פרויקט סיום – רובוטיקה HIT

רובוט שמצייר לפי תמונה

חברי הקבוצה : ספיר נאוגאוקר 206542375

דן אברין 204358394

דניאל בוגדרי 315846501

מהות הפרויקט – תיאור:

מטרת הרובוט היא לצייר לפי תמונה או כמות לחיצות על הכפתור.

הרובוט מסוגל לצייר תרשימים ידועים מראש (ריבוע, משולש, קו, בית) לפי כמות לחיצות וכיבוי עצמי בעת לחיצה ארוכה (3 שניות).

בנוסף לכך, לרובוט מחוברת מצלמה אשר מזהה צבע, לפי הצבע המתאים היא שולחת לרובוט את הצורה הנדרשת לציור.

הפעלה של הרובוט:

הפלטפורמה בה בחרנו להשתמש היא EV3 Lego עם כרטיס SD של שפת פייתון.

לטובת כך נדרשנו להתקין image מתואם של פייתון על כרטיס SD ולעלות את האבן החכמה עם התוכנה, בנוסף לכך גם הגדרנו סביבת עבודה ב VS code באמצעות התקנה של פייתון וספריות מוכנות ל EV3.

**פירוט שלבי הפיתוח:**

מצאנו שרטוט מוכן מראש של הרובוט ובנינו אותו על פיו, היו חסרים שלושה חלקים אשר נדרשנו להדפיס אותם במדפסת תלת מימד.

התכנון הראשוני של הרובוט היה יצירת רובוט שמצייר תמונה אשר מציגים לו. הרעיון היה שילוב של קוד עיבוד תמונה יחד עם התנועה של הרובוט. לאחר חקירה על הרובוט, הבנו כי הזרועות של הרובוט והשילוב של שני מנועים ביחד, יוצרים זוויות וחישובים מתמטיים מסובכים.

מקוצר הזמן הוחלט לשנות את הדרך פעולה של הרובוט, אשר שומר על התפקיד שלו – מקבל קלט מהמשתמש ומסוגל לצייר תרשימים ידועים מראש על פי הקלט.

**פירוט חלקי העבודה ואתגרים:**

1. תכנון וכתיבה של תנועות הרובוט –

נדרשנו להבין את המרחב תנועה של הזרועות והטווח שהם יוכלו להגיע ולצייר. בנוסף לכך, גם הבנה ושימוש נכון במנוע הקטן (המשמש להרמה של הזרועות עם העט מהדף).

נתקלנו בלא מעט אתגרים כאשר מספר חלקים של הרובוט לא ישבו בצורה טובה. למשל הזחל אשר הניע את הזרוע השמאלית - גלגל השיניים לא ישב בדיוק על הזחל וגרם לקפיצות של הזרוע, פתרנו את הבעיה באמצעות תוספת קטנה לזחל.

בנוסף לכך, גם המנוף של המנוע הקטן אשר תפקידו להרים/להוריד את הזרועות, לא ביצע את העבודה שלו במלואו הכוח. הוספנו עוד מספר חלקי לגו בשביל ליצור משטח דחיפה רחב יותר ובכך הגדלנו את הכוח המופעל על ידו.

בנוסף, מנגנון ההרמה היה מנתק לעיתים את הקוביה החכמה מן גוף הרובוט ולכן נעזרנו באזיקון על מנת לקבע את הקוביה החכמה אל הגוף.

1. מילון שרטוטים –

היינו צריכים להגדיר מראש אילו צורות רצינו לצייר עם הרובוט, לשם כך עבור כל צורה חישבנו את תנועות המנוע הנדרשות ( קו, 2 קוים, משולש, מרובע, בית) ויצרנו סט פקודות שנדרש לבצע לפי סדר, עבור כל צורה, וסט הפקודות הזה נשלח לפונקציה ייעודית שמבצעת את הפעולה ושולחת פקודות אל המנועים.

