הרצת התכנית תיערך משורת ההרצה (command line) בעזרת הסינטקס:

הנתיב לקובץ ה שמתאר את סביבת ההרצה.

קובץ הjson מקודד מילון שמכיל את שם הקובץ תחת השם ,  
את גודל המבוך תחת השדה (המבוך יהיה בתוך ריבוע בגודל , וימורכז בנקודה ) ועוד 2 רשימות:

הרשימה הראשונה, maze מקודדת את המבוך שיבנה בסביבת הpybullet

הרשימה תכיל רשימות של מסילות פוליגונליות, כל מסילה פוליגונלית מיוצגת על ידי רשימת נקודות   
המסילה הפוליגונלית המיוצגת על ידי רשימת הנקודות

שקולה לרשימת הקטעים:

בפועל, למכשולים יש עובי שמוגדר בתור בקובץ .  
לכן בפועל כל ישר מתורגם למלבן עם עובי ואורך .

הרשימה השניה, "positions", תהיה רשימה של מילונים של נקודת ההתחלה של רכב זווית ההתחלה של רכב ביחס לציר ה: , ונקודת הסיום שלו: .

הם וקטורים באורך 2, שמקודדים את ערך ה של הנקודות.

הערך הוא זווית ברדיאנים, כלומר מספר ממשי בטווח .

לכל איבר ברשימה התכנית תייצר מכונית בסביבת הסימלציה בנקודת ההתחלה עם הזווית הנתונה (ביחס לציר הx), ותגדיר את המטרה שלה להיות נקודת הסוף.

מצורפים עם החבילה מספר קבצי json לדוגמה בתיקייה mazes.

**שים לב:** התכנית מניחה שהמבוך תקין, מבוך תקין מכיל שרשראות פוליגונאליות שלא חותכות את עצמן או זו את זו, והיא עלולה שלא לפעול כראוי בהינתן קלט לא מתאים.

*:*

,   
התכנית תדפיס ערכים אינפורמטים לגבי הריצה כמו מספר הצעדים שעברו בכל 500 צעדים, ומידע על ההחלטות של המכוניות[צריך ניסוח יותר טוב]

,   
התכנית תדפיס בסוף הריצה גרף של matplotlib עליו המפות שנוצרו מסריקות הרכבים, והדרך שכל מכונית נסעה.

v-, visualize--   
התכנית תריץ את הסימולציה בסביבת הbullet באופן ויזואלי.\*

*--*plot\_maze,-m *התכנית תדפיס את המבוך הנתון בעזרת , במקום להריץ את הסימולציה*

--max\_time MAX\_TIME,-t MAX\_TIME

התכנית תרוץ במשך MAX\_TIME *צעדים, ואז תעצור.*

\*בהרצת זמן הריצה יארך משמעותית בגלל חישובים גרפים נוספים של המעבד.