

SHINORI

MAKING OFF

```
        sub e_x(iy)      ; A = (obs.w + obs.x) - player.x
        ld a, #0         ; A = 0
        jr c, no_colisiona_en_x_enemigo ; Si da <0, no colisiona

        ; Se cumple la 1º, ahora comprobamos la segunda
        ; (player.x + player.w) - obs.x >= 0
        ld a, e_x(ix)    ; A = player.x
        add e_w(ix)      ; A = player.x + player.w
        sub e_x(iy)      ; A = (player.w + player.x) - obs.x
        jr c, no_colisiona_en_x_enemigo ; Si da <0, no colisiona

        ; Por ultimo si colisiona en X metemos un 1 en A y devolvemos
        ld a, #1
        ret

no_colisiona_en_x_enemigo:
    ld a, #0 ; Devolvemos un 0 si no colisiona
    ret
;=====
```

comprobar_colisiones_player_obstacles:

```
    ; Ponemos que no esta colisionando por debajo a nada
    ld a, #0
    (colisiona_abajo), a
```

```
    entityman_getEntityArray_IX
    collisionman_getObstaclesNivel_IV
```

```
    entityman_getNumObstaclesNivel_A ; A = num_obstacles
    ld a, a ; numero_obs = num_obstacles
```

; En IX el primer elemento de los

introduction

how *Shinobu* was made

When we start to study Multimedia Engineering degree, we knew that the way wasn't going to be easy. But not for that we were going to quit.

This degree caught our attention. Some for the creative part, which merges with computer science through design, modeling and post production programs. Others for knowing the logic and IA of the machine, games and programs.

Infinity Entertainment, a group of fourth-year students. Rubén Rubio, Jesús Más and Judith Mula present Shinobu, a game created to participate in the #CPCRetroDev after a few weeks of hard work. Shinobu.

first days

In the beginning, anyone knows what was going to come down on us. At the end of the last course, Fran Gallego told us that during the summer we should take a course made by him, to learn basic concepts of machine code and Z80 assembler.

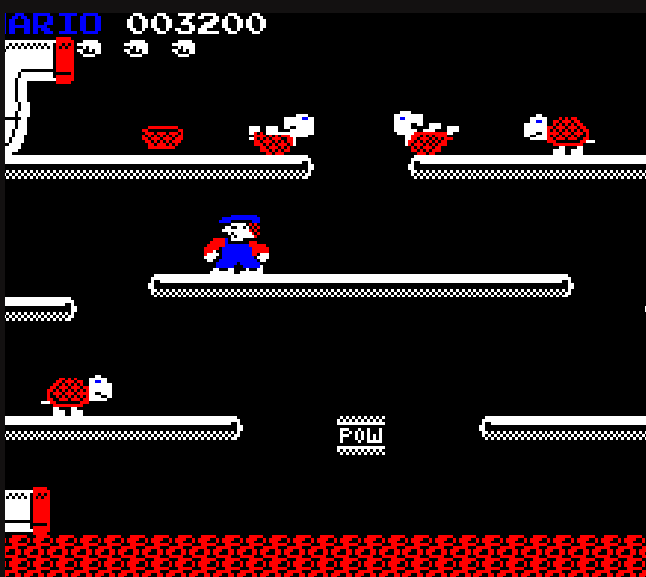


Students of higher years had already warned us that we had to make an assembly game at the beginning of the course. But we didn't imagine the challenge this meant. We can safely say that most of our classmates and we have had a lot of pressure.

Then, on September 10, the first day of Videogames 1, Fran Gallego appeared with the planning. The game had to be delivered on October 30. We are not going to lie, with this planning we don't need 5 minutes to get nervous.

Although we had a week to continue learning Z80 assembler, in the beginning we discovered why Fran gave so much importance to start learning this language in summer.

We dedicate the first week especially to continue learning. Meanwhile, we think of ideas for the game. There was only one condition. The game was to work in Amstrad CPC 464.



After thinking for a few days, the idea came up. We thought a lot about the typical platform shooter. However, we didn't want to kill enemies, we wanted to have a goal. Then, we describe that someone had made a Super Mario for Amstrad. It is not official, but we loved that this game existed for this machine. However, we wanted the game to be harder to win. Our screens weren't going to be side, we wanted to play against gravity. We continued with the idea of the shooter, so our character was going to be able to shoot to kill enemies.

With this idea, then, we begin to develop. But starting with an idea does not mean that it could not change over time. What was the result?

description of the game



Everything was going to be based on the story of a ninja who needed to get a jewel. We want that you should pass between levels going up, so we thought of a tower. Everything was linked.

A ninja had to go to a tower guarded by guardians. He wants to get what his family had been looking for years. But, nobody could get out of there alive. How complicated was it?



