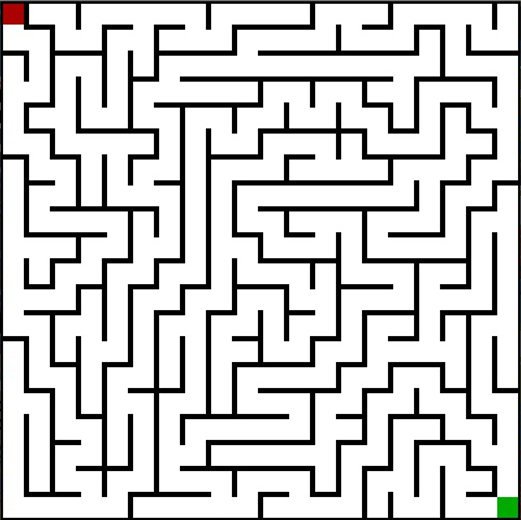
מיצר מבוך – Maze generator



מגיש: ראם מיקולסקי

ת"ז: 326542982

כיתה: י'2

שם המורה: שמואל מעודה

שימושים בתוכניות וטכנולוגיות:

* סביבת פיתוח: Visual Studio Code
* קומפיילר + לינקר: Tasm + TLink בשביל שפת Assembly 8086
* אימולטור: DosBox
* סביבת יצירת תוכן וויזואלי: Aseprite

**תוכן עניינים תיק פרויקט:**

1. שער פרויקט + פרטי פרויקט
2. תוכן עניינים
3. הסבר על הנושא
4. תרשימי זרימה
5. תרשימי זרימה – המשך
6. מדריך למשתמש
7. מדריך למשתמש – המשך
8. תיאור משתנים קבועים ופונקציות עד הדף האחרון
9. מקורות מידע

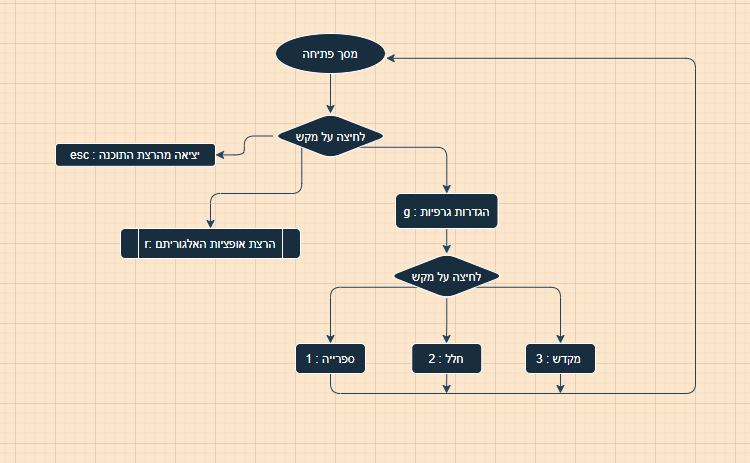
למה בחרתי בנושא זה:

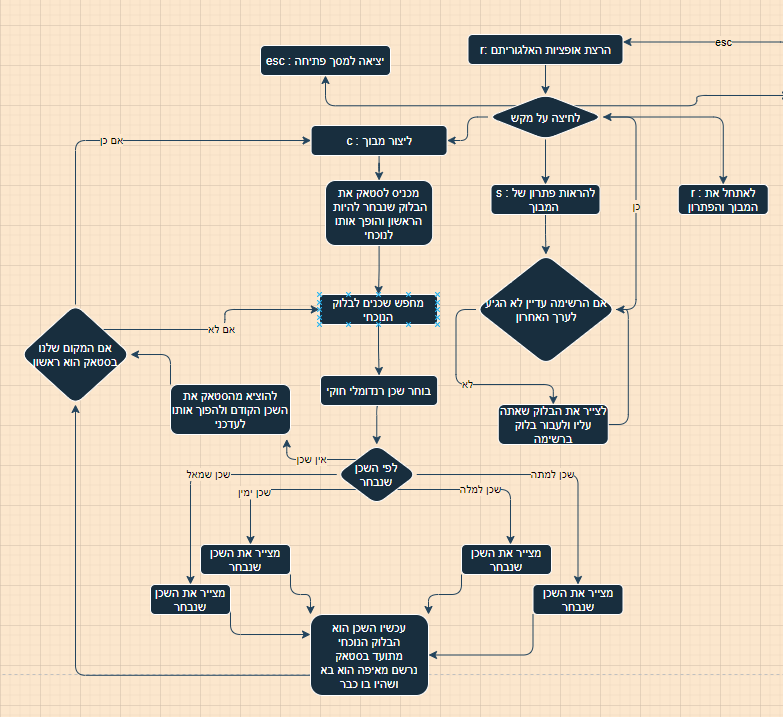
מיצר מבוכים הוא אחד הנושאים הנפוצים יותר באלגוריתמיקה. הרעיון מתבסס על כך שהאלגוריתם בוחר רנדומלית לאיפה להמשיך את המבוך ואם אין לו מקום להמשך את המבוך הוא הולך אחורה עד לאזור שהוא יכול להמשיך.

