נושאים ברובוטיקה ובינה מלאכותית

מנחה: ד"ר עמי ברלר

אוניברסיטת בן גוריון

סמסטר סתיו תשע"ז

קבוצה 5

לירז רייכנשטיין

גסאן נבארי

אמיר טל

גלעד השמנד

**הקדמה**

**המשימה:** הפעלת רובוט בסביבת הדמייה של עולם פתוח, Gazebo, בו מבוך עם שלוש קוביות בצבעים שונים. הרובוט ימצא את הקבויות בתוך חדרי המבוך ויעמוד מולן, לפי סדר אקראי המתקבל בתחילת הרצת התכנית.

על הרובוט לזהות את המבוך, לזהות את הקביות וצבעיהן ולהחליט על סדר פעולה לפי המידע שצבר עד כה.

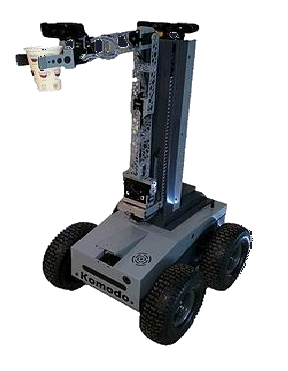
**הערות:**

* המיקום ההתחלתי של הרובוט אקראי.
* סידור צבעי הקוביות אקראי.
* סדר הביצוע הנדרש אקראי. (בפועל - נבחר באקראי ע"י המרצה).

**הרובוט:**

נשתמש ברובוט מסוג Komodo with elevator של חברת RoboTiCan.

לרובוט הזה מודלים וספריות מוכנות לשימוש בסימולטור Gazebo בנוסף לפונקציונליות שמתאימה לביצוע המשימה שלנו.



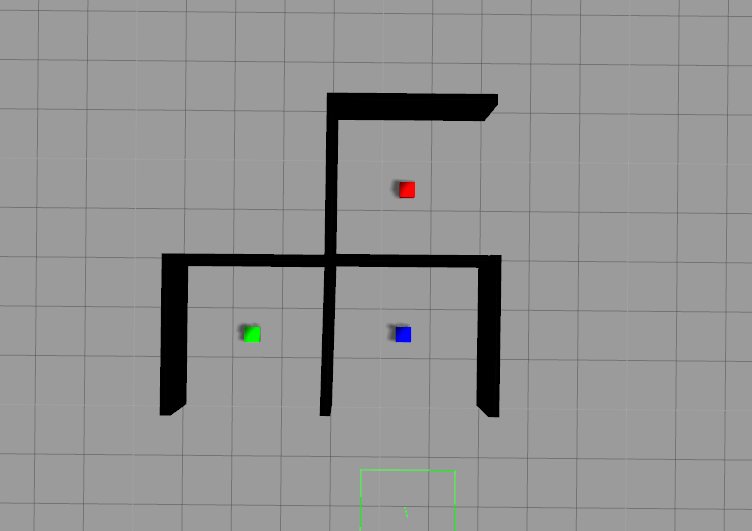
**סביבת העבודה:**

סביבת הסימולציה שלנו היא Gazebo, מערכת ההפעלה של הרובוט היא ROS וקבצי ההפעלה והתקשורת נכתבו בpython.

במהלך הפרוייקט עבדנו במעבדת הרובוטיקה בבניין המתמטיקה באוניברסיטת בן גוריון.

**המבוך והסביבה הוירטואלית:**

יצרנו את כל המפות דרך הגזיבו ובעריכה ישירה של קבצי xml. המפות כוללות מבוך בעל שלושה תאים ובכל תא קובייה בצבע אחר.



**אופן מימוש הפתרון:**

עלינו לממש אלגוריתם שמסוגל להתמודד עם כל סידור קוביות ועם נקודות התחלה שונות במפה. כלומר, על האלגוריתם לדעת להתמודד עם אתגרי מרחב וסדר פעולות משתנה.

השתמשנו בnode אחד דרכו נעשית כל העבודה מול הרובוט. בעזרת הnode שלנו קיבלנו נתונים מחיישני הרובוט השונים (מצלמה, לייזר), עיבדנו אותם ולפי הנתונים גם שלחנו פקודות הכוונה אל הרובוט.

**אלגוריתם פעולה ראשי**

1. אתחולים
2. בצע אלגוריתם סריקה ראשונית.
3. האם רואים פחות משני צבעים?
   1. תיקון עמדה
   2. חזרה לשלב 2
4. הסק את סדר הביקור בתאים לפי הידע שנצבר עד כה.
5. האם אחד משני הצבעים הנראים מופיע ראשון בסדר? /\* בשלב הזה כבר רואים שני צבעים וידוע לנו סדר הפעולות\*/
   1. בצע אלגוריתם סריקה והתקדמות לקוביה.
   2. הוצא מהתור את הצבע שנמצא || קדם אינדקס.
   3. בצע אלגוריתם מעבר לתא הבא.
   4. הוצא מהתור את הצבע || קדם אינדקס.
   5. בצע מעבר לתא הבא.
6. אחרת -
   1. בצע אלגוריתם עקיפת מבוך.
   2. סרוק עד מציאת הקובייה.
   3. בצע אלגוריתם סריקה והתקדמות לקוביה.
   4. הוצא מהתור את הצבע שנמצא || קדם אינדקס.
   5. בצע אלגוריתם מעבר לתא שני, מבוסס על מיקום נוכחי ומידע שהושג בסריקה ראשונית.
   6. הוצא מהתור את הצבע שנמצא || קדם אינדקס.
   7. בצע מעבר לתא שלישי, מבוסס על מיקום נוכחי ומידע שהושג בסריקה ראשונית.

