

```
1  //-----
2  //      Title:MASTER DEGREE THESIS by ANTONIO SCAZZI
3  //
4  //  Description:Addon for Simconnect linked to Prepar3D
5  //      It has to be attached to a C2 Plane
6  //-----
7
8  #include "headers/globalvar.h"
9  #include "headers/readwrite.h"
10 #include "headers/dispatchfun.h"
11 #include "headers/autopilots.h"
12 #include "headers/miscellaneous.h"
13
14 //title of the addon
15 const char* TITLE_STRING = "C2 PLANE";
16
17 //main function of the addon
18 int __cdecl _tmain(int argc, _TCHAR* argv[])
19 {
20
21     // Apertura del simconnect
22     if (SUCCEEDED(SimConnect_Open(&hSimConnect, TITLE_STRING, NULL, 0, 0, 0)))
23     {
24         //connesso al simulatore
25         printf("Connected to Prepar3D\n");
26
27         //lettura file configs.txt
28         LetturaConfigs();
29
30         //richiesta test
31         printf("Vuoi eseguire un test dei controlli?? ");
32         flag_test = AskYesNo();
33         if (flag_test==1)
```

```
34     {
35         printf("Su cosa vuoi eseguire il test?\n");
36         flag_test = SelectTest();
37     }
38 }
39
40 //richiesta di creazione e apertura outputfile
41 printf("Vuoi salvare i dati della simulazione? ");
42 flag_stampa=AskYesNo();
43 if (flag_stampa == 1)
44 {
45     CreaFile();
46     ApriFile();
47 }
48
49 //ciclo principale dell'applicazione
50 while (0 == flag_quit)
51 {
52     //funzione che gestisce gli eventi del simulatore
53     SimConnect_CallDispatch(hSimConnect, MyDispatchProc, NULL);
54
55     //controllo se il simulatore è in pausa o meno
56     if (flag_isrunning == 1)
57     {
58         //request data on user
59         hr = SimConnect_RequestDataOnSimObjectType(hSimConnect, REQUEST_1, DEFINITION_1, 0,
60             SIMCONNECT_SIMOBJECT_TYPE_USER);
61
62         //request data on other plane
63         hr = SimConnect_RequestDataOnSimObjectType(hSimConnect, REQUEST_0, DEFINITION_1, 100000,
64             SIMCONNECT_SIMOBJECT_TYPE_AIRCRAFT);
65
66         //ceck if is on ground and correct the position
```

```
65         IsOnGround();
66
67         //controllo del carrello e retrazione
68         GearCheck();
69
70         // se al suolo autopilota C2, manovra di decollo
71         if (flag_initialgroundcheck == 1)
72         {
73             switch (flag_decollo)
74             {
75
76                 //inizializzazione
77                 case 1:
78                 {
79                     // manetta al 90%
80                     SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER,
81                                                     EVENT_THROTTLE_SET, 14743, SIMCONNECT_GROUP_PRIORITY_HIGHEST,
82                                                     SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
83
84                     // rimuovo il freno di stazionamento
85                     SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER,
86                                                     EVENT_PARKING_BRAKES_SET, 0, SIMCONNECT_GROUP_PRIORITY_HIGHEST,
87                                                     SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
88                     SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER,
89                                                     EVENT_RUDDER_SET, 0, SIMCONNECT_GROUP_PRIORITY_HIGHEST,
90                                                     SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
91                     SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER,
92                                                     EVENT_AILERON_SET, 0, SIMCONNECT_GROUP_PRIORITY_HIGHEST,
93                                                     SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
94                     SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER,
95                                                     EVENT_ELEVATOR_SET, 0, SIMCONNECT_GROUP_PRIORITY_HIGHEST,
96                                                     SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
97                 }
98             }
99         }
```

```
88          //passo alla prossima fase
89          flag_decollo = 2;
90
91          //invio il comando al drone per decollare
92          SendCommandTakeOff();
93      }
94      break;
95
96      //fase di rullaggio
97      case 2:
98      {
99          // manetta al 90%
100         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER,
                                     EVENT_THROTTLE_SET, 14743, SIMCONNECT_GROUP_PRIORITY_HIGHEST,
                                     SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
101
102         //attivo l'autopilota di heading con controllo di timone
103         double rudder = Headingtakeoff(initial_heading, UserPlane.heading);
104         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER,
                                     EVENT_RUDDER_SET, static_cast<DWORD>(round(rudder)),
                                     SIMCONNECT_GROUP_PRIORITY_HIGHEST, SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
