Portfolio

Germain Le Chapelain

College projects

Intro

In order to give you a rough idea of the projects I realized during my earliest years at EPITA I put together in that document a short presentation each of them, along with the project report I gave to validate the subject.

The `project' subject at EPITA

Epita, a famous French computer Science School in Paris, is a private university which emphasis on a very hands- on approach. The program takes place in 5 years. If most of engineer school require people to graduate from generalist `preparatory classes' (two year of college, being taught mostly theoretical mathematic and physics.) Epita students are enrolled right after high school in an integrated preparatory program, providing additionally to those subjects a very strong introduction to computer science.

As part of those teaching, students are required to realize a project in the programming language taught during the scholar year. That means Pascal programming language the first year, and C language the second year.

The subject of those projects is rather free, nobody is required to develop a video game. That was a personal choice of me. They are to be developed in team of two to four students.

Pirates !

Year	1999 (Epita 1)
Platform	PC.
Language	Pascal
Technology	Delphi, DirectX Direct3D (retained mode)
Tools used	3DS Max (models) Cool Edit (Sounds effects) Rebirth (music)

Genesis

It was kind of difficult to make our mind on a subject not too hard to have something working, since none of us had a serious programming experience at this time, nor even known anything about what was programming exactly at that time.

Overboard, from Psygnosis, was our model for « Pirates! »

We all four had a brainstorming, going throw all ours Playstation demo CD trying to find some originals ideas. We stopped on **Overboard** (Psygnosis, 1997), which was, if not amazingly innovative, still pretty unique, fun to play and very well realized. As far as I know no adaptation for PC was available at that time.

Development

The use of Pascal programming language was mandatory. The use of Delphi was a prerequisite for DirectX.

We took great advantage of the profusion of tutorials and examples shipped with DirectX SDK. The website Delphi Jedi was of a great help in the purpose of using Delphi and DirectX together.

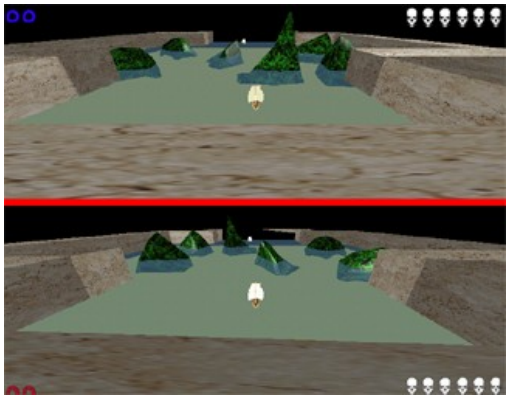
For the enemy artificial intelligence, I came out with a GDI prototype showing basic moves and behavior of a herd of enemy, thus validating further research in that interesting field with my teachers.

We took great advantage of teamwork and parallel assessment, taking advantage of every team members own competences. One working more specifically on the graphic aspect, one other on the camera management. My natural curiosity and my ability to overcome problems led me as a matter of fact to work on the overall game play engine, the 3D engine rendering and the physic engine. I also keep up the work on AI having a personal passion for it.

Results

Well what did we got of all that good work ? A well achieved game project, containing two levels and quite fun to play. The final release contains two complete levels and five different kinds of enemy. Additionally, we successfully integrated a two player mode that brings a new dimension to the game.

Menu items are in French to be even more scary



« Duel to the death »

Being harassed in near your home port.

That hurts.

Darkwave

Year	2000 (Epita 2)
Platform	Sony Playstation
Language	C
Technology	gcc Sony Net Yaroze
Tools used	Amapi 3D (models) Photoshop (graphics and texture) Net Yaroze Support tools (graphics integration)

Genesis

In second year, students are not supposed to develop a video game unless it brings interesting innovation, especially in the field of enemy intelligence. For a long time I would have like to develop a game on a gaming console. The Sony Playstation, for which EPITA had a bunch of the amateur development toolkit `Net Yaroze', made it the platform of choice for that endeavor.

