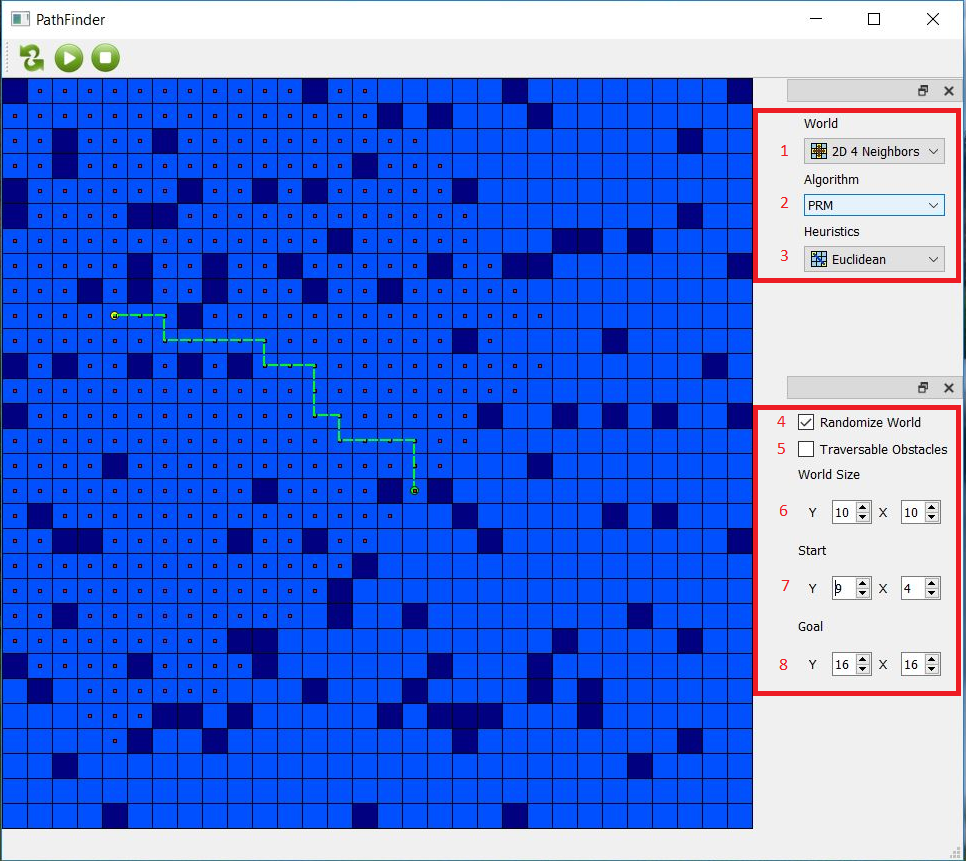
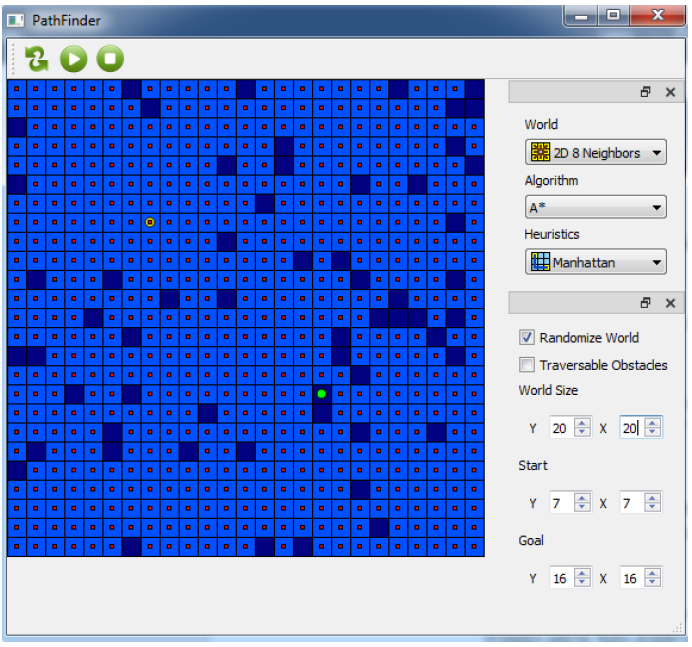
**מטלה 2**

**שאלה 1**



הסבר על החלקים המסומנים:

1. האם לכל צומת יש 4 או 8 שכנים, כלומר, האם ניתן להתקדם גם באלכסון או לא.
2. בחירת האלגוריתם לחיפוש המסלול הקצר ביותר: דייקסטרה, \*A, PRM או מושבת נמלים.
3. בחירת ההיוריסטיקה: מרחק מנהטן או מרחק אוקלידי.
4. האם העולם והמכשולים שבו נקבעים בצורה רנדומלית או לא.
5. האם המסלול יכול לחצות מכשולים.
6. גודל המפה.
7. נקודת ההתחלה.
8. נקודת הסיום.



בהרצה הנ"ל ניתן לראות כי לא נמצא מסלול מהנקודה (7, 7) לנקודה (16, 16), מכיוון שהאחרונה נמצאת על מכשול שלא ניתן לעבור דרכו.

**שאלה 2**

1. Dijkstra: מציאת המסלול הקצר ביותר ע"פ האלגוריתם הקלאסי של דייקסטרה. מקבל כקלט גרף מכוון ממושקל (עם משקלים לא שליליים) ומחזיר את המסלול הקצר ביותר מנקודת ההתחלה אל נקודת הסיום.
2. \*A: מציאת המסלול הקצר ביותר ע"פ האלגוריתם \*A. מקבל כקלט גרף מכוון ממושקל (עם משקלים לא שליליים) ומחזיר את המסלול הקצר ביותר מנקודת ההתחלה אל נקודת הסיום ע"י הפונקציה plan\_gen שמתכננת את המסלול האופטימלי לבעיה.

**שאלה 3**

הקוד הראשי מורץ ומוצג באמצעות GUI. ראשית, התכנה מציירת את המפה כברירת מחדל והמשתמש יכול לשנות את הפרמטרים, כגון גודל המפה, נקודות התחלה וסיום, מיקום המכשולים וכו', בנוסף המשתמש יכול לבחור באיזה אלגוריתם להשתמש, האם אפשר ללכת באלכסון, את נקודת ההתחלה ואת נקודת הסיום, וכאשר האלגוריתם מופעל, ישנן מספר פונקציות שאחראיות לצייר את המסלול שהתקבל בהתאם לאלגוריתם הנבחר (דייקסטרה, \*A, PRM או מושבת נמלים). בסיום, מוצג המסלול הקצר ביותר מנקודת ההתחלה אל נקודת הסיום.

**שאלות 4+5**

|  |
| --- |
| דייקסטרה בלי אלכסונים; 279 מילי שניות |
| דייקסטרה עם אלכסונים; 276 מילי שניות |
| \*A בלי אלכסונים; 60 מילי שניות |
| \*A עם אלכסונים; 25 מילי שניות. |

**מסקנות:**

* אלגוריתם \*A מהיר יותר מאשר דייקסטרה.
* כאשר האלגוריתם יכול לעבור דרך אלכסונים במסלול שהוא היוצר, שני האלגוריתמים, דייקסטרה ו-\*A מצאו את המסלול בפחות זמן מאשר מסלול שלא יכול לעבור דרך אלכסונים.