



תיק פרויקט – אולטימייט איקס עיגול – Ultimate Tic Tac Toe

מגיש: גיא גור-אריה

ת"ז: 212499347

בית ספר: "איש שלום"

מקצוע: סייבר – אסמבלר

שמות המנחים: אופיר שביט, רחל רדומסקי

תיאור הפרויקט:

בחרתי לתכנת את המשחק אולטימייט איקס עיגול, או סופר איקס עיגול (Ultimate Tic Tac Toe/Super Tic Tac Toe), שהוא גרסא מתוחכמת יותר ומסובכת יותר של משחק האיקס עיגול המקורי (Tic Tac Toe) שהומצא לפני הרבה מאוד שנים עוד במצרים העתיקה, בוא צריכים לקבל מצב של 3 אריחים (איקס או עיגול) ברצף, אנכית אופקית או באלכסון על מנת לנצח.

הגרסא הזאת של איקס עיגול שונה לחלוטין מצורת החשיבה שיש באיקס עיגול המקורי, והיא הרבה יותר מעניין ומאתגרת.

לא ידוע באיזה שנה הומצא האולטימייט איקס עיגול, אך סביר להניח בשנת 2000+.

חוקי המשחק:

- בכל פעם אתה יכול לשים איקס/עיגול רק בשטח המסומן בירוק, על ידי לחיצה על העכבר במיקום שבוא אתה רוצה לשים איקס/עיגול.
- בכל פעם שאתה שם איקס/עיגול, זה שולח את השחקן השני למשבצת אחרת, כלומר קובע את השטח הירוק שבוא מותר לשים, באופן הבא:

אם שחקן שם איקס באמצע של טבלה קטנה, לא משנה איפה הטבלה הקטנה, זה שולח את השחקן לשים באמצע של הטבלה הגדולה, כלומר איפשהו בטבלה הקטנה האמצעית. אותו דבר לגבי בפינות וכו'.
- אם הגיעו למצב שבו טבלה קטנה אחת נכבשה, כל פעם ששחקן שולח את השני לשם, יש לשני אפשרות לשים איפה שהוא רוצה, והשטח הירוק מכסה את כל הטבלאות שוב, לכן מבחינה אסטרטגית, יש להימנע מלשלוח את השחקן היריב לשם. טבלה נחשבת "כבושה" גם אם התרחש שם מצב של "תיקו", אך לא האיקס ולא העיגול כבשו אותה.
- אם למישהו יש 3 שטחים כבושים (איקס/עיגול גדולים) ברצף, בצורה אופקית, אנכית, או באלכסון, הוא המנצח והמשחק נגמר. כמובן גם יכול להימצא מצב של "תיקו", אך מצב זה נדיר מאוד, וזה מה שמבדיל את המשחק הזה מהאיקס עיגול הרגיל.

הוראות הפעלה:

יש להריץ את הקובץ `game.asm`, שצריך להימצא באותה תיקיה של `codseg1.asm – codeseg8.asm`, ו-`dataseg1.asm – dataseg4.asm`, הכוללת גם את התמונות של מסך הבית, דף ההוראות, ולוח המשחק.

כאשר מריצים את הקובץ מגיעים למסך הבית, בוא ישנם 3 כפתורים:

"Play" – שיתחיל את המשחק,

"Rules" – שיפתח את דף ההוראות שבוא מוסברות ההוראות בקצרה,

ו-"Exit" – שמסומן בתור חץ שמצביע על דלת, אם ילחצו עליו התכנית תיסגר.

ניתן ללחוץ על הכפתורים רק באמצעות העכבר, ולחיצה על החצים או על המקשים במקלדת, לא תעשה כלום.

אם לוחצים על "Play", המשחק מתחיל, ובדף של המשחק ישנם עוד 2 כפתורים:

"Restart" – שיתחיל את המשחק מחדש, כלומר ינקה את הלוח. ניתן ללחוץ עליו בכל זמן נתון, אך אם השחקנים רוצים משחק הוגן הם לא ילחצו עליו עד שאחד ינצח.