Metal Gear Solid (Konami)

Descent (Interplay)

Being required to come with game with a string level of intelligence and taking place in a smart background, we all were interested in doing something in the spirit of **Metal Gear Solid** (Konami, 1998). Modeling wholes characters was still a bit of an issue, both in terms of difficulty for us, and in terms of computation for the fairly equipped Net Yaroze development environment, which when it has been abandoned by Sony Computer Entertainment was at a fairly advanced level of maturity, some of the feature mentioned in the documentation being simply not implemented ! For all those reasons, we decided to come with that liberal adaptation **Metal Gear Solid** but involving spacecraft inspired from **Descent** (Interplay, 1994), easier to model.

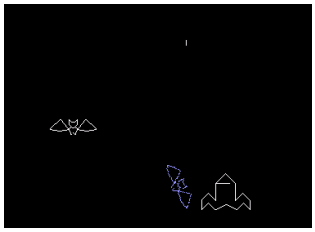
Development

Our team was reduced. Only two peoples from the previous team remained (EPITA is pretty selective). Only one new member was enrolled, keeping the team to reduce amount. Unlike our previous experience, that work strongly focus on the understanding of the platform we were to work one. The lack of examples and the support from Sony being terminated for that project even before having a mature library for amateur development made a whole job of the task of experimenting and studying the API from the available reference documentation only.

My biggest concern was then to quickly have something working on it, and even though far from the actual result I would have expected, which would approach it in its outer form.

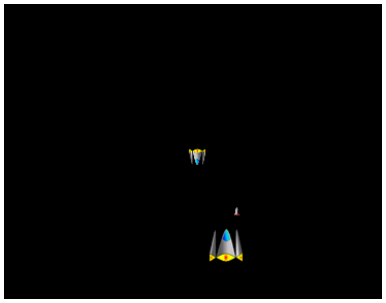
I started developing a very simple game, inspired from eliminator (loriciel) in order to insure our plain understanding of Net Yaroze resources handling, and validate our tool chain of development on Playstation. (We were lent a Net Yaroze Memory card and a serial cable, but no OS CD, so we had to play around with a mod chipped personal Playstation.)

Eliminator on Thomson TO7-70



Unnamed prototype demonstrating Net Yaroze primitives handling

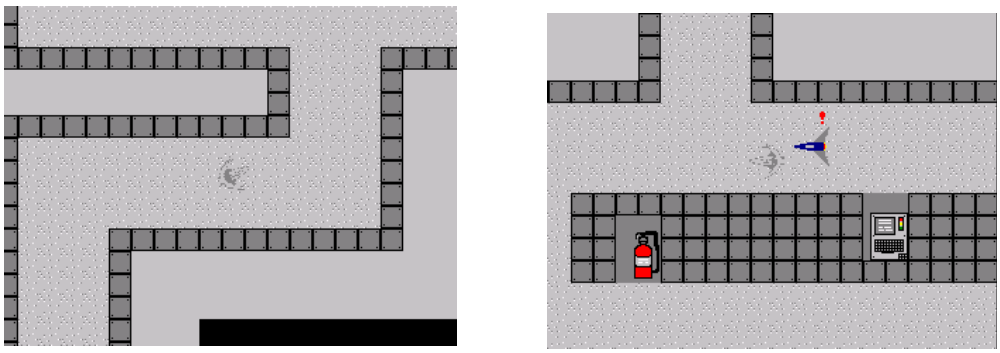
In the meanwhile, another team member worked on sound handling but on the emulator, as we had only one set of development.



Another prototype, emulator validated, to understand sound handling

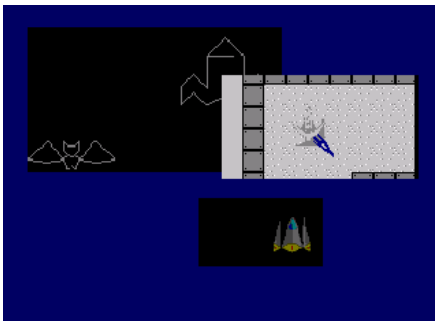
To finish with, feeling self confident of our success managing basic 2D functionality of the Yaroze, I quickly came with a 2D prototype closer to the idea we had of the finished game.

