**פרויקט איקס-עיגול – אנגולר**

צרו אתר באנגולר, המאפשר למשתמש לשחק במשחק איקס-עיגול נגד המחשב.

1. **חוקים:**

* קיימת מטריצה בגודל 3X3 המאפשרת בכל תא להציג X או O.
* ברגע שיש שלושה איקסים בשורה או בטור או באלכסון – האיקס מנצח.
* ברגע שיש שלושה עיגולים בשורה או בטור או באלכסון – העיגול מנצח.
* המשתמש תמיד מתחיל והוא תמיד איקס.
* כאשר המשתמש מקליק עם העכבר על תא במטריצה – התא הזה מציג תמונה של איקס.
* לאחר Delay של שנייה אחת, המחשב מוצא תא מתאים ושם בו תמונה של עיגול.
* אם המשתמש לוחץ על תא שכבר יש בו תמונה – לא קורה כלום.
* אם המשתמש לוחץ מחוץ למטריצה – לא קורה כלום.
* אם המשתמש מנצח – לאחר Delay של 3 שניות – מוצגת ההודעה: "ניצחת את המחשב!".
* אם המחשב מנצח – לאחר Delay של 3 שניות – מוצגת ההודעה: "המחשב ניצח!".
* אם יש תיקו (רק אם אין יותר תאים פנויים) – לאחר Delay של 3 שניות – מוצגת ההודעה: "תיקו".

1. **ה-Components שיש לבנות:**

* AppComponent – מכיל את תוכן הדף כולו – כותרת עליונה, מטריצה, כותרת תחתונה.
* HeaderComponent – מכיל את הכותרת העליונה.
* FooterComponent – מכיל את הכותרת התחתונה.
* MatrixComponent – מכיל את המטריצה.
* CellComponent – מכיל תא אחד בלבד שקיים בתוך המטריצה.
* כל Component נוסף שאתם חושבים שצריך לבנות.

1. **האחריות הויזואלית והלוגית של כל Component:**

* ה-AppComponent מכיל את ה-HeaderComponent, את ה-MatrixComponent ואת ה-FooterComponent.

כמו כן הוא הרכיב הראשון שמוצג בהרצת האתר.

* ה-HeaderComponent מכיל כותרת ראשית וכותרת משנית.
* ה-FooterComponent מכיל זכויות יוצרים.
* ה-MatrixComponent מכיל טבלה של 3X3 שבכל תא יש את ה-UI של ה-CellComponent.

