**פרויקט באסמבלי – מבוך תלת מימדי**

שם הפרויקט: המבוך

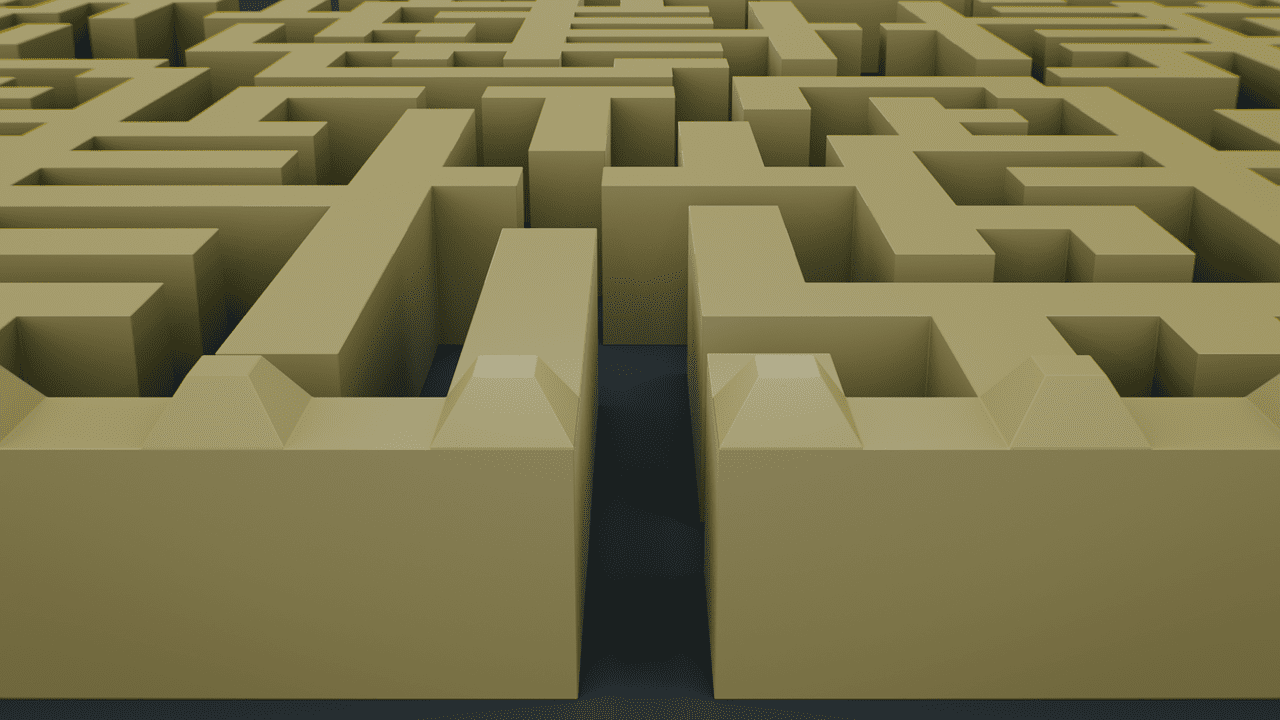
שם המתכנת: אילן קיסלבסקי

תעודת זהות: 214447591

מורה מלווה: יוסי זהבי

כיתה: י'4

בית ספר: תיכון ע"ש חיים הרצוג כפר סבא



תוכן עניינים

[מבוא 3](#_Toc10407712)

[נושא העבודה 4](#_Toc10407713)

[אופן ההפעלה 4](#_Toc10407714)

[גרסאות המערכת 4](#_Toc10407715)

[תיעוד והסבר הפתרון 5](#_Toc10407716)

[תרשים זרימה 8](#_Toc10407717)

[רשימת פעולות 9](#_Toc10407718)

[דוגמאות הרצה 11](#_Toc10407719)

[סיכום אישי 12](#_Toc10407720)