בתור מתכנת מתחיל יחסית רציתי ליצור פרויקט שיעזור לי כמה שיותר להשתפר בתכנות ושיהיה מהנה. ראיתי ושמעתי בהרבה מקומות שכתיבת אלגוריתמים לפרויקטים היא דבר מאוד משפר ומהנה אז החלטתי ללכת לכיוון הזה. עשיתי קצת מחקר וראיתי שאלגוריתם יותר נפוץ היה Maze Generator ופתירת מבוך. לכן החלטתי לעשות את הפרויקט על זה.

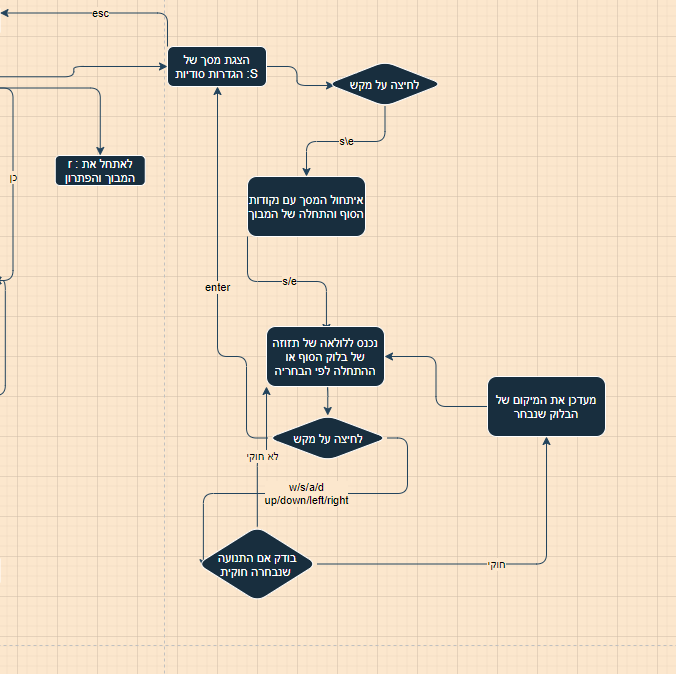
*תרשימי זרימה:*

מסך פתיחה:



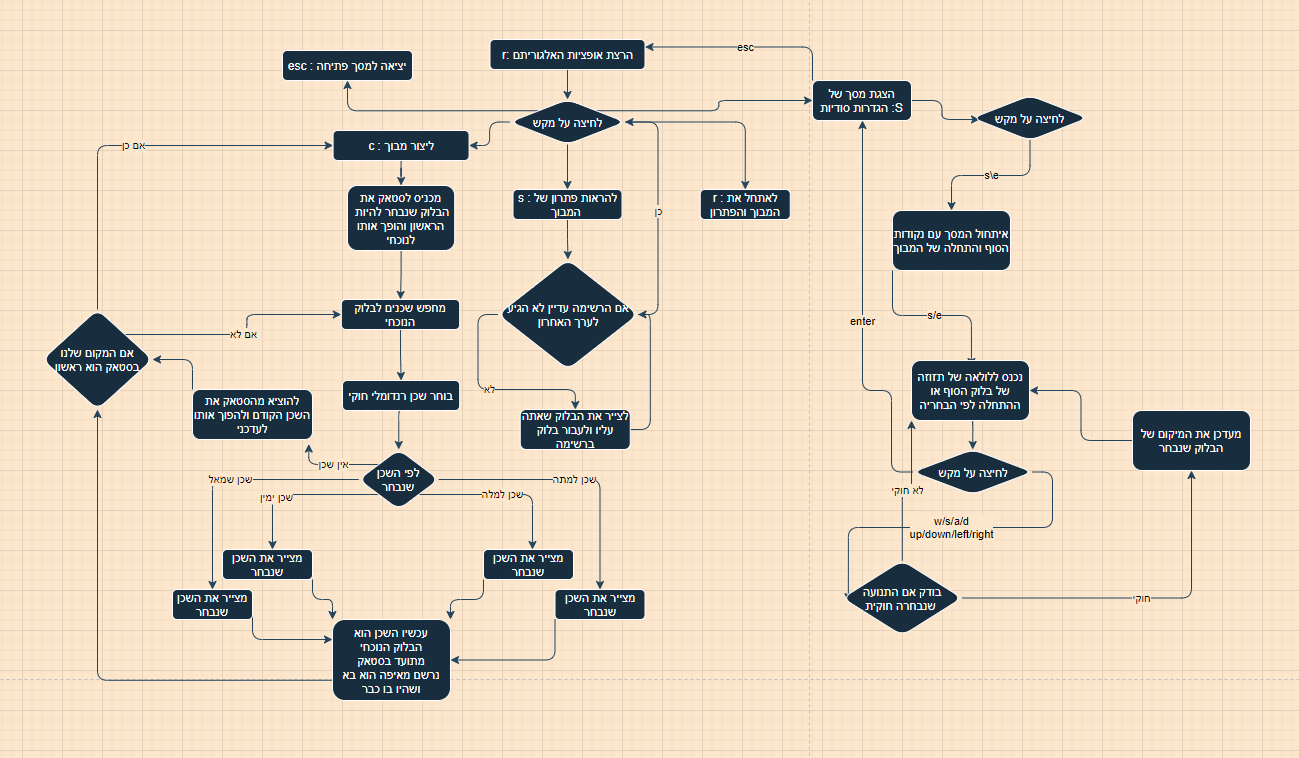


### **מסך אלגוריתם:**

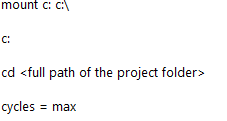


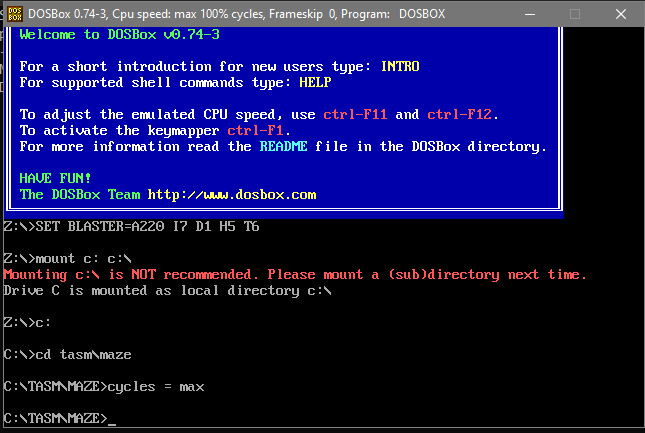
**מסך הגדרות סודיות:**

כל מסך הרצת האלגוריתם:



*מדריך למשתמש:*

* להתקין את תיקיית הפרויקט
* להתקין את התוכנה dosBox מהלינק הזה: <http://data.cyber.org.il/assembly/dosbox.exe>
* לפתוח את dosBox ולכתוב את השורות האלו:
* דבר זה אמור להראות כך:

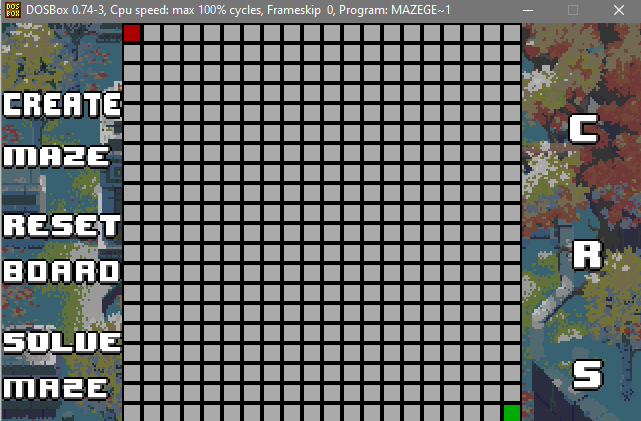


* לאחר מכן כתבו run maze ואז לחצו tab תמחקו את הסיומת הקובץ של השם שיצא לכם ותלחצו Enter - 