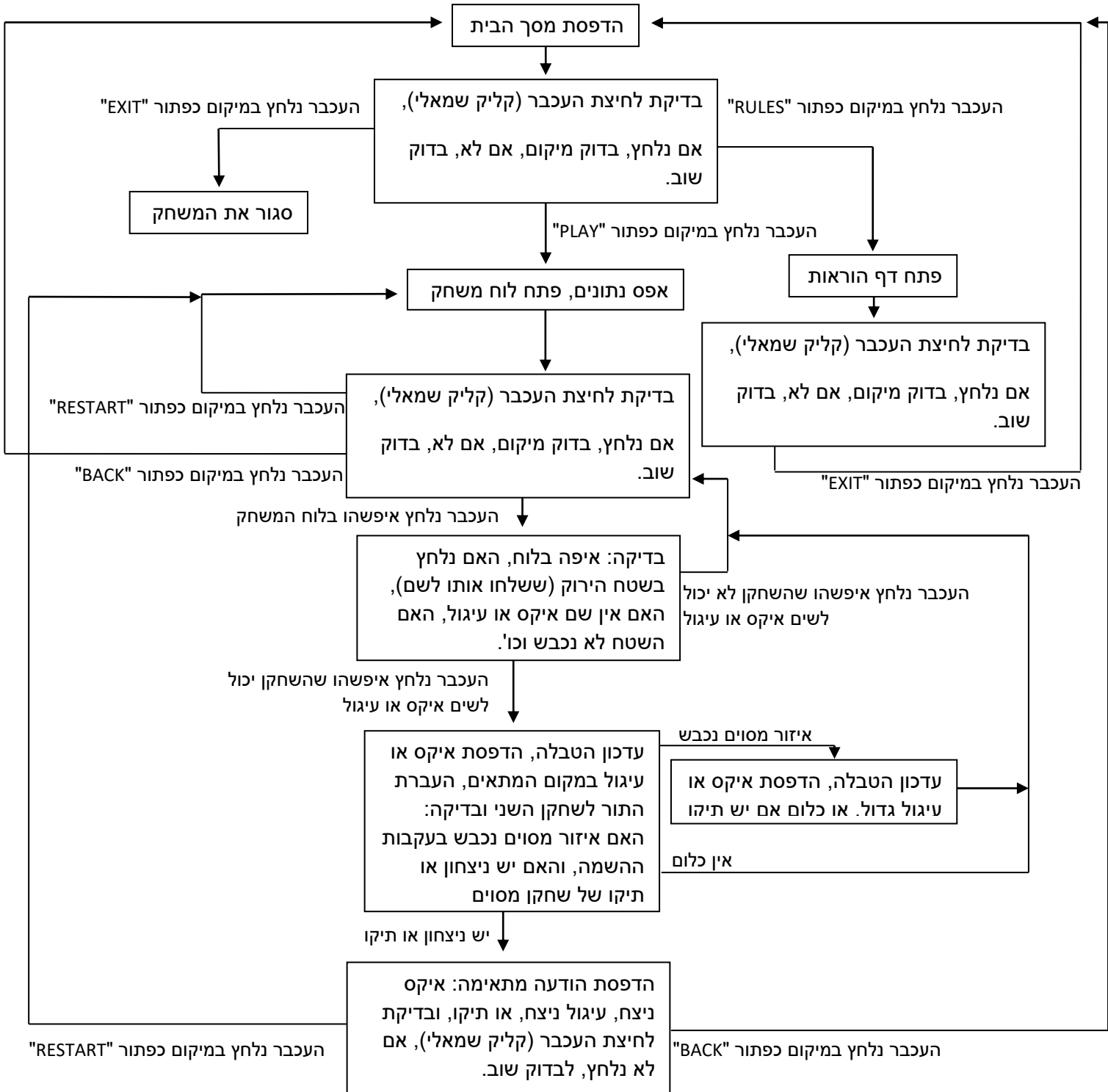
ו-"Back" – שפשוט חוזר למסך הבית.

בדף ההוראות יש רק מקש שחוזר לדף הבית.

אם אחד השחקנים ניצח, יש ללחוץ על כפתור ה"Restart" או לחזור לדף הבית.

לצד לוח המשחק (הטבלה) ישנו עוד לוח, המסמן תור מי זה (איקס/עיגול), והאם איקס ניצח, עיגול ניצח, או יש "תיקו" בסיום משחק.

תיאור הפתרון



ב"איפוס נתונים" הכוונה היא לאיפה יש איקס ועיגול, כך שיהיה ניתן ללחוץ בכל מקום אחרי שהטבלה מתאפסת, ה"נתונים" זה בעצם מערך שמדמה מערך דו מימדי שנקרא TableXO, שבו 90 משתנים, כלומר 10 לכל טבלה:

1-9 למיקום שבו נלחץ איקס או עיגול בטבלה (1 לאיקס, 2 לעיגול, 0 אין שם כלום), ומיקום 0 שיכול להיות 4 דברים: 1 אם איקס כבש את הטבלה, 2 אם עיגול כבש, 3 אם יש שם תיקו, ו-0 אם לא נכבש בכלל.

כמו כן יש עוד מערך חד מימדי שבו 10 משתנים, הנקרא CubesThatWon, והוא לצורך עזר בעיקר, ומתייחס לטבלה הגדולה:

1-9 זה לטבלה קטנה שנכבשה (1-4 כמו מקודם), ומיקום 0 במערך הוא המנצח.

גרפיקה:

הגרפיקה של המשחק לא כל כך מסובכת, השתמשתי רק ב-3 תמונות (מסך פתיחה, הוראות, ולוח משחק), ועדכנתי דברים על התמונה של הלוח כדי להגיש מי ניצח או האם יש תיקו.

בכדי לצייר את האיקס והעיגול, יש לי 2 מערכים של 13 על 13 פיקסלים (ככה יצא לי, זה לא משהו חשוב), אחד מצייר איקס קטן, ואחד שמצייר עיגול קטן, ועוד 2 מערכים גדולים של 51 על 51 פיקסלים, אחד מצייר איקס גדול, ואחד מצייר עיגול גדול.

דוגמא: המערך של העיגול הקטן:

circle db 16,16,16,16,6 ,6 ,6 ,6 ,16,16,16,16

db 16,16,6 ,6 ,6 ,6 ,6 ,6 ,16,16

db 16,6 ,6 ,6 ,16,16,16,16,6 ,6 ,6 ,16

db 16,6 ,6 ,16,16,16,16,16,16,6 ,6 ,16

db 6 ,6 ,16,16,16,16,16,16,16,16,6 ,6

db 6 ,6 ,16,16,16,16,16,16,16,16,6 ,6

db 6 ,6 ,16,16,16,16,16,16,16,16,6 ,6

db 6 ,6 ,16,16,16,16,16,16,16,16,6 ,6

db 6 ,6 ,16,16,16,16,16,16,16,16,6 ,6

db 16,6 ,6 ,16,16,16,16,16,16,6 ,6 ,16

db 16,6 ,6 ,6 ,16,16,16,16,6 ,6 ,6 ,16

db 16,16,6 ,6 ,6 ,6 ,6 ,6 ,6 ,6 ,16,16

db 16,16,16,16,6 ,6 ,6 ,6 ,6 ,16,16,16,16

אני משתמש בפרוצדורה מסוימת הקוראת את האיברים במערך הנתון
ומדפיסה אותם פיקסל אחרי פיקסל במיקום הנתון לה.

