

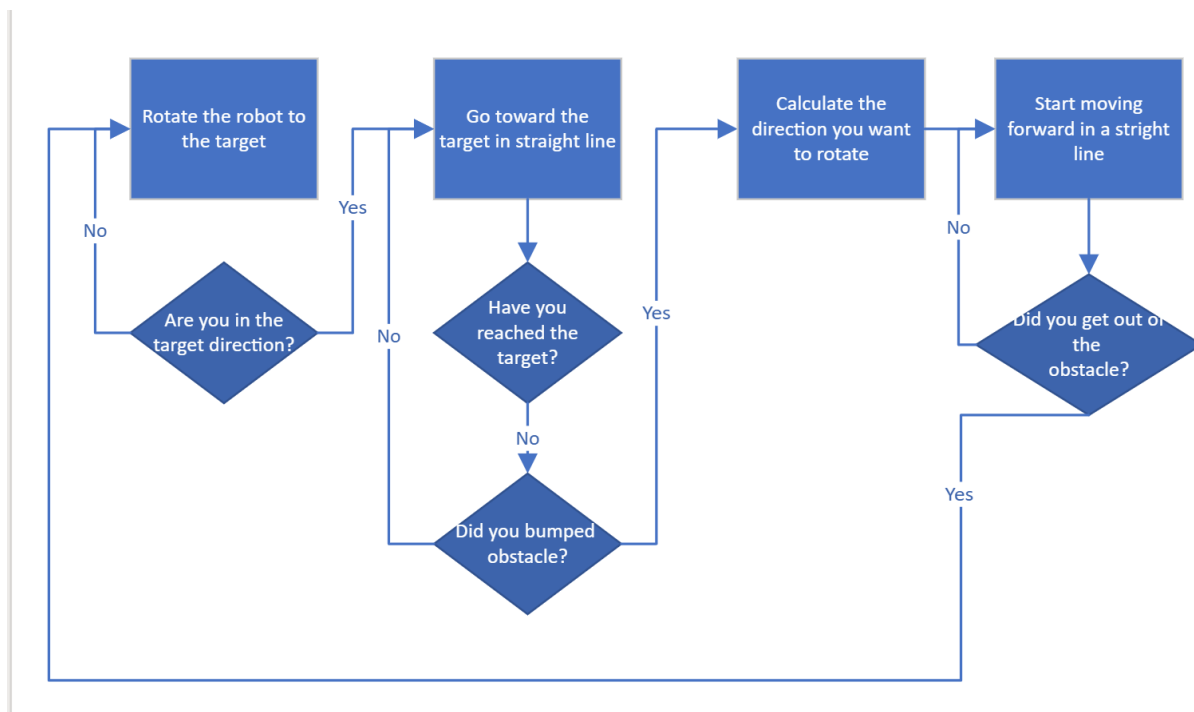
Yaniv lukin 213377013

Io Trabelssi 316334994

:Pseudo code

1. תסובב את הרובוט לכיוון המטרה
a. תבדוק האם הרובוט בכיוון של המטרה: אם לא המשיך להסתובב, אם כן אז תעבור למצב הבא
2. התחל להתקדם לכיוון המטרה בקו ישר
3. האם הגעת למטרה?
a. אם כן אז עצור אם לא תמשיך אלה
4. האם התנגשת באובייקט?
a. אם כן אז תקבע את הכיוון שאתה רוצה לסובב את הרובוט אליו אם לא אז תחזור ל2
5. לאחר שקבעת את הכיוון שאתה רוצה להסתובב תתחיל להתקדם ישר עד שאתה יכול לצאת מהמכשול
6. האם סיימת את המכשול?
a. אם כן אז תחזור ל1 אם לא אז חזור ל5

Flowchart של הקוד:



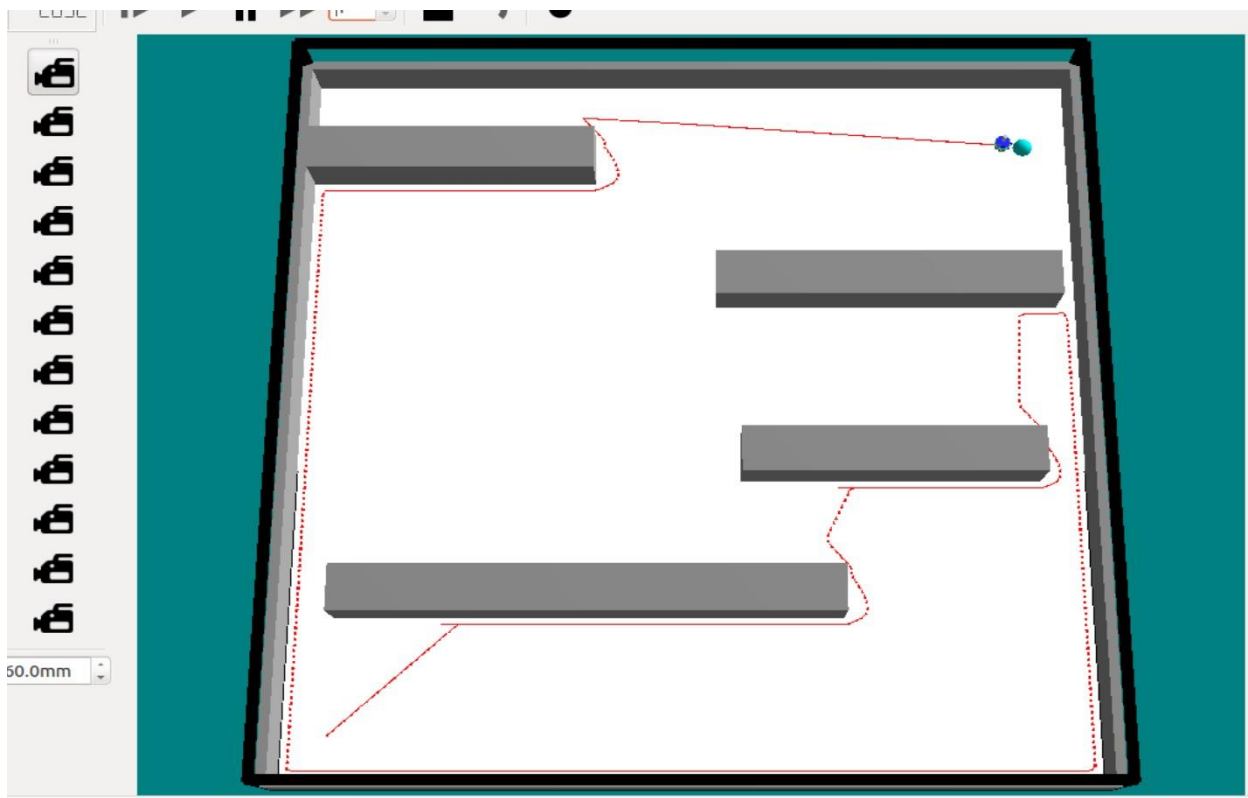
כיצד בחרנו את Threshold?:

אז בקוד שלנו היה לנו מספר thresholds שונים כמו כמה רחוק הרובוט היה יכול להיות מהמטרה שאותו בחרנו שהיה 0.05 מפני שרביעיית מהרדיוס של הרובוט הרגיש לנו נכון כי זה אומר שכבר רוב הרובוט נמצא במטרה וזה תוצאה מספקת בשבילנו

את המהירות סיבוב המינימלית שהיא 5, פה לא הייתה לנו הצדקה לזה יותר מדי פשוט נדיר שזה היה 5 ואם זה היה ב 1 זה ייקח לנו יותר מדי זמן 5 נתן לנו תוצאות מספקות ומהירות מספיק טובה, היה חשוב לנו שהמהירות תהיה מסיק נמוכה שלא יקרה מצב שבו אנחנו מדלגים על הזווית הרצויה כמו שקרה לנו במהירויות יותר גדולות.

וקירוב של הזווית של הרובוט לכיוון המטרה, פה בחרנו שזה היה עד 1 כלומר אם הזווית קטנה מ 1 או גדולה מן 359 בסכום. את הזווית הזו שוב רצינו לבחור משהו שייתן לנו קירוב למטרה אבל מה שהיה באמת חשוב לנו זה שהרובוט לא ידלג בטעות על הסכום הזה ואז יסתובב פעמים. לדוגמה אם הזווית של הרובוט היא 538.9 והוא עולה ב 2.3 מעלות כל פעם אז הוא ידלג על הזווית שרצינו ואז נקבל שהרובוט יסתובב סתם פעמים. ראינו תוצאות טובות עם קירוב של 1 וכמו כן אנחנו מעדכנים את הזווית של הרובוט כל כמה שניות ככה שזה לא משפיע עליו יותר מדי שככה שגם אם הוא בחר סטייה יחסית רחוקה אז תוך כמה שניות היא תדעךן לזוויות יותר עדכניות.

הרצת הקוד:



ניתן לראות שהרובוט אכן מגיע למטרה. אם זו הוא לא מגיע בדרך האופטימלית שאנחנו מצפים ממנו.

הסיבה לכך היא מפני שבנקודה מסויימת הרובוט שלנו מתנגש בקיר אך הוא חושב שהקיר הוא מכשול אחד ענק ולכן הוא ממשיך להתקדם על כל הקיר עד שהוא מצליח לצאת ממנו וכשהוא מצליח לצאת ממנו הוא מגיע למטרה שלו.