105         //autopilota di controllo del pitch per tenere il velivolo a terra
106         double elevator = Pitchhold(0.8, UserPlane.pitch, UserPlane.pitchrate);
107         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER,
                                     EVENT_ELEVATOR_SET, static_cast<DWORD>(round(elevator)),
                                     SIMCONNECT_GROUP_PRIORITY_HIGHEST, SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
108
109         //comando per richiamata
110         if (UserPlane.velocity > 300)
111         {
112             flag_decollo = 3;
113             // setto l'alettone a 0 (non servirà se implemento l'ari)
114             SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER,
```

```
        EVENT_RUDDER_SET, 0, SIMCONNECT_GROUP_PRIORITY_HIGHEST,
        SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
115     }
116 }
117 break;
118
119 // fase di richiamata e salita rettilinea
120 case 3:
121 {
122     // manetta al 90%
123     SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER,
        EVENT_THROTTLE_SET, 14743, SIMCONNECT_GROUP_PRIORITY_HIGHEST,
        SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
124
125     //autopilota di mantenimento dell'heading con controllo di alettone
126     double aileron = Headinghold(initial_heading, UserPlane.heading, 10, UserPlane.bank,
        UserPlane.rollrate);
127     SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER,
        EVENT_AILERON_SET, static_cast<DWORD>(round(aileron)),
        SIMCONNECT_GROUP_PRIORITY_HIGHEST, SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
128
129     //autopilota di controllo della quota con controllo dell'equilibratore
130     double elevator = Pitchhold(-15, UserPlane.pitch, UserPlane.pitchrate);
131     SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER,
        EVENT_ELEVATOR_SET, static_cast<DWORD>(round(elevator)),
        SIMCONNECT_GROUP_PRIORITY_HIGHEST, SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
132
133     //comando per virata di allineamento alla crociera
134     if (UserPlane.altitude > quota_crociera / 3)
135     {
136         flag_decollo = 4;
137     }
138 }
```

```
139         break;
140     }
141 }
142
143 //controllo se è stato richiesto un test
144 if (flag_test == 0)
145 {
146     //fase di immissione in rotta di crociera e mantenimento della quota in crociera
147     if (flag_decollo == 4)
148     {
149         // manetta al 70%
150         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_THROTTLE_SET, ↗
            static_cast<DWORD>(round(16383 * .7)), SIMCONNECT_GROUP_PRIORITY_HIGHEST, ↗
            SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
151
152         //autopilota di controllo della quota con controllo dell'equilibratore
153         if (UserPlane.altitude < quota_crociera - 250 )
154         {
155             double elevator = Pitchhold(-15, UserPlane.pitch, UserPlane.pitchrate);
156             SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, ↗
                EVENT_ELEVATOR_SET, static_cast<DWORD>(round(elevator)), ↗
                SIMCONNECT_GROUP_PRIORITY_HIGHEST, SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
157         }
158         else if ( UserPlane.altitude > quota_crociera + 250)
159         {
160             double elevator = Pitchhold(15, UserPlane.pitch, UserPlane.pitchrate);
161             SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, ↗
                EVENT_ELEVATOR_SET, static_cast<DWORD>(round(elevator)), ↗
                SIMCONNECT_GROUP_PRIORITY_HIGHEST, SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
162         }
163         else
164         {
165             double elevator = Altitudehold(quota_crociera, UserPlane.altitude, 15, UserPlane.pitch, ↗
```

```
        UserPlane.pitchrate);
166      SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER,
        EVENT_ELEVATOR_SET, static_cast<DWORD>(round(elevator)),
        SIMCONNECT_GROUP_PRIORITY_HIGHEST, SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
167    }
168
169    //autopilota di mantenimento dell'heading con controllo di alettone
170    double aileron = Headinghold(heading_crociera, UserPlane.heading, 20, UserPlane.bank,
        UserPlane.rollrate);
171    SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_AILERON_SET,
        static_cast<DWORD>(round(aileron)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
        SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
172
173    if (UserPlane.altitude < (quota_crociera + 10.0) && UserPlane.altitude >(quota_crociera -
        10.0))
174    {
175        if (flag_isincruise == 0)
176        {
177            initial_simtime = UserPlane.simtime;
178            flag_isincruise = 1;
179        }
180        if (UserPlane.simtime - initial_simtime > 5)
181        {
182            flag_decollo = 5;
183            SendCommandReachC2();
184        }
185    }
186    else