1. מצבי עבודה -

3.1 מצב מצלמה -

רצינו להשתמש בעיבוד תמונה על מנת לזהות צורות גאומטריות ולפיהן לצייר צורות בהתאם, אך נתקלנו בקושי בתהליך עיבוד התמונה שכן המצלמה שעובדת עם הev3 אינה מעבירה את התמונה עצמה אל הקוד\רובוט ויודעת בעיקר לזהות צבעים\עצמים ולבצע אחריהם עקיבה (כפי שמפורט במסמך הרשמי של המצלמה).

לכן, החלטנו להשתמש במצלמה בתור חיישן צבע, ולימדנו את המצלמה את הצבעים שרצינו להשתמש בהם כחול, אדום, ירוק – לפי כל צבע שנקלט במצלמה, יבוצע ציור אחר שהגדרנו מראש – \*\*ראוי לציין שהמצלמה רגישה לתאורה ולכן עדיף לתכנת אותה בסביבת העבודה המיועדת.

ניתן להתממשק אל המצלמה בעזרת :

(א) פרוטוקול I2C (ב) להגדיר את המצלמה כסנסור של EV3 (Ev3devSensor)

אנחנו השתמשנו בדרך (ב) שהייתה קצת יותר פשוטה.

כתובת לאתר המצלמה:

<http://www.mindsensors.com/vision-for-robots/191-vision-subsystem-v5-for-nxt-or-ev3-with-fixed-lens>

קישור להורדת תוכנת המצלמה אשר בעזרתה מלמדים אותה צבעים:

<http://www.mindsensors.com/largefiles/NXTCam5/>

כתובת לקובץ pdf המסביר על אופן תכנות המצלמה, ודרכי התממשקות אל המצלמה בעזרת פרוטוקול i2c (נמצא בסוף הקובץ):

[www.mindsensors.com/pdfs/NXTCam5-User-Guide.pdf](http://www.mindsensors.com/pdfs/NXTCam5-User-Guide.pdf)

הסבר התממשקות אל המצלמה כEv3devSensor-:

[http://docs.ev3dev.org/projects/lego-linux-drivers/en/ev3dev-stretch/sensor\_data.html#ms-nxtcam5](http://docs.ev3dev.org/projects/lego-linux-drivers/en/ev3dev-stretch/sensor_data.html" \l "ms-nxtcam5)

* 1. מצב כפתור –

הגדרנו שליטה על פיקוד ההדפסה על בסיס מספר לחיצות על חיישן לחיצה:

|  |  |
| --- | --- |
| מספר לחיצות | ציור מבוצע |
| 1 | קו |
| 2 | 2 קוים |
| 3 | משולש |
| 4 | מרובע |
| 5 | בית |

מסקנות והמלצות להמשך הפיתוח:

**במשך תהליך העבודה הגענו למספר מסקנות:**

1. **השרטוטים לא היו הכי מדויקים ולכן הזרועות ומומנט לא היו מספיק טובים. שדרוג של החלקים וההבנה של הבעיה עזרה לנו למצוא פתרונות יצירתיים.**

**בהמשך, נוכל לפתח את הרובוט במספר דרכים שונות:**

1. **יצירת מילון מושגים רחב יותר עם יותר דוגמאות ושרטוטים מסובכים. הדבר יאתגר גם את המפתח וגם את הרובוט בניסיון להבחין בין יותר תרשימים.**
2. **שימוש במצלמה ועיבוד תמונה טוב יותר, כרגע המצלמה משתמשת כחיישן צבע, ניתן לשפר אותה או להחליף במצלמה אשר יכולה לזהות צורה. מכאן, שילוב של עיבוד תמונה יחד עם המילון יאפשר לצייר לפי תמונה ולא צבע.**
3. **ניתן לשלב זיהוי תמונה לפי SVG (פורמט תמונה אשר מתורגם לוקטורים), ובעזרת שינוי בקוד וחישוב זוויות מתאימות, ניתן לצייר תמונה על ידי הרובוט.**