ESAT
INNOVATION SCHOOL
ESCUELA SUPERIOR DE ARTE Y TECNOLOGÍA

Antekeland 2077

Subject: Programación avanzada
Teacher: Gustavo Aranda

Year: 2020/2021
Author: Ricardo Beltrán Muriel & Javier Benito Abolafio

Index

1. Which are the steps from the code writing to the execution of a program?

Firstly, we start with a simple text editor (e.g. Notepad++ Visualcode, etc),to make our source code. This code is written in a programming language that has a special syntax and its own rules.

One we have or source code we use the compiler to create the object code. The process of compilation translete our source code written in a programming language to a machine language. This machine language is specific to each architecture, e.g. is different to compile in a PC to compile in a PS4.

After that we use the linker to do the executable file, but also, it joins all the libraries, all your files, and makes the header of our executable file.

2. What is an Integrated Development Environment (IDE)? What is the difference between working with vs without one?

An IDE is a set of regulations and tools designed to co-work. This includes some development rules, a specific compiling way, and the project's hierarchy and version control system that will be used.

In regard to advantages, IDEs offer a more automated work flow specially when working in a company where everyone uses the same IDE, making it easier to help and communicate each other. Moreover, IDEs are great for typing code and developing in the same framework, so the developer does not feel the need to switch between different applications to develop and debug.

Nonetheless, it can be hard to learn at first and may take some days to get used to it.

3. What is the debugging process with an IDE and how does it help develop an application?

When debugging with an IDE, users are allowed to set breakpoints along the code that will make the execution to stop in there, so that all the memory could be read in runtime, allowing developers to check memory allocations, data and all the information needed.

4. Explain with your own words the meaning of Procedural Programming, Object-Oriented Programming and Events-Oriented Programming, and their characteristics and the relationship among them.

Procedural Programming

It is a kind of programming that follows a determined flow.

OOP

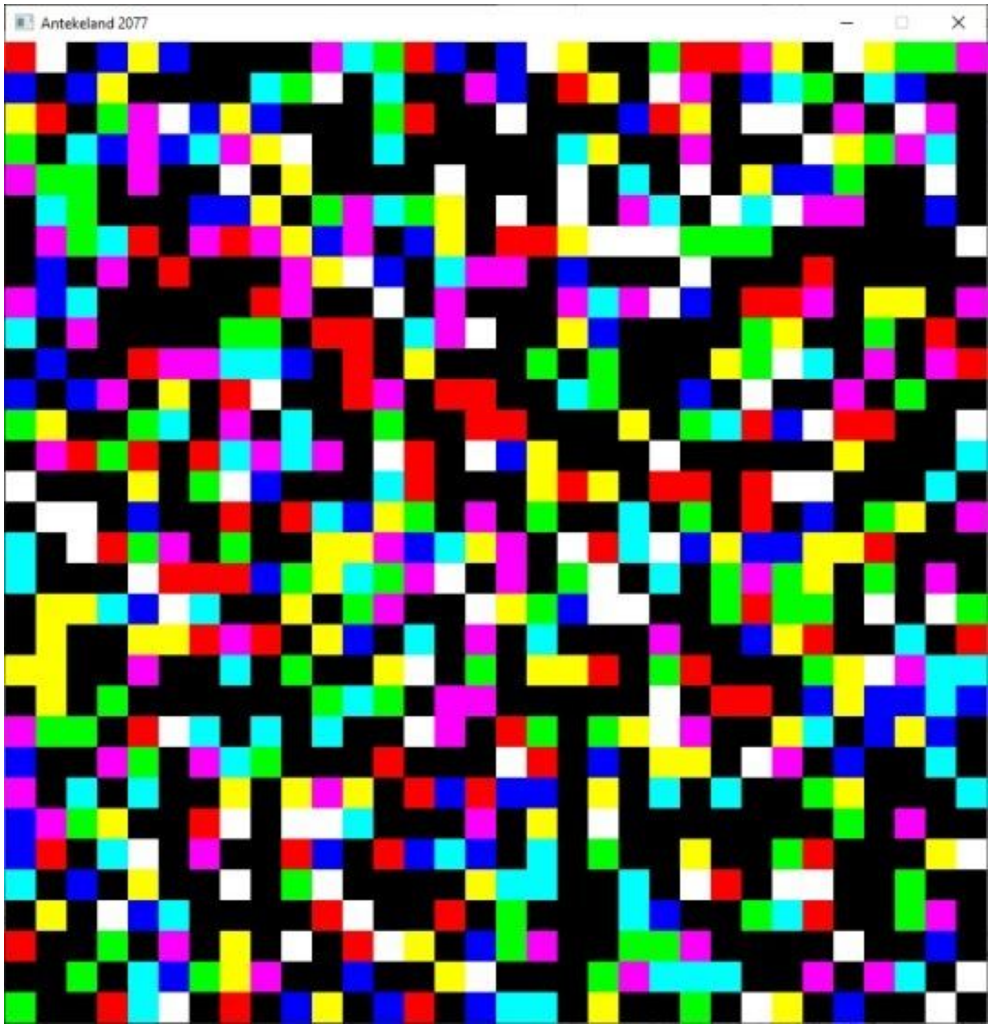
This kind of programming is based on the concept of "objects", focusing on the encapsulation of behaviours.