187    {
188        flag_isincruise = 0;
189    }
190 }
191
```

```
192
193         //fase di crociera a velocità di crociera
194         if (flag_decollo == 5)
195         {
196             // imposto la velocità a 850 feet/sec
197             double throttle = Autothrottle(850, UserPlane.velocityZ, UserPlane.accelerationZ);
198             SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_THROTTLE_SET,
199                                     static_cast<DWORD>(round(throttle)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
200                                     SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
201
202             //autopilota di controllo della quota con controllo dell'equilibratore
203             if (UserPlane.altitude < quota_crociera + pitchang - 250)
204             {
205                 double elevator = Pitchhold(-15, UserPlane.pitch, UserPlane.pitchrate);
206                 SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER,
207                     EVENT_ELEVATOR_SET, static_cast<DWORD>(round(elevator)),
208                     SIMCONNECT_GROUP_PRIORITY_HIGHEST, SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
209             }
210             else if (UserPlane.altitude > quota_crociera + pitchang + 250)
211             {
212                 double elevator = Pitchhold(15, UserPlane.pitch, UserPlane.pitchrate);
213                 SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER,
214                     EVENT_ELEVATOR_SET, static_cast<DWORD>(round(elevator)),
215                     SIMCONNECT_GROUP_PRIORITY_HIGHEST, SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
216             }
217             else
218             {
219                 double elevator = Altitudehold(quota_crociera + pitchang, UserPlane.altitude, 15,
220                     UserPlane.pitch, UserPlane.pitchrate);
221                 SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER,
222                     EVENT_ELEVATOR_SET, static_cast<DWORD>(round(elevator)),
223                     SIMCONNECT_GROUP_PRIORITY_HIGHEST, SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
224             }
225         }
```

```
216
217         //autopilota di mantenimento dell'heading con controllo di alettone
218         double aileron = Headinghold(heading_crociera+bankang, UserPlane.heading, 20,
219                                     UserPlane.bank, UserPlane.rollrate);
220     }
221 }
222
223 //test pitchhold
224 else if(flag_test == 101)
225 {
226     double throttle = Autothrottle(800, UserPlane.velocityZ, UserPlane.accelerationZ);
227     SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_THROTTLE_SET,
228                                   static_cast<DWORD>(round(throttle)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
229                                   SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
230
231     double aileron = Bankhold(0, UserPlane.bank, UserPlane.rollrate);
232     SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_AILERON_SET,
233                                   static_cast<DWORD>(round(aileron)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
234                                   SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
235
236     if (flag_initial == 0 && UserPlane.simtime > 0)
237     {
238         initial_simtime = UserPlane.simtime;
239         flag_initial = 1;
240     }
241     if (UserPlane.simtime - initial_simtime < 5.0)
242     {
243         pitchang = 0;
244         double elevator = Pitchhold(pitchang, UserPlane.pitch, UserPlane.pitchrate);
245         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_ELEVATOR_SET,
246                                       static_cast<DWORD>(round(elevator)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
```

```
        SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);  
241     }  
242     else if (UserPlane.simtime - initial_simtime < 15.0)  
243     {  
244         pitchang = -20;  
245         double elevator = Pitchhold(pitchang, UserPlane.pitch, UserPlane.pitchrate);  
246         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_ELEVATOR_SET,   
            static_cast<DWORD>(round(elevator)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,   
            SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);  
247     }  
248     }  
249     else if (UserPlane.simtime - initial_simtime < 25.0)  
250     {  
251         pitchang = 0;  
252         double elevator = Pitchhold(pitchang, UserPlane.pitch, UserPlane.pitchrate);  
253         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_ELEVATOR_SET,   
            static_cast<DWORD>(round(elevator)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,   
            SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);  
255     }  
256     }  
257     else if (UserPlane.simtime - initial_simtime < 35.0)  
258     {  
259         pitchang = -10;  
