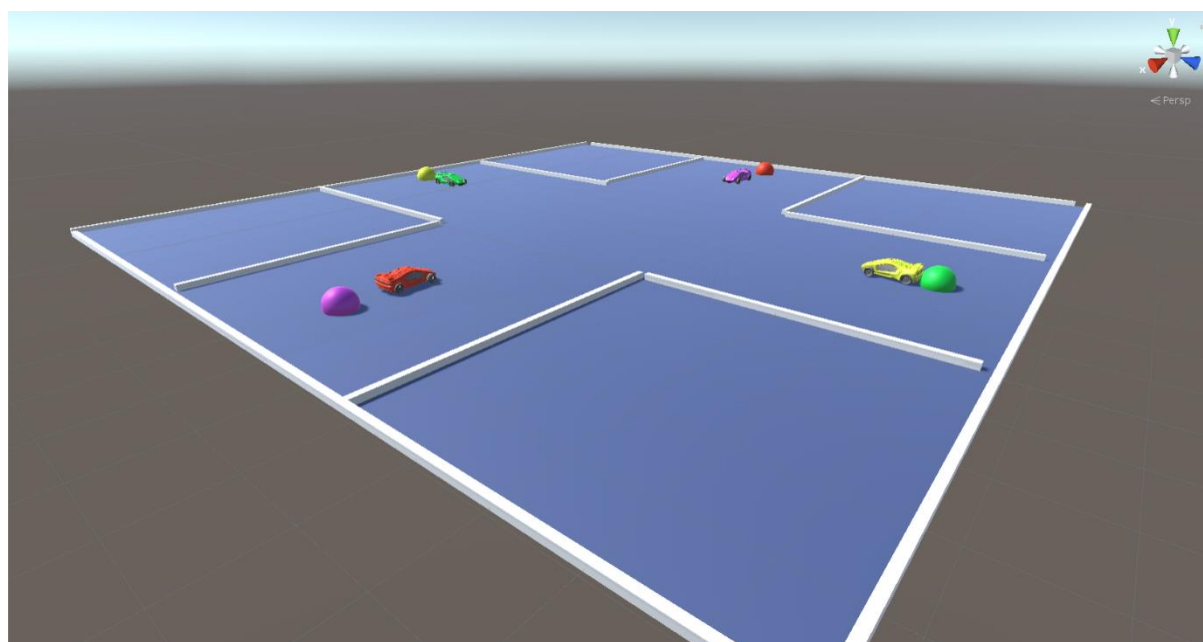


למידת חוקים חברתיים לביצוע תוכניות בסביבות מרובות סוכנים



רועי גנץ | יונתן קסלברנר

מנחה: אייל טייטלר

CRML lab, Technion

תוכן עניינים

2	מבוא:
3	תיאור הבעיה:
3	בניית המודל
3	הפתרון האלגוריתמי
4	רקע מדעי
4	רשת נוירונים
5	למידה מחזזקים Reinforcement learning
5	אלגוריתם Proximal Policy Optimization (PPO)
5	סוגיית bias-variance
6	סביבת העבודה
8	קשיים בפתרון הבעיה:
8	מידול הסביבה
8	ייצוג מרחב המצב
8	ייצוג מרחב הפעולה
8	הגדרת המודל הדינאמי של הבעיה ואופי השליטה ברכבים
9	פונקציית התגמול
10	קביעת קונפיגורציית הרשת וקביעת ההיפר-פרמטרים
11	פתרון הבעיה
11	מידול הסביבה
14	מימוש ואימון האלגוריתם הלומד
17	תוצאות
18	גישות אחרות לפתרון הבעיה
18	אלגוריתם מחקה
18	פתרון מבוסס חיפוש יוריסטי
19	רעיונות להמשך
19	תרחישים תחבורתיים מורכבים יותר
19	חוקים חברתיים עבור עבודת צוות של סוכנים
19	סיכום ומסקנות
20	ביבליוגרפיה ומשאבים

מבוא:

בחברה אנושית נהוגים סטים ברורים של כללים חברתיים ונורמות התנהגות אשר נקבעו, עוצבו ונלמדו במרוצת השנים ומטרתם לאפשר את המשך הקיום החברה. הכללים הללו רבים וחולשים על כל תחומי החיים, למשל, נורמות התנהגות בחברה. אנו, בני האדם, פועלים עפ"י החוקים והנורמות הללו מתוך הבנה שיש צורך בכללי יסוד לצורך חיים בחברה אנושית.

חלק מהכללים הללו הינם כללים א-פורמליים (למשל, "מפני שיבה תקום והזרת פני זקן") וחלק מהם פורמליים ומעוגנים בחוקי המדינות. בפרויקט זה, אנו מתעניינים בחוקים חברתיים ונורמות התנהגות בתחבורה. החוקים הללו נועדו להבטיח בראש ובראשונה את ביטחון הנוסעים, אך גם את יעילות מערכת התחבורה.

בעתיד הנראה לעין, ככל הנראה ייסעו בכבישים רכבים אוטונומיים רבים. אחת הבעיות הכרוכות בכך הינה תזמון תנועת כלי הרכב ללא התנגשויות וללא סיכון חיי אדם. בעיה זו באה לידי ביטוי בין היתר בעת חציית צמתים. מכיוון שהרכבים הללו יישלטו ע"י תוכנה חכמה, השאיפה תהא שהצמתים לא יהיו מרומזרים כלל, ושהרכבים יחלפו בצומת, מבלי לעצור ומבלי לסכן חיי אדם.



תמונה להמחשת צומת לא מרומזר בעולם האמיתי

לצורך משימה זו, על הרכבים ללמוד חוקים חברתיים, דהיינו, קודי התנהגות, אשר יש לנהוג לפיהם על מנת ליצור סדר ולהימנע מתאונות. מטרת פרויקט זה הינה לבצע למידה של חוקים אלו אשר יאפשרו לרכבים אוטונומיים לפעול על פי המדיניות המיטבית – כזו אשר תמזער את הסיכון ותמקסם את היעילות.

כדי לפתור בעיה זו, בחרנו להשתמש ב"למידה מחיזוקים" (Reinforcement Learning) באמצעות אלגוריתם PPO. כמו כן, נעזרנו במנוע הגרפי Unity ובעזר ml-agents. שילוב שני הכלים הללו מאפשרים למדל את התרחישים המבוקשים ולאמן את הרכבים האוטונומיים על מנת שאלו ילמדו את החוקים החברתיים ואת נורמות ההתנהגות הדרושות לצורך צליחת המשימה.

תיאור הבעיה:

הבעיה בה אנו עוסקים היא יצירת סביבה המכילה מספר סוכנים (כלי רכב אוטונומיים) אשר ילמדו כללי התנהגות שונים שיאפשרו להם לנווט בצמתים בצורה יעילה ככל הניתן, מבלי לסכן חיי אדם. בעוד שכבישי העולם נשלטים ע"י סטים של חוקים ידועים וברורים, בעתיד, כאשר רכבים אוטונומיים הנשלטים ע"י סוכנים נבונים יחליפו את הנהגים האנושיים, ייתכן ויהיו חוקים אחרים המאפשרים תחבורה יעילה יותר ומסוכנת פחות. בפרויקט זה, ננסה למדל עולם שכזה ונאפשר לסוכנים הללו ללמוד חוקים חברתיים ונורמות התנהגות בכבישים.

במסגרת הפרויקט התמודדנו עם שתי בעיות מרכזיות:

- בניית מודל המדמה את הבעיה האמיתית אותה נרצה לפתור.
- קושי אלגוריתמי במציאת פתרון באמצעות למידה מחיזוקים.

בניית המודל:

בניית מודל המדמה מערכת מהעולם האמיתי היא בעיה מורכבת הטומנת בחובה קשיים רבים. הסביבה, המכילה סוכנים ואובייקטים שונים, נדרשת לפעול על פי חוקים פיזיקליים כמו תאוצה, מהירות וחיוך. כמו כן, על סביבת הסימולציה להיות מקושרת לאלגוריתם הלומד אשר לפיו יפעלו הסוכנים.

לצורך מידול הבעיה, נעזרנו במנוע הגרפי Unity ובנינו סביבה המדמה צומת לא מרומזר ובו רכבים אוטונומיים. מימשנו סביבה ובה הסוכנים המפעילים את הרכבים האוטונומיים יכולים לשלוט בכיוון ובמהירות הנסיעה. כמו כן, סביבה זו מאפשרת להגדיר יעד לכל רכב ומאפשרת לרכבים לדגום את סביבתם באמצעות חיישנים שונים.

