MAKING OF

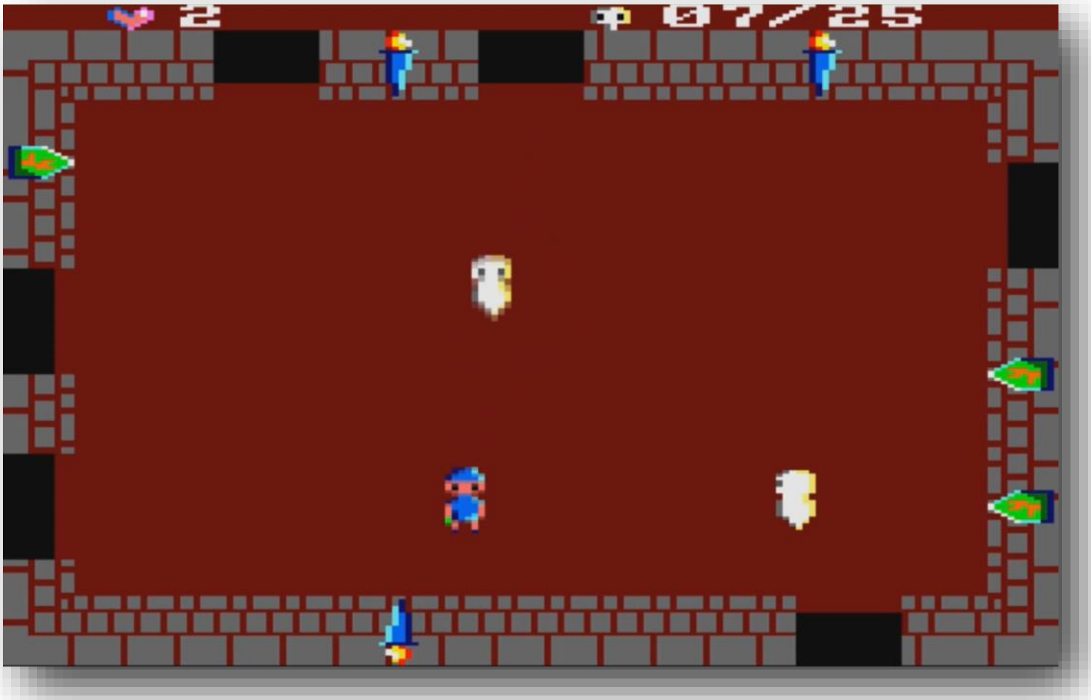
Index

Introduction

Castle Kid is a game created in assembly Z80 language for Amstrad CPC. Firstly, as it was the first time we created a game for Amstrad, we had to learn about the technology. To do this, we did a course of machine code and assembly language in September.

Once, we had learnt the basics, we could start with the **first prototype** of our game which we did during the first week in October.

Finally, we added new enemies, multiplayer mode, customized controls, etc.



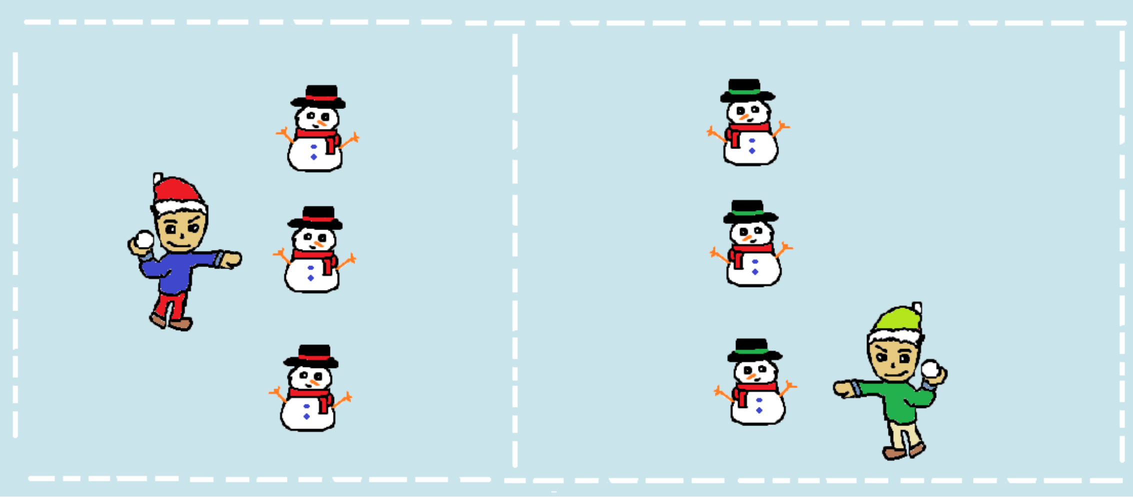
Now, we are going to go in depth with every single aspect in the game to explain how it was made.

