

תכנות מתקדם 2 - אפליקציית תחקור טיסה.

להגשה עד: **12.04.2021** במודל. ניתן להגיש בקבוצות של עד 4 סטודנטים.

הקדמה

סטודנטים יקרים שלום רב, הפעם אנו עולים מדרגה. בדומה לעולם האמיתי, אבני הדרך הסמסטר מוגדרים כאוסף של סיפורי משתמש (user stories), הניתנים לצוות פיתוח מאחראי המוצר (Product Manager), עם כמה דגשים והכוונות אקדמיות. כל סיפור משתמש מגלם תכולה אחת או יכולת אחת של האפליקציה שתבנו ונועד לחלק את המשימות בצורה לוגית בין חברי הצוות. אתם תעבדו בצוותים של עד 4 סטודנטים, ואתם נדרשים לתכנן את המשימה, להשלים בעצמכם את פרטי המימוש, לחלק את העבודה בצורה יעילה בצוות, להסכים על ארכיטקטורה משותפת ועוד שלל אתגרים המדמים את העולם האמיתי. אתם רשאים להחליט כיצד נראית ומתנהגת האפליקציה שלכם, כל עוד סיפורי המשתמש ממומשים בצורה סבירה. זה לא פשוט בהתחלה, אך זו התנסות מאד חשובה ואף אפשרית בשלב זה.

בהצלחה.

רקע

בפרויקט זה נבנה אפליקציה שתאפשר לנו להציג נתוני טיסה על גבי סימולטור ייעודי ולחקור אותם. המשתמשים שלנו הם חוקרי טיסה או טייסים שרוצים לצפות בנתונים, שנדגמו בקצב מסוים במהלך טיסה כלשהי. נתוני הטיסה כוללים את מצב ההגאים, המהירות, הכיוון הגובה וכו', והם מוקלטים לתוך קובץ טקסט, שאותו ניתן לטעון באפליקציה שלנו. האפליקציה תנגן את הנתונים כמו סרט מתחילת ההקלטה ועד סיומה, היא תציג גרפית את המטוס ביחס לכדור הארץ, את מצב ההגאים, ונתוני טיסה נוספים במספר תצוגות שונות, כולל תצוגה המיועדת למציאת חריגות בנתונים.

הדרישות בתרגיל יינתנו על ידי מספר סיפורי משתמש קצרים, שילוו לעיתים בשרטוט בסיסי לביאור הכוונה. אתם **רשאים** לממש את סיפור המשתמש בצורה שונה מהשרטוט, מלבד סיפור משתמש אחד, אין הגבלה בבחירת הפקדים או העיצוב של האפליקציה.

מטרת התרגיל היא שימוש בתבנית העיצוב MVVM, אנחנו ממליצים להשתמש ב-C# וב-WPF, אתם רשאים לבחור בכל טכנולוגיה מתאימה אחרת. לא כל פרטי המימוש השונים מאופיינים על ידי סיפורי המשתמש, ועליכם להשלים את החסר כראות עינכם.

התוצר הסופי

במודל יש להגיש מסמך CSV עם העמודות הבאות, מופרדות בפסיק: student_id, student_name, link

- student_id מספר תעודת הזהות של המגיש
- student_name שם המגיש באנגלית, **אותיות ascii בלבד**
- link לינק לבראנץ' ייעודי בחשבון github המכיל את הגרסא הסופית של הפרויקט. אין להוסיף דבר ל-branch הנ"ל לאחר תאריך ההגשה. עמודה זו אמורה להיות זהה לכלל המגישים.

- מספיקה הגשה בודדת לכל פרויקט (סטודנט אחד מגיש). אנא בדקו שניתן להוריד את הפרויקט באמצעות git clone על ידי כל משתמש.
- על התיקייה הראשית ב-git להכיל קובץ README, זהו קובץ טקסט, ניתן לכתוב בו תיעוד באמצעות שפת ה-markup של GitHub. הקובץ הזה מחולק לכמה חלקים:
- (1) תיעוד והסבר קצר על הפרויקט ועל פיצ'רים מיוחדים שמימשתם, והבודק צריך לשים לב אליהם.
 - (2) תיעוד והסבר כללי על מבנה התיקיות והקבצים הראשיים בפרויקט.
 - (3) התקנות נדרשות מכל מפתח שרוצה לעבוד על הפרויקט שלכם. גרסאות ספציפיות של כלים ו-frameworks שהשתמשם וכו..
 - (4) הוראות התקנה וריצה ראשונית של האפליקציה שלכם על מחשב חדש.
 - (5) קישור לקבצי תיעוד נוספים בתוך git המכילים תיעוד מפורט על המחלקות הראשיות בפרויקט, על זרימת המידע, תרשימי UML של המחלקות הראשיות, יש לתת דגש על תבנית העיצוב MVVM בתרשימים ובתיעוד הנוסף.
 - (6) קישור לסרטון עד 6 דקות, בו אתם מדגימים את סיפורי המשתמש, לפי הסדר בו הם מופיעים במסמך.
- כמו כן, התיקייה הראשית ב-GIT צריכה להכיל תיקייה בשם **plugins** עם שני אלגוריתמי זיהוי חריגות הניתנים לשימוש יחד עם האפליקציה, כמו שיפורט בהמשך.
- מבנה שאר התיקיות והקבצים הוא לשיקול דעתכם.

בנוס אתגר ההיסטוריה הליניארית -

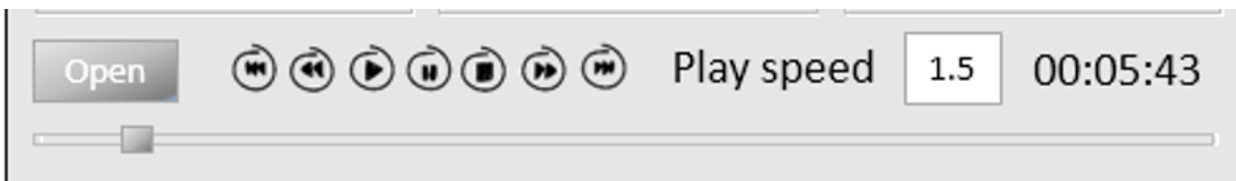
במידה ועבדתם ברציפות על הפרויקט ויש מספר גדול של commits ב-GIT ממשתמשים פעילים שונים (כלומר רוב המגישים היו contributors פעילים), מתעוררת בדרך כלל בעיה בשמירה על עץ גרסאות לא מסובך, ללא merges אלא רק fast-forwards. הדבר ניתן לביצוע על ידי rebase תכוף בכל push. פרויקט שיגיש עץ commits ליניארי לחלוטין ללא אף merge, אך יכיל contributions משמעותיים ומרובים ממשתמשים פעילים שונים יזכה ב-3 נק' בנוס.

סיפורי המשתמש

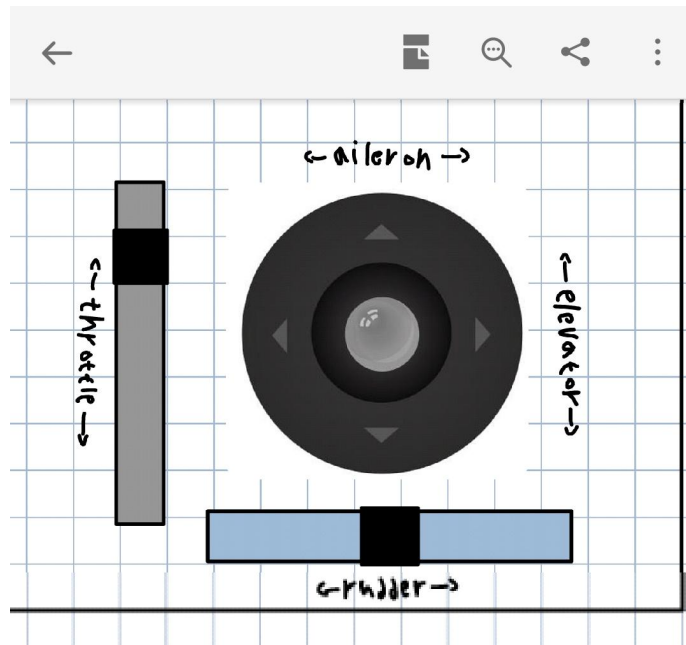
1. בתור חוקר טיס, הייתי רוצה לטעון קובץ CSV שבו מוקלטים נתוני טיסה שנדגמו בקצב כלשהו (למשל של 10Hz). דוגמא לקובץ CSV בפורמט הנכון נמצאת במודל של ההרצאה. לפני הטעינה יש לוודא כי קובץ ההגדרות (playback_small.xml שגם נמצא במודל) נקרא על ידי האפליקציה. בקובץ זה מופיעות הגדרות לקצב הדגימה, וכן שמות העמודות הרלוונטיות בקובץ ה-CSV והתיאור שלהם. (הערה: עליכם להחליט כיצד לממש את טעינת שני הקבצים בדרך נוחה למשתמש). לאחר טעינת הקובץ הייתי רוצה שהאפליקציה תנגן בפקד ייעודי סרט המראה את המטוס בכל רגע נתון מתחילת הטיסה ועד סופה בקצב הנדגם. התמונה בסרט מופקת באמצעות תוכנה חיצונית בשם Flightgear. זהו סימולטור טיסה המשמש טייסים חובבים להתלמד בשליטה על מטוס. אנחנו נשתמש בו כמקור שמציג את תמונת המטוס בהתאם למצב ההגאים ונתונים נוספים. משמעות ההקרנה היא להורות לסימולטור למקם את המטוס בדיוק במיקום הנוכחי שלו בכדור הארץ, ובגובה, כיוון, וה attitude (ה roll, pitch, yaw) הנוכחיים שלו. כאשר נותנים לטיסה לרוץ בקצב כלשהו זה יראה כאנימציה של הטיסה. על מנת לקבל תמונה עדכנית יש לשדר את נתוני הטיסה מהקובץ לתוך FG, ולקבל ממנו תמונה באמצעות API ייעודי. קיים מסמך במודל עם הסברים נוספים כיצד להתממשק מול FG. להלן תמונה מתוך FG:



2. בתור חוקר טיס, לאחר טעינת נתוני הטיסה, הייתי רוצה לדלג לכל זמן בטיסה באמצעות פקד דומה לפס גלילה שיש בסרטוני youtube. אני אוכל להזיז את הפקד קדימה או אחורה לכל זמן שארצה, ותצוגת המטוס (יחד עם תצוגות נתונים נוספות) ידלגו לזמן זה וימשיכו את סרט הטיסה משם. ניתן לממש את הפונקציונליות הזאת גם בצורות אחרות, כל עוד זה נוח למשתמש. להלן דוגמא בסיסית:



3. בתור חוקר טיס, אני רוצה לראות את מצב ההגאים הראשיים של המטוס בתצוגה דמויית ג'ויסטיק (זוהי דרישת העיצוב הקשיחה היחידה, מכיוון שטייסיים רגילים להשתמש בג'ויסטיק). הג'ויסטיק לא משמש לשליטה על המטוס, אלא רק מציג את מיקום ההגאים כפי שהם מוזנים מתוך הקובץ. הג'ויסטיק וההגאים משנים את מיקומם בהתאם לזמן הטיסה.
להלן דוגמא:



4. בתור חוקר טיס, לפעמים ארצה לצפות במהירות גבוהה בקטעי טיסה שונים שבהם לא קרה דבר, ולעיתים להתמקד בקטעי טיסה קריטיים לצפות בהם במהירות נמוכה מאוד עד כדי עצירה. לכן ארצה לשלוט על קצב ה- playback, באמצעות פקד ייעודי. יש לכם חופש להחליט כיצד הפקד נראה ואיך משתמשים בו, העיקר שיהיה נוח לשימוש שוטף, ניתן לקבל השראה מהתרשימים הקודמים, אך אין דרישה לממש אחד לאחד.

5. בתור חוקר טיס, הייתי רוצה לראות לצד התמונה של המטוס ומצב ההגאים גם מספר נתונים שונים:

1. גובה הטיסה (ע"פ ה altimeter)
 2. מהירות הטיסה (airspeed)
 3. כיוון הטיסה
 4. ומדדי yaw | roll, pitch
- היו יצירתיים בהצגת הנתונים הללו.

6. בתור חוקר טיס, הייתי רוצה לבחור נתון מסוים מתוך כל הנתונים המופיעים בקובץ, ולחקור אותו. ברגע שאבחר בנתון מסוים, למשל GPS ALT (גובה כפי שנמדד ב-GPS) הייתי רוצה לראות גרף נקודות מתעדכן של הנתון הנ"ל. ציר X הוא הזמן, וציר Y הוא ערך הנתון. להלן תרשים לדוגמה (צד ימין למעלה):



7. בתור חוקר טיס, ברגע שאני בוחר לחקור נתון מסוים, הייתי רוצה לראות את גרף הנתון הכי קורלטיבי (Pearson) מבין נתוני הטיסה השונים כפי שנמצאים בקובץ, יש להציג את הגרף של הנתון הנ"ל, (וכמובן את שמו) בצמוד לגרף מהסעיף הקודם. על הגרף להתעדכן עם הזמן.

