

SORCERERS

MAKING OF..

Whenever I start a new project it is to test new features of the tools to be used (in this case CPCtelera 1.5), or different game modes.

This time I wanted to make a game for two players at the same time, and dispense with the double video buffer in order to have as much memory as possible.

It was also time to use the data compression offered by CPCtelera 1.5 with the "zx7b" and that it is especially useful for reducing the map size and graphic resources.

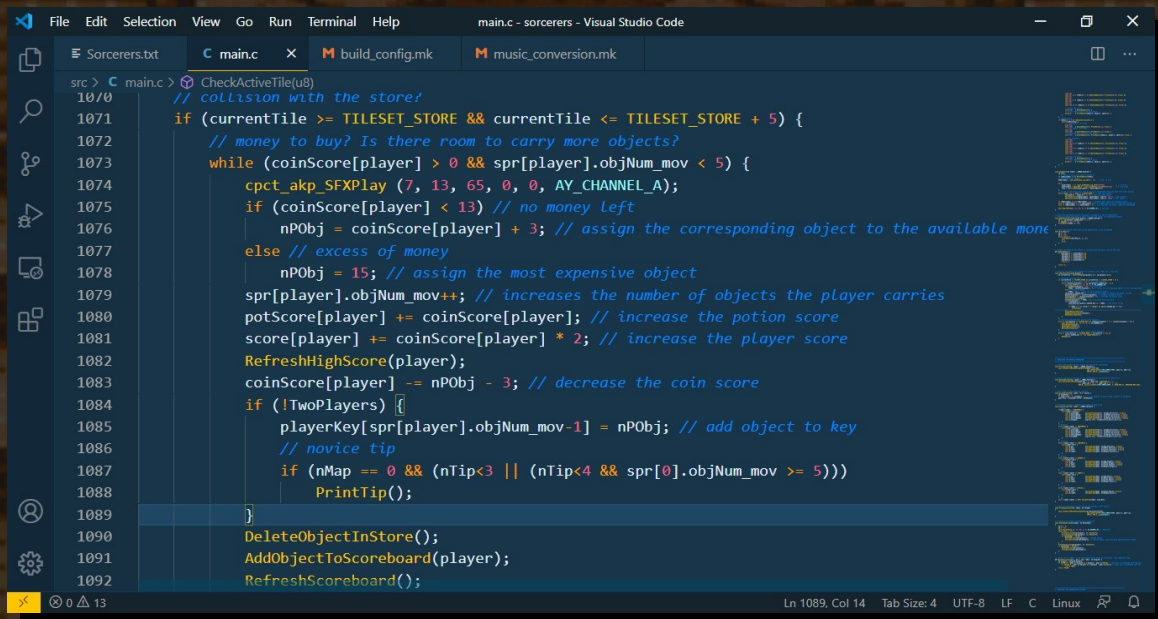
This attempt to have as much memory as possible after writing the game logic was necessary to be able to create an extensive map, but so much memory was freed that unexpectedly it became possible to incorporate two very different game modes into the same tape load, for 1 and 2 players. And so it was done even at the cost of a map not so great, although enough to go developing a progressive difficulty and dropping less skilled players before arriving to the final screen.

In addition, I have allowed myself to create different sprites for each direction of movement. The normal thing is to do just one and "mirror" it, but although it requires less memory it has the disadvantage that you can't apply highlights and shadows to characters because they wouldn't be consistent.

Regarding the theme, the story, the graphics... they emerged sporadically in the first moments.

As for the sound section, I had wanted to work with Mr. McKlain for a long time. Genius and figure of the musical notes for the CPC. A few years ago we met in Malaga on a RetroPixel and we agreed work together on a project. McKlain has also complemented the work of Atila (Betatester and official counselor of the kingdom), testing the game and proposing improvements that have finally been implemented.

The language used has been mostly "C", with some small third-party functions in assembler. Compiled with SDCC 3.6.8, the compiler included in CPCtelera 1.5.



```
src > C main.c > CheckActiveTile(U8)
1070 // collision with the store?
1071 if (currentTile >= TILESET_STORE && currentTile <= TILESET_STORE + 5) {
1072     // money to buy? Is there room to carry more objects?
1073     while (coinScore[player] > 0 && spr[player].objNum_mov < 5) {
1074         cpct_akp_SFXPlay (7, 13, 65, 0, 0, AY_CHANNEL_A);
1075         if (coinScore[player] < 13) // no money left
1076             nPObj = coinScore[player] + 3; // assign the corresponding object to the available money
1077         else // excess of money
1078             nPObj = 15; // assign the most expensive object
1079         spr[player].objNum_mov++; // increases the number of objects the player carries
1080         potScore[player] += coinScore[player]; // increase the potion score
1081         score[player] += coinScore[player] * 2; // increase the player score
1082         RefreshHighScore(player);
1083         coinScore[player] -= nPObj - 3; // decrease the coin score
1084         if (!TwoPlayers) {
1085             playerKey[spr[player].objNum_mov-1] = nPObj; // add object to key
1086             // novice tip
1087             if (nMap == 0 && (nTip<3 || (nTip<4 && spr[0].objNum_mov >= 5)))
1088                 PrintTip();
1089         }
1090         DeleteObjectInStore();
1091         AddObjectToScoreboard(player);
1092         RefreshScoreboard();

