

נספח – התממשקות עם FlightGear

FlightGear הוא סימולטור בקוד פתוח שמאפשר גם מחקר. אולם, אנו לא נשתמש בתכונות הסימולטביזציה שלו, אלא נשתמש בו "כמקרן" עבור תצוגה של נתוני טיסה.

ניתן להוריד את הסימולטור מ <https://www.flightgear.org/> לכל מערכות ההפעלה הנפוצות.

הגרסאות האחרונות הפכו קצת כבדות ואם המחשב שלכם לא מצליח להריץ את הסימולטור חלק גם אחרי שהורדתם רזולוציה ורינדורים מיותרים אז שווה לשקול להוריד גרסאות ישנות וקלות יותר כמו למשל של 2018.

FlightGear יכול לפתוח כל אמצעי קלט-פלט (למשל http, socket tcp / udp, קבצים, פורטים סיריאליים וכו') ולכתוב או לקרוא נתוני טיסה. כדי להגדיר לו באלו נתונים מדובר יש להזין לו קובץ XML לתיקיה בה התקנתם את FlightGear, ובפנים ל 0/data/protocol

דוגמה עבור משתמשי חלונות:

C:\Program Files\FlightGear 2020.3.6\data\Protocol

אנו נעתיק לשם את הקובץ playback_small.xml שהכנתי עבורכם. הביטו בקובץ, יש 2 תגיות עיקריות <input> ו <output> בתוכן יש את הגדרת הנתונים שנרצה לאסוף \ להכתיב לסימולטור בהתאמה.

למשל:

```
<chunk>
  <name>aileron</name>
  <type>float</type>
  <format>%f</format>
  <node>/controls/flight/aileron[0]</node>
</chunk>
```

בכל chunk יש הגדרה של משתנה ששמו נמצא בתגית name מסוג המוגדר בתגית type וקישור למיקום של הערך בסימולטור בתגית node.

בהגדרות שנזין לסימולטור נוכל לומר לו לאסוף את הנתונים המוגדרים ב playback_small.xml ולשמור אותם למשל בקובץ csv, לשדר אותן באמצעות socket וכו'.

את ההגדרות מזינים ב command line או במסך ה GUI שנפתח עם ההרצה בלשונית ה settings.



למשל, עם ההגדרה הזו הקלטנו טיסה לתוך קובץ csv:

```
--generic=file,out,10,c:\flights\reg_flight.out,playback_small
```

שמשמעותה ליצור את הקובץ c:\flights\regflight.csv ולשפוך לשם את הנתונים שמוגדרים ב playback_small.xml בקצב של 10 פעמים בשנייה.

עם ההגדרה הבאה לדוגמה הסימולטור יכול לשדר כלקוח את אותם הנתונים דרך tcp socket לשרת לוקאלי שמאזין בפורט 6400:

```
--generic=socket,out,10,127.0.0.1,6400,tcp,playback_small
```

כך למשל אפליקציה יכולה לקבל נתוני טיסה שמתרחשת בזמן אמת בסימולטור.

הסימולטור יכול גם בעצמו לפתוח שרתים מסוגים שונים שאנו נתחבר אליהם כלקוחות. למשל:

```
--telnet=socket,in,10,127.0.0.1,5402,tcp
```

הגדרה זו תאפשר לנו להתחבר באמצעות טרמינל של telnet client לסימולטור הטיסה ולשוטט במבנה התיקיות שייצר עבור ההגדרות שלו.

כך למשל נוכל לכתוב

```
get /controls/flight/aileron[0]
```

ולקרוא את הערך של ה aileron, או למשל

```
set /controls/flight/aileron[0] -0.8
```

וממש להזין לו ערך של -0.8 (מה שיגרום לגלגול המטוס)

אם נזין בהגדרות לסימולטור למשל --httpd=8080 אז הסימולטור יפתח שרת HTTP על פורט 8080 שנוכל להתחבר אליו באמצעות הדפדפן בכתובת <http://localhost:8080/> ולמצוא שם אפליקציית רשת שהם יצרו לעזר. גם שם תוכלו לדפדף בין כל המשתנים שיש לסימולטור.

לעומת זאת כשנרצה להשתמש בסימולטור כמקור, נצטרך להגדיר לו לא להפעיל את התכונות הסימולטיביות שלו וכן להגדיר לו מקור שממנו נכתיב לו את הנתונים.

למשל:

```
--generic=socket,in,10,127.0.0.1,5400,tcp,playback_small
```

```
--fdm=null
```

הגדרנו לסימולטור לפתוח הפעם שרת TCP בפורט 5400 ובאמצעות socket לקוח יתחבר אליו ויכתיב לו נתונים לפי הסדר המוגדר ב playback_small.xml בקצב של 10 פעמים בשנייה. ההגדרה --fdm=null אומרת לו לא להפעיל סימולציה.

הסימולטור בתגובה יקרא (in) מהלקוח שורה אחר שורה של ערכים המופרדים בפסיק, ועבור כל ערך הוא אוטומטית יזין אותו למשתנה המתאים בסימולטור בזכות ההגדרה של playback_small.xml, ובתגובה לכך אנו נראה "אנימציה" של הטיסה.

ניתן בקלות לכתוב לקוח שקורא קובץ CSV (שהערכים בעמודות שלו כמובן תואמים את סדר המשתנים ב playback_small.xml) ומשדר את הנתונים הללו לסימולטור.

הנה למשל דוגמה ב Java שקוראת שורה שורה מהקובץ reg_flight.csv ומשדרת אותו ל FlightGear שנפתח עם ההגדרה לעיל:

```
Socket fg=new Socket("localhost", 5400);
BufferedReader in=new BufferedReader(new FileReader("reg_flight.csv"));
PrintWriter out=new PrintWriter(fg.getOutputStream());
String line;
while((line=in.readLine())!=null) {
    out.println(line);
    out.flush();
    Thread.sleep(100);
}
out.close();
in.close();
fg.close();
```

כמובן ה sleep(100) דואג שהנתונים ישודרו בקצב של 10 פעמים בשנייה. תוכלו לשחק עם הפרמטר הזה כדי להקריין את הטיסה מהר יותר או לאט יותר. כמובן עדיף לטעון את הקובץ פעם אחת, ולשדר את הנתונים לסימולטור מהזיכרון ולא מהדיסק.