

IU Internationale Hochschule

Weiterbildung: "Software Engineering - Python"

Modul : DLMCSPSE01\_D - Projekt: Software Engineering

Tutor: Prof. Dr. David Kuhlen



## **Prüfungsleistung: Portfolio**

### **Projektdokumentation**

Eingereicht am 16.09.2025.

#### Verfasser:

Djahan Bayrami Latran

Denninger Straße 198

81927 München

E-Mail: [djahan.latran@gmail.com](mailto:djahan.latran@gmail.com)

Matrikelnummer: UPS10672478

## 1. Projektübersicht

Das folgende Projekt "AlgoLab" zielt darauf ab, eine Desktop-Applikation zu entwickeln, über die sich Nutzer/Lernende über die funktionsweise grundlegender Algorithmen informieren und den Ablauf anhand von grafischen Animationen visuell betrachten können.

Ziel des Projekts ist die Entwicklung einer unter Windows lauffähigen Anwendung mit grafischer Benutzeroberfläche, die durch Animationen und Interaktivität die Funktionsweise der Algorithmen darstellt und einfacher begreifbar machen soll.

Den Nutzern soll eine Auswahl an Algorithmen zur Verfügung stehen, sowie die Möglichkeit, deren Ablauf zu beobachten und verschiedene Parameter wie z.B. die Ablaufgeschwindigkeit oder die Größe des Input-Arrays verändern zu können. Zu den verschiedenen Algorithmus-Kategorien soll jeweils eine passende Visualisierungsart gewählt werden, um das Verständnis bestmöglich zu fördern. Die Hauptzielgruppe sind Studenten und Lernende informationstechnischer Studiengänge oder ähnlichem.

## 2. Risikomanagement

Technische Risiken:

**RT1:** Fehlerhafte Algorithmen-Logik

- **Beschreibung:** Ein fehlerhaft implementierter Algorithmus liefert falsche Ergebnisse und verfehlt seinen Zweck
- **Eintrittswahrscheinlichkeit:** Mittel
- **Schadensausmaß:** Sehr hoch
- **Frühwarnindikatoren:** Die visuelle Ausgabe ist falsch. Bspw. ein Sortieralgorithmus platziert den höchsten Balken/größten Wert nicht am Ende der Liste. Frühzeitige Tests schlagen fehl.
- **Präventive Maßnahme:** Frühzeitiges Testing der Algorithmen
- **Reaktive Maßnahme:** Fehleranalyse der Implementierung und Korrektur.

**RT2:** Leistungsinstabilität bei Verarbeitung großer Datensätze

- **Beschreibung:** Die Algorithmen laufen bei großen Eingabemengen zu langsam und die Anwendung erleidet Performanceprobleme
- **Eintrittswahrscheinlichkeit:** Mittel
- **Schadensausmaß:** Hoch

- **Frühwarnindikatoren:** Tests mit großen Datenmengen benötigen eine lange Ausführungszeit.
- **Präventive Maßnahme:** Korrekte Umsetzung der Algorithmus-Logik. Maximale Größe des Eingabe Datensatzes sinnvoll begrenzen. Performante Auswahl der genutzten Datenstrukturen.
- **Reaktive Maßnahme:** Fehleranalyse des Algorithmus-Moduls. Analyse der Zeitkomplexität des Algorithmus-Codes.

#### RT3: Mangelhafte Eingabesicherung

- **Beschreibung:** Absturz der Applikation bei falschen/ungültigen Eingaben
- **Eintrittswahrscheinlichkeit:** Niedrig
- **Schadensausmaß:** Sehr hoch
- **Frühwarnindikatoren:** Absturz der Applikation bei unerwarteten Eingaben
- **Präventive Maßnahme:** Auf sinnvolles Exception-Handling achten. Anstatt direkte Eingabe von Werten durch den User, Auswahl an selektierbaren Optionen auf dem User-Interface.
- **Reaktive Maßnahme:** Hinweise zu falschen Eingaben oder Richtlinien zu den Eingabeparametern.

#### RT4: Hoher Ressourcenverbrauch

- **Beschreibung:** Darstellung der Algorithmen ist ineffizient umgesetzt und verursacht Instabilität der Anwendung durch zu hohen Speicher-/Ressourcenverbrauch.
- **Eintrittswahrscheinlichkeit:** Niedrig
- **Schadensausmaß:** Hoch
- **Frühwarnindikatoren:** Hoher Speicherverbrauch durch unnötige Berechnungen in der Darstellung der Algorithmen
- **Präventive Maßnahme:** Geeignete, effiziente Bibliothek mit passenden Datenstrukturen zur Visualisierung der Algorithmen verwenden. Auf Optimierung in der Umsetzung achten.
- **Reaktive Maßnahme:** Tiefe der Darstellung vereinfachen. Größe der Eingabeparameter verringern. Code optimieren.

