

```
1  //-----
2  //      Title:MASTER DEGREE THESIS by ANTONIO SCAZZI
3  //
4  //  Description:header file with all global variables
5  //-----
6  #pragma once
7  #include <windows.h>
8  #include <tchar.h>
9  #include <stdio.h>
10 #include <strsafe.h>
11 #include <vector>
12 #include <stdlib.h>
13 #include <string.h>
14 #include <cmath>
15 #include <fstream>
16 #include <ctime>
17 #include <cstring>
18 #include <math.h>
19 #include "SimConnect.h"
20 #include <iostream>
21
22
23 //define the event group
24 enum GROUP_ID {
25     GROUP_0, GROUP_1, GROUP_2
26 };
27 //define the event id
28 enum EVENT_ID {
29     EVENT_SIM_PAUSED, EVENT_SIM_UNPAUSED, EVENT_SIM_START, EVENT_SIM_STOP, EVENT_THROTTLE_SET, EVENT_RUDDER_SET,
30     EVENT_AILERON_SET, EVENT_ELEVATOR_SET, EVENT_FLAPS_SET, EVENT_PARKING_BRAKES_SET, EVENT_LANDING_GEAR,
31     EVENT_0, EVENT_1, EVENT_2, EVENT_3, EVENT_4, EVENT_5, EVENT_6, EVENT_7, EVENT_TRANSPONDER, EVENT_PRINT_1,
32     EVENT_PRINT_2
33 };
34 };
```

```
31 //define the input id
32 enum INPUT_ID {
33     INPUT_0, INPUT_1, INPUT_2, INPUT_3, INPUT_4, INPUT_5
34 };
35 //define data structure od the sim data
36 enum DATA_DEFINE_ID {
37     DEFINITION_1, DEFINITION_2
38 };
39 //define how many data request we need
40 enum DATA_REQUEST_ID {
41     REQUEST_1, REQUEST_0, REQUEST_2, REQUEST_3
42 };
43 // define the data structure
44 struct ObjectDataStruct
45 {
46     char    title[256];
47     double simtime;
48     double altitude;
49     double latitude;
50     double longitude;
51     double pitch;
52     double bank;
53     double heading;
54     double velocity;
55     double velocityX;
56     double velocityY;
57     double velocityZ;
58     double elevator;
59     double rudder;
60     double aileron;
61     double throttle;
62     double transponderCode;
63     double alpha;
```

```
64     double beta;
65     double pitchrate;
66     double rollrate;
67     double yawrate;
68     double accelerationZ;
69     double accelerationX;
70     double accelerationY;
71     double windX;
72     double windY;
73     double windZ;
74 };
75
76 //flags
77 int flag_created = 0, flag_quit = 0, flag_throttle = 0, flag_flap = 0, flag_isrunning = 0, flag_parkingbrake = 0, ↗
    flag_initial = 0, flag_output = 0, flag_decollo = 0, flag_landgear = 0, flag_dronefound = 0, flag_partenzadrone = ↗
    0, flag_isincruise = 0, flag_stampa = 0, flag_initialgroundcheck = 0, flag_roll = 0, flag_pitch = 0, ↗
    flag_comandoverride = 0, flag_test=0, flag_timer=0;
78 //definisco pi greco
79 double PI = 3.14159265358979323846;
80
81 //variabili da pulire
82 double Vz;
83 double heading;
84 double initial_latitude, initial_longitude;
85 double vs;
86
87 //variabili globali configurazione
88 double daticonfigs[3];
89 double quota_crociera = 0;
90 double heading_crociera = 0;
91 double initial_heading = 0;
92 double initial_transponder = 0;
93 double initial_simtime = 0, initial_simtime2=0;
```

```
94
95
96
97 //title of output file
98 char fileTitle[50];
99 char fileTitle2[50];
100
101 FILE* configs_file;
102 FILE* output_file;
103 FILE* output_file2;
104 errno_t err;
105 errno_t err2;
106
107 errno_t err0;
108
109 //other variables
110 ObjectDataStruct* V1;
111 ObjectDataStruct* V2;
112 ObjectDataStruct UserPlane,OtherPlane;
113 HANDLE hSimConnect = NULL;
114 HRESULT hr;
115 DWORD UserID;
116 DWORD ObjectID2;
117
118
119
120 const double WGS84_A = 6378137.0; // Earth's semi-major axis in meters
121 const double WGS84_B = 6356752.3; // Earth's semi-minor axis in meters
122
123 struct AircraftPosition
124 {
125     double Heading;
126 };
```

```
127 AircraftPosition initial;
128
129 double bankang = 0, pitchang = 0, altitude = 0, testheading = 0, comtestheading = 0,
    comvelocity=0,comDOWN=0,comEAST = 0,comNORD = 0;
130 ObjectDataStruct TestPlane;
131 std::vector<double> testNED(3, 0.0);
132
133 double deltatime = 0, previoustime = 0, testVel = 0;
```