דבר זה אמור להביא אתכם למסך הפתיחה של האלגוריתם:



* עכשיו ישנן שתי אופציות או שתתחילו להריץ את מסך הרצת האלגוריתם בלחיצה על מקש r או ללחוץ לשינוי המצב הגרפי g 
* g: יופיע מסך ואז תבחרו מה יהיה המצב הגרפי 1 – ספרייה 2 – חלל 3 – מקדש (המצב ההתחלתי)
* r: יופיע מסך הרצת האלגוריתם אם תלחצו על מקש:
  + c – המבוך יבנה
  + s – פותר את המבוך
  + r – מאתחל את כל המבוך
  + מקש סודי – להגדרות סודיות
* אם תצליחו להגיע להגדרות הסודיות יקפוץ לך מסך הסבר – אתה יכול עכשיו לשנות את מיקום התחלת המבוך ומיקום הסתיימותו, פירוט במסך הסודי – רק למגניבים.



* תיאור משתני וקבועי התוכנה:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | שם משתנה או קבוע | סוג | גודל | תפקיד |
| 1 | filehandle | משתנה | Word – 2 byte | חלק מ- BMP |
| 2 | Header | משתנה | מערך של 54 בתים | הדר לש הBMP |
| 3 | Palette | משתנה | מערך של 1024 בתים | פלטה של הBMP |
| 4 | ScrLine | משתנה | מערך של 320 בתים | חלק מ - BMP |
| 5 | ErrorMsg | משתנה | מערך של 4 בתים | חלק מ - BMP |
| 6 | Secret | משתנה | מחרוזת – מספר בתים לפי גודל המחרוזת | תמונה להגדרות הסודיות |
| 7 | libraryS | משתנה | מחרוזת – מספר בתים לפי גודל המחרוזת | שם תמונה של פתיחת האלגוריתם |
| 8 | library | משתנה | מחרוזת – מספר בתים לפי גודל המחרוזת | שם התמונה של הרצת התוכנית |
| 9 | libraryG | משתנה | מחרוזת – מספר בתים לפי גודל המחרוזת | שם של התמונה של האופציות הגרפיות |
| 10 | spaceS | משתנה | מחרוזת – מספר בתים לפי גודל המחרוזת | שם תמונה של פתיחת האלגוריתם |
| 11 | spaceB | משתנה | מחרוזת – מספר בתים לפי גודל המחרוזת | שם התמונה של הרצת התוכנית |
| 12 | spaceG | משתנה | מחרוזת – מספר בתים לפי גודל המחרוזת | שם של התמונה של האופציות הגרפיות |
| 13 | templeS | משתנה | מחרוזת – מספר בתים לפי גודל המחרוזת | שם תמונה של פתיחת האלגוריתם |
| 14 | templeB | משתנה | מחרוזת – מספר בתים לפי גודל המחרוזת | שם התמונה של הרצת התוכנית |
| 15 | templeG | משתנה | מחרוזת – מספר בתים לפי גודל המחרוזת | שם של התמונה של האופציות הגרפיות |
| 16 | is\_maze | משתנה | word | אומר אם נוצר כבר מבוך או לא |
| 17 | start\_pointer | משתנה | Word | מצביע על מיקום של השם של התמונה למסכי הפתיחות |
| 18 | bg\_pointer | משתנה | Word | מצביע על מיקום של השם של התמונה למסכי הרצת האלגוריתם |
| 19 | graphics\_pointer | משתנה | Word | מצביע על מיקום של השם של התמונה למסכי אופציות הגרפיות |
| 20 | stack\_pointer | משתנה | Word | מצביע למיקום הstackC הכי עליון |
| 21 | graphics\_args | משתנה | Word – מערך של 12 | מערך של מצביעים למשתנים שרציתי להעביר לפונקציית הגרפיקה |
| 22 | run\_args | משתנה | Word – מערך של 13 | מערך של מצביעים למשתנים שרציתי להעביר לפונקציית הרצת האלגוריתם |
| 23 | maze\_start | משתנה | word | מחזיק את מקום התחלת המבוך |
| 24 | maze\_end | משתנה | word | מחזיק את מקום סיום המבוך |
| 25 | current\_cell | משתנה | word | מחזיק את הבלוק העכשווי שהאלגוריתם נמצא בו |
| 26 | Xn | משתנה | word | מחזיק את המספר הרנדומלי של בחירת השכן להמשיך לסלול את המבוך 0-3 |
| 27 | stackC | משתנה | Word – מערך של 400 | זה מערך של stack מיוחד שיצרתי בשביל אלגוריתם המבוך |
| 28 | neighbors\_list | משתנה | Word – מערך של 4 | מערך של ארבעת השכנים והאם השחקן יכל לעבור לשכן או לא |
| 29 | board | משתנה | Word – מערך של 800 | במערך זה נשמר כל המידע של המבוך |
| 30 | GREEN | קבוע | אין | הצבע הירוק |
| 31 | WHITE | קבוע | אין | הצבע הלבן |
| 32 | RED | קבוע | אין | הצבע האדום |
| 33 | GRAY | קבוע | אין | הצבע האפור בהיר |
| 34 | P\_XN | קבוע | אין | הגישה בstack למשתנה |
| 35 | P\_STACKC | קבוע | אין | הגישה בstack למשתנה |
| 36 | P\_NEIGHBORS\_LIST | קבוע | אין | הגישה בstack למשתנה |
| 37 | P\_STACK\_POINTER | קבוע | אין | הגישה בstack למשתנה |
| 38 | P\_CURRENT\_CELL | קבוע | אין | הגישה בstack למשתנה |