Let's talk about the game. **This ninja can do three things:**

- He can move sideways.
- He can jump to change the platform and dodge the traps.
- He can also throw shuriken so he can kill enemies. But beware! Only 3 by screen.

We established 3 lives so you had several opportunities to pass the tower. This is divided into 6 floors or levels. To pass between them, you must reach the top of the screen and jump. Also, you have to try to avoid getting close to the traps.

- Traps hurt you, so you lose a life.
- The traps are delighted. They make you return to the first level.

Also, you should not approach the guardians. They can't harm you, but they push you. Maybe just so you don't pass. Or maybe to throw you into a trap. **Easy? Hard? Well, one of us still hasn't been able to pass the game.**

technologies

```
Welcome to...

This setup script will help you configure CPCtelera in your system.

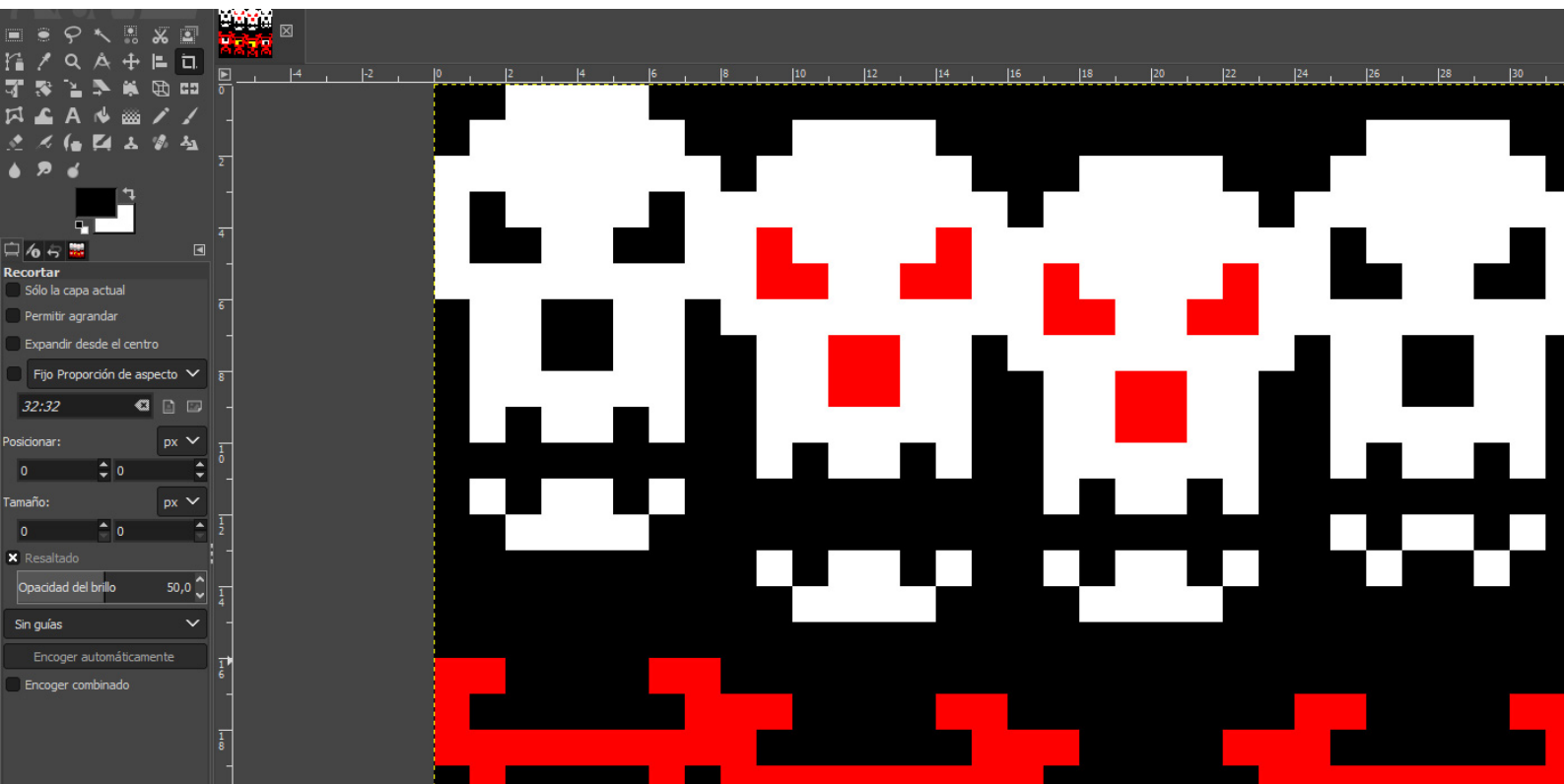
=====
== 1: CPCtelera initial tests
=====
```

1.

