

Aufgabe: Erstelle die Benutzeroberfläche für ein Memory-Spiel in C# mit Windows Forms

Ziel: Du wirst die grundlegende Benutzeroberfläche (UI) für ein Memory-Spiel erstellen, das Karten anzeigt, ohne jedoch die Spiellogik (wie den Timer oder die Kartenübereinstimmung) zu implementieren. Deine Aufgabe ist es, die UI-Elemente zu gestalten und die Formulare korrekt zu organisieren.

Schritt 1: Erstelle ein neues Windows Forms-Projekt

1. Öffne Visual Studio.
2. Erstelle ein neues Projekt:
 - Wähle "Windows Forms App (.NET)".
 - Gib dem Projekt einen Namen (z. B. "MemoryGame") und speichere das Projekt in dem Git-Repository-Ordner.

Schritt 2: Gestalte das Hauptformular

1. **Formulargröße:**
 - Setze die Größe des Formulars so, damit genügend Platz für die Karten vorhanden ist.
2. **Fenstertitel:**
 - Setze den Titel des Formulars auf Memory Game.

Schritt 3: Füge UI-Elemente hinzu

1. Füge ein TableLayoutPanel hinzu. Dies wird verwendet, um die Karten in einem Raster anzuzeigen.
2. Füge eine Hintergrundfarbe hinzu
3. Setze CellBorderStyle auf Inset (somit sieht man Abstände zwischen den Zellen)
4. Stelle das TableLayoutPanel so ein, dass es 4 Zeilen und 4 Spalten hat (insgesamt 16 Karten).
5. Passe die Größe der Zellen so an, dass sie alle gleich groß sind (100:4 -> 25 %)
6. Setze die Dock-Eigenschaft des Panels auf Fill, sodass es das gesamte Formular ausfüllt.

7. Karten-Buttons:

- Füge 16 Buttons zum TableLayoutPanel hinzu (4x4 Raster). Dock wieder auf Fill setzen

(Mittels Copy-Paste können schnell neue Buttons hinzugefügt werden)

- Ändere die Hintergrundfarbe auf die gleiche wie vom Layout

8. Erzeuge in Form1 eine Variable:

```
List<string> icons = new List<string>()
{
    "!", "!", "N", "N", " ", " ", "k", "k",
    "b", "b", "v", "v", "w", "w", "z", "z"
};
```

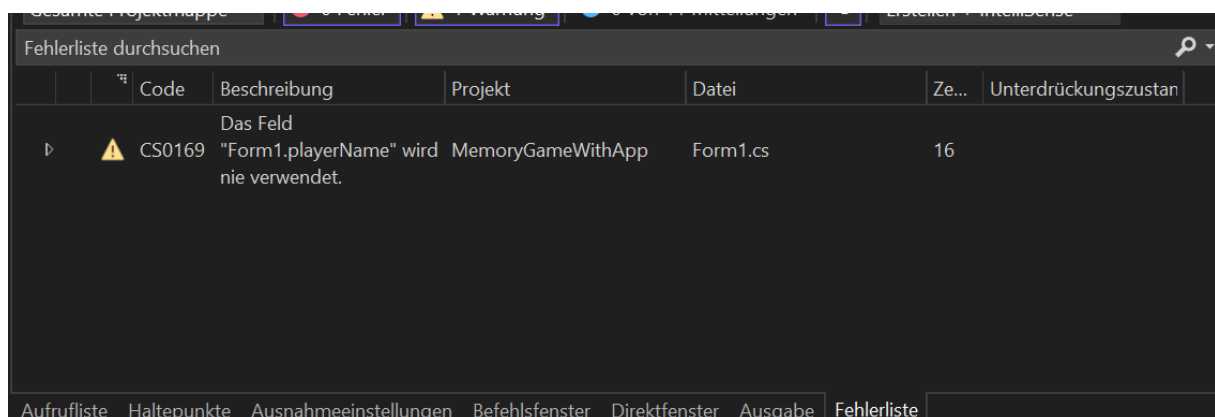
Tipp: Anzahl der Elemente mittels icons.Count

9. Gib jedem dieser Buttons ein Element dieser Liste. Nutze die Klasse Random, die einen Random-Index berechnet. In der Liste soll das Element an dem Random-Index dann gelöscht werden. Nutze das Beispiel in der Vorlesungsfolie von heute auf Seite 15.

Schritt 4: Ereignisse hinzufügen

1. Button-Events:

- Klicke auf einen Button doppelt.
- Gebe vorerst nur aus, dass beim Klicken eines Buttons etwas in der Konsole ausgegeben werden soll
- Gib jedem dieser Buttons die gleiche on-click-Methode



Unter Ausgabe ist dann während des Programmlaufs sichtbar, was in der Konsole ausgegeben wird.

2. **Optional** (wird aber später benötigt): Wenn ihr das Programm ausführt, sollen die Buttons erst keine Symbole zeigen. Beim Klicken soll das Symbol gezeigt werden.

Zuerst soll die Vordergrundfarbe die des Hintergrundes sein.

Nach dem Klicken soll die Vordergrundfarbe wieder sichtbar sein (z. B. auf `Color.Black` gesetzt werden), sodass das Symbol angezeigt wird.