הפתרון האלגוריתמי:

הקושי האלגוריתמי הינו הגדרת בעיית למידה מורכבת, מבוססת למידה מחיזוקים, המייצגת נכונה את הבעיה אותה אנו מנסים לפתור. האלגוריתם המורכב מרשת נוירונים מקבל כקלט נתונים רבים אודות מצב הסוכן והסביבה ומספקת כפלט את הפעולה שעל הסוכן לבצע. יתר על כן, עלינו להגדיר את הקונפיגורציה המתאימה לפתרון הבעיה.

רקע מדעי

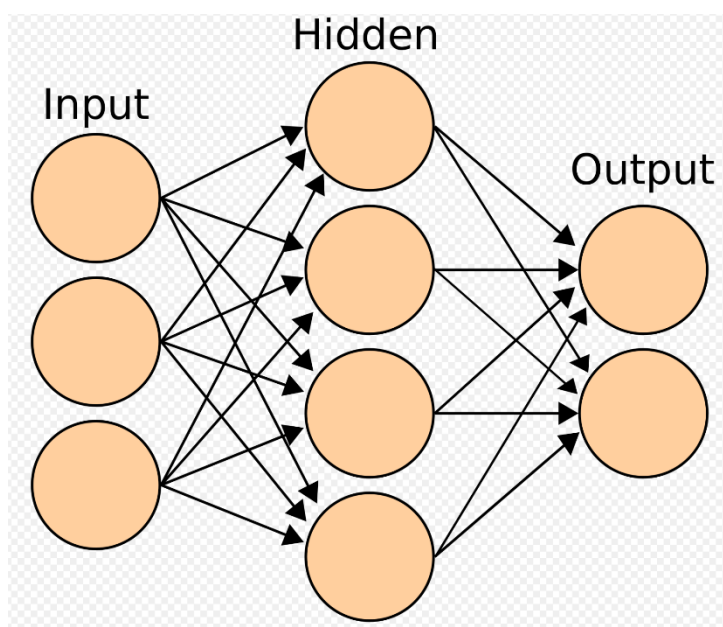
רשת נוירונים

רשת נוירונים הינה מודל מתמטי אשר פותח בהשראת המוח האנושי. רשת זו מורכבת ממספר רב של יחידות חישוב המכונות נוירונים ומקשתות המחברות אותם זה לזה. נוירון הינה יחידת מידע המפיקה מידע כתלות הקלט אותו היא מקבלת. בעוד שכל נוירון יכול ללמוד פונקציות בסיסיות בלבד, שילוב של נוירונים רבים עם פונקציית אקטיבציה מאפשר למערכת ללמוד פונקציות מורכבות מאוד.

באופן כללי, רשת הנוירונים מורכבת משכבת כניסה (הקלט), משכבת מוצא (הפלט) ולרוב תכיל גם שכבות פנימיות (שכבות חבויות). כל שכבה כזו מורכבת מנוירונים אשר מחוברים באמצעות קשתות ממושקלות לנוירונים בשכבה הבאה. למשל, בבעיה שלנו קלט רשת הנוירונים הינו המצב הנכחי של הרכב (מיקום, כיוון למטרה וכו') ופלט רשת הנוירונים הינו הפעולה שעל הרכב לבצע (היגוי מתאים).

בשלב הלמידה, הרשת מקבלת דוגמאות רבות ומכוונת את משקולות הקשתות שברשת, בהתאם לדוגמאות. התהליך שבאמצעותו מתבצע כיוון המשקלים מכונה "פעפוע לאחור" (back propagation). בתהליך זה, מספקים לרשת קלט, מחשבים את הפלט שהרשת מפיקה ומחשבים את פונקציית השגיאה המתקבלת. לאחר מכן, מבצעים גזירה של השגיאה כתלות במשקולות השונים, ומעדכנים את המשקולות כדי למזער את השגיאה העתידית.

לאחר שלב הלמידה, משקולות הרשת מכוונים וניתן להשתמש בה.

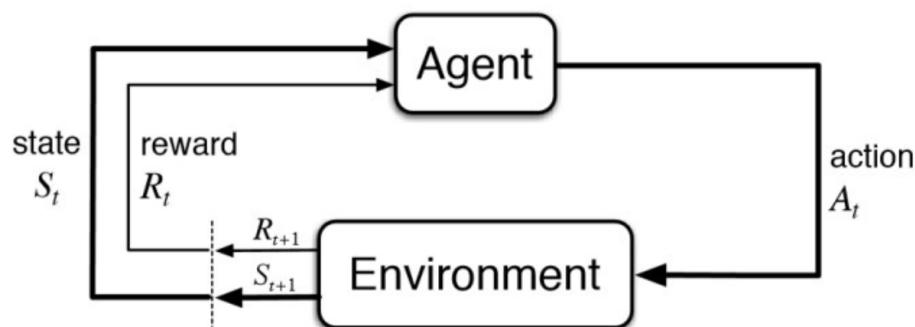


דוגמה לארכיטקטורה של רשת נוירונים

למידה מחיזוקים Reinforcement learning

למידה מחיזוקים הינה תת-תחום של למידת מכונה. בתחום זה, סוכנים שונים נוקטים פעולות בסביבה כלשהי בכדי למקסם רווח מצטבר. המטרה הינה שהסוכנים ילמדו מדיניות פעולה אשר תביא למיקסום של הגמול אותו הם יקבלו.

מודל הלמידה מחיזוקים מבוסס על ניסוי וטעיה שמתבצע בסדרת אינטראקציות בין סוכן לסביבתו ולסוכנים אחרים. בכל אינטראקציה הסוכן מקבל מידע אודות מצבו ובוחר לבצע פעולה מתוך מגוון הפעולות הנתונות בידיו, עפ"י מדיניותו. עבור כל פעולה, מקבל הסוכן תגמול (ערך מספרי) המבטא את טיב הפעולה. מטרת הסוכן הינה לגבש מדיניות פעולה אשר מתאימה בין מרחב המצבים האפשריים לבין מרחב הפעולות, כך שבכל מצב בו יימצא, הסוכן יבצע פעולה אשר תביא לגמול המרבי אותו ניתן להשיג.



המחשת עקרון הפעולה של למידה מחיזוקים

אלגוריתם Proximal Policy Optimization (PPO)

אלגוריתם PPO הינו אלגוריתם של למידה מחיזוקים אשר נעזר ברשת נוירונים. האלגוריתם משנה את משקלי רשת הנוירונים בהתאם למדיניות אותה אנו מכתיבים לו בעזרת פונקציית התגמול ובכך מקרב את הפונקציה האידיאלית אשר ממפה ממרחב המצב למרחב הפעולה (כלומר, הפונקציה שתמקסם את התגמול). אלגוריתם זה הינו SOTA שכן הוא מספק תוצאות מעולות תוך סיבוכיות חישובית מועטה יחסית לאלגוריתמים מתחרים, קלות כיוון פרמטרים וצורך מועט יחסית בדגימות. בשל יתרונותיו הרבים, בחרנו להשתמש בו.

סוגיית bias-variance

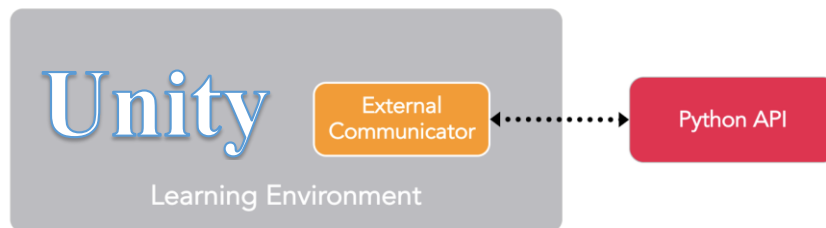
bias-variance אלו שני מאפיינים של מודלים בלמידת מכונה, אשר לרוב בעלי קורלציה הפוכה – מודל עם bias קטן, לרוב יהיה עם variance גדול, ולהיפך. כעיקרון, bias מבטא את הקירבה בין הפונקציה הטובה ביותר השייכת למרחב ההיפותזות של המודל לבין פונקציית המטרה. לעומת זאת, ה-variance מבטא את טיב הפונקציה שנלמדה בפועל ע"י האלגוריתם ביחס לפונקציה הטובה ביותר אשר שייכת למרחב ההיפותזות של המודל.