```

Once the project has been completed, I can say that the most complex or laborious points have been:

- Obtain the proper order of execution of all the game logic in the main loop and place the vertical retrace so that the dreaded flickering of sprites does not appear. Although it has not been possible to eliminate completely in some screens very charged, it can be said that the objective has been met, and there is hardly any flickering in general.

- The creation of graphics with such low resolution. It's an odyssey designing 8x8 pixel "ingredients" that really look like what you want to represent. The same happens with the main sprites, 10x12 pixels. There is an anecdote related to this; In the preliminary versions the characters wore a Gandalf-type peaked hat, but to get to the level of graphic detail that was intended with the main sprites, the hat had to be "sacrificed", wearing both protagonists in their definitive version a shiny bald head.



It was not a game designed to compete in the CPC Retrodev, but it started many months ago (according to my notes, on August 3, 2019), and has been progressing very slowly taking advantage of vacations or some free weekends. The coincidence in time and the fact that it was developed for the Amstrad CPC 464 is what has encouraged us to give it one last push to present it to competition.

*This text has been automatically translated. We apologize for the inconvenience.

SORCERERS

COMO SE HIZO...

Siempre que empiezo un nuevo proyecto es para probar nuevas características de las herramientas a utilizar (en este caso CPCtelera 1.5), o modalidades distintas de juego.

En esta ocasión quería hacer un juego para dos jugadores en una misma partida, y prescindir del doble buffer de video para disponer de la mayor cantidad posible de memoria.

También era el momento de usar la compresión de datos que ofrece CPCtelera 1.5 con el algoritmo "zx7b" y que es especialmente útil para reducir el tamaño del mapeado y los recursos gráficos.

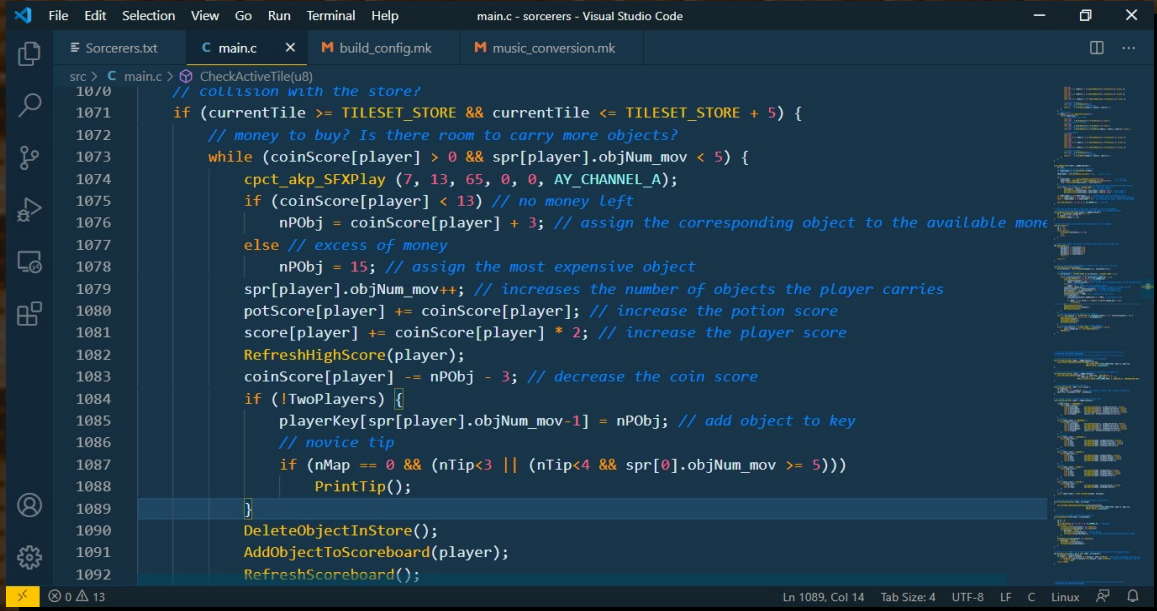
Este intento de disponer del máximo de memoria posible tras escribir la lógica del juego era necesario para poder crear un mapeado extenso, pero fue liberada tanta memoria que inesperadamente se hacía posible incorporar en una misma carga desde cinta dos modalidades de juego muy distintas, para 1 y 2 jugadores. Y así se hizo aun a costa de un mapeado no tan grande, aunque suficiente para ir desarrollando una dificultad progresiva y haciendo caer a los jugadores menos hábiles antes de llegar a la pantalla final.