* תיאור פונקציות:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | שם פונקציה | קלט | מיקום של קלט | מה היא מבצעת | פלט | מיקום של פלט |
| 1 | drawCurrentPos | צבע ומקום לצבוע בלוק | מחסנית | מציירת בלוק על לוח המבוך לפי מקום בין 0 -399 | ריבוע למסך הגרפי | מסך |
| 2 | drawCell | אורך רוחב צבע ומקום | מחסנית | מציירת ריבוע על המסך הגרפי במקום בין 0 – סוף המסך הגרפי | ריבוע למסך הגרפי | מסך |
| 3 | drawBoard | גודל של ריבוע, מיקום, צבע | מחסנית | מציירת את הלוח | לוח של ריבועים | מסך |
| 4 | graphics | מערך של מצביעים למקומות משתנים שימושים בפונקציה | מחסנית | מעדכנת את מצביעי המסך – משנה מוד גרפי | שינוי משתנים בdata | Data |
| 5 | delay | אין | אין | לולאות – ובכך עושה דילאי קצר | אין | אין |
| 6 | drawY | הבלוק נוכחי והבלוק הבא ללכת | מחסנית | מציירת חיבור בין שתי בלוקים ציר Y | ציור חיבור | מסך |
| 7 | drawX | הבלוק נוכחי והבלוק הבא ללכת | מחסנית | מציירת חיבור בין שתי בלוקים ציר X | ציור חיבור | מסך |
| 8 | fileOffsets | רשימה לשים בה את המצביעים. מצביע ראשון | מחסנית | מוצאת את המיקום של כל מחרוזת בזיכרון ושמה אותם במערך | מערך של מצביעים | Data |
| 9 | pushC | מידע לשים במחסנית ומצביע למצביע המחסנית | מחסנית | שמה במחסנית מידע ומעדכנת את מצביע המחסנית בהתאם | מחסנית שלי מצביע למחסנית | Data |
| 10 | popC | מצביע למצביע המחסנית | מחסנית | מוציאה מהמחסנית מידע ומעדכנת את מצביע המחסנית בהתאם | מצביע למחסנית | Data |
| 11 | indexToPlace | מקום בין 0 -399 | מחסנית | ממיר את המיקום הנתון למיקום על המסך הגרפי | אוגר – דרך המחסנית | מחסנית |
| 12 | boardToData | מקום בין 0 -399 מיקום של הלוח במידע | מחסנית | ממיר את המיקום הנתון למיקום בdata segment | אוגר דרך המחסנית | מחסנית |
| 13 | dataToBoard | מקום במידע מיקום של הלוח במידע | מחסנית | ממיר את המיקום הנתון למיקום בין 0 -399 | אוגר דרך המחסנית | מחסנית |
| 14 | leftRight | מיקום עכשווי לאן רוצים ללכת | מחסנית | בודק אם אפשר ללכת לאותו מיקום ימינה או שמולאה | מחסנית | מחסנית |
| 15 | validNeighbor | מיקום הלוח במידע, מיקום נוכחי, לאן הוא רוצה ללכת | מחסנית | בודק אם אפשר ללכת לכיוון הרצוי מחזיר – את המיקום אם אפשר ואם לא אז 1- | מחסנית | מחסנית |
| 16 | findNeighbors | מערך השכנים, מיקום הלוח במידע, מיקום נוכחי | מחסנית | מעדכן את מערך השכנים – אפשר ללכת – את המיקום אי אפשר – 1- | מעדכן את מערך השכנים | Data |
| 17 | randomSeed | אין | אין | מוציא מהשעון מספר התחלתי רנדומלי – seed | מחסנית | מחסנית |