השגיאה בין הפונקציה הנלמדת לבין פונקציית המטרה מורכבת מגורם התלוי ב-bias ומגורם התלוי ב-variance. לכן, היינו רוצים להקטין את שניהם, אך בשל אופיים זה לא אפשרי: tradeoff.

סביבת העבודה

בפרויקט שלנו נעזרנו במנוע הגרפי unity ובעזר ml-agents. Unity הינו מנוע גרפי המשמש, בין היתר, לפיתוח של משחקים וסימולציות תחת החוקים הפיזיקליים הקיימים בעולם האמיתי (כבידה, חיכוך וכו'). ml-agents הינו עזר קוד פתוח אשר מהווה ממשיך בין המנוע הגרפי Unity לבין אלגוריתם הלמידה ובכך מאפשר לאמן סוכנים נבונים בסימולציות ומשחקים.

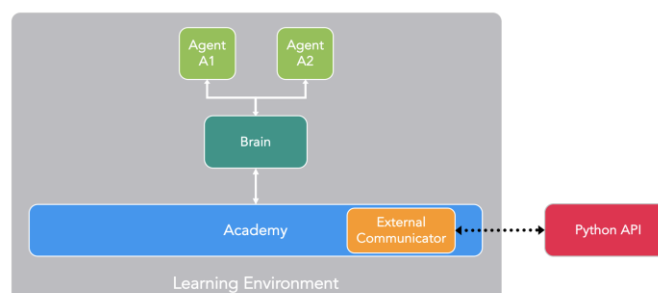
הסוכנים יכולים להתאמן וללמוד מדיניות פעולה באמצעות שיטות שונות של למידת מכונה ובהן למידה מחיזוקים, בה אנו משתמשים בפרויקט. עזר זה מקשר בין הסימולציה במנוע הגרפי Unity לאלגוריתם הלמידה הממומש בPython : ml-agents מממשק בין הסימולציה לבין האלגוריתם הלומד ופועל כדלקמן - הסוכנים אשר פועלים בסביבת הסימולציה מספקים לאלגוריתם הלמידה את וקטור המצב שלהם (אשר מהווה קלט לרשת הניורונים), בעוד שאלגוריתם הלמידה מספר לסוכנים את הפעולה הבאה שעליהם לעשות (הפלט של רשת הניורונים). כמו כן, הסוכנים מעדכנים את אות התגמול, אשר המדיניות הנלמדת תנסה למקסם.



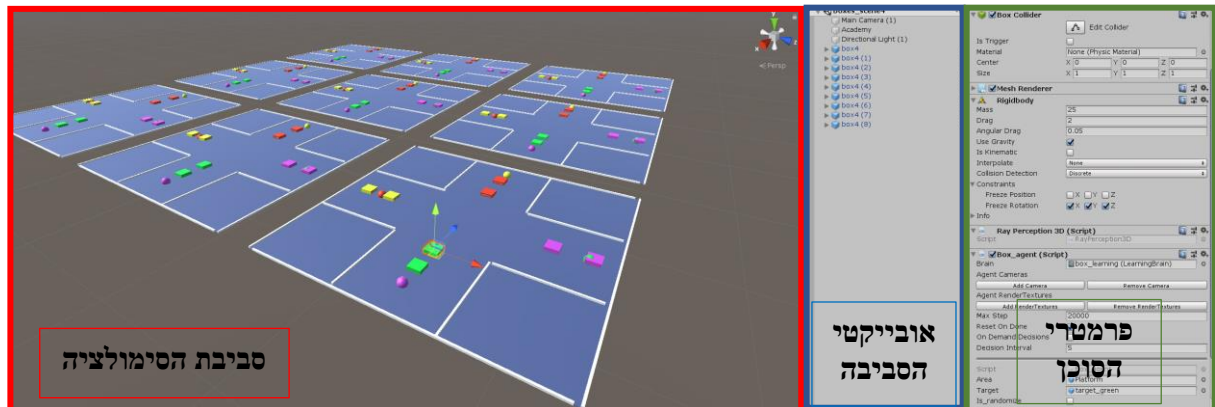
ml-agents מכיל מספר מודולים אשר יש לממשם:

- **אקדמיה** – אובייקט המאגד תחתיו את הסביבה והאובייקטים בסצנת Unity. מודול זה אמון על אתחול הסביבה, איפוסה והפעלת הסימולציה בכל צעד, תוך ניהול הקשר מול האלגוריתם הלומד.
- **סוכן** – שחקן אשר מסוגל להתבונן בסביבה שלו ולקבל החלטה כיצד עליו לפעול, בהתאם לתצפיותיו. בעת שימוש בלמידה מחיזוקים, תפקידיו החשובים ביותר עליהם אמון הסוכן הינם הגדרת וקטור המצב (התצפיות שלו) ואות התגמול המבוסס על המצב הנוכחי. הסוכן מוסר אותם למדיניות (לאלגוריתם הלומד) אשר מחזירה לו איזו פעולה עליו לבצע. בכל צעד בסימולציה, הסוכן מדווח אודות מצבו הנוכחי למדיניות ומקבל בחזרה הוראה כיצד עליו לפעול.
- **מוח** – אובייקט אשר מגדיר את מבנה הקלט והפלט של רשת הניורונים – מגדיר את גודל הקלט ואת גודל הפלט וסוגו: בדיד או רציף. לכל סוכן ממופה מוח אשר מממש את המדיניות של הסוכן. כמו כן, מוח אחד יכול להתמפות למספר סוכנים. משמעות הדבר היא שכל הסוכנים הללו יפעלו עפ"י אותה המדיניות.

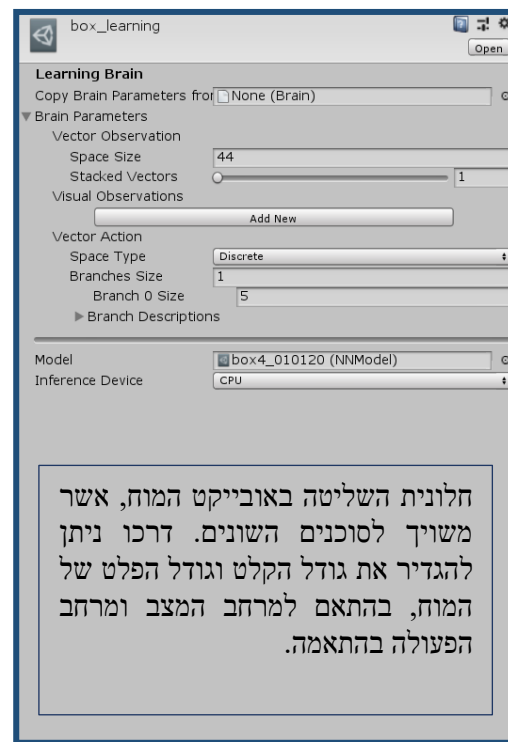
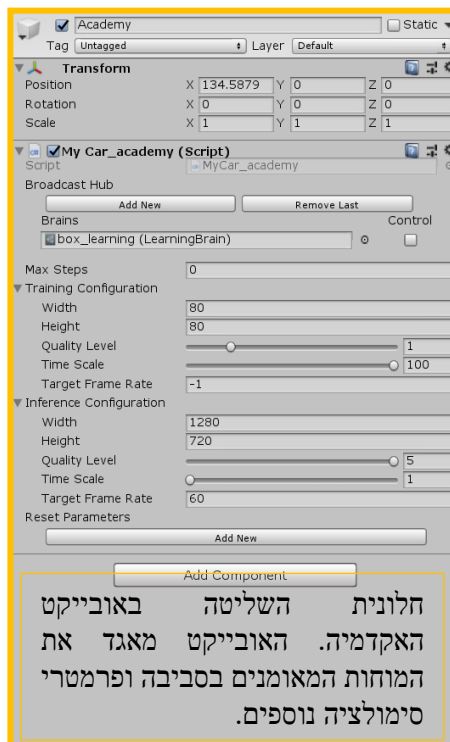
מצורפת דיאגרמה הממחישה את מבנה הסביבה ואת הממשק שבין המודולים השונים בה:



להלן תמונות הממחישות את נראות ממשק סביבת הunity וה-ml-agents על רכיביהם השונים:
חלון הסביבה הראשי כפי שמופיע ב-Unity:



חלוניות חשובות נוספות:



קשיים בפתרון הבעיה:

לצורך פתרון בעיית הניווט, היה עלינו להתמודד עם מספר קשיים הן טכניים והן אלגוריתמיים. בין הקשיים הללו נמנים מידול הסביבה, בחירת ייצוג מרחב המצב ומרחב הפעולה, בחירת פונקציית התגמול, בחירת ארכיטקטורה לרשת הניורונים והיפר-פרמטרים למודל. נציין כי הסוגיות הללו משפיעות זו על זו, דבר אשר מקשה על פתרון – לא ניתן לשפר כ"א בנפרד בשל ההשפעה ההדדית. למשל, בחירת מרחב מצב ומרחב פעולה עשירים מאוד, יצריכו קונפיגורציית רשת שונה מכאלו מצומצמים. להלן הרחבה אודות הקשיים השונים עמם התמודדנו בפרויקט:

מידול הסביבה – היה עלינו ליצור סביבה באמצעות מנוע גרפי אשר תתאר נכונה את הבעיה, כפי שהיא בעולם האמיתי. לשם כך, בחרנו להשתמש במנוע Unity – דבר אשר הצריך מאיתנו לרכוש ידע ומיומנויות רבות אשר לא היו ברשותנו בתחילת הפרויקט. נדרשנו להבין וללמוד איך להשתמש במנוע זה – כיצד הוא בנוי, ממשק קוד-סימולציה בשפת C# (אותה לא הכרנו), היכרות עם אובייקטים פיזיקליים מורכבים וכיצד להשתמש בהם. כמו כן, למדנו על האופן בו ספריית העזר ml-agents בנויה, על כל מרכיביה הרבים, וכיצד לקשר בין האלגוריתם הלומד לבין סביבת הסימולציה.

ייצוג מרחב המצב – בחירה חכמה של מרחב המצב הכרחית לפתרון הבעיה. כמו בכל בעיית למידה, המידע אותו יקבל המודל הלומד קריטי להצלחת הלמידה (במקרה שלנו, מרחב המצב הינו הקלט לאלגוריתם הלומד). במידה והמידע לא תמציתי דיו ו/או אינו מכיל את כל המידע הדרוש, הלמידה תיכשל. כלומר, מדובר בtradeoff – יש להימנע מלהכיל מידע לא רלוונטי, דבר אשר עשוי לפגום בלמידה ובמהירות ההתכנסות, אך גם יש להימנע מלהחסיר מידע חיוני. לכן, לצורך בחירת מרחב המצב היה עלינו לפתור בעיית אופטימיזציה – מציאת ייצוג תמציתי דיו, אך כזה שיכיל את המידע הקריטי לצורך התכנסות האלגוריתם ורכישת מדיניות טובה.

ייצוג מרחב הפעולה – מרחב הפעולה (רציף או בדיד) מכיל את הפעולות אותן הסוכן יכול לבצע ומהווה את הפלט של אלגוריתם הלמידה – בהינתן מצב כלשהו, על האלגוריתם להורות על הפעולה אותה יש לבצע. קביעת מרחב המצב מהווה גם כן tradeoff – מחד גיסא, על מרחב המצב לאפשר לסוכן לבצע את המשימה המוטלת עליו בהצלחה, אך מאידך גיסא, עליו להיות תמציתי דיו כדי לאפשר הגעה להתכנסות הלמידה. יתר על כן, פרט לפעולות אותן נרצה לאפשר לסוכן לבצע, יש להגדיר את אופיין: האם הערכים שיכולים להתקבל במרחב הפעולה הם רציפים או בדידים. בחירה במרחב פעולה רציף מאפשר לסוכן מגוון רחב יותר של פעולות אשר יאפשר לסוכן לממש מגוון רחב יותר של התנהגויות, אולם הדבר יכביד מאוד על תהליך הלמידה – הפונקציה אותה יש ללמוד תהא מורכבת משמעותית. בשל הגורמים הללו, היה עלינו לבחור במרחב פעולה אשר יאפשר פתרון של הבעיה ורכישת מדיניות טובה, אך כזה שיאפשר התכנסות של הליך הלמידה.

הגדרת המודל הדינאמי של הבעיה ואופי השליטה ברכבים – סוגיה זו מתקשרת בצורה ישירה להגדרת מרחב הפעולה ועיקרה טמון ב"סדר הבעיה" – האם הסוכנים יישלטו ע"י פעולות שישפיעו על מיקום כלי הרכב השפעה מסדר שני ע"י שליטה בתאוצה או מסדר ראשון ע"י שליטה במהירות.

פונקציית התגמול – בעת שלב הלמידה, אלגוריתם ה-PPO מנסה לגבש מדיניות אשר תמקסם את פונקציית התגמול. פונקציית זו אמורה להכווין את אלגוריתם הלמידה ברכישת המדיניות – לתגמל את הסוכן עבור התנהגות טובה, ולקנוס אותו על התנהגות רעה. ישנן 2 גישות מרכזיות בקביעת פונקציית תגמול:

- גישת Sparse Rewarding – גישה זו גורסת כי יש לתגמל את הסוכן רק עבור הגעה לתוצאה הרצויה (במקרה שלנו, בעת הגעה למטרה מבלי לבצע תאונה). החיסרון המרכזי של גישה זו הינו שעבור בעיות קשות, סביר להניח מבחינה הסתברותית כי הסוכן לא יצליח להגיע כלל לתוצאה הרצויה ללא הכוונה. הדבר נובע מכך שכדי להגיע למטרה עליו לבצע רצף פעולות ספציפי וארוך אשר ההסתברות לכך שיוגל בדיוק בסדר הנחוץ – אפסית. בגישת Sparse Rewarding, הגעה לרצף הפעולות הדרוש לצורך הגעה לתוצאה הרצויה שקול בקירוב להגרלה אקראית. הסיבה לכך טמונה בעיקרון הפעולה של למידה מחיזוקים – ניסוי וטעייה המבוסס על מיקסום פונקציית התגמול. בגישה זו, הסוכן לא יגיע כלל לתוצאה הרצויה ולכן לא ימצא את המדיניות אשר תמקסם אותה.

- גישת Reward Shaping – גישה זו מנסה להתגבר על מגבלות Sparse Rewarding בכך שהיא מתגמלת גם עבור התנהגות חיובית ולא רק עבור הגעה לתוצאה הרצויה. העקרון הטמון בבסיס גישה זו הוא חיפוש יוריסטי אחר המדיניות האופטימלית, במקום חיפוש עיוור. כעת, רצף הפעולות לא יוגרל באקראי אלא יוכוון ע"י ההתנהגות אותה נתגמל. במידה וההתנהגות אותה אנו מתגמלים והאופן בו אנו מתגמלים אותה, עולה בקנה אחד עם המדיניות האופטימלית, גישה זו תפעל טוב יותר מגישת ה-Sparse Rewarding. במידה ולא, אלגוריתם הלמידה עלול למצוא מדיניות שאינה אופטימלית כלל אשר ממקסמת את פונקציית התגמול אך מבלי להגיע לתוצאה הרצויה. לדוגמא, במידה ונקנס את הסוכן על הזמן שחולף (כדי להאיץ את הפתרון) יותר מדי, הוא יעדיף להגיע לסיום הריצה במינימום זמן, בלי קשר לתוצאה הסופית (למשל, ע"י התנגשות אשר תסיים את הריצה).

קביעת קונפיגורציית הרשת וקביעת ההיפר-פרמטרים – בעת קביעת מרחב המצב והפעולה, אנו למעשה מגדירים את הקלט והפלט של הרשת. אולם, עלינו לקנפג את כמות "השכבות החבויות" ואת מספר הנוירונים בכל שכבה. בחירה ברשת מורכבת תקטין את bias על חשבון variance: תאפשר ללמוד סט רחב יותר של מדיוניות פעולה אך חיפוש בסט רחב כזה יקשה על מציאת מדיניות טובה. לעומת זאת, בחירה ברשת קטנה מדי תקטין את variance על חשבון bias: אמנם סט המדיניות האפשרי קטן יחסית ויאפשר חיפוש יעיל, אולם הסיכויים שסט כזה יכיל את מדיניות המטרה, נמוכים. עלינו לבחור פרמטרי רשת כאלו אשר יהוו את עמק השווה ב-tradeoff זה. מעבר לקונפיגורציית הרשת, היה עלינו לקבוע מספר היפר-פרמטרים אשר משפיעים על הצלחת הליך הלמידה:

- גודל ה-batch – מגדיר את מספר הניסיונות אשר עליהם יש לבסס את עדכון משקולות רשת הנוירונים בעת ביצוע ה-gradient descent: גודל גדול יהיה יציב יותר ויכול להביא להתכנסות מהירה יותר, אולם עשוי להיתקע בנק' קיצון מקומית (בעוד שגודל קטן להיפך).
- גמול אינטרינזי ואקסטרינזי – הגמול אותו אנו נותנים לסוכן מכונה גמול אינטרינזי. בנוסף, קיימת אפשרות להגדיר גמול חיצוני – אקסטרינזי. גמול זה נועד לעידוד סקרנות של הסוכן במהלך הלמידה בכך שמתגמל הגעה למצבים שהסוכן טרם ביקר בהם. במידה ומשתמשים בשני הגמולים הללו, יש צורך למשקל את היחס ביניהם. עידוד הסקרנות יכול לגרום לסוכן לראות יותר מצבים ואף להביא להיחלצות ממינימום מקומי בעת חיפוש אחר מדיניות המטרה.

