

פרויקט פיתוח תוכנה מתקדם 2

הקדמה

סטודנטים יקרים שלום רב,

הפעם בפת"מ 2 אנו עולים מדרגה. בדומה לעולם האמיתי, הפרויקט יוגדר כאוסף של סיפורי משתמש (user stories) אך בכל זאת עם כמה הכוונות ודרישות אקדמיות. אתם תעבוד בצוותים של עד 4 סטודנטים, ותידרשו לצלוח בשלב זה משוכה – להחליט לבד. למעשה אתם נדרשים:

- לתכנן את סדר העדיפויות שבו "תשרפו" סיפורי משתמש בכל "ספרינט" של עבודה.
- לכל סיפור משתמש עליכם לבצע את כל מעגל הפיתוח: לנתח-לתכנן-לממש-לבדוק.

זה לא פשוט בהתחלה, אך זו התנסות מאד חשובה ואף אפשרית בשלב זה.

לקראת סוף הסמסטר תצטרכו להגיש לתיבת הגשה במודול מספר תוצרים שיוגדרו בהמשך בדף המודול של הקורס. תוצרים אלו יכללו אלמנטים כגון תכנון לוח הזמנים המקורי, תרשים מחלקות, לינק לקוד ב GIT, בדיקות שנערכו, הוכחה למצב הסיום של כל user story, וגם סרטון דמו של כל הפיצ'רים.

כדי להימנע מדחיינות תצטרכו להגיש **אבני דרך**:

1. מסמך שיכיל את שמות אנשי הצוות ותכנון זמנים למימוש הפרויקט.
2. Design – תרשים מחלקות ותרשימי UML אחרים של התכנון.
3. הפרויקט הסופי על כל תוצריו.

לפרויקט תהיה בדיקה ידנית.

כל פרטי ההגשה כאמור יינתנו בהמשך הסמסטר בדף המודול של הקורס.

בהצלחה!

תיאור המשתמש

המשתמשים העיקריים שלנו הם איילת ואיל פלג, תאומים בשנות ה 20 לחייהם, ששירתו יחד ביחידה טכנולוגית בחיל האוויר, וכיום סטודנטים לתואר שני במדעי המחשב עם תזה בתחום תחקור טיסות בלתי מאוישות וגילוי חריגות. הם חכמים, חרוצים, ענייניים מאד, וחיים את הטכנולוגיה על כל גווניה מגיל צעיר. לאיילת יש בעיית ראייה והיא מרכיבה משקפיים עם מספר די גבוה. לאייל יש עיוורון צבעים קל. אייל משתמש בעיקר בלינוקס בתפוצת אובונטו ואוהב לעבוד דרך ה terminal, ואילו איילת מעדיפה מק ולעבוד בד"כ מול GUI. באוניברסיטה הם עובדים על חלונות.

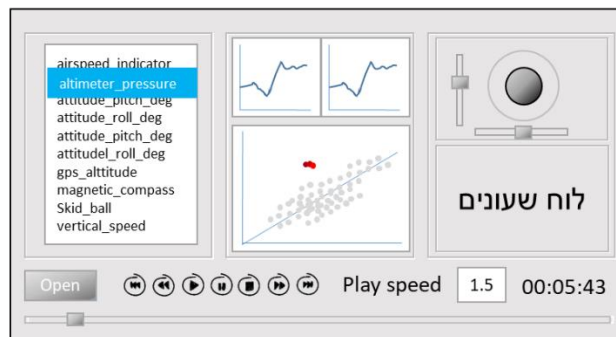
במסגרת המחקר שלהם, הם קבלו קובצי טיסה – קובצי CSV המהווים time-series data של נתוני טיסה שונים. לפני שיתחילו לנקות את המידע ולהפעיל כל מיני אלגוריתמים של למידת מכונה, הם מבינים שהם צריכים לבצע ויזואליזציה של המידע כדי להבין מה בכלל קורה בכל טיסה. הם מבינים שהם צריכים "להקריין" את הטיסה מתוך הנתונים ופשוט לראות בעיניים מה קורה שם.

כמו כן, כאשר הם נפגשים עם המנחה שלהם גל מאיר, הם רוצים להשתמש באפליקציה הזו כדי להדגים כיצד אלגוריתמים שונים זיהו רגעי חריגה שונים בטיסות שונות. הם מצפים שגל יבקש מהם להריץ שוב, קדימה ואחורה, לעצור בנק' זמן מסוימות ולהחליף אלגוריתמים כדי לראות, לנתח ולהפיק תובנות.

