games that came with the computer. Oh Mummy was the first one. Later, my father got me some books and magazines with type-in games and a book for learning BASIC. From the moment I opened that book my life changed forever.

Making a game was since then a dream. As a kid I did different simple programs in BASIC, but I never really understood how sprites, gate array, sound chip, etc. worked, so my dream never came true.

Not long ago I dusted off my Amstrad CPC6128 that I still keep since I was a kid and started replaying games. Some months later I discovered CPCtelera and I decided to make a game and make my childhood dream become true.

My girlfriend and I love to play Bomberman together, therefore I decided to make a game that can be played by two players, so we could play it together.

Gameplay

The game is quite simple. Two Cannons are located at each side of the screen and can shoot at each other. At the centre of the screen, between the Cannons, there is a group of bricks. Cannon Balls will destroy the first brick they encounter in their path or subtract a live from the opposite player if hitting the Cannon.

When hitting a Brick there is chance to get a Power Up. Each one is activated by random and with different properties.

When a Cannon is hit by a Cannon Ball, it loses a live. Reach 0 and the game is over.

A player gets an Extra Live every time collects 464 points.

Starting

When developing software it is very easy to fall in the trap of adding more and more functionality as the time goes, and never really finish anything. It is a must to have a scope and a deadline.

Hence, first of all I defined the scope of the project; What type of game I want to do? What is the minimum I need to have something I can call it a finished product?

And finally, and very important, I defined my deadline.

From this I decided: by early September I will have a game capable of showing different levels/maps, with a loading screen, a menu with music and a game over screen. The game will be playable by either one or two players, so I will need some sort of AI. A simple AI that just moves the Cannons in a random way and shoots at random times will suffice.

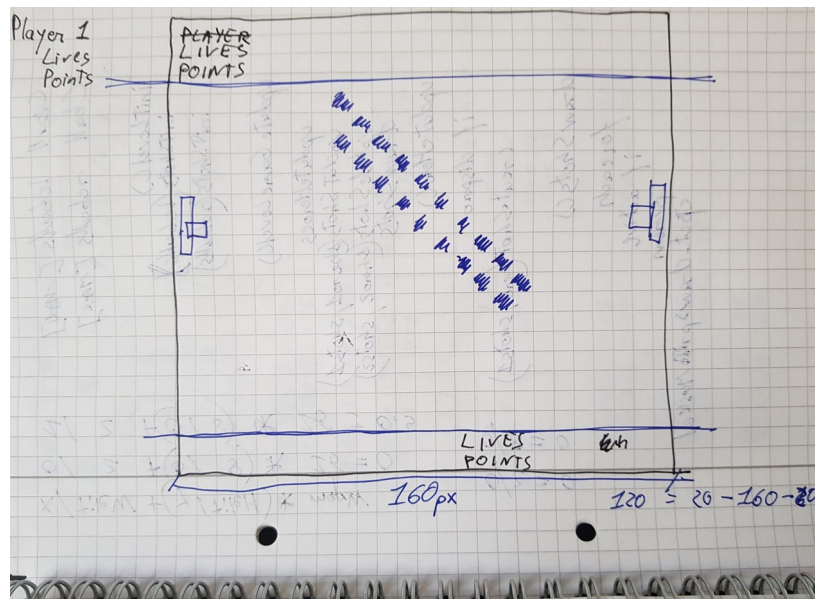
I started coding, watching CPCtelera tutorials and reading manuals. By the first week of September I had my final product with everything I defined. Was then when I decided to send this game to the #CPCRetroDev 2018. I would like to see the popularity of the game (if any) and I hope to receive feedback for future games.

As I still had time until the deadline for the #CPCRetroDev, I decided on other features I wanted to implement; Power Ups, sound effects during the game play, better AI (maybe even multiple ones) and more maps for levels.

I started with the Power Ups, followed by the better AI and a quick way of implementing maps with a spreadsheet that translates to C arrays. Sound effects proved to be a bit difficult to get it working, but finally I got it.

Step by Step

First of all I did a design on paper of how the game screen would look.



After installing CPCtelera, I started creating the sprites for the Cannons, Cannon Balls and Bricks. First executable contained an almost empty screen with one Cannon at each side that could shoot Cannon Balls, and some indestructible bricks at the centre of the screen.

From there I moved into calculating which Brick has been hit by a Cannon Ball. I am not happy at all on how I did it, but I have not figured out another way. Well, it is not pretty but it works.

Next came the scoreboards for showing points and lives. I added the logic of cannons being hit by a Cannon Ball and of decreasing number of lives, as well as accumulating points when a Cannon Ball hits a Brick.

Though I only had one brick map designed, I developed a way of changing the brick map after all Bricks were destroyed. And created a simple brick map to test this. So at this point I had two different brick maps.

AI came at this point. A very dumb AI (more like an A without I), just getting a random number and depending of which number go up, down or shoot. Dumb, but enough for testing.

Next I wrote the Game Menu, so I could select 1 or 2 players and thus test the AI.

I re-used the logo I created for the Game Menu and I created a loading screen. Once I tested I could load it before the game with a small program written in BASIC, I created a small shell script to use *iDSK* to add the loader, the loading screen and the game to a DSK file. Was not able to make the same for the tape file using *2cdt* and I was spending too many hours, so I parked this for a while.

I added a simple Game Over screen just saying "Game Over" and which player was the winner. The screen was shown as soon as any of the players lost all lives.

Next I did the music for the Game Menu with ArkosTracker.

And with this I had covered the initial scope I defined!