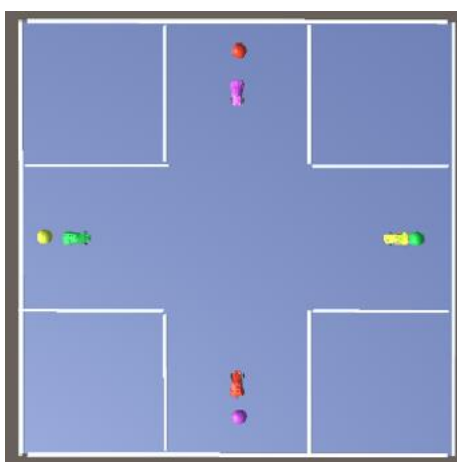
פתרון הבעיה

לצורך פתרון הבעיה, פעלנו בשני המישורים הבאים:

- מידול הסביבה באמצעות unity
- מימוש ואימון האלגוריתם הלומד

מידול הסביבה

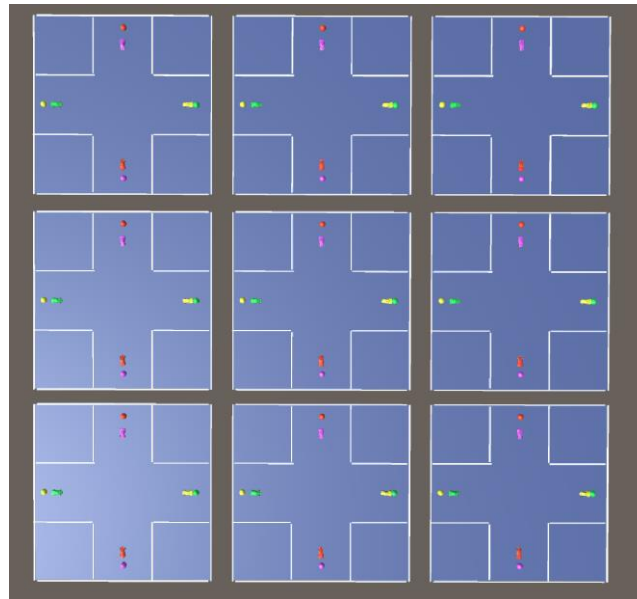
מימשנו את הסביבה כצומת cross לא מרומזר אשר מכיל שוליים בנויים. הצומת מכיל מספר סוכנים אשר לכל אחד מהם יעד אחר. מסלולי הסוכנים השונים חוצים בחלקם זה את זה, דבר אשר יאלץ את האלגוריתם ללמוד חוקים חברתיים שימנעו התנגשויות.



כדי לאפשר אימון עתידי של הסוכנים בסביבת הסימולציה, היה עלינו להגדירה על פי הכללים שml-agents מכתיב: הגדרת אקדמיה, מוחות וסוכנים. הגדרנו כי כל הסוכנים (agents) בסביבה שלנו משויכים לאותו המוח (brain), דהיינו, נוקטים באותה מדיניות ופועלים עפ"י אותם חוקים חברתיים. כמו כן, הגדרנו ומימשנו אקדמיה (academy) אשר אמונה על ריכוז ושליטה על תהליך האימון ותקשורת בין הסביבה לבין האלגוריתם הלומד.

בנוסף, מימשנו את הscript'ים והמתודות הדרושים עבור הסוכנים לצורך מימוש סביבה תקינה המסוגלת להתממשק ולתקשר עם האלגוריתם הלומד.

יתר על כן, על מנת להאיץ את הליך האימון, בחרנו לממש מספר זירות ולאמן אותן במקביל. הדבר מייצר יותר דוגמאות לאלגוריתם הלומד באותו פרק הזמן ומזרז את משך האימון.



סביבה מרובת זירות לאימון מואץ

פרט למצוין לעיל, היה עלינו לבחור את המודל הדינאמי של השליטה בסוכנים. עמדו לרשותנו שתי אפשרויות עיקריות:

- **מודל דינאמי מסדר שני** – שליטה במיקום הסוכנים באמצעות שינויי תאוצה. במודל זה, נממש כל סוכן כרכב (אובייקט ספריה ב-unity) אשר מהירותו נשלטת ע"י לחיצה על דוושת הגז ושינוי כיוון התנועה ע"י מערכת היגוי. כאמור, $\alpha(t) = x''(t)$ ולכן מדובר במודל סדר שני. כמו כן, שינוי רגעי בהיגוי הרכב אינו משפיע ישירות על הרכב, אלא מדובר בהשפעה מצטברת. כלומר, שינוי רגעי של ההיגוי באופן מסוים לא מביא לשינוי זהה בכיוון תנועת הרכב בו ברגע – מדובר בשינוי מתמשך.



אובייקט מכונית ב-unity

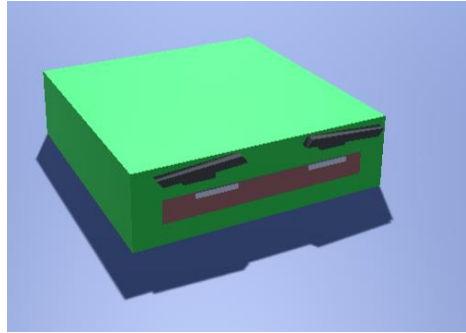
יתרונות הגישה –

- מודל שליטה מציאותי יותר
- ייצוג מדויק יותר של הבעיה בעולם האמיתי

חסרונות הגישה-

- מודל שליטה מורכב יותר אשר יקשה על האלגוריתם הלומד

- **מודל דינאמי מסדר ראשון** – שליטה במיקום הסוכנים באמצעות שינוי מהירות, ובכיוון התנועה ע"י סיבוב הסוכן על צירו. בגישה זו, נממש את המודל הפיזיקלי של הסוכן בעצמנו – נגדיר מנגנון בקרה אשר קובע את מהירותו ואת כיוון תנועתו בכל רגע נתון. היות והשליטה כאן היא על המהירות ולא על התאוצה, מדובר במודל דינאמי מסדר ראשון $x'(t) = v(t)$. כמו כן, אנו נשלט ישירות בכיוון תנועתו של הסוכן באמצעות סיבוב על צירו ולא דרך הגה כמו במודל הדינאמי מסדר שני.



אובייקט בעל מודל דינאמי מסדר ראשון אשר מימשנו

יתרונות הגישה-

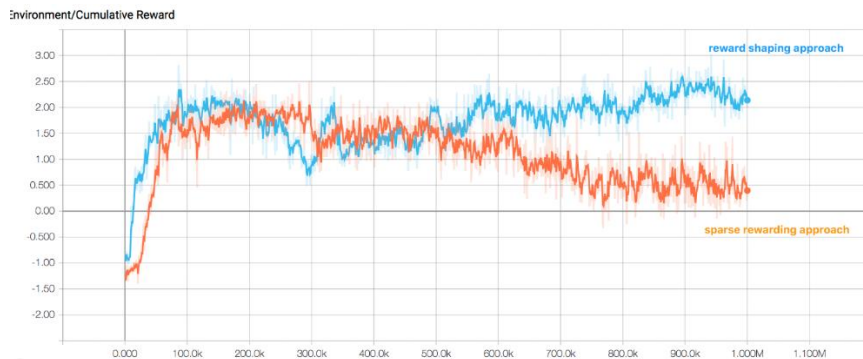
○ מודל שליטה פשוט יותר המקל על למידת האלגוריתם

חסרונות הגישה-

○ צורך במימוש מאפס של אובייקטי הסוכנים

- **בחירת המודל הדינאמי:** היה עלינו לבחור בין מודל שליטה מסדר שני (תאוצה) למודל שליטה מסדר ראשון (מהירות). בתחילת הפרויקט, בחרנו לעבוד עם מודל מסדר שני, דהיינו, הסוכנים מומשו ע"י מכוניות עם שליטה באמצעות דוושת הגז (שליטה בתאוצה) ומערכת היגוי. אולם, חרף הניסיונות הרבים שכללו כיוונוני הפרמטרים ושינוי פונקציית התגמול, מרחב המצב, מרחב הפעולה ופרמטרי הרשת, לא הצלחנו להגיע לתוצאות אליהן ייחלנו. בנוסף, ניסינו להגדיל את רשת הניורונים אך גם ניסיון זה העלה חרס בידינו. אי לכך, עברנו למודל מסדר ראשון שכן הוא פשוט ממנו. נדגיש כי הודות לשימוש במודל מסדר ראשון, הצלחנו לרכוש מדיניות טובה עם רשת ניורונים קטנה מזו שהשתמשנו בה עבור מודל מסדר שני. יתר על כן, השתמשנו בגישת גמול sparse, אשר לא הביאה להתכנסות עבור מודל מסדר שני. לסיכום, מודל דינאמי מסדר ראשון אפשר לנו להשיג את התוצאות המיוחלות.