Fran Gallego told us that first of all, the game should work in Amstrad CPC 464. But we could not test our game continuously on the machine, therefore, we could use a couple of emulators to work on our computers, Winape and Retro Virtual Machine. It has been really useful to work with them. RVM especially, because it looked a lot like the machine and its loading interface was very similar to the physical machine. Winape, however, had a very easy way to debug the game.

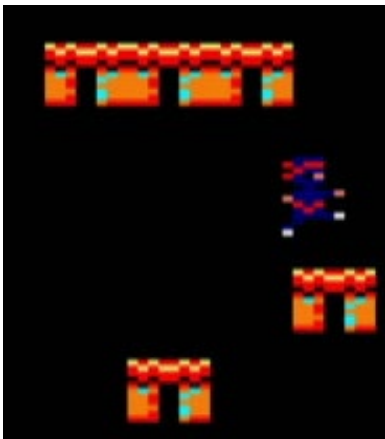
2.

We use a library to simplify. This is CPCTelera, a library written by our teacher and very useful. Given the basic technologies to use, we could choose the rest. GIMP to draw sprites and sprite sheets, Tiled to create the maps and Arkos Tracker for music and sound effects.



day a day

The work during all these weeks has not been easy. We divide the tasks between the three components of the team. However, not everything was so easy. Time limited us a lot.



collisions

One of us started with the most complicated, collisions. We knew this was critical, because it would make the machine have to do many calculations.

Solution? The enemies would float. In this way, they did not have to be on platforms and so we could put them where we wanted.

In addition, our game was divided into levels, why calculate the collisions with everything that exists? Pointers to obstacles and enemies per level.



shoots

At first, we thought that the shots should be infinite. While the game was developing, we saw that it was better to have bullets limited by level. The trap? Well, you'll have to play to find out.



enemies

Our idea was to really add 2 types of enemies. Our idea was to really add 2 types of enemies. Some with patrol and other fixed. The last ones shot when they saw you. That was our initial idea; however, the last days we started having problems with fixed enemies, so we decided to remove them.

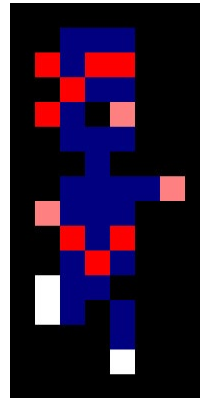
These three things were the most important, but there was still much to do. Do the levels, the sprites, animations, music, changing screens... What was the main problems? Time and technical problems. That your computer isn't pleasant. Luck he had an easy solution.

art

The only thing we knew was that the player was going to be a ninja. We were drawing things in GIMP. It was not easy, considering that the different modes of the machine would make our designs stretch, so it was frustrating to know that we could not have details of a pixel.



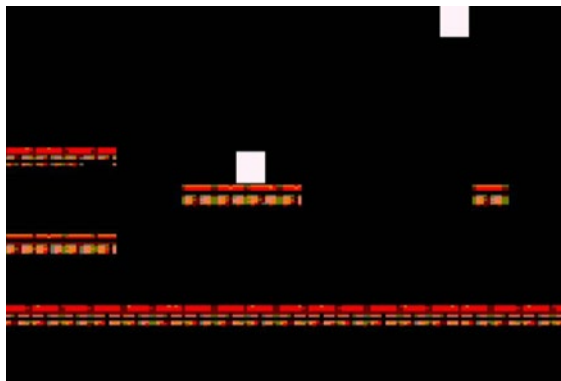
NINJA FIRST DESIGN



NINJA FINAL DESIGN

The tiles, initially we used an orange design that contrasted with black, although they do not convince us too much. One of us found a game in which they used a similar color, so we were inspired by him. This game is called Golden Tail.

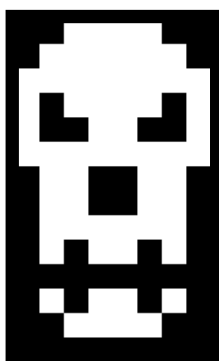
TILES FIRST DESIGN



TILES FINAL DESIGN



ENEMIES



ODOKURO



ONI

music

In the sound section, we have taken inspiration from classical music because we needed a clear rhythm and identified a specific instrument. To do this, we have selected the following musical pieces: Saber Dance by Aram Khachaturian and Moonlight, written by Ludwig van Beethoven.



With the Arkos Tracker software, we have been able to represent these musical works as faithfully as possible. All this, thanks to the score editor that contains the software.

credits

Senior programmer: Rubén Rubio

Junior programmer: Jesús Mas & Judith Mula

2D Art: Jesús Mas & Judith Mula

Music and sound effects: Jesús Mas

Rubén Rubio (@ruben.rubiom)

Judith Mula (@judithmmolina)