Then I saw the announcement for the #CPC RetroDev 2018 and thought, why not? Maybe I can get useful feedback about my game. Which parts people like and which parts do not.

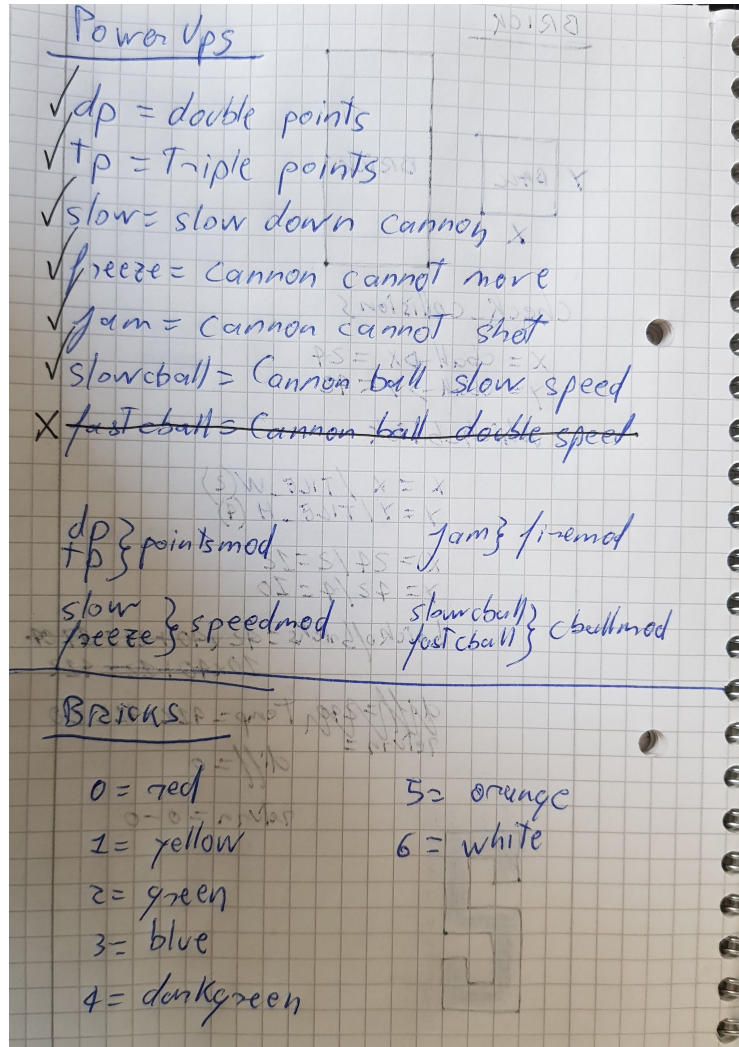
So I made a list of improvements and I prioritised them to take them in order.

First I went for creating more brick maps. I made a spreadsheet to easily create more brick maps. I can design the position and colours for the Bricks and the spreadsheet translates it into a C array that I can use in my code.



With the dumb AI I had at that moment I was not going to get much fun, so I wrote two more algorithms and decided to use them depending on the level. So as the player progresses, the AI becomes more challenging.

To give a more interesting play I decided to add Power Ups.



By hitting Bricks a player can activate Power Ups. Some give extra points, some annoy the opponent by reducing the speed, freezing or even jamming the Cannon.

At this point I showed the game to my girlfriend and we started playing it on my Amstrad CPC6128. From this time of play testing came some ideas (position of the scoreboard, colours of some bricks, logo). My girlfriend gave me the idea for the lives' counter to flick when the player loses a live.

Last days of development I spent creating the last improvements we gathered from our fun sessions of testing.

Challenges

I have never done a game before, but more importantly I have never programmed in C, so first I had to get to know the language. I am familiar with other imperative languages like COBOL and Pascal, so I just needed to get a clear idea of how header files work, how to pass parameters between functions and how pointers work. *The C Programming Language* book by Brian W. Kernighan and Dennis M. Ritchie was an excellent read.

Personally, my challenges were to learn how the Amstrad works internally (at chip level) and get used to the different CPCtelera tools. But the biggest challenge by far was creating graphics. I have zero skills for drawing. For that reason I decided to create a game that did not need amazing graphics. After all, it is my first game ever.

The YouTube videos from Fran Gallego giving classes at the university have been a gold mine for gathering knowledge about CPCtelera and the Amstrad CPC itself.

Doing the music was fine. I had some experience with trackers in machines like Atari ST and Commodore Amiga, so at least I did not have to learn how trackers work.

Technologies

All code was written using CPCtelera v1.4.2 libraries, using Visual Studio Code.

For testing during coding phase I used WinAPE 2.0 Beta 2 until I reached the final phase testing, which I did on my Amstrad CPC6128.

Sprites were created with Retro Game Asset Studio v1.2.5.0

ArkosTracker v1.0 was used for the music and audio effects.

For the game logo and tiles I used GIMP v2.8.22 and then converted the graphics to C arrays with the macro IMG2SPRITES from CPCtelera.

Background tile map and brick maps were created with a spreadsheet I made myself which translates to C arrays.

For creating the disc file (DSK) I used iDSK from CPCtelera, and for the tape file (CDT) I used 2cdt also from CPCtelera.

The GameManual.pdf and this MakingOf.pdf were created with LibreOffice Writer v5.4.1.2

The gamplay.mp4 and the trailer.mp4 were created with OpenShot Video Editor v2.4.2

Conclusion

All in all an amazing experience of creativity, discovery and experimentation. I have learned a lot C and how the Amstrad CPC works internally. After more than 30 years playing games, now I can play a game that I made myself. My dream has finally come true!