הפרוצדורות: (לרוב הפרוצדורות אין קלט ופלט, לכן ציינתי קלט או פלט רק כאשר יש)

שם הפרוצדורה:	מה היא עושה:	היכן היא מצויה:
drawCircle	קולטת X ו-Y, ומדפיסה שם עיגול קטן	codeseg2.asm
drawCross	קולטת X ו-Y, ומדפיסה שם עיגול קטן	codeseg2.asm
drawInSmallCube	הפרוצדורה הכי מסובכת בתכנית. אורכה כ-300 שורות. היא קולטת איפה העכבר לוחץ (X ו-Y), מעדכנת את הקורדינטות לפינה של כל קוביה קטנה בלוח, לפי המיקום שבו העכבר לחץ, בודקת אם זה בתוך הלוח, בודקת איזו קוביה זו, האם יש שם כבר איקס או עיגול, האם מותר לו ללחוץ בשטח הזה, וכמו כן מעדכנת את נתוני הטבלה אך בלי להדפיס כלום. את הX וה-Y שהיא מדפיסה, משתמשים אחרי זה הפרוצדורות שמדפיסות עיגול קטן ואיקס קטן.	codeseg3.asm
SmallCubeFixProc	משנה את כל מה שירוק בטבלה לשחור, כלומר מאפסת את השטח הירוק, על מנת שפרוצדורה אחרת תקבע את השטח הירוק החדש.	codeseg4.asm
SmallCubeProc	קובעת את השטח הירוק החדש, כולל אם שלחו מישהו לשטח כבוש אז היא צובעת את הכל בירוק.	codeseg4.asm

codeseg5.asm	גם 3 הפרוצדורות הבאות הן ד"י ארוכות אך לא כל כך מסובכות. הפרוצדורה הזאת בודקת בכל טבלה קטנה האם איקס או עיגול ניצח בצורה אופקית, ומעדכן את נתוני הטבלה אם כן.	CheckIfWonColumn
codeseg5.asm	בודקת בכל טבלה קטנה האם איקס או עיגול ניצח בצורה אנכית, ומעדכן את נתוני הטבלה אם כן.	CheckIfWonRow
codeseg5.asm	בודקת בכל טבלה קטנה האם איקס או עיגול ניצח בצורה אלכסונית, ומעדכן את נתוני הטבלה אם כן. בעצם חילקתי את הבדיקה של כבישת שטח ל-3 פרוצדורות.	CheckIfWonDiagonal
codeseg5.asm	בודקת אם בכל טבלה קטנה יש תיקו. מקרה זה נדיר מאוד אך יכול לקרות ולכן היא קיימת. בעצם היא בודקת אם בכל שורה טור או אלכסון אין ניצחון. משתמשת בפרוצדורת עזר "MidCheck".	CheckIfDraw
codeseg5.asm	קולטת 3 נתונים ובודקת אם כולם שווים אחד לשני. עוזרת בפרוצדורה "CheckIfDraw" בכך שמחזירה 1 אם 3 המשתנים אינם שווים.	MidCheck
codeseg6.asm	כל הפרוצדורות עכשיו הן גרסא ערוכה של הפרוצדורות הקודמות, כך שיבדקו את הטבלה הקטנה או יציירו בה, במקום בטבלאות הקטנות. הפרוצדורה הזאת	DrawBigCircle

	<p>בודקת אם יש ניצחון על פי נתוני הטבלה שעודכנו מהפרוצדורות הקודמות ומדפיסה עיגול גדול אם עיגול ניצח.</p>	
codeseg6.asm	<p>רשמתי "Cube" במקום "Cross", אך זה לא משבש את התוכנית בשום אופן ואני מעדיף להשאיר את זה ככה. הפרוצדורה הזאת בודקת אם יש ניצחון על פי נתוני הטבלה שעודכנו מהפרוצדורות הקודמות ומדפיסה איקס גדול אם איקס ניצח.</p>	DrawBigCube
codeseg7.asm	<p>בדיוק כמו "CheckIfWonColumn", רק מתיחס לטבלה הגדולה. הפרוצדורה הזאת בודקת אם יש ניצחון סופי בצורה אופקית של איקס או עיגול בטבלה הגדולה, ומעדכנת את הנתונים של הטבלה.</p>	CheckIfFinalyWonColumn
codeseg7.asm	<p>הפרוצדורה הזאת בודקת אם יש ניצחון סופי בצורה אנכית של איקס או עיגול בטבלה הגדולה, ומעדכנת את הנתונים של הטבלה.</p>	CheckIfFinalyWonRow
codeseg7.asm	<p>הפרוצדורה הזאת בודקת אם יש ניצחון סופי בצורה אלכסונית של איקס או עיגול בטבלה הגדולה, ומעדכנת את הנתונים של הטבלה.</p>	CheckIfFinalyWonDiagnal
codeseg7.asm	<p>הפרוצדורה בודקת אם המשחק הסתיים בתיקו ומעדכנת את הנתונים</p>	CheckIfFinalyDraw