תיאור כללי של האפליקציה – רק כדי לקבל הקשר

CSV Flight Data Input

	A	B	C	D	E	F	G	H
1	airspeed_i	altimeter_	altimeter_	attitude_ir	attitude_ir	attitude_ir	attitude_ir	encod
2	14.0113	-54.6073	-53.4138	3.072681	1.355898	2.720353	0.181473	-49.9
3	12.04959	-56.7784	-55.5849	3.031284	1.218807	2.72001	0.181227	-49.9
4	11.15854	-57.3682	-56.1747	3.056030	1.068761	2.710742	0.181102	-50.0



FlightGear
Simulator:



בהינתן קובץ CSV המכיל time-series data של נתוני טיסה (שנדגמו בקצב של 10Hz למשל) נרצה להציג את הטיסה כבסרטון וידאו. לשם כך נתחבר לסימולטור הטיסה FlightGear אך במקום להשתמש בו כסימולטור נשתמש בו כמקרן. נשדר לו את נתוני הטיסה והוא יציג אותם.

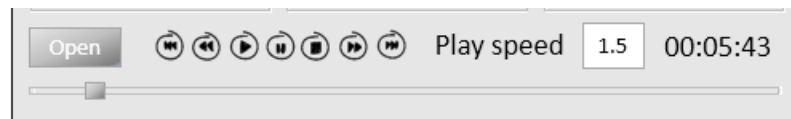
נרצה יכולת לשדר את הנתונים בקצב אמתי, מהיר, או איטי או מוגדר ע"י המשתמש. נרצה גם איזשהו פס גלילה שניתן להריץ באמצעותו את הטיסה קדימה או אחורה וכמובן הסימולטור יגיב מידית כאילו היה מדובר בסרטון.

ב GUI שלנו נרצה לראות 4 אלמנטים עיקריים:

- ג'ויסטיק וירטואלי שמציג את מצב ההגאים (Aileron & Elevators), מייצב הכיוון (Rudder) והמצערת (Throttle)
- לוח שעונים - פאנל מכשירים שרואים בו שעוני גבוה, כיוון, מהירות, yaw, pitch, roll
- רשימה של עמודות מקובץ ה CSV
- קנבס שמציג את גרף גילויי החריגות, ע"פ האלגוריתם שהוגדר בהגדרות האפליקציה.

כפתור ה Open יפתח דיאלוג לפתיחת קובץ CSV.

באמצעות **הפקדים** נוכל להזיז את הטיסה לכל נקודת זמן שנרצה, ולצפות בטיסה באיזה קצב שנרצה.



כמובן, האפליקציה תשדר (באמצעות תקשורת TCP) את הנתונים של נק' הזמן הנוכחית לסימולטור הטיסה כדי שיקרין את תמונת המצב הנוכחית.

בנוסף האפליקציה תציג את מצב ה**ג'ויסטיק** (העיגול שבתוך המעגל). כאשר ציר ה X הוא ה aileron (השולט בגלגול המטוס) וציר ה Y הוא ה elevators (השולטים ב pitch). **שימו לב:** זה לא ג'ויסטיק לשליטה ידנית, אלא לתצוגת המצב של ההגאים כפי שמופיעים בקובץ ה CSV. כמו כן נציג את מצב המצערת בפס גלילה אנכי, ואת מצב מייצב הכיוון באופקי האופקי.

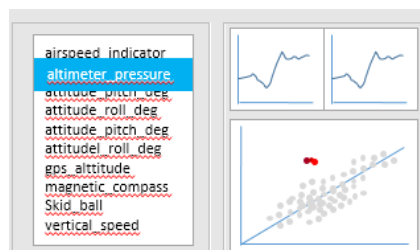


לוח השעונים נרצה להציג את הנתונים הנוכחיים של:

- גובה הטיסה (ע"פ ה altimeter)
- מהירות הטיסה (airspeed)
- כיוון הטיסה
- ומדדי yaw, roll, pitch.

אתם בוחרים כיצד להציג את הנתונים. זה יכול להיות סתם טקסט (שזה די עלוב) או תצוגה גרפית כלשהי כמו למשל של השעונים במטוס.

תצוגת הנתונים:



צריכה להיות לנו רשימה של כל האטריביוטים (עמודות) שיש ב CSV. כאשר בוחרים אחד מהם, למשל ה altimeter, אז נראה בצד שמאל למעלה גרף נע של הערכים שלו מתחילת הטיסה ועד נק' הזמן הנוכחית.

