**פרויקט סיום – מיקרו-מעבדים ושפת אסמבלי:**

הלוגיקה בתוכנית:

טענו את המסך הגרפי של ה-DOSBOX שגודלו 320x200. "חילקנו" את המסך ל"לקוביות" בגודל 8x8 פיקסלים כך שגודל המבוך הוא: 40x25. למעשה נתייחס למסך כאל מסך בגודל 40x25 ונשתמש בפונקציות מתאימות כדי להמיר את מיקומי ה"קוביות" למיקומי פיקסלים במסך הגרפי.

יש מערך בשם Maze שמכיל 1000 איברים שכל אחד מהם מייצג פיקסל. אנחנו שומרים איפה נמצא השחקן במבוך. אם הוא זז ימינה אנחנו בודקים את המערך במקום +1. אם הוא זז שמאלה אנחנו בודקים את המערך במקום -1. אם השחקן זז למעלה אחננו בודקים את המערך במקום -40 (כלומר שורה אחת למעלה) ואם אם יורד למטה אנחנו בודקים את המערך במקום +40. (כלומר המקום הנוכחי במערך שהשחקן נמצא בו וממנו זזים מספר איברים תלוי לאיזה כיוון זזים). בודקים אם הערך של המבוך שאותו מקום שווה ל-1 ואם כן אז אפשר ואז משנים את המשתנים ששומרים את מיקום השחקן בהתאם וטוענים את הקובייה שהשחקן היה בה (מבחינת הצבעים) ואז שומרים את הקובייה מבחינת הצבעים שהשחקן הולך להיטען אליה (שהתמונה של השחקן עומדת להיטען אליה) וטוענים את השחקן לפי הכיוון שזזנו.

אם השחקן טוען מבוך דיפולטיבי (בתחילת המשחק או לחיצה על 1) אז נטען מבוך מוכן מראש שהוא אותו מבוך כמו בדוגמה שהיה במבוך לדוגמה בקובץ הפרויקט שבמודל.

אם השחקן טוען מבוך רנדומלי נבחר באופן אקראי (בתחילת התוכנית אנחנו לוקחים מספר שהוא מכפלת השעות והדקות וכל פעם שאנחנו טוענים מבוך אנחנו מכפילים את המספר הזה פי 5 ומוסיפים 1 כדי "להכין" מספר רנדומלי חדש. בטעינה מבוך רנדומלי אנחנו לוקחים את המספר ומחלקים אותו פי 7 והשארית של החלוקה קובעת איזה מבוך ייטען.

המבוכים הרנדומלים למעשה הם מבוכים מוכנים מראש שבכל פעם אחננו בוחרים מבוך בצורה רנדומלית.

SaveCell, LoadCell - לפני שאנחנו ממקמים את השחקן במבוך. אנחנו שומרים במערך בגודל 64 איברים את הצבעים של הפיקסלים של אותו קוביה שאנחנו שמים את השחקן עליה. ואז טוענים את השחקן במצב של "חץ שפונה ימינה". לאחד מכן אם אחננו זזים לאיזשהו קובייה אנחנו קודם טוענים בחזרה את הקובייה של השחקן היה בה. ואז שומרים את הקובייה שהשחקן עומד להיות מודפס בה ואחרי ששמרנו את הקוביה במערך אחננו מדפיסים את השחקן באותה קובייה.

SetXY – מחשבים את הפיקסל השמאלי-העליון לפי ה-X וה-Y של הקובייה.

ResetVariables - כשמפעילים את התוכנית (או לוחצים 1 או 2 לטעינה מחודשת של מבוך), המשתנים של ה-DataSegment מתאפסים.

ResetMaze – מאתחל את ערכי המערך של המבוך ל-1 (שמשמעותו קובייה כחולה).

DefineGraphicScreen – לאחר מכן מוגדר המסך הגרפי ונטען.

ColorCube – חילקנו את המסך ל-"קוביות" בגודל 8 על 8 ואנחנו נותנים לפונקציה את הפיקסל השמאלי-העליון שממנו מתחילים לצבוע את הקוביה ואת הצבע של הפיקסלים שנצבעים.

ColorScreen – צובע את המסך בכחול.

ClearBlack – מעלימים את הפיקסלים השחורים מסביב לטקסט של ה-START וה-END. וצובעים בכחול.

MatrixPosition – מחשבים איזה מיקום במערך אחננו צריכים לבדוק לפי המיקום של השחקן.

CreateMatrix - יוצר את המטריצה. רץ על המיקומים במסך ובודק קובייה קובייה. אם הקובייה כחולה יהיה במערך ערך 1 ואם הקובייה צהובה הערך במערך ישתנה ל-2. ואז יהיה לנו מערך שמכיל את הערכים של המבוך מבחינת אם זה דרך שניתן ללכת בה או שזה קיר.

Print\_Start, Print\_End – מדפיס את המחרוזות במיקומים שצריך להדפיס אותם.

SetPlayer – ממקם את השחקן אחרי טעינת המבוך ומשנה את משתני המיקום שלו בהתאם.

DrawRow – מקבל מיקום, X ו-Y וגודל השורה ואם זה שורה (אופקית) או טור (אנכי) ואז צובע את הקוביות בהתאם, בעזרת פונקציה זו נצבע את המסך ונכין אותו. ואז נשתמש בפונקציה CreateMatrix שתיארנו קודם כדי להכין מהמבוך הזה מערך שמתאר אותו.

DefineDefaultScreen, DefineRandomScreen1, DefineRandomScreen2 – צובע את השורות והטורים של המבוך בצהוב בעזרת סאברוטינת DrawRow.

DrawCarUp, DrawCarRight, DrawCarLeft, DrawCarDown – מקבל מיקום של החץ שמתאר את השחקן. וצובע את הדמות של השחקן פיקסל-פיקסל לפי "סוג" השחקן שצריך לצייר שתלוי לאיזה כיוון השחקן פה.