יתר על כן, כפי שהסברנו בחלקים הקודמים, ככל שהבעיה אותה אנו מנסים לפתור קשה יותר, קטנה הסבירות שפונקציית תגמול sparse תאפשר רכישת מדיניות טובה. לכן, במקרים כאלו, גישת תגמול של reward shaping תביא לתוצאות טובות יותר. אנו גורסים כי מודל דינאמי מסדר שני מסובך משמעותית ממודל סדר ראשון וכראיה לכך, ביצועי המדיניות הנרכשת בשיטת reward shaping עבור מודל מסדר שני, עדיפים על פני גישת sparse reward.



מימוש ואימון האלגוריתם הלומד

לצורך מימוש האלגוריתם הלומד נדרשנו להתמודד עם מספר סוגיות בעלות חשיבות מכרעת בהצלחת תהליך הלמידה. להלן עיקרן:

- הגדרת מרחב מצב – מרחב המצב הינו אוסף התצפיות של הסוכן אשר מסכמות את מצבו. עפ"י

וקטור המצב של הסוכן, המדיניות תכריע אודות הפעולה עליו לבצע. כפי שהסברנו בחלקים הקודמים, על וקטור זה להיות אינפורמטיבי ככל שניתן, אולם תמציתי דיו כדי לאפשר למידה אפקטיבית, שכן וקטור זה מהווה קלט לרשת הניורונים שמגדירה את מדיניות הפעולה של כל סוכן. לכן, על וקטור זה להכיל אך ורק את הנתונים הקריטיים לצורך קביעת הפעולה שעל הסוכן לבצע. להלן המשתנים בוקטור מרחב המצב:

- תצפיות ממנגנון דמוי LIDAR - מדידת מרחק מאובייקטים בכיוונים שונים ע"י יריית קרני לייזר בזוויות שונות ומדידת מרחקי האובייקטים הנמצאים בכל אחת מהן. מנגנון זה נקרא RayPerception במנוע Unity. כמו כן, מכניזם זה מאפשר לנו את זיהוי סוג האובייקט – סוכן אחר, מכשול או מטרה. בחרנו להוסיף את התצפיות הללו למרחב המצב שכן הן קריטיות לצורך קביעת הפעולה הבאה של הסוכן – המדיניות חייבת להכיר את סביבת הסוכן לצורך קביעת פעולה אופטימאלית.
- בחרנו לירות 7 קרני לייזר בזוויות שונות כאשר תוצאת כל קרן מיוצגת ע"י 5 מספרים – 3 אשר מייצגים את סוג האובייקט ביוצג 1-hot-vector, ייצוג המרחק מהאובייקט הנצפה והאם האובייקט הנצפה הינו מסוג ידוע. לכן, בסה"כ מנגנון זה מייצר לנו 35 מספרים למרחב המצב.
- כיוון מנורמל למטרה – וקטור כיוון למטרה אשר מיוצג ע"י 3 מספרים (x,y,z) . וקטור זה חיוני כדי שהמדיניות תוכל להכווין את הסוכן לעבר מטרתו.
- כיוון התנועה הנוכחי של הסוכן – וקטור המייצג את כיוון תנועת הסוכן אשר מיוצג גם כן ע"י 3 מספרים. וקטור זה הכרחי שכן נתון זה בנוסף לכיוון תנועת הסוכן יאפשרו למדיניות לקבוע כיצד לסובב את הסוכן כדי שינוע לכיוון מטרתו.

○ מהירות הסוכן – מיוצגת ע"י וקטור בן 3 מספרים – מהירות בכל אחד מהצירים.

- **הגדרת מרחב הפעולה** - מרחב הפעולה הינו אוסף הפעולות האפשריות שהסוכן יכול לבצע. בהינתן וקטור מצב מסוים, על המדיניות לקבוע מה הפעולה מתוך מרחב הפעולה שעל הסוכן לבצע כדי למקסם את הגמול שלו. כפי שהסברנו בפרקים הקודמים, מרחב הפעולה צריך להיות עשיר דיו כדי לאפשר לסוכנים מגוון פעולות, אולם לא עשיר מדי שכן הדבר יפגע בהתכנסות למדיניות אופטימאלית.

סביבת ה-ml-agents מאפשרת להגדיר מרחב מצב בדיד ומרחב מצב רציף. משמעות הדבר היא אופי הערכים שאותם ניתן לקבל עבור כל פעולה. למשל, האם ניתן לסובב את הסוכן לכל זווית או רק לאחת מתוך סט זוויות מוגדר מראש. שימוש במרחב פעולה רציף מאפשר אוסף פעולות רחב יותר, אולם מגדיל משמעותית את ה-variance ומקשה על למידת המדיניות האופטימאלית. בסקר ספרות שביצענו, גילינו כי עבור משימות ניווט כאלו, מומלץ להשתמש במרחב פעולות בדיד. להלן פירוט אודות מרחב הפעולות שהגדרנו:

○ שינוי מהירות הסוכן – ניתן לבצע אחת מבין הפעולות הבאות:

- הגדלת המהירות ביחידה אחת
- הקטנת המהירות ביחידה אחת
- אי שינוי המהירות

○ שינוי כיוון תנועת הסוכן – ביצענו דיסקרטיזציה של כיווני הסיבוב האפשריים לערכים הבאים:

- שינוי כיוון התנועה ימינה ב- $3\frac{1}{3}^{\circ}$
- שינוי כיוון התנועה שמאלה ב- $3\frac{1}{3}^{\circ}$
- אי שינוי כיוון התנועה

- **הגדרת פונקציית התגמול** – לפונקציית התגמול אחריות מכרעת בתהליך רכישת המדיניות שכן באמצעותה אנו מגדירים מהן ההתנהגויות הרצויות (הגעה למטרה) והלא רצויות (התנגשות בסוכן אחר או קיר). בעת תהליך הלמידה, אנו משנים את המדיניות כך שתפעל לצורך מיקסום פונקציית התגמול. אי לכך, בחירת פונקציית תגמול מתאימה, תביא לרכישת מדיניות רצויה ומכאן חשיבותה הרבה.

כפי שצינו קודם לכן, ישנן שתי דוקטרינות מרכזיות לפונקציות תגמול – גישת sparse rewarding וגישת reward shaping. לכל אחת מהגישות הללו יתרונות וחסרונות. אנו בחרנו להיעזר בגישה הנוטה לכיוון גישת sparse rewarding. להלן עיקרי אופן התגמול שבחרנו:

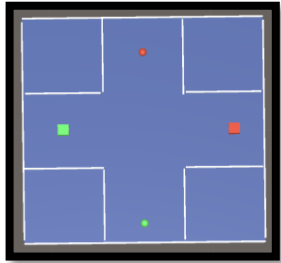

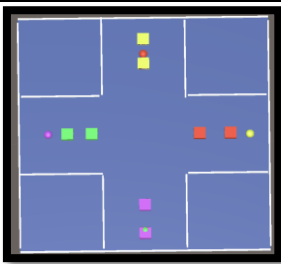

- גמול חיובי גדול יחסית עבור הגעה למטרה
- גמול שלילי גדול יחסית עבור התנגשות
- גמול שלילי קטן על כל frame – נועד לעודד הגעה מהירה למטרה

כדי למקסם גמול זה, על המדיניות לעודד הגעה מהירה למטרה, וזו בדיוק ההתנהגות שנרצה שהסוכנים ירכשו. כדי למקסם פונקציה זו, על המדיניות להגדיר חוקים חברתיים וכללי התנהגות אשר יאפשרו לסוכנים להגיע למטרתם ביעילות ובבטחה.

