

**Beschreibung des Spiels:**

Auf einem quadratischen 5x5-Feld aus 25 Zellen, die anfangs leer sind, soll durch Anklicken einzelner Zellen erreicht werden, dass abschließend alle Zellen 'X'-markiert sind.

Durch Anklicken einer Zelle werden aber außer dieser auch die darüberliegende und darunterliegende sowie die links und rechts nebenan liegende Zelle bearbeitet. Wird eine Randzelle (z.B. am linken Rand) angeklickt, so wird statt einer (nicht vorhandenen) links danebenliegenden die am rechten Rand in dieser Zeile liegende Zelle bearbeitet. Enthält eine Zelle bereits eine Markierung und wird erneut bearbeitet, so verliert sie die Markierung wieder.

Ziel des Spieles ist es, mit möglichst wenig Zügen (Klicks) alle Zellen des Spielfeldes zu markieren.

**Aufgabenstellung:**

- Erstellen Sie zum Projekt Lampe1 ein 480x600 Pixel-Formular mit
  - 5x5-StringGrid ohne Scrollbars, ohne feste Zeilen/Spalten in der Größe 400x400 Pixel mit Schriftgröße 50pt
  - Buttons zum Starten eines neuen und Abbrechen des laufenden Spiels (oder versuchen Sie auch einmal BitButtons (Register Zusätzlich), für die es in der Eigenschaft Kind bereits vordefinierte Arten gibt.
  - Button zum Beenden des Programms
  - Label und nicht-bearbeitbares Editfeld zur Anzeige der benötigten Züge
- Formulieren Sie einen Grobalgorithmus
  - zum Problem der Kontrolle, ob alle Felder den Text 'X' enthalten (also markiert sind, da dann das Spielende erreicht ist)
  - zur Methode, die beim Anklicken einer Zelle gestartet wird
- Verfeinern Sie die obigen Grobalgorithmen zu Delphimethoden:
  - zur OnSelect-Methode einer StringGridzelle (gehen Sie schrittweise vor: Implementieren Sie zuerst nur das Bearbeiten der aktuellen Zelle, und prüfen Sie, ob der Zyklus Markierung 'X' / keine Markierung korrekt arbeitet. Erweitern Sie dann um die links/rechts benachbarten Zellen, prüfen Sie auch die Korrektheit für Randzellen. Rufen Sie dazu die lokale Prozedur Wechseln (nSpalte,sZeile) auf; erstellen Sie für die Aufrufe bezüglich der vier Nachbarzellen Formeln, um diese Parameter korrekt zu übergeben. Tipp: Versuchen Sie, mit  $x \text{ MODULO } 5$  das erkennbare Problem zu lösen. Vervollständigen Sie erst dann für die oben/unten-liegenden Zellen.)
  - zur lokalen Funktion SpielIstAus innerhalb der OnSelect-Methode einer StringGridzelle
  - zur lokalen Prozedur Wechseln (spalte,zeile), die in der OnSelect-Methode fünfmal mit den entsprechenden Parametern aufgerufen wird
  - Zusätzlich benötigen Sie noch eine Klickmethode zum Starten eines Spiels. Hier sind alle Zellen zu initialisieren, die Anzeige benötigter Züge sichtbar zu machen, diese Anzahl auf 0 zu initialisieren.
  - Die Klickmethode zum Spielabbruch - nur aktiv während eines laufenden Spiels - soll das StingGrid unsichtbar machen.
- Testen Sie das Programm ausgiebig:
  - Werden die gewünschten Zellen bearbeitet?
  - Verändert sich bei nochmaliger Zellenbearbeitung der Inhalt korrekt?
  - Wird die Anzahl benötigter Züge jeweils aktualisiert angezeigt?
  - Beendet das Programm erwartungsgemäß?
- Wenn Sie mit allem fertig sind: Erweitern Sie das Projekt zum Projekt Lampe2: Das Formular erhält zusätzlich:
  - (zusätzlich) GroupBox mit MemoFeld zur Anzeige der Spielinstruktionen aus einer Datei lampenspiel\_info.txt
  - (zusätzlich) GroupBox mit Optionen:
    - wählbare Anzahl Zellen (4-9) in Spalte/Zeile durch Label und SpinEdit
    - wählbare Länge des Zyklus (2-9) statt Ein/Aus-Markierung
 (Ein Zyklus der Länge 3 bewirkt, dass die Zellen nacheinander die Inhalte 2-1-0-2-1-0-2-1-0-... annehmen. Spielende ist dann, wenn alle Zellen gleichzeitig den Wert 0 enthalten.)
- Passen Sie die Click-Methode zum Start eines neuen Spieles an: Wegen der frei wählbaren Zellenanzahl muss nun hier das StringGrid hinsichtlich folgender Eigenschaften bearbeitet werden:
  - Position Top und Left der oberen Ecke (z.B. 8 und 8)
  - Höhe und Breite auf 400, Zeilen- und Spaltenanzahl wie im SpinEdit gewünscht,
  - Zellhöhe und -breite mittels Division (z.B. Zellbreite = Breite-x DIV Spalten)
  - testen Sie, wie groß der Pixelabzug x sein muss, damit alle Zellen sichtbar sind
  - Schriftgröße entsprechend der Zellhöhe-y (testen Sie, wie groß y sein muss)
 Tipp: Verwenden Sie einen global deklarierten String cZyklus, in den Sie die zyklischen Stationen (z.B. '012345') hineinschreiben. Die Zellen werden dann mit cZyklus [ Length (cZyklus) ] initialisiert.
- Formulieren Sie den Algorithmus für das Wechseln der Zelleninhalte neu, wobei nun die zyklische Veränderung korrekt implementiert werden muss.
 

Tipp: Arbeiten Sie mit Modulo statt mit CASE oder IF, notieren Sie, welche Zahlen auf welche folgen (z.B. 5->4, 4->3, ..., 1->0, 0->5) und erarbeiten Sie dazu eine Formel, die auch für alle kleineren/größeren Zyklen gilt. Auch die Funktion SpielIstAus muss nun den gewünschten Endzustand prüfen, der in cZyklus (wo dort?) steht.

