

Testkonzept für Tic-Tac-Toe Spiel

1. Einleitung

Das vorliegende Testkonzept beschreibt die Teststrategie für das Tic-Tac-Toe Spiel. Ziel ist es, die Funktionalität, Zuverlässigkeit und Korrektheit der Spielimplementierung sicherzustellen.

2. Testarten

2.1 Unit Tests

- Fokus auf einzelne Komponenten und Methoden
- Überprüfung der Logik unabhängig von externen Abhängigkeiten
- Abdeckung verschiedener Szenarien und Randfälle

2.2 Integrationstests

- Überprüfung des Zusammenspiels verschiedener Komponenten
- Validierung der Interaktion zwischen Model, View und Controller

3. Testbereiche

3.1 Model-Komponente

- Initialisierung des Spielfelds
- Spielerzuglogik
- Gewinnbedingungen
- Verschiedene Spielfeldgrößen
- Zustandsmanagement und Rückgängig-Funktion

Testfälle Model:

- Spielfeld korrekt initialisieren
- Spieler wechseln
- Gültige und ungültige Züge erkennen
- Gewinnbedingungen für verschiedene Brettgrößen
- Rückgängig-Funktion korrekt implementieren

3.2 Controller-Komponente

- Spielablaufsteuerung
- Interaktion zwischen Model und View
- Züge ausführen
- Spielstatus verwalten

Testfälle Controller:

- Spielzüge korrekt delegieren
- Spielerwechsel
- Gewinnererkennung
- Fehlerzustände behandeln

3.3 View-Komponente

- Korrekte Darstellung des Spielfelds
- Benutzereingaben verarbeiten
- Spielstatus visualisieren

Testfälle View:

- Spielfeld korrekt zeichnen
- Spielerinformationen anzeigen

- Fehlermeldungen darstellen

4. Testszenarien

4.1 Grundlegende Spielszenarien

- Spiel auf 3x3 Brett
- Spiel auf 5x5 Brett
- Spiel auf 7x7 Brett

4.2 Spezielle Testfälle

- Unentschieden erkennen
- Rückgängig-Funktion
- Ungültige Züge verhindern
- Grenzwerte und Randfälle testen

5. Testabdeckung

Angestrebte Abdeckung

- Methodenabdeckung: $\geq 80\%$
- Bedingungsabdeckung: $\geq 70\%$
- Pfadabdeckung: $\geq 60\%$

6. Testumgebung

- Entwicklungsumgebung: Visual Studio
- Testframework: MSTest
- Zusätzliche Tools:

- Code Coverage
 - Statische Codeanalyse

7. Testdokumentation

- Jeder Testfall wird dokumentiert mit:
 - Kurzbeschreibung
 - Vorbedingungen
 - Eingabedaten
 - Erwartete Ergebnisse
 - Tatsächliche Ergebnisse
 - Status (Bestanden/Fehlgeschlagen)

8. Testphasen

1. Entwicklungsbegleitende Tests
2. Kontinuierliche Integration
3. Abschliessende Gesamtsystemtests

9. Risiken und Herausforderungen

- Komplexität der Gewinnbedingungen
- Unterschiedliche Spielfeldgrössen
- Zustandsmanagement

10. Empfehlungen

- Regelmässige Testsuite-Ausführung

- Continuous Integration
- Automatisierte Testausführung