Idea of the game

Choosing the **main idea** of the game was a little bit difficult because we were not able to have an idea with which we could develop a game funny and with good AIs. Additionally, we did not know what we were going to be able to do in assembly language due to our lack of awareness in this field.

First, one of us proposed a **shooter game** in which the player must shoot to an enemy inside a map like Pac Man. To be original, the player could only shoot backward so if the player were moving to the right he only could shoot to the left.

Another idea we had, was a 1 vs 1 **Snowball Battle**. Each player should be on one side of the map in which there would be three snowmen to protect from the enemy's snowballs. Here it is a sketch we did then:



However, finally we all agree with the idea of surviving from zombies inside a dungeon. Like this, we could have different AIs and include a multiplayer mode in case we had time.

Artificial Intelligences

From the beginning, we knew that the AIs are one of the most important parts in the game. So, at first, we wanted to have a very complex one. Consequently, we could not even specify one for the first enemy. Soon, we learnt that simples AI could also look so good. In fact, it can easily happen that complex behaviours look bad.

So, we defined a lot of simple behaviours that could be in the game. As the first idea of the game was surviving from zombies, the first AI we introduced was chasing (just moving to the player's position).

However, we knew that we needed more enemies with different behaviours, so we wrote a list with all of them:

- **Zombie** (the current ghost): they chase you
- **Archer/wizard** (the current cannon): they throw an arrow from the distance
- **Squire** (the current crabs): they had a shield to protect themselves from the player shots.
- **Ball:** they move and rebound off the walls.
- **Sprinter**: they change their speed periodically

Some days later, we changed the last two enemies and we added:

- **Squeleton**: they move only in one direction (only X or Y axis)
- **Duck**: they move randomly

Level Design

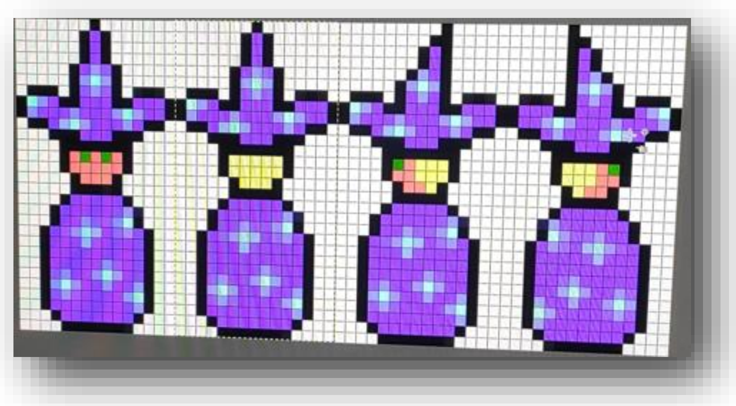
In order to do the level design, at the beginning we spawned the enemies randomly periodically. However, soon we realised that if we had in a table the enemies spawn data, we could better customize times and positions of every single enemy. And that was exactly what we did.

Once, we had our system working we designed six levels. In each one we showed a new enemy, but we made a big mistake: introducing the enemies too fast. As we had been developing the game, we knew how the enemies would attack and we could predict their movements so if a new player had played it would have been very chaotic and even frustrating.

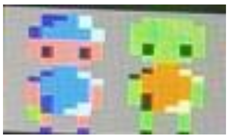
From that moment, we had to redesign all the levels we had to introduce new enemies bit by bit and have a progressing difficulty.

Art

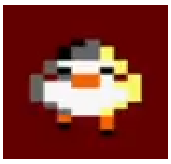
Regarding the art, we started creating a wizard and a zombie.