260         double elevator = Pitchhold(pitchang, UserPlane.pitch, UserPlane.pitchrate);  
261         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_ELEVATOR_SET,   
            static_cast<DWORD>(round(elevator)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,   
            SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);  
262     }  
263     else if (UserPlane.simtime - initial_simtime < 45.0)  
264     {  
265         pitchang = 0;  
266         double elevator = Pitchhold(pitchang, UserPlane.pitch, UserPlane.pitchrate);
```

```
267         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_ELEVATOR_SET,
            static_cast<DWORD>(round(elevator)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
            SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);

268     }
269 }
270 else
271 {
272     exit(0);
273 }
274 }
275
276 //test per il bankhold
277 else if (flag_test == 102)
278 {
279     double throttle = Autothrottle(800, UserPlane.velocityZ, UserPlane.accelerationZ);
280     SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_THROTTLE_SET,
            static_cast<DWORD>(round(throttle)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
            SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);

281
282     double elevator = Altitudehold(6000, UserPlane.altitude, 20, UserPlane.pitch,
            UserPlane.pitchrate);
283     SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_ELEVATOR_SET,
            static_cast<DWORD>(round(elevator)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
            SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);

284
285     if (flag_initial == 0 && UserPlane.simtime > 0)
286     {
287         initial_simtime = UserPlane.simtime;
288         flag_initial = 1;
289     }
290     if (UserPlane.simtime - initial_simtime < 5.0)
291     {
292         bankang = 0;
```

```
293         double aileron = Bankhold(bankang, UserPlane.bank, UserPlane.rollrate);
294         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_AILERON_SET,
        static_cast<DWORD>(round(aileron)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
        SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
295     }
296     else if (UserPlane.simtime - initial_simtime < 15.0)
297     {
298         bankang = -20;
299         double aileron = Bankhold(bankang, UserPlane.bank, UserPlane.rollrate);
300         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_AILERON_SET,
        static_cast<DWORD>(round(aileron)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
        SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
301     }
302     else if (UserPlane.simtime - initial_simtime < 25.0)
303     {
304         bankang = 0;
305         double aileron = Bankhold(bankang, UserPlane.bank, UserPlane.rollrate);
306         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_AILERON_SET,
        static_cast<DWORD>(round(aileron)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
        SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
307     }
308     else if (UserPlane.simtime - initial_simtime < 35.0)
309     {
310         bankang = -10;
311         double aileron = Bankhold(bankang, UserPlane.bank, UserPlane.rollrate);
312         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_AILERON_SET,
        static_cast<DWORD>(round(aileron)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
        SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
313     }
314     else if (UserPlane.simtime - initial_simtime < 45.0)
315     {
316         bankang = 0;
317         double aileron = Bankhold(bankang, UserPlane.bank, UserPlane.rollrate);
```

```
318         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_AILERON_SET,
        static_cast<DWORD>(round(aileron)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
        SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
319     }
320     else
321     {
322         exit(0);
323     }
324 }
325
326 //test altitude hold
327 else if (flag_test == 103)
328 {
329     double throttle = Autothrottle(800, UserPlane.velocityZ, UserPlane.accelerationZ);
330     SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_THROTTLE_SET,
        static_cast<DWORD>(round(throttle)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
        SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
331
332     double aileron = Bankhold(0, UserPlane.bank, UserPlane.rollrate);
333     SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_AILERON_SET,
        static_cast<DWORD>(round(aileron)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
        SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
334
335     if (flag_initial == 0 && UserPlane.simtime > 0)
336     {
337         initial_simtime = UserPlane.simtime;
338         flag_initial = 1;
339     }
340     if (UserPlane.simtime - initial_simtime < 35.0)
341     {
342         altitude = 6000;
343         double elevator = Altitudehold(altitude, UserPlane.altitude, 15, UserPlane.pitch,
