שולה מוקשים – גירסת חלונות

משחק המבוסס על שולה המוקשים המפורסם של מיקרוסופט.

**עקרונות המשחק:**

נתון לוח משחק (מטריצה) ובה מתחבאים מוקשים.

בשלב הראשון הלוח מוסתר.

לחיצה על משבצת מסוימת עלולה להיות מסוכנת, ולפוצף מוקש.

אם זה קרה – המשחק יסתיים.

אחרת, יופיע על הלוח כמה שכנים עם מוקשים יש למשבצת עליה לחצנו.

אם אין לה שכנים מסוכנים, יופיע המספר 0, ויחשפו גם המספרים של השכנים כל עוד גם הם עם 0 שכנים, התהליך מסתיים עד אשר מגיעים למשבצת עם לפחות שכן אחד. בלוח הבא ניתן לראות משבצות שלא נחשפו עדיין (בצבע אפור בהיר), משבצות עם מספר בין 1 ל-8, המציינים כמה מוקשים שכנים יש למשבצת, ומשבצות באפור כהה יותר, המציינות שאין להן שכנים עם מוקשים.



**האלגוריתם הבסיסי**

המשחק מתנהל כל עוד לא עלינו על מוקש, ועד שכל המשבצות ללא מוקשים נחשפו.

על המסך יש להציג בנוסף למצב המשבצות החשופות, את מספר הצעדים שבוצעו עד כה, נחמד יהיה להציג גם את הזמן שעבר מתחילת המשחק.

בסיום משחק, נצחון או הפסד, תוצג הודעה מתאימה, ויוצע לשחקן להתחיל משחק חדש.

**הדרישות:**

1. פיתוח משחק שולה מוקשים בסביבת Windows Form
2. הצגת מצב הלוח אחרי כל צעד (לחיצה)
3. הצגת מספר הצעדים המצטבר
4. הודעת ניצחון או הפסד בסוף
5. אופציונאלי והרחבות:
	1. הצגת שעון זמן
	2. אפשרות להתחיל את השלב מחדש
	3. אפשרות לסמן משבצת חשודה, ולמחוק את הסימון
	4. מספר רמות קושי
	5. חשיפת כל המוקשים על ידי פקודה מיוחדת.

**תהליך פיתוח המשחק וחישוב הציון**

העבודה היא אישית! יהיו מספר דפי הנחיה עם הסברים ורמזים. אותם אעלה למודל בשלבים

לצורך פיתוח המשחק נגדיר את האובייקט מוקש - Cell:

Cell– יכיל את כל האינפורמציה על המשבצת. למשל:

* האם זה מוקש?
* כמה שכנים מוקשים יש לו (לא כולל אותו)?
* האם הוא מוצג?
* האם יש עליו דגל סימון?

בהצלחה ובהנאה!

**שולה מוקשים - דף הנחיה 1**

איך מתחילים?

1. תחת הפרוייקט נגדיר מחלקה Cell כפי שלמדנו עד היום, נגדיר לה את כל התכונות והפעולות.
2. נפתח פריט חדש מסוג Windows Form, זהו הבסיס לעבודה.
3. נדאג לפתוח במקביל (דרך תפריט view) את ארגז הכלים – toolbox.
4. הוסיפו לטופס שדות מסוג Label. שם יופיעו מספר המוקשים, מספר התור, זכיה/הפסד, וכדומה.

בהמשך תוכלו להוסיף שעון טיימר המראה כמה זמן חלף, וכפתורי שליטה שונים..

1. נוסיף/נגרור לחלון שלנו אובייקט מסוג PictureBox, מומלץ שיהיה בחלק הימני של הטופס, על מנת שנוכל להגדילו מבלי לפגוע בפריטים שהוספנו בסעיף הקודם.



pictureBox

1. את קובץ הקוד נפתח על ידי קליק ימני של העכבר על שם הטופס (בחלונית מצד ימין), ונבחר view code
2. שימו לב שהפעולה הבונה כבר קיימת! הוסיפו לה קריאה לפעולה פנימית שלכם המייצרת את לוח המשחק.
3. רשימת תכונות שתזדקקו להן. הערה: ניתן להוסיף עוד...

private int x0, y0; // matrix origin position

private float side = 20F; // grid size

private int rows, cols, mines; // number of rows, columns and mines

private Cell[,] board;

private Graphics gr; // the Graphics Object that serves all methods

private static Random rnd;

private bool gameOver;

private int turn;

private static Font fontS = new Font("Wingdings", 14, FontStyle.Regular);

private static Font fontT = new Font("Arial", 12, FontStyle.Regular);

private static Pen pen = new Pen(Color.Black);

private static Brush bg;

private Timer timer;

private System.Diagnostics.Stopwatch stopWatch;

1. הקוד הבא דואג לצייר טבלה/מטריצה על המסך הגרפי. כדאי להחזיקו בפעולה נפרדת המזומנת מתוך הפעולה הבונה.

// Set screen size

כאן נקבע את גודלי איזורי העבודה שלנו:

Width ו-Height הם מימדי כל הטופס

לעומת canvas.Width ו- canvas.Height שהם מימדי איזור ה-picturebox.

השם canvas הוא שם המשתנה של איזור הציור של לוח המשחק.

המשתנים cols, rows, ו-side מתארים את מספר השורות, העמודות וגודל כל משבצת.

אתם אחראים לתת להם ערכים מתאימים.

 int wMargin = Width - canvas.Width; // חישוב הרוחב מעבר לאזור הלוח

 int hMargin = Height - canvas.Height; // חישוב הגובה מעבר לאזור הלוח

 int w = cols \* side + 1;

 int h = rows \* side + 1;

 Width = wMargin + w;

 Height = hMargin + h;

 // disable screen resize

 MaximizeBox = false;

 FormBorderStyle = FormBorderStyle.FixedSingle;

 // initialize canvas

 Bitmap bm = new Bitmap((int)(cols \* side)+1, (int)(rows \* side)+1);

 canvas.Image = bm;

 canvas.Size = new Size(w, h);

 gr = Graphics.FromImage(bm);

 gr.TextRenderingHint = System.Drawing.Text.TextRenderingHint.AntiAlias;

 DrawMatrix(0, 0, rows, cols, side); // זו הפעולה המציירת את הלוח

 bg = new SolidBrush(BackColor); // נשמור על צבע הרקע בכדי "למחוק"

 Play();

1. ציור הלוח:

private void DrawMatrix(int x0, int y0, int rows, int cols, int gridSize)

מציירת את המטריצה החל מ-x0,y0. מציירת rows שורות ו-cols עמודות, במרווחים של gridSize

המשתנים x0, y0, gridsize נקבעים במהלך בניית הטופס (הפעולה הבונה). זה לשיקולכם.... ומשפיע על גודל איזור הלוח והטופס בכללותו.

ציור כל קו נעשה באמצעות DrawLine הנמצאת באובייקט הגרפיקה gr שבנינו למעלה.

**חישבו כיצד נצייר את המטריצה?**

**משימתכם**: להחליט על גודל הלוח, ולצייר אותו בחלון.

בשלב הראשון מומלץ להחליט על גודל אחד קבוע מראש. בשלב מאוחר יותר ניתן להוסיף למשתמש אפשרות בחירת רמת משחק, הקובעת אל גודל הלוח.



**שולה מוקשים - דף הנחיה 2**

**כיצד נתפוס לחיצות עכבר על גבי הלוח?**

בחלון התכנון הגרפי, בחרו את איזור הלוח ה-canvas, ובתכונות מצד ימין למטה, חפשו את האירוע MouseClick.

הקשה כפולה על התא הריק מימינו תיצור פעולה העונה לאירוע כזה. סטודיו יעביר אתכם ישירות לקובץ הקוד אל תוך הפעולה. שימו לב שהפעולה מקבלת שני פרמטרים: אובייקט בשם sender, ואובייקט אירועים בשם e. באובייקט השני תוכלו למצוא את הקואורדינטות שבו נלחץ העכבר, וגם האם נלחץ בכפתור ימני או שמאלי.

