

# Debugging

---

Aufgabe "Debugging", Modul 450

Die folgenden Aufgaben beziehen sich auf die Beispiele **ConnectFour** (die eigentlich Logik), **ConnectFour.Demo** (das interaktive Demoprogramm für die Spiellogik) und **ConnectFour.Test** (das xUnit-Test-Projekt, welches noch keine Tests enthält).

Das Spiel ist im deutschsprachigen Raum als Vier gewinnt bekannt. Zwei Spieler lassen abwechselnd Steine in ein 6x7-Gitter fallen. Das Spiel endet, wenn ein Spieler vier seiner Steine in eine Reihe (horizontal, vertikal, diagonal) gebracht hat und so zum Sieger wird.

## Aufgabe 1: Manueller Test

Starte das Spiel gemäss den Anweisungen im **README.md**. Reagiert das Spiel in bestimmten Situationen bzw. bei bestimmten Eingaben unerwartet? Halte deine Erkenntnisse in einer neuen Datei namens **Testprotokoll.md** fest. Du solltest zumindest folgende Fehler finden:

1. Programmabsturz: Das Programm kann durch eine valide oder invalide Eingabe zum Absturz gebracht werden.
2. Spiellogik: Die Spiellogik ist nicht gemäss den gängigen Regeln implementiert, sondern fehlerbehaftet. Beschreibe die gefundenen Fehler mitsamt Zustand (Spielbrett) und Eingabe, die zum Fehler führte.

## Aufgabe 2: Test-Driven Debugging

Suche dir einen Fehler aus, dem du mithilfe von **Test-Driven Development** auf den Grund gehen möchtest. Schreibe einen Testfall, der den Fehler zuverlässig reproduziert und dabei *"erfolgreich scheitert"*.

## Aufgabe 3: Debugging

Suche dir Debugging-Techniken aus dem Theorieteil aus (zusätzliche Ausgaben auf die Kommandozeile, Einsatz des Debuggers, Durchspielen des Codes im Kopf bzw. auf Papier usw.). Wenn du der Fehlerursache auf den Grund gekommen bist, dokumentiere dein Vorgehen und deine Erkenntnisse in einer neuen Datei namens **Fehlersuche.md**. Korrigiere den Fehler anschliessend, sodass der Testfall erfolgreich durchläuft.