

תרגיל מס' 2 – דמקה אנגלית - לקונסול

מטרות

- הטמעה של עבודה עם מחלקות ותכנות מכון עצמים
- עבודה עם Properties, Constructors, Enums
- 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
- שימוש ב-Dll (אסמבלי) חיצוני
- עבודה עם מערכים ואוספים

הכינותי מראש

- Microsoft Visual Studio מותקן על המחשב
- קובץ Ex02.ConsoleUtils.dll שנמצא כחלק מחבילת ההורדה מהאתר.

התרגיל

יש לממש משחק "דמקה" כאשר ה-console הוא הממשק למשתמש.

התוכנה תאפשר לשני שחקנים לשחק אחד נגד השני, ותאפשר לשחק נגד המחשב, כאשר במצב כזה המחשב מחליט החלטות רנדומאליות לגבי המהלכים שלו, מתוך מבחר המהלכים החוקיים שיש לרשותו. בונס של עד 7 נקודות יינתן למי שיממש גם בינה מלאכותית במהלכי המחשב (כלומר מהלכי המחשב מבוססי החלטות מחושבות). מי שמממש את הבונס נדרש לציין זאת בגוף המייל של ההגשה בחלק של ההערות, ורצוי גם לפרט קצת אודות אופן המימוש.

מהלך המשחק

1. המשתמש מתבקש להזין את שמו (מחרוזת נטולת רווחים, מקסימום 20 תווים)
2. המשתמש מתבקש לקבוע את גודל הלוח (6 או 8 או 10)
3. המשתמש מתבקש להחליט האם המשחק הוא של שני שחקנים או נגד המחשב.
4. במידה והוא בוחר באפשרות של שני שחקנים, הוא מתבקש להזין את גם את שם השחקן השני.
בתחילת סיבוב, יופיע לוח כמו בדוגמא הבאה (בהיתן גודל של 8x8):

```

      A   B   C   D   E   F   G   H
=====
a|   | O |   | O |   | O |   | O |
=====
b| O |   | O |   | O |   | O |   |
=====
c|   | O |   | O |   | O |   | O |
=====
d|   |   |   |   |   |   |   |   |
=====
e|   |   |   |   |   |   |   |   |
=====
f| X |   | X |   | X |   | X |   |
=====
g|   | X |   | X |   | X |   | X |
=====
h| X |   | X |   | X |   | X |   |
=====

```

Ron's turn:

(סידור החיילים הוא כזה שתמיד יש שתי שורות רווח בין השורות ללבנים. כלומר בלוח של 10 על 10, יש 4 שורות לכל אחד, בלוח של 8, יש 3 שורות לכל אחד ובלוח של 6 יש 2 שורות לכל אחד).

5. כל משתמש בתורו יתבקש לבחור את המשבצת שממנו הוא רוצה לזוז ואת המשבצת שאליה הוא רוצה לעבור. בפורמט הבא: COLrow>COLrow (למשל Af>Bf).
6. אם המהלך אינו מהלך חוקי, המשתמש יקבל הודעה מתאימה, ויתבקש להזין מהלך שונה.
7. אחרי שהמשתמש מזין מהלך חוקי, ינוקה המסך, הלוח יציור שוב תוך שהוא מייצג את מצב המשחק החדש (דוגמא אחרי שהמשתמש הראשון הזין את המהלך Gf>He):

```

      A   B   C   D   E   F   G   H
=====
a|   | O |   | O |   | O |   | O |
=====
b| O |   | O |   | O |   | O |   |
=====
c|   | O |   | O |   | O |   | O |
=====
d|   |   |   |   |   |   |   |   |
=====
e|   |   |   |   |   |   |   | X |
=====
f| X |   | X |   | X |   |   |   |
=====
g|   | X |   | X |   | X |   | X |
=====
h| X |   | X |   | X |   | X |   |
=====

```

Ron's move was (X): Gf>He

Tal's Turn (O):

8. במידה והמהלך היה דילוג מעל שחקן יריב, המטבע של שחקן היריב מושמד.
כדי לאפשר מספר דילוגים ברצף, אחרי כל דילוג תתבצע בדיקה האם קיימת אפשרות דילוג נוספת של כלי המשחק שביצע את הדילוג הנוכחי. אם כן, התור נשאר אצל השחקן שביצע את הדילוג הנוכחי, והדילוג הבא הופך להיות הצעד החוקי היחיד שלו!
9. כאשר מטבע מגיע לשורה האחרונה (האחרונה של היריב) הוא הופך ל"מלך". מלך של O מסומן ע"י U, מלך של X מסומן ע"י K. מלך, בניגוד למטבע רגיל, יכול לצעוד ולקפוץ גם אחורה.
10. בכל שלב ניתן לפרוש מהמשחק ע"י הזנת "Q" כקלט במקום הזנת מהלך.
השחקן שפרש, מפסיד במשחקו.
11. במידה ואחרי המהלך עדיין אין ניצחון או תיקו, התור עובר לשחקן היריב (סעיף 4)
12. ומידה ויש ניצחון (כל מטבעות היריב הושמדו או שליריב אין מהלכים חוקיים) או תיקו (לשני הצדדים אין מהלכים חוקיים) – יוכרז תיקו, או המנצח (שיקבל נקודות שהן ההפרש בין כמות החיילים שלו לכמות החיילים של המפסיד. מלך שווה 4 חיילים) ויוצג מצב הנקודות המצטבר.
13. במצב של סיום סיבוב (תיקו, ניצחון, או פרישה) המשתמש יישאל אם ברצונו לשחק סיבוב נוסף.
סיבוב נוסף הוא עם אותו גודל לוח ועם אותם שחקנים שממשיכים לצבור ניקוד.
14. במצב של משחק נגד המחשב, מהלך המשחק לא ישתנה, מעבר לעובדה שהמחשב לא מחכה שיבקשו ממנו לבצע תור אלא מזין למשחק את המהלך שלו.

עוד לגבי חוקי המשחק:

- את חוקי המשחק ניתן למצוא כאן:
<http://www.netintellgames.com/checkersrules.htm>
- כאן ניתן להוריד התקנה של המשחק לצורך אימון/הבנה:
<http://www.netintellgames.com/archives/netcheck.exe>
 - מהלך חוקי רגיל הוא צעד אלכסוני אחד בכיוון היריב ("קדימה").
 - "דילוג" (או "אכילה") מעל כלי יריב הוא מהלך חוקי נוסף שבו יש כלי יריב במשבצת סמוכה (אלכסונית) למטבע של השחקן, ומעבר לכלי של היריב (באותו כיוון אלכסוני) יש משבצת פנויה אליו רוצים לדלג.
 - כאשר ניתן לדלג - חובה לבצע אותו! אם ישנן מספר דילוגים אפשריים, רשאי השחקן לבחור מביניהם.
 - אם אחרי דילוג יש דילוג חוקי נוסף **בעזרת אותו הכלי**, חובה לבצע גם אותו.
 - למטבע אסור לצעוד אחורה ואסור לדלג אחורה.
 - שימו לב, בניגוד לחוקי הדמקה הרוסית, בדמקה האנגלית גם מלך רשאי לנוע רק צעד אחד בלבד באלכסון. (היתרון היחיד שלו זה שהוא רשאי לנוע ולקפוץ גם אחורנית).

[המשך בעמוד הבא]

הוראות כלליות

- לצורך המהלכים של המחשב יש להגריל מהלך מבין המהלכים החוקיים שלו. לצורך כך ניתן להשתמש במתודה Next של המחלקה Random. (כלומר אין חובה לממש אינטליגנציה מלאכותית כלשהיא עבור המהלכים של המחשב).
- יש לבדוק תקינות קלט בכל בקשת קלט מהמשתמש ולהציג הודעה מתאימה במקרה של קלט לא תקין.
- 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)
- אחרי מהלכים של המחשב, אפשר להפעיל את המתודה `System.Threading.Thread.Sleep(3000)` שמעכבת את המשך ריצת התוכנית ב-3 שניות כדי שהמשתמש יוכל לראות מה היה המהלך של המחשב.
- **ארכיטקטורה והנדסת תוכנה:**
 - יש להשתמש בתכנות מונחה עצמים! בחירה נכונה של מחלקות וחלוקה נכונה למתודות מהווים נקודות חשובות בתרגיל זה.
 - יש להפגין שימוש נכון ביכולות C# 2.0 ודוט-נט.
 - **יש לבצע הפרדה מתאימה בין המחלקות שמממשות 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!
 מימוש אינטליגנציה מלאכותית (AI) כלשהיא למהלכים של המחשב יזכה בבנוס של עד 7 נקודות.
 - ניתן להיעזר בקבוצת הפייסבוק של הקורס כדי לשאול שאלות בנוגע לתרגיל.
 - יש לעמוד בתקנים לכתיבת קוד כפי שמפורט במסמך הרלוונטי שניתן למצוא באתר הקורס. נקודות ירדו למי שלא יעמוד בתקנים אלו.
 - יש לעמוד בהוראות ההגשה כפי שמפורט במסמך הרלוונטי שניתן למצוא באתר הקורס. נקודות ירדו למי שלא יפעל לפי הוראות אלה.
 - נא להימנע מהעתקות! (הן מתגלות מאוד בקלות)

הגשה

- תאריך ההגשה הוא יום ד', 27 לאפריל 2022, בשעה 22:00.
כל יום איחור מוריד 2 נקודות מציון התרגיל.

בהצלחה ☺