# PENGUIN CROSS

## MAKING OF

# INTRODUCTION

Penguin Cross is a video game for Amstrad CPC 464 programmed in assembly by 3 Multimedia Engineering students at the University of Alicante.

The authors of the game are Francesc Bellido Delgado (paco.bellido.delgado@gmail.com), David Mulero Pérez (davidmuleroperez@gmail.com) and Wei He (weicocn1@gmail.com).

Our group is **Cuboid Studio** and you can contact us through Twitter @CuboidStudio

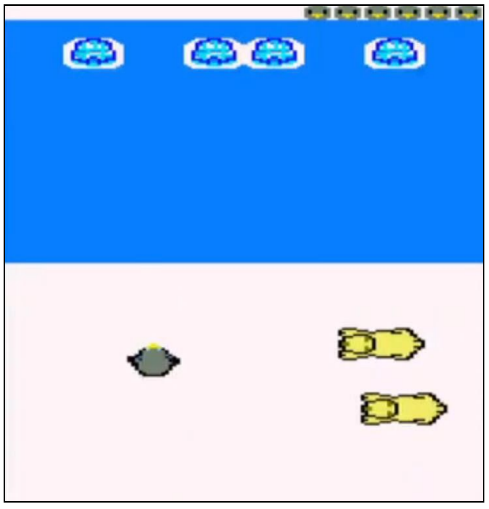
In this file we will explain how it has been programmed during the 5 weeks in which it has been developed.

# HOW WAS THE GAME DEVELOPED?

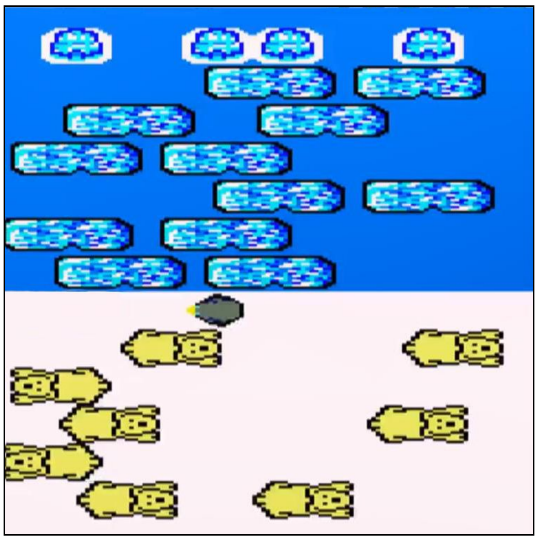
In order to develop the game we have used assembly language. For 2 weeks we began to follow online courses to understand machine code and started to use assembly language. First we draw some sprites in machine code, programmed some animations in assembly and then we start to make an initial ECS (Entity-Component-System) structure to make a Starfield effect.

Once we got a good base of learning we started the development of our game that has lasted 5 weeks.

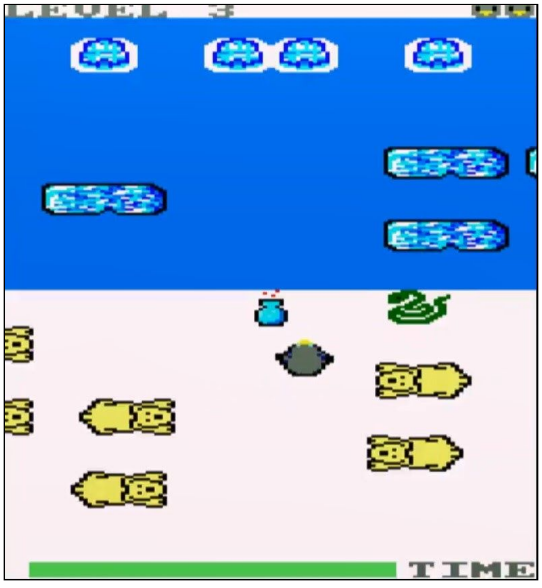
- In the first week we made a playable prototype where a main character could move and collide with enemies. We also implement igloos that work as a goal.



