

שולה מוקשים - דף הנחיה 2

דף הנחיה זה מכיל שני חלקים:
בראשון תמצאו הסבר נוסף על האלגוריתם ובשני הדגמת שימוש באובייקט Mine בצורה קלה יותר.

חלק א'

האלגוריתם לפי שלבים:

1. בניית לוח משחק בגודל שרוצים, למשל 10x10. זיכרו הלוח נמצא בתוך האובייקט הראשי – לוח משחק. לאובייקט זה יש מספר תכונות: לוח המשחק - מערך דו מימדי, מספר המוקשים, מספר התור, ואובייקט האקראיות Random. המערך הדו מימדי הינו מערך של האובייקט הבסיסי (ראו חלק שני דף זה).
2. יש להגריל את מיקומם של המוקשים. ההגרלה נעשית בלולאה כמספר המוקשים. בכל פעם מגרילים שורה ועמודה ומציבים שם מוקש.
חישבו: מה קורה אם כבר יש שם מוקש? איך מתגברים על זה?
3. עכשיו נוכל להוסיף לכל תא את מספר שכניו. יש לעבור על כל הלוח (מערך דו מימדי) ולספור כמה מוקשים מסביבו, ולאחסן באובייקט התא.
זהירות: מה קורה שנמצאים בקצה הלוח?
4. חישבו על מגוון הפעולות הדרושות לכם. למשל:
פעולת Click שתגיב ללחיצה על משבצת מסויימת. מתפקידה יהיה לבדוק האם מוקש, ואם לא אז לחשוף את המקומות הדרושים להיחשף. חשיפה ראשונה היא המשבצת עצמה. ההמשך בפעולות נוספות.
הפעולה תחזיר האם עלינו על מוקש.
פעולה נוספת היא **פעולה החשיפה**. פעולה זו תקבל אינדקס לשורה ואינדקס לעמודה. הפעולה תחשוף את כל 8 השכנים.
בתור שלב ביניים, ניתן לעצור כאן.
במשחק האמיתי עלינו להמשיך להתקדם ולחשוף את כל ה-'0' עד למשבצות שאין 0.
פעולת הדפסה – תנקה את המסך ותדפיס את הלוח. במהלך ההדפסה הפעולה גם תספור ותבדוק האם כל המשבצות שאין מוקשים נחשפו. אם כן סיימנו.
חשיפת מוקשים – היא פעולה המתבצעת לאחר שדרכנו על מוקש, ואז המשחק יראה לנו את כל המוקשים שהסתתרו בלוח.

חלק ב'

בבואנו להגדיר אובייקט, נחשוב על הצורך ואופן השימוש בו. לפעמים נרצה אובייקטים שיפעילו פעולות, למשל הצב, ולפעמים נרצה אובייקטים שעיקר שימושם הוא מארז של תכונות.

האובייקט הראשי של המשחק הוא מהסוג הראשון, ושם רוב עבודתכם.
האובייקט Mine מייצג מוקש או תא במערך, יחזיק אוסף של תכונות: האם מוקש? מספר שכנים? האם מוצג?

על מנת להקל על חיינו, ולהימנע מהצורך בשימוש בהרבה פעולות, נגדיר תכונות של אובייקט כזה עם הרשאת גישה ציבורית (להבדיל ממה שלמדנו עד כה, הרשאה פרטית), וכך נוכל לגשת לתכונות באופן ישיר.
שימו לב, בהמשך נדגים כיצד נראה אותו האובייקט כאשר תכונותיו פרטיות.

```
public class Mine
{
    public bool isMine;
    public int neighbors;
    public bool visible;
    public int id;
    private static int idAll = 0;

    public Mine()
    {
        isMine = false;
        neighbors = 0;
        visible = false;
        id = idAll++;
    }
}
```

על מנת לגשת לתכונות של Mine נבצע את הפעולות הבאות:

```
Mine mine = new Mine();
if (mine.visible)
{
    Console.WriteLine("Mine is visible");
}
mine.neighbors = 5;
```

סדרת הפעולות הנ"ל בונה אובייקט חדש, מבררת מה מצב ה-`visible` שלו, ומציגה הודעה בהתאם, ולבסוף מעדכנת את מספר השכנים שלו.

וכעת נראה את אותו האובייקט עם תכונות פרטיות:

```
public class Mine
{
    private bool isMine;
    private int neighbors;
    private bool visible;
    private int id;
    private static int idAll = 0;

    public Mine()
    {
        isMine = false;
        neighbors = 0;
        visible = false;
        id = idAll++;
    }
}
```

```

public bool IsMine()
{
    return isMine;
}
public void SetMine(bool mine)
{
    this.isMine = mine;
}
public int GetNeighbors()
{
    return neighbors;
}
public void SetNeighbors (int neighbors)
{
    this. neighbors = neighbors;
}

// Here come the rest of the methods
}

```

על מנת לגשת לתכונות של Mine נבצע את הפעולות הבאות:

```

Mine mine = new Mine();
if (mine.IsVisible())
{
    Console.WriteLine("Mine is visible");
}
mine.SetNeighbors(5);

```

בהצלחה!