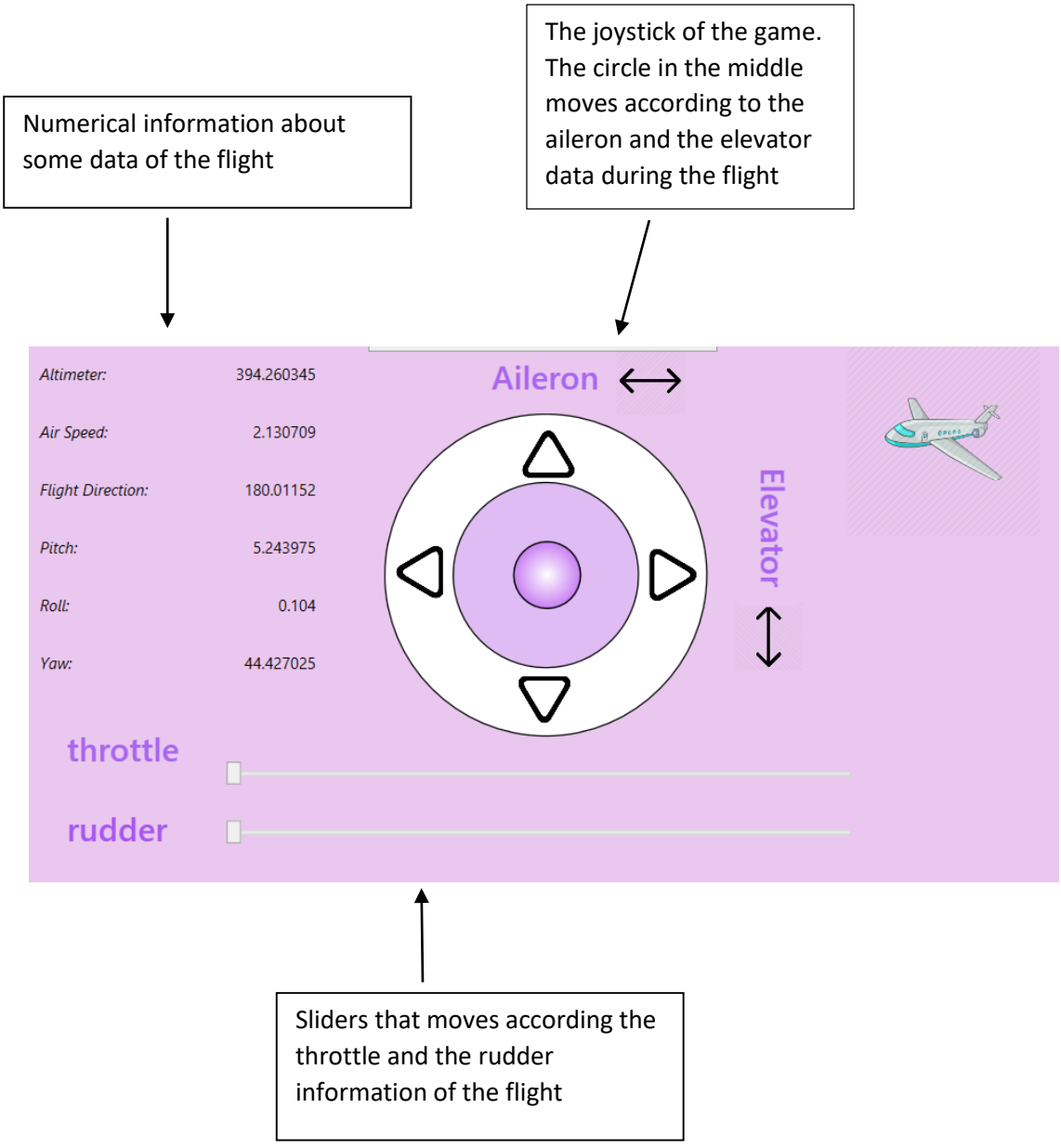
Wheel-

The wheel classes are also divided to: view, model and viewModel.

The wheel classes represent the steering wheel data for the whole flight, showing different flight data such as: elevator, rudder, speed and so forth. The joystick represents the elevator and the aileron information. The sliders represent the rudder and the throttle information.

As for the implementation of the MVVM:

There are fields and Properties for each data at the model and at the view model. When some data is updated at the CSV file, the setter of the model changes the matching private field, and notify the viewModel about the field that changed- using the NotifyPropertyChanged method.



The fields in the viewModel are with the same names, with a "VM" at the beginning. For example, in the model the data field will be: rudder for the private field and: Rudder for the public property, and at the viewModel the field will be: VM_Rudder (also public property). The value of the properties at the viewModel will be the one's of the model (the get method returns model.rudder).

Between the viewModel and the view of the Wheel there is Data Binding. When a property at the viewModel changes, it also appears at the view and changes the right element at the wheel screen.

The model of the wheel can receive changes from the user and update the correct element's values in the viewModel and view of the wheel, so they will display the most recent data as the user's request.