* מה הפרוייקט?
  + המשחק Catch me if you can, מיועד ל2 שחקנים, שחקן אחד יהיה מרגל ושחקן שני מפעיל הסוכנים. לוח המשחק הוא מפת עולם עם 40 שדות תעופה, אשר מחוברים במידה וקיימת בינהם טיסה. מטרת המרגל היא להגיע ליעד הסופי מבלי להתפס, ואילו מטרת מפעיל הסוכנים היא לתפוס את המרגל לפני שיגיע ליעד הסופי.
  + הנתונים הם מקובץ JSON המכיל מידע על שדות התעופה, ולכל שדה תעופה יש מערך המכיל את השדות אליהם מקושר ע"י טיסה. (הוספת תמונה של מערך הטיסות בReact)
  + הפרוייקט הוא לממש את 2 השחקים כך שישוחקו ע"י המחשב באמצעות מודל של Reinforcement Learning
* מזה Reinforcement Learning?
  + זהו מודל למידת מוכנה אשר מתבסס על תגמול להתנהגות רצויה וענישה על התנהגות לא רצויה. באופן כללי, המודל מסוגל לתפוס ולפרש את סביבתו לביצוע פעולות וללמוד ע"י ניסוי וטעיה.
  + שיטה זו מייחסת ערכים חיוביים לפעולות הרצויות לעידוד הסוכן וערכים שליליים להתנהגויות לא רצויות. זה מתכנת את הסוכן לחפש תגמול כולל לטווח ארוך ומקסימלי כדי להשיג פתרון אופטימלי. כך לומד הסוכן עם הזמן להימנע מהשלילי ולחפש את החיובי.
* איך ממימשנו:
  + שימוש בgym
    - Gym היא ספריית קוד פתוח של OpenAI אשר מספקת את הסביבה והמעטפת לפיתוח ובדיקות של מודלי Reinforcement Learning. Gym מספקת מספר סטים של סביבות ללימוד Reinforcement Learning וקיימת אפשרות לבנות סביבה בהתאמה אישית (כפי שעשינו), כל עוד מספקים 3 פונקציות חובה לסביבה (init, step, reset).

Gym בעצם עושה את הקשר בין הסביבה למודל.

* בחרנו בGym כיוון שזוהי הספרייה הפופלרית ביותר ברשת והDocumentation שלהם הוא המפורט והברור ביותר.

* + שימוש באלגוריתם PPO (Proximal Policy Optimization)
    - אלגוריתם פופולרי ללימוד מודלים של Reinforcement Learning, לקבלת החלטות בסביבה על מנת להשיג את המטרה.
    - האלגוריתם מתוכנן לאזן בין חקירה של צעדים חדשים, לבין שימוש בצעדים שכבר סווגו כצעדים טובים.

* + הסבר על action space, הסבר על Action mask
    - זהו סט הפעולות האפשריות למודל , לדוגמא סנייק ישנן 4 פעולות אפשריות (למעלה, למטה, ימינה, שמאלה). בכל צעד המודל בוחר פעולה מתוך הaction space.
    - במשחק שלנו, כל actionמייצג שדה תעופה, והaction space הוא כל שדות התעופה.

כלומר בחירה בaction מסויים תעביר את השחקן לשדה התעופה הנבחר.

* + - כיוון שלא ניתן להגיע מכל שדה לכל שדה, בכל שלב ישנם פעולות מתוך הaction space שהן לא חוקיות ולכן השתמש בAction mask.
    - Action mask מגביל את מספר הפעולות מתוכן אפשר לבחור, בהתאם למצב הנוכחי בו נמצא השחקן. זוהי פונקציה שנקראת לפני כל צעד ומחזירה למודל אילו פעולות חוקיות באותו הצעד.

* + הסבר על observation – תמונת מצב
    - זוהי בעצם הסביבה (העולם) אותו רואה המודל בכל שלב ועל פיה הוא בוחר את הצעד הבא שלו.
    - במשחק שלנו, הobservation מייצג את מיקומי השחקנים והיעד, כלומר מערך בגודל 4. [spy\_position, agent1\_position, agent2\_position, target\_position]
  + הסבר step
    - הפוקציה step מקבל את הaction שנבחר על ידי המודל, בפוקציה זו מיישמים את לוגיקת המשחק, ומחזירים למודל את הobservation החדש והreward המתאים.
  + Reward - על פי מודל הSpy
    - 0 : צעד כאשר המודל לא נתפס או הגיע ליעד.
    - -1: כאשר המודל הפסיד.
    - 1 : כאשר המודל ניצח.
    - ~~2 : כאשר המודל ניצח ע"י המסלול הקצר ביותר.~~
* הצגת ממצאים בצורה גרפית
  + גרף עקומה למידה וממוצע reward
  + Pie chart נצחונות/הפסדים (לפני / אחרי האימון)
* הצגת משחק אבטיפוס בui