**יישומי בינה מלאכותית**

**סמסטר חורף תשע"ז**

**מטלה תכנותית 1**תאריך הגשה: 18.1.17 23:55

* המטלה התכנותית בשפת התכנות C#.
* יש לשאול שאלות במודל, בפורום מטלה תכנותית 1.
* יש להגיש קובץ word או pdf המכיל 1-2 דפים ובהם הסבר של האלגוריתם שהחלטתם לממש וכן את הקובץ: "player1.cs" בלבד.
* בקובץ "player1.cs" יש לממש את השיטה: " playYourTurn" בלבד (ניתן להוסיף שיטות עזר פרטיות וכן מחלקות עזר פרטיות לקובץ עצמו, אין להגיש קבצים נוספים).
* אין ליצור בנאי למחלקה זו ואין להשתמש במשתנים ושיטות סטטיות.
* העבודה להגשה בזוגות בלבד.
* לפני שניגשים לממש את המטלה מומלץ לעיין רבות בהסברים וכן בקוד הקיים.
* פתרון המטלה שתגישו ייבדק מול שאר ההגשות ע"י תוכנת העתקות, וכן מגיש אשר יימצא כי העתיק עלול לפגוע בהמשך לימודיו או בציונו בקורס.
* הפרוייקט נכתב ויבדק בסביבת העבודה MS Visual Studio 2013.
* לאחר מתן הציון על עבודה זו נקיים טורניר בין ההגשות בעלות ההישגים הטובים ביותר. שלושת הזוגות שינצחו בטורניר יקבלו בונוס לציון הסופי (מקום 3 – 3 נק', מקום 2 – 5 נק', מקום 1 – 8 נק').
* יש למלא את תעודות זהות המגישים בשיטה getPlayers.

**הקדמה**:  
המטלה הינה בנושא משחקים בבינה מלאכותית. במטלה ממומש המשחק "הריבועים הנעלמים", המשחק נערך בלוח בגודל rows X cols המכיל ריבועים.

כל שחקן בתורו בוחר ריבוע מהלוח, כתוצאה מכך נעלם הריבוע הנבחר וכן נעלמים כל הריבעים שהינם מתחתיו **וגם** מימינו.

מטרת המשחק הינה **לא** להיות השחק שלוחץ על הריבוע השמאלי עליון בלוח.

מומלץ להריץ את הקובץ " Example.exe" בו ממומש המשחק ולשחק על מנת להבין טוב יותר את החוקים.

cols ו- rows הינם משתנים שאינם ידועים לכם (כלומר: לא ידוע לכם מה יהיה גודל לוח המשחק עליו אריץ את הבדיקות).  
במטלה יהיה עליכם לממש את אחד משחקני המשחק (שחקן ‘1’), כאשר השחקן השני (‘2’) כבר מומש.

מטרת המטלה היא לבנות אלגוריתם לכל צעד\תור במשחק כך שתנצחו את השחקן הממומש. תור שחקן במשחק הינו תלוי זמן, כלומר, ישנו טיימר הבודק כי לא חרגתם מהזמן המוגדר לתור.  
למשחק קיימת דרגת קושי (כפי שתוגדר בהמשך). ציונכם יקבע על פי רמות הקושי השונות בהן תצליחו לנצח וכן במגוון הלוחות השונים בהם תנצחו.

באפשרותכם לממש את האלגוריתם בכל דרך שתמצאו לנכון.

**מחלקות הפרוייקט**:

מחלקת לוח (Board)  
מחלקת לוח המשחק הינה המחלקה המחזיקה את לוח המשחק.  
לוח המשחק מוגדר כמערך דו-מימדי בגודל שנקבע מראש.  
בתא יחיד בלוח יכול להיות אחד מהתווים הבאים:  
'X' – משבצת המכילה ריבוע וניתן לבחור בה.  
' ' – משבצת ריקה (בה היה ריבוע ונעלם).  
  
