

COSMIC GARDENS

Making-Of







TABLE OF CONTENTS

1.HOW IT WAS MADE	4
2.TECHNOLOGIES USED	5
3.PROBLEMS & SOLUTIONS	6
4.PICTURES AND SCREENSHOTS	7
CREDITS	13



1. HOW IT WAS MADE

Cosmic Gatlin has been an interesting challenge in our career as developers and students.

First of all, we started coding an Entity-Component-System engine, basically the base of the game. Our main programmer improved the code while our graphics designer started creating sprites.

In just one week, the team made a basic playable demo of our game. It was just 2 squares, red and yellow, if you pressed P, a bullet would appear and destroy the yellow square.

During the following days, sprites were added to the game. Now our “red square” had a *very cool sheriff shape*, and our enemies transformed into robots. The team started very strong, but we slowed down a bit in the middle of the project. Maybe it was just stress for being only 2 people carrying with all the development.

Near the end, we did a review of the video game and it was too simple, so we added more enemies and a difficult system.

We are very proud of the final version of the video game and we hope you will enjoy it.



2. TECHNOLOGIES USED

For our game we used these technologies:

- **CPCTelera**: An Amstrad CPC game engine to basically make our lives easier.
- **Visual Studio Code (VSCode)**: Main source-code editor.
- **Aseprite**. Main sprite editor and pixel art tool.
- **Gimp**. Secondary sprite editor.
- **Arkos Tracker 1.0**. Main music tool.
- **WinAPE 2.0b2**. Main Amstrad CPC emulator for testing and debugging.
- **Retro Virtual Machine (RVM)**. Secondary Amstrad CPC emulator for testing and debugging.



3. PROBLEMS & SOLUTIONS

In the course of creating the video game, we encountered not so many problems. The main problem was that we are only 2 people making a game devised for groups of 3 people.

We had to work harder than other groups to make a videogame equal or better than theirs. This has taken us a lot of stress and burnouts, but we achieved something that we are very proud of.

Being a newbie programming an Amstrad CPC game is a total chaos. Both of us had to learn from scratch and keep rising.

Another problem that we encountered was the size of the screen. In mode 0 screen, pixels are wider (1x1 mode 1, 2x1 mode 0). This has given us a lot of headaches and made us lose a lot of time.



4. PICTURES AND SCREENSHOTS





ChillBoat Studio | *COSMIC GATLIN* 🌈

@ChillBoatStudio



here it is, the first playable version of cosmic gatlin! 🌈



(but there's a lot more to do, this is just the beginnig...)



#CPCRetroDev #IndieDev #UAGames #AmstradCPC
#Amstrad #GameDev




537 reproducciones

0:03 / 0:23



1:34 p. m. · 12 oct. 2021 · Twitter Web App



ChillBoat Studio | *COSMIC GATLIN* 
@ChillBoatStudio



now we have **POWER UPS** ⚡ (and also some fancy
sprite animations) #CPCRetroDev #IndieDev #UAGames
#AmstradCPC #Amstrad #GameDev #WIP



11:42 p. m. · 14 oct. 2021 · Twitter Web App









ChillBoat Studio | COSMIC GATLIN 

@ChillBoatStudio



NEW FEATURES:

- 1 now the enemies *literally explode* when you shoot them  
- 2 if the enemies hit you, you will lose 1 star (if you lose 4, it's game over)  
- 3 the ^{damned} flickering bug is finally fixed  

#CPCRetroDev #IndieDev #Amstrad



11:55 p. m. · 19 oct. 2021 · Twitter Web App



ChillBoat Studio | COSMIC GATLIN' 🌈

@ChillBoatStudio



we tested Cosmic Gatlin' on the original Amstrad CPC computer! it works great 😎¹⁰⁰

#CPCRetroDev #IndieDev #UAGames #AmstradCPC
#Amstrad #GameDev #RetroGaming



6:04 p. m. · 19 oct. 2021 · Twitter for Android



ChillBoat Studio | *COSMIC GATLIN* 🌈
@ChillBoatStudio



👁️ take a look at the loading screen and the controls and powerups help screen! 📖 🎮

#amstrad #amstradcpc #cpcetrodev #pixelart #retro #indiedev



6:38 p. m. · 27 oct. 2021 · Twitter Web App



CREDITS

Main Programmer

Javier Botella Martínez

Graphics & Music Artist

Francesc Martínez Torregrosa