

# ARTICLE DEVIL Making-of



Universitat d'Alacant  
Universidad de Alicante

Multimedia Engineering

ABP 2023-2024

# Index

1. About Us

2. Development

3. Technologies

4. Problems and Lessons  
Learned

# 1. About Us

Hi! We are part of Darken Studios, the creators of “Little Devil”. Our team is composed of 3 Multimedia Engineering students from UA.

- Javier Gascón
- Antonio Sanchez
- Gema Martinez

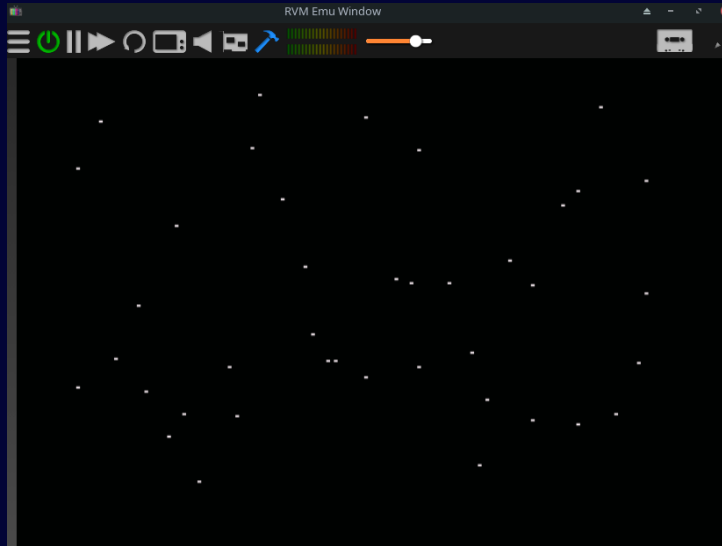
At the beginning of this academic year, our main objective has been to focus on the development of our videogame for Amstrad CPC.

Our primary goal was to take part in #CPCRetrodev 2023 and do a great performance.

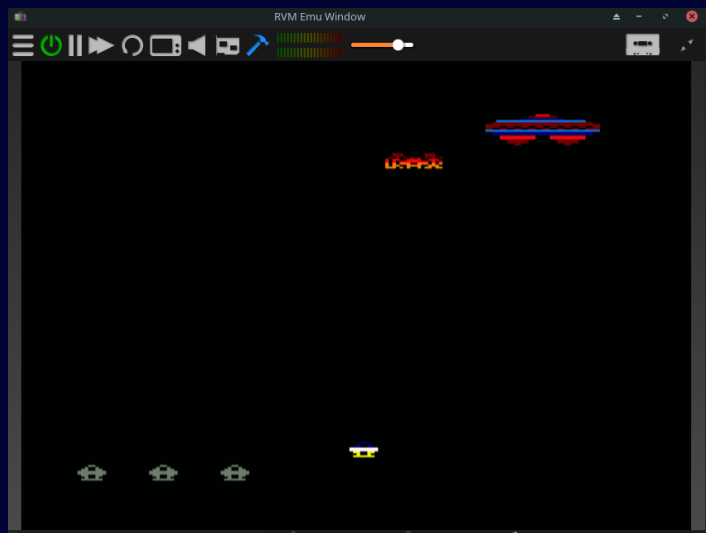
All of this is to establish a strong foundation to face the challenge of creating a 3D video game for the ABP project.

## 2. Development

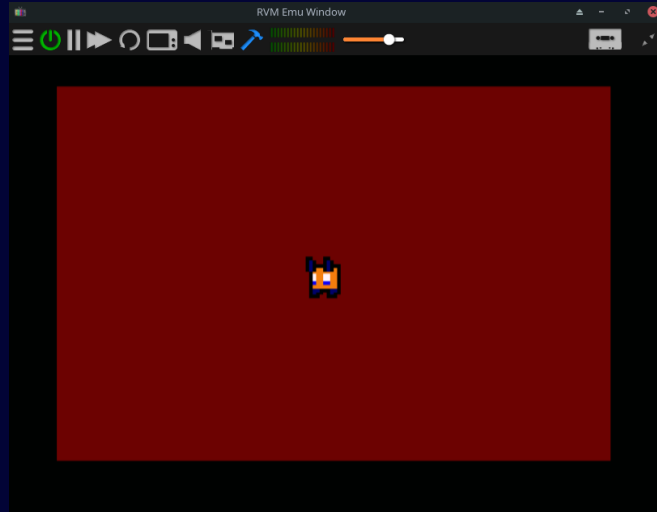
To begin with, we worked on a Starfield effect:



Then we created a clone of Atari Assault to have a robust ECS engine that would allow us to develop our game in a more structured and straightforward manner. This took us quite some time since it was the first few weeks of class, and we weren't familiar with assembly language.



When we started our game, we first created the player entity and provided keyboard input to allow it to move in all four axis.



