

תרגיל מס' 2 – משחק הזיכרון לקונסול

מטרות

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

ידע נדרש

- עבודה עם מערכים / מחלקות אוספים
- עבודה עם מחלקות (Access modifiers, Constructors, Properties, Indexers)
- שימוש במחלקה string
- שימוש ב-Dll (אסמבלי) חיצוני

הכינותי מראש

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

התרגיל

עליכם לממש את משחק הזיכרון (באופן פרימיטיבי יחסית). כאשר ה-console הוא הממשק למשתמש. התוכנית תבקש תחילה מהמשתמש את גודל הלוח הרצוי (מספר עמודות ומספר השורות). המצב ההתחלתי יהיה לוח בגודל סימטרי של עמודות X שורות לפי בקשת המשתמש. התוכנית לא תאפשר בחירת כמות משבצות אי זוגית (למשל 5 X 5). דוגמא ללוח התחלתי במקרה שהמשתמש ביקש 6 עמודות ו-4 שורות:

	A	B	C	D	E	F
1						
2						
3						
4						

בכל שלב המשתמש ייתבקש לבחור משבצת בלוח כדי לחשוף את האות שנמצאת במשבצת הנ"ל. דוגמא: המשתמש בחר לגלות את משבצת E3 ומאחוריה הוגרלה האות T. התוכנית תצייר את הלוח כך:

	A	B	C	D	E	F
1						
2						
3					T	
4						

בשלב זה ייתבקש המשתמש לבחור משבצת נוספת שבה הוא חושב שנמצאת האות T הנוספת. המשבצת שבוחר תחשף באופן דומה לזה שבו נחשפה המשבצת הראשונה (הלוח יצויר שוב).

מהלך המשחק

1. המשתמש מתבקש להזין את שמו (הוא השחקן הראשון).
2. המשתמש מתבקש להחליט האם המשחק הוא של שני שחקנים או נגד המחשב.
3. במידה והוא בוחר באפשרות של שני שחקנים, הוא מתבקש להזין את שמו של השחקן השני.
4. המשתמש מתבקש לקבוע את גובה ורוחב הלוח (מינימום 4x4 מקסימום 6x6 וחובה כמות משבצות זוגית).
5. בתחילת סיבוב המטריצה ריקה ותצויר באופן כזה (במקרה של 6x4):

	A	B	C	D	E	F
1						
2						
3						
4						
6. כל שחקן בתורו יתבקש לבחור את המשבצת הראשונה שאותה הוא רוצה לחשוף.
- אם הבחירה אינה חוקית תינתן הודעה מתאימה וכך הלאה עד אשר הוא יבחר משבצת חוקית (משבצת חוקית היא משבצת מוסתרת בגבולות הלוח. כל קלט אחר נחשב לא חוקי ויש להוציא הודעה שמסבירה את הסיבה לאי-החוקיות שלו).
7. אחרי שהשחקן בוחר משבצת, ננקה את את המסך, נצייר את הלוח כפי שהוא כרגע אחרי שחשפנו את האות במשבצת הרלוונטית.
8. בשלב זה על השחקן לבחור את המשבצת שמהווה לדעתו התאמה, לפי ההוראות בסעיף 6 ו-7.
9. במידה והמשתמש הצליח לחשוף זוג תואם, הזוג יישאר חשוף עד סוף המשחק, השחקן יקבל נקודה, והתור יישאר אצלו לקבלת קלט חדש לפי הסעיף הקודם (כלומר חוזר לסעיף 6).
10. במידה והמשתמש חשף זוג לא תואם, הזוג הלא תואם יישאר חשוף למשך 2 שניות וכעבור 2 שניות שתי המשבצות שבחר ישובו להיות ריקות והתור עובר לשחקן השני לקבלת קלט (כלומר חוזר לסעיף 6).
11. מצב הלוח אחרי שהשחקן הראשון בחר את המשבצות E3 ו-B4:

	A	B	C	D	E	F
1						
2						
3					T	
4		R				

12. כשהזוג האחרון נחשף, המשחק נגמר ונקבע ניצחון לשחקן עם מספר החשיפות הגדול יותר (תצא הודעה מתאימה הכוללת את מצב הנקודות בין השחקנים. כל שחקן מקבל נקודות לפי מספר החשיפות שלו).
13. אחרי סיום משחק, המשתמש יישאל אם ברצונו להתחיל סיבוב נוסף. אם כן חוזרים לסעיף 4. אם לא, תוצג הודעה והתוכנית תסתיים.
14. החל מסעיף 6, בכל שלב ניתן לפרוש מהמשחק ע"י בחירת "Q" במקום משבצת (פרישה כללית מהתוכנה).

הוראות כלליות

- לצורך המהלכים של המחשב יש להגדיל משבצות מבין המשבצות שמהוות מהלך חוקי. לצורך כך ניתן להשתמש במתודה Next של המחלקה Random. (כלומר אין צורך לממש אינטליגנציה מלאכותית כלשהיא עבור המהלכים של המחשב)
- לצורך מימוש ההמתנה של 2 שניות יש להשתמש במתודה Sleep של המחלקה System.Threading.Thread
- יש לבדוק תקינות קלט בכל בקשת קלט מהמשתמש ולהציג הודעה מתאימה במקרה של קלט לא תקין (יש להבדיל בין קלט לא תקין תחבירית לבין קלט לא תקין עניינית).
- 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!)
לחוץ 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). (לבודק יש את הקובץ אצלו במקום זהה)
- 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 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. קחו בחשבון שחלק מהמימושים אמורים לשרת אתכם גם בשלב שבו נרצה לפתח את המשחק לסביבת "חלונות" והשאיפה היא להשאיר כמה שיותר חלקים ללא שינוי!
- מימוש אינטליגנציה מלאכותית (AI) כלשהיא למהלכים של המחשב יזכה בבנוס של עד 6 נקודות!
- אין צורך לתעד את הקוד בתיעוד XML.
- ניתן להעזר בקבוצה בפייסבוק כדי לשאול שאלות בנוגע לתרגיל.
- יש לעמוד בתקנים לכתיבת קוד כפי שמפורט במסמך הרלוונטי שניתן למצוא באתר הקורס. שימו לב בעיקר לתקנים לגבי שמות של שדות מחלקה ופרמטרים לפונקציה. נקודות יירדו למי שלא יעמוד בתקנים אלו.
- יש לעמוד בהוראות ההגשה כפי שמפורט במסמך הרלוונטי שניתן למצוא באתר הקורס. נקודות יירדו למי שלא יפעל באופן מדויק לפי הוראות אלה. נא להמנע מהעתקות (הן מתגלות מאוד בקלות)

הגשה

- יש להגיש את התרגיל לא יאחר מיום א', 31 למאי 2020, בשעה 22:00.
- לא תאושרנה בקשות לדחייה שלא קשורות לסיבות פורמליות (כגון מילואים/מחלה).

בהצלחה ☺