8. בתור חוקר טיס, ברגע שאני בוחר נתון מסוים לתחקור ומתבונן בנתונים המתואמים, הייתי רוצה לראות גם את קו הרגרסיה הליניארית ביניהם. יש להדגיש ויזואלית את כל הנקודות בקו הרגרסיה השייכות לשלושים השניות האחרונות בטיסה. שימו לב שציר ה-X וה-Y מיועדים מציגים את שני הנתונים, ולא את הזמן בהם נדגמו. שרטוט לביאור בתרשים הקודם צד ימין למטה.

9. בתור חוקר טיסה, אני מעוניין להבין מתי קרתה חריגה במדידת המכשירים או בנתונים השונים על מנת לזהות תקלות אפשריות במטוס. זכרו שלחריגה בנתונים אין הגדרה אחת, ולרוב אני אשתמש בכל מיני אלגוריתמים שונים לזיהוי. אלו יכולים להיות אלגוריתמים שפותחו אחרי שהאפליקציה כבר נמצאת אצלי בשימוש, כלומר לא אצטרך לעדכן את הקבצים הבינארים באפליקציה על מנת להשתמש באלגוריתם גילוי חריגות חדש.

אני יכול לטעון את נתוני הטיסה, ולבחור אלגוריתם לגילוי חריגות. האלגוריתם יזהה באילו רגעים בזמנים התרחשה חריגה ויסמן זאת בצורה בולטת, כך שאוכל לקפוץ ביעילות לאותו רגע בזמן ולחקור אותו. בנוסף, כל אלגוריתם צריך לממש תצוגה שונה לחקר החריגה, בהתאם לאופיו ולסוג הנתונים שהוא עובד עליהם האלגוריתמים הראשונים שאני רוצה לעבוד עמם הם:

a. מבוסס רגרסיה (מימשתם בסמסטר א')

b. מבוסס מעגל פנימי (מימשתם בסמסטר א')

אני יכול לטעון אלגוריתם א', לחקור באמצעותו את הטיסה, ואז לטעון את אלגוריתם ב', אני מצפה שהחריגות יחושבו מחדש.

האלגוריתמים הללו ימומשו בנפרד **כפלאג-אין**, משמע, קבצי DLL הנטענים **דינאמית** באפליקציה לאחר עלייתה. אני כמשתמש מקבל לזיהוי חריגות כקובץ DLL חיצוני לאפליקציה (בתיקיה אחרת שאינה ידועה מראש) וטוען אותו דינאמית במהלך הריצה, בלי להפעיל מחדש, וכמובן שלא לקמפל מחדש את האפליקציה.

הערה: עליכם לשים את שני ה-DLL לשני האלגוריתמים הראשונים בתיקייה בשם plugins תחת התיקייה הראשית ב-GIT. כמו כן יש לפרט בתרשימי ה-UML ובתיעוד המחלקות כיצד מומש הסיפור הנ"ל בדגש על הטעינה הדינאמית של ה-DLL. ניתן להשתמש ב-Plug-In שפותח בסמסטר א'. אנא הסבירו במידה ותפתחו אלגוריתם נוסף לזיהוי חריגות, כיצד ניתן לשלבו בקלות באפליקציה בלי לשנות את הקוד ובלי לקמפל אותה מחדש.

דרישות אקדמיות

כחלק מדרישות הקורס, יש לממש את הדרישות הנ"ל.

1. שימוש בתבנית העיצוב MVVM

יש לממש ModelView ו-View לכל חלק לוגי ומשמעותי בעיצוב המשתמש. חלוקה הגיונית לדוגמא: פקד ההגאים, פקד הניגון ופקד הנתונים צריכים כל אחד לקבל View/ViewModel משל עצמם. כל אחד יכול לתקשר עם כל פקד אחר, למשל על ידי העברת הודעות או הקפצת events לפקד המרכזי של האפליקציה.

2. שימוש ב-Plug In

כמתואר בסיפור המשתמש, יש לממש את אלגוריתמי גילוי חריגות באמצעות DLL חיצוני הנטען דינאמית לאחר עליית האפליקציה. בעת עליית האפליקציה ה-DLL הללו לא זמינים, והם מסופקים על ידי המשתמש לאחר שהוא טוען אותם ממיקום כלשהו על המחשב שלו.

3. תיעוד קוד, תרשימי מחלקות וזרימה

כמתואר בחלק "התוצר הסופי" יש להוסיף תרשימי UML תיעוד מסודר והסברים לכל המחלקות המרכזיות. לשים דגש על מימוש MVVM ומימוש ה-Plugin.

4. קונבנציות קוד

עליכם להשתמש בקונבנציות הקוד הבאות:

(לינק לקונבנציות קוד יעלה בהקדם)

מפתח ניקוד

יעודכן בהקדם

בהצלחה

צוות הקורס.