

פורטפוליו

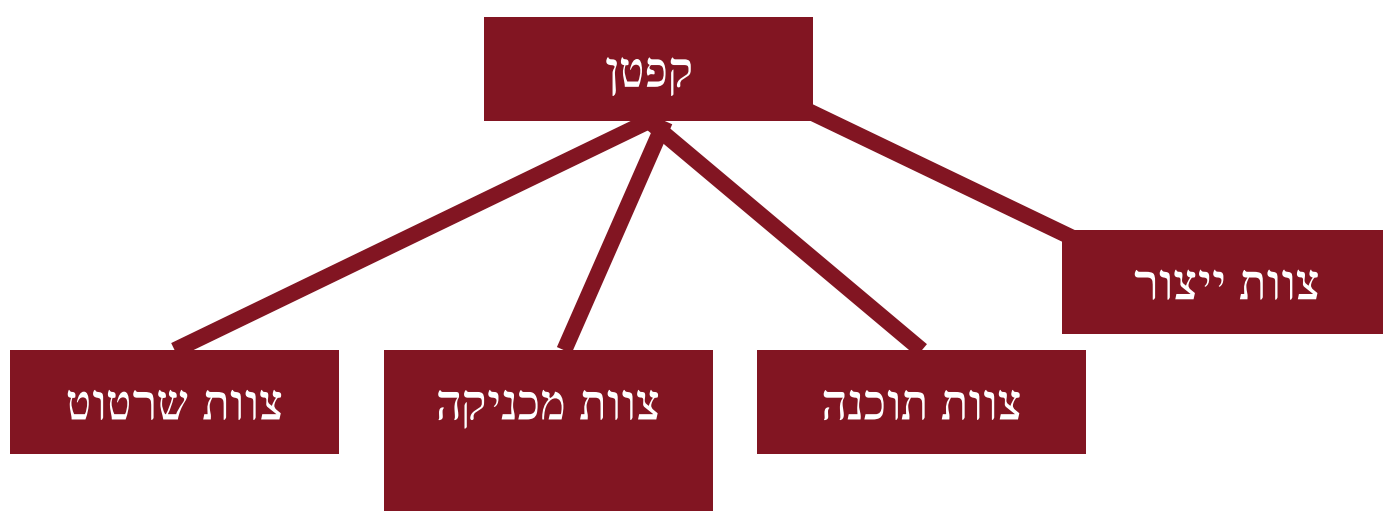


על הקבוצה



אנחנו קבוצה Makers Assemble 13146 מבית ספר תיכונט שבתל אביב. הקבוצה שלנו מורכבת מ-15 ילדים וילדות בכיתה ח', עבור כולנו זאת השנה השניה בארגון FIRST. בשנה שעברה השתתפנו בליגת ה-FLL. את המתחם אנחנו חולקים עם קבוצות ה-FLL.

הקבוצה שלנו מחולקת לצוותים וקפטן שפועלים זה עם זה במטרה להשיג את התוצאות המקסימליות תוך שמירה על תהליכי קבלת החלטות.



על הקבוצה

חלוקת חברי קבוצה לתפקידים

צוות מכוניקה	צוות שרטוט	צוות תוכנה	צוות ייצור
ליעוז מרואני	גיא ציגרינסקי	ליאור זוסמן	בן אחיעזרא
אלה סוסנה	גיא אהרוני	איתמר אורן	אורי רוזנברג
עילי גריאני		סהר ברשם	רועי מוגרבי
עילי בירן		דור בייפוס	
הראל שוסטיאל		עמית דגן	



יצור

בחזר היצור שלנו יש כמה וכמה מכונות אבל אנחנו לא משתמשים בכלם, שתי המכונות שאנחנו משתמשים בהם הכי הרבה הם "המכרתה", שהיא מכונה שמקצרת ומעגלת צירים. אנחנו משתמשים בה הרבה כי היא מאפשרת לנו להגיע בדיוק מרבי, אפילו לא ס"מ אחד פחות מהמידה הנדרשת. יש לנו גם את "הכרסומת", שהיא מכונה לחור המתקיימת בבלוקים. אנחנו משתמשים בה כאשר צריך לבצע חורים מדויקים במתכת. יש לנו עוד מספר מכונות, אך השימוש בהן פחות תדיר. כל אחד מאיתנו עבר הכשרה על מנת ללמוד איך לעבוד עם המכונות בצורה בטיחותית ולמנוע סכנות.



