

# مشروع برمجة 2

اسم المشروع:

**Light out**

باستخدام

C# و

Windows Forms

تقديم الطالبان:

صفاء محمود الطرشان

بيان محمد غياث صعيدي

## تقرير مفصل عن تطوير لعبة Lights Out باستخدام C# و Windows Forms

اسم المشروع: لعبة Lights Out

بيئة التطوير: C# باستخدام Windows Forms

نوع التصميم: تصميم برمجي وليس يدوي، حيث تم إنشاء عناصر الواجهة والتحكم في منطق اللعبة باستخدام الكود.

أهداف المشروع

تطوير لعبة **Lights Out** حيث يحتاج اللاعب إلى إطفاء جميع الأضواء باستخدام الضغط على الأزرار .  
تحقيق تجربة لعب منطقية وتفاعلية تعتمد على تبديل حالة الأزرار المجاورة .  
إضافة إمكانية الفوز والخسارة بحيث تكون اللعبة متوازنة وتحديها معقول .  
تصميم واجهة ديناميكية تسمح بتعديل حجم الشبكة من خلال خيارات قابلة للتخصيص.

### مراحل التنفيذ

#### ◆ تصميم الواجهة البرمجية (UI) باستخدام Windows Forms

- إعداد ComboBox لاختيار حجم الشبكة.
- إنشاء Label لعرض عدد الحركات.
- إضافة Button لبداية لعبة جديدة.
- إنشاء شبكة ديناميكية للأزرار بناءً على الحجم المختار.

#### ◆ تطوير المنطق الأساسي للعبة (LightsOutGame.cs)

- إنشاء مصفوفة Grid تمثل حالة تشغيل وإطفاء الأزرار.
- تطوير ToggleLights() لتحديث حالة الأزرار بطريقة صحيحة.
- تحسين CheckWinCondition() بحيث يتحقق من الفوز عند إطفاء جميع الأضواء.
- تطوير CheckLossCondition() بحيث يخسر اللاعب عند الوصول إلى وضعية غير قابلة للحل.

#### ◆ إضافة ميزات متقدمة

- إمكانية تغيير حجم الشبكة ديناميكيًا (3x3, 4x4, 5x5, 6x6).
- حالة الفوز عندما يتم إطفاء جميع الأضواء مع رسالة تهنئة .
- إمكانية الخسارة عندما تصبح الشبكة غير قابلة للحل مع عرض تقرير مفصل .
- تحليل الشبكة لمعرفة ما إذا كانت هناك خطوات منطقية لإكمالها.

#### ◆ التحديات والحلول

- مشكلة إعادة تشغيل الأزرار بعد الضغط عليها  
تم حلها عبر تعديل ToggleLights() بحيث لا يتم تبديل الأزرار التي تم إطفائها بالفعل.
- عدم ظهور رسالة الفوز عند إطفاء جميع الأضواء  
تم تحسين CheckWinCondition() بحيث يتم التحقق من الفوز بعد كل حركة.
- إضافة طريقة للخسارة بدون تحديد عدد الحركات  
تم استخدام CheckLossCondition() لتحليل الشبكة ومعرفة إذا كانت قابلة للحل أم لا.