Epitech - 2019 Promotion

Avanced C++ Team

KUSH COMA

Alban Cloquet
Guillaume Proquin
Lucien Perouze
Matthieu Lambert
Robin Kerdiles
Samuel Darracq

Project R-Type

R-Type is a side scrolling shoot-em-up arcade game produced by Irem in 1987. The player controls a space fighter named the R-9 to defend humanity against a mysterious powerful alien life-form known as the «Bydo».

Summary

A : Project managment

A 1 : Team

Alban Cloquet

Project manager

Robin Kerdiles

Developper

Guillaume Proquin

Developper

Lucien Perouze

Developper

Samuel Darracq

Developper

Matthieu Lambert

Developper

A 2 : Managment platform

Slack

Slack is used to communicate in our team about the totality of the project. The instant chat allows us to always be aware of the project advancement and to maintain an agreement between our team members. Moreover Slack is really useful to implement links with other applications and project management tools. Indeed, our team uses two channels on Slack which are respectively linked to our Github repository and to our Asana.

Asana

Asana is a task management tool. It can handle several projects on the same time and contains a list of tasks which can be described, assigned to people, and dated. Asana allows us to follow the advancement of each task and to check parts are done before their deadlines. Thanks to the calendar part of Asana, our tasks can be sorted by priority order, advancement and people who are in charge of.

GitHub

GitHub is our online repository manager. It allows to have a git repository and to share our work with the all team via the latter. GitHub is our interface to manage our code advancement, merging of our branches and our versioning on the R-Type project.

A 3 : Build project

CMake

CMake is a usefull tool created to simplify the team work in IT projects. It permits developpers to specify the architecture of the project by listing all the sources and then create the compilation tools specific to the client's platform. From the same project we can easilly create a Makefile or a windows based compilation script.

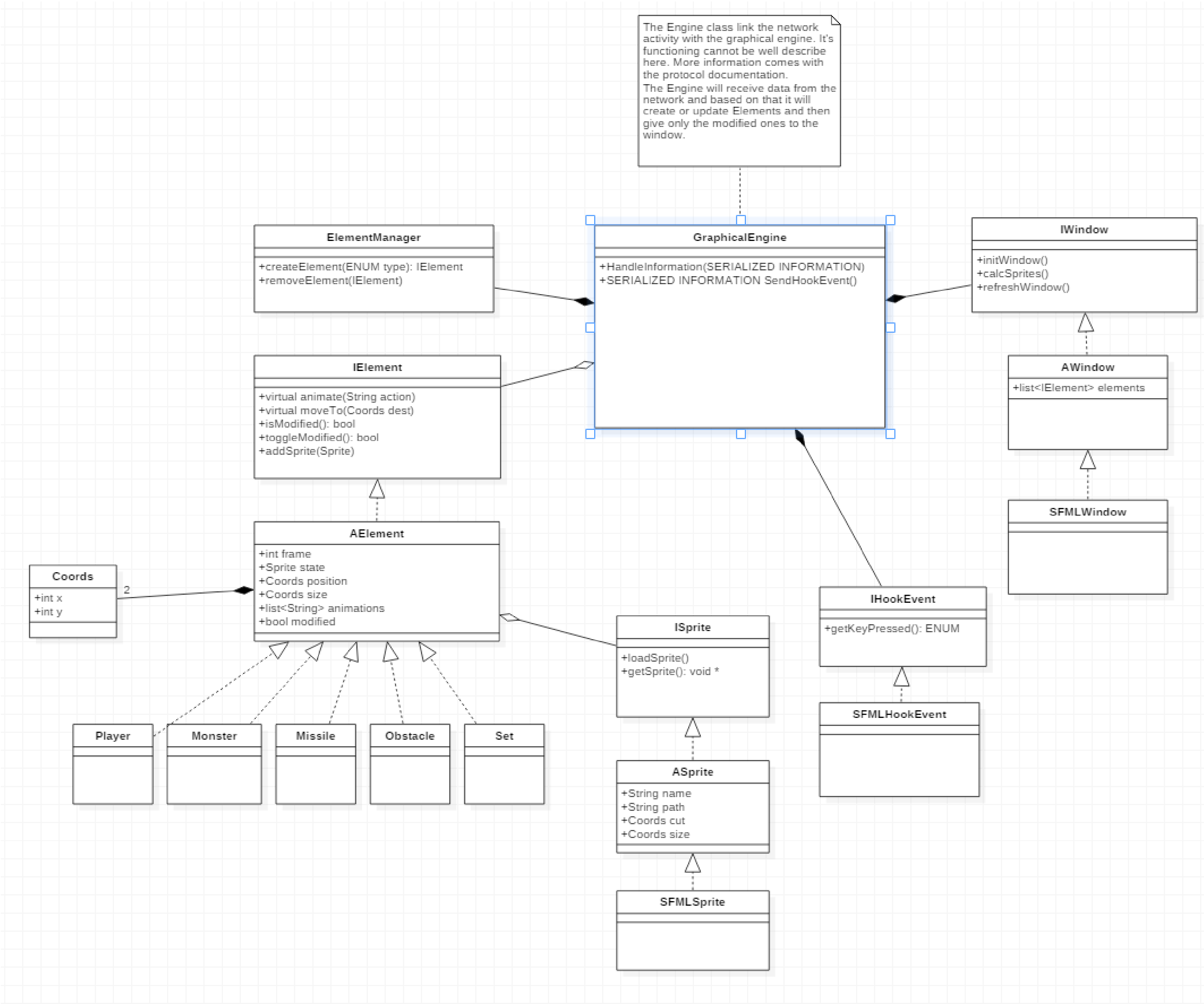
Create compilation script

In order to build the project on any plateform we want, we use CMake. The architecture of our project is set in the files named CMakeLists.txt. To create the appropriate compilation script you just have to execute the following command :

```
$> cmake . -G«IDE Generator»
```

B : Game logic

B 1 : Graphical engine



The Engine class link the network activity with the graphical engine. It's functioning cannot be well describe here. More information comes with the protocol documentation.
The Engine will receive data from the network and based on that it will create or update Elements and then give only the modified ones to the window.

See appendix for better quality.

C : Communication

C 1 : TCP initialisation

The connection of a new client is in TCP to keep the packet integrity. We also exchange the necessery data required by the client such as new monsters or sets.

ⓘ *The protocol is further described in our RFC.*

C 2 : UDP stream

During the game execution, each client streams the server in UDP to display game informations with its graphical engine.

ⓘ *The protocol is further described in our RFC.*

C 3 : Protocol (RFC)

ⓘ *See appendix for RFC and protocol specification.*

KUSH
COMA