# מבוא

שם העבודה: המבוך

שם הקובץ: Maze3D.asm

קבצים נלווים: run.bat

סביבת הפיתוח: Notepad++

סביבת הרצה: DOSBox

סביבת עבודה: Turbo Assembler, Turbo Linker, Turbo Debugger

**כיצד יוצרים קובץ הרצה?**

את הקובץ מריצים בתוכנית tasm, המתרגמת את פקודות האסמבלי לשפת מכונה. כתוצאה מכך, נוצר קובץ חדש עם אותו שם אבל עם סיומת .obj את הקובץ הזה נעביר בתוכנית tlink, תכנית המקשרת קבצים ויוצרת קובץ הרצה, אותו שם סיומת exe. את קובץ ההרצה אפשר להריץ ואפשר לדבג באמצעות TD.

# נושא העבודה

בפרוייקט שלי זזים בתוך מבוך שנראה תלת מימדי בעזרת המקלדת. על מנת לנצח, יש להגיע מנקודת ההתחלה אל נקודת הסיום, אשר אותה יראו כקיר עם מסגרת אדומה. חלק חשוב בעבודה היה לחשב את מאפייני משתמש (כיוון, X,Y במבוך, עומק ווקטור מימין ומשאל(יש או אין פנייה)) על מנת לצייר נכון את המבוך.

# אופן ההפעלה

בתחילת המשחק יוצג מסך שמציג את הוראות המשחק שממנו מתחילים את המשחק בעזרת לחיצה על מקש מקלדת. בתחילת המשחק יצוייר המבוך בנקודת ההתחלה, המשתמש ילחץ על החצים בהתאם לכיוון הרצוי והמבוך ישתנה בהתאם לנקודת המבט חדשה. במהלך המשחק ניתן להדפיס מפה דו מימדית של המבוך בעזרת המקש M. כמו כן, יהיה ניתן להדפיס את מאפייני המשתמש (כיוון, X,Y במבוך) בעזרת המקש P. כאשר המשתמש יגיע אל נקודת הסיום תוצג הודעת נצחון "Victory Royale".

# גרסאות המערכת

בגרסה הנוכחית ניתן להתקדם ע"י החץ העליון והחץ התחתון. כמו כן ניתן לפנות בעזרת החץ הימני והחץ השמאלי ומיד לאחר הפניה רואים את המבוך מנקודת המבט החדשה. בגרסה הבאה חשבתי להוסיף צליל של צעדים וניגון שיר בעת הניצחון. בנוסף, בעת הפניה יוצג מבט של מסלול התנועה מנקודת המוצא עד לנקודת המבט החדשה.

# תיעוד והסבר הפתרון

בתחילת המשחק מוצג מסך הוראות המשחק .

לאחר מכן נכנסים ללולאה הראשית של המשחק כאשר רק לחיצה על מקש היציאה (esc) או ניצחון ע"י הגעה אל המקום הניצחון יוציא מן הלולאה והתוכנית תפסק.

**בתוך הלולאה הראשית:**

1. חישוב פרטי המשתמש על מנת לצייר בהתאם את המבוך.
2. ציור המבוך.
3. קליטת קלט מהמשתמש.
   1. אם המקש שנלחץ לא תואם לאף אחת מהאפשריות יש חזרה אל קליטת הקלט.
4. ביצוע הפעולה בהתאם לקלט.
   1. אם הפעולה אינה יכולה לקרות (למשל, לעבור דרך קירות) יש חזרה אל קליטת הקלט.
5. בדיקה: האם יש ניצחון?
   1. השמה במשתנה בוליאני ifwin בהתאם לתוצאת הבדיקה.
6. ניקוי המסך על מנת לצייר בפעם הבאה את המבוך.
7. בדיקה: האם צריך להדפיס את מאפייני המשתמש או לצייר את מפת המבוך?
   1. פועל בהתאם, אם צריך לצייר – מציירים, אחרת חוזרים להתחלה.