- **קביעת פרמטרי רשת הניורונים** – רשת הניורונים היא האלגוריתם אשר מממש את מדיניות הסוכנים – היא מקבלת כקלט את המצב הנוכחי של הסוכן (וקטור המצב) ומחזירה כפלט את הפעולה שעל הסוכן לבצע (פעולה ממרחב הפעולות). הגדרת מרחב המצב ומרחב הפעולה קובעת את הקלט והפלט לרשת הניורונים ומותירה לנו להגדיר את מספר השכבות החבויות, מספר הניורונים בכל שכבה ואת הפרמטרים השונים של תהליך הלמידה.

לצורך קביעת הפרמטרים הללו, התחלנו מפרמטרי רשת של פרויקטים דומים וביצענו hyper parameter tuning עבור פרמטרי הרשת האידיאליים למשימתנו. בסופו של דבר, בחרנו רשת עם 2 שכבות חבויות שבכל אחת מהן 256 ניורונים.

- **Curriculum learning** – שיטה זו נועדה לשפר את תהליך הלמידה. בשיטה זו, הלמידה מחולקת לשלבים כאשר בכל שלב הרמה עולה ביחס לשלב הקודם. בשיטה זו, בשלב הראשון, מתחילים מדוגמאות קלות יחסית וככל שמתקדמים בשלבים רמת הקושי עולה. שיטה זו שואבת השראה מאופן הלמידה של בני אדם. כדי לממש עיקרון זה, אימנו את הרשת בשלבים, מהקל אל הקשה ומימשנו מספר זירות ברמות קושי שונות. הגדרנו שתי תוכניות אימונים: תוכנית 1 - שלב 1, שלב 2 ושלב 3.1 ותוכנית 2 - שלב 1, שלב 2 ושלב 3.2.

שלב 1		מדיניות היעד – הגעה למטרה תוך התחמקות בסיסית מסוכנים אחרים, והימנעות מהתנגשויות עם דפנות הזירה.
שלב 2		מדיניות היעד – הגעה למטרה תוך התחמקות מסוכנים אחרים והימנעות מהתנגשויות עם דפנות הזירה.
שלב 3.1		מדיניות היעד – הגעה למטרה תוך התחמקות משמעותית מסוכנים אחרים והימנעות מהתנגשויות עם דפנות הזירה. בזירה זו, יש חיתוך בין מסלולים מצומצם יחסית.
שלב 3.2		מדיניות היעד – הגעה למטרה תוך התחמקות משמעותית מסוכנים אחרים והימנעות מהתנגשויות עם דפנות הזירה. בזירה זו, יש חיתוך מקסימאלי בין מסלולי הסוכנים.

תוצאות

לאחר תהליך האימון, הסוכנים רכשו מדיניות אופטימאלית – הם מצליחים לעבור את הצומת במהירות מבלי להתנגש זה בזה או במכשולים. כמו כן, נלמדו חוקים חברתיים מעניינים מאוד. למשל, בשלב 3.1, הסוכנים למדו שעל מנת למנוע התנגשויות והגעה מהירה ככל הניתן למטרה עליהם להיצמד לצד שמאל. בשלב 3.2, הסוכנים למדו להאט ולתת זכות קדימה, כאשר הם עמדו לקראת התנגשות עם סוכנים אחרים.

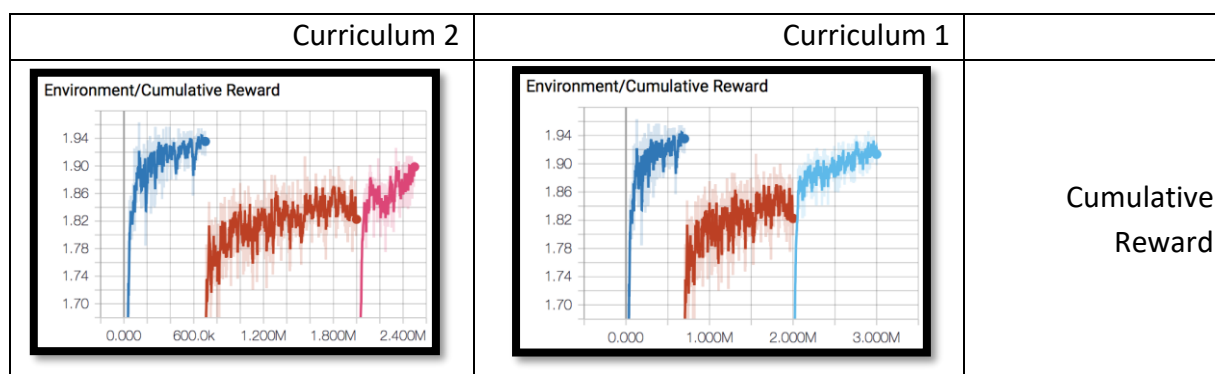
להלן קישור לאתר הפרויקט אשר מכיל סרטונים הממחישים את התוצאות הסופיות :

<https://sociallawlearning.carrd.co/>

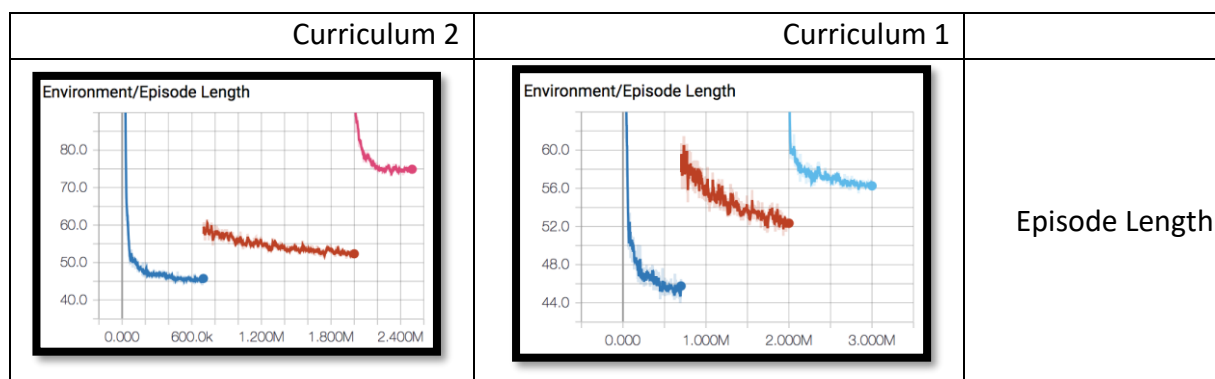
ניתוח ביצועים

להלן גרפים שנלקחו מtensor board אשר מכילים מידע אודות התכנסות הרשת. בגרפים השונים, כל צבע מייצג שלב אחר curriculum learning.

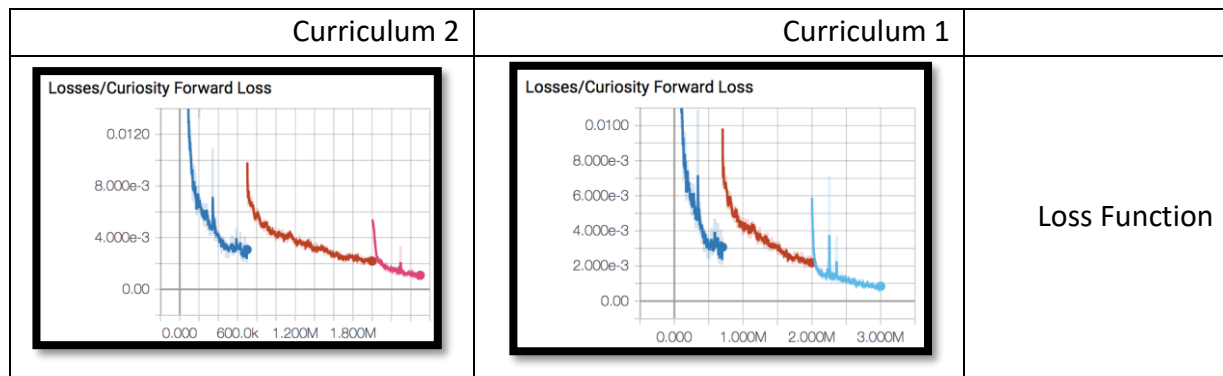
- Cumulative reward - מראה את הגמול המצטבר כתלות במספר האיטרציות. ניתן לראות כי בכל שלב, המדיניות הנרכשת משתפרת ומגדילה את הגמול המצטבר. במעבר בין השלבים ישנה ירידה בגמול המצטבר בשל העלייה ברמת הקושי.



