22/04/2016

Airport Simulator

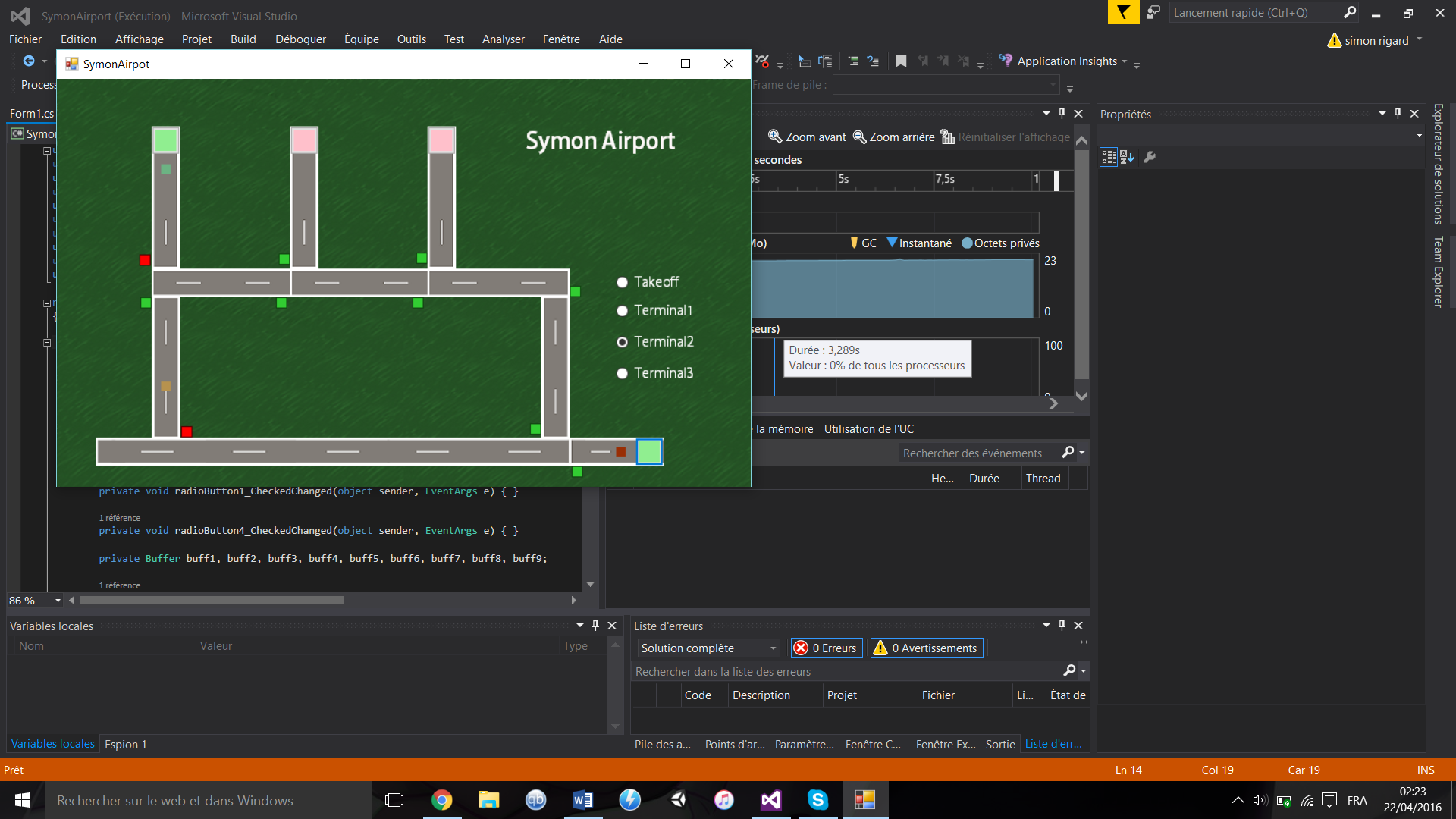
CONCURRENT PROGRAMMING IN C#

Simon Rigard

The Project

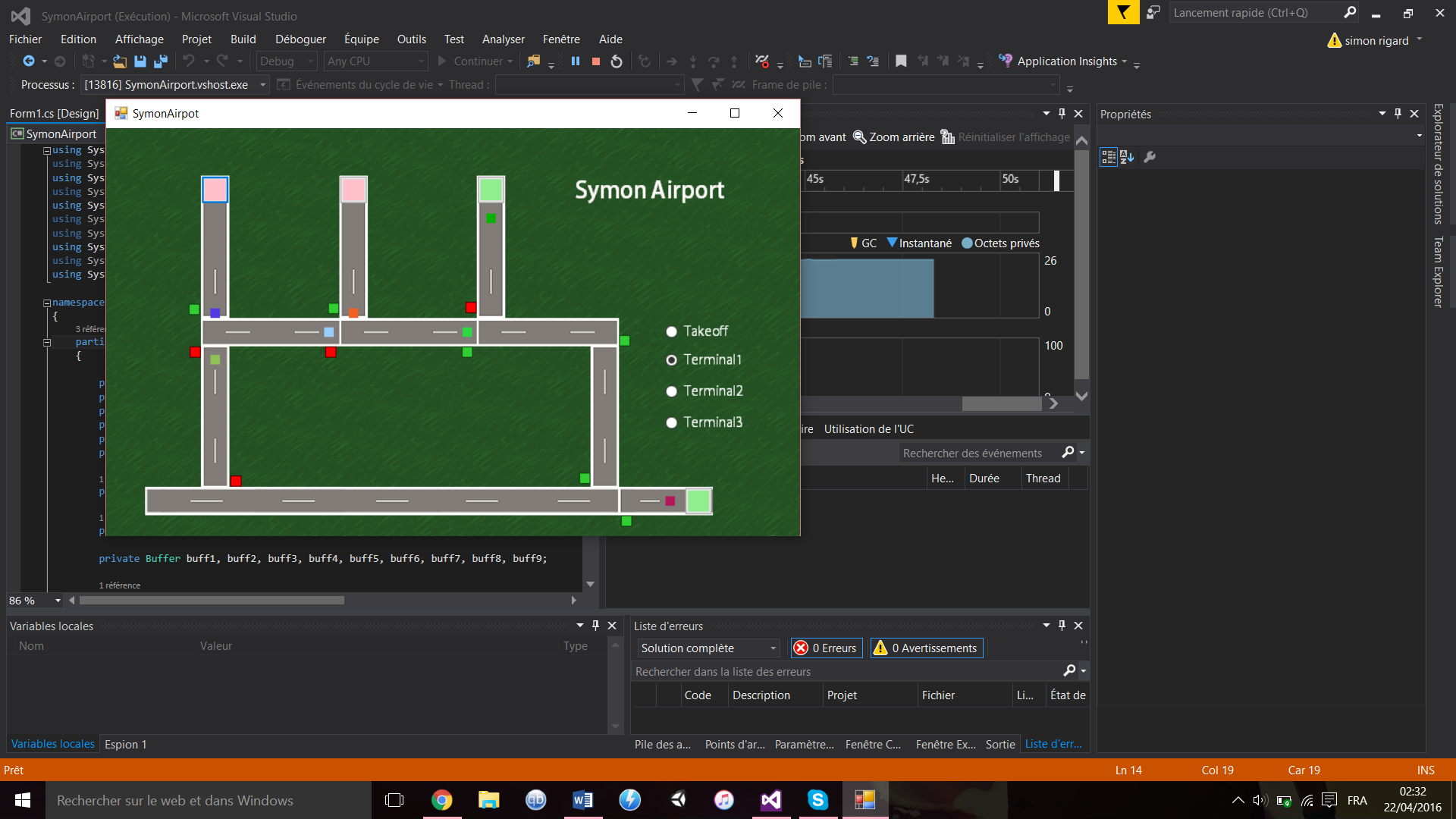
The goal of the project was to make a fully functional program which can simulate landings and take off inside an airport. This also includes 3 terminals and the program will avoid planes from colliding by making them wait when a section is occupied by another plane.

This is done with different Threads, semaphores and buffers. Each section (panel) will have a thread, a semaphore with a thread and a buffer with a thread. The program will be able to move all the plane at the same time as long as the buffer allows the next plane to pass on another panel.



The program allows the user to simulate the landing of a plane choosing his destination with the buttons on the right of the screen. The user can click the terminals to make the planes takeoff or he can click on the green button on the right bottom to make a plane land.

The buffers actually contain an object plane, which is instantiated from the class plane.cs That class gives the plane a color and a destination from 0 to 3 corresponding to takeoff and the terminals.



Traffic lights also indicates the plane if the way is open. If the light is red, that means another plane is currently on the next panel so the plane will wait until the way is clear.

A random color is also assigned to each plane.

Panel with terminals have a different class from the regular panels because they can host planes but also they can release them. That is why we have two classes WaitPanelThread and ButtonPanelThread.

Also every parameter like the distance the planes are traveling are in the call of the constructor in form1.cs

WaitPanelsThreads have their current, next and the terminal semaphore/buffer while ButtonPanelThreads have their current and next semaphore/buffer.

Threads, Semaphores and Buffers

Main Thread