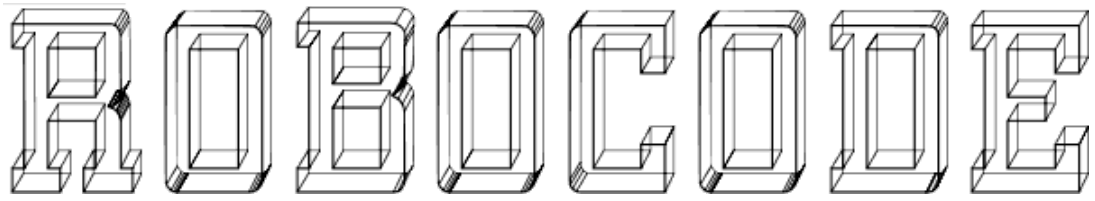
Université de Franche-Comté - UFR ST

Project report - Advanced Web



Students :
Guillaume Weider
Anthony Colez

2017/2018

Table of Contents

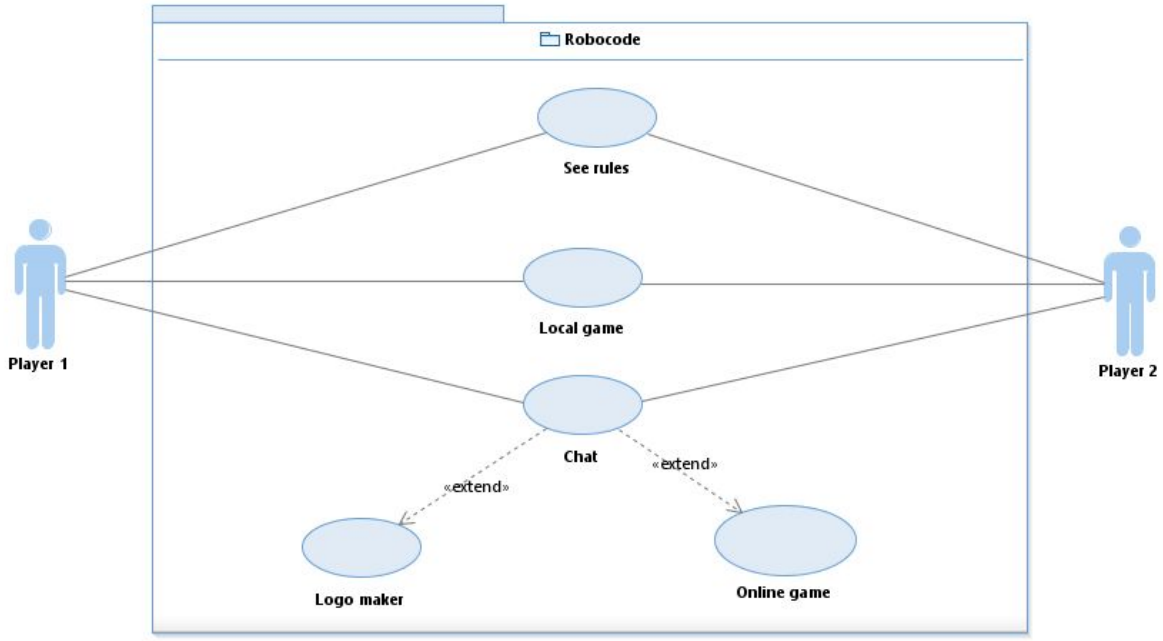
We haven’t succeed to complete the Robocode development for the online game, this project isn’t finished. We try our best to deliver a local game functional. Unfortunately we didn’t succeed to implement the transition for the robot movement. This lack of functionality brings us difficulties to make the communication protocol diagram and the transition diagram.

Introduction

Robocode is a web game, you can play it with only a computer if you have an internet connection. It means you don’t need a game console to play it.
The game is a turn-based strategy and it's a game for two players only.
Robocode is a game in which two robots compete by fetching flags in order to put them in the opposing base. The first robot to drop two flags in the opponent's base wins the game.
The game board is a square of 9 × 9 boxes. The bases of the robots are arranged on 4 boxes at the west (blue robot) and east sides (red robot). The flags to pick up are arranged on the north and south sides of the game board.

At the beginning of the round, each player chooses 5 "code" cards that will compose the 5 instructions of the program that will be executed by his robot. When both players have written and validated their program, they are executed.
The robots follow their instructions one by one, in turn : the first robot executes its first instruction, then the second robot executes its first instruction, and so on, until the 5 instructions have been executed. At the end of each program (set of 5 instructions), players are asked to write their next program.
The game ends when one of the two players has managed to place two flags in the opposing base.