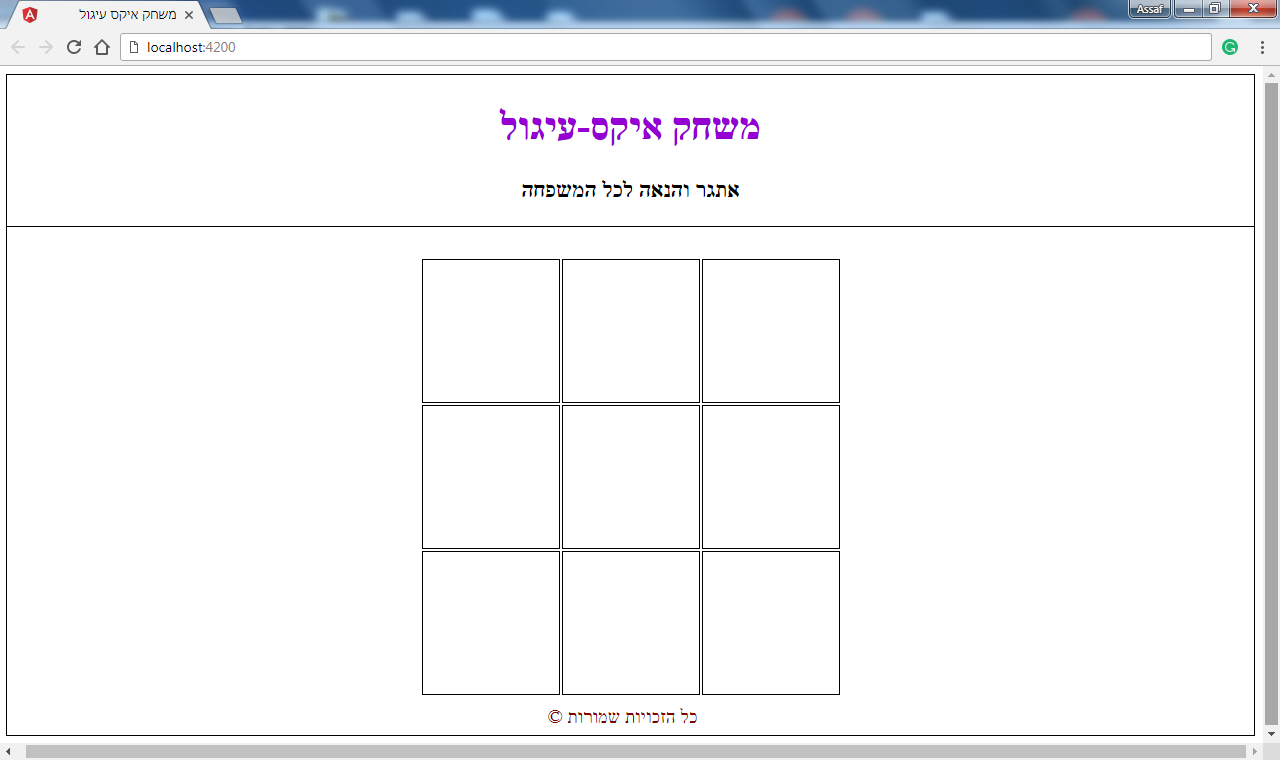
כמו כן, ה-MatrixComponent מכיל במחלקת ה-TypeScript שלו מטריצה (מערך בגודל 3, של מערכים בגודל 3) שצריכה להכיל בכל שלב במשחק ערכים מתאימים לפי הצעדים של המשתמש והמחשב. יש להחליט מה הסוג של המטריצה הזו (כלומר מה סוג הנתונים של כל תא). המלצה: enum.

ה-MatrixComponent אחראי גם על הלוגיקה של המשחק, כלומר:

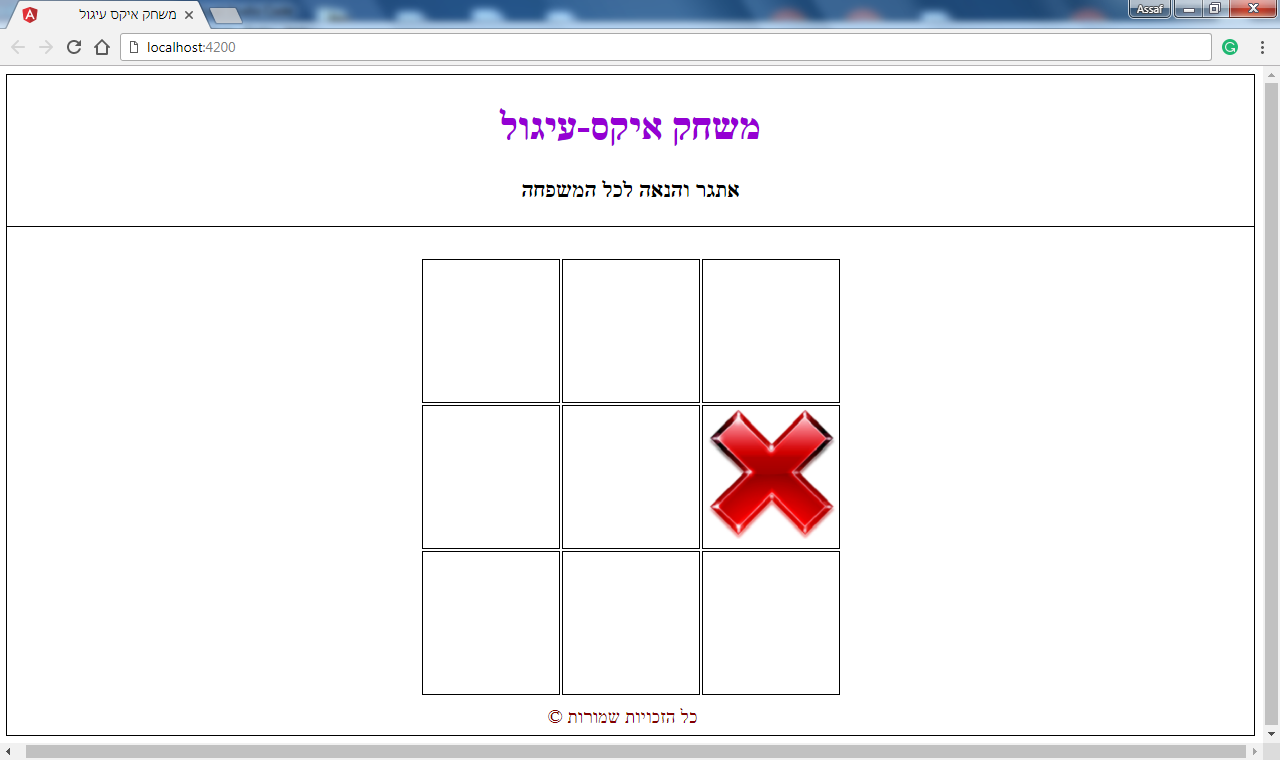
1. לנהל את המטריצה כמו שצריך מבחינת השמת הערכים הנכונים בה.
2. לקבל את צעד המשתמש (דרך העלאת ארוע מה-CellComponent אל ה-MatrixComponent).
3. לבדוק האם המשתמש ניצח, האם המחשב ניצח או האם יש תיקו.
4. למצוא את הצעד המתאים של המחשב.
5. לקבוע עבור ה-CellComponent איזו תמונה להציג.
6. להציג את ההודעה המתאימה בסיום המשחק (אם המשתמש ניצח או המחשב ניצח או שיש תיקו).
7. **הנוהל שבו המחשב מוצא את הצעד שלו:**
8. על המחשב לבדוק ראשית אם הוא יכול לנצח. אם כן – עליו לבצע זאת ולנצח.
9. אם אין אפשרות לנצח בצעד הנוכחי – על המחשב לבדוק האם הוא יכול לחסום את המשתמש מלנצח בצעד הבא, כי אם המשתמש יכול לנצח בצעד הבא, והמחשב לא יחסום אותו – אז כמובן שהמשתמש יוכל לנצח בצעד הבא. לכן אם יש מצב כזה – על המחשב לחסום את המשתמש מלנצח בצעד הבא.
10. אם אין אפשרות לנצח או לחסום את המשתמש מלנצח – על המחשב להגריל את מיקום הצעד הנוכחי, כמובן מהתאים הריקים.

להלן סדרה של תמונות הממחישות את ה-UI של המשחק:

בתחילת המשחק – כאשר האתר מורץ – מוצגת טבלה ריקה:



המשתמש תמיד מתחיל, ותמיד הוא X. להלן תמונה של צעד ראשון שהמשתמש יכול לבחור:



כעת תור המחשב. לאחר שנייה אחת, המחשב מוצא צעד מתאים. בגלל שהמחשב לא יכול לנצח כעת ובגלל שאין למחשב אפשרות לחסום את המשתמש מלנצח, המחשב מגריל מיקום כלשהו מבין המקומות הפנויים. לדוגמה:



כעת תור המשתמש, שיכול לדוגמה לבחור ב:



בשלב הזה תור המחשב, והוא בודק האם הוא יכול לנצח – והוא לא, ולכן הוא בודק האם הוא יכול לחסום את המשתמש מלנצח בשלב הבא – והוא כן! לכן המחשב בשלב הזה חייב לחסום את המשתמש מלנצח, אחרת המשתמש ינצח בשלב הבא. לפיכך, המחשב יהיה חייב לבחור בדוגמה הזו את הצעד הבא:



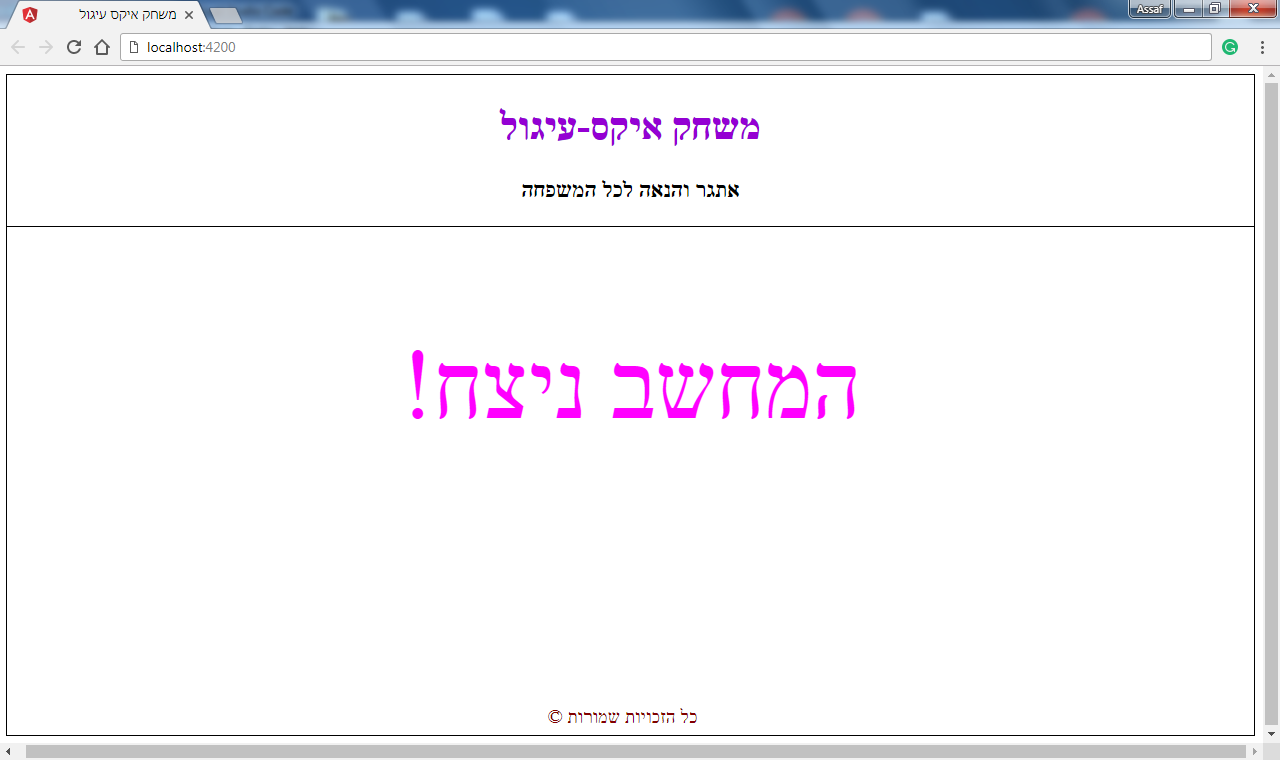
כעת תור המשתמש. נניח והמשתמש בוחר את הצעד הבא:



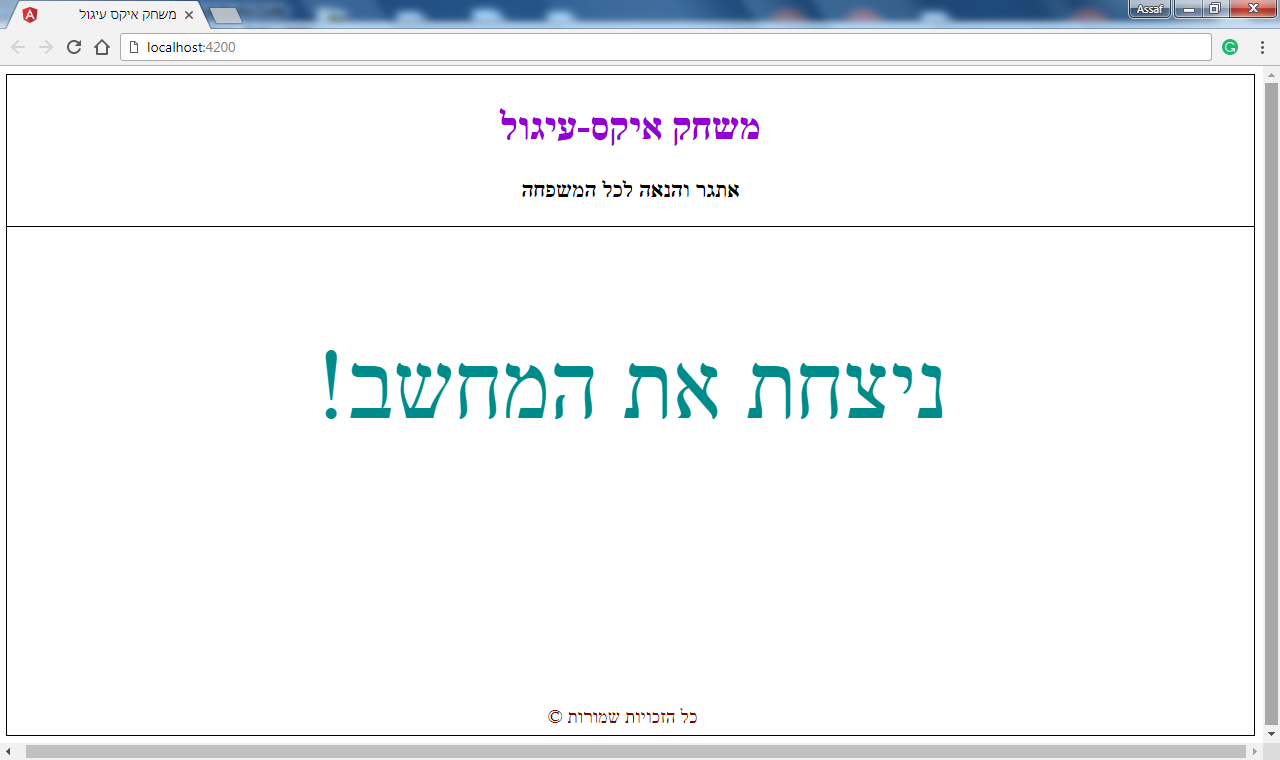
כעת תור המחשב, והוא כמו תמיד בודק ראשית אם הוא יכול לנצח – והוא אכן יכול לנצח! לכן הוא חייב לבחור במיקום שיביא לו את הניצחון, שזה המיקום הבא:



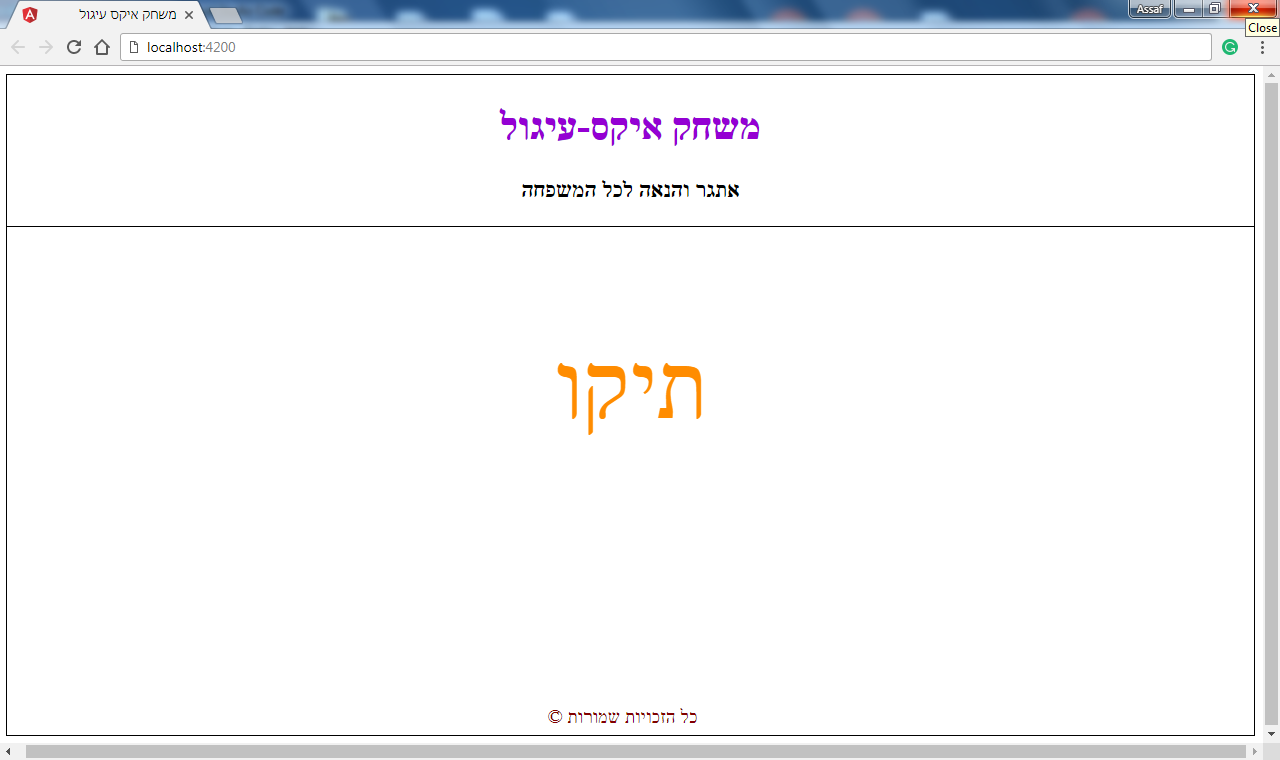
כאן, לאחר 3 שניות Delay, על האתר להציג את ההודעה הבאה:



במידה והמשתמש מנצח, יש להציג את ההודעה הבאה:



במידה ויש תיקו – יש להציג את ההודעה הבאה:



שימו לב: ניתן למצוא מקום בדף להצגת ההודעות כך שהמטריצה תמשיך להיות מוצגת ביחד עם ההודעות. כלומר לא להעלים את המטריצה בסיום המשחק כך שהמשתמש יוכל לראות את ההודעה ואת המטריצה ביחד.

יש להשתמש בתמונות הבאות עבור האיקס והעיגול:





**דגשים נוספים:**

* במידה והגיע מצב שבו חייב להיות תיקו, למרות שהמשחק לא נגמר – אין להפסיק את המשחק אלא יש לאפשר למשתמש להמשיך ולשחק (אפילו שזה לא יוביל לשום ניצחון או הפסד) ורק בסיום כל הצעדים של המשתמש והמחשב, כאשר אין יותר תאים ריקים, להציג את הודעת ה-"תיקו".
* על הפרויקט לעבור קומפילציה ולרוץ ללא שגיאות.
* אסור לפרויקט לקרוס בשום מצב.
* יש לתעד את הקוד כמו שצריך.
* יש לכתוב קוד אסתטי הכולל ריווח, הזחות (Indentations), רווח אחד בין פונקציה לפונקציה וכו'.

**בהצלחה ☺**