**פירוט המימוש:**  
**Uav class:**  
class זה מייצג כטב"מ ומכיל את כל הפונקציות הרלוונטיות לכלי.  
  
**פונקציית excecuteCommands:**  
פונקציה זו אחראית לביצוע כל פקודה כאשר מגיע הזמן שצריך לבצע אותה, ומעדכנת בהתאם את היעד החדש של הכלי הספיציפי.  
  
**פונקציית updateUav**:  
פונקציה זו אחראית בכל שלב בסימולציה לעדכן את המיקום (x ,y) של הכלי ואת הזווית בהתאם למצב שלו באותו רגע(מצב המתנה\תנועה אל היעד). אם הוא הגיע ליעד אז היא מעבירה אותו למצב המתנה ומעדכנת שהוא הגיע ליעד ומחשבת את הערכים בהתאם.  
**כאשר הכלי נע בקו ישר**:  
נעזר בנוסחה של תנועה במהירות קבועה של גוף:(נפרק את הנוסחה ע"פ רכיבים של ציר x וציר y):  
xnew=xold+v\*cos(azimuth)\*dt

ynew=yold+v\*sin(azimuth)\*dt

כאשר v – מהירות קבועה , dt זה צעד הזמן הנוכחי של הסימולציה

**כאשר הכלי נע אל היעד**:

את הazimuth החדש אנו נחשב כל פעם ע"י: הפעלת הפונקציה arctan על dx/dy כאשר dx הוא תוצאה של המרחק הנוכחי של שיעור הX של הכלי משיעור הx של היעד וdy הוא תוצאה של המרחק הנוכחי של y של הכלי משיעור הy של היעד. ונבדוק עבור כל כטבם בכל מחזור זמן בסימולציה האם הוא הגיע ליעד שלו.   
   
**כאשר הכלי הגיע אל היעד**:  
כדי לדעת אם כלי הגיע אל היעד נחשב את ההפרש בין המיקום הנוכחי של הכלי(x,y) לבין היעד שלו destX,destY)). ההפרשים האלה מייצגים את המרחק שהכלי צריך לעבור בציר x וציר y כדי להגיע ליעד שלו. ואז נשתמש במשפט פיתגורס(חישוב של 2 נקודות במישור), כדי לדעת את המרחק הנוכחי מהיעד. אם המרחק קטן מצעד הזמן הנוכחי של הסימולציה:(dt) \* המהירות שלו((v , זה אומר שבצעד הזמן הזה הכלי יגיע ליעד שלו.  
נשתמש בנוסחה של מיקום בתנועה במעגל ושינוי זווית בתנועה מעגלית כדי לחשב את הערכים.  
azimuth=v/r\*dt  
x=destX+r\*cos(azimuth)  
y=destY+r\*sin(azimuth)  
  
**פונקציית createSimulationCycleOutput:**  
כותבת את הפלט הנדרש עם הנתונים בכל צעד זמן של הסימולציה לתוך קובץ טקסט. החישובים בסימולציה נעשים ברדיאנים, ולכן, בשביל לכתוב את הזוית במעלות נמיר את התוצאה מרדיאנים למעלות.  
  
**simulation class:**  
class זה אחראי לקריאת 2 קבצי הקלט של הסימולציה והכנסה של הנתונים לתוך וקטורים , כך שנוכל להשתמש בהם כדי לאתחל את כל הכטבמים שלנו ולאתחל את הפקודות עבור הסימולציה.  
  
**RowData class:**

Class זה נועד לשמור בתוכו בנפרד כל פקודה בסימולציה מהקובץ **SimCmds.txt** ולהכניס אותה לוקטור. היה לי יותר נוח להסתכל על כל פקודה כך שהיא תהיה בתור אובייקט שמכיל את כל 4 הנתונים. בנוסף ישנה פונקציה (sortCommandsByTime) שממיינת את הערכים בוקטור לפי הtime שלהם כך שנהיה בטוחים שאנחנו מבצעים את הפקודות לפי סדר הזמן הכרונולוגי.  
  
\*מצורף המסלול של 2 הכטבמים עבור קבצי הקלט אותם צירפתי להגשה.  