#### RT5: Integrationsproblem der animierten UI-Komponente

- **Beschreibung:** Technische Probleme durch Integration des Animationsfensters in das User-Interface. Durch unzureichende Entkopplung der Komponenten werden Fehler oder Probleme im Animationsfenster auf das restliche UI übertragen und führen zu Instabilität der gesamten Anwendung.

- **Eintrittswahrscheinlichkeit:** Hoch
- **Schadensausmaß:** Sehr Hoch
- **Frühwarnindikatoren:** Einfrieren des UI's während die Anwendung läuft und ein Algorithmus abgespielt wird
- **Präventive Maßnahme:** Entkopplung frühzeitig in der Architekturplanung berücksichtigen und kontinuierliches Testen der Integration.
- **Reaktive Maßnahme:** Überarbeitung/Verbesserung der Entkopplung. Notfalls alternative Bibliothek wählen.

#### Management Risiken:

##### **RM1:** Unklare/sich verändernde Anforderungen

- **Beschreibung:** Vielzahl von möglichen Algorithmen und Darstellungen machen das Projekt zu komplex oder verursachen unkontrollierten Wachstum des Projektumfangs.
- **Eintrittswahrscheinlichkeit:** Hoch
- **Schadensausmaß:** Hoch
- **Frühwarnindikatoren:** Mehrmalige Änderung der Anforderungen. Fehlen an klarer Abgrenzung der Aufgaben.
- **Präventive Maßnahme:** Frühzeitige, saubere und gut strukturierte Anforderungsanalyse.
- **Reaktive Maßnahme:** Geplante Änderungen ausführlich dokumentieren und die Auswirkungen auf den Gesamtverlauf auswerten. Notfalls den Änderungswunsch verneinen.

##### **RM2:** Unzureichende Nutzereinbindung/Abstimmung an Zielgruppe

- **Beschreibung:** Unzureichende Einbindung der Nutzergruppe verursacht, dass die Anwendung an den Bedürfnissen vorbei entwickelt wird. Der Nutzen der Applikation wird verfehlt.
- **Eintrittswahrscheinlichkeit:** Gering
- **Schadensausmaß:** Hoch
- **Frühwarnindikatoren:** Kein oder mangelnder Austausch mit dem Tutor
- **Präventive Maßnahme:** Saubere, strukturierte Zwischenabgaben und regelmäßiges Feedback des Tutors einholen.
- **Reaktive Maßnahme:** Gespräch mit dem Tutor organisieren und Anwendung/Projekt gemäß Feedback anpassen.

### **RM3: Unrealistische Zeitplanung oder unvorhersehbare Problematiken**

- **Beschreibung:** Eintreten von unvorhergesehenen technischen Hürden, die Verzögerungen im Zeitplan verursachen. Unterschätzung des zeitlichen Aufwands von Organisation und Entwicklung. Z.B. Probleme mit verwendeten Bibliotheken und deren Kompatibilität.
- **Eintrittswahrscheinlichkeit:** Hoch
- **Schadensausmaß:** Mittel
- **Frühwarnindikatoren:** Abweichungen vom ursprünglichen Zeitplan. Meilensteine werden erst später erreicht, als im Zeitplan festgelegt.
- **Präventive Maßnahme:** Zusätzlichen Handlungsspielraum mit einplanen. Festlegen von kleinteiligen und einfacher erreichbaren Zielen.
- **Reaktive Maßnahme:** Anpassungen am Projektzeitplan. Auf die wesentlichen/wichtigeren Funktionen fokussieren. Ggfs. auf weniger Wichtige Teilaspekte verzichten. Verlängerung der Weiterbildung beantragen.

### **RM4: Zeitmangel durch Beruf**

- **Beschreibung:** Zur Verfügung stehende Zeit für das Projekt schwankt stark, da es nebenberuflich erarbeitet wird. Stoßzeiten im Beruf führen zu Verschiebungen im Zeitplan und verursachen eine verspätete Abgabe als ursprünglich geplant.
- **Eintrittswahrscheinlichkeit:** Sehr hoch
- **Schadensausmaß:** Niedrig-Mittel
- **Frühwarnindikatoren:** Die geplante wöchentliche Zeit zur Bearbeitung des Projektes wird regelmäßig unterschritten.
- **Präventive Maßnahme:** Minimale Bearbeitungszeit festlegen, die jede Woche erreicht werden soll. Zeit kann flexibel über die Woche verteilt werden, solange das Ziel am Ende der Woche erreicht wird.
- **Reaktive Maßnahme:** Aufholen der Rückstände an Wochenenden. Notfalls Beantragung einer Verlängerung der Bearbeitungszeit.

### 3. Link zum Github-Repository

[https://github.com/djahan-latan/algorithm\\_visualizer](https://github.com/djahan-latan/algorithm_visualizer)

### 4. Zeitplan & Arbeitspakete

#### Aktualisierter Zeitplan:

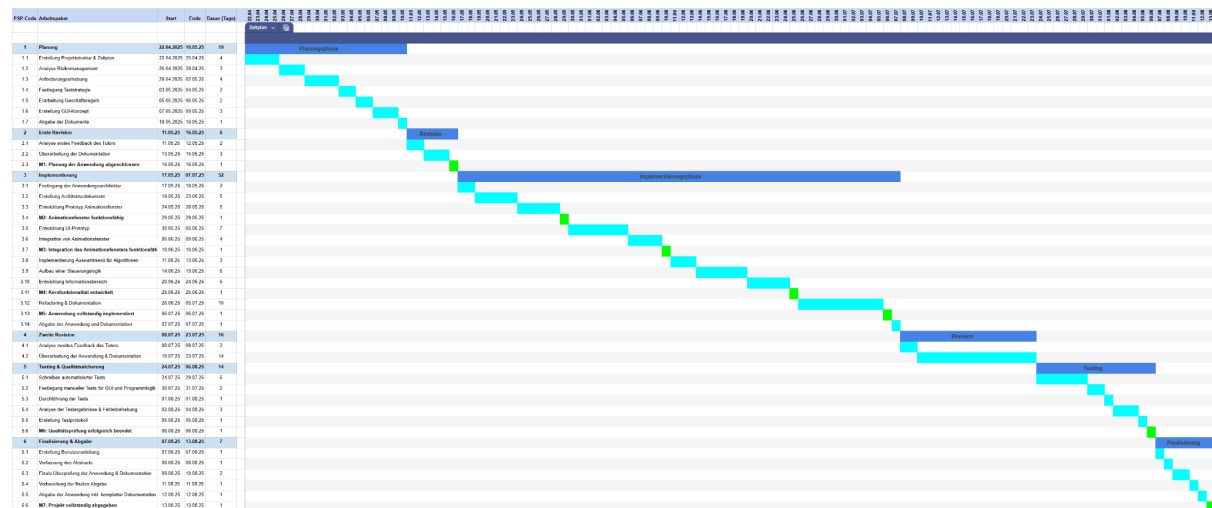


Abbildung 1: Zeitplan des Projektablaufs, erstellt mit Google Sheets.

## Arbeitspakete mit PSP-Codes und Meilensteinen (M1-M7):

PSP-Code	Phase	Arbeitspaket	Beschreibung	Dauer (Tage)	Start	Ende	Meilenstein
<b>1</b>	<b>Planung</b>			<b>19</b>	<b>22.04.2025</b>	<b>10.05.25</b>	
1.1		Erstellung Projektstruktur & Zeitplan	Erstellung eines Projektplans inkl. Zeitplan und Arbeitspaketetabelle zur Entwicklung der Anwendung	4	22.04.2025	25.04.25	
1.2		Analyse Risikomanagement	Analyse welche Risiken während der Entwicklung entstehen könnten, sowie Einschätzung des Auftretens und des Schadens, um das Auftreten möglichst zu verhindern und den Schaden zu minimieren	3	26.04.2025	28.04.25	
1.3		Anforderungserhebung	Erhebung der funktionalen und nichtfunktionalen Anforderungen an die Applikation auf Basis der Projektidee	4	29.04.2025	02.05.25	
1.4		Festlegung Teststrategie	Erarbeitung einer passenden Teststrategie der Applikation gemäß den zuvor erhobenen Anforderungen zur besseren Qualitätssicherung	2	03.05.2025	04.05.25	
1.5		Erarbeitung Geschäftsregeln	Analyse und Beschreibung der Prozesse und Regeln zum Verhalten der Anwendung. Dazu zählt, wie ein Algorithmus abläuft, wie dieser gesteuert werden kann und wie die Benutzerinteraktionen von der Applikation verarbeitet werden	2	05.05.2025	06.05.25	
1.6		Erstellung GUI-Konzept	Erarbeitung eines visuellen GUI-Konzepts und Darstellung als 2D-Bildcollage, um frühzeitig das GUI Layout zu veranschaulichen	3	07.05.2025	09.05.25	
1.7		Abgabe der Dokumente	Einreichen der bisherigen Dokumente aus der Planungsphase, um Feedback des Tutors einzuholen. Dies umfasst das Spezifikationsdokument, das Anforderungsdokument und die Projektdokumentation	1	10.05.2025	10.05.25	
<b>2</b>	<b>Erste Revision</b>			<b>6</b>	<b>11.05.25</b>	<b>16.05.25</b>	
2.1		Analyse erstes Feedback des Tutors	Auswertung des Tutor-Feedbacks zur Planungsphase und dadurch Ableitung notwendiger Korrekturen	2	11.05.25	12.05.25	
2.2		Überarbeitung der Dokumentation	Überarbeitung des Projektplans und entsprechenden Dokumenten aus der Planungsphase gemäß Feedback des Tutors	3	13.05.25	15.05.25	
2.3		<b>Planung der Anwendung abgeschlossen</b>	<b>Die Planung der Anwendung und deren Entwicklung ist abgeschlossen. Es kann nun in die Implementierungsphase übergegangen werden</b>	1	16.05.25	16.05.25	<b>M1</b>
<b>3</b>	<b>Implementierung</b>			<b>52</b>	<b>17.05.25</b>	<b>07.07.25</b>	
3.1		Festlegung der Anwendungsarchitektur	Festlegung einer passenden Systemarchitektur gemäß den Anforderungen aus der Planungsphase	2	17.05.25	18.05.25	
3.2		Erstellung Architekturdokument	Erstellung des Architekturdokuments und Dokumentation der Architektur mit UML Diagrammen. Hierzu zählt ein Klassendiagramm, sowie ein Sequenzdiagramm, das einen zentralen Prozess der Anwendung beschreibt	5	19.05.25	23.05.25	
3.3		Entwicklung Prototyp Animationsfenster	Entwicklung eines Prototypen zur visuellen Darstellung der Algorithmen. Sortier- und Suchalgorithmen werden mittels animierter Balkendiagramme (Balken entsprechen Listenwerten) dargestellt und Graph-Traversierungs-Algorithmen mit Gitternetzen (2D Array als Graph).	5	24.05.25	28.05.25	
3.4		<b>Animationsfenster funktionsfähig</b>	<b>Das Animationsfenster wurde erfolgreich implementiert. Das Verhalten der Algorithmen und die Visualisierung funktioniert wie vorgesehen. Das Fenster kann nun in ein User-Interface integriert werden</b>	1	29.05.25	29.05.25	<b>M2</b>
3.5		Entwicklung UI-Prototyp	Entwicklung eines Prototypen auf Basis des GUI-Entwurfs aus der Planungsphase. Der Fokus liegt dabei auf dem Layout, der Gestaltung und Platzierung der Steuerelemente. Die Funktionalität dieser wird in diesem Abschnitt noch nicht entwickelt	7	30.05.25	05.06.25	
3.6		Integration von Animationsfenster	Technische Integration des Animationsfensters in das User-Interface. Verknüpfung mit der UI- und Steuerlogik. Sicherstellung dass die Darstellung performant und flüssig läuft. Entkopplung der Bildwiederholrate von UI-Steuerung und Animationsfenster.	4	06.06.25	09.06.25	
3.7		<b>Integration des Animationsfensters funktionsfähig</b>	<b>Vollständige funktionsfähige Integration des Animationsfensters in das UI erreicht</b>	1	10.06.25	10.06.25	<b>M3</b>
3.8		Implementierung Auswahlmenü für Algorithmen	Erweiterung des UI-Prototyps. Das Auswahlmenü der Algorithmen wird um die zusätzlichen Such-/Sortier- und Graphtraversierungs-Algorithmen ergänzt. Hierbei handelt es sich konkret um die Algorithmen "Binary Search", "Selection Sort", "Insertion Sort", "Depth-First-Search" und "Dijkstra's Algorithm"	3	11.06.25	13.06.25	
3.9		Aufbau einer Steuerungslogik	Erweiterung der aus 3.5 entwickelten Buttons zur Steuerung der Algorithmen. Entwickelt wird die Funktion Algorithmen zu Starten, Pausieren und Zurücksetzen	6	14.06.25	19.06.25	
3.10		Entwicklung Informationsbereich	Entwicklung des Info-Bereichs. Der Bereich dient als zusätzlicher Informationsbereich für den User. Hier werden Informationen über den gewählten Algorithmus in Form von Text dargestellt. Zusätzlich wird der Source Code des Algorithmus mit Kommentaren angezeigt, damit dieser besser verständlich ist. Der Infobereich öffnet sich, nachdem ein Algorithmus gewählt wurde und wird aktualisiert sobald ein anderer Algorithmus gewählt wird	5	20.06.25	24.06.25	
3.11		<b>Kernfunktionalität entwickelt</b>	<b>Vollständige Kernfunktionalität der Applikation wurde entwickelt. Eine Code-Säuberung, sowie Kommentierung und Testing stehen noch aus. Die Applikation kann jedoch ausgeführt werden. Alle geplanten Algorithmen wurden integriert und können visualisiert und untersucht werden</b>	1	25.06.25	25.06.25	<b>M4</b>
3.12		Refactoring & Dokumentation	Säuberung und Kommentierung des Programm Codes. Der Code soll für externe Betrachter sinnvoll und nachvollziehbar kommentiert werden. Sich wiederholender Code wird in Funktionen/Methoden umgeschrieben. Unnötiger Code für die korrekte Ausführung der Applikation wird entfernt.	10	26.06.25	05.07.25	
3.13		<b>Anwendung vollständig implementiert</b>	<b>Vollständige funktionsfähige Applikation erreicht</b>	1	06.07.25	06.07.25	<b>M5</b>
3.14		Abgabe der Anwendung und Dokumentation	Abgabe der überarbeiteten Dokumente aus der Planungsphase mit Github-Link zum Repository, sowie des neu hinzugekommenen Architekturdokuments, um Feedback des Tutors einzuholen	1	07.07.25	07.07.25	
<b>4</b>	<b>Zweite Revision</b>			<b>16</b>	<b>08.07.25</b>	<b>23.07.25</b>	
4.1		Analyse zweites Feedback des Tutors	Auswertung des Tutor-Feedbacks zur Implementierungsphase und dadurch Ableitung von notwendigen Korrekturen	2	08.07.25	09.07.25	
4.2		Überarbeitung der Anwendung & Dokumentation	Überarbeitung der Arbeitspakete und entsprechenden Dokumenten aus der Implementierungsphase gemäß Feedback des Tutors	14	10.07.25	23.07.25	
<b>5</b>	<b>Testing &amp; Qualitätssicherung</b>			<b>14</b>	<b>24.07.25</b>	<b>06.08.25</b>	
5.1		Schreiben automatisierter Tests	Entwickeln von automatisierten Tests. Implementierung von Unit-Tests zum automatisierten Testen wichtiger Komponenten	6	24.07.25	29.07.25	
5.2		Festlegung manueller Tests für GUI und Programmlogik	Festlegung von Testscenarien um manuelle Prüfungen der Benutzeroberfläche und Steuerung auszuführen	2	30.07.25	31.07.25	
5.3		Durchführung der Tests	Ausführung aller Tests aus den beiden Arbeitspaketen 5.1 und 5.2 um festzustellen, ob die Anwendung fehlerfrei funktioniert	1	01.08.25	01.08.25	
5.4		Analyse der Testergebnisse & Fehlerbehebung	Auswertung der Testergebnisse und ggfs. Behebung der Mängel	3	02.08.25	04.08.25	
5.5		Erstellung Testprotokoll	Erstellen des Testprotokolls und Dokumentation der Testergebnisse	1	05.08.25	05.08.25	
5.6		<b>Qualitätsprüfung erfolgreich beendet</b>	<b>Abschluss der Testphase mit vollständiger und korrekt funktionsfähiger Applikation erreicht</b>	1	06.08.25	06.08.25	<b>M6</b>
<b>6</b>	<b>Finalisierung &amp; Abgabe</b>			<b>7</b>	<b>07.08.25</b>	<b>13.08.25</b>	
6.1		Erstellung Benutzeranleitung	Verfassen einer Anleitung für die Benutzung der Anwendung. Diese sollte leicht verständlich sein für Laien.	1	07.08.25	07.08.25	
6.2		Verfassung des Abstracts	Verfassen eines zweiseitigen (Din A4) Abstracts des Projekts, in dem die Umsetzung kompakt zusammengefasst wird und der Prozess kritisch reflektiert wird	1	08.08.25	08.08.25	
6.3		Finale Überprüfung der Anwendung & Dokumentation	Letzte Überprüfung der Vollständigkeit der Dokumentation und Funktionsfähigkeit der Anwendung, um die Vorbereitung der Abgabe starten zu können	2	09.08.25	10.08.25	
6.4		Vorbereitung der finalen Abgabe	Zusammenstellung aller bearbeiteten und durch Aufgabenstellung geforderten Dateien und Dokumenten in finaler Form	1	11.08.25	11.08.25	
6.5		Abgabe der Anwendung inkl. kompletter Dokumentation	Finale Abgabe mit vollständiger Projektdokumentation, Abstract und Benutzeranleitung. Link zum Github Repository der Anwendung mit Quellcode befindet sich in der Dokumentation	1	12.08.25	12.08.25	
6.6		<b>Projekt vollständig abgegeben</b>	<b>Vollständiger Abschluss des Projekts. Alle Dateien wurden per PebblePad eingereicht.</b>	1	13.08.25	13.08.25	<b>M7</b>

Abbildung 2: Arbeitspakete und Meilensteine, erstellt mit Google Sheets.

## 5. Benutzeranleitung

Folgende Anleitung erklärt, wie man das Programm lokal zum Laufen bringt.

### Voraussetzungen

- Python 3.10 oder höher
- pip (Python Package Manager)

### Installation

- Kclone das Repository:  
`git clone https://github.com/djahan-latran/algorithm_visualizer.git`
- Navigiere zum Projektverzeichnis:  
`cd algorithm_visualizer`
- Installiere die Abhängigkeiten/Packages:  
`pip install -r requirements.txt`

### Ausführung der Anwendung

- Führe im Verzeichnis, wo sich main.py befindet den folgenden Befehl aus:  
`python main.py`
- Die Anwendung sollte nun im Pygame-Fenster starten

### *Hinweis:*

Alternativ kann vom Endnutzer auch die mitgelieferte .exe-Datei "AlgoLab" ausgeführt werden (auch im Repository unter algorithm\_visualizer/dist auffindbar).



## Nutzung

Über das Hauptmenü kann ein Algorithmus ausgewählt werden:



Abbildung 3: Screenshot aus dem Hauptmenü von "AlgoLab".

Anschließend öffnen sich weitere Fenster.

Oben rechts können, je nach Kategorie, unterschiedliche Parameter individualisiert werden.

Die Ablaufgeschwindigkeit kann auch während des Ablaufs des Algorithmus verändert werden.

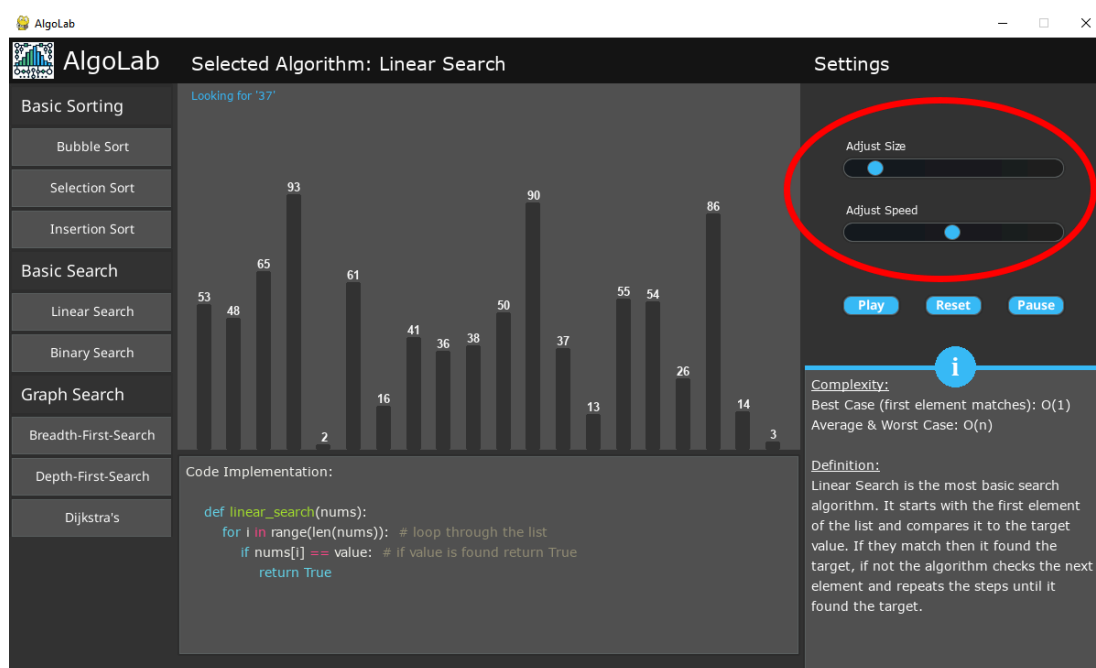


Abbildung 4: Screenshot aus dem Linear-Search-Fenster von "AlgoLab".

Unter den einstellbaren Parameter befindet sich die Steuerung.

Wenn wie im unteren Beispiel der Such-Wert von Linear Search gefunden wurde, kann durch Reset und anschließend Play der Vorgang erneut mit neuen Werten simuliert werden.

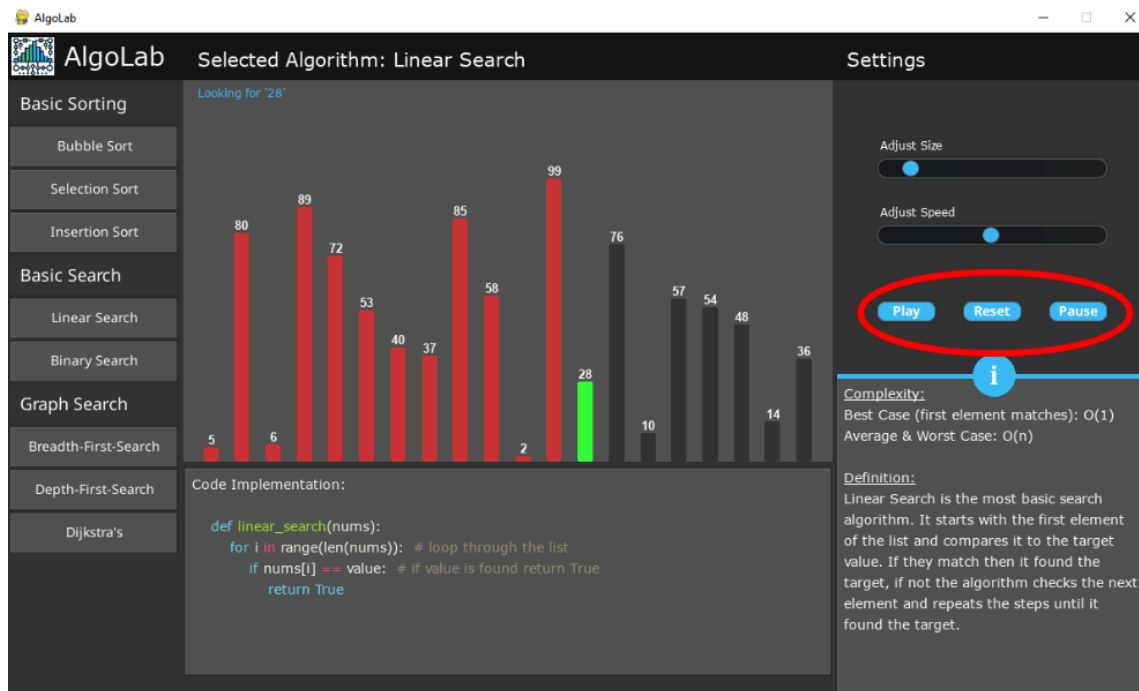


Abbildung 5: Screenshot aus dem Linear-Search-Fenster von “AlgoLab” nach erfolgreicher Suche.

Ein Sortier-Algorithmus ist dann beendet, wenn alle Werte grün erscheinen, wie folgend abgebildet.

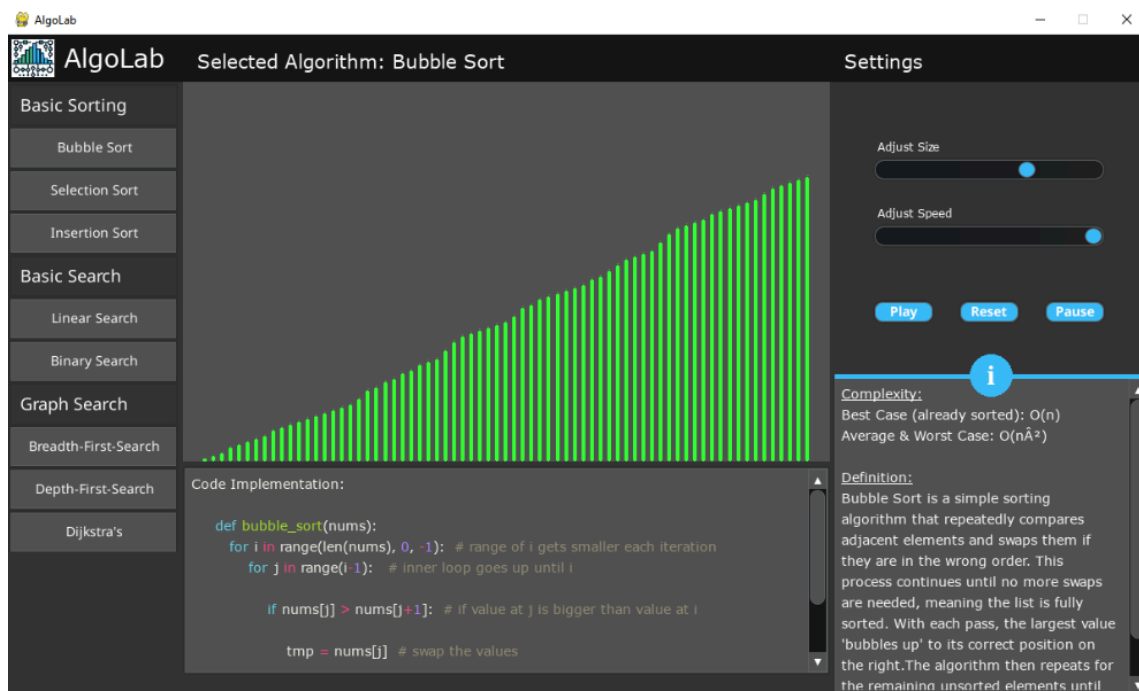


Abbildung 6: Screenshot aus dem Bubble-Sort-Fenster von “AlgoLab” nach erfolgreichem Sortiervorgang.

Bei Graphenalgorithmen kann durch Klicken der “Set target” Schaltfläche und anschließendem Klicken auf der Gitternetzstruktur ein Ziel platziert werden.

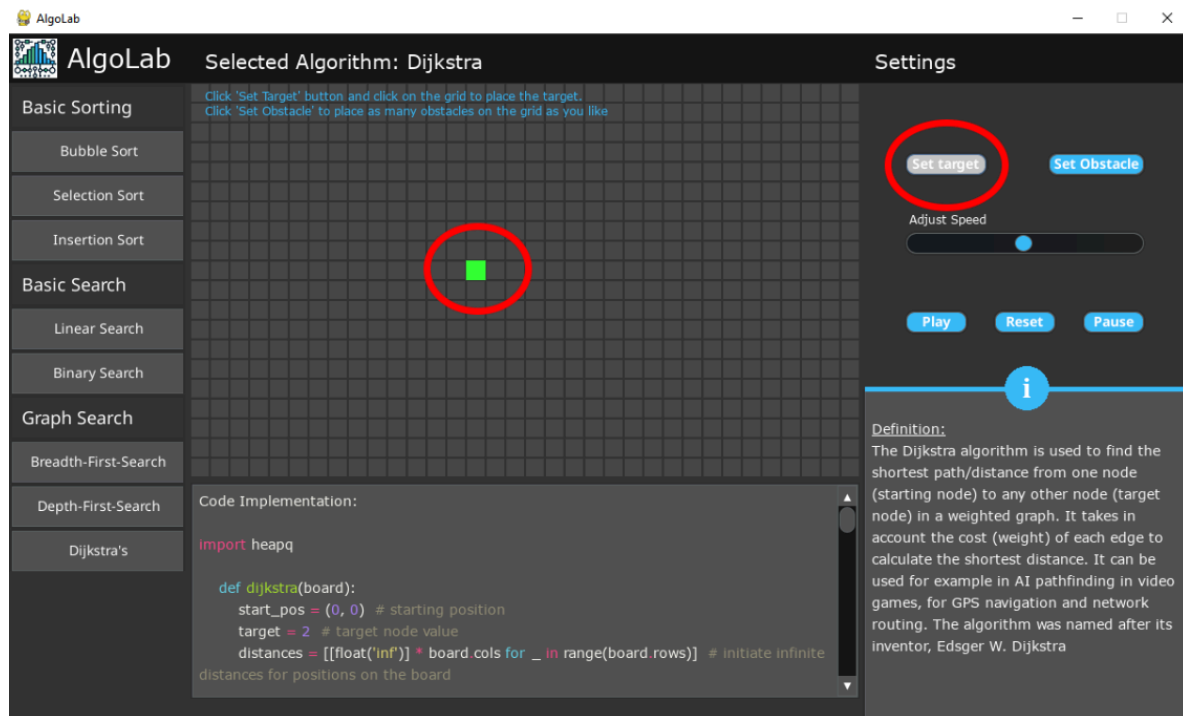


Abbildung 7: Screenshot aus dem Dijkstra-Fenster von “AlgoLab” bei Zielauswahl.

Anschließend können durch den gleichen Vorgang mit der “Set-Obstacle” Schaltfläche beliebig viele Hindernisse platziert werden.

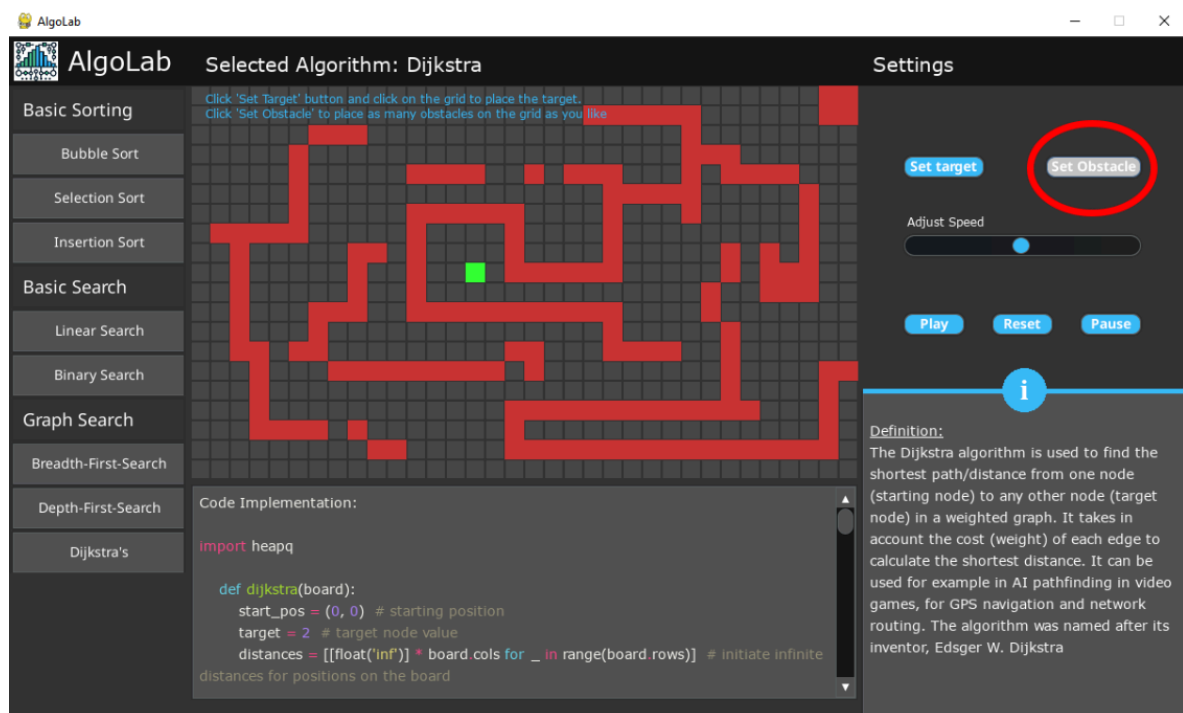


Abbildung 8: Screenshot aus dem Dijkstra-Fenster von “AlgoLab” nach Hindernis Auswahl.

Die Graphenalgorithmen stoppen die Suche, sobald sie den gesuchten Wert gefunden und registriert haben. Speziell im Falle des Dijkstra-Algorithmus wird zusätzlich noch der kürzeste Pfad zum Ziel durch das Gitternetz in hellblauer Farbe angezeigt:

