

מטלה מספר 2 – תכנות מתקדם 2

חלק א'- היכרות עם סימולטור הטיסה

כל מי שלא עשה את הפרוייקט של תכנות מתקדם 1 בסמסטר שעבר, אין חשש (אנחנו נשתמש רק בסימולטור טיסה ולא בפרוייקט מסמסטר שעבר)

מצ"ב, דף הסבר הקרוי היכרות עם סימולטור הטיסה המציג את מערכת סימולטור הטיסה, וכיצד ניתן להפעילה.

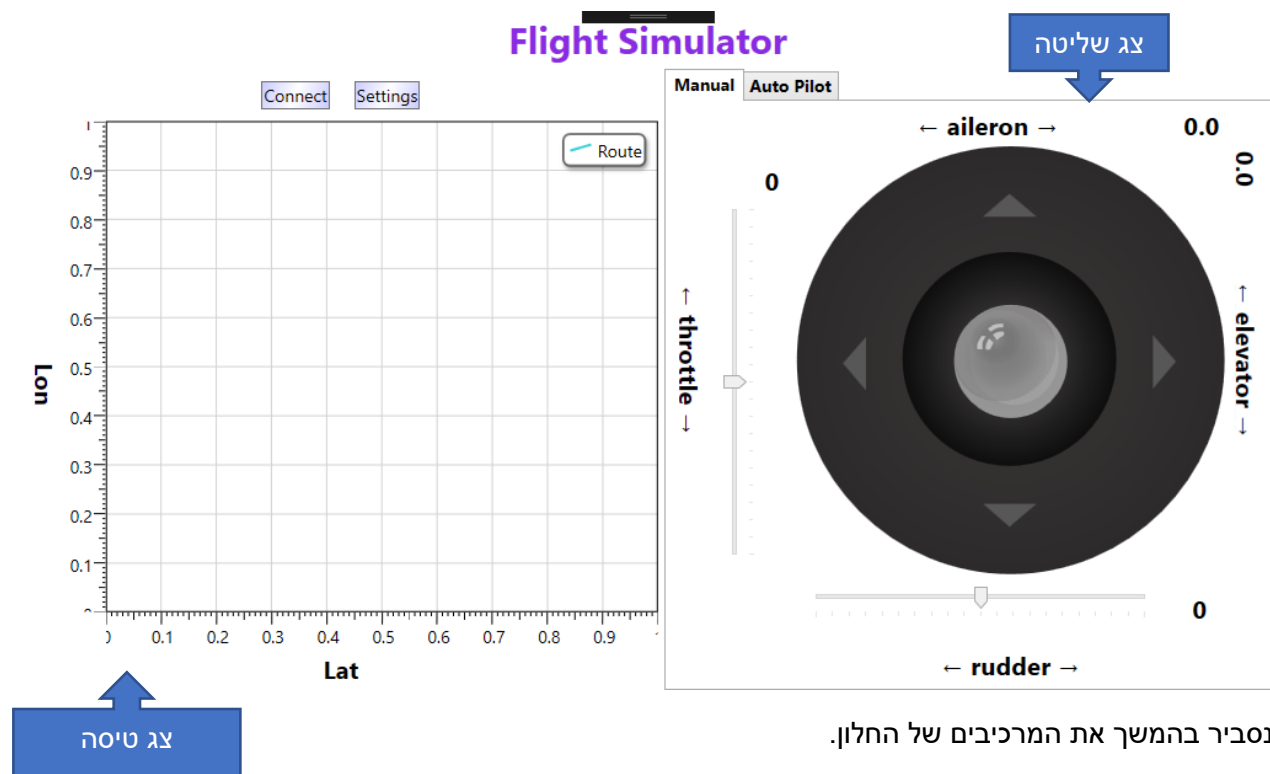
שימו לב!

- על מנת להשתמש בפקד של הצגת מסלול טיסה **חובה עליכם להוסיף את ה-dll הבא:** `DynamicDataDisplay.dll`, הנמצא בתקיית `dll`, אל הפרוייקט שלכם בתור Reference.
- השימוש ב-Routed Events עבור כפתורים הינו **אסור!** חובה עליכם להשתמש במנגנון ה-Command אשר נלמד בתרגולים! (משמע בקוד מאחורי הקלעים של ה-VIEW אסור להגדיר events!)
- חובה להשתמש בעקרון ה-MVVM שנלמד בהרצאה ובתרגולים, כאשר כל אחד מהצגים (שליטה אוטומטית, שליטה ידנית וטיסה) יוגדרו כ-UserControl **כאשר לכל אחד יהיה ViewModel מתאים משלו.**

חלק ב'- הסבר כללי על המערכת

באבן דרך זו (אבן דרך מספר 3) אנחנו נבנה ממשק Gui (אפליקציית WPF) אשר תאפשר לנו לשלוט על המטוס שלנו.

להלן חלון לדוגמא אשר מתאר את מבנה האפליקציה:



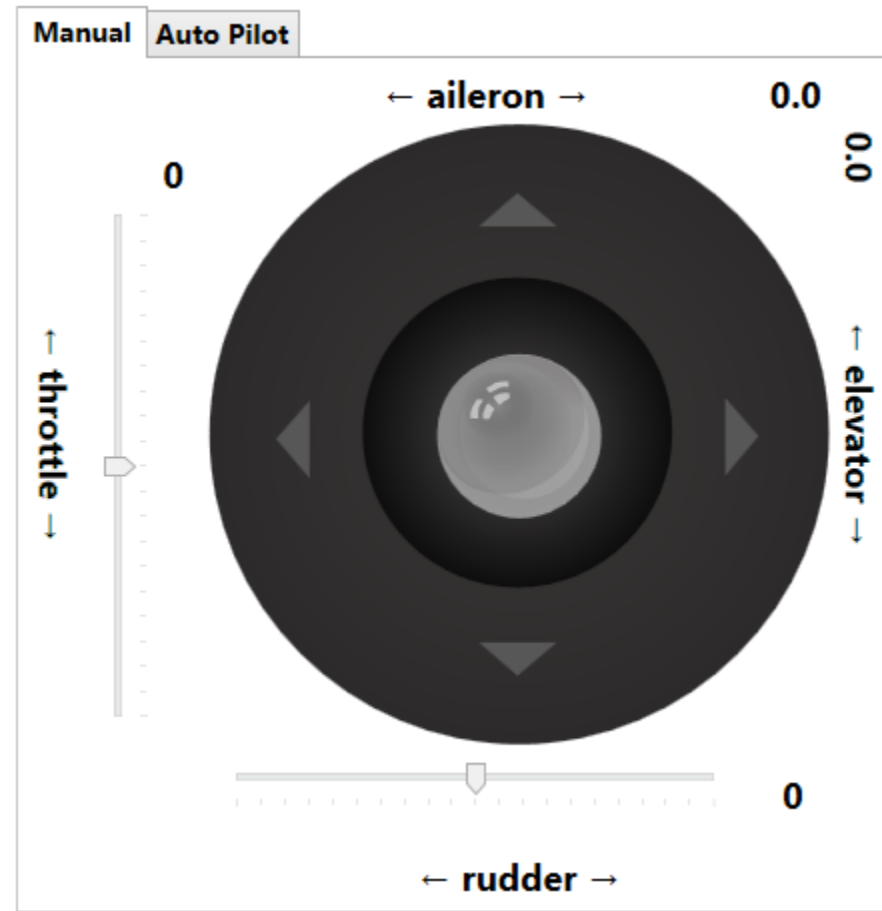
נסביר בהמשך את המרכיבים של החלון.

לצורך תרגיל זה, נצטרך להגדיר **2 ערוצי תקשורת** של TCP (שייבנו בקוד שלנו) באופן הבא:

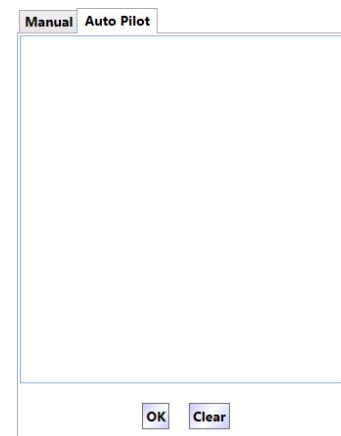
- **ערוץ Commands** - בערוץ זה, אנחנו נתחבר **כלקוח** אל סימולטור הטיסה (אשר ישמש כשרת). בערוץ זה אנחנו נעביר לסימולטור הטיסה **פקודות set בסיסיות** (ראו מסמך **היכרות עם סימולטור הטיסה** עבור הסבר נוסף). אין צורך להשתמש ב- Interpreter אשר יצרתם סמסטר הקודם).
- **ערוץ Info** - בערוץ זה הסימולטור יתחבר **כלקוח** אלינו, וישלח לנו את הנתונים אודות המטוס (הנתונים יוגדרו בקובץ generic_small.xml). לפיכך, עלינו ליצור **שרת** אשר יאזין לנתוני המטוס (בפרט Lon, Lat).

את המטוס נוכל להטיס ב-2 שיטות:

- **ידנית** - יהיה לנו Joystick אשר באמצעותו נוכל לקבוע את ה- aileron, elevators, rudder, throttle. כך שכל שינוי בjoystick יעביר **פקודת set** מתאימה בערוץ ה- Command. להלן תצוגתו:



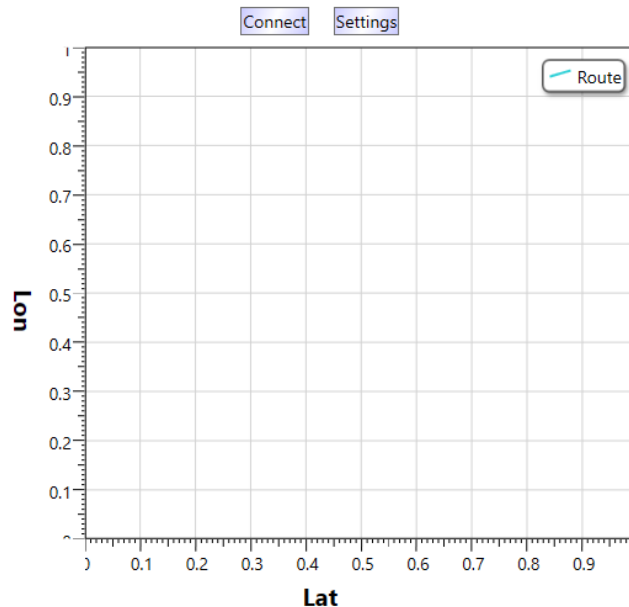
- **אוטומטית** - יהיה לנו שדה לכתיבת פקודות set. אשר לאחר הזנתן נוכל לשלוח אותם בערוץ ה- Command. להלן תצוגת הפקד:



חלק ג' – פירוט של חלקי המערכת

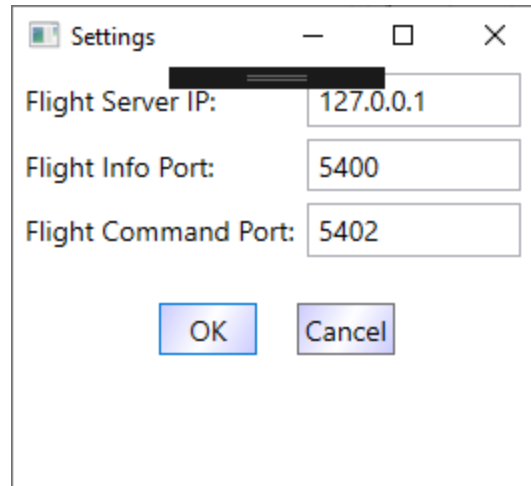
רכיב צג הטיסה

רכיב צג הטיסה, נראה כך:



ברכיב זה, ישנם 2 כפתורים:

- כפתור Connect – כפתור אשר מבצע חיבור ל-2 הערוצים (ערוץ ה- Commands וערוץ ה- Info). ניתן להניח כי החיבור לערוצים מבוצע בהצלחה ולכן אין חובה לבדוק את תקינות החיבור.
- כפתור Settings – כפתור זה מאפשר לשנות את ההגדרות הקשורות להפעלת המערכת. חלון זה נראה כך:



חלון זה מאפשר לשנות את ההגדרות של כתובת ה-IP של השרת וכמו כן שינוי של PORT ערוצי התקשורת.

