

IU Internationale Hochschule

Weiterbildung: "Software Engineering - Python"

Modul : DLMCSPSE01_D - Projekt: Software Engineering

Tutor: Prof. Dr. David Kuhlen



Prüfungsleistung: Portfolio

Testdokument

Eingereicht am 16.09.2025.

Verfasser:

Djahan Bayrami Latran

Denninger Straße 198

81927 München

E-Mail: djahan.latran@gmail.com

Matrikelnummer: UPS10672478

1. Teststrategie

In der Teststrategie wurden manuelle Tests mit automatisierten Unit-Tests kombiniert.

Dabei war das Ziel, sowohl die Datenverarbeitung und Logik der Anwendung als auch die Interaktion durch den Nutzer über die grafische Oberfläche zu prüfen.

Bei den automatisierten Tests lag der Fokus auf der Kernlogik der Anwendung. Mit Unittests wurden die Methoden der Module "Parameters", "Board", "FileReader" und "AppController" (Teile der "Model" und "Controller" Schicht) auf ihre Korrektheit getestet und die Fehlerbehandlung (Exception-Handling) überprüft. In den manuellen Tests wurden visuelle und interaktive Tests der grafischen Benutzeroberfläche, die nicht vollständig automatisiert werden können, durchgeführt. Z.B. die Auswahl eines Algorithmus, die Steuerung über Buttons und die visuelle Darstellung der Algorithmsgschritte. So wurde sichergestellt, dass die Anwendung auch während der Laufzeit aus Sicht des Nutzers korrekt funktioniert. Durch die hybride Teststrategie wurden sowohl die Nutzererfahrung als auch die funktionale Korrektheit der Anwendung getestet.

2. Manuelle Tests

Die Tests wurden mit einer eindeutigen Test-ID, Beschreibung, Vorbedingung, erwarteten Ergebnissen, tatsächlichen Ergebnissen und Status dokumentiert und in die folgenden 4 Kategorien aufgeteilt: Kat.01 - UI Interaktion, Kat.02 - Funktionalität der Algorithmen, Kat.03 - Anzeige von Inhalten externer Dateien, Kat.04 - Stabilität/Robustheit.

Kat.01 - UI Interaktion

Test-ID	Beschreibung	Vorbedingungen	Auszuführende Schritte	Erwartetes Ergebnis	Tatsächliches Ergebnis	Status
UI-01	Starten der Anwendung	Anwendung ist installiert oder Repository geklont	Anwendung starten oder main.py ausführen	Anwendung startet und Algorithmus-Auswahl-Menü wird angezeigt	Anwendung startet und Algorithmus-Auswahl-Menü wird angezeigt	Erfolgreich
UI-02	Algorithmus wählen	Anwendung wurde gestartet	Auswahl eines Algorithmus im Menü	Steuerung, Infos und Animationsfenster werden angezeigt, sowie Name des ausgewählten Algorithmus	Steuerung, Infos und Animationsfenster werden angezeigt, sowie Name des ausgewählten Algorithmus	Erfolgreich
UI-03	"Play"-Button funktioniert	Algorithmus wurde ausgewählt	"Play"-Button klicken	Algorithmus und Visualisierung starten	Algorithmus und Animation starten	Erfolgreich
UI-04	"Pause"-Button funktioniert	Algorithmus wurde gestartet	"Pause"-Button klicken	Algorithmus und Visualisierung pausieren, Stand wird weiterhin angezeigt	Algorithmus und Animation pausieren, Stand wird weiterhin angezeigt	Erfolgreich
UI-05	"Reset"-Button funktioniert	Algorithmus wurde ausgewählt	"Reset"-Button klicken	Zustand von Algorithmus und Visualisierung werden zurück gesetzt - bei "Basic Sort" und "Basic Search" Kategorie werden neue Zufallswerte generiert, bei "Graph Traversal" Kategorie werden die Werte der Nodes zurückgesetzt	Zustand von Algorithmus und Visualisierung werden zurück gesetzt - bei "Basic Sort" und "Basic Search" Kategorie werden neue Zufallswerte generiert, bei "Graph Traversal" Kategorie werden die Werte der Nodes zurückgesetzt	Erfolgreich
UI-06	"Size"-Slider funktioniert	"Basic Sort" oder "Basic Search" Algorithmus wurde ausgewählt	"Size"-Slider bewegen	Anzahl der Werte in der Liste vergrößert sich beim Bewegen nach rechts und verringert sich beim Bewegen nach links - Werte werden während des Bewegens kontinuierlich neu generiert	Anzahl der Werte in der Liste vergrößert sich beim Bewegen nach rechts und verringert sich beim Bewegen nach links - Werte werden während des Bewegens kontinuierlich neu generiert	Erfolgreich
UI-07	"Speed"-Slider funktioniert	Algorithmus wurde gestartet	"Speed"-Slider bewegen	Algorithmus und Visualisierung laufen langsamer beim Bewegen nach links und schneller beim Bewegen nach rechts	Algorithmus und Visualisierung laufen langsamer beim Bewegen nach links und schneller beim Bewegen nach rechts	Erfolgreich
UI-08	"Obstacle"-Button funktioniert	"Graph Traversal"-Algorithmus wurde ausgewählt	"Obstacle"-Button klicken und anschließend auf ein oder mehrere Felder in der Gitterstruktur der Visualisierung klicken	angeklickte Felder in der Gitterstruktur färben sich rot	angeklickte Felder in der Gitterstruktur färben sich rot	Erfolgreich
UI-09	"Set Target"-Button funktioniert	"Graph Traversal"-Algorithmus wurde ausgewählt	"Set Target"-Button klicken und anschließend auf ein Feld in der Gitterstruktur der Visualisierung klicken	angeklickte Feld in der Gitterstruktur färbt sich grün - anschließend weitere Auswahl von Feldern deaktiviert	angeklickte Feld in der Gitterstruktur färbt sich grün - anschließend weitere Auswahl von Feldern deaktiviert	Erfolgreich

Abbildung 1: Testprotokoll Kategorie 01, erstellt mit Google Sheets

Kat.02 - Funktionalität der Algorithmen

Test-ID	Beschreibung	Vorbedingungen	Auszuführende Schritte	Erwartetes Ergebnis	Tatsächliches Ergebnis	Status
ALGO-01	Bubble-Sort sortiert Werte aufsteigend	Bubble-Sort wurde ausgewählt	Algorithmus starten und zu Ende laufen lassen	Algorithmus stoppt und Visualisierung zeigt die Werte in sortierter Reihenfolge, grün gefärbt	Algorithmus stoppt und Visualisierung zeigt die Werte in sortierter Reihenfolge, grün gefärbt	Erfolgreich
ALGO-02	Selection-Bubble-Sort sortiert Werte aufsteigend	Selection-Sort wurde ausgewählt	Algorithmus starten und zu Ende laufen lassen	Algorithmus stoppt und Visualisierung zeigt die Werte in sortierter Reihenfolge, grün gefärbt	Algorithmus stoppt und Visualisierung zeigt die Werte in sortierter Reihenfolge, grün gefärbt	Erfolgreich
ALGO-03	Insertion-Sort sortiert Werte aufsteigend	Insertion-Sort wurde ausgewählt	Algorithmus starten und zu Ende laufen lassen	Algorithmus stoppt und Visualisierung zeigt die Werte in sortierter Reihenfolge, grün gefärbt	Algorithmus stoppt und Visualisierung zeigt die Werte in sortierter Reihenfolge, grün gefärbt	Erfolgreich
ALGO-04	Linear-Search findet Wert	Linear-Search wurde ausgewählt	Algorithmus starten und zu Ende laufen lassen	Algorithmus stoppt und Visualisierung zeigt den gesuchten Wert (grün gefärbt)	Algorithmus stoppt und Visualisierung zeigt den gesuchten Wert (grün gefärbt)	Erfolgreich
ALGO-05	Binary-Search findet Wert	Binary-Search wurde ausgewählt	Algorithmus starten und zu Ende laufen lassen	Algorithmus stoppt und Visualisierung zeigt den gesuchten Wert (grün gefärbt)	Algorithmus stoppt und Visualisierung zeigt den gesuchten Wert (grün gefärbt)	Erfolgreich
ALGO-06	BFS findet Wert	BFS wurde ausgewählt	Ziel und optional Hindernisse auf der Gitterstruktur setzen - Algorithmus starten und zu Ende laufen lassen	Algorithmus stoppt nachdem er den Wert in der Gitterstruktur gefunden hat - alle durchsuchten Felder sind hellgrau gefärbt - alle nicht durchsuchten Felder sind dunkelgrau	Algorithmus stoppt nachdem er den Wert in der Gitterstruktur gefunden hat - alle durchsuchten Felder sind hellgrau gefärbt - alle nicht durchsuchten Felder sind dunkelgrau	Erfolgreich
ALGO-07	DFS findet Wert	DFS wurde ausgewählt	Ziel und optional Hindernisse auf der Gitterstruktur setzen - Algorithmus starten und zu Ende laufen lassen	Algorithmus stoppt nachdem er den Wert in der Gitterstruktur gefunden hat - alle durchsuchten Felder sind hellgrau gefärbt - alle nicht durchsuchten Felder sind dunkelgrau	Algorithmus stoppt nachdem er den Wert in der Gitterstruktur gefunden hat - alle durchsuchten Felder sind hellgrau gefärbt - alle nicht durchsuchten Felder sind dunkelgrau	Erfolgreich
ALGO-08	Dijkstra-Algorithmus findet Wert	Dijkstra wurde ausgewählt	Ziel und optional Hindernisse auf der Gitterstruktur setzen - Algorithmus starten und zu Ende laufen lassen	Algorithmus stoppt nachdem er den Wert und den kürzesten Pfad zu diesem in der Gitterstruktur gefunden hat - alle durchsuchten Felder sind hellgrau gefärbt - alle nicht durchsuchten Felder sind dunkelgrau gefärbt - der kürzeste Pfad ist hellblau gefärbt	Algorithmus stoppt nachdem er den Wert und den kürzesten Pfad zu diesem in der Gitterstruktur gefunden hat - alle durchsuchten Felder sind hellgrau gefärbt - alle nicht durchsuchten Felder sind dunkelgrau gefärbt - der kürzeste Pfad ist hellblau gefärbt	Erfolgreich
ALGO-09	Dijkstra-Algorithmus findet den kürzesten Pfad	Dijkstra wurde ausgewählt	Ziel und optional Hindernisse auf der Gitterstruktur setzen - Algorithmus starten und zu Ende laufen lassen	Algorithmus stoppt nachdem er den Wert und den kürzesten Pfad zu diesem in der Gitterstruktur gefunden hat - alle durchsuchten Felder sind hellgrau gefärbt - alle nicht durchsuchten Felder sind dunkelgrau gefärbt - der kürzeste Pfad ist hellblau gefärbt	Algorithmus stoppt nachdem er den Wert und den kürzesten Pfad zu diesem in der Gitterstruktur gefunden hat - alle durchsuchten Felder sind hellgrau gefärbt - alle nicht durchsuchten Felder sind dunkelgrau gefärbt - der kürzeste Pfad ist hellblau gefärbt	Erfolgreich

