

## תרגיל מס' 2 – ארבע בשורה אחת - קונסול

### מטרות

- הטמעה של עבודה עם מחלקות ותכנות מונחה עצמים
- שימוש במחלקה string
- שימוש ב-Dll (אסמבלי) חיצוני
- עבודה עם מערכים/מחלקות אוספים

### ידע נדרש

- היכרות עם כלי הפיתוח Microsoft Visual Studio .NET
- הכרות עם תחביר בסיסי של שפת C#
- שימוש במחלקה string
- עבודה עם מחלקות (מצייני גישה, בנאים, מאפיינים)
- שימוש ב-Dll (אסמבלי) חיצוני
- עבודה עם מערכים/מחלקות אוספים

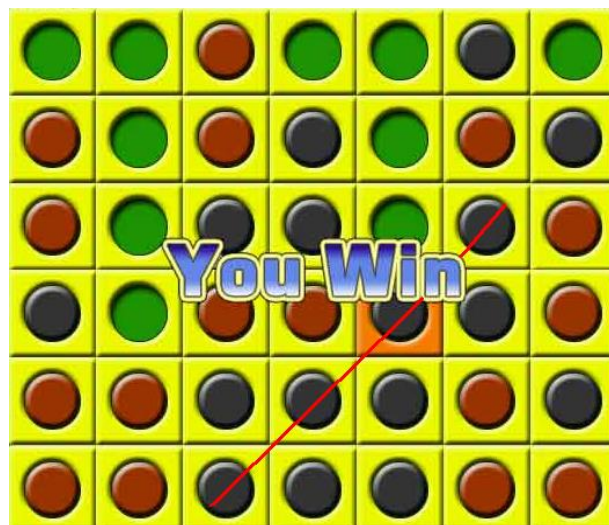
### הכינותי מראש

- קובץ Ex02.ConsoleUtils.dll שנמצא כחלק מחבילת ההורדה מהאתר.

### התרגיל

יש לממש משחק "ארבע בשורה אחת" כאשר ה-console הוא הממשק למשתמש. למי שלא מכיר, המשחק הוא בעצם שיכלול של המשחק "איקס מיקס דריקס".

**במקור**, לוח המשחק הוא בעצם מטריצה ריקה, שיכולה להכיל מטבעות משני סוגים, לכל משתמש יש את הסוג שלו. כל משתמש בתורו מכניס ז'יטון למטריצה, במטרה ליצור רצף של 4 מטבעות מהסוג שלו.



התוכנה תאפשר לשני שחקנים לשחק אחד נגד השני, ותאפשר לשחק נגד המחשב (אשר יחליט החלטות רנדומליות לגבי המהלכים שלו). בונס של עד 10 נקודות יינתן למי שיממש גם בינה מלאכותית במהלכי המחשב (כלומר מהלכי המחשב מבוססי החלטות מחושבות).

## מהלך המשחק

1. המשתמש מתבקש לקבוע את גודל המטריצה (לא פחות מ-4x4, לא יותר מ-8x8. לא חייב להיות ריבועי)
2. המשתמש מתבקש להחליט האם המשחק הוא של שני שחקנים או נגד המחשב.
3. מטריצה ריקה תופיע כמו בדוגמא הבאה (בהינתן גודל של 6x5):

```

      1   2   3   4   5   6
    |   |   |   |   |   |
    =====
    |   |   |   |   |   |
    =====
    |   |   |   |   |   |
    =====
    |   |   |   |   |   |
    =====
    |   |   |   |   |   |
    =====
    |   |   |   |   |   |
    =====

```

4. כל משתמש בתורו יתבקש לבחור את העמודה שלתוכה הוא רוצה להטיל את המטבע שלו.
5. אם העמודה כבר מלאה, תינתן הודעה מתאימה, עד אשר הוא ייבחר עמודה שיש בה מקום.
6. אחרי שהמשתמש בוחר את העמודה, יתנקה המסך, הלוח יצייר שוב, כשהוא מכיל את הז'יטון הנוסף שהוטל בקצה ערימת הז'יטונים בעמודה:

```

      1   2   3   4   5   6
    |   |   |   |   |   |
    =====
    |   |   |   |   |   |
    =====
    |   |   | O | O |   |   |
    =====
    |   | X | X | X |   |   |
    =====
    |   | O | X | O | O |   |
    =====

```

7. במידה ואין רצף של 4 מטבעות מאותו סוג, התור עובר לשחקן השני.
8. במידה ויש (מאונך, מאוזן או אלכסון) – יוכרז על המנצח שיקבל נקודה ויוצג מצב הנקודות.
9. במידה והלוח התמלא לגמרי, יוכרז על תיקו ויוצג מצב הנקודות.
10. במצב של משחק נגד המחשב, מהלך המשחק לא ישתנה, מעבר לעובדה שהמחשב לא מחכה שיבקשו ממנו לבצע תור.
11. בכל שלב ניתן לפרוש מהמשחק ע"י בחירת "Q" במקום בחירת עמודה (השחקן השני יקבל נקודה).
12. במצב של סיום משחק (תיקו, ניצחון, או פרישה) המשתמש יישאל אם ברצונו לשחק סיבוב נוסף. סיבוב נוסף הוא עם אותו גודל לוח ועם אותם שחקנים שממשיכים לצבור ניקוד.

## הוראות כלליות

1. לצורך המהלכים של המחשב יש להגריל עמודה מבין העמודות הפנויות. לצורך כך ניתן להשתמש במתודה Next של המחלקה Random. (כלומר אין חובה לממש אינטליגנציה מלאכותית כלשהיא עבור המהלכים של המחשב).
2. ניתן להשתמש במחלקה `List<T>` שמממשת מערך שניתן להוסיף לו איברים בצורה דינאמית (בדומה ל-`Vector<T>` ב-CPP). דוגמת קוד לשימוש בה נמצאת באתר הקורס.
3. מימוש אינטליגנציה מלאכותית (AI) כלשהיא למהלכים של המחשב יזכה בבונוס של עד 10 נקודות.
4. יש לבדוק תקינות קלט בכל בקשת קלט מהמשתמש ולהציג הודעה מתאימה במקרה של קלט לא תקין.
5. לפני כל הדפסה של מצב הלוח יש לבצע ניקוי מסך. **לשם כך, יש להיעזר בקובץ `Ex02.ConsoleUtils.dll` שמכיל את מחלקת השירות `Ex02.ConsoleUtils.Screen` שמכילה את המתודה הסטטית `.Clear()`.**  
הדרך לפניה ל-dll שלי מתוך הקוד שלכם היא פשוטה:  
**יש למקם את ה-dll בתיקייה `C:\Temp`.** (חשוב מאוד. לא לשים במקום אחר!)  
ללחוץ כפתור ימני של עכבר בחלון ה-Solution Explorer, מעל ה-References של הפרויקט.  
לבחור באופציה Add Reference, ואז לבחור ב-dll באמצעות האפשרות Browse.  
כעת, ניתן לגשת ל-Namespace שמוגדר בתוך ה-dll ולהפעיל מתודות מתוך ה-dll באופן זהה להפעלת מתודות ספריה רגילות. חשוב: **אין לצרף את הקובץ הזה בהגשה (אחרת GMAIL ידחה את העבודה שלכם).** (לבדוק העבודות יש את הקובץ אצלו במקום זהה)  
6. **ארכיטקטורה והנדסת תוכנה:**
  - a. יש להשתמש בתכנות מונחה עצמים! בחירה נכונה של מחלקות וחלוקה נכונה למתודות מהווים נקודות חשובות בתרגיל זה.
  - b. יש להפגין שימוש נכון ביכולות C# 2.0 ודוט-נט.**יש לבצע הפרדה מתאימה בין המחלקות שמנהלות את הלוגיקה והנתונים של המשחק לבין המחלקות שמציגות את מצב המשחק ומייצרות אינטראקציה עם המשתמש.**  
**במילים אחרות:** הפרדה בין המימוש של ממשק המשתמש (UI) למימוש של הלוגיקה של המערכת. כלומר צריך לקחת בחשבון שהחלקים הרלוונטיים במימוש אמורים לשרת אתכם גם במצב שבו תרצו לפתח את המשחק לסביבת "חלונות" והשאיפה היא להשאיר את שכבת הלוגיקה ללא שינוי!

## הגשה

- יש להגיש עד ליום ב', 16 למאי 2016, בשעה 22:00.
- ההגשה תתבצע לתיבת המייל של התרגילים של הקורס כמפורט במסמך הוראות הגשה שניתן למצוא באתר הקורס. נקודות ירדו למי שלא יפעל בדיוק לפי ההוראות.
- אין צורך לתעד בפורמט XML (עוד לא למדנו)
- יש לעמוד בתקנים לכתיבת קוד כפי שמפורט במסמך הרלוונטי שניתן למצוא באתר הקורס.
- אי עמידה בתקנים גוררת הורדה בניקוד.
- כל יום איחור יוריד 2 נקודות מציון התרגיל.
- ניתן להיעזר בקבוצת הפייסבוק של הקורס כדי לשאול שאלות בנוגע לתרגיל.
- נא לא לענות לשאלות של אחרים ולא לפרסם שאלות שמציגות פתרון.
- **נא לממש את העבודה באופן עצמאי לחלוטין ולהימנע מהעתקות (הן מתגלות מאוד בקלות)**

בהצלחה ☺