

פרויקט ברובוטיקה – תכנון מסלול לרובוט

עליכם לבנות תוכנה עבור רובוט PcBot, המצויד בחיישן לייזר (סורק), הכוללת את התכונות הבאות:

- א. קריאת מפה ותכנון מסלול.
- ב. תנועה רציפה לאורך המסלול תוך כדי המנעות ממכשולים.
- ג. מעקב אחר מיקום הרובוט לאורך המסלול.

התוכנה תקבל כקלט קובץ קונפיגורציה שיכלול את הפרמטרים הבאים:

- נתיב במחשב בו נמצאת המפה
- מיקום התחלתי של הרובוט על המפה (X, Y, Yaw)
- נקודת היעד (X, Y)
- גודל הרובוט
- רזולוציית המפה הנתונה (גודל כל פיקסל במפה בעולם האמיתי)
- רזולוציית ה-grid המכיל מידע על המכשולים בסביבה (גודל כל תא ב-grid בעולם האמיתי)

בשלב ראשון, על התוכנה לנפח את המפה (בהתאם לגודל הרובוט והרזולוציה), לקרוא את המפה ל-grid, לבנות מה-grid גרף מתאים ולהפעיל את אלגוריתם A^* לתכנון מסלול מהמקור ליעד (דאגו לכך שהמסלול לא יתקרב מדי למכשולים). לצורך מימוש השלב הנ"ל, השתמשו בקוד שכתבתם באבן דרך ב'.

בשלב הבא, בחרו נקודות ביניים (waypoints) לאורך המסלול שהחזיר A^* . נקודות אלו יסייעו לכם לוודא שהרובוט אינו סוטה מהמסלול שנקבע.

לאחר מכן, על התוכנה לגרום לרובוט לנוע לאורך המסלול שהחזיר A^* תוך שימוש בנקודות הביניים שבחרתם. ברגע שהרובוט מגיע ל-waypoint מסוים, התאימו את כיוון התנועה שלו על-מנת שיוכל להגיע ל-waypoint הבא (הגדירו tolerance עבור המרחק המינימלי של הרובוט מה-waypoint הדרוש כדי שייחשב שהוא השיג את ה-waypoint).

תוך כדי תנועת הרובוט, עליכם לחשב את מיקומו בעזרת particle filters עליהם למדנו בכיתה. עליכם לעקוב אחר Particles, ליצור חדשים ולמחוק ישנים בהתאם לאלגוריתם הנלמד. כמובן שמחיקת particles תהיה בהתאם לסף שתקבעו (רשמו אותו כ-define או בקובץ קונפיגורציה).

על הרובוט להדפיס את מיקומו תוך כדי תנועה בסביבה (תדירות הדפסת המיקום תיקבע על ידכם). מיקום הרובוט יהיה ב-3DOF: X, Y, Yaw . כאשר הרובוט מגיע לנקודת היעד שלו, עליו לעצור ולהדפיס את מיקומו הסופי.

כמו-כן עליכם לממש את התנהגויות התנועה במרחב והימנעות ממכשולים ע"ג מערכת קבלת ההחלטות שלמדנו.

להלן מחלקות מרכזיות אשר רצוי שתממשו:

- **Map**: מחלקה זו תכיל את נתוני המפה, גודל, רזולוציה, וכו'. כל תא במפה יכיל ערך 0 (free) או 1 (occupied). מחלקה זו תכיל מתודות לקריאת מפה מקובץ png לתוך grid וניפוח של המפה. כמו-כן תאפשר מתודות המרה ממיקום אמיתי בעולם לתא במפה והפוך. כמובן שתמצאו לנכון להוסיף מתודות כאוות נפשכם.
 - **PathPlanner**: מחלקה זו תהיה אחראית על ביצוע תכנון המסלול על-ידי הפעלת אלגוריתם A*.
 - **WaypointsManager**: מחלקה זו תהיה אחראית על בחירת נקודות הביניים על-גבי המסלול שהחזיר אלגוריתם A*.
 - **Particle**: מחלקה זו תכיל מידע אודות Particle – מיקומו וה-belief שלו. מחלקה זו תכיל מתודת update אשר תעדכן את ה-belief, והמיקום של ה-Particle. יש לממש מתודות נוספות כגון יצירת particles חדשים מה-Particle הנוכחי.
 - **LocalizationManager**: מחלקה זו תנהל את עדכון ה-Particles. מתודת ה-update של מחלקה זו תעבור בלולאה על כל ה-Particles ותריץ update עבור כל אחד מהם. כמוכן תמחק ותיצור Particles בהתאם.
 - **ConfigurationManager**: מחלקה זו אחראית על קריאת הנתונים מקובץ הקונפיגורציה.
 - **Manager, Behaviors, Robot**: מערכת קבלת ההחלטות שלמדנו.
- שימו לב: חיבור בין מערכת קבלת ההחלטות לאלגוריתמיקת המיקום תתבצע במתודת ה-run של ה-Manager. בכל איטרציה בלולאה שב-manager תתבצע קריאה ל-update של LocalizationManager.
- עליכם להריץ את המערכת הן בכלי הסימולציה והן ע"ג הרובוטים במעבדה.
- בנוסף:** בונים יינתן עבור ציור של המסלול שהחזיר A* על-גבי המפה הנתונה.

נוהל הגנה על הפרויקט

תוכלו להיבחן על הפרויקט באחד מבין שני מועדי הגנה הבאים:

1. מועד הגנה ראשון יתקיים בשיעור האחרון של הסמסטר.
2. מועד הגנה נוסף יתקיים במהלך תקופת הבחינות. תאריך מדויק יפורסם בהמשך.

ניתן לבצע את הפרויקט בצוותים של עד שלושה סטודנטים. במהלך ההגנה על כל סטודנט להיות מוכן לענות על שאלות הנוגעות לכלל הפרויקט ולא רק על החלק שהוא כתב.

אנא הקפידו על כללי הנדסת תוכנה (מודולריות של הקוד, גמישות לשינויים) וכתובה של קוד נקי וקריא.

בהצלחה לכולם,

יהודה ורועי