For the model, we were inspired by the NES Zelda and decided to focus on the dungeon combat aspect without puzzles, similar to Castle Kid, another game from CPCRetrotev. To do this, we created the following room:



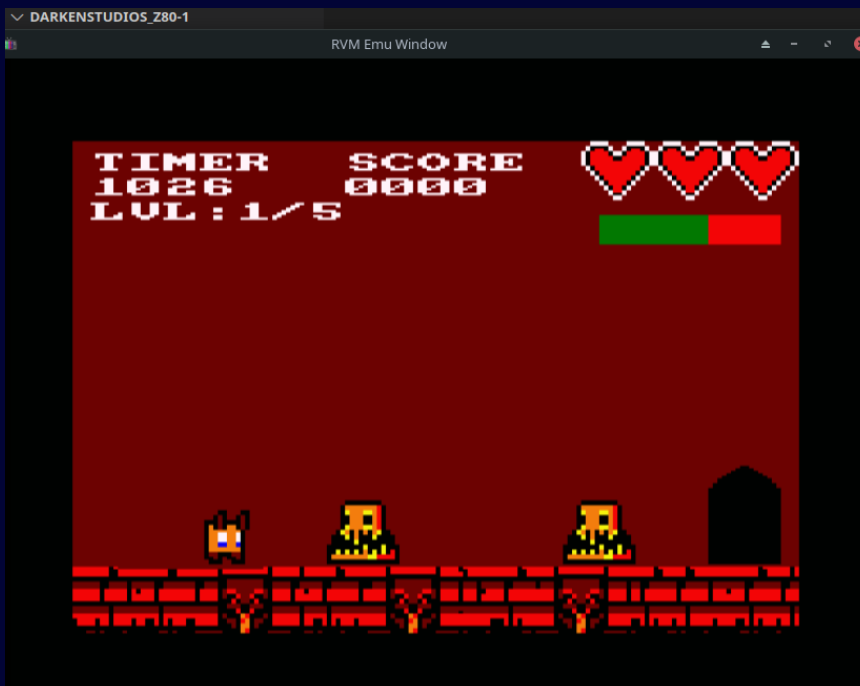
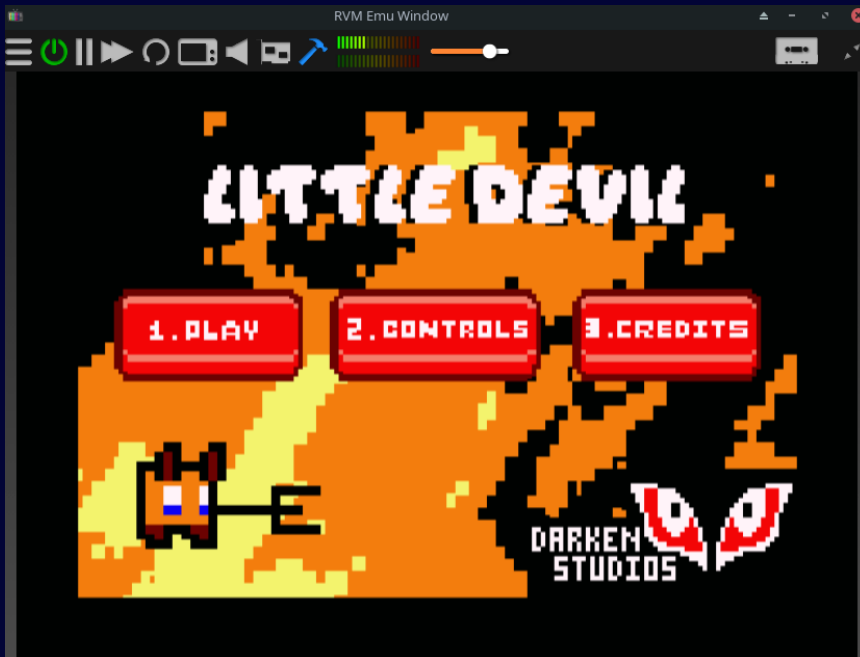
Due to the limited time remaining and the difficulty this idea presented, it was discarded, and we opted for a combat model similar to Crimson Knight Adventures but without side-scrolling.

In the following weeks after the model change, we focused on leveraging what we had from the previous work, creating the enemies and their AI, as well as implementing the scoring system, time, and lives.

In the following table, the entities we have in the game:

<b>PLAYER</b>	
<b>LAVA ENEMY</b>	
<b>WIZARD ENEMY</b>	
<b>FIREBALLS</b>	
<b>STONES</b>	

Finally, the level system and the creation of enemies based on the current level were introduced. The final look is as follows:



## 3. Technologies

### CODE:

- Visual Studio Code: Streamlined code editor.

### ART and MUSIC:

- Aseprite. Animated sprite editor and pixel art tool.
  - Gimp. Secondary art tool.
- Arkos Tracker: Ultimate musical tool for 8/16 bits computers from the 80's.

### EMULATION:

- CPCTelera: Astonishingly fast Amstrad CPC game engine for C and Assembler developers.
- WinAPE: Amstrad emulator for PCs.
- Retro Virtual Machine (RVM): Zx Spectrum, Amstrad CPC, MSX-1, SG-1000 and Sega Master System Emulator.



## 4. Problems and Lessons Learned

### PROBLEMS:

The first serious problem we had to face was creating a proper ECS model through the Starfield effect. Initially, we didn't understand how assembly language worked well, and this task was completed two weeks after its initial deadline. This also set us back in having a more robust ECS engine for the Atari Assault.

A problem that has been haunting us for quite some time is figuring out a rather unusual bug that occurs occasionally when you keep the sword active for an extended period and kill an enemy. It seems that the weapon entity remains alive even after deactivating the sword.

Understanding collisions also took us quite some time to implement in assembly language.

Another problem we've encountered is debugging in WinAPE, as it's a tedious task that requires a high level of concentration.

### LESSONS:

The most important lesson we've learned is that asking for help on a task you can't solve on your own doesn't make you worse. Sometimes, we just need someone else's perspective on the problem to find the solution on our own.

Another interesting point has been the concept of 'Less to More.' We have been more efficient when we divided tasks into shorter and simpler ones and then combined them in the more challenging parts.