WaitingKeyPress – כאשר התוכנית מחכה לקלט מהמקלדת:

אם השחקן הגיע לקצה המבוך, (האות D של END התוכנית תסתיים).

אם לא, התוכנית תבדוק אם השחקן לחץ על ESC ואם כן היא תצא מהתוכנית.

1 – טעינת מבוך רגיל.

2 – טעינת מבוך רנדומלי.

48h – לחיצה למעלה.

50h – לחיצה למטה.

4Bh – לחיצה שמאלה.

4Dh – לחיצה ימינה.

אם לא הסאברוטינה של המתנה ללחיצת מקש תופעל שוב.

שימוש בתוכנית:

התוכנית מציגה את המבוך כפי שיש בקובץ הפרויקט.

בלחיצה על '1' הוא טוען מחדש את המבוך הדיפולטיבי.

בלחיצה על '2' הוא טוען מבוך רנדומלי.

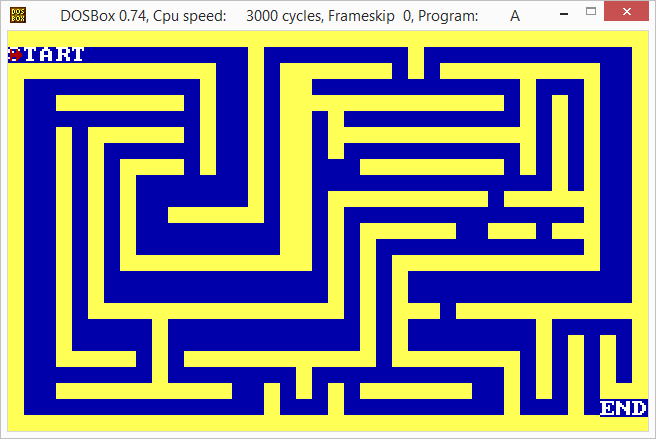
יש מצשתנים שמכילים את המיקום של השחקן.

את המבוך עצמו מבחינת אם זה כחול או צהוב.

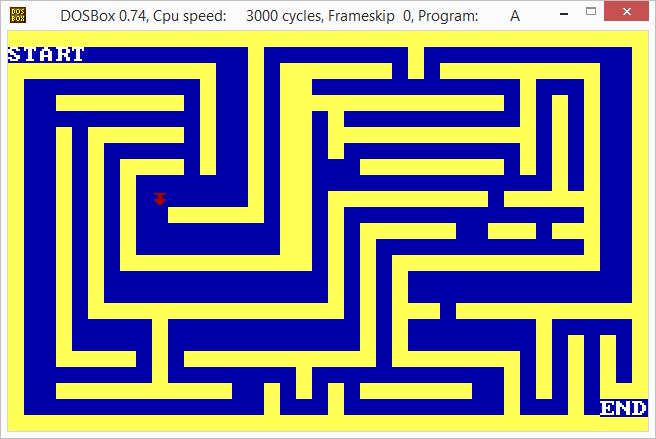
כשלוחצים על מקש אז הוא בודק במערך של המבוך האם המקום שאני עומד להגיע אליו כחול או לא. אם כחול הוא מזיז את השחקן ואם לא אז הוא לא מזיז.

התוכנתי נגמרת כאשר וחצים על ESC או כשמגיעים ל-END.

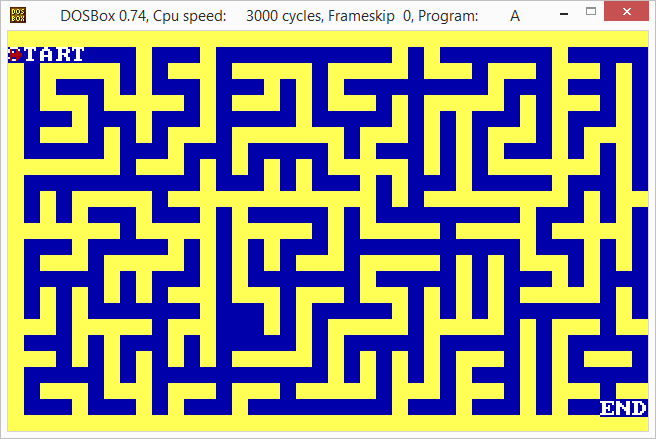
כשמפעילים את התוכנית:



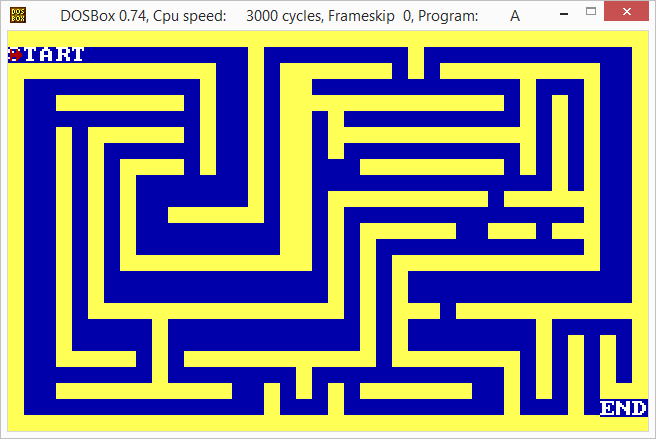
וניתן להזיז את השחקן רק דרך הריבועים הכחולים:



לחיצה על לחצן 2 מפעיל מבוך רנדומלי:



לחיצה שוב על 1 מפעילה את המבוך הדיפולטיבי (המבוך לדוגמה):



כשמגיעים ל-END (כשהשחקן על האות D), התוכנית מסתיימת:

