תיאור הבעיה:

ישנם רכבים בתוך מבוך שאינו ידוע מראש.  
כל רכב מתחיל בנקודת התחלה שונה, ומקבל נקודת סוף שונה, אליה הוא רוצה להגיע.  
כל רכב סורק את המבוך, ומנסה לעדכן בזמן אמת את תכנון המסלול שלו בהתחשבות במכשולים חדשים.  
לרכב יש מידע מדויק אודות איפה הוא נמצא ביחס לנקודת ההתחלה שלו.  
כאשר 2 רכבים קרובים מספיק זה לזה, הם יכולים לשתף מידע שהם צברו אודות המבוך.  
המטרה היא לכתוב אלגוריתם שמביא כל מכונית לנקודת הסוף, בזמן מינימלי.

מדריך למשתמש:

ניתן להכניס מבוך כרשימת קווים פוליגונליים בתוך קובץ json,  
באותו הקובץ ניתן לבחור את מספר המכוניות, ונקודות התחלה וסיום של כל אחת מהמכוניות (קיימת קבוצת מבוכים לדוגמה בתיקייה mazes)  
התוכנה בונה סימולציה של תנועת המכוניות במבוך והנסיון שלהן להגיע לנקודת הסיום.  
ניתן לבקש מהתוכנה לבחור באופן אקראי את אחד מהארגומנטים, או לבחור מצב דיפולטי נתון.

הערה:

לצערנו, נכון לעכשיו הקוד לא קונסיסטנטי לחלוטין מסיבה שלא הצלחנו לזהות עדיין ולכן ייתכנו תוצאות שונות מהותית עבור הרצות עם קלטים זהים. לכן במקרה ובו תוצאת האלגוריתם אינה מספקת, ייתכן שלאחר הרצה נוספת תתקבל תוצאה רצויה.

מדריך למתכנת:

הפרויקט בנוי ממספר קבצי פייתון ותיקיית מבוכים לדוגמה:  
הקובץ מכיל את כל הקבועים של האלגוריתם, מחולקים לפי קטגוריות.  
הקובץ מכיל פונקציות עזר, בעיקר לחישובים גיאומטרים ולדיסקרטיזציה של הסביבה.  
הקובץ המרכזי PRM שפותר את הבעיה לרכב אחד במבוך ידוע מראש.  
הקובץ בונה מפה שהיא קירוב של המבוך – בעזרת הסריקות שהמכונית סורקת.  
הקובץ מכיל את כל הנתונים לגבי הסביבה, המכוניות בה והמבוך.  
הקובץ מממש מחלקה של מכונית, ואחראי על כל הנתונים של המכונית, עדכון וקבלת ההחלטות שלה.  
הקובץ מכיל פונקציות שעושות שימוש ב על מנת לבנות את המבוך בסביבה.  
הקובץ מממש מחלקה של גרף ממושקל (וגם מחלקות של קודקודים וקשתות), ותומך בפעולות של הורדת והוספת קשתות וקודקודים.  
הקובץ מממש את אלגוריתם ה-   
התיקייה mazes מכילה מספר מבוכים דיפולטים, עם נקודות התחלה וסיום מוגדרות מראש.

הקובץ מכיל את כל הקבועים החשובים בפרויקט שייתכן שמספק קבצים ירצו לגשת אליהם.  
הוא מחולק למקטעים על פי מקורות הקבועים בקבצים, כל הקבועים מהקובץ למשל בבלוק אחד של הגדרות וקוד.

הקובץ   
אחראי על בניית המבוך בסביבת ה.   
הפונקציה המרכזית מייצרת ,בהינתן רשימה של מסילות פוליגונליות, את המבוך המתאים בסביבה של .

הקובץ :  
תפקיד הקובץ הוא לממש את הסביבה ולרכז את פעילות התוכנית, קובץ ה קורא ל שמתחיל ומריץ את הסימולציה.  
בקובץ 2 פונקציות מרכזיות,  
פונקצית ה שמייצרת את הסביבה: פותחת את סביבת ה, מאתחלת את גרף ה, וקוראת לבניית כל האובייקטים מסוג ,   
פונקציית ה שאחראית על לקיחת צעד של כל מכונית, עדכון הגרף וסיום התוכנית במקרה הצורך: אם כל הרכבים הגיעו ליעדן או שאחת המכוניות התנגשה במכונית אחרת או בקיר.

הקובץ :

תפקיד הקובץ לממש את המחלקה ,  
המחלקה מכילה את כל הנתונים על המכונית: המיקום שלה, הצומת על גרף הPRM עליה היא נמצאת, המטרה שלה.  
היא אחראית על בניית המכונית בסביבת ה ועל התנועה שלו בסביבה- בדיקת התנגשויות, בחירת מסלול, סריקת מבוך בעזרת , וקבלת החלטות (כמו זכות קדימה לרכב אחר).  
בדומה למחלקה , מממשת 3 פונקציות מרכזיות:  
 שמאתחלת משתנים, ומוסיפה את הרכב לסביבת ה   
 שמבצעת תנועה אחת (תנועה של יחיד) של המכונית, בהתאם לאלגוריתם ה ההולונומי.  
 שבודקת את המצב החדש של המכונית: האם התנגשה בגוף אחר או האם הגיעה לנקודת המטרה.