Además me he permitido el lujo de crear sprites diferentes para cada sentido del movimiento. Lo normal es hacer uno solo y "espejarlo", pero aunque requiere menos memoria tiene el inconveniente de que no puedes aplicar brillos y sombras a los personajes porque no serían coherentes.

Respecto a la temática, la historia, los gráficos... fueron surgiendo de manera esporádica en los primeros momentos.

En cuanto al apartado sonoro, llevaba tiempo queriendo trabajar con el Sr. McKlain. Genio y figura de las notas musicales para el CPC. Hace ya unos años coincidimos en Málaga en una RetroPixel y acordamos trabajar juntos en algún proyecto, que al final ha resultado ser este. McKlain además ha complementado la labor de Atila (Betatester y consejero oficial del reino), testeando el juego y proponiendo mejoras que finalmente han sido implementadas.

El lenguaje utilizado ha sido mayoritariamente "C", con algunas pequeñas funciones de terceros en ensamblador. Compilado con SDCC 3.6.8, el compilador incluido en CPCtelera 1.5.



```
1070 // collision with the store?
1071 if (currentTile >= TILESET_STORE && currentTile <= TILESET_STORE + 5) {
1072     // money to buy? Is there room to carry more objects?
1073     while (coinScore[player] > 0 && spr[player].objNum_mov < 5) {
1074         cpct_akp_SFXPlay (7, 13, 65, 0, 0, AY_CHANNEL_A);
1075         if (coinScore[player] < 13) // no money left
1076             nPObj = coinScore[player] + 3; // assign the corresponding object to the available money
1077         else // excess of money
1078             nPObj = 15; // assign the most expensive object
1079         spr[player].objNum_mov++; // increases the number of objects the player carries
1080         potScore[player] += coinScore[player]; // increase the potion score
1081         score[player] += coinScore[player] * 2; // increase the player score
1082         RefreshHighScore(player);
1083         coinScore[player] -= nPObj - 3; // decrease the coin score
1084         if (!TwoPlayers) {
1085             playerKey[spr[player].objNum_mov-1] = nPObj; // add object to key
1086             // novice tip
1087             if (nMap == 0 && (nTip<3 || (nTip<4 && spr[0].objNum_mov >= 5)))
1088                 PrintTip();
1089         }
1090         DeleteObjectInStore();
1091         AddObjectToScoreboard(player);
1092         RefreshScoreboard();

```

Una vez finalizado el proyecto, puedo decir que los puntos más complejos o laboriosos han sido:

- Obtener el orden adecuado de ejecución de toda la lógica de juego en el bucle principal y situar las esperas del retrasado vertical para que no apareciese el tan temido parpadeo de sprites al no disponer del segundo buffer de video. Aunque no se ha conseguido eliminar del todo en algunas pantallas muy cargadas, se puede decir que se ha cumplido el objetivo, y apenas se aprecia parpadeo en general.

- La creación de gráficos con tan baja resolución. Es una odisea diseñar "ingredientes" de 8x8 píxeles que se parezcan realmente a lo que quieres representar. Igual ocurre con los sprites principales, de 10x12 píxeles. Hay una anécdota relacionada con esto; En las versiones preliminares los personajes llevaban un sombrero picudo tipo Gandalf, pero para llegar al nivel de detalle gráfico que se pretendía con los sprites principales, el sombrero tuvo que ser "sacrificado", luciendo ambos protagonistas en su versión definitiva una brillante calva.



No ha sido un juego pensado para competir en la CPC Retrodev, sino que se inició hace ya muchos meses (según mis notas, el 3 de agosto de 2019), y ha ido avanzando muy poco a poco aprovechando vacaciones o algunos fines de semana libre. La coincidencia en el tiempo y el hecho de haber sido desarrollado para el Amstrad CPC 464 es lo que nos ha animado a darle un último empujoncito para presentarlo a concurso.