Abbildung 2: Testprotokoll Kategorie 02, erstellt mit Google Sheets

Kat.03 - Anzeige von Inhalten externer Dateien

Test-ID	Beschreibung	Vorbedingungen	Auszuführende Schritte	Erwartetes Ergebnis	Tatsächliches Ergebnis	Status
FR-01	themes.json Datei wird geladen - themes werden in die grafische Oberfläche übernommen	themes.json existiert in src/gui	Anwendung starten	Optik der Anwendung entspricht den Werten der themes-Datei	Optik der Anwendung entspricht den Werten der themes-Datei	Erfolgreich
FR-02	code_texts.yaml Datei wird geladen - Inhalt wird im Code-Fenster angezeigt	code_texts.yaml existiert in src/gui	Algorithmus auswählen	Code-Fenster öffnet sich und zeigt Source Code mit Kommentaren an	Code-Fenster öffnet sich und zeigt Source Code mit Kommentaren an	Erfolgreich
FR-03	info_texts.yaml Datei wird geladen - Inhalt wird im Code-Fenster angezeigt	info_texts.yaml existiert in src/gui	Algorithmus auswählen	Info-Fenster öffnet sich und zeigt textuellen Inhalt an	Info-Fenster öffnet sich und zeigt textuellen Inhalt an	Erfolgreich

Abbildung 3: Testprotokoll Kategorie 03, erstellt mit Google Sheets

Kat.04 - Stabilität/Robustheit

Test-ID	Beschreibung	Vorbedingungen	Auszuführende Schritte	Erwartetes Ergebnis	Tatsächliches Ergebnis	Status
ER-01	Bei "Graph Traversal" Algorithmen kann nur ein Target gleichzeitig existieren (grün eingefärbt)	"Graph Traversal" Algorithmus ausgewählt und Target platziert	Algorithmus starten und während dem Ablauf versuchen ein Target zu platzieren	Platzierung eines weiteren Targets funktioniert nicht	Platzierung eines weiteren Targets funktioniert nicht	Erfolgreich
ER-02	Während ein "Graph Traversal" Algorithmus läuft kann weder ein Target noch ein Obstacle platziert werden	"Graph Traversal" Algorithmus ausgewählt	Algorithmus starten und während dem Ablauf versuchen ein Target zu platzieren	Platzierung des Targets funktioniert nicht	Platzierung des Targets funktioniert nicht	Erfolgreich

Abbildung 4: Testprotokoll Kategorie 04, erstellt mit Google Sheets

3. Automatisierte Tests

Die automatisierten Tests decken folgende Module, aus den “Model” und “Controller” Schichten der MVC-Architektur ab:

- “Algorithms”: Überprüfung der Korrektheit der Algorithmen und ob vordefinierte Exceptions richtig ausgelöst werden.
- “Parameters”: Testen ob Werte korrekt erzeugt werden (Listen & Zielwerte).
- “Board”: Validierung, ob die Gitterstruktur (2D-Liste) korrekt erzeugt und wiederhergestellt wird.
- “FileReader”: Überprüfung der Korrektheit der Lade- und Parse-Vorgänge von YAML-Dateien, sowie der Fehlerbehandlung mit Hilfe von Mocks.
- “AppController”: Testen von unterschiedlichen Logik-Vorgängen wie z.B. der Reaktion auf Benutzeraktionen unter anderem mit Hilfe von Mocks.

Mocks:

In den Tests zu “FileReader” und “AppController” kommen sogenannte “Mocks” zum Einsatz. Diese ermöglichen es, die Module isoliert zu testen, ohne dass Abhängigkeiten zu anderen Klassen oder Ressourcen berücksichtigt werden müssen. Dadurch kann überprüft werden, ob im Modul “AppController” bestimmte Methodenaufrufe stattfinden. Im “FileReader” - Modul werden durch Mocks Dateioperationen getestet, ohne dabei echte Dateien zu verwenden.

Die implementierten automatisierten Tests befinden sich im Repository unter src/tests.