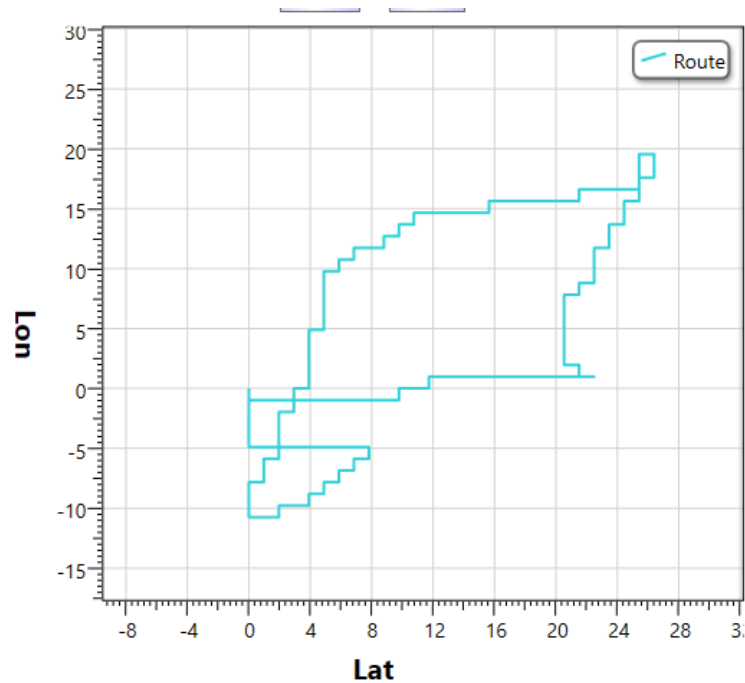
הגדרות אלו נשמרות בקובץ *App.config*, אשר נראה כך:

```
<userSettings>
  <FlightSimulator.Properties.Settings>
    <setting name="FlightServerIP" serializeAs="String">
      <value>127.0.0.1</value>
    </setting>
    <setting name="FlightInfoPort" serializeAs="String">
      <value>5400</value>
    </setting>
    <setting name="FlightCommandPort" serializeAs="String">
      <value>5402</value>
    </setting>
  </FlightSimulator.Properties.Settings>
</userSettings>
```

למענכם יצרתי את המחלקה *ApplicationSettingsModel.cs* אשר יודע לקרוא הגדרות אלו, ולקשר אותם ל- *Properties* מתאימים.

לאחר שינוי כתובות ה-IP וה-PORT, נתונים אלו יישמרו (אולם ההתחברות מחדש של ערוצי התקשורת תתקיים רק לאחר לחיצה מחודשת על כפתור ה- *Connect* או הפעלה מחודשת של המערכת).

בנוסף, בצג הטיסה, אנחנו רואים פקד דמוי גרף. בגרף זה אנחנו נציג את מסלול הטיסה של המטוס שלנו. מסלול הטיסה שלנו ייקבע עפ"י הנתונים הנשלחים בערוץ ה- *Info*, אשר מהם נרצה לדגום את ערכי ה- *Lat* ו- *lon* ולהציגם בגרף. לדוגמא:



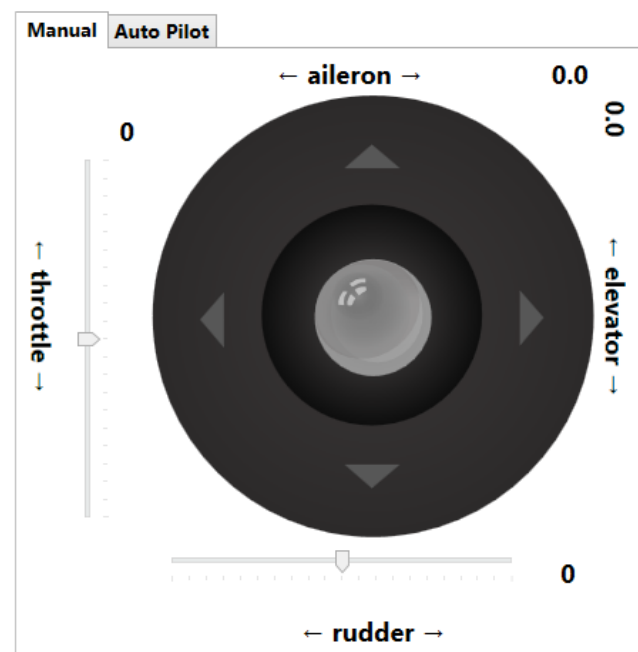
לשימושכם יצרתי את ה- UserControl הקרוי FlightBoard.xaml. (רכיב זה מאפשר גלילה של קדימה ואחורה, כך שאין חשש כי המטוס ייצא מגבולות הצג).
המסלול שמוצג הוא המסלול שהמטוס עבר! (ולא חישוב מסלול מסמסטר הקודם)

צג שליטה

בצג זה יהיה לנו 2 אפשרויות שליטה, אשר המעבר מאפשרות אחת למשנה הינה ע"י שימוש ב- TabControl.

שליטה ידנית

ברכיב זה עליכם לממש Joystick אשר באמצעותו נוכל לקבוע את ערכי aileron, elevators, rudder, throttle.



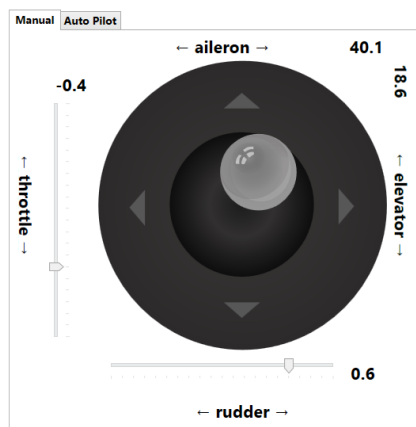
את ערכי ה- throttle ו- rudder נקבע באמצעות Slider אשר בעל ערכים בטווח [-1,1].

את ערכי ה- aileron, elevators נקבע באמצעות פקד הקרוי Joystick. יצרתי למענכם את הפקד הזה בקובץ Joystick.xaml, אמנם עליכם לקבוע **לוגיקה לקביעת הערכים של ה- aileron, elevators** בהינתן מיקום הכפתור. משמע עליכם יהיה לשנות את המתודה

```
private void Knob_MouseMove(object sender, MouseEventArgs e)
```

הקיימת בקובץ Joystick.xaml.cs.

כל שינוי של אחד הערכים יוביל לעדכון המטוס ע"י ערוץ ה- Commands.
(אין צורך לחכות ולהמתין לאימות הנתונים ע"י סימלטור הטיסה).



רכיב זה מכיל פקד TextBox אשר מאפשר לכתוב פקודות Set בערוץ ה- Commands. (ניתן להניח כי הפקודות תקינות) **אם ממשתם את המפרש מסמסטר קודם, אז רצוי שתשלחו את הפקודות של המפרש במקום פקודות set.**

הפקד מכיל 2 כפתורים:

- כפתור OK – מבצע שליחה של **כל הפקודות** אחת אחרי השנייה בערוץ ה- Commands כאשר בין כל פקודה לפקודה יהיה delay של 2 שניות על מנת שנוכל לראות זאת בסימלטור (**אין זה אומר שהצג יהיה קפוא במהלך שליחת המידע.. חשבו על דרך יצירתי לוודא ששליחת הפקודות תתבצע ברקע**). **יש להניח כי כפתור ה- OK יילחץ פעם אחת בלבד עבור כל כתיבת פקודות, ולא תהיה שליחה במקביל של 2 סטים שונים של פקודות!** (שוב, אין צורך להשתמש בקוד מסמסטר א', אך ורק בתוכנת הסימלטור!)
- כפתור Clear – מבצע ניקוי של מסך הפקודות. (אין צורך לנקות את המסך לאחר לחיצה על כפתור ה- OK).

כאשר המשתמש מזין פקודות אשר עדיין לא הגיעו אל השרת, הרקע של ה- TextBox יהיה צבוע בצבע אדום בהיר באופן הבא:

ולאחר שליחת המידע יחזור להיות לבן.

(אין צורך לחכות ולהמתין לאימות הנתונים ע"י סימלטור הטיסה).

צורת עבודה:

יש לקרוא לתרגיל Ex2 או Exercise2 או FlightSimulator בלבד!

את התרגיל ניתן להגיש **לכל היותר בזוגות**. ניתן, אך לא חובה.
את התרגיל יש לעשות בעזרת SOURCE CONTROL ומומלץ להשתמש ב git בעזרת bitbucket.
גם אם אתם עובדים לבד, חובה לעבוד ולהשתמש ב- source control אשר צורת העבודה אתו נלמדה ותורגלה בקורס תכנות מתקדם 1.

עם זאת, במודל תוכלו למצוא קישורים שיסייעו לכם, בפרט הדגמה של עבודה עמו ב. VS -
את התרגיל כולו כותבים ב C# - אין להשתמש בספריות אשר מחייבות התקנה נוספת של דברים מעבר ל Net Framework. בפרט, לא ניתן להשתמש בספריות או קוד מוכן אשר עושה עבורכם דברים שאתם אמורים לממש.

לתרגיל ישנו פורום ייעודי בפיאצה. את כל השאלות יש לשאול אך ורק דרך הפורום בפיאצה. חובה להתעדכן בפורום. כל הנאמר בו מחייב את כולם.

עליכם לכתוב את הקוד ע"פ ה Naming Convention המקובלים של שפת C# .
פירוט תוכלו למצוא בקישור הבא:

[https://msdn.microsoft.com/en-us/library/ms229002\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/ms229002(v=vs.110).aspx)

עליכם להגיש את כל הקוד של הפרויקט מתועד ע"פ הסטנדרט הבא:

<https://msdn.microsoft.com/en-us/library/aa288481%28v=vs.71%29.aspx>

צורת ההגשה:

עליכם להגיש קובץ zip שיכיל 3 תיקיות:

1. **תיקיית Src:** אשר תכיל את כל הקוד של ה- Solution הכולל את כל הפרויקטים.
2. **תיקיית exe:** אשר תכיל את כל קבצי ההרצה (קבצי exe) של כל אחד מהפרויקטים – כל אחד בתיקייה נפרדת. פשוט להעתיק את תיקיית bin של אחד כל אחד ואחד מהפרויקטים.
3. **תיקייה בשם etc:** אשר תכיל קובץ בשם info.txt אשר יכיל את שמות המגישים, ת"ז, קבוצת תרגול של כל אחד. בנוסף היא תכיל את ה- log של העבודה מול ה- git.

כל קבצי ההרצה יורצו על מערכת windows עם net framework. חדש אשר אמור לתמוך בכל הגרסאות אחורה. מחובתכם לוודא שכל הקבצים שהגשתם תקינים, ובפרט שהגשתם את **כל הקבצים** הנדרשים. מומלץ לוודא שהקבצים שלכם רצין תקין במחשב שונה משלכם טרם ההגשה.

לא ניתן יהיה להגיש קבצים מחדש או רטרואקטיבית או תיקונים או שינויים לקבצים לאחר מועד ההגשה. רק אחד מבני הזוג מגיש את התרגיל. **חובה** להגיש אך ורק לקבוצת התרגיל שאתם רשומים אליה.

את התרגיל יש להגיש עד השעה 22:00 דרך המודל.

בפועל מערכת ההגשה תאפשר הגשה עד 23:59 של היום הנקוב, אך הגשה לאחר 22:00 היא "על אחריותכם" בלבד. **ניתן להגיש באיחור עם הפחתת ניקוד בהתאם.**

בהצלחה

דור ואלי