This realization should demonstrate smooth moving and collision detection, as well as a fairly enjoyable artificial intelligence for enemy. It should have also integrated the `alert' modes we required as in Metal Gear Solid. This means that when an enemy ship would detect the player ship it would enter in a special state of mind making it aggressively pursuing the player.



Mini- Darkwave, a prototype much closer to our vision of the final product.

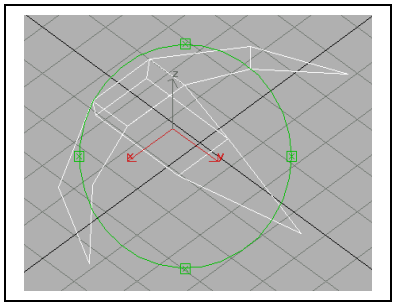
The school evaluation of that subject is mainly a speech about the work that as been done during the preceding period. It is very import that the project looks stable, serious and well managed. For the earliest evaluations, teacher won't expect students to come with achieved work, but they will judge the ability of seriously manage the project on their own based on the consistency of the few work they done so far as well as their ability to communicate and display results itself.

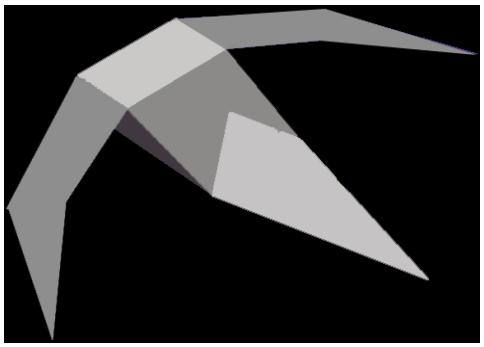


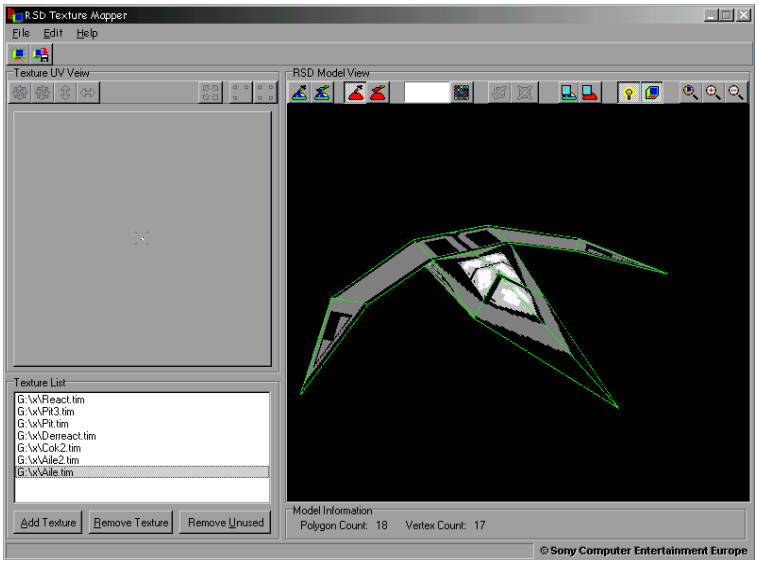
Our intermediate prototype, our how to nicely integrate together heterogeneous cases studies.

That is, I integrated those three small projects in a demo CD rather minimalist, but still nice and intuitive, following a little bit what was done by Sony Computer Entertainment in their promotion CD.

Eventually, with strong understanding of Playstation internals and a good knowledge of the Yaroze API and the development tool chain, we were fully able to handle the 3D rendering.

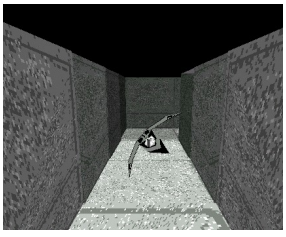






Mesh modeling is done with Amapi, texture mapping with Net Yaroze TIM tool.