**פעולת ציור המבוך למסך -Drawmaze :**

1. חישוב מאפייני המלבן המלא על מנת לציירו.
2. לפי העומק, ציור מלבן מלא ב-X,Y המתאימים שנשמרו במערכים ArrX,ArrY.
3. ציור קווים אלכסוניים מפינה עליונה של המסך לפינה תחתונה בצבע המבוך.
4. בדיקה: האם רואים את קיר הניצחון?
   1. אם כן - ציור מסגרת אדומה מסביב למלבן המלא.
5. **לולאה באורך העומק שבתוכה:**
   1. ציור קו אנכי בצד שמאל בX,Y- השמורים במערכים הנ"ל, מתחילת המסך ועד לתחילת המלבן.
   2. בדיקה: האם יש פנייה בצד שמאל? (לפי המערך VectorLeft)
      1. אם כן, ציור קו אופקי מהקו הקודם ועד לקו הנוכחי, לאחר מכן ציור מלבן בצבע שחור על מנת למחוק את האלכסון. את שתי הפעולות הרשומות ב-4.2.1 עושים גם על סוף הקו (אותו X, ולY מוסיפים את אורך הקו). אם לא, קופצים לבדיקה בצד ימין.
   3. ציור קו אנכי בצד ימין ב320-X,Y השמורים במערכים הנ"ל, מסיום המלבן ועד לסוף המסך.
   4. בדיקה: האם יש פנייה בצד ימין? (לפי המערך VectorRight)
      1. אם כן, ציור קו אופקי מהקו הנוכחי ועד לקו הבא, לאחר מכן ציור מלבן בצבע שחור על מנת למחוק את האלכסון. את שתי הפעולות הרשומות ב-4.4.1 עושים גם על סוף הקו (אותו X, ולY מוסיפים את אורך הקו.

**פעולת חישוב העומק –** **CalcDeep :**

הפעולה היא מעין סוג של פעולה עוטפת, הקוראת לפעולה המחשבת את העומק באמת לפי הכיוון של המשתמש.

בתוך הפעולות המחשבות את העומק:

1. חישוב האינדקס בזיכרון בהתאם למיקום המשתמש במבוך.
2. ביצוע לולאה באורך המתאים לכיוון שבתוכה:

(0 - ה Yבמבוך פעמים, 1 - גודל המבוך פחות הY במבוך פעמים , 2 - גודל המבוך פחות הX במבוך פעמים, 3 – הX במבוך פעמים).

1. ביצוע test בין הערך במבוך לבין הקיר המתאים.

( 0 - הקיר העליון (2), 1 - הקיר הימני (1), 2 - הקיר התחתון (8), 3 - הקיר השמאלי (4)).

1. אם התוצאה היא 0, אין קיר, מעלים את העומק באחד וחוזרים ללולאה הבאה. אם התוצאה היא 1, יש קיר, לא מעלים את העומק ויוצאים מהלולאה.

**פעולה הבודקת אם רואים את קיר הנצחון – Showwin:**

פעולה זו בודקת אם ממקום המשתמש רואים את קיר הנצחון לפי Xman,Yman ו- Xwin,Ywin.

1. בדיקה מהו כיוון המשתמש
2. בודקת לפי הכיוון אם Yman=Ywin או Xman=Xwin (0,2 X, 1,3 Y)
3. הוספה/החסרת העומק מהמשתנה (המשתנה שנבדק קודם) , (0,3 החסרה, 1,2 הוספה).
4. בדיקה אם הסכום/ההפרש שווה לX/Y הנצחון.
5. השמה של 0/1 ב showwin1 בהתאם.

# 

# תרשים זרימה

# רשימת פעולות

DrawMap - פעולה המציירת את המבוך כתמונה דו מימדית, מקבלת את המערך השומר את המבוך.

- DrawDir0, DrawDir1, DrawDir2, DrawDir3פעולות המציירות חץ, לפי כיוון המשתמש, במשבצת בו נמצא במפה הדו מימדית, מקבלת את כיוון משתמש במשתנה DirMan ואת הX,Y של תחילת המשבצת המייצגת את המשתמש.

