

Software Engineering UML & Java The BattleShip Game Project

Sylvain Théry

thery@unistra.fr

ICube

From definition of the game rules:

<https://www.youtube.com/watch?v=q0qpQ8doUp8>

- define the requirements (hardware / software) to develop a software version of the game for:
 - local 1 player + computer
 - local 2 player on 1 screen
 - local 2 player on 2 screen (opposite direction)
 - 2 players / 2 machines over the network (socket)
- for each configuration do iteratively (agile) the specification with UML
 - activity diagrams
 - usecase diagrams
 - sequence diagrams
 - state diagrams
 - and finally when all is clear the class diagrams

You have to model the interface:

- what happened when we launch the game ?
- what happen if you loose ?
- what happen if you win ?
- ...

The game:

- what happen when we touch a boat ?
- what happen when we miss a boat ?
- ????

The ???

Once you have an enough precise modeling,
translate it into programming language Java

activity, sequence, state diagrams help you with
algorithm to implement

class diagram help you with class definition.

Program by pair (on the same computer) to
apply XP rules.

Do incremental development

Begin with a very simple version

- not playing the game for example

And generate iteratively new versions

- enhance the spec
- code the evolution
- test the program
- and go back to spec

Dead-line 22 April

You must provide a git repository on github

Each prototype versions

- must be accessible through a tag
- must have a release.
- must include the modelling diagrams
 - with files or url to access (store on seafile.unistra.fr)

Sources (and headers) should be respect a coding convention of your choice (example <https://google.github.io/styleguide/javaguide.html>) be commented and documented with doxygen.

You provide also a report to explain your development in the README.md (markdown format)