	של הטבלה. מקרה זה גם נדיר, אך בכל זאת אפשרי. משתמשת בפרוצדורת עזר "MidFinalCheck".	
codeseg7.asm	קולטת 3 נתונים ובודקת אם כולם שווים אחד לשני. עוזרת בפרוצדורה "CheckIfFinallyDraw" בכך שמחזירה 1 אם 3 המשתנים אינם שווים.	MidFinalCheck
codeseg8.asm	codeseg8.asm מיועד רק לבדיקת מיקום כפתורים. הפרוצדורה הזאת בודקת אם נלחץ כפתור ה RESTART בלוח המשחק ומאתחלת את הלוח אם כן.	CheckRestartButton
codeseg8.asm	הפרוצדורה הזאת בודקת אם נלחץ כפתור ה QUIT/BACK בלוח המשחק ומחזירה למסך הבית אם כן.	CheckQuitButton
codeseg8.asm	הפרוצדורה הזאת בודקת אם נלחץ כפתור ה PLAY במסך הפתיחה ומאתחלת את נתוני הלוח ומדפיסה את לוח המשחק אם כן.	CheckStartButton
codeseg8.asm	הפרוצדורה הזאת בודקת אם נלחץ כפתור ה RULES במסך הפתיחה ופותחת את דף ההוראות אם כן.	CheckRulesButton
codeseg8.asm	הפרוצדורה הזאת בודקת אם נלחץ כפתור ה EXIT במסך הפתיחה ויוצאת מהמשחק אם כן.	CheckExitButton
codeseg8.asm	הפרוצדורה הזאת בודקת אם נלחץ כפתור ה BACK בדף ההוראות וחוזרת למסך הבית אם	CheckBacktoMenuButton

	כן.	
codeseg8.asm	קוראת לכל הפרוצדורות שבודקות כפתורים בלוח המשחק (אתחול או חזור)	CheckButton
codeseg8.asm	קוראת לכל הפרוצדורות שבודקות את הכפתורים המסך הבית (התחל, חוקים או צא)	CheckMenuButtons
game.asm	רושמת תור מי זה (איקס או עיגול)	WriteTurn
game.asm	רושמת מי ניצח (איקס, עיגול או תיקו)	WriteWon
game.asm	מאתחלת את נתוני הטבלה	Reboot
game.asm	קוראת לפרוצדורות שבודקות כיבוש של טבלה קטנה (בצורה אנכית, אופקית, אלכסון או תיקו).	CheckIfWonAll
game.asm	קוראת לפרוצדורות שבודקות ניצחון או תיקו סופי בטבלה הגדולה (בצורה אנכית, אופקית, אלכסון או תיקו).	CheckIfFinallyWonAll

כמו כן, ישנן את הפרוצדורות שמדפיסות תמונה, אך זה פשוט העתקה מהספר ולא אני כתבתי ולכן אני לא חושב שיש צורך לציין אותן.

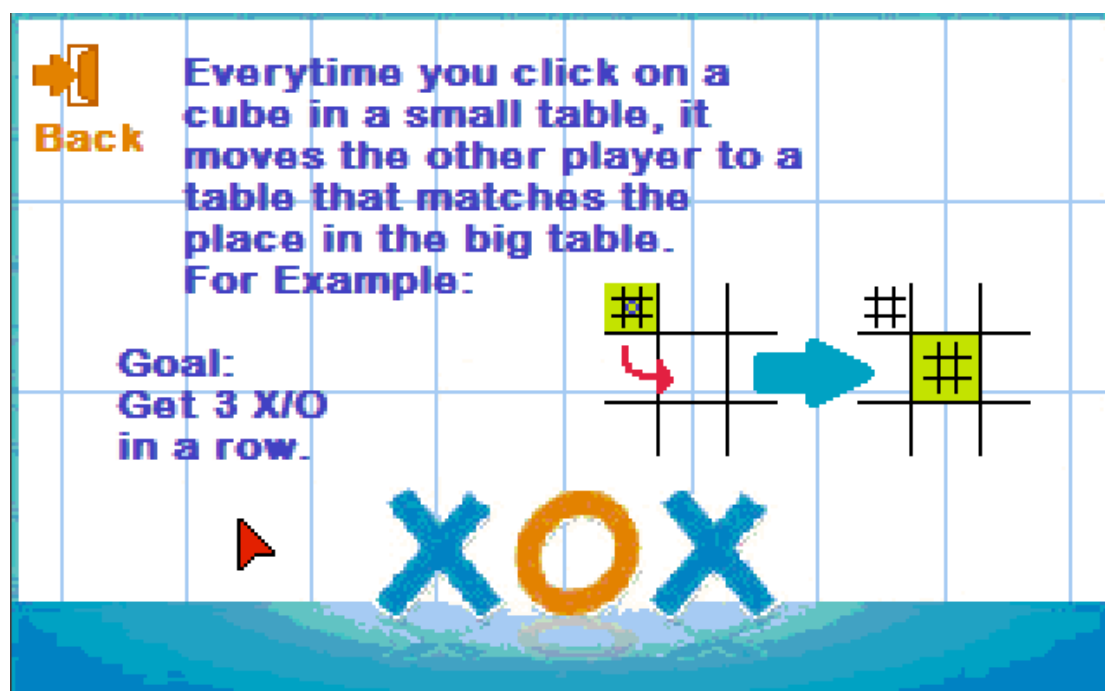
צילומי מסך מהמשחק:

מסך פתיחה: (הוספתי עכבר משל עצמי, אין בזה שום צורך, זה רק לצורך הנאה)