הקובץ :

תפקיד הקובץ לטפל בבניית המפה – ייצוג המבוך ב"מוח" של המכונית, בעזרתו מחושב גרף הPRM ובהתאם לגרף - המסלול המינימלי למטרה.  
אובייקט מסוג מכיל 2 תכונות:

1. מפה: ייצוג של המבוך בעזרת רשימה של מסילות פוליגונליות.  
   המסילה הפוליגונלית המיוצגת על ידי רשימת הנקודות

שקולה לרשימת הקטעים:

בפועל, למכשולים יש עובי שמוגדר בתור בקובץ .  
לכן בפועל כל ישר מתורגם למלבן עם עובי ואורך .

1. גודל: מימדי הסביבה, ערך הx המקסימלי (אנו מניחים שכל סביבה היא ריבועית וממורכזת ב)

בעת ריצת התוכנה, כאשר מכונית תפגוש במכשול, עליה לעדכן את המסלול שלה בהתאם.  
על מנת לעשות זאת, היא תעדכן את המפה שלה, באמצעותה מחושב מחדש הגרף, ובאמצעותו מחושב מחדש המסלול.  
סריקות של המכשול החדש יכולות להיות מאוד רועשות, מרכבות מספר קטעים שונים, ומגלמות בתוכן טעויות מדידה וקירובים.  
על מנת למצות מהמידע הזה את המידע הרלוונטי, אנו משתמשים באלגוריתם (RDP).  
האלגוריתם מקבל רשימת נקודות והופך אותן למסילה פוליגונלית כך שהמלבנים המתאימים לה מכילים את כל נקודות הסריקה.  
  
הפונקציה העיקרית שבעזרתה מוסיפים נקודות למפה היא .  
היא מסננת נקודות לא רלוונטיות, שכבר מוכלות במלבן ששייך למפה  
מפרקת את רשימת הנקודות הנותרת לנקודות שנבעו ממכשולים נפרדים, בעזרת קבוע מרחק בין 2 נקודות סמוכות – אם 2 סריקות סמוכות (בזמן) רחוקות זו מזו , הקוד יפרק את הישרים ויניח שנבעו מ2 מכשולים שונים.  
כעת לכל קבוצת נקודות, מחושבת המסילה הפוליגונלית בעזרת .

הקובץ :

תפקיד הקובץ הוא לממש את האלגוריתם לפי המאמר <https://www.cs.cmu.edu/~maxim/files/dlite_tro05.pdf> (הפסאודו-קוד הרלוונטי נמצא בעמוד 8, תרשים 9 ומומש על ידינו בקוד).

הוא הרחבה של האלגוריתם לניווט במבוך שנועד להתמודד עם ניווט בסביבה לא מוכרת ע"י כך שהוא מניח שלא קיימים מכשולים ובעת גילוי מכשול ועדכון מחירים של קשתות (אצלנו, הסרת קשתות קבועה או זמנית מהגרף), האלגוריתם מבצע שינויים מקומיים במסלול האופטימלי כדי להגיע למסלול מהיר חדש במהירות גבוהה יחסית.

בקובץ ממומשת גם המחלקה PriorityQueue שמממשת תור עדיפויות עבור האלגוריתם.

הקובץ :

תפקיד הקובץ הוא לנהל את ה-roadmap שבקוד (זהו לא בדיוק PRM כי הנקודות לא מוגרלות באקראיות אלא מפוזרות באופן אחיד במעין רשת). ה-roadmap הוא גרף מכוון וממושקל במרחב מדגם תלת-מימדי שכולל שתי דרגות למיקום ועוד דרגה לזווית המכונית. קודקודים בגרף המתאים מחוברים ע"י קשת אם המיקומים המתאימים לקודקודים קרובים מספיק אחד לשני ואם ניתן להגיע מקודקוד אחד לשני ע"י נסיעה של הרובוט בזווית קבועה של הגלגלים (עד כדי זווית מקסימלית שהוגדרה באזור הקבועים בקוד). מכיוון שהגרף בנוי בצורה של רשת, נובעת סימטריה בין המיקומים השונים ולכן על מנת להקל על החישוב המקדים (בניית הגרף ההתחלתי), אנו מחשבים את הקשתות המתאימות עבוד קודקוד ספציפי ומעתיקים את הקשתות האלו לשאר הקודקודים. במפה סטנדרטית מהמפות בתיקייה mazes הגרף ההתחלתי יכיל בערך מיליון קודקודים ובערך 10 מיליון קשתות, על כן תחילת הריצה שבה המכוניות יפגשו במספר רב של מכשולים חדשים תהיה ככל הנראה איטית יותר מהמשכה שבה הגרף קטן יותר ולא מתבצעים בו שינויים רבים.

המחלקה PRM בקוד מייצרת את הגרף המתאים שמשותף בין כל הרובוטים בריצה. כאשר רובוט נתקל במכשול, הקודקודים והקשתות שלא ניתנים לשימוש בגלל המכשול (כלומר המכשול חוסם אותם) נמחקים אצל כל הרובוטים והם מחשבים את מסלול התנועה שלהם מחדש בהתאם למידע החדש בעזרת האלגוריתם .