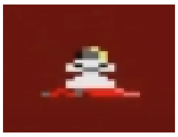


However, when we rendered them in the game, we realised that they were too big, so we had to changed them. Then, we did the blue character who has remained as the main character until the current version of the game. The other one was the zombie that would turn into a ghost.



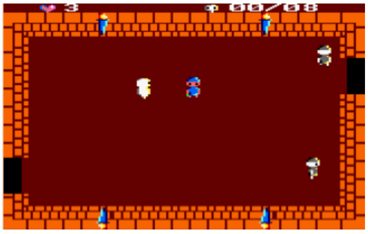
From that moment, we created more enemies as an archer and a squire but then we realised that they could be mixed up with the player. So, we had to redesign once again the enemies. To do that we used a palette with gray colours.



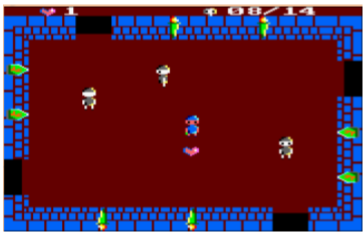


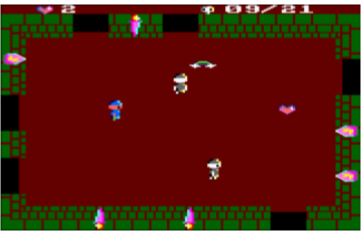
Finally, as we added a multiplayer mode, we also created the player2 (with green colours).

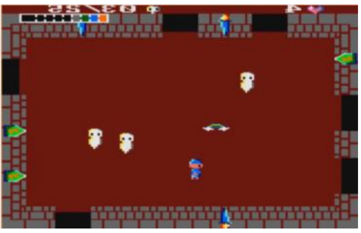
As for the dungeon, in the beginning the background colour was gray but we changed it to red so that the player and the enemies could be seen correctly.



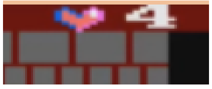
We also changed the walls as the player is completing rounds in the game to create a feeling of progressing.

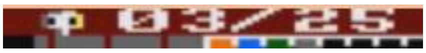






Regarding the HUD and the objects, we made simple sprites and eady to understand like a heart for the lives, a skull for the deaths and a rainbow for a special attack