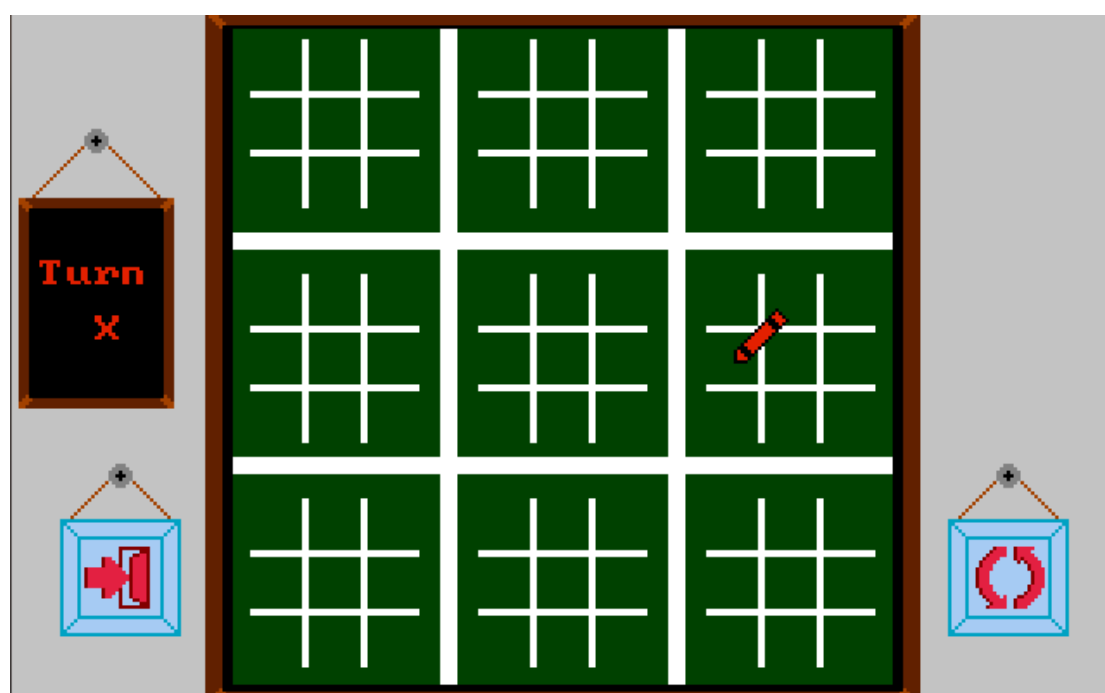
את הכיתוב של "TIC TAC TOE" והרקע מצאתי באינטרנט, אך את הכפתורים ושאר הכיתוב עשיתי בעצמי.



דף הוראות: (מקוצר מאוד, ניתן ללמוד על המשחק ולפתח אסטרטגיות ככל שמשחקים יותר)

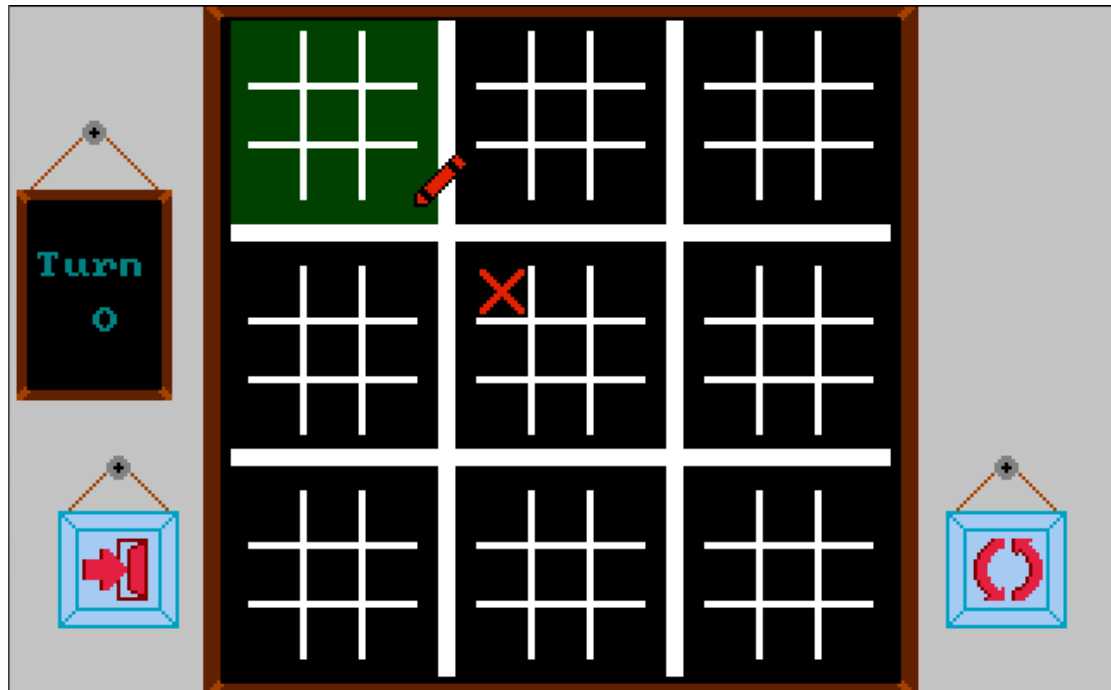


לוח המשחק: (ריק) גם פה הוספתי עכבר שונה קצת)