            UserPlane.pitchrate);
```

```
344      SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_ELEVATOR_SET, ↗
      static_cast<DWORD>(round(elevator)), SIMCONNECT_GROUP_PRIORITY_HIGHEST, ↗
      SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);

345
346  }
347  else if (UserPlane.simtime - initial_simtime < 70.0)
348  {
349      altitude = 6000+250/0.3048;
350
351      double elevator = Altitudehold(altitude, UserPlane.altitude, 15, UserPlane.pitch, ↗
      UserPlane.pitchrate);
352      SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_ELEVATOR_SET, ↗
      static_cast<DWORD>(round(elevator)), SIMCONNECT_GROUP_PRIORITY_HIGHEST, ↗
      SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);

353
354  }
355  else if (UserPlane.simtime - initial_simtime < 105.0)
356  {
357      altitude = 6000;
358
359      double elevator = Altitudehold(altitude, UserPlane.altitude, 15, UserPlane.pitch, ↗
      UserPlane.pitchrate);
360      SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_ELEVATOR_SET, ↗
      static_cast<DWORD>(round(elevator)), SIMCONNECT_GROUP_PRIORITY_HIGHEST, ↗
      SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);

361
362  }
363  else
364  {
365      exit(0);
366  }
367
368 }
```

```
369         //test heading hold
370         else if (flag_test == 104)
371         {
372             double throttle = Autothrottle(850, UserPlane.velocityZ, UserPlane.accelerationZ);
373             SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_THROTTLE_SET,
                 static_cast<DWORD>(round(throttle)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
                 SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
374
375             double elevator = Altitudehold(6000, UserPlane.altitude, 15, UserPlane.pitch,
                 UserPlane.pitchrate);
376             SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_ELEVATOR_SET,
                 static_cast<DWORD>(round(elevator)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
                 SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
377
378             if (flag_initial == 0 && UserPlane.simtime > 0)
379             {
380                 initial_simtime = UserPlane.simtime;
381                 flag_initial = 1;
382                 testheading = UserPlane.heading;
383             }
384             if (UserPlane.simtime - initial_simtime < 5.0)
385             {
386                 comtestheading = testheading;
387
388                 double aileron = Headinghold(comtestheading, UserPlane.heading, 35, UserPlane.bank,
                 UserPlane.rollrate);
389                 SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_AILERON_SET,
                 static_cast<DWORD>(round(aileron)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
                 SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
390             }
391             else if (UserPlane.simtime - initial_simtime < 35.0)
392             {
393                 comtestheading = testheading +20;
```

```
394
395
396         double aileron = Headinghold(comtestheading, UserPlane.heading, 35, UserPlane.bank,      ↗
            UserPlane.rollrate);
397         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_AILERON_SET,  ↗
            static_cast<DWORD>(round(aileron)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
            SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
398
399     }
400     else if (UserPlane.simtime - initial_simtime < 65.0)
401     {
402         comtestheading = testheading;
403
404         double aileron = Headinghold(comtestheading, UserPlane.heading, 35, UserPlane.bank,      ↗
            UserPlane.rollrate);
405         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_AILERON_SET,  ↗
            static_cast<DWORD>(round(aileron)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
            SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
406
407     }
408     else if (UserPlane.simtime - initial_simtime < 95.0)
409     {
410         comtestheading = testheading +10;
411
412         double aileron = Headinghold(comtestheading, UserPlane.heading, 35, UserPlane.bank,      ↗
            UserPlane.rollrate);
413         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_AILERON_SET,  ↗
            static_cast<DWORD>(round(aileron)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
            SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
414     }
415     else if (UserPlane.simtime - initial_simtime < 125.0)
416     {
417         comtestheading = testheading;
```

```
418
419         double aileron = Headinghold(comtestheading, UserPlane.heading, 35, UserPlane.bank,
420                                     UserPlane.rollrate);
421                                     SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_AILERON_SET,
422                                     static_cast<DWORD>(round(aileron)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
