

תכנות מתקדם 2 – תרגיל מס' 3

בתרגיל זה נבנה ממשק Web-י למשחק שלנו. ממשק זה יכיל את הדפים הבאים:

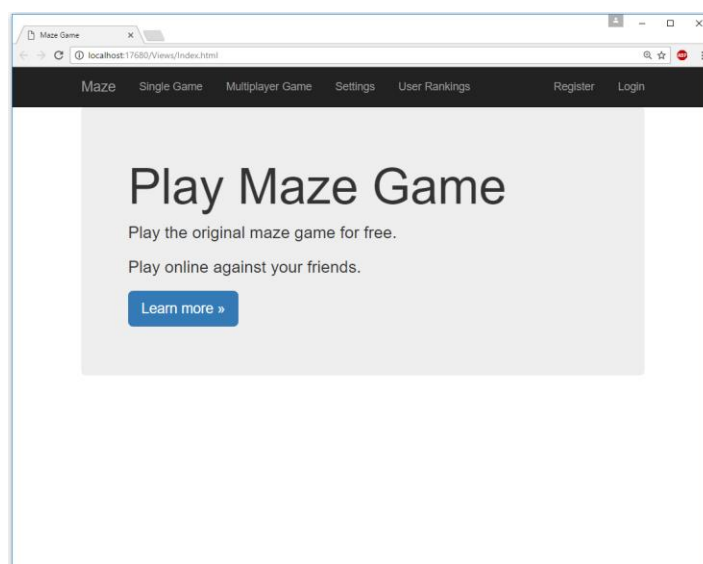
דף הבית:

בכותרת הדף יופיע תפריט עם האפשרויות הבאות:

- משחק יחיד
- משחק נגד שחקן יריב
- הגדרות
- הצגת טבלת שיאים
- כניסה / רישום משתמש חדש

תפריט זה יוצג באופן קבוע בכל דפי האתר (איך ניתן לבצע זאת מבלי לשכפל את קוד התפריט?)
 כל הדפים של האתר יהיו נגישים לכלל המבקרים פרט לדף של משחק נגד שחקן יריב, המחייב רישום של המשתמש לאתר.

דף הבית יכול להיראות למשל כך:



דף הרשמה:

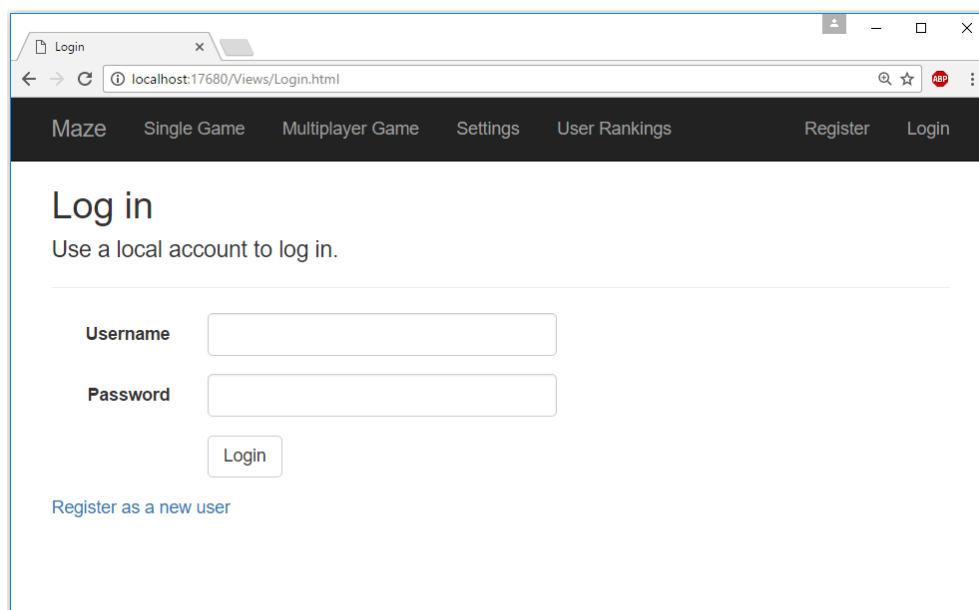
בדף יופיע טופס שביקש מהמשתמש להזין את הפרטים הבאים:

- שם משתמש
- סיסמא
- אימות סיסמא
- Email

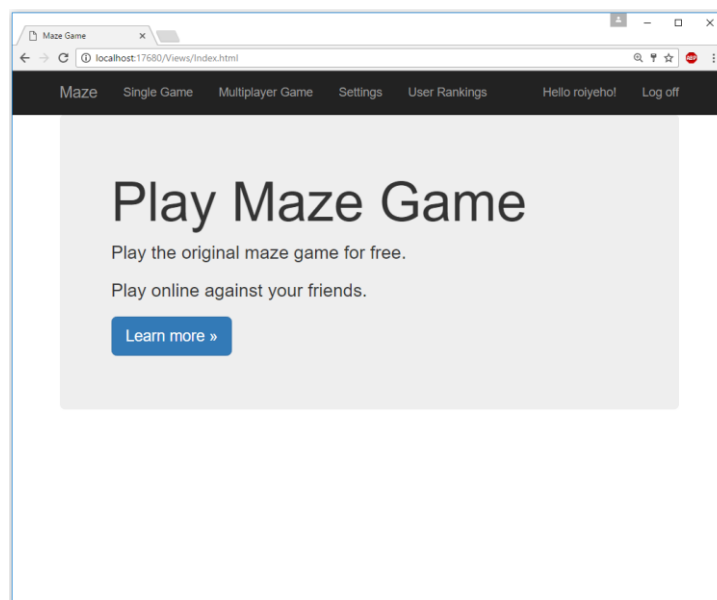
טרם שליחת הטופס אל השרת יש לוודא שכל השדות אינם ריקים ומכילים ערכים תקינים (למשל שאימות סיסמא תואם לסיסמא). לאחר שליחת הטופס אל השרת – יש לשמור את הפרטים של המשתמש בבסיס הנתונים. את הסיסמא יש להצפין באמצעות פונקציית Hash (כדוגמת SHA1) לפני שמירתה בטבלה. במידה ורישום המשתמש בשרת הצליח, יש לשמור את שם המשתמש שנרשם ב- SessionStorage בקליינט ולהעביר את המשתמש אל דף הבית.

דף התחברות (Login):

לאחר הרישום הראשוני, המשתמש יוכל מעתה והלאה להתחבר אל המשחק דרך מסך ההתחברות בעזרת שם המשתמש והסיסמא שהזין במסך ההרשמה. במידה ותהליך ההתחברות הסתיים בהצלחה, המשתמש ישאר מחובר עד אשר יסגור את חלון הדפדפן או עד אשר יבצע התנתקות (Logout).
דוגמא לדף התחברות:



לאחר כניסה למערכת יופיע בפניה הימנית של סרגל הכלים, שם המשתמש שהתחבר וכפתור 'התנתק' אשר יבטל את ה- Session הנוכחי שלו ויעביר אותו לדף הבית:



משחק יחיד:

בצד שמאל של הדף המשתמש יוכל להזין את פרטי המשחק: שם המבוך, מספר השורות ומספר העמודות.

לחיצה על Start Game תגרום לשליחת בקשה א-סינכרונית לשרת (באמצעות AJAX). בזמן ההמתנה לתשובה יוצג למשתמש Loader (חיווי טעינה). כאשר התשובה חוזרת מהשרת, יוצג המבוך שחזר למשתמש והתמונה של השחקן תמוקם בנק' ההתחלה.

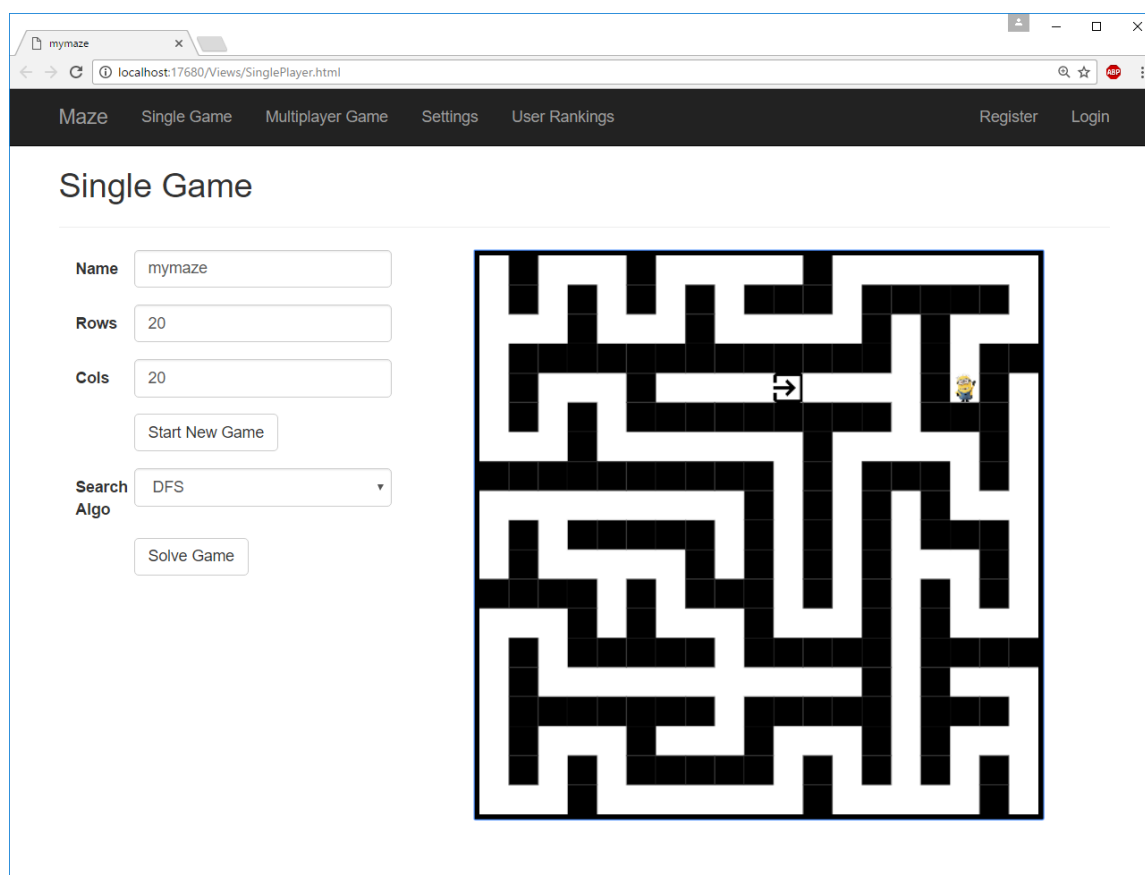
המשחק עצמו (תזוזה במבוך) אינה מצריכה פניה אל השרת. השחקן יוכל לזוז עם החצים במבוך ומטרתו להגיע אל נק' היעד. כאשר השחקן מגיע ליציאה תוצג לו הודעת הצלחה.

בנוסף למשתמש תהיה יכולת לבקש את פתרון המבוך מהשרת. הפתרון יזיז את הדמות באנימציה מנקודת הכניסה לנקודת היציאה מהמבוך.

בכותרת החלון יש להציג את שם המבוך.

במידה ואין תקשורת מול השרת (מכל סיבה שהיא), יש להציג הודעה מתאימה.

הדף יכול להיראות למשל כך:



את ציור המבוך בצעו ע"י בניית רכיב jQuery Plugin בשם mazeBoard. רכיב זה יקבל אלמנט Canvas המוגדר בדף ה-HTML ויציר עליו את המבוך. בנוסף הוא יטפל באירועים שונים של המבוך (תזוזת השחקן, הצגת פתרון וכו').

הפעלת ה- Plugin תבצע ע"י:

```
var myMazeBoard = $("#canvasId").mazeBoard(
    mazeData,           // the matrix containing the maze cells
    startRow, startCol, // initial position of the player
    exitRow, exitCol,   // the exit position
    playerImage,        // player's icon (of type Image)
    exitImage,          // exit's icon (of type Image)
    true,               // is the board enabled (i.e., player can move)
    function (direction, playerRow, playerCol) {
        // a callback function which is invoked after each move
    }
}
```

משחק נגד שחקן יריב:

בחלק השמאלי של הדף המשתמש יוכל לבחור האם להצטרף למשחק קיים או לפתוח משחק חדש. במידה והמשתמש בחר לפתוח משחק חדש עליו להזין את פרטי המבוך בדומה למשחק יחיד (שם המבוך, מספר השורות ומספר העמודות).

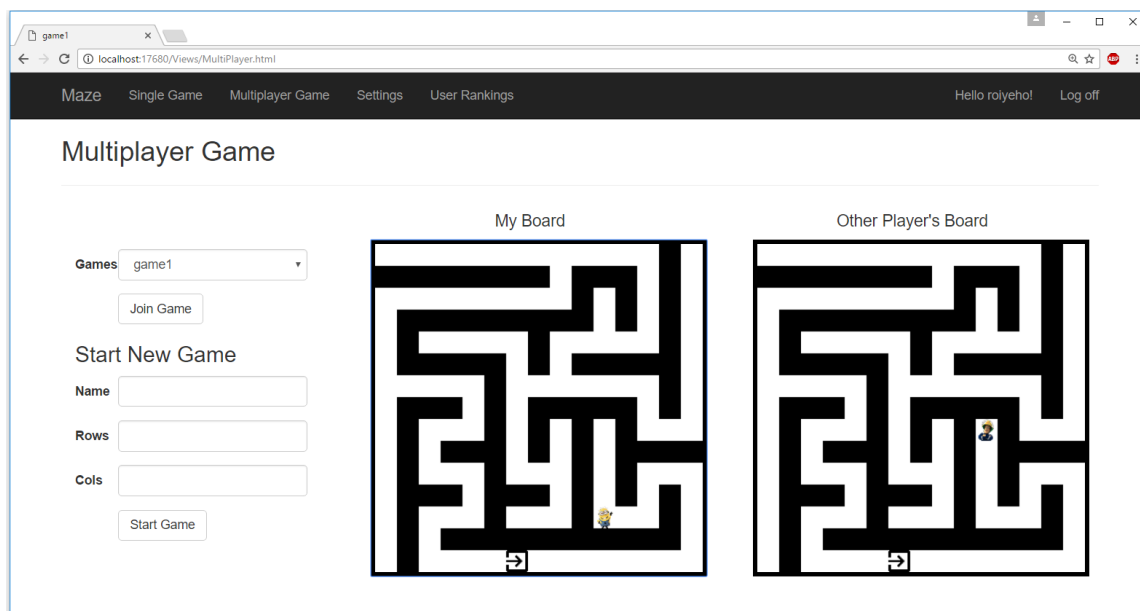
לאחר כניסה למשחק יוצגו שני מבוכים בחלק הימני של הדף. מבוך אחד יציג את המבוך שלי ויתנהג בדיוק כמו משחק יחיד. המבוך השני יציג את המבוך של השחקן השני ויצג את ההתקדמות של השחקן השני. כל פעולה שהשחקן יבצע תופיע בממשק של השחקן השני ולהיפך, כל פעולה שהיריב מבצע תופיע על הממשק של השחקן הנוכחי (כמובן בחלק המתאים של המסך). הראשון שיסיים את המשחק יוכרז כמנצח.

שימו לב שרוב הפעולות במסך זה מחייבות שליחת בקשות א-סינכרוניות (הודעה למשתמש השני על התזוזות שלי, קבלת עדכונים על התזוזות של השחקן השני והצגתן וכו').

לאחר סיום המשחק, ישלח דיווח לשרת מי המשתמש שניצח במשחק ומי המשתמש שהפסיד. נתונים אלה יישמרו בבסיס הנתונים בשרת.

בתפריט החלון יוצג שם המשחק. במידה ואין תקשורת מול השרת (מכל סיבה שהיא), יש להציג הודעה מתאימה. במידה וממתינים לשחקן השני שיתחבר יש להציג חיווי על כך במסך.

הדף יכול להיראות למשל כך:



דף הגדרות:

- דף זה יאפשר למשתמש לשמור את הגדרות ברירת המחדל הבאות:
- גודל המבוך הדיפולטיבי (מספר שורות ומספר עמודות). כאשר יוצרים משחק חדש, גודל המבוך שיופיע בברירת המחדל במסך יקבע עפ"י מה שהוזן במסך ההגדרות.
 - אלגוריתם החיפוש הדיפולטיבי (DFS או BFS)
- הגדרות אלו ישמרו ב- Local Storage של המשתמש.

דף הצגת שיאים:

לחיצה על כפתור "User Rankings" בתפריט תציג את רשימת השחקנים ממוינת לפי הניקוד שלהם בסדר יורד. ניקוד של שחקן יקבע לפי מספר הנצחונות שלו פחות מספר ההפסדים שלו.

דף זה יכול להיראות כך:

User Rankings

Rank	Username	Wins	Losses
1	roiyeho player since 21/4/2017	2	1
2	test player since 21/4/2017	1	1
3	inbary player since 21/4/2017	0	1

צורת עבודה:

שרת ה- Web מחליף את השרת שכתבתם בתרגילים הקודמים. את הלוגיקה של יצירת משחקי המבוך ופתרונם עליכם להעביר לשרת ה- Web (רצוי על-ידי העברתם ל- Class Library נפרד, שיוכל לשמש הן את השרת הקודם והן את השרת החדש).

אותו שרת Web מאחסן גם את כל קוד צד הקליינט שיש להכיל כדי לנהל את המשחק (כל דפי ה- HTML, הסקריפטים, התמונות וכו').

הגישה לשירותים של יצירת מבוך ופתרונם יתבצעו באמצעות Web API. שימו לב שאת מחרוזת ה- JSON המכילה את התשובה לא תוכלו להחזיר ישירות לקליינט (כיוון שה- Web API ינסה לבצע לה שוב סריאליזציה ל- JSON). במקום זאת, עליכם להחזיר את ה- JObject המקורי, למשל:

```
[HttpGet]
public JObject GenerateMaze(string name, int rows, int cols)
{
    Maze maze = model.GenerateMaze(name, rows, cols);
    JObject obj = JObject.Parse(maze.ToJSON());
    return obj;
}
```

את הניהול של משחק מול יריב יש לבצע באמצעות ספריית SignalR, המאפשרת תקשורת דו-כיוונית בין הקליינט לשרת. הספרייה ניתנת להורדה דרך NuGet Package Manager.

את נתוני המשתמשים יש לשמור בטבלת Users בבסיס הנתונים SQL Server Express המגיע כחלק מהתקנת ה- Visual Studio. הגישה לבסיס הנתונים תבצע באמצעות ORM (Entity Framework). עליכם לוודא שבספריית ההגשה שלכם הקובץ של בסיס הנתונים (קובץ .mdf) נמצא בתיקיית App_Data של פרויקט ה- Web.

הנחיות כלליות:

1. דפי ה-HTML יכילו קוד HTML בלבד. סקריפטים והגדרות עיצוביות יופיעו בקבצי JS או CSS מתאימים בלבד.
2. אין לבצע "ריפרושים" מיותרים. טעינה של מידע משרת ה-Web אל הקליינט צריך להתבצע בצורה א-סינכרונית (בעיקר רלוונטי למקרה של משחק בין שני שחקנים).
3. קוד ה-JavaScript צריך להיות עפ"י ה-naming conventions המקובלים של השפה (למשל שמות של פונקציות מתחילות באותיות קטנות ולא גדולות כמו ב-C#). יש לזכור לתעד גם את הקוד הכתוב בצד הקליינט.
4. את קוד ה-JavaScript יש לכתוב בצורה מודולרית עם חלוקה הגיונית למודולים/פונקציות. יש להימנע משימוש במשתנים גלובליים.

צורת הגשה:

- את התרגיל ניתן להגיש בזוגות. ניתן, אך לא חובה.
- עליכם להגיש קובץ zip שיכיל 3 תיקיות:
1. תיקיית src: אשר תכיל את כל הקוד של ה-Solution הכולל את כל הפרוייקטים, כולל פרויקט ה-Web.
 2. תיקיית exe: אשר תכיל את קבצי ההרצה (קבצי exe) של כל אחד מהפרוייקטים – כל אחד בתיקה נפרדת. במקרה שלנו את קובץ ההרצה של הקליינט ושל הסרבר (כמובן בנפרד), וכמובן את הקבצים הנלווים הנדרשים כמו ה-dll-ים וקובץ ההגדרות. שימו לב שעל הקבצים להיות standalone ולא להסתמך על קבצים מחוץ לתיקייה.
 3. תיקייה בשם etc: אשר תכיל קובץ בשם info.txt אשר יכיל את שמות המגישים, ת"ז, קבוצת תרגול של כל אחד. בנוסף, היא תכיל את ה-log של העבודה מול ה-git.
- כל קבצי ההרצה יורצו על מערכת windows עם .net framework. חדש אשר אמור לתמוך בכל הגרסאות אחורה. מחובתכם לוודא שכל הקבצים שהגשתם תקינים, ובפרט שהגשתם את כל הקבצים הנדרשים. מומלץ לוודא שהקבצים שלכם רצים תקין במחשב שונה משלכם טרם ההגשה.
- לא ניתן יהיה להגיש קבצים מחדש או רטרואקטיבית או תיקונים או שינויים לקבצים לאחר מועד ההגשה.
- רק אחד מבני הזוג מגיש את התרגיל. **חובה** להגיש אך ורק לקבוצת התרגיל שאתם רשומים אליה.
- את התרגיל יש להגיש עד ה-15.6 בשעה 22:00 דרך המודל.
- בפועל המערכת תאפשר הגשה עד 23:59 של היום הנקוב, אך הגשה לאחר 22:00 היא "על אחריותכם" בלבד. לא ניתן להגיש באיחור כלל.

בהצלחה

רועי וואלי