

לאחר שהסימולטור פעל, פיתחו את הדפדפן בכתובת <http://localhost:8080> ותוכלו לראות את האפליקציה ה-web-ית שבאה עם הסימולטור.

כעת, תפתחו telnet client בשורת הפקודה, על local host ופורט 5402 (בהתאם להגדרות שראינו בסימולטור)

הערה: בלינוקס ניתן ישר לפתוח מהטרמינל, בחלונות יש לוודא שהתקנתם telnet client, מי שלא התקין צריך עזרה יכול לחפש בגוגל "install telnet client windows 10 command line" ולמצוא מדריכים.

כאמור, ב CMD של חלונות הקלידו telnet 127.0.0.1 5402 ובעצם התחברתם כלקוח לשרת שפתח הסימולטור על המחשב שלכם. הממשק הזה נוח מאד מכיוון שהוא בנוי בצורה של file system (קבצים ותיקיות). כתבו ls כדי לראות את "התיקיות והקבצים" במיקום הנוכחי שלכם. תוכלו להיכנס לתיקיה באמצעות הפקודה cd (שזה change directory) למשל cd controls. תטיילו קצת בין אינספור ההגדרות השונות כדי לקבל תחושה מה יש שם.

כעת הביטו בו זמנית בסימולטור וב telnet. בסימולטור אתם יכולים לשנות זוויות צפייה ע"י לחיצה על V ואף משחק עם העכבר. הביטו על המטוס ממבט אחורי. כעת בואו נזין פקודה להזזת מייצב הכיוון של המטוס (rudder):

לא משנה היכן אתם ב telnet, כתבו:

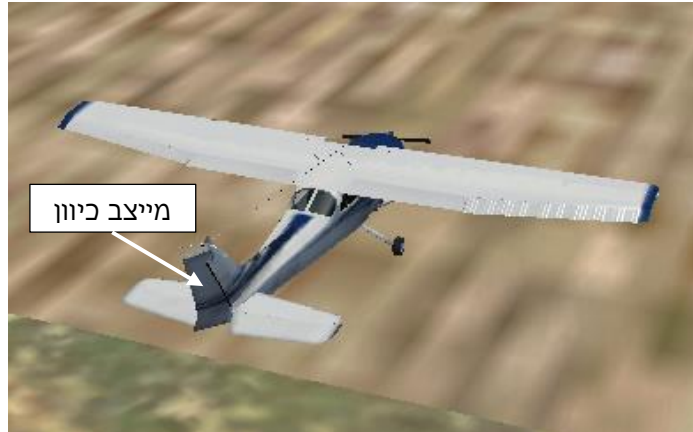
```
set controls/flight/rudder 1
```

ראו בסימולטור כיצד מייצב הכיוון זז
עד הסוף ימינה.

באופן דומה כתבו

```
set controls/flight/rudder -1
```

וראו כיצד מייצב הכיוון של המטוס זז
עד הסוף שמאלה. כל ערך בין 1- ל 1
יסובב את מייצב הכיוון בהתאמה.



חוץ מלתת פקודות להגאים של המטוס, ניתן גם לדגום את הערכים השונים שנמדדו ע"י מכשירי הטיסה,
כמו כיוון, מהירות, גובה וכו'. כתבו ב telnet את השורה הבאה:

```
get /instrumentation/altimeter/indicated-altitude-ft
```

ניתן לראות בתגובת השרת את הערך של הגובה הנוכחי כפי שנמדד במכשיר טיסה שנקרא altimeter.

אבל, את הערכים של הטיסה אנו נדגום בצורה מחוכמת יותר. יחד עם ההפעלה של הסימולטור נוסיף את ההגדרה:

```
--generic=socket,out,10,127.0.0.1,5400,tcp,generic_small
```

משמעותה היא שהפעם הסימולטור יתחבר כלקוח לשרת (שאנו נבנה) באמצעות socket, לצורך פלט
(out) בתדירות של 10 פעמים בשנייה, על ה local host בפורט 5400 מעל tcp/ip. הערכים שנדגום
מוגדרים בקובץ שנקרא generic_small.xml המצורף כנספח לפרויקט. את הקובץ הזה עליכם לשתול
במיקום בו התקנתם את FlightGear בתיקייה data/protocol. פתחו את קובץ ה XML כדי להתרשם
אלו נתונים נדגום.