423                                     SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
424
425     }
426     else
427     {
428         exit(0);
429     }
430 }
431 //test autothrottle
432 else if (flag_test == 105)
433 {
434
435     double elevator = Altitudehold(6000, UserPlane.altitude, 15, UserPlane.pitch,
436                                     UserPlane.pitchrate);
437                                     SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_ELEVATOR_SET,
438                                     static_cast<DWORD>(round(elevator)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
439                                     SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
440     double aileron = Bankhold(0, UserPlane.bank, UserPlane.rollrate);
441     SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_AILERON_SET,
442                                     static_cast<DWORD>(round(aileron)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
443                                     SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
444
445     if (flag_initial == 0 && UserPlane.simtime > 0)
446     {
447         initial_simtime = UserPlane.simtime;
```

```
443         flag_initial = 1;
444
445     }
446     if (UserPlane.simtime - initial_simtime < 30.0)
447     {
448         comvelocity = 800;
449
450         double throttle = Autothrottle(comvelocity, UserPlane.velocityZ, UserPlane.accelerationZ);
451         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_THROTTLE_SET, ↗
            static_cast<DWORD>(round(throttle)), SIMCONNECT_GROUP_PRIORITY_HIGHEST, ↗
            SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
452     }
453     else if (UserPlane.simtime - initial_simtime < 70.0)
454     {
455         comvelocity = 900;
456
457         double throttle = Autothrottle(comvelocity, UserPlane.velocityZ, UserPlane.accelerationZ);
458         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_THROTTLE_SET, ↗
            static_cast<DWORD>(round(throttle)), SIMCONNECT_GROUP_PRIORITY_HIGHEST, ↗
            SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
459     }
460 }
461
462 else if (UserPlane.simtime - initial_simtime < 120.0)
463 {
464     comvelocity = 800;
465
466     double throttle = Autothrottle(comvelocity, UserPlane.velocityZ, UserPlane.accelerationZ);
467     SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_THROTTLE_SET, ↗
        static_cast<DWORD>(round(throttle)), SIMCONNECT_GROUP_PRIORITY_HIGHEST, ↗
        SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
468 }
469 }
```

```
470         else
471         {
472             exit(0);
473         }
474     }
475     //test NORD
476     else if (flag_test == 106)
477     {
478
479         if (UserPlane.longitude != 0)
480         {
481             if (flag_initial == 0 && UserPlane.simtime > 0)
482             {
483                 initial_simtime = UserPlane.simtime;
484                 flag_initial = 1;
485                 TestPlane = UserPlane;
486             }
487
488             if (UserPlane.simtime - previoustime < 1)
489             {
490                 deltatime = UserPlane.simtime - previoustime;
491                 testVel = testVel + (1 / (111000.0 * cos(TestPlane.latitude / 180 * PI)) * 800
492                 * .3048 * deltatime);
493
494
495                 double elevator = Altitudehold(6000, UserPlane.altitude, 15, UserPlane.pitch,
496                 UserPlane.pitchrate);
497                 SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER,
498                 EVENT_ELEVATOR_SET, static_cast<DWORD>(round(elevator)),
499                 SIMCONNECT_GROUP_PRIORITY_HIGHEST, SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
500                 double aileron = Bankhold(0, UserPlane.bank, UserPlane.rollrate);
501                 SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER,
```

```
EVENT_AILERON_SET, static_cast<DWORD>(round(aileron)),  
SIMCONNECT_GROUP_PRIORITY_HIGHEST, SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);  
499 if (UserPlane.simtime - initial_simtime < 210.0)  
500 {  
501     comNORD = 0;  
502     testNED = EcefToNEDPhi(TestPlane.latitude, TestPlane.longitude + testVel,  
TestPlane.altitude, UserPlane.latitude, UserPlane.longitude, UserPlane.altitude,  
TestPlane.heading);  
  
503  
504     double throttle = ForwardSeparation(testNED, comNORD, UserPlane.velocityZ,  
UserPlane.accelerationZ);  
505     SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER,  
EVENT_THROTTLE_SET, static_cast<DWORD>(round(throttle)),  
SIMCONNECT_GROUP_PRIORITY_HIGHEST, SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);  
  
506  
507 }  
508 else if (UserPlane.simtime - initial_simtime < 310.0)  
509 {  
510     comNORD = 50;  
511     testNED = EcefToNEDPhi(TestPlane.latitude, TestPlane.longitude + testVel,  
TestPlane.altitude, UserPlane.latitude, UserPlane.longitude, UserPlane.altitude,  