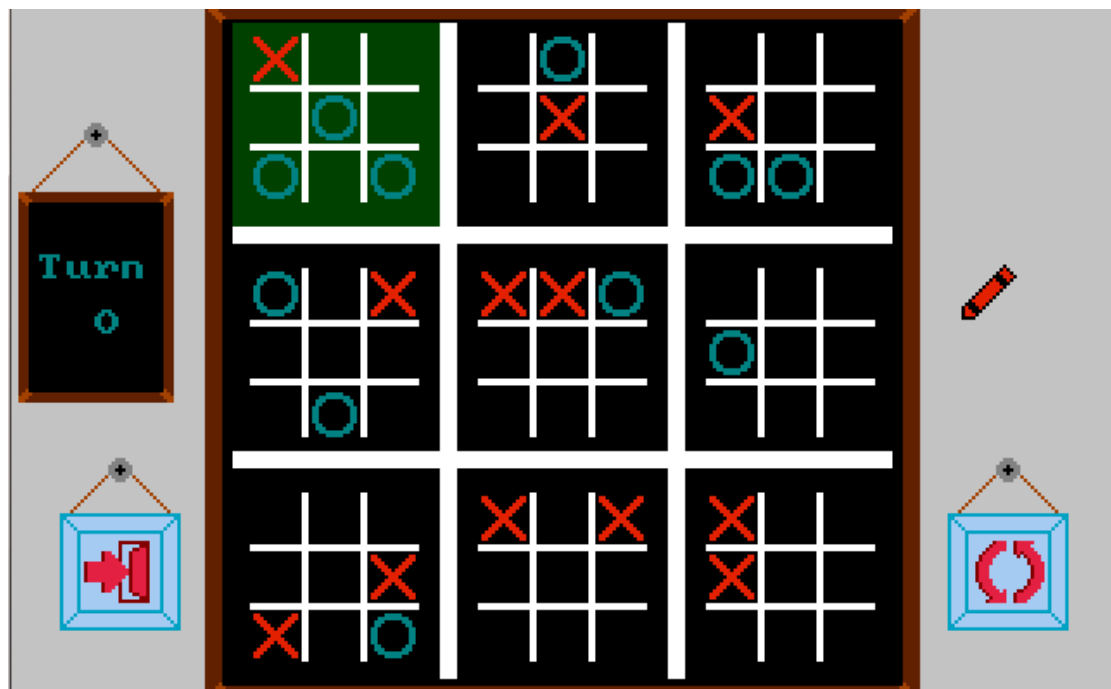


לוח המשחק אחרי תור אחד: (לראות מה משתנה)

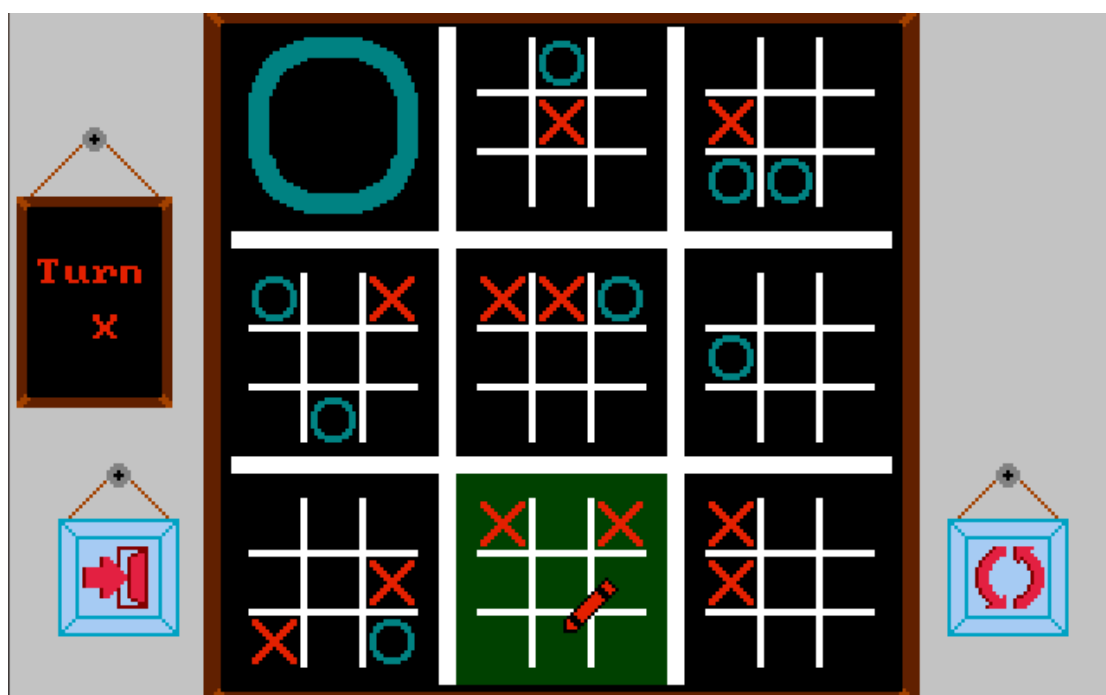
כמו שאפשר לראות, בלוח בצד רשום שתור עיגול עכשיו, והשטח הירוק התעדכן לאחר ההשמה של איקס.



