DiveJely

ORIGINAL IDEA

At first, when we were deciding what type of game we wanted to make, we immediately thought about making a game that takes place in the depths of the ocean. From there we considered different ideas, such as making a puzzle-solving game in some underwater ruins or a harpoon fishing game, among others. However, we wanted to convey the feeling of helplessness that one feels at the bottom of the ocean when adventuring the unknown in a territory in which you are not designed to survive and that is why we decided to not provide the player with any means to defend itself, since in the reality, when divers dive into the sea, they are at its mercy. From here the final idea was born, we wanted our protagonist to have to face the sea creatures only with his reflexes and ability to maneuver underwater since in real life what you carry with you underwater will be of little use to you when the time to face a creature that has been evolving under the sea for thousands of years come's. Whether they were jellyfish was more of a personal preference of the team. Finally we came up with the idea of instead of putting a number of lives or a life bar, as most games do, we could put instead an oxygen bar that would decrease over time since when diving it is extremely important to be aware of the remaining oxygen and not to get nervous to not spend it faster (which is why in the game, when you get hit by a jellyfish you lose part of it), after this our game was already born.

MAKING OF

The first thing we did in the development of our game was to create an entity manager suitable for our idea, this incorporated basic things such as their position, speed and dimensions but also included an additional property that we would use to be able to provide them with different AI behaviors. The entire process of creating this manager was a complex challenge because it is the first time we did something similar and also in a language that we had not used before. Little by little we managed to learn and shape the engine of our game. Once we managed to finish the entity manager we started working on the render, input, collisions and AI systems. The latter is the one we are most proud of since thanks to it we were able to provide each jellyfish with totally different behaviors, which makes the game undoubtedly more entertaining. After this we only had to create the levels and the oxygen system, unfortunately for this last step we did not have enough time to extend it as we had originally planned due to other university work and some personal issues. However, we managed to have 3 quite varied and entertaining levels. The realization of these was quite a challenge since we had to take into account the trajectory of each jellyfish that appeared in the level, so that it did not collide with others and that the game was balanced (neither too difficult nor too easy) we tried to make that each level was a small challenge for the player but that at the same time there were no unfair situations and that there was always a way to avoid danger (sometimes easier to see than others).

Thanks to this project we have learned many things such as how to structure a game engine as well as strengthen our programming bases.

ART

Let's talk about the artistic aspect of the game. First and foremost, let's focus on the menu, which was the first illustration we created. We dedicated 5 days of continuous work to perfect it, despite the limitations I faced with the software I was using. At that time, we didn't have much experience in pixel art, and we had never worked on a project of this scale. Thanks to hard work and effort, we managed to master the basics of pixel art through an online course and many trials.

Initially, DiveJely was going to be a survival-focused game. We planned to include 7 types of jellyfish, each with different movement patterns, colors, and sizes. We drew inspiration from the most iconic jellyfish in the world's seas. Unfortunately, the resources we had couldn't represent these jellyfish because they had such complex patterns and a wide range of colors that couldn't be illustrated in a single 50 x 50-pixel image, which represented almost a quarter of the screen.

It was when we incorporated the images into the game that we realized the color restrictions. Many colors didn't look the same as in the original image. Additionally, the pixels on the Amstrad CPC are taller than they are wide, causing a slight distortion in the image and a decrease in its quality. In my future projects, I will try to consider the pixel ratio on the screen, although we admit that it's a considerable challenge as it involves working on a distorted image to achieve a good final result.

Because of this, we had to reinvent the jellyfish, always taking real-world references but not copying them directly. The first jellyfish we created required 7 images for its animation, and we spent a whole day working on it. It was our first experience in creating pixel art animations, and we were really proud of it. However, we realized that we couldn't reach the goal of 7 jellyfish at that pace. So, we decided to create female and male versions of that jellyfish and develop other jellyfish with simpler animations of 2-3 images at most.

One of the most important things in pixel art is the choice of the program you use. In our case, we used LibreSprite, a completely free open-source environment that is very similar to Aseprite, a leading program in pixel art creation. We didn't find any functionality in Aseprite that LibreSprite didn't have. It's a very comprehensive and professional software with features like onion layers for animations, pixel rotation tools, text generation, color palette creation, and many other functions. Once we got familiar with this tool, the artistic creation process of the game significantly improved in terms of quality and speed. We could create 6 jellyfish in a single morning!

Then, we began considering the possibility of including a comic at the end of the game, which was a completely new experience for us. It was in the first panel, which showed a treasure with a diver in the foreground, that we realized the importance of shading and lighting. These elements bring the image to life and make it look significantly better. The second image

depicted a mystical sunset, with a sun that blended with the moon and even a jellyfish, under a starry sky with the shop where the diver would presumably buy their console. Despite the limited color palette available, we discovered that shading could be done by taking the colors adjacent to the main color, even if they seemingly had no relation. The result impressed even myself. Although it took us two days to complete this illustration, we are truly proud of it, and we didn't know we would be able to create pixel art of such quality.

The last image had to be done hastily as we were running out of time for the delivery, so it's not as detailed as the others.

Before concluding, we would like to mention a particular jellyfish, the most dangerous one in the game, the "Eye Jelly." This jellyfish blinks in its animation and synchronizes with its teleportations; in a way, we could consider it the final boss of the game. This jellyfish is inspired by Sauron from "The Lord of the Rings," omnipresent and omniscient, present in every corner of the game.

After all the hard work, we encountered a significant limitation: the computer memory. Unfortunately, we couldn't include the final comic as it exceeded the available memory capacity. Additionally, when we tried to display many jellyfish on the screen at once, they started to flicker, and the game experienced significant slowdown.

To overcome this obstacle, we had to moderate the number of jellyfish in the levels and compress several images. This way, we managed to include both the menu and the comic in the game. If we had more memory at our disposal, we would have wanted to add another introductory comic.

Refeering to music, we also had no experience with encoding it. We learned how to use Arkos Tracker to generate the corresponding signals. Regarding the artistic aspect, we chose to follow the tradition in video games of using a waltz for underwater areas. So, broadly speaking, the songs are a succession of chords in a 3/4 time signature. We had to sacrifice some quality due to technical limitations in terms of tracks or the complexity of creating instruments.

Conclusion:

In this project, we have experienced a significant growth in our pixel art creation skills, and we are really satisfied with this improvement. We are excited about the possibility of applying these skills in our future personal projects.

Additionally, we have gained knowledge in assembly programming. We have also learned to achieve goals while considering constraints, including time limitations, even if it means adjusting the original objectives.