TestPlane.heading);  
  
512     double throttle = ForwardSeparation(testNED, comNORD, UserPlane.velocityZ,  
UserPlane.accelerationZ);  
513     SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER,  
EVENT_THROTTLE_SET, static_cast<DWORD>(round(throttle)),  
SIMCONNECT_GROUP_PRIORITY_HIGHEST, SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);  
  
514  
515 }  
516 else if (UserPlane.simtime - initial_simtime < 410.0)  
517 {  
518     comNORD = 0;  
519     testNED = EcefToNEDPhi(TestPlane.latitude, TestPlane.longitude + testVel,
```

```
TestPlane.altitude, UserPlane.latitude, UserPlane.longitude, UserPlane.altitude,
TestPlane.heading);
520     double throttle = ForwardSeparation(testNED, comNORD, UserPlane.velocityZ,
UserPlane.accelerationZ);
521     SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER,
EVENT_THROTTLE_SET, static_cast<DWORD>(round(throttle)),
SIMCONNECT_GROUP_PRIORITY_HIGHEST, SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
522
523     }
524     else
525     {
526         exit(0);
527     }
528     printf("com%.5f\ttest %.5f\tuser %.5f\tNORD%.5f\n", comNORD, TestPlane.longitude +
testVel, UserPlane.longitude, testNED[0]);
529     previoustime = UserPlane.simtime;
530 }
531
532
533
534 }
535 //test EAST
536 else if (flag_test == 107)
537 {
538     if (flag_initial == 0 && UserPlane.simtime > 0)
539     {
540         initial_simtime = UserPlane.simtime;
541         flag_initial = 1;
542         TestPlane = UserPlane;
543     }
544     double throttle = Autothrottle(800, UserPlane.velocityZ, UserPlane.accelerationZ);
545     SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_THROTTLE_SET,
static_cast<DWORD>(round(throttle)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
```

```
        SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
546     double elevator = Altitudehold(6000, UserPlane.altitude, 15, UserPlane.pitch,
        UserPlane.pitchrate);
547     SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_ELEVATOR_SET,
        static_cast<DWORD>(round(elevator)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
        SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
548
549     if (UserPlane.simtime - initial_simtime < 5.0)
550     {
551
552         comEAST = 0;
553         testNED = EcefToNEDPhi(TestPlane.latitude, TestPlane.longitude, TestPlane.altitude,
        UserPlane.latitude, UserPlane.longitude, UserPlane.altitude, TestPlane.heading);
554         double aileron = LateralSeparation(testNED, comEAST, 35, UserPlane.bank,
        UserPlane.rollrate);
555         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_AILERON_SET,
        static_cast<DWORD>(round(aileron)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
        SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
556
557     }
558     else if (UserPlane.simtime - initial_simtime < 35.0)
559     {
560         comEAST = 100;
561         testNED = EcefToNEDPhi(TestPlane.latitude, TestPlane.longitude, TestPlane.altitude,
        UserPlane.latitude, UserPlane.longitude, UserPlane.altitude, TestPlane.heading);
562         double aileron = LateralSeparation(testNED, comEAST, 35, UserPlane.bank,
        UserPlane.rollrate);
563         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_AILERON_SET,
        static_cast<DWORD>(round(aileron)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
        SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
564
565     }
566     else if (UserPlane.simtime - initial_simtime < 65.0)
```

```
567     {
568         comEAST = 0;
569         testNED = EcefToNEDPhi(TestPlane.latitude, TestPlane.longitude, TestPlane.altitude,
570                                UserPlane.latitude, UserPlane.longitude, UserPlane.altitude, TestPlane.heading);
571         double aileron = LateralSeparation(testNED, comEAST, 35, UserPlane.bank,
572                                            UserPlane.rollrate);
573         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_AILERON_SET,
574                                       static_cast<DWORD>(round(aileron)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
575                                       SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
576     }
577     else if (UserPlane.simtime - initial_simtime < 95.0)
578     {
579         comEAST = 50;
580         testNED = EcefToNEDPhi(TestPlane.latitude, TestPlane.longitude, TestPlane.altitude,
581                                UserPlane.latitude, UserPlane.longitude, UserPlane.altitude, TestPlane.heading);
582         double aileron = LateralSeparation(testNED, comEAST, 35, UserPlane.bank,
583                                            UserPlane.rollrate);
584         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_AILERON_SET,
585                                       static_cast<DWORD>(round(aileron)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
586                                       SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
