***מסמך דרישות תוכנה (SRS)***

1. **שם הפרויקט : MazeRun**  
   קורס: אלגוריתמים מתקדמים  
   תאריך: יולי 2025  
   שמות השותפים: אלעד אהרון, רועי טרבינוביץ, רותם רוזנברג, מיכאל גרינברג
2. **תקציר:**  
   MazeRun הוא משחק מבוך דו־ממדי רספונסיבי לדפדפן, בו שחקן אנושי מנסה להגיע ליציאה לפני שאויב אחד או שניים תופסים אותו. המשחק מבוסס על לוגיקה בזמן אמת, הכוללת בינה מלאכותית שמבצעת חישובי מסלול בזמן אמת בעזרת אלגוריתם A\*. המבוך נבנה אקראית אך נבדק לפתרונות, וישנם פריטים כמו מטבעות, רנדומיזר ותור נוסף שנכנסים כחלק מהאתגר. המשחק כתוב כולו ב־HTML, CSS ו־JavaScript ללא תלות בשרת.
3. **קישורים:**

* הדגמת משחק: [לחץ כאן](https://rotemrosenberg.github.io/MazeRun/html/index.html)
* GitHub: [לחץ כאן](https://github.com/RotemRosenberg/MazeRun)
* מצגת הפרויקט: [לחץ כאן](https://docs.google.com/presentation/d/1nBWyMiAreDt2IcaTmkbH-jCorvXP-vho-sqRngnGzPY/edit?usp=sharing)

1. **תיאור הבעיה:**  
   המשחק נועד לתת מענה לחוויה אינטראקטיבית שבה השחקן צריך להתמודד מול אויבים שמגיבים באופן חכם למיקומו בזמן אמת. האתגר היה לבנות AI שאינו צפוי אך הוגן, מבוך שתמיד פתיר, ולשלב אלמנטים אקראיים תוך שמירה על ביצועים בזמן אמת.
2. **מאפיינים פונקציונליים**

* משחק רספונסיבי הפועל ב־Canvas
* תנועה של השחקן באמצעות חיצים (keydown events)
* יצירת מבוך אקראי בגודל 30x30 עם 25% או 35% קירות
* אויב אחד (מצב רגיל) או שניים (מצב Hard)
* Power-ups מיוחדים:  
  - Extra Turn מאפשר תור נוסף לשחקן   
   Randomizer - מגריל מבוך חדש מיידית
* AI מבוסס A\* עם ארבעה מצבים:  
  • תקיפה (אם קרוב מאוד)  
   • חסימה (אם במרחק בינוני)  
  • רדיפה (במרחק רגיל)  
   • חיזוי (אם רחוק מאוד)
* מנגנון ניקוד לפי מטבעות שנאספו
* מסך תפריט ראשי (index.html), הדרכה (tutorial.html), מסך משחק (game.html)
* הודעות סוף משחק עם SweetAlert2

1. **אלגוריתמים ולוגיקה:**

* A\* Pathfinding:  
   •חישוב f(n) = g(n) + h(n)  
   h(n): מרחק מנהטן מותאם עם בונוס קרבה לשחקן ועונש קרבה ליציאה  
  • עלות נוספת אם יש Power-up במשבצת או סמיכות לקירות
* BFS:  
  • משמש לבדיקת תקפות המבוך בכל יצירה  
  • מוודא שיש מסלול גם לשחקן וגם לאויב
* לוגיקת AI :  
  • ניתוח מיקום שחקן ואויב בכל תור  
  • מעבר בין אסטרטגיות לפי מרחק ומצב

1. **ארכיטקטורת מערכת:**

* HTML + CSS + JS
* Canvas לייצוג ויזואלי של לוח המשחק (800x800)
* הפרדה לקבצים:  
   index.html תפריט ובחירת רמת קושי  
   game.html לוח המשחק  
   tutorial.html עמוד הסבר  
   game.js מנוע המשחק הראשי  
   index.js, tutorial.js התנהגות העמודים האחרים  
   style.css עיצוב בסיסי לכל הרכיבים
* ללא שימוש בספריות צד שלישי, מלבד SweetAlert2

1. **מבנה תיקיות וקבצים:**

* /html:  
  index.html, game.html, tutorial.html
* /js:  
  game.js, index.js, tutorial.js
* /css:  
  style.css

1. **אתגרים מרכזיים ופתרונות:**

* מבוכים לא פתירים: פתרנו עם BFS בעת כל יצירת לוח
* התנגשות "חילוף מקום": זיהוי מצב בו שחקן ואויב מחליפים תאים באותו תור
* איזון התנהגות האויב: פונקציית heuristic מותאמת שגרמה לו להתקרב אך לא לשהות על היציאה
* שמירה על ביצועים: חישובי A\* נעשים רק בתורו של האויב

1. **חוזקות ומרחב לשיפור  
   חוזקות:**  
    ✔ חוויית משחק מאתגרת ודינמית  
   ✔ בינה מלאכותית אדפטיבית עם 4 מצבים  
   ✔ ביצועים טובים בזמן אמת גם בדפדפן  
   ✔ קוד מודולרי ונוח להרחבה  
   ✔ ממשק ויזואלי נעים ואינטואיטיבי

**שיפורים עתידיים:**

* הוספת צלילים ואנימציות
* מעבר בין שלבים
* שמירת שיאים או תוצאות
* הוספת מצב מרובה שחקנים