**מסמך סוף סמסטר ב׳ – פרויקט גמר**

פרויקט מס׳ 7 – פיתוח משחק סוכנים נגד מרגלים

מנחה: בני בורנפלד

מגישים:

|  |  |
| --- | --- |
| שם מלא | תעודת זהות |
| יואב הראל | 303168231 |
| סתיו הרצוג | 208852327 |
| משה מנע | 313266223 |

**תיאור המשחק:**

המשחק מיועד ל2 שחקנים, שחקן אחד יהיה מרגל ושחקן שני מפעיל הסוכנים.   
מטרת המרגל היא להגיע ליעד הסופי מבלי להיתפס, ואילו מטרת מפעיל הסוכנים היא לתפוס את המרגל לפני שיגיע ליעד הסופי.

**מטרה:**

פיתוח המשחק סוכנים נגד מרגלים ומימוש 2 השחקנים כך שישוחקו ע"י המחשב באמצעות מודל של Reinforcement Learning.

פיתוח UI לאינטראקציה עם המשחק.

**נתונים:**

קובץ Json אשר מכיל מידע על שדות התעופה והקשרים ביניהם.

השתמשנו בid של שדות התעופה, בשם שלהם, מיקומם הגיאוגרפי (Lat, Long).

**ארכיטקטורת הפתרון**

תרשים:

A diagram of a server

Description automatically generated with medium confidence

הסבר:

Client – מומש ב React

Server – מומש ב Phyton ע״י FastAPI

Models – מומשו ב Phyton בשימוש בספריות Gym Open AI, Stable baseline

בקשת post מצד הclient – שליחת Observation כפרמטר שמורכב מ: Spy\_Position,Agent1\_Position,Agent2\_Position,Target\_Postion

השרת מעביר את ה Observation למודל המתאים. לאחר מכן, המודל מבצע Predict לפי הObservation שהתקבל.

במידה והקריאה למודל Spy, יוחזר Action יחיד לשרת, בקריאה למודל Agents יוחזר 2 Actions מהמודל לשרת והוא מעביר אותו כתשובה ל Client.

בהתאם ל response שהתקבל מהשרת, הClient מעדכן את הUI בהתאם למצב במשחק.

תיאור ארכיטקטורת מודלי למידת מכונה – Reinforcement learning:

[ מיקומי השחקנים והיעד [

A picture containing text, symbol, diagram, map

Description automatically generated

[ שדה תעופה / 2 שדות תעופה [

[ 20 שדות תעופה [

התרשים מיועד לשני המודלים, לשניהם יש את אותה Observation, Environment אך השוני ביניהם זה הAction וה - Reward.

Reward

-1 : הפסד.

1 : ניצחון.

-2 : תיקו (משחק מעל 30 צעדים).

צעד ניטרלי, מופעלת פונקציה היוריסטית (ניקוד מצטבר):

מודל Spy:

* + -0.1 : כאשר המרגל מרחק צעד מהמטרה, אך ביצע צעד שונה.
  + -0.1 : כאשר הצעד הנוכחי של המרגל קירב אותו למרחק של צעד אחד מאחד הסוכנים.

מודל Agents:

* + -0.1 : כאשר שני הסוכנים בחרו את אותו הצעד.
  + -0.1 : כאשר סוכן היה במרחק צעד מהמרגל אך ביצע צעד שונה.

**טכנולוגיות וספריות מרכזיות:**

* Gym Open AI - שימוש בספריה לצורך בניית Environment עבור 2 המודלים.

קישור לספרייה: [https://www.gymlibrary.dev](https://www.gymlibrary.dev/)

* Stable baseline 3 - שימוש באלגוריתם PPO ועזרים נוספים.

קישור לספרייה: [https://stable-baselines3.readthedocs.io/en/master](https://stable-baselines3.readthedocs.io/en/master/)

* amCharts 5 - שימוש בספרייה ליצירת UI למשחק.

קישור לספרייה: [https://www.amcharts.com](https://www.amcharts.com/)

* FastAPI – ליצירת שרת ב Python.
* Material UI – לעיצוב UI המשחק
* React – ספרייה שלJavaScript ל UI

**אלגוריתמים מרכזיים:**

PPO (Proximal Policy Optimization)

אלגוריתם פופולרי ללימוד מודלים של Reinforcement Learning, לקבלת החלטות בסביבה על מנת להשיג את המטרה.

לאלגוריתם PPO יש 2 רכיבים מרכזיים: רשת מדיניות אשר ממפה states ל actions, ורשת ערכים שמעריכה את ה reward הצפוי של כל state. רשת המדיניות ורשת הערכים מאומנות בעזרת הנתונים שנאספו ע״י המודל לאחר מס׳ איטרציות בסביבה.

שלבים של אלגוריתם PPO:

1. איסוף נתונים: המודל אוסף נתונים ע״י ביצוע המדיניות הנוכחית בסביבה ושומר את התוצאות.
2. חישוב יתרונות: רשת הערכים משמשת להערכת ה reward הצפוי של כל state שהמודל היה בו. היתרונות של כל action שנעשה ע״י המודל מחושבים כהפרש בין ה reward הצפוי ל state הבא לבין ה state הנוכחי.
3. עדכון מדיניות: רשת המדיניות מתעדכנת ע״י חישוב יחס ההסתברויות בין המדיניות הישנה לחדשה. המדיניות מתעדכנת בעזרת Gradient descent כדי למזער את ההפסד.
4. עדכון רשת הערכים: רשת הערכים מתעדכנת ע״י מזעור פונקציית ההפסד בין היתרונות הממשיים לחזויים.

חוזרים על שלבים 1-4 מס׳ איטרציות עד שהמדיניות מתקרבת לאופטימליות.

האלגוריתם מתוכנן לאזן בין חקירה (explore) של צעדים חדשים, לבין שימוש בצעדים שכבר סווגו כצעדים טובים (exploit).

BFS (Breadth First Search)

אלגוריתם למציאת מסלול קצר ביותר, השתמשנו באלגוריתם בפונקציה היוריסטית כדי לבדוק את המרחקים בין הסוכנים למרגל ובין המרגל ליעד המטרה.

קישור לאפליקציה:

Catch me if you can - <http://proj.ruppin.ac.il/cgroup4/prod/finalproj/build1/index.html#/>