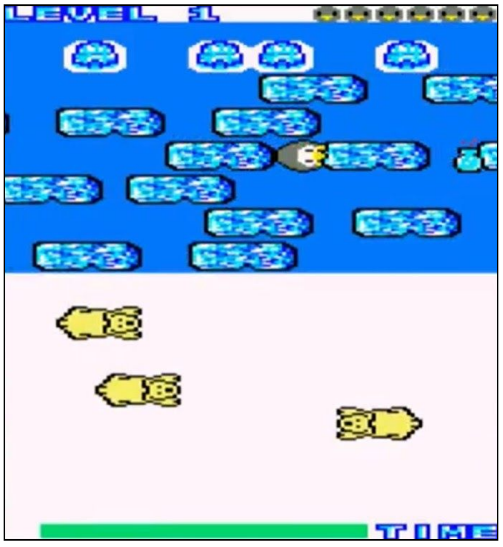
- In the second week we implemented the platforms that the player can get on to avoid drowning in water and the enemy generation system.



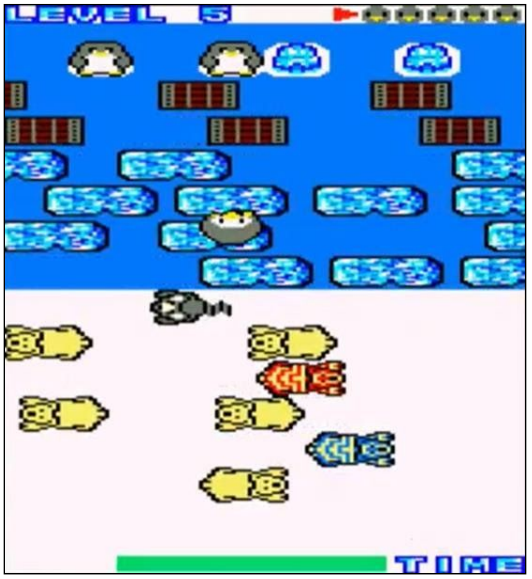
- In the third week we started the development of items, created the level system, new enemies (raccoon) and new behavior for the platforms and implemented the clipping of the enemies and the use of the mask to draw the items and the main character.



- In the fourth week we developed the double buffer to avoid flickering, implemented interruptions to control the playing time and made a first contact with the music.




- In the last week we have finished the music, improved the game design, including enemies and alternative platforms, implemented the menus (credits, controls, game over) and drafted the necessary documentation to be able to present the game to CPCRetroDev 2020.



# USED TECHNOLOGIES

- We used the library **CPCtelera**, posted under GNU GPL with Copyright (C) 2018 ronaldo / Fremos / Cheesetea / ByteRealms (@FranGallegoBR).
- Sublime Text 3 / Visual Studio Code
- Git Hub
- Arkos Tracker 1
- WinAPE/Retro Virtual Machine
- Gimp / Photoshop

# PROBLEMS FOUND

During the development we have encountered some difficulties that may have delayed our programming pace. WinAPE has caused us some problems such as not being able to open or the strings not appearing (in case of opening several instances of the program the configuration is corrupted and there is no ROM, the fix to this problem was to reset WinAPE). Also, it was the first time that we used Git so we had to learn to use it in parallel with the development of the game.

Regarding the development,we had several problems when reducing the size of the screen since we had to edit many of the CPCtelera functions, also in the implementation of the double buffer we had to change many of our functions in order to make the change of buffer correctly in every moment. For the implementation of clipping we need to perform the calculations for deletion and it took us a long time to do it correctly.

On some occasions we have encountered bugs that corrupted the game (functions that destroyed registers, functions that we had to edit when double buffering, interrupt management ...) and we had to debug with WinAPE to solve them.