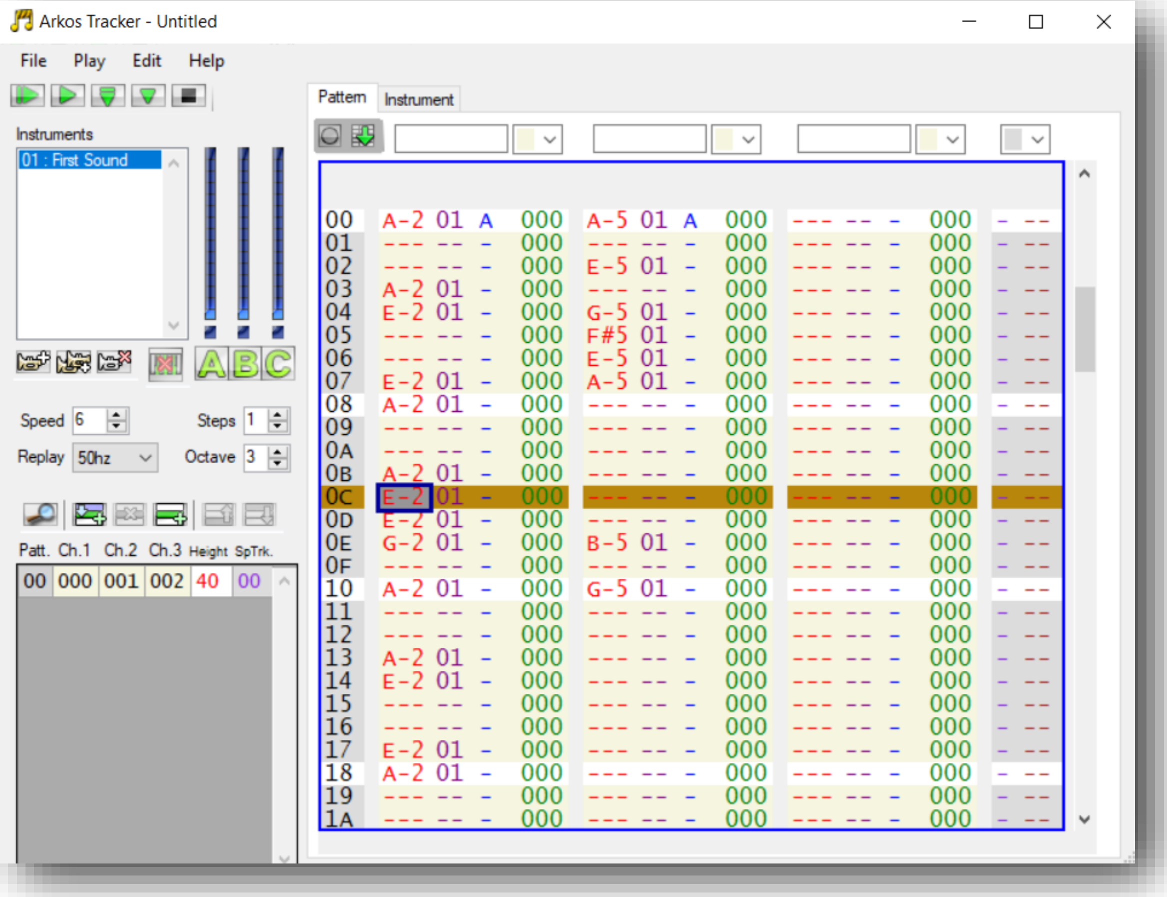




Music

The music and the SFX have been created with Arkos Tracker 1. Firstly, we created a trial music and sound effects just to check that we could hear everything well. Later we got in touch with a friend so that he could help us to compose the music in the game.

To create the music, we used a single instrument and we had to change the volume so that the SFX could be heard.



01 : First Sound

Speed 6
Steps 1
Replay 50hz
Octave 3

Patt.	Ch.1	Ch.2	Ch.3	Height	SpTrk.
00	000	001	002	40	00

```
00  A-2 01 A  000  A-5 01 A  000  --- -- -  000  - --
01  --- -- -  000  --- -- -  000  --- -- -  000  - --
02  --- -- -  000  E-5 01 -  000  --- -- -  000  - --
03  A-2 01 -  000  --- -- -  000  --- -- -  000  - --
04  E-2 01 -  000  G-5 01 -  000  --- -- -  000  - --
05  --- -- -  000  F#5 01 -  000  --- -- -  000  - --
06  --- -- -  000  E-5 01 -  000  --- -- -  000  - --
07  E-2 01 -  000  A-5 01 -  000  --- -- -  000  - --
08  A-2 01 -  000  --- -- -  000  --- -- -  000  - --
09  --- -- -  000  --- -- -  000  --- -- -  000  - --
0A  --- -- -  000  --- -- -  000  --- -- -  000  - --
0B  A-2 01 -  000  --- -- -  000  --- -- -  000  - --
0C  E-2 01 -  000  --- -- -  000  --- -- -  000  - --
0D  E-2 01 -  000  --- -- -  000  --- -- -  000  - --
0E  G-2 01 -  000  B-5 01 -  000  --- -- -  000  - --
0F  --- -- -  000  --- -- -  000  --- -- -  000  - --
10  A-2 01 -  000  G-5 01 -  000  --- -- -  000  - --
11  --- -- -  000  --- -- -  000  --- -- -  000  - --
12  --- -- -  000  --- -- -  000  --- -- -  000  - --
13  A-2 01 -  000  --- -- -  000  --- -- -  000  - --
14  E-2 01 -  000  --- -- -  000  --- -- -  000  - --
15  --- -- -  000  --- -- -  000  --- -- -  000  - --
16  --- -- -  000  --- -- -  000  --- -- -  000  - --
17  E-2 01 -  000  --- -- -  000  --- -- -  000  - --
18  A-2 01 -  000  --- -- -  000  --- -- -  000  - --
19  --- -- -  000  --- -- -  000  --- -- -  000  - --
1A  --- -- -  000  --- -- -  000  --- -- -  000  - --
```