משתניי המחלקה:

int \_rows מספר שורות הלוח -

int \_colsמספר עמודות הלוח -

char[,] \_board לוח המשחק -

int \_squaresLeft כמות הריבועים שנותרו בלוח -

שיטות המחלקה (באפשרותכם להשתמש בהם לעזר):

* בנאי המחלקה –

public Board(int rows, int cols)

* בנאי המחלקה (מעתיק לוח קיים) -

public Board(Board toCopy)

* שיטה הבודקת האם המשחק נגמר -

public bool isTheGameEnded()

* שיטה המנסה למלא את התא בלוח בתו השחקן שהגיע כקלט, השיטה מחזירה: "אמת"- אם הצליחה, "שקר"- אחרת -

public bool fillPlayerMove(int row, int col, char player)

* שיטה המנסה למחוק תו הקיים בלוח במיקום שהתקבל כקלט, השיטה מחזירה: "אמת"- אם הצליחה, "שקר"- אחרת -

public bool unFillPlayerMove(int row, int col)

* שיטה המחזירה "אמת" אם התא במיקום הנבחר ריק, "שקר" אחרת-

public bool checkIfCellIsEmpty(int row, int col)

* שיטה המדפיסה את תכולת הלוח הנוכחי-

public void printTheBoard()

מחלקת שחקן (Player) - מחלקה אבסקטית

מחלקת שחקן הינה מחלקה אבסטרקטית ממנה תצטרכו לבצע ירושה ולממש את שיטתה.

משתניי המחלקה:

* שחקן, תו המייצג מי השחקן במשחק (במקרה שלכם יקבל 'X')-

ֹ\_player

* יריב, תו המייצג מי היריב במשחק (במקרה שלכם יקבל 'O')-

ֹ\_opponent

שיטות המחלקה:

* שיטה אבסטרקטית אותה תצטרכו לממש,   
  השיטה מקבל כקלט:   
  1) board – לוח המשחק הנוכחי   
  2) timesup – משך הזמן בו תצטרכו לעמוד, כלומר: עליכם להחזיר מיקום בו אתם בוחרים לשחק לפני שנגמר הזמן, אי עמידה בזמן תגרור הפסד במשחק הנוכחי.   
  השיטה מחזירה:  
  זוג המייצג את המיקום בלוח בו אתם בוחרים.

public Tuple<int, int> playYourTurn(Board board, TimeSpan timesup)

מחלקת שחקן1 (1Player)

מחלקת "שחקן1" היא מחלקת השחקן שלכם במשחק.

כעת ממומש אלגוריתם רנדומאלי, עליכם לשנותו לאלגוריתם לשלכם.

עליכם לממש בה את השיטה:

public Tuple<int, int> playYourTurn(Board board, TimeSpan timesup)

המקבלת את הלוח הנוכחי במשחק ואת הזמן לתור ומחזירה זוג המייצג את בחירת השחקן.  
  
מחלקת שחקן2 (2Player)   
מחלקת "שחקן2" היא המחלקה של השחקן היריב שלכם במשחק (מחלקה שכבר מומשה).  
  
מחלקת Program  
המחלקה הראשית, המחלקה שמנהלת את מהלכו התקין של המשחק.  
לפני הרצת התוכנית יקבעו מספר משתנים:  
במחלקה מוגדרים מספר משתנים קבועים מראש אותם ניתן לשנות לפני הרצה כדי להבין טוב יותר את תפקודכם במשחק.  
  
המשתנים:

* מספר המשחקים שהמחלקה תריץ-

public const int m\_numberOfGames

תפקודכם במשחק יקבע על סמך משחקים שלמים רבים, כאשר כל משחק יתחיל במהלך של שחקן אחר.

* מספר שורות לוח המשחק-

public const int m\_boardRows

* מספר עמודות לוח המשחק-

public const int m\_boardCols

* רמת קושי-

public const int m\_gameLevel

רמת הקושי במשחק תקבע את כמות הזמן שיהיה לשחקן1 (לכם) על מנת לשחק תור במשחק:  
רמת קושי 1 – 200ms  
רמת קושי 2 – 150ms

רמת קושי 3 – 100ms

רמת קושי 4 – 80ms

רמת קושי 5 – 50ms

ל-"שחקן2" (השחקן הממומש) יש 100ms לכל תור.

* האם להדפיס את תוצאת כלל המשחקים?-

public const bool m\_printAllResults

בהצלחה!