

פרויקט סיום - רובוטים אוטונומיים

מגישים: אריאל יחזקאל 211356449 ואלמוג אמיגא 211543285

הקדמה

בפרויקט זה יצרנו סביבת OpenAI Gym המיועדת ללמד סוכני Reinforcement learning את היכולת לשלוט ברחפן דו מימדי. אנחנו עשינו את הקורס למידה מונחית חיזוקים של פרופ' עמוס עזריה. ורצינו לשלב את הנושא עם הקורס הנוכחי ולכן בחרנו במשימה מספר 2. (כאן, הרעיון הוא לבנות פרויקט קוד פתוח איכותי שיאפשר סימולציה של אלגוריתמי בקרה שונים וכן יכולת להוות פלטפורמה למכונות לומדות כדוגמת (Reinforcement Learning)).

מחלקות

יצרנו את מחלקה של הסביבה של הרחפן ובנוסף יצרנו מחלקה שמייצגת מבחינה פיזיקלית ודינמית את הרחפן והשפעות הכוללות על הרחפן (למשל, כוח הכבידה).


את מחלקת הרחפן שלנו (drone.py) יצרנו על ידי ספריית pymunk (ספריה בפייתון של פיזיקת אובייקטים דו ממדיים), היא בעלת המון פונקציות מובנות ולכן נעזרנו בקישור זה כדי ליצור מחלקה זו.

<http://www.pymunk.org/en/latest/pymunk.html#pymunk.Space>

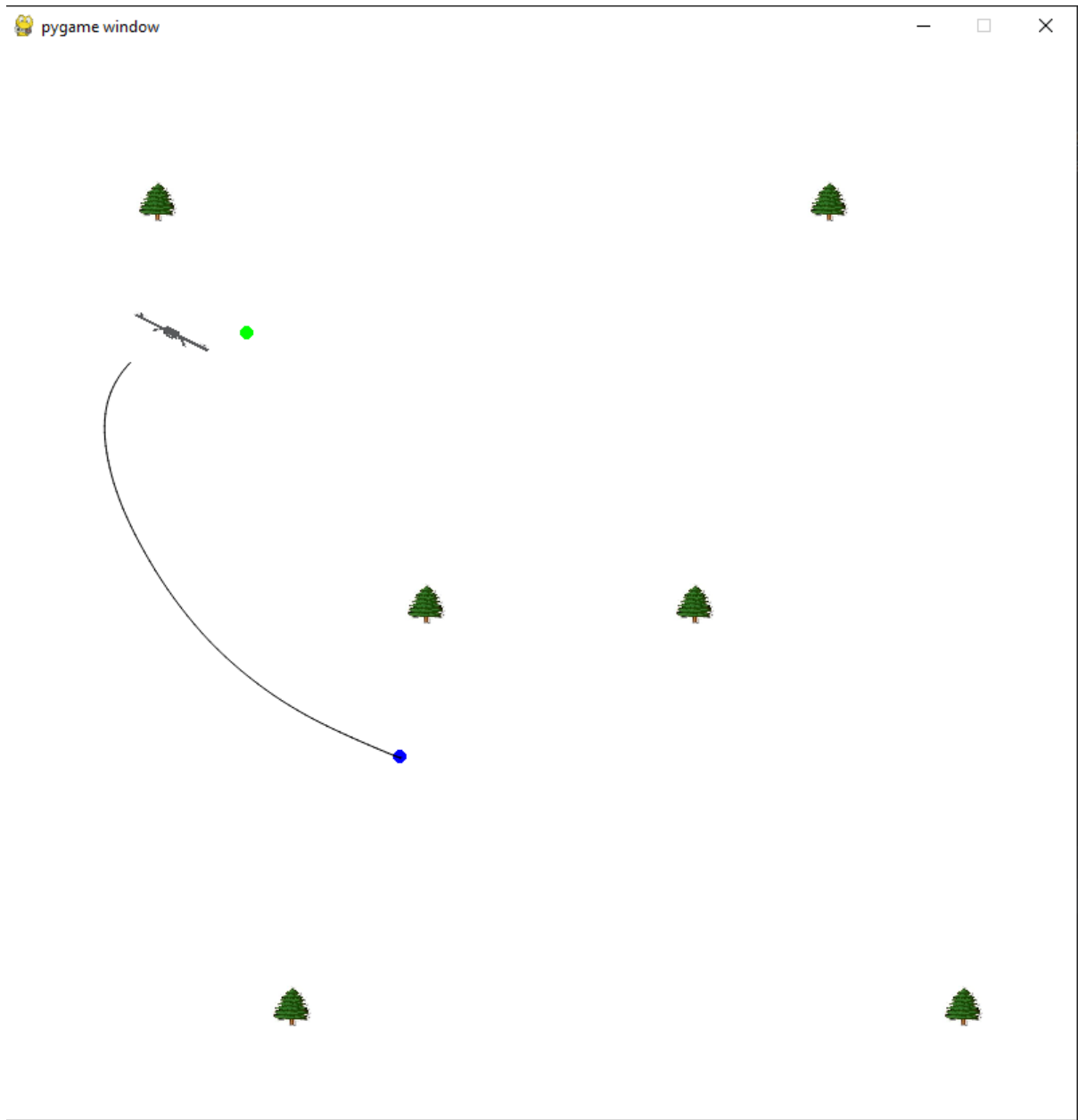
הסבר קצת על הרחפן:

הרחפן מורכב משלושה חלקים גוף, מנוע שמאלי ומנוע ימני. הכוחות שהם מפעילים הם אלה שמייצגים את מרחב הפעולות (actions). על מנת להראות זאת בצורה ויזואלית יפה לעין הצגנו את הרחפן כתמונה ובנוסף ישנו קו שמייצג את הדרך שעבר הרחפן מנקודת ההתחלה (נקודה כחולה) עד לאן שהגיע (בין אם התרסק או הגיע ליעד).

בשביל ליצור את מחלקת הסביבה של הרחפן נעזרנו בסרטון יוטיוב זה

.  [Creating a Custom OpenAI Gym Environment for your own game!](#)

בסרטון ניתן לראות מכונית מירוץ דו מימדית שצריכה ללמוד לסיים מסלול בעזרת AI. השתמשנו ברעיון זה, רק שבמקום מכונית מירוץ לקחנו רחפן דו מימדי שמטרתו להגיע לנקודה אקראית במפה, נקודת היעד אשר מסומנת כנקודה ירוקה, תוך כדי שהוא נזהר ממכשולים בדרך (בתוכנה המכשולים מסומנים כעצים ובגודל של 30X30 פיקסלים). בסרטון ניתן לראות כיצד יש לפתח סביבת OpenAI Gym. וזה בדיוק מה שעשינו. נעזרנו בפונקציות הקבועות ובקבצים הספציפים שצריך ליצור על מנת לפתח סביבה זו.



בנוסף יש מחלקת `z` אשר משתמשת באלגוריתמי `z` על מנת לאמן ולבחון את הסוכן שלנו. כל מה שצריך לעשות זה לשנות את שורה מספר 9 בקובץ לפי ההוראות שנמצאות שם. האלגוריתמים שניתן להשתמש הם אלגוריתמי RL מובנים של ספריית `stable_baselines3` כל מה שצריך לעשות זה לשנות את `import` של האלגוריתם אותו אתה רוצה לבדוק.

אלגוריתמים לדוגמא: `A2C`, `Q-learning`, `Proximal Policy Optimization`.

ניתן למצוא את מימושי `import` כאן:

<https://stable-baselines3.readthedocs.io/en/master/modules/ppo.html>

בנוסף יש מחלקה אחרונה שנקראת `running_env` שהיא סוגרת את הGUI.

תהליך האימון והסבר על הסביבה

בתחילת כל ריצה, הרחפן ממוקם אקראית בסביבה (נקודה כחולה) ובנוסף עם זווית שיפוע אקראית מ-45 מעלות עד 45- מעלות.

כאשר הריצה מסתיימת, הסוכן מקבל פרס שלילי של -10 אם הרחפן עולה על זווית של 90 מעלות או אם הרחפן טס מחוץ לאזור המותר או אם הוא מתנגש באחד העצים בסביבה. בנוסף, כל ריצה מוגבלת במשך המספר המרבי של שלבי זמן - שבחרנו בו להיות 500. בכל שלב זמן, אם הרחפן לא הגיע לנקודת היעד הסוכן מקבל פרס שלילי של -0.01.

כאשר הריצה מסתיימת, הסוכן מקבל פרס חיובי של 10 אם הוא מגיע לנקודה הירוקה שנבחרה להיות כיעד לרחפן.

מרחב הפעולות (actions) שעומד לרשות סוכן ה-RL מורכב משני ערכים בין -1 ל 1. הערך 1- פירושו אין כוח, וערך 1 הוא הכוח המקסימלי האפשרי. הערכים הם: הכוח במנוע שמאלי והכוח במנוע הימני.

מרחב התצפית (observation) מורכב משמונה ערכים, כולם נעים בין -1 ל 1. הערכים הם: speed_x, speed_y, phi, angle, distance_x, distance_y, position_x, position_y

שני הראשונים מייצגים את המהירות של הרחפן בצירי x ו-y.

המספר השלישי מייצג את המהירות הזוויתית הנוכחית של הרחפן.

המספר הרביעי מייצג את זווית הרחפן.

המספרים החמישי והשישי מייצגים את המרחק של הרחפן מהיעד (הנקודה הירוקה) בצירי x ו-y.

המספרים השביעי והשמיני מייצגים את מיקומו של הרחפן בסביבה.

נקודות להמשך:

- הוספת סנסורים למרחב התצפית (למשל סנסור שמודד מרחק מעץ)

עבודות דומות:

- <https://github.com/ZikangXiong/two-d-nav-gym>
- <https://github.com/Dormey/Python-Block-Battle-Game>