באופן דומה נראה לצידו גרף של ה אטריביוט שנמצא הכי קורלטיבי אליו (ע"פ Pearson) בטיסה שנחשבת תקינה (למשל GPS ALT).

בגרף הגדול התחתון נראה תצוגה גרפית של פעילות האלגוריתם שנבחר עבור גילויי החריגות. למשל עבור האלגוריתם מפת"מ 1 נראה את קו הרגרסיה שהם יצרו (altimeter X GPS ALT) ונק' ברקע שמציינות את הערכים שראינו בטיסה תקינה. לעומת זאת נרצה לראות באופן בולט את הערכים של נק' הזמן הנוכחית, למשל בצבע אדום עם שובל של 30 הנק' האחרונות לדוגמה.

שימו לב: נרצה את האפשרות להחליף את האלגוריתם (ותצוגתו) גם בזמן ריצה.

אם זוהתה חריגה בנק' זמן כלשהי יש להתריע על כך, למשל ע"י לצבוע את הרקע של הגרף בצבע אדום או בכל דרך אחרת שתבחרו.

דרישות אקדמיות

1. האפליקציה צריכה להיות כתובה ב **Java**
2. ארכיטקטורת התוכנה צריכה להיות **MVVM** ששומרת על כל עקרונות ההפקדה בין השכבות
3. ה **GUI** צריך להיות כתוב ב **JavaFX**
4. כל אלמנט מ 4 האלמנטים ב **GUI** שהוגדרו לעיל (למשל הג'ויסטיק) צריך להיות **עצמאי**.
 - a. צריך שיהיה ניתן להוסיף או להסיר אותו כתגית ב **FXML** בדיוק כמו שמוסיפים או מסירים כפתור למשל.
 - b. לשם כך כל אלמנט צריך לספק אפשרויות הגדרה והתממשקות גנריות באמצעות **data binding**.
 - c. הימנעו מחלון שמכיל את כל הרכיבים הבסיסיים שמרכיבים את התצוגה הסופית. החלון צריך להכיל כ 4 אלמנטים בלבד שכל אחד בתורו הוא אובייקט מורכב מאלמנטים בסיסיים יותר.
5. גילויי החריגות יכול להתבצע לוקאלית או ע"י תקשורת לשרת שכתבנו בסמסטר א'.
6. האלגוריתם לגילויי החריגות ותצוגתו הגרפית, יהיו **פלאג-אין** של האפליקציה. כלומר ניתן יהיה לטעון גם אלגוריתם שימוש ויקומפל אפילו אחרי שהאפליקציה עצמה כבר רצה. אל דאגה, נלמד איך לעשות זאת בנספח שיפורסם בנפרד במודול.
7. במסמך נפרד שיפורסם במודול תקבלו דרישות למימוש של אלגוריתמים נוספים לגילויי חריגות.
8. בנספח נפרד שיפורסם במודול אכיר לכם את סימולטור הטיסה וכיצד להתממשק איתו.

סיפורי משתמש

זמן:	עדיפות:	אייל עורך הגדרות
סיפור משתמש: בקובצי CSV שונים ייתכנו שמות שונים לעמודות, למשל נתון גובה הטיסה יכול להיקרא ALT או altitude וכו', בנוסף ייתכנו גם ערכים מינימאליים ומקסימאליים שונים, קצב דגימה שונה וכו'. לכן כאייל אני רוצה להגדיר את משמעות העמודות שבקובץ ה CSV (ודברים נוספים) כדי שאוכל להציג באפליקציה ולשדר אותן לסימולטור הטיסה.		
אייל פותח קובץ XML באמצעות עורך שקיים במערכת ההפעלה שלו. שם הוא יוכל להגדיר קשר אסוציאטיבי בין שמות העמודות בקובץ ה CSV לבין המשמעות שלהם. כך האפליקציה תדע מאלו עמודות יש לקחת את הנתונים הבאים: מיקום בכדור הארץ, גובה, מהירות, כיוון, roll, pitch, yaw, מצערת, צירי ה X ו ה Y של הג'ויסטיק, והגאי מייצב הכיוון. לכל נתון שכזה אייל מגדיר את הערך המקסימלי והמינימלי שיש לכל עמודה שכזו. לדוג' מייצב הכיוון נע בין 1- ל 1. בנוסף יש להגדיר את קצב דגימת הנתונים בקובץ, למשל 10 פעמים בשנייה, מהו שמו של הקובץ שמגדיר טיסה תקינה, והקובץ בו שמור אלגוריתם גילוי החרیגות.		
קריטריון לסיום: הקובץ נשמר במערכת ההפעלה		

זמן:	עדיפות:	אייל טוען הגדרות
סיפור משתמש: כאייל, לאחר שערכתי את ההגדרות בקובץ ה XML, ארצה לטעון אותן לאפליקציה.		
אייל פותח את האפליקציה ובוחר את קובץ ה XML שממנו ייטענו ההגדרות. אם ישנה בעיה בטעינה, האפליקציה תציג הודעת שגיאה מתאימה (למשל אם אחד הנתונים חסרים, או פורמט ה XML פגום וכדומה). אחרת, היא תציג הודעה שהטעינה בוצעה בהצלחה. אם אייל לא בוחר קובץ XML האפליקציה תזכור את ההגדרות מהקובץ שנטען בפעם האחרונה.		
טיפ: זיכרו שניתן לשמור \ לטעון אובייקט כ XML באמצעות מחלקות פשוטות ומתאימות.		
קריטריון לסיום: הודעה מתאימה על הצלחה או כישלון הטעינה של קובץ ה XML.		

זמן:	עדיפות:	איילת טוענת קובץ טיסה
סיפור משתמש: כאיילת אני רוצה לטעון קובץ טיסה כדי שאוכל להציגה בצורה ויזואלית באמצעות האפליקציה		
איילת לוחצת על כפתור Open. יופיע דיאלוג לבחירת קובצי CSV. איילת תבחר קובץ. האפליקציה תטען ותבדוק את הקובץ. אם ישנה בעיה בפורמט הקובץ, למשל נתונים חסרים, או שלא נמצאו עמודות המתאימות להגדרות בקובץ ה XML אז תופיע הודעת שגיאה מתאימה. אחרת הקובץ יטען והטיסה תהיה מוכנה לתצוגה		
קריטריון לסיום: <ul style="list-style-type: none"> האטריביוטים של הטיסה (כותרות העמודות ב CSV) יופיעו ברשימה זמן הטיסה יוגדר ל 0 (תחילת הטיסה) והטיסה עדין לא תופעל לוח השעונים והג'ויסטיק יציגו את המצב של זמן 0 		

זמן:	עדיפות:	אייל מקרין טיסה
סיפור משתמש: כאייל במקום להסתכל על המון נתונים יבשים בטבלה, אני רוצה להציג את הטיסה השמורה בקובץ באמצעות סימולטור שישמש אותי כמקור על מנת שאוכל לראות הרבה פרמטרים באותו הזמן ולהסתכל על הטיסה "מבחוץ" ולקבל איזשהו הקשר שיעזור לי להבין מה קורה בה.		
בדומה להפעלת סרטון ב YouTube, אייל מפעיל פקדים של play, pause, rewind, forward, stop, וכן תזוזה של הטיסה לנק' זמן מסוימת באמצעות פס גלילה (scrollbar / slider). בהתאמה יוצג זמן הטיסה הנוכחי וכן אם הוא פתוח, סימולטור הטיסה FlightGear יקרין את נתוני הטיסה שבנק' הזמן הנוכחית. אם הוא לא פתוח אז לא תבוצע הקרנה אך האפליקציה תמשיך להתנהג כרגיל (למשל להראות את גובה הטיסה בלוח השעונים)		
הסבר: משמעות ההקרנה היא להורות לסימולטור למקם את המטוס בדיוק במיקום בנכחי שלו בכדור הארץ, ובגובה, כיוון, וה attitude (ה roll, pitch, yaw) הנוכחיים שלו. כאשר נותנים לטיסה לרוץ בקצב כלשהו זה יראה כאנימציה של הטיסה.		
טיפ: יש לכם נספח במודול שמגדיר לכם כיצד להתממשק ולתקשר עם FlightGear.		
קריטריון לסיום: סימולטור הטיסה מציג באופן רצוף את הטיסה ע"פ ההפעלה של אייל		

זמן:	עדיפות:	איילת צופה בנתוני הטיסה A
סיפור משתמש: כאיילת אני רוצה לראות את הפקודות שנתן הטייס למטוס וכיצד המטוס הגיב להן על מנת שאוכל לזהות מקרים חריגים.		
איילת מקרינה את הטיסה. בהתאמה לנק' הזמן הנוכחית נתוני ההגאים מוצגים כג'ויסטיק וירטואלי. הג'ויסטיק מוצג כעגול בתוך מעגל וינוע בציר ה X ובציר ה Y בגבולות המעגל בהתאם לפקודות הטייס וביחס לערכי המינימום והמקסימום האפשריים. כמו כן נתון המצער (throttle) יוצג באמצעות פס גלילה אנכי, ונתון הגאי מייצב הכיוון (rudder) יוצג באמצעות פס גלילה אופקי.		
באופן דומה, בהתאמה לנק' הזמן הנוכחית לוח השעונים יציג את הנתונים הבאים: גובה, מהירות, כיוון, roll, pitch, yaw		
קריטריון לסיום: התאמה מלאה בין תצוגת לוח השעונים והג'ויסטיק לנתונים הנוכחיים בקובץ ה CSV ע"פ נק' הזמן שמוקרנת.		

זמן:	עדיפות:	איילת צופה בנתוני הטיסה B
סיפור משתמש: כאיילת אני רוצה לראות כיצד אטריביוטים שונים מתנהגים לאורך הטיסה על מנת שאוכל להסיק מסקנות בדבר חריגות אפשריות.		
איילת מקרינה את הטיסה. איילת בוחרת מתוך הרשימה את אחד האטריביוטים (למשל airspeed indicator). האפליקציה תציג גרף בו ציר ה X הוא ציר הזמן שמתחיל ב 0 ונגמר בנק' הזמן הנוכחית. ציר ה Y הוא ציר הערכים שמתחיל מהערך המינימלי ומסתיים בערך המקסימלי שנצפו מתחילת הטיסה ועד נק' הזמן הנוכחית. הגרף יציג קו שמראה כיצד נעו הערכים לאורך הזמן.		
באופן דומה יוצג גרף נוסף עבור האטריביוט הקורלטיבי ביותר (ע"פ פירסון) בטיסה תקינה.		
קריטריון לסיום: <ul style="list-style-type: none"> התאמה מלאה בין תצוגת הגרף לנתונים הנוכחיים בקובץ ה CSV עד לנק' הזמן שמוקרנת. גרף מתעדכן תוך כדי הטיסה או שינוי נק' הזמן 		

זמן:	עדיפות:	אייל בוחר אלגוריתם לגילוי חריגות
סיפור משתמש: כאייל אני רוצה תוך כדי הקרנת הטיסה להחליף בין אלגוריתמים שונים לגילוי חריגות על מנת לקבל המחשה ויזואלית לפעולתם בתגובה לנתוני הטיסה המוקרנים.		
אייל לוחץ על כפתור או תפריט מתאים באפליקציה. האפליקציה פותחת דיאלוג לבחירת קובץ class. אייל בוחר קובץ class שמייצג אלגוריתם לגילוי חריגות. האפליקציה טוענת את הקובץ ומודיעה האם הטעינה הצליחה או נכשלה.		
טיפ: יש לכם נספח במודול שעוסק בטעינת קובצי class בזמן ריצה.		
קריטריון לסיום: הודעה מתאימה על הצלחה או כישלון הטעינה		

זמן:	עדיפות:	אייל צופה באלגוריתם לגילוי חריגות
סיפור משתמש: כאייל אני רוצה תוך כדי הקרנת הטיסה לצפות בוויזואליזציה של גלאי החריגות בפעולה ולקבל דיווח על גילוי חריגה.		
אייל מקרין את הטיסה. בהתאמה לנתוני הטיסה האפליקציה מפעילה את האלגוריתם הנבחר ומציגה את הוויזואליזציה שלו. אם האלגוריתם מגלה חריגה בנק' הזמן הנוכחית האפליקציה תתריע על כך. (אתם מחליטים על הדרך)		
שימו לב: במסמך נפרד שיפורסם במודול יוגדרו לכם האלגוריתמים הדיפולטיביים שבאים עם האפליקציה והתצוגה שלהם.		
<ul style="list-style-type: none"> אלגוריתם Univariate מבוסס zScore אלגוריתם מסמסטר א' (תוכלו פשוט לתקשר עם השרת) אלגוריתם היברידי 		
קריטריון לסיום: האלגוריתם מוצג ופועל בהתאמה להגדרתו ולקלט נתוני הטיסה בנק' הזמן הנוכחית.		

שימו לב לפרסומים במודול בדבר תוצרי ומועדי הגשה של אבני הדרך השונות.

בהצלחה!