אסטרטגיה

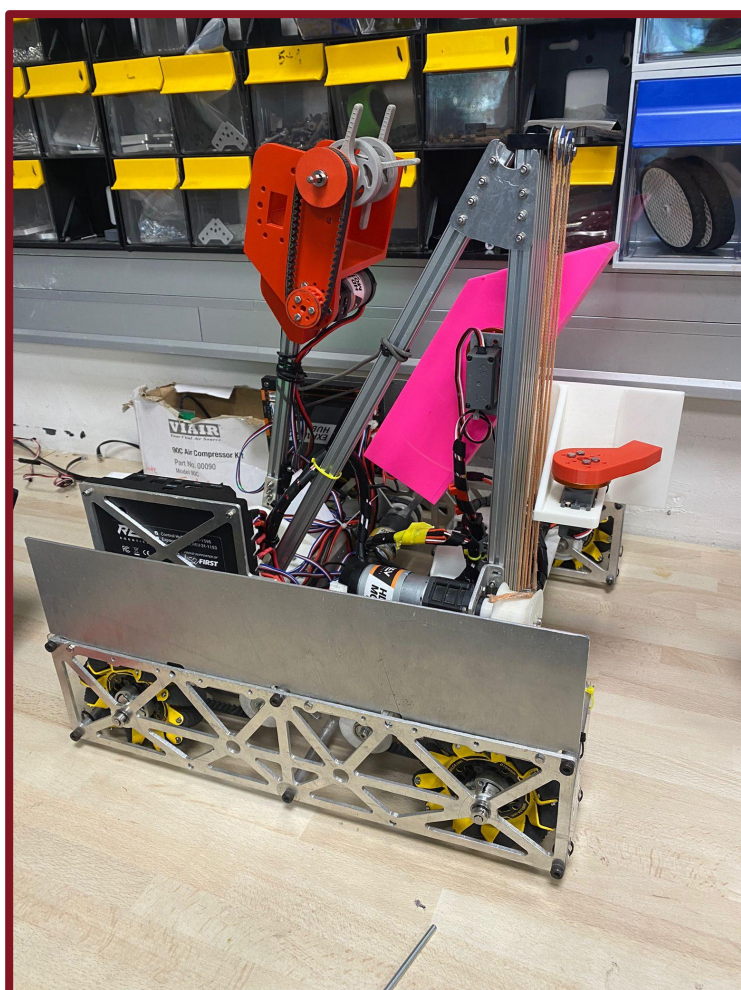
התחלנו בגיבוש האסטרטגיה שלנו על ידי לימוד מעמיק של חוקי המשחק, תוך בחינה של כל המשימות האפשריות ודרכי ביצוען. יצרנו טבלה המפרטת את כל המשימות, הניקוד שהן מניבות והזמן הנדרש לביצוען, מה שסייע לנו להבין אילו משימות הן הקריטיות ביותר להשגת התוצאה הרצויה. לאחר גיבוש האסטרטגיה, עברנו לשלב הגדרת דרישות המערכת לרובוט. חילקנו את הרובוט לשלוש מערכות עיקריות – הרמה, הנעה ואיסוף – כל מערכת קיבלה תפקידים ברורים ומטרות מוגדרות בהתאם לצרכים השונים של המשחק בנוסף, פיתחנו "טבלת מטרות", שמסווגת את הדרישות לשלוש קטגוריות עיקריות: "חובה", "צריך" ו"נחמד". קטגוריית "חובה" כוללת את כל הפעולות והמאפיינים הקריטיים שהרובוט חייב לבצע על מנת לפעול בצורה תקינה ולהשיג את מטרות המשחק. קטגוריית "צריך" מתארת שיפורים שיכולים לשדרג את הרובוט ולייעל את הביצועים שלו, אך הם אינם קריטיים להשגת היעדים הבסיסיים. קטגוריית "נחמד" מתארת תוספות אפשריות, שיכולות להוסיף ערך, אך יתבצעו רק אם הזמן והמשאבים יאפשרו זאת. גישה זו אפשרה לנו לתכנן בצורה מסודרת וממוקדת, תוך שמירה על סדרי עדיפויות ברורים וגמישות למהלך העבודה במידת הצורך.

דוגמא - דרישות מערכת ההרמה:

חובה	צריך	נחמד
הגעה לסל העליון ספסימן עליון	הגעה לסל התחתון ספסימן נמוך	טיפוס
5	3	

תהליך תכנון הרובוט

בהתחלה העלנו רעיונות לאיך הרובוט והמערכות יראו
אחר כך התחלנו לבנות את המעלית ואת מערכת איסוף
לשאר המערכות לא הכנו סקיצה מכיוון שהיו קלות ולא היה צורך בדוגמא להם
ראשון בנינו את מערכת הנעה ואז את שאר הרובוט



מערכת הנעה

השתמשנו בהנעת מכנום המאפשרת תזוזה לכל הכיוונים ללא צורך לשנות את הפוזיציה של הרובוט

אורכה של מערכת ההנעה שלנו היא 45 ס"מ. ה control hub וה expshion hub מחוברים בצד הפנימי של הפלטה, מה שמאפשר לנו גישה מיידית לבקרים.

השתמשנו בפרופילים שיצרנו במתחם שלנו וזה סייע בכך שבמקום לקנות פרופילים ולחכות הרבה זמן ולהיות מוגבלים לפי היצרן הפורפילים היו מוכנים בזמן שרצינו ולמדנו תוך כדי הייצור שלהם על מכונת ה CNC ושיפרנו את יכולות העיבוד השבבי שלנו

מערכת ההרמה

מערכת המעלית פועלת באמצעות חוט ארוך שנמשך לאורך כל שלושת השלבים על גלגלות ומסתובב במהירות על כננת, מה שמוביל להעלאת הצירים.

בהתחלה, ערכנו חישובים ובדיקות על מנת לקבוע את מספר השלבים הדרוש לכלל המערכת, כך שהמעלית תוכל לעמוד בדרישות הספציפיות שלנו. בנוסף, בחנו את גובה כל שלב, מתוך מטרה לוודא שנוכל להגיע בצורה נוחה ואופטימלית לארבעת הגבהים שאנחנו צריכים – מנמוך לגבוה. היה חשוב לנו לוודא שהמנוע יוכל להרים את המעלית בצורה מהירה ויציבה, במיוחד כיוון שיש לנו משקל בסוף המעלית. כדי להגיע לגבהים המבוקשים בצורה מיטבית, החלטנו לבנות את המעלית עם כמה פרופילים, כאשר כל פרופיל עולה כאשר הקודם הגיע לגובה המקסימלי שלו.

מערכת איסוף

אחרי שניתחנו את המשחק בצורה מעמיקה, נדרשנו להגדיר את הדרישות שלנו עבור מערכת האיסוף. היה לנו חשוב מאוד שהמערכת תוכל לפעול בצורה היעילה ביותר, כך שהיא תוכל לאסוף את החלקים במהירות, בלי לעכב את זרימת המשחק. חיפשנו פתרון שיהיה גם נוח לתפעול וגם לא ידרוש הרבה זמן או מאמץ בפועל, כך שהמערכת תהיה מהירה, מדויקת ואפקטיבית לאחר דיונים ושקלול האפשרויות השונות, הגענו למסקנה שפתרון אינטייק יתאים לנו בצורה הטובה ביותר. האינטייק שלנו מורכב משני גלגלים. כאשר מסובבים אותם יחד – גלגל ימין עם כיוון השעון וגלגל שמאל נגד כיוון השעון – הסאמפל נכנס. כדי להוציא את הסאמפל, פשוט מחליפים את כיוון הסיבוב: גלגל ימין נגד כיוון השעון וגלגל שמאל עם כיוון השעון. ברגע שהחלק נכנס לחלל הגלגל, הגלגלים עוצרים את סיבובם כדי למנוע תקלות או סיבוכים. ל

היתרון במערכת כזו הוא שבזכות הפשטות של העיצוב והעבודה המהירה שלה, אנחנו יכולים להמשיך במשחק מבלי להשקיע הרבה זמן בתהליך האיסוף עצמו. המערכת פועלת בצורה אוטומטית ברוב הזמן, כך שהיא חוסכת זמן ומביאה אותנו לתוצאות בצורה מהירה ויעילה.

תוכנה

את השנה פתחו בללמוד JAVA שאיתה אנחנו נתכנת את הרובוט. אנחנו עובדים בתוך Android Studio ומשתמשים בגיט-האב כדי לגבות את כל הקוד שלנו. אנחנו מחלקים לקבצים ול Subsystems. אנחנו מארגנים את כל הקוד בתיקות מסודרות כמו תיקייה נפרדת לאוטונואומי Auto. בקרת PID:

קוד Pid נועד בשביל לגרום למערכות לעבוד ביותר דיוק ע"י שימוש בשלושה מארכים הפרופורציונלי-P, האינטגרלי-I והדיפרנציאלי-D.

- **פרופורציונלי-P:** הביטוי, מכפיל את ערך אות השגיאה הנוכחי בקבוע "KP" ומתייחס לערכה העדכני של השגיאה בלבד. זהו החלק הפשוט ביותר בבקר.
- **אינטגרלי-I:** החלק, מבצע אינטגרציה בזמן על אות השגיאה מתחילת המדידה ועד לרגע החישוב, ומכפיל את התוצאה בקבוע "KI" חלק זה מתייחס לערכי השגיאה מזמן תחילת המדידה עד לרגע החישוב, ובכך בעצם מושפע מערכי העבר של השגיאה.
- **דיפרנציאלי-D:** החלק, מבצע גזירה של אות השגיאה לפי הזמן, ומכפיל את התוצאה בקבוע "KD". חלק זה כביכול מתייחס להתנהגות העתידית של אות השגיאה על ידי בדיקה של קצב השינוי שלה, או במילים אחרות- באיזו מהירות היציאה מתקרבת או מתרחקת מאות הייחוס הרצוי.

תוכנה - טלאופ

- **הנעה** - אנחנו משתמשים בחישוב ווקטורים בשביל לחשב את המהירות של כל אחד מהמנועים בהנעה ואנחנו משתמשים בגיירו כדי לסובב את ווקטור התנועה סביב הגיירו ובכך להשיג נהיגת "מבט מגרש",
 - **מעלית** - במעלית אנחנו משתמשים בבקרת PID בשביל להעלות אותה בגובה הרצוי בדיוק רב ולהשאיר אותה במקום..
 - **זרוע** - בשביל לשלוט על הזרוע, השתמשנו בחיישני מגע בשביל להגיד לה מתי לעצור, מתי לעלות, ומתי לרדת.
 - **אינטייק** - אנחנו הכנו פונקציות כדי לעצור את האינטייק כשהוא מזהה חלק משחק. בשביל הזיהוי, השתמשנו בחיישן מרחק.
 - **צבת** - בנוגע לצבת, יצרנו פונקציה שיוודע אם הוא פתוח או סגור, ולפי זה, סגרנו ופתחנו אותו במצבים שהוא היה צריך.
- אנחנו תכנתנו את הטלאופ על ידי תנאים שהוא עדה מה לעשות בכל מצב. במקום שלכל מערכת יהיה כפתור משלה, השתמשנו במלא תנאים בצורה יעילה כדאי לצמצם את המספר כפתורים שאנו משתמשים בהם.

```
private void intakeAutomation() { 1 usage
```

```
if (robotState == "INTAKE" && !intake.isGamePiece()) {
    dashTele.addData( caption: "intakeAuto state", value: 1);
    intake.setPower(-0.5);

    if (!touchBottom.isPressed()) {
        arm.setPower(-0.3);
    } else {
        arm.setPower(0);
    }
}
```

```
if (robotState == "INTAKE" && intake.isGamePiece() && !touchTop.isPressed()) {
    intake.setPower(0);
    if (!touchTop.isPressed()) {
        arm.setPower(0.3);
    } else {
        arm.setPower(0);
    }
}
```

```
private void highScoringAutomation() { 1 usage
if (robotState == "HIGHSCORING" && elevatorPControl.atPoint() && !isGamePieceDoingScoring)
if (elevatorPControl.getSetPoint() == 0) {
    basket.setPosition(0.95);
    elevatorPControl.setSetPoint(105);

    dashTele.addData( caption: "basketStages", value: 1);

} else if (elevatorPControl.getSetPoint() == 105 && elevatorPControl.atPoint()) {
    basket.closeBasket();
    isGamePieceDoingScoring = true;

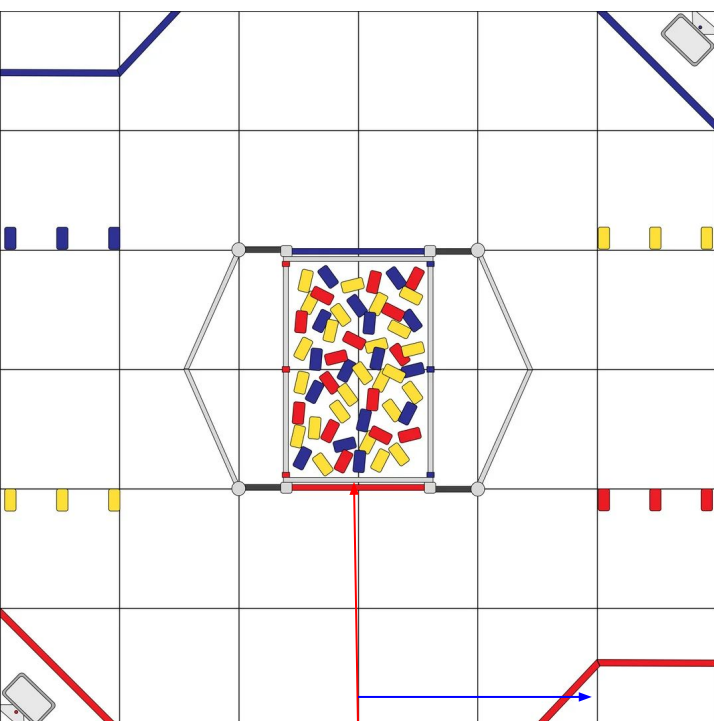
    dashTele.addData( caption: "basketStages", value: 2);

    robotState = "IDLE";
}
```

תוכנה - אוטונומי

בשלב האוטונומי אנחנו משתמשים בספריית Road Runner כדי לשלוט על הרובוט ולעקוב אחרי מסלולים מורכבים תוך כדי שאנחנו עושים פעולות אחרונות.

אחרי שערכנו את המבדקים של Road Runner. כתבנו את הקוד עם Trajectories שבהם הגדרנו את המיקום התחלה של הרובוט ואז הוא הלך לאיפה שאמרנו לו במפה כך שהוא תמיד יגיע לאותו מקום כל פעם. בשביל להפעיל את המערכות שלנו, השתמשנו בסוגים שונים של Markers.



שלב 1 - הרובוט נוסע ישר ותולע ספייסימן על הבאר הגבוה.

שלב 2 - הרובוט חוזר וחונה אובזרביישן זון.

התנדבויות

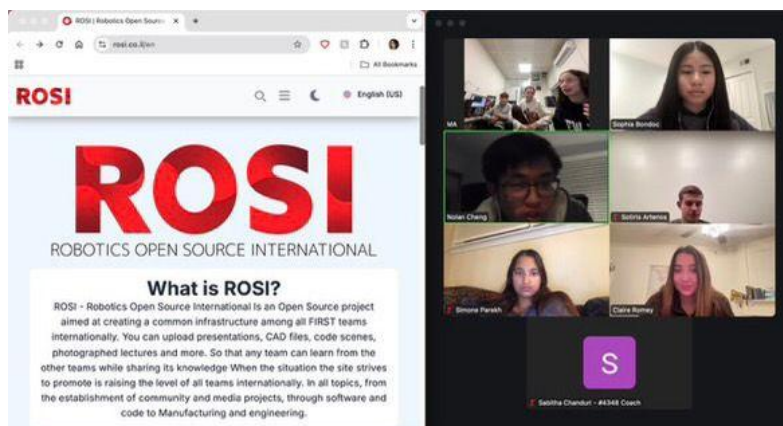
במהלך השבועות, כל חברי הקבוצה שלנו מתנדבים בבתי ספר שונים. אנחנו מתנדבים ב9 בתי ספר שונים, ו5 מתוכם הם בתי ספר לחינוך מיוחד. בהתנדבויות בונים רובוטים ומתכנתים אותם ביחד. FIRST, אנחנו מעבירים לילדים את ערכי מטרתה של ההתנדבות היא לחשוף ילדים מאוכלוסיות שונות ברחבי העיר למדע שמפגישה את הילדים עם עולמות, FIRST, ולטכנולוגיה, ובייחוד לתוכנית חדשניים ומאפשרת להם ללמוד בצורה חוייתית ומעשית.



ROSI

open source פרויקט ROSI - Robotics Open Source International שמטרתו יצירת תשתית משותפת בין כל קבוצות FIRST בארץ. לאתר שלנו אפשר להעלות מצגות, קבצי CAD, קטעי קוד, הרצאות מצולמות ועוד. כך שכל קבוצה, תוכל ללמוד מקבוצות האחרות תוך שיתוף הידע שלה כאשר המצב שהאתר שואף לקדם הוא העלאת הרמה של כל הקבוצות בארץ בכל הנושאים, מהקמת פרויקטים קהילתיים ומדיה, דרך תוכנה וקוד ועד להנדסת מכונות והנדסה. יש באתר יותר מ-170 מצגות מ-30 קבוצות שונות מ-15 מדינות שונות

אנחנו עזרנו ויצרנו קשר עם 100 קבוצות מסביב העולם. חלק העלו מצגות ואפילו נפגשנו בזום עם כמה קבוצות מארצות הברית!



לסיכום אנחנו קבוצה מליגת **FTC** המשתתפת בתחרות **FIRST** ועובדים יחד על פיתוח רובוט מתקדם. זו השנה השנייה שלנו בארגון, אחרי שהתחרינו ב-FLL, וכעת אנו מתמודדים עם אתגרים חדשים בתחומים כמו תכנות, מכניקה, נהיגה ואסטרטגיה.

הרובוט שבנינו משלב **מערכת הנעה מכאנום**, **מעלית רב-שלבית** לאיסוף והרמה, ותכנון מדויק של כל הרכיבים כדי להגיע לביצועים מיטביים. אנו מתכנתים ב-**Java** עם **PID** לשליטה מדויקת, ומשתמשים ב-**Road Runner** לתנועה אוטונומית חכמה.

מעבר לבנייה, אנחנו פעילים בקהילה, מתנדבים בעשרה בתי ספר ומפיצים את ערכי **FIRST** בקרב ילדים, כולל בחינוך מיוחד. בנוסף, יצרנו קשרים עם קבוצות מרחבי העולם במסגרת **פרייקט ROSI**, ושיתפנו ידע עם יותר ממאה קבוצות.

דרך העבודה למדנו איך לעבוד בצוות, לפתור בעיות טכניות ולשפר את יכולות התכנון, הייצור והחשיבה האסטרטגית שלנו. אנו ממשיכים לשאוף קדימה, לשפר את הרובוט ולהפיץ את האהבה לרובוטיקה!

עם למידת הערכים של **FIRST** והיכולות הטכניות שלנו, עד תרומה לקהילה – החזון שלנו מסתכם במשפט אחד – אנחנו לא נעצור לעולם ונגשים את החזון שלנו – **MAkers lead the way**