| 18 | LCG | מצביע למשתנה שמחזיק את המספר הרנדומלי הקודם | מחסנית | מעדכן את המשתנה עם המספר הרנדומלי הקודם במספר רנדומלי חדש | משתנה | Data |
| 19 | randomNeighbor | מצביעים למערך השכנים, והמשתנה של הרנדום | מחסנית | בוחר שכן רנדומלי חוקי | מיקום בזיכרון של השכן הנבחר – מחסנית | מחסנית |
| 20 | updateCellHeader | בלוק קודם, מיקום של מקום חדש | מחסנית | מעדכן את שדות הבלוק אומר שהוא בוקר כבר ומאיפה הוא הגיע | מעדכן בלוח מידע | Data |
| 21 | updateChoosenCell | מיקום בלוק קודם, מיקום של מקום חדש, מיקום בזיכרון של מצביע הstack | מחסנית | מכניס למחסנית את הבלוק החדש ומעדכן את הheader שלו בעזרת updateCellHeader | מעדכן בלוח מידע וגם במערך המחסנית | Data |
| 22 | Solution | האם נוצר כבר מבוך, תחילת המבוך, סוף המבוך, מיקום הלוח | מחסנית | מראה את פתרון המבוך | מצייר את הפתרון | מסך |
| 23 | resetMaze | האם נוצר כבר מבוך, תחילת וסיום המבוך מצביע לתמונת רקע, מיקום של מיקום עכשווי, מיקום של הלוח | מחסנית | יאפס את הלוח במסך ויאפס את כל הערכים של הלוח במידע ושל המשתנים של האלגוריתם | מאפס ערכי מידע  וגרפי | מסך  Data |
| 24 | carveMaze | מצביעים למשתני המבוך | מחסנית | יוצר את המבוך במידע ותוך כדאי מציג את תהליך היצירה וויזואלית | משנה משתנים ומצייר את המבוך | מסך  Data |
| 25 | endCrave | מצביע למשתנה דגל האם יש מבוך, תחילת וסיום המבוך | מחסנית | מצייר את שתי נקודות ההתחלה והסוף של המבוך ומעדכן את המשתנה לכך שיש מבוך | משנה משתנים ומצייר גרפי | מסך  Data |
| 26 | Run | מערך של מצביעים למשתנים הדרושים לפונקציה | מחסנית | נותנת למשתמש את האופציה לבוחר איזה פונקציה להפעיל- בניית המבוך פתירה שלו, אתחול שלו, הגדרות סודיות | איו | אין |
| 27 | secretSettings | מצביעים ל- האם יש מבוך, תמונת רקעת מיקום עכשווי, לוח, תמונה , התחלה וסוף המבוך | מחסנית | נותן למשתמש לבחור איזה בלוק – התחלה או סוף לשנות | משנה ערכים של משתנים  ומציג במסך הגרפי | מסך  Data |
| 28 | chooseCell | צבע, נקודה להזיז ונקודה להשאיר | מחסנית | מזיז לפי wsad את הבלוק שנתון בגרפי ובdata | משנה משתנים ומצייר למסך | Data  מסך |
| 29 | updateMovement | צבע, נקודה להשאיר, מיקום עכשווי מקום ללכת אליו | מחסנית | מעדכן וויזואלית ומחזיר את המיקום החדש | מיקום חדש למחסנית ומסך מצייר | מחסנית  מסך |
| 30 | touch\_edges | כיוון, נקודה להשאיר, המיקום הבא | מחסנית | בודק אם המשתמש יכול ללכת לבלוק מסוים – קירות ונקודה שנשארת במקום | מחזיר 0 אם אי אפשר ללכת  1 אם אפשר | מחסנית |

**מקורות חיצוניים:**

<https://data.cyber.org.il/assembly/assembly_book.pdf>

קישור לGitHub:

<https://github.com/programmer1024r/Maze_Generator>