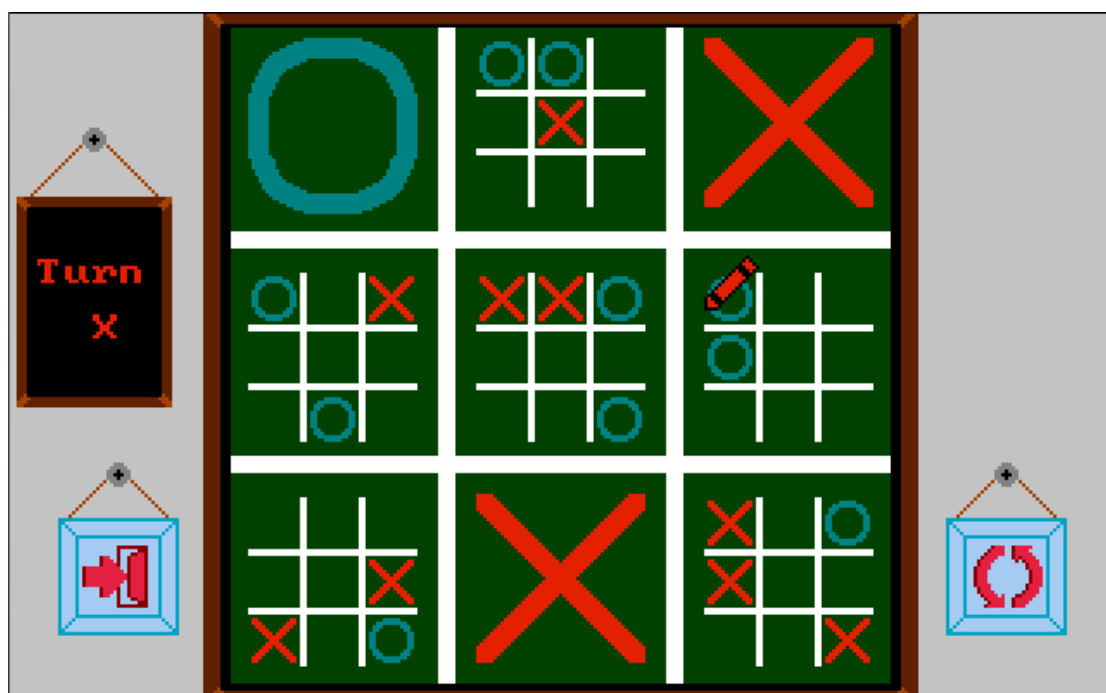
לוח המשחק לאחר כמה תורות:



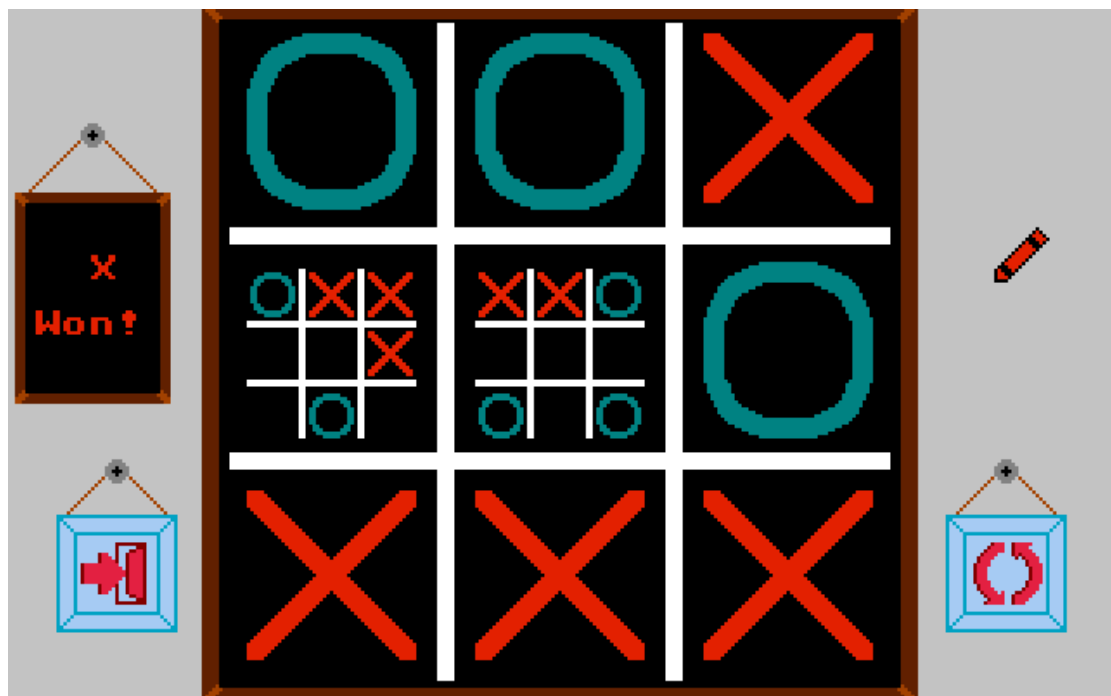
לוח המשחק לאחר ששטח נכבש:



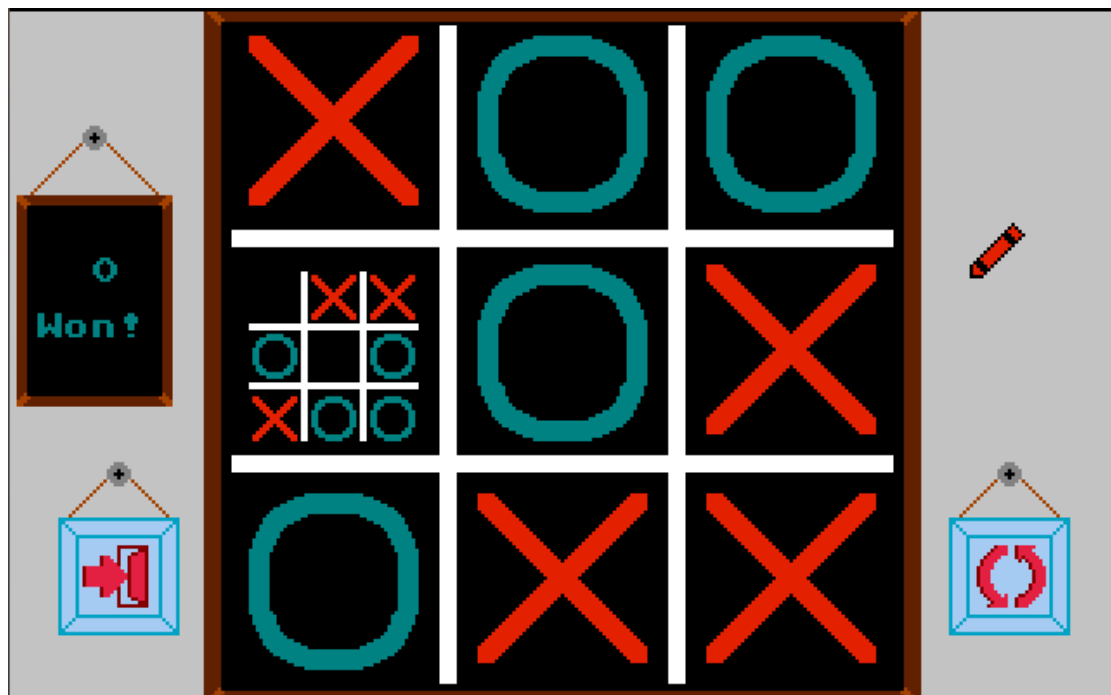
לוח המשחק לאחר שמספר שטחים נכבשו, ושלחו שחקן למשבצת שנכבשה (השטח הירוק מכסה את כל הלוח שוב):



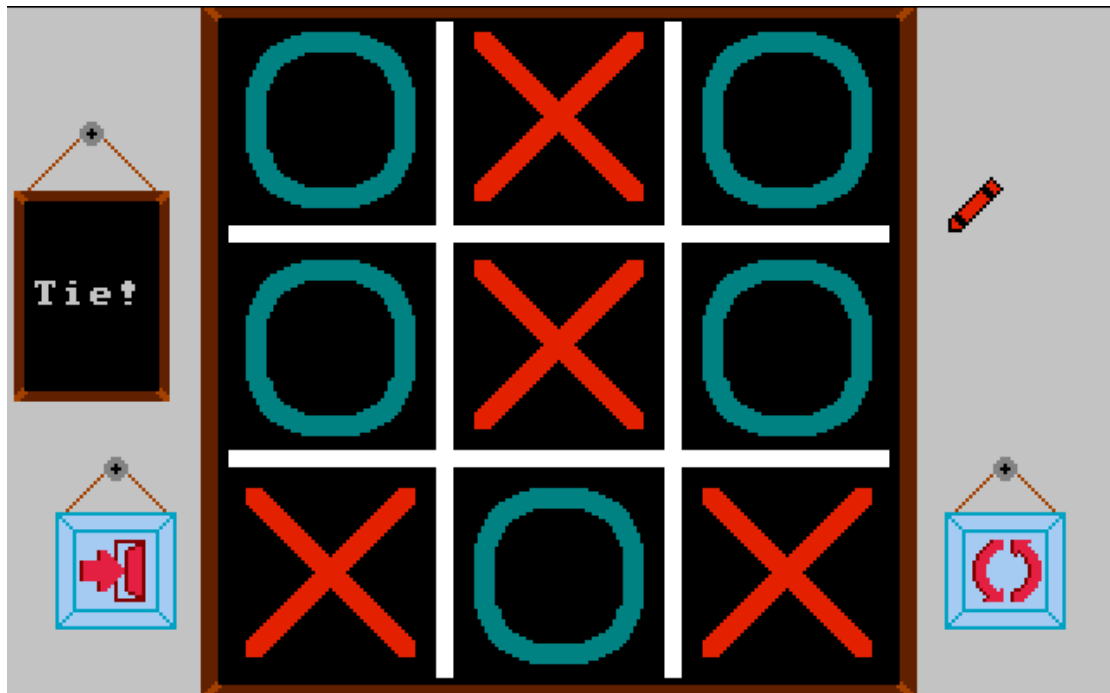
לוח המשחק כאשר איקס ניצח:



לוח המשחק כאשר עיגול ניצח (הפעם עם יותר שטחים כבושים):



לוח המשחק לאחר תיקו (עם הודעה מתאימה בלוח ליד):



אני בחרתי בפרויקט הזה מכמה סיבות:

1. אני אוהב איקס עיגול ואני גם טוב בזה, ומאז שגיליתי את הגרסא הזאת אני משחק הרבה בה, ואני חושב שיש בזה הרבה יותר אסטרטגיה וחשיבה מסתם איקס עיגול רגיל.
2. אני וחבר שלי תכננו לתכנת את המשחק הזה בשפת תכנות אחרת לפני הרבה זמן, ואך פעם לא הייתה לנו הזדמנות, והוא כבר פרש מזה, וכשהייתה לי אפשרות לעשות משחק ישר חשבתי על האפשרות הזאת.
3. רציתי אתגר. ידעתי שהפרויקט הזה יהיה יותר מסובך משאר הדברים שיכולתי לעשות, אבל דעתי שאני אעמוד בזה וזה השתלם בסוף. לעשות איקס עיגול רגיל כל אחד יכול, זה קצת יותר אתגר בשבילי.