ClearScreen – פעולה המנקה את כל המסך.

PrintProp - פעולה המדפיסה את הX,Y של המשתמש במבוך ואת כיוונו, מקבלת את המשתנים DirMan,Xman,Yman.

Drawmaze- פעולה המציירת את המבוך באופן תלת מימדי, מקבלת את המערך השומר את המבוך.

CalcVector- פעולה הקוראת לפי כיוון המשתמש לפעולה אחרת המחשבת אם יש קיר או אין קיר מימין ושמאל עד העומק, מקבלת את המשתנה DirMan.

CalcVector\_dir0, CalcVector\_dir1, CalcVector\_dir2, CalcVector\_dir3 – פעולות המחשבות אם יש קיר או אין קיר מימין ושמאל עד העומק בהינתן כיוון המשתמש והעומק, מקבלת את המשתנים DirMan, Xman, Yman, Deep. ושמה 0/1 בהתאם לאם יש קיר או אין במערכים VectorLeft, VectorRight.

CalcDeep - פעולה הקוראת לפי כיוון המשתמש לפעולה אחרת המחשבת את העומק, מקבלת את המשתנה DirMan.

Deep\_Dir0, Deep\_Dir1, Deep\_Dir2, Deep\_Dir3 – פעולות המחשבות את עומק המשתמש בהינתן כיוונו, מקבלת את המשתנים DirMan, Xman, Yman. ומחזירה במשתנה Deep את העומק המחושב.

MoveInmaze – פעולה הבודקת את כיוון התזוזה (קדימה או אחורה) ואת כיוון המשתמש (לפי המשתנה DirMan) ומשנה את הX,Y של המשתמש במבוך בהתאם, הפעולה גם בודקת שהמשתמש לא יוכל לעבור דרך קירות.

Goback\_dir0, Goback\_dir1, Goback\_dir2, Goback\_dir3 – פעולות הבודקות אם המשתמש עלול לעבור קיר בתזוזה אחורה, מקבלת את המשתנים DirMan, Xman, Yman. קוראים לפעולה רק אם נלחץ חץ למטה (לזוז אחורה).

ChangeLook – פעולה המשנה את כיוון המשתמש בהתאם לקלט מהחצים.

FullRect – פעולה המציירת מלבן מלא, מקבלת X,Y של נקודה למעלה בשמאל, אורך רוחב וצבע.

EmptyRect - פעולה המציירת מלבן ריק, מקבלת X,Y של נקודה למעלה בשמאל, אורך רוחב וצבע.

DrawHorizontalLine – פעולה המציירת קו אופקי, מקבלת X,Y של נקודה הכי עליונה, אורך הקו וצבע.

DrawVerticalLine – פעולה המציירת קו אנכי, מקבלת X,Y של נקודה הכי שמאלית, אורך הקו וצבע.

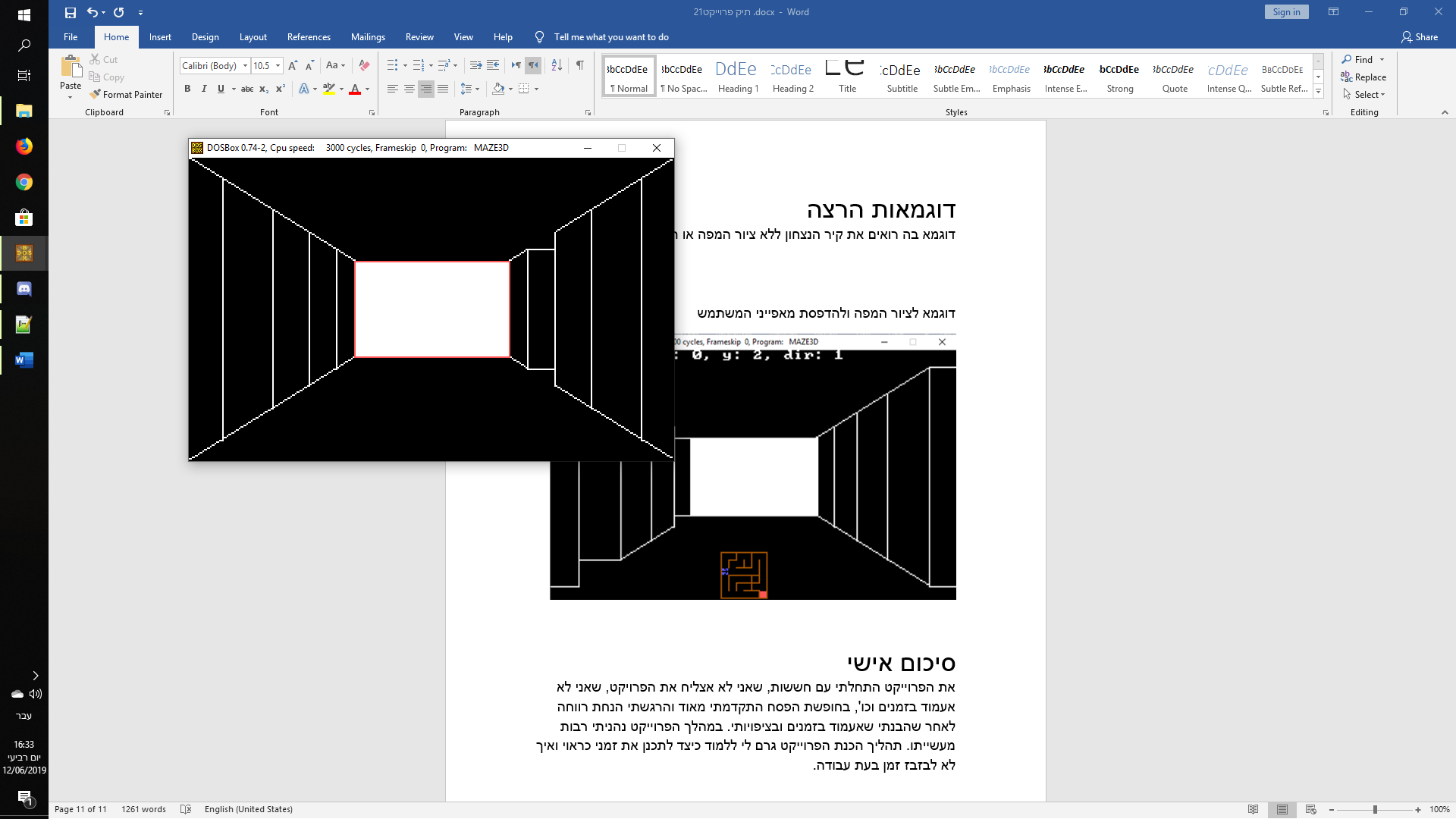
DrawLine2D – פעולה המציירת קו אלכסוני, ומקבלת שתי נקודות התחלה סיום של הקו.

PIXEL – פעולה המקבלת נקודה וצובעת אותה.

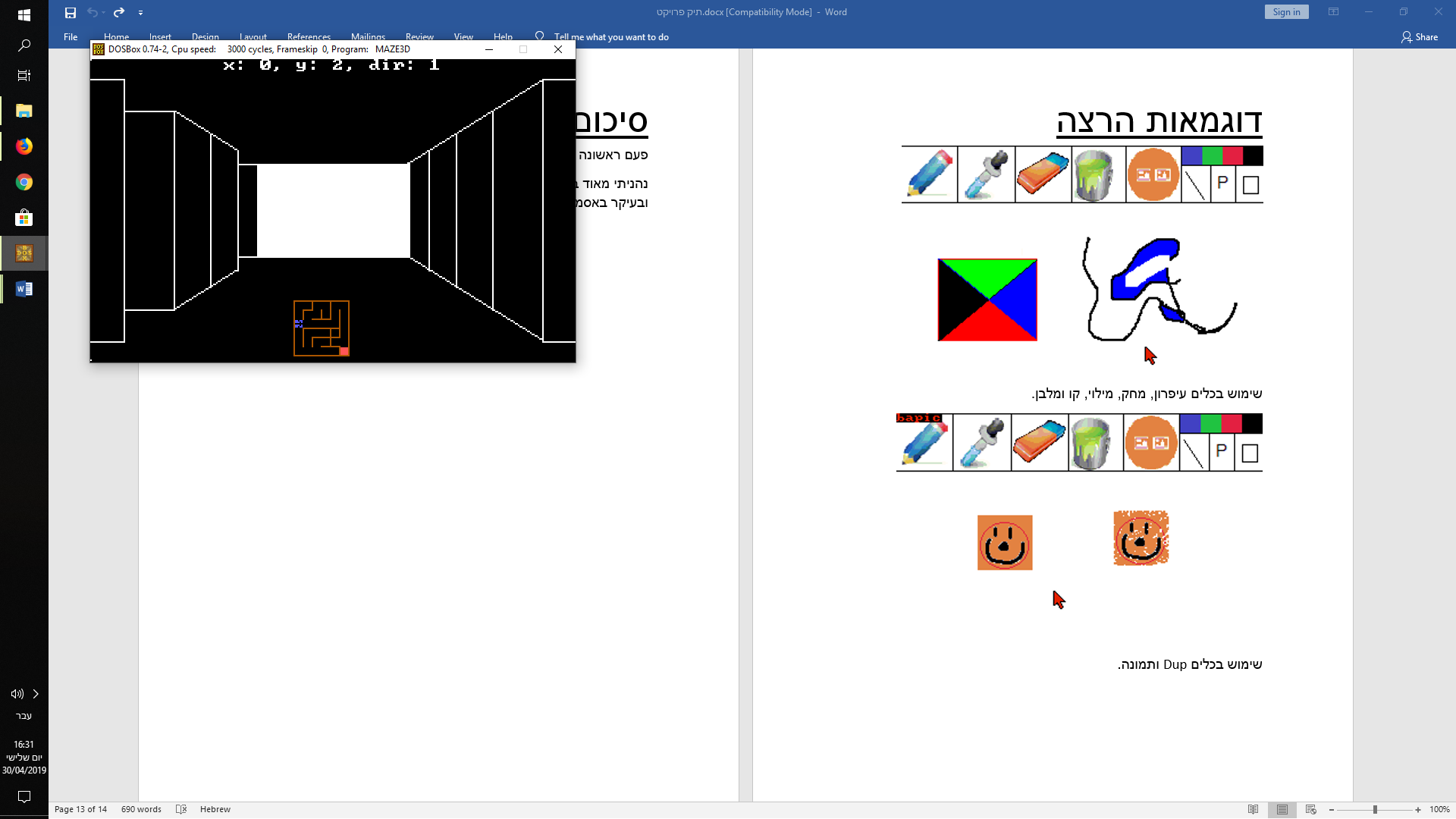
Showwin – פעולה הבודקת עם רואים את קיר הנצחון, מקבלת במשתנים את DirMan, Xman, Yman, Deepועושה חישוב מתאים על מנת לבדוק, השמה של 0/1 ב showwin1 בהתאם לתוצאת הבדיקה.

# דוגמאות הרצה

דוגמא בה רואים את קיר הנצחון ללא ציור המפה או הדפסת מאפייני המשתמש



דוגמא לציור המפה ולהדפסת מאפייני המשתמש



# סיכום אישי

את הפרוייקט התחלתי עם חששות, שאני לא אצליח את הפרויקט, שאני לא אעמוד בזמנים וכו', בחופשת הפסח התקדמתי מאוד והרגשתי הנחת רווחה לאחר שהבנתי שאעמוד בזמנים ובציפויותי. במהלך הפרוייקט נהניתי רבות מעשייתו. תהליך הכנת הפרוייקט גרם לי ללמוד כיצד לתכנן את זמני כראוי ואיך לא לבזבז זמן בעת עבודה.