# LEARNED LESSONS

During the development of Penguin Cross we have learned to better understand language at a low level. During the first weeks of the course we began to study machine code and assembly language.

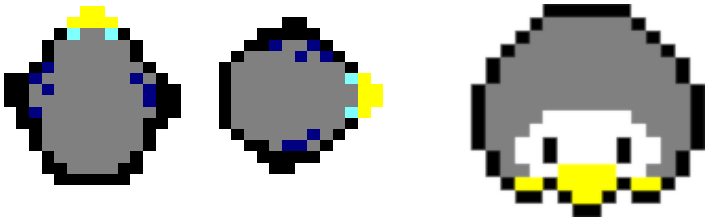
In addition, we have learned to make use of Git and GitHub, which has been very useful to us to be able to perform version control of our code and has greatly facilitated teamwork.

On the other hand, we have improved our ability to create sprites using Gimp / Photoshop and 8-bit music using Arkos Tracker 1.

# SPRITE EVOLUTION

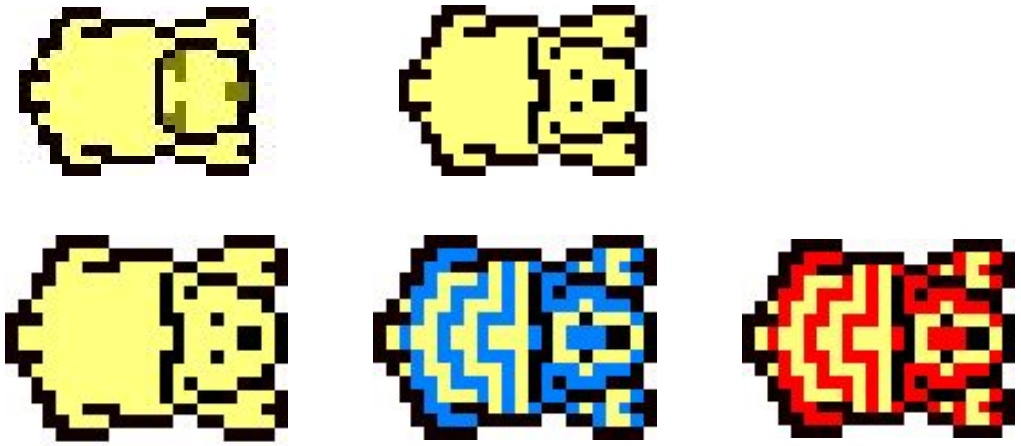
Now we are going to show the evolution of some of our sprites. The creation of sprites has been maintained throughout development and many of them have needed to improve their initial version to be able to look correctly. As our ability to create them has improved, it has become easier to create them with a better look.

## Player



## Goal

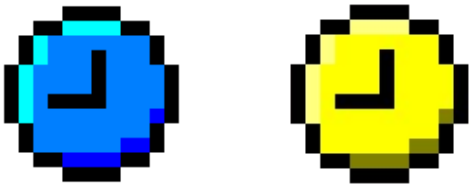
## Enemy



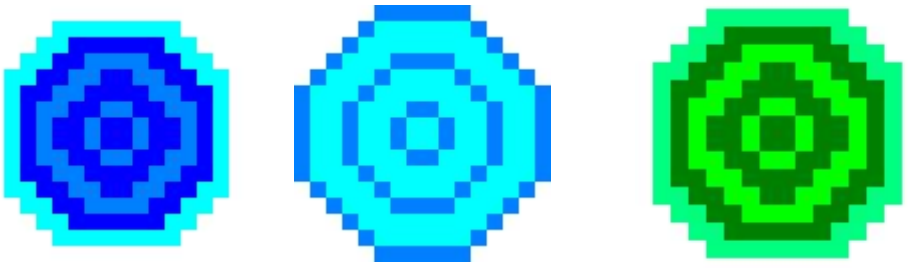
Platform

Sunken Platform

Clock



Portals



Text font