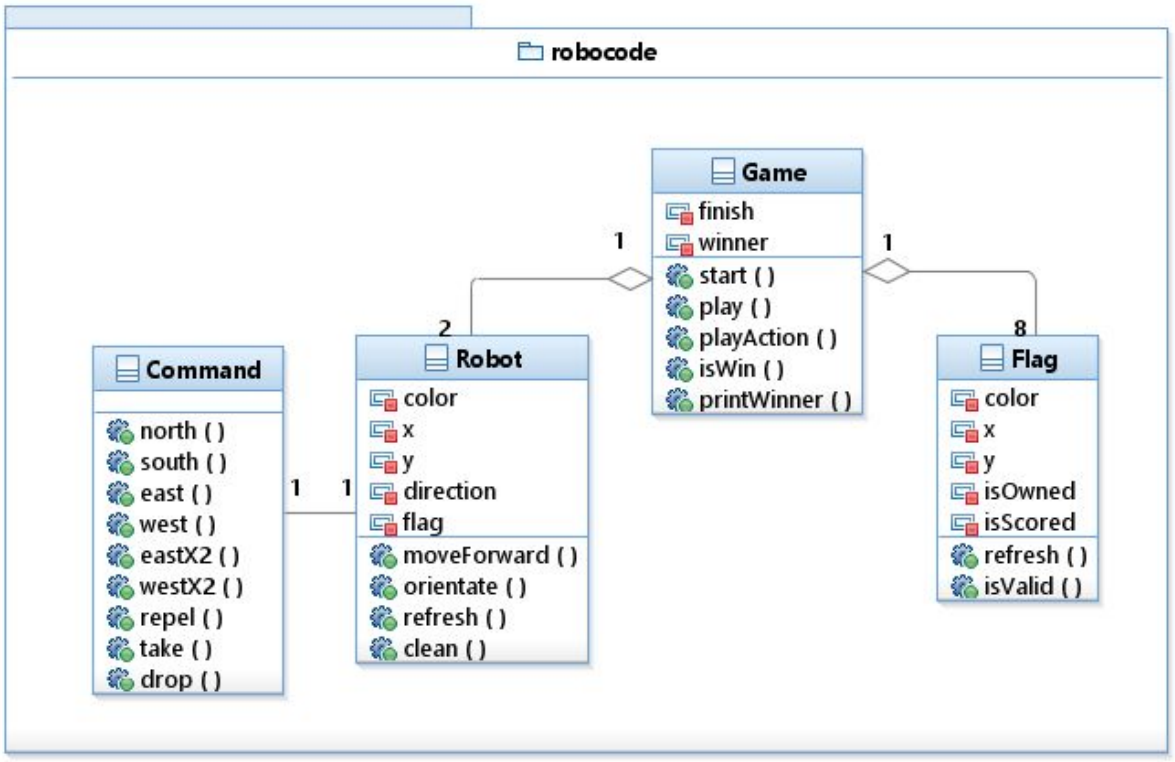
Features



Robocode allow two player to play the game. Players can read the rules on the menu page. They can play in local, which mean that they are on the same device playing one after the other. Finally Players can join a chat where they can speak together, they can create their logo and they can launch an online game, every player play on is own device and the actions can be chose at the same time.

Data model

This classes diagram represent the robocode game in local. We haven't succeed to create the online game so this diagram represent what we implemented. The online diagram would have a Player class which contain the information about the play as the name or the logo of the player.



We have four classes. A Game class which have two instances of the Robot class and eight instances of the Flag class. The Flag class represent a flag. The Robot class represent a robot and have commands represented by the Command class.

The Game class have two arguments, a boolean finish false by default, and which we turn true when the game is finish. The second argument is a string which is initialized with the string "none" and is changed with the color of the winner, or by the string "equality" if the game finish on an equality. Game class have five methods, the start() method to initialize the game and different object like the flags and the robots. The play() method which manage a turn, this method is called when all the actions of the players are validated. The playAction() method play a specific action of the action's array. The isWin() method check if a team reach the score limit and the method printWinner() which display the winner and redirect to the menu.

The Flag class have five arguments, the color of the flag, the horizontal and vertical position, if a robot carry the flag and if the flag has been dropped in the good camp and is scored. The function refresh() manage the display of the flag and the function isValid() check if the flag has been dropped in the good camp and increase the concerned team point.

The Robot class have five arguments, the color of the robot, the vertical and horizontal position, the direction of the robot and a integer flag which represent the number of the flag carry by the robot. Flag argument is initialized to -1 by default which mean that the robot didn't carry a flag. The moveForward() fonction update the horizontal and vertical position of the robot in the next cell according to the orientation of the robot. The orientate() method orientate the robot in the orientation chosen. The refresh() method manage the display of the robot and clean() function clear the robot.

Finally, Command is a static class which manage all the action usable by the player.

About the data conception, the improvement axis would be to create an abstract class called Entity which would compose the Game class. The flag class and Robot Class would inherit from Entity. This will allow us to put the position and color argument in Entity, as well than the refresh method. This will certainly allow us to have a lighter code.

Work organization

This activity diagram represent our organization of work for the binomial.