**אלגוריתמי עזר:**

**אלגוריתם סריקה ראשונית ושמירת תוצאות:**

1. בצע סיבוב שלם עד זיהוי שתי קוביות בעלות צבעים שונים.
2. אם לא נמצאו שני צבעים:
   1. בצע צעד קדימה || שני צעדים אחורה לסירוגין.
   2. חזור לשלב 1.

**אלגוריתם עקיפת מבוך:**

1. קדם את הרובוט לנקודת המוצא הקבועה, מול התא הקידמי-שמאלי.
2. צא בתנועה קבועה לכיוון התא האחורי.

**ניתוח תמונה:**

ע״י שימוש בספריות openCV ביצענו המרה של התמונה המתקבלת ממצלמת הרובוט לתמונה בפורמט שניתן לעבוד איתו עם הפונקציות של openCV (כמובן ע״י פונקציה של openCV). לאחר מכן ביצענו המרה של התמונה מתמונת בגווני צבע של rbg (אדום-כחול-ירוק) לתמונה בפורמט של hsv (צבע, רוויה וערך ביניהם) כיוון שהדבר אפשר לנו פונקציונליות קלה יותר בעיבוד התמונה. יצרנו מסיכה – מעין מסננת – לכל צבע כך שאם נשים את המסיכה האדומה (למשל) על התמונה בפורמט hsv נקבל רק את הפריטים האדומים בתמונה.

בשלב הראשון שמנו את כל המסיכות על התמונה ובדקנו האם נותרנו עם 2 קוביות בתמונה. ברגע שזה קרה, עדכנו את מיקומי הקוביות במערך ששמרנו בnode. לאחר מכן לפי סדר המעבר בין הקוביות, בכל פעם מתבצע סינון של התמונה ע״י מסיכה של הצבע הנוכחי של הקוביה בה אנו מעוניינים. לפי מיקום הקוביה בתמונה, אנו קובעים את כיוון ההתקדמות של הרובוט (השאיפה להסתובב עד שהקוביה במרכז התמונה, ורק אז להתקדם ישר לכיוונה).

**קבצים:**

כתבנו שני קבצי python ועיצבנו 6 מפות.

המפות נמצאות בתקייה project/worlds/.

Our\_robot.py - הקובץ נמצא בתקייה project/scripts/. הוא מייצג את ה node שלנו, כאן מתבצע עיבוד הנתונים וההוצאה לפועל של כל פעולות הרובוט.

Launch.py - הקובץ נמצא בתקיית הראשית של הפרוייקט project/. הקובץ זה טיפל בפעולת הרנדומיזציה של נתוני העולם בעת האתחול וביצע את פעולת ההרצה עם הנתונים האקראיים שהוגרלו.

**סיכום:**

הפרוייקט הזה נתן לנו הזדמנות לעבוד כצוות. עבדנו כל משך הפרוייקט במעמדת הרובוטים בבניין מתמטיקה, קומה (1-). מלבד התחלה איטית, העבודה התנהלה בקצב אחיד שהוגבל על ידי למידה של סביבה חדשה. חילקנו בצוות את העבודה למטלות ליחיד ולזוגות. הדבר נתן לנו אפשרות לעבוד לבד ויחד לסירוגין. כולם היו שותפים לבעיות ולפתרונות, דבר שנתן לנו יתרון בעת היעדרות חבר מהצוות.

עמדנו בפני מספר אתגרים בפרוייקט. תחילה היה בלתי ניתן להתחיל את הפרוייקט עקב בעיות טכניות רבות במעבדה. משנפתרו, האתגר הראשי, כמו שכבר הזכרנו, היה למידה עצמאית של סביבות העבודה החדשות לנו. ברגע שהתגברנו על המכשול הראשי העבודה התקדמה וההתמודדויות היו ליצור פתרון לאתגר דינמי ולצייד את הרובוט במספיק כלים להתגבר על המצבים השונים מולם יתמודד.

מבחינת הלמידה האישית שלנו ניתן להגיד שלמעשה כל דבר שעשינו בפרוייקט היה חדש (לנו) ולכן למדנו הרבה דברים. האתגר איפשר לנו ללמוד שפה, סביבת תכנות ומערכת הפעלה חדשים לנו. בנוסף התמודדנו מול אתגר דינמי, דבר שאינו בהכרח חדש לנו, אך בהחלט היווה נקודת דיון ולמידה בין חברי הצוות.

הוראות הפעלה:

1. יש להעתיק את תיקיית הפרוייקט לתקייה catkin\_ws/src. (לשמור בתיקיה project)
2. לבצע את הפקודה: cd ~/catkin\_ws.
3. לאחר מכן: chmod –R 777 src/project
4. לעבור למצב bash. (במידת הצורך)
5. לבצע: catkin\_make.
6. לעבור לתיקיית הפרויקט: cd src/project.
7. להריץ את הקובץ launch.py. (במידת הצורך: python launch.py)
8. לחכות שהגזיבו יעלה ולבחור ב bash סדר הגעה לקוביות.

קישורים:

1. קישור לפרוייקט בgoogle drive, בתיקיה ישנו קובץ read\_me.txt מפורט: https://drive.google.com/drive/folders/0B3Y2GM5aZuhnTnEwTmJXcDFzaHM?usp=sharing