- Episode length - מעיד על משך זמן הריצה של סוכן – מתחילת תנועה עד להגעה למטרה / התנגשות. כפי שניתן לראות, עבור כל שלב, המדיניות מצליחה לקצר את משך זמן הנסיעה. היות וגם הגמול משתפר, משמעות הדבר היא שהסוכנים מצליחים להגיע למטרה במהירות רבה יותר.



- Loss Function – זוהי בעצם פונקציית ההפסד אותה מנסה האלגוריתם למזער. בגרף אנו רואים כי האלגוריתם משתפר וההפסד קטן ככל שמספר הניסיונות עולה.



גישות אחרות לפתרון הבעיה

כדי לפתור את הבעיה עימה התמודדנו, בחרנו להיעזר באלגוריתם לומד המבוסס על למידה מחיזוקים (Reinforcement Learning). בחרנו בגישה זו, שכן היא מאפשרת לסוכנים לרכוש חוקים חברתיים ונורמות התנהגות בצורה חופשית, ללא התערבות שלנו.

פרט לפתרון אותו מימשנו, ישנם פתרונות אלגוריתמיים אפשריים נוספים לבעיה זו. נסקור כמה מהם:

אלגוריתם מחקה: באמצעות העזר ml-agents, ניתן להגדיר כי רשת הניורונים שמממשת את המדיניות תלמד לחקות התנהגות רצויה. ההתנהגות הרצויה יכולה להיות הקלטה של בן אדם שמפעיל בעצמו את הסוכנים. יתר על כן, ניתן להגדיר שהאלגוריתם המחקה יהווה warm start (אתחול חכם) ללמידה מניסוי וטעיה (למשל, לאלגוריתם הלומד אותו מימשנו בפרויקט). אולם, גישת פתרון זו מגבילה את חופש הפעולה של הסוכנים בלמידת חוקים חברתיים ומקבעת אותם לדפוסי התנהגות מסוימים, לפיהם פעל המפעיל האנושי שסיפק באמצעות "הקלטות" את ההתנהגות הרצויה.

פתרון מבוסס חיפוש יריסטי: ניתן להסתכל על הבעיה כעל בעיית חיפוש בגרף – מצבי הסוכן השונים הם צמתי הגרף, בעוד שניתן לעבור בין מצבים באמצעות הפעולות השונות המהוות את קשתות הגרף. כל סוכן מתחיל מנקודת התחלה כלשהי ושואף להגיע לנקודת סיום מוגדרת, כאשר לרשותו סט של פעולות אפשריות. היות והגרף המוגדר ע"י מרחב המצב ומרחב הפעולות הינו עצום, חיפוש בגרף כזה הינו משימה קשה מאוד חישובית, ואף לא אפשרית. כדי להתגבר על קושי זה, ניתן לבצע חיפוש יריסטי. יוריסטיקה הינה מטריקה שמעריכה את טיבו של כל צעד אפשרי אשר ניתן להיעזר בה לצורך חיפוש יעיל בגרף שלנו.

בפתרון מבוסס חיפוש יריסטי, בכל רגע נתון, כל סוכן יבצע חיפוש בגרף, אשר יתחיל מהמצב הנוכחי שלו ויבחר בפעולה אשר תספק לו את הערך היוריסטי העתידי הטוב ביותר, מבין כל הפעולות האפשריות.

אולם, המגבלה בגישה זו טמונה בכך שהיא לא מאפשרת לסוכנים לרכוש בעצמם דפוסי התנהגות, אלא לבצע את הפעולות השונות בהתבסס על היוריסטיקה אותה הגדרנו.

רעיונות להמשך

הפרויקט אותו מימשנו מהווה בסיס ידע ובסיס טכני לשימוש ב-unity וב-ml-agents לרכישת מדיניות של סוכנים נבונים באמצעות למידה מחיזוקים. סביבת הקוד אותה יצרנו מכילה את כל הקוד הדרוש ליצירה נוחה ופשוטה של סביבות מרובות סוכנים, הגדרת האלגוריתם הלומד ולתקשורת בין סביבת הסימולציה לרשת הניורונים. לכן, פרויקט זה מהווה בסיס טוב לשלל רעיונות להמשך הקשורים בלמידה בסביבות מרובות סוכנים. להלן כמה רעיונות לפרויקטי המשך.

תרחישים תחבורתיים מורכבים יותר: בפרויקט זה הסוכנים למדו נורמות התנהגות וכללים חברתיים לצורך תנועה בצומת לא מרומזר. הפרויקט שלנו מכיל את כל הדרוש לצורך הגדרת סביבות תחבורה שונות ולאמן אותן. לכן, ניתן להרחיב את הפרויקט שלנו ולהביא ללמידת חוקים חברתיים בתרחישים תחבורתיים מורכבים יותר, למשל, מערכת כבישים המכילה ריבוי סוכנים, מספר צמתים, מעגלי תנועה וכיוצא בזאת.

חוקים חברתיים עבור עבודת צוות של סוכנים: ניתן להיעזר בפרויקט זה לצורכי למידה של חוקים חברתיים שאינם קשורים לתחבורה, למשל עבור עבודת צוות של סוכנים. ניתן להגדיר משימה ולהטיל אותה על קבוצה של סוכנים. כדי לפתור אותה בצורה אופטימלית, על הסוכנים לשתף פעולה ולעבוד בצוות בצורה יעילה. בעת אימון אלגוריתם של למידה מחיזוקים, הסוכנים ירכשו מדיניות מתאימה אשר תתבסס על חוקים חברתיים עבור עבודת צוות ושיתוף פעולה.

סיכום ומסקנות

מטרת פרויקט זה הינה למידת חוקים חברתיים ונורמות התנהגות בסביבה מרובת סוכנים, המדמה צומת לא מרומזר. החוקים והנורמות הללו יביאו לחצייה בטוחה ומהירה של הצומת. לצורך מימוש הפרויקט, היה עלינו להתמודד עם קשיים טכניים ואלגוריתמיים לא מבוטלים וללמוד רבות על מבנה סביבת הסימולציה, תכנותה, תיאוריה של למידת מכונה בכלל ולמידה מחיזוקים בפרט ותכנון אלגוריתם לומד מבוסס למידה מחיזוקים על שלל פרמטריו הרבים.

במהלך העבודה על הפרויקט בחנו שיטות שונות למידול הבעיה (מודל דינאמי מסדר ראשון ושני) וגישות אלגוריתמיות שונות (הגדרת אופי ומבנה מרחבי המצב, הפעולה ופונקציית התגמול). בסופו של דבר, מצאנו קונפיגורציה מתאימה אשר הביאה לרכישת מדיניות אשר מביאה לחצייה מוצלחת של הצמתים השונים במהירות, וללא תאונות. יתר על כן, הסוכנים למדו מספר חוקים חברתיים ברורים: היצמדות לשמאל כאשר יש סוכן הנע מולם ועצירה ומתן זכות קדימה, במקרים בהם צפויה התנגשות עתידית. ניתן לראות הדגמה של החוקים הללו באתר הפרויקט - <https://sociallawslearning.carrd.co/>.

ביבליוגרפיה ומשאבים

נושא	מקורות
נורמות וחוקים חברתיים	https://en.wikipedia.org/wiki/Social_norm
רשתות נוירונים	https://en.wikipedia.org/wiki/Deep_learning
למידה מחזזקים	https://en.wikipedia.org/wiki/Reinforcement_learning
Proximal Policy Optimizer	https://openai.com/blog/openai-baselines-ppo/
Unity	<ul style="list-style-type: none"> ▪ https://en.wikipedia.org/wiki/Unity_(game_engine) ▪ https://learn.unity.com/
ml-agents	<ul style="list-style-type: none"> ▪ https://github.com/Unity-Technologies/ml-agents ▪ https://www.youtube.com/watch?v=x2RBxmooh8w ▪ https://www.youtube.com/watch?v=YDm8EIjpqa8 ▪ https://www.youtube.com/watch?v=y9ZkCG9hbfQ&t=1549s