EOP

This last kind of programming is based on interruptions. It is so common among Operative Systems as we are constantly communicating through interruptions.

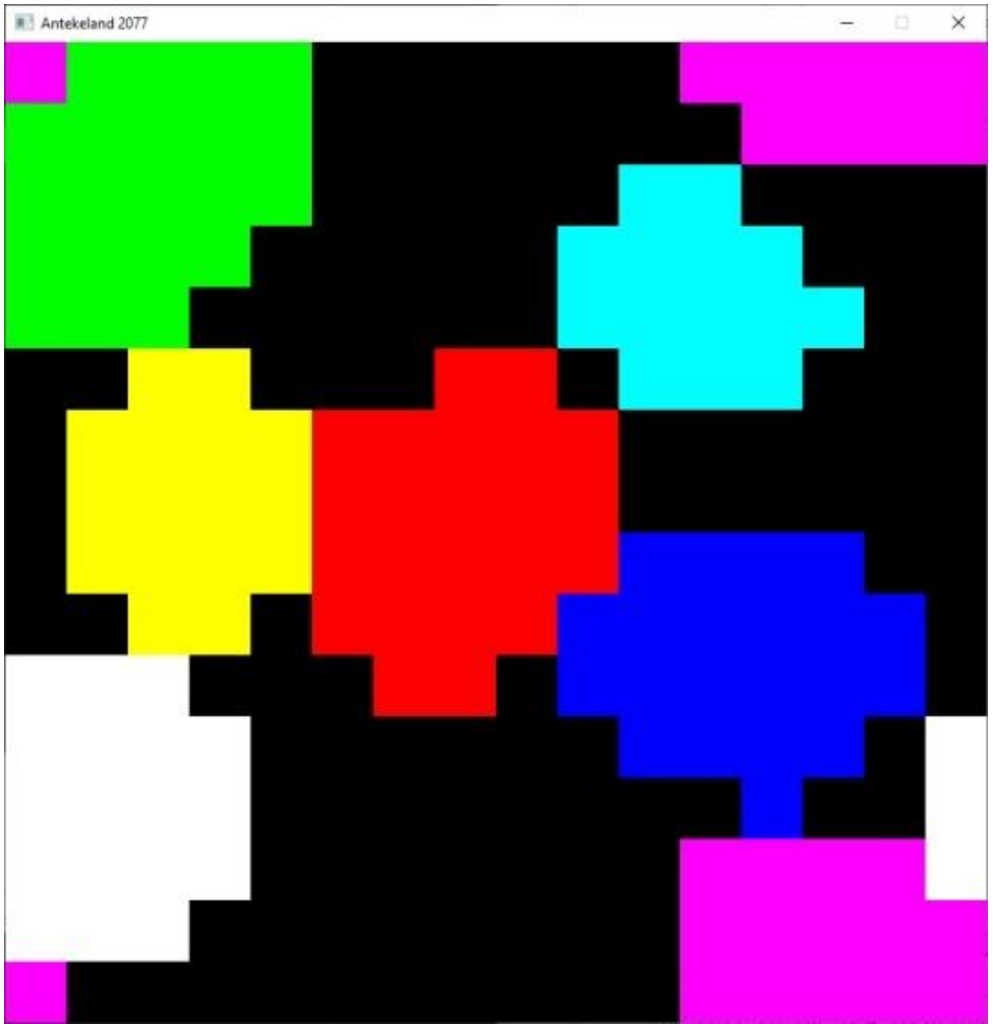
5. Algorithms

Algorithms are a succession of well-defined steps (instructions) that must be followed in order to solve a problem or to perform a specific result. They are normally optimized so it is very recommendable to use them.
Programming Paradigms
Procedural Paradigm
It is in sort way an algorithm since both follow well-defined steps. However, procedural programming is designed to obtain different results in every execution of the application.
In our case, we have implemented a Cellular Automata Algorithm designed to create procedural maps. As we wanted to go a step further, we thought it was a good idea to implement this feature.
This algorithm spreads over a grid different cells, each one with a specific state (between 1-7) so each state will correspond to a color. Once the first map is created, we will obtain a result like this:

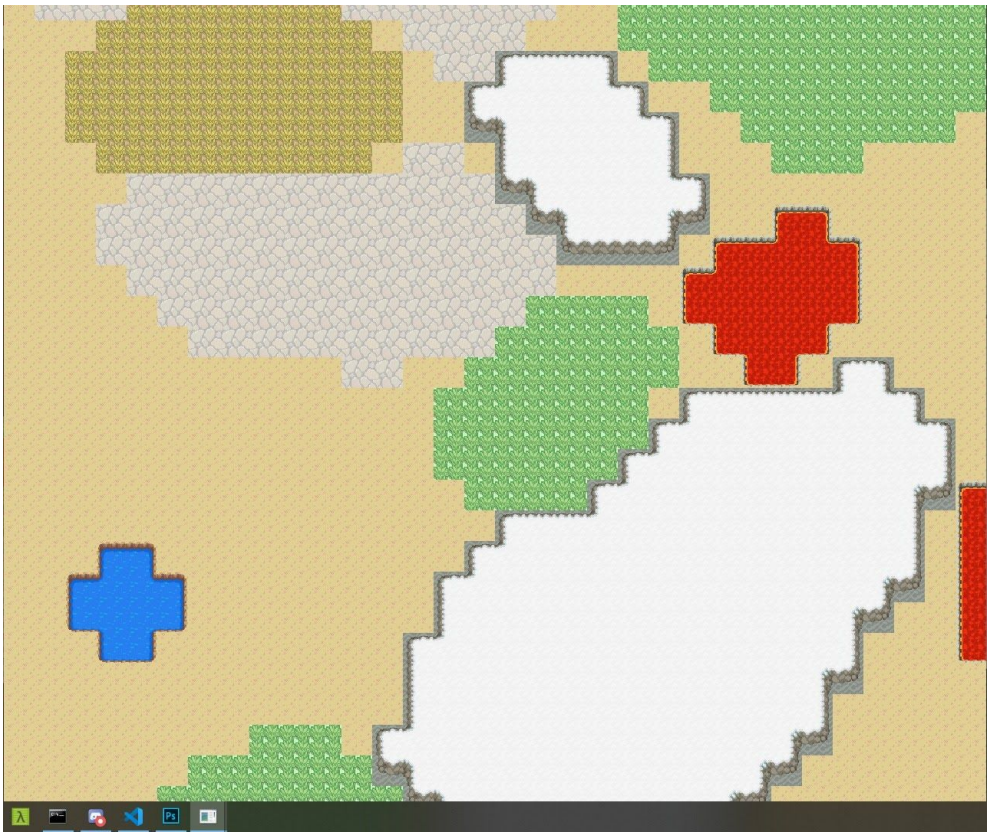


Now, the algorithm tries to order every cell. In order to do that, each cell will check its 3x3 matrix being itself the center of it. Said in other words, each cell

will check its 8 neighbours. Depending on the number neighbours, that cell will suffer different situations. When the map is ordered, we should have something similar to this:



Now it only lasts to replace every single cell into a determined sprite, and the final result will be something like this:

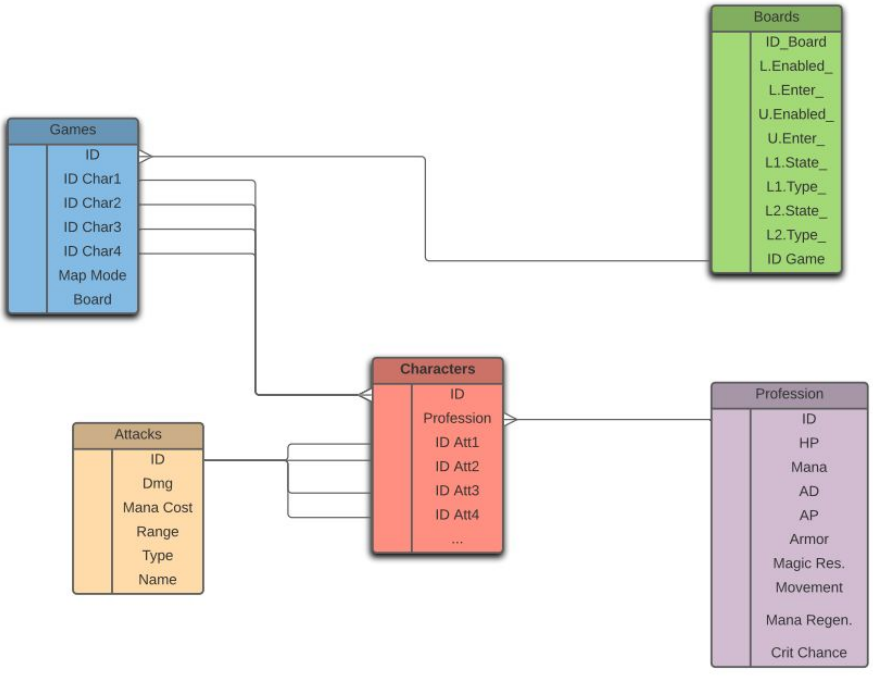


6. Object-Oriented Programming

This kind of programming is focused on encapsulation. This guarantees integrity and privacy to all variables and methods.
Regarding our cases, we have tried to used OOP as much as possible, creating a unique class for the board, character, label, database…
Event-Object Programming
This last kind of programming is based on interruptions. It is so common among Operative Systems as we are constantly communicating through interruptions.
However, we have not made use of any of this.
All of them can be programmed in Visual Studio, probably the best IDE due to its huge debugger. In fact, we have used this IDE over the development of the game in order to detect errors, memory leaks (when loading and not freeing textures and surfaces), or checking some variables' values.
Implementation and Debugging
In our Character Creation, we load different textures as long as the player is changing the character's outfit. At first, the FPS drop was extreme when loading 12 different textures. That was because we were creating and destroying 12 surfaces and textures every frame. Fortunately, we detected that performance error without the help of Visual Studio. However, we found out the game crashed when randomizing the character at least 20 times. In that case we did need to use Visual Studio to debug the execution of the game, and indeed, we found a memory leak when creating and not releasing textures and surfaces. We

actually saw how the memory increased exponentially every time the outfit was changed.
Basically the use of an IDE helped us a lot. It made us detect errors and check the memory in runtime. However, none of us was used to it, so we have to get used to Visual Studio and to debugging.
On the other hand, using plane text editors is easier as they only offer the basic functions to type code. Nevertheless, it is less useful in regard to find errors.
Debugging Process
At first, and in case of weird crashes, there is no need to set a breakpoint. Executing the application will throw an error so we can track it back.
But in case we don't have crashes but bad performances, at least one breakpoint must be set. After that, we execute the application and it will run normally until reaching that breakpoint. From now on, we can go line by line, instruction by instruction or function by function to debug the code.
Programming Normative
It has been one of the key points of our project, the usage of a determined normative. It is really useful for development groups or companies, since they all must develop the same application, coding different parts of it and later on it must be joined. This could be really tedious and hard. However, using a normative obliges everyone to follow the same rules, something that will ease the work.

7. Database



For the database , it is very small but enough for our game, to save a game use the gam table and we save the game id and its character, to know the board of that game we use the board table that it uses to save all necessary information to create a board and the game that it belongs to. Also we have an auxiliary table that we use to store information and it is not necessary to be in the code.

8. Work

Ricardo Beltrán. Did the most of the tools for the project, math tools and other classes such as entity, label, rect, ect. Also did the AI of the character, and the movement of the character, the scenes and the database.
Javier Benito. He did a lot of work with textures, he did the character selection, all the UI and the character attack , and once we got the procedural generation he implemented in the game and put sprites.

Both helped each other to do the task. We do the first approsh of the procedural generation together and most of the ideas had been through together.

9. Post mortem

We run out of time thinking that we can do lots of things but we can't reach all of them. WE work very hard from the beginning but I think we lose time doing things or trying things that maybe we should forget about them and do other things in order to finnish the project. Both finish with a bittersweet flavour we obviously want to finish and have the best project but we can't. the way we posed this project was too big for 2 we didn't realise until the last day.