עם הגדרה זו נצטרך לפתוח את השרת שלנו **לפני** שנפתח את הסימולטור כדי שהוא יוכל להתחבר
אלינו כלקוח. הסימולטור ישלח 10 פעמים בשנייה את הערכים שדגם מופרדים בפסיק (בדיוק כמו ב
CSV) ולפי הסדר שהוגדרו ב XML. בהמשך אבן הדרך תכתבו שרת קטנטן שיאזין לערכים האלו.

מי שרוצה לקבל עוד קצת רקע לגבי שליטה במטוס יוכל לקרוא (בוויקפדיה למשל) אודות

- ההגאים: aileron, elevators, rudder (מייצב כיוון, מייצב גובה, מאזנות בהתאמה)
- וכיצד הם משפיעים בעת טיסה על ה roll, pitch, yaw בהתאמה.
 - ובעברית סבסוב, עלרוד, גלגול בהתאמה.

אין כל חובה לדעת להטיס מטוס בפרויקט זה, אך מעט רקע בהחלט יכול לעזור.

מפרש קוד לשליטה בטיסה

כאמור, ברצוננו לכתוב מפרש לשפת תכנות חדשה שמטרתה להטיס את המטוס שבסימולטור. נתחיל מלהגדיר קוד לדוגמא שמטרתו לגרום למטוס להמריא בצורה ישרה. בהמשך נסביר את המשמעות של שורות אלו וכיצד נכתוב מפרש שיריץ אותן.

קוד לדוגמא:

```
1. openDataServer 5400 10
2. connect 127.0.0.1 5402
3. var breaks = bind "/controls/flight/speedbrake"
4. var throttle = bind "/controls/engines/engine/throttle"
5. var heading = bind "/instrumentation/heading-indicator/offset-deg"
6. var airspeed = bind "/instrumentation/airspeed-indicator/indicated-speed-kt"
7. var roll = bind "/instrumentation/attitude-indicator/indicated-roll-deg"
8. var pitch = bind "/instrumentation/attitude-indicator/internal-pitch-deg"
9. var rudder = bind "/controls/flight/rudder"
10. var aileron = bind "/controls/flight/aileron"
11. var elevator = bind "/controls/flight/elevator"
12. var alt = bind "/instrumentation/altimeter/indicated-altitude-ft"
13. breaks = 0
14. throttle = 1
15. var h0 = heading
16. while alt < 1000 {
17.   rudder = (h0 - heading)/20
18.   aileron = - roll / 70
19.   elevator = pitch / 50
20.   print alt
21.   sleep 250
22. }
23. print "done"
```

הסבר:

נרצה ששורה 1 תגרום לפתיחה של ת'רד ברקע, שפותח **שרת** המאזין על פורט 5400 וקורא שורה שורה בקצב של 10 פעמים בשנייה. את הערכים הנדגמים יש לאכסן במבנה נתונים שבאמצעותו נוכל לשלוף ב (1) את הערך העדכני של משתנה כלשהו שבחרנו.

נרצה ששורה 2 תתחבר כלקוח לשרת שנמצא ב 127.0.0.1 ומאזין על פורט 5402.

בשורות 3-12 אנו מגדירים את המשתנים שאיתם נעבוד במהלך התוכנית. המשמעות של bind היא כריכה בין ערך המשתנה בתוכנית שלנו לבין מיקומו בסימולטור הטיסה.

כך למשל בשורה 14 כשביצענו השמה `throttle = 1` שלחנו למעשה לסימולטור את הפקודה:

```
set /controls/engines/engine/throttle 1
```

וגרמנו למצערת להיפתח עד הסוף (כוח מלא למנוע כדי שהמטוס יתחיל לנוע)

נשים לב שהכריכה היא דו-כיוונית, למשל המשתנה `heading` כרוך למכשיר `heading indicator` שנמצא ב:

```
/instrumentation/heading-indicator/offset-deg
```

בכל פעם בתוכנית שנשתמש ב `heading` נקבל את הערך הנוכחי של הכיוון מסימולטור הטיסה. למשל, בשורה 15 המשתנה `h0` מקבל את ערכו הנוכחי של `heading` - כיוון הטיסה, כפי שנגדם באותו הרגע ע"י הת'רד שרץ ברקע שפתחנו בשורה 1.