נסו לכתוב פעולה המשתמשת ב-MessageBox.Show(“string”) ובה תודיעו איזה כפתור נלחץ, ומה הקואורדינטות. רמז: בידקו מהן התכונות הזמינות לכם מהאובייקט המועבר כפרמטר e.

לאחר מכן חשבו מתוך הקואורדינטות, על איזו משבצת על הלוח לחצתם. הציגו מספר זה, ובידקו האם צדקתם.

בהצלחה!

**שולה מוקשים - דף הנחיה 3**

**האובייקט Cell**

תפקידו של אובייקט זה לשקף מה קורה בכל משבצת במשחק. הוא יכיל את כל התכונות הנחוצות לנו בכדי להתקדם במשחק.

תכונות האובייקט:

 public class Cell

 {

 private bool isMine; // האם מוקש?

 private int neigbors; //כמה שכנים מוקשים?

 private bool visible; // האם נחשף (לצורכי תצוגה)?

 }

הפעולה הבונה אחראית לאיתחול התכונות, אבל עדיין איננה יודעת על כמות השכנים. תכונה זו תקבל ערך באמצעות פעולת Set מתאימה.

במחלקת המשחק (המחלקה המכילה את הטופס הראשי), נגדיר מערך דו-מימדי של Cell בגודל שציירנו את המטריצה. לאחר בנית המערך (בניית אובייקט לכל תא), נגריל את המוקשים לפי הכמות שהחלטנו.

טיפ:

רמת קושי סבירה: 9% מוקשים (כלומר 9 מוקשים על 100 משבצות)

רמת קושי גבוהה: 18% מוקשים.

בנוסף, ככל שהמערך גדול יותר, רמת הקושי עולה.

**ספירת השכנים:**

בדומה למשחק החיים, עלינו לספור כמה שכנים מוקשים יש לכל תא במערך. פעולה זו מתבצעת פעם אחת בלבד! משום שמפת המוקשים אינה משתנה.

נעשה זו באמצעות פעולה חיצונית ל-Cell. ועבור כל חישוב שנבצע נעדכן את האובייקט המתאים.

בהצלחה!

**שולה מוקשים - דף הנחיה 4**

**חשיפה**

עם לחיצה על עכבר, נבדוק איזו משבצת נלחצה. אם היא כבר נחשפה (visible) אין מה להמשיך.

אם לא נחשפה, עלינו לחשוף אותה, אם משבצת זו היא מוקש, המשחק הסתיים! השחקן הפסיד.
נחשוף את כל המוקשים על הלוח, ונודיע לו הודעה מתאימה.

אם איננה מוקש, נציג את כמות השכנים שיש למשבצת. אם יש לה לפחות מוקש אחד, התהליך מסתיים כאן.
אחרת (יש לה 0 שכנים מוקשים), נמשיך לחשוף את שכניה עד שנגיע למשבצת שאיננה עם אפס שכנים, או שכבר נחשפה.
כאן נשתמש ברקורסיה.... חישבו איך.

**סיום המשחק:**

אחרי תהליך החשיפה, נבדוק האם רק המוקשים נותרו לא חשופים? אם כן יש ניצחון, אחרת, המשחק ממשיך.

בהצלחה!

**שולה מוקשים - דף הנחיה 5**

**תוספות ושיפורים**

ניתן להוסיף למשחק הרבה שיפורים מעבר לבסיס, דוגמאות:

1. בחירת רמת הקושי לפני תחילת המשחק
2. ספירה והצגת מספר התורות/מהלכים שבוצעו
3. הצגת שעון זמן שייעצר עם סיום המשחק
4. לחצן ימני לסימון דגל, העוזר לשחקן
5. כפתור המאפשר להתחיל את אותו המשחק (אותה חלוקת מוקשים) פעם נוספת
6. כפתור המאפשר להתחיל משחק חדש
7. מקש סודי החושף את כל המוקשים לכמה שניות ו"מחביא אותם שוב"
8. שימוש בצבעים שונים למטרות שונות

בהצלחה!