587     }
588     else if (UserPlane.simtime - initial_simtime < 125.0)
589     {
590         comEAST = 0;
591         testNED = EcefToNEDPhi(TestPlane.latitude + comEAST, TestPlane.longitude,
592                                TestPlane.altitude, UserPlane.latitude, UserPlane.longitude, UserPlane.altitude,
593                                TestPlane.heading);
594         double aileron = LateralSeparation(testNED, comEAST, 35, UserPlane.bank,
595                                            UserPlane.rollrate);
596         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_AILERON_SET,
597                                       static_cast<DWORD>(round(aileron)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
```

```
        SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);  
588  
589    }  
590    else  
591    {  
592        exit(0);  
593    }  
594  
595  
596    }  
597    //test DOWN  
598    else if (flag_test == 108)  
599    {  
600  
601        double throttle = Autothrottle(800, UserPlane.velocityZ, UserPlane.accelerationZ);  
602        SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_THROTTLE_SET, ↗  
        static_cast<DWORD>(round(throttle)), SIMCONNECT_GROUP_PRIORITY_HIGHEST, ↗  
        SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);  
603  
604        double aileron = Bankhold(0, UserPlane.bank, UserPlane.rollrate);  
605        SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_AILERON_SET, ↗  
        static_cast<DWORD>(round(aileron)), SIMCONNECT_GROUP_PRIORITY_HIGHEST, ↗  
        SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);  
606  
607  
608        if (flag_initial == 0 && UserPlane.simtime > 0)  
609        {  
610            initial_simtime = UserPlane.simtime;  
611            flag_initial = 1;  
612            TestPlane = UserPlane;  
613        }  
614        if (UserPlane.simtime - initial_simtime < 35.0)  
615        {
```

```
616
617         comDOWN = 0;
618         testNED= EcefToNEDPhi(TestPlane.latitude, TestPlane.longitude,
        TestPlane.altitude,UserPlane.latitude,UserPlane.longitude,UserPlane.altitude,TestPlane.heading);
619         double elevator = VerticalSeparation(testNED, comDOWN, 20,
        UserPlane.pitch,UserPlane.pitchrate);
620         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_ELEVATOR_SET,
        static_cast<DWORD>(round(elevator)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
        SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
621
622     }
623     else if (UserPlane.simtime - initial_simtime < 70.0)
624     {
625         comDOWN = 150;
626         testNED = EcefToNEDPhi(TestPlane.latitude, TestPlane.longitude, TestPlane.altitude,
        UserPlane.latitude, UserPlane.longitude, UserPlane.altitude, TestPlane.heading);
627         double elevator = VerticalSeparation(testNED, comDOWN, 20, UserPlane.pitch,
        UserPlane.pitchrate);
628         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_ELEVATOR_SET,
        static_cast<DWORD>(round(elevator)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
        SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);
629
630     }
631     else if (UserPlane.simtime - initial_simtime < 105.0)
632     {
633         comDOWN = 0;
634         testNED = EcefToNEDPhi(TestPlane.latitude, TestPlane.longitude, TestPlane.altitude,
        UserPlane.latitude, UserPlane.longitude, UserPlane.altitude, TestPlane.heading);
635         double elevator = VerticalSeparation(testNED, comDOWN, 20, UserPlane.pitch,
        UserPlane.pitchrate);
636         SimConnect_TransmitClientEvent(hSimConnect, SIMCONNECT_OBJECT_ID_USER, EVENT_ELEVATOR_SET,
        static_cast<DWORD>(round(elevator)), SIMCONNECT_GROUP_PRIORITY_HIGHEST,
```

```
        SIMCONNECT_EVENT_FLAG_GROUPID_IS_PRIORITY);  
  
637  
638        }  
639        else  
640        {  
641            exit(0);  
642        }  
643  
644  
645  
646    }  
647    //calcolo ned se trovo drone per scriverlo su file  
648    if (flag_dronefound == 1)  
649    {  
650        testNED = EcefToNEDPhi(UserPlane.latitude, UserPlane.longitude, UserPlane.altitude,  
                                OtherPlane.latitude, OtherPlane.longitude, OtherPlane.altitude, UserPlane.heading);  
651    }  
652  
653    //comandi manuali  
654    if (flag_decollo == 1001)  
655    {  
656  
657    }  
658  
659    //scrivo su file di output  
660    if (flag_stampa == 1)  
661    {  
662        StampaFile();  
663    }  
664 }  
665  
666 //definisco la frequenza dell'add-on  
667 Sleep(50);
```

```
668     }
669
670     //chiusura applicazione e file output
671     hr = SimConnect_Close(hSimConnect);
672     if (flag_stampa == 1)
673     {
674         ChiudiFile();
675     }
676     printf("\nDisconnected from Prepar3D ");
677     system("pause");
678 }
679 else
680 {
681     printf("\nFailed to Connect to Prepar3D ");
682     system("pause");
683 }
684 system("pause");
685 return 0;
686 }
687
```