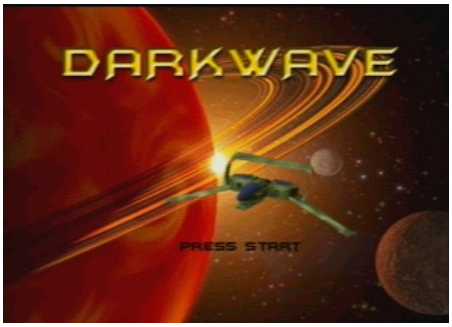
Models as been done using Amapi 3D, a very interesting modeler that allows a good grip on the complexity of meshes in terms of polygons. Graphics as been gathered around the web and adapted thanks to Adobe Photoshop. All of it has been integrated more or less smoothly using tools provided as part of the tool chain.



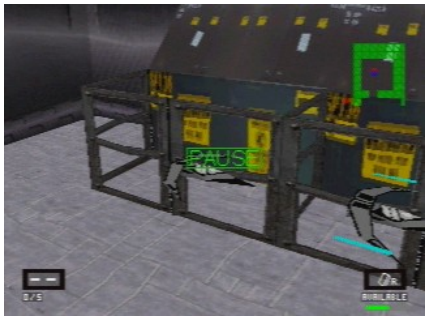
Last prototype before release.

Result

Here are a few screen shots of what you can find in the game. You will need either a mod chipped version of the Sony Playstation able to play backed-up games, or an emulator. ePSXe works just fine.

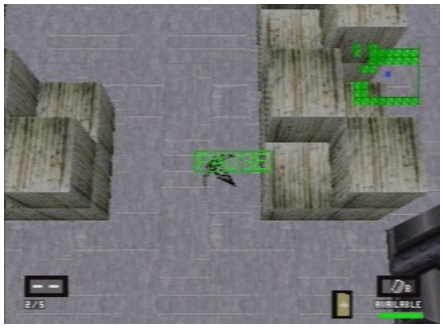


Title screen, done with Photoshop



That piece of the set was my favorite

Gather the key to progress!



Bonus are hidden here and there

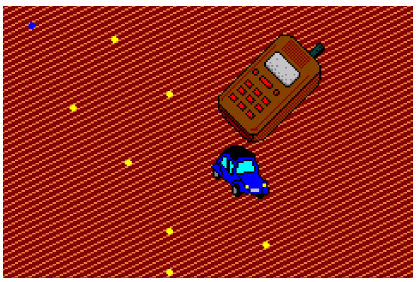
SmallRacer

Year	2001 (Epita 3)
Platform	Gameboy advanced
Language	C++
Technology	Nintendo API
Tools used	Gcc MSPaint

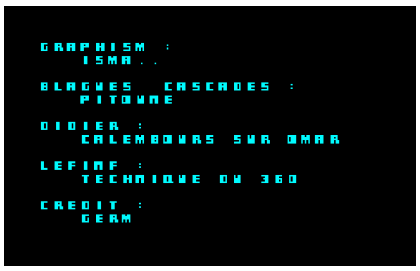
SmallRacer is a very simple project which as no other purpose than to provide a basis for students from the game development laboratory to work with the Nintendo Gameboy Advanced. It shows the usage of register for hardware stretching and transformation of sprite. It also demonstrates the proper way to set memory transfer when using scrolling background. This is it. It is still fun to play for few minutes.

Presentation in mode 0

Title screen in mode 3



The game



Credits

And then ..

And then ? I unrolled the master degree, starting as of the third year in EPITA. I focused a lot on artificial intelligence. Actually, I managed so well in that field that I was allowed to integrate at the same time the public university Paris VI. I kept on doing tons of very interesting project, thus being quite far from video gaming domain (although I worked on mesh simplification during my master thesis.) I would still be very glad to discuss about them in an upcoming interview.

During all that time, the passion for video games has never left me. What I want more than everything else is integrating such an innovative company as Pseudo Interactive to fully express my talents.