# Tests

## Vorgehen

Wir werden Unit Tests schreiben. Wir werden <http://msdn.microsoft.com/en-us/library/ms243147%28v=vs.80%29.aspx> benutzen. Wir werden die Tests im Nachhinein schreiben, also kein Test Diven Development. Wir werden jedoch zu jeder Klasse sicher einen Test schreiben.

## Testfälle

Folgende Tests werden sicher gemacht. Weitere Tests werden Agile beim Programmieren erstellt.

### Test Spielfeld

* Einfacher Zug nach vorne sollte fehlschlagen
* Einfacher Zug auf leeres Feld
* Zug auf Feld ausserhalb des Spielfelds sollte fehlschlagen
* Einfacher Zug rückwärts sollte fehlschlagen
* Einfacher Zug mit Dame rückwärts sollte funktionieren
* Einfacher Zug auf Feld in letzter Linie sollte Stein in einen Damenstein umwandeln
* Einfacher Zug auf besetztes Feld schlägt Fehl
* Angriffszug sollte zwei Felder nach vorne und Feindlichen Spieler töten
* Angriffszug über freundlichen Spieler sollte nicht funktionieren
* Angriffszug über leeres Feld sollte nicht funktionieren
* Angriffszug auf Feld in letzter Zeile ergibt eine Dame
* Nach Angriffszug sollte ein weiterer Angriffszug möglich sein
* Nach Angriffszug sollte kein einfacher Zug mehr möglich sein
* Wenn ein Spieler eine Angriffszug verpasst, sollte dieser gefressen werden können, jedoch nicht wenn der Spieler einen anderen Angriffszug gemacht hat

### Test KI

* Spiel gegen andere KI ohne einmal Regeln zu verletzten

### Test LAN

Wir müssen noch abklären wie gut man Netzwerktests in C# machen kann.

* LAN Spieler in Netzwerk sollte gefunden werden
* LAN Spieler kann sich Verbinden
* LAN Spieler erfüllt alle Tests die in „Test Spielfeld“ beschrieben sind