

מבוא לרובוטיקה – תרגיל 4**States**

- 1. Move
- 2. findBase
- 3. turn
- 4. block
- 5. addHomeBase
- 6. collision
- 7. passRobot

הסבר שעונים

על מנת שלא נבזבז מלא זמן בלחסום את בסיס האויב ונבצע גם עוד משימות, הגדרנו שני שעונים:

- 1. frequencyBlockTimer – כל כמה זמן נוכל לחסום את בסיס האויב
 - 2. blockTimer – כמה זמן הרובוט חוסם את בסיס האויב
- שעונים נוספים:
- 3. frequencyTurnTimer - כל כמה זמן נסתובב
 - 4. turnTimer – כמה זמן הרובוט מסתובב
 - 5. startCover – מאתחל את frequencyTurnTimer כל כמה זמן
 - 6. frequencyAddPosTimer - כל כמה זמן נוכל להוסיף מיקום של בסיס למיקומי הבסיס הקיימים

זכירת מיקום הקו

יצרנו שני ווקטורים:

- 1. homePos – שומר את מיקומי הבסיס שלנו המדויקים
- 2. homeArea – שומר אזורים סמוכים לבסיס שלנו

כאשר הרובוט מוריד אוכל בבסיס בפעם הראשונה, נשמור את מיקום הבסיס המדויק ב-homePos, אך אם הרובוט רואה בסביבתו את הבסיס (לאו דווקא בהורדת האוכל), נשמור את המיקום הזה כאזור סמוך לבסיס שלנו, כך בפעם הבאה שהרובוט יאסוף אוכל הוא יוכל לנסוע באופן ישיר לבסיס שלנו או לאזור סמוך.

ישנה עדיפות למיקום בסיס מדויק מאשר למיקום סמוך, ולכן כאשר הרובוט מוצא אוכל, אנו קודם נבדוק אם יש מיקום מדויק, ורק אם אין מיקום מדויק נחזיר מיקום סמוך. כמו כן, אם יש כמה מיקומים מדויקים, כלומר יש כמה בסיסים שזכרנו- נחזיר את הקרוב ביותר, ובאותו אופן נפעל גם לגבי האזורים הסמוכים.

פירוט על המצבים

move

1. אם לרובוט יש אוכל – עובר למצב findBase
2. אם הרובוט רואה מולו/מימינו/משמאלו את בסיס האויב וגם frequencyBlockTimer הסתיים – נוסע לכיוון הבסיס, מאתחל את blockTimer ועובר למצב block
3. אם הרובוט רואה מולו/מימינו/משמאלו את הבסיס שלנו וגם frequencyAddPosTimer הסתיים – נוסע לכיוון הבסיס ועובר למצב addHomeBase
4. אם הרובוט בהתנגשות חזיתית – מאתחל את turnTimer ועובר למצב collision
5. אם הרובוט רואה מולו רובוט אחר - מאתחל את turnTimer ועובר למצב passRobot
6. אם הרובוט רואה מולו מכשול – מאתחל את turnTimer ועובר למצב turn
7. אם frequencyTurnTimer הסתיים - מאתחל את turnTimer ועובר למצב turn
8. אחרת – תיסע ישר

findBase

1. אם לרובוט אין אוכל – עוצר, זוכר את מיקום הקן על ידי הוספת המיקום ל-homePos (אם לא קיים), מאתחל את frequencyAddPosTimer, עובר למצב Move, ומאתחל את startCover
2. אם הרובוט רואה את הבסיס שלנו מולו/מימינו/משמאלו - נוסע לכיוונו

3. אם הרובוט רואה מולו מכשול – נוסע לכיוון הנגדי
4. אם הרובוט רואה מולו רובוט אחר – ממשיך לנסוע ישר
5. אחרת –
- א. אם homePos ריק וגם homeArea ריק – נוסע ישר
- ב. אחרת – מאתחל את turnTimer ועובר למצב turn

turn

1. אם לרובוט יש אוכל וגם homePos או homeArea לא ריקים – מוצא את הבסיס הכי קרוב לפי זכירת הבסיס (כפי שהוסבר לעיל), מחשב את הזווית שצריך להסתובב אליה כדי לנסוע לכיוון הבסיס, מסתובב לזווית, כאשר הוא מגיע לזווית נוסע ישר ועובר למצב findBase
2. אם הרובוט רואה מולו מכשול – מסתובב ועובר למצב move
3. אם ה-turnTimer הסתיים – מאתחל את frequencyTurnTimer ועובר למצב move.
4. אחרת – מסתובב

Block

1. אם לרובוט יש אוכל – עובר למצב findBase
2. אם הרובוט נדחף בזמן החסימה – נוסע ישר, מאתחל את frequencyBlockTimer, ועובר למצב move
3. אם blockTimer הסתיים – נוסע ישר, מאתחל את frequencyBlockTimer, ועובר למצב move
4. אם הרובוט לא רואה מולו/מימינו/משמאלו את בסיס האויב – עוצר
5. אם הרובוט רואה מולו/מימינו/משמאלו את בסיס האויב – נוסע לכיוונו

addHomeBase

1. אם לרובוט יש אוכל – עובר למצב findBase

2. אם הרובוט לא רואה מולו/מימינו/משמאלו את בסיס האויב – עוצר, מוסיף את המיקום שלו ל-homeArea, מאתחל את frequencyAddPosTimer, ועובר למצב move
3. אם הרובוט רואה מולו את הבסיס שלנו – נוסע לכיוונו ועובר למצב move

Collision

1. אם לרובוט יש אוכל – עובר למצב findBase
2. אם turnTimer הסתיים – עובר למצב move
3. אחרת – מסתובב

passRobot

1. אם לרובוט יש אוכל – עובר למצב findBase
2. אם turnTimer הסתיים – עובר למצב move
3. אחרת – הרובוט נוסע בכיוון ההפוך מאיפה שראה רובוט אחר

תגובת הרובוט לרובוטים אחרים

הרובוט מגיב בצורה זהה לכל רובוט שהוא רואה מולו, בין אם זה רובוט מהקבוצה שלו ובין אם זה רובוט מהקבוצה הנגדית. הסיבה שאין הפרדה בין התגובות היא שאין לנו יכולת לדעת כיצד רובוטים מקבוצות אחרות יגיבו, ואנו נעדיף להימנע ממצבים שיכולים לגרום לבזבוז זמן ולפגוע ביעילות חיפוש המזון.

בחינת הקבוצה שלנו נגד קבוצות אחרות

את הקבוצה שלנו הרצנו נגד 3 קבוצות שונות, להלן צילום מסך של התוצאות:

arena	seed	time	first-name	first-score	second-name	second-score	winner
easy	12	70000	foraging_8_controller	27	opponent_1	10	foraging_8_controller (first team)
easy	12	70000	foraging_8_controller	144	opponent_2	52	foraging_8_controller (first team)
easy	12	70000	foraging_8_controller	124	opponent_3	126	opponent_3 (second team)

ניתן לראות שלאחר ניסוי דיי ארוך, הקבוצה שלנו סיימה פעמיים עם ניקוד גבוה יותר מהקבוצה המתחרה, ופעם אחת בניקוד צמוד מאד בהפרש של 2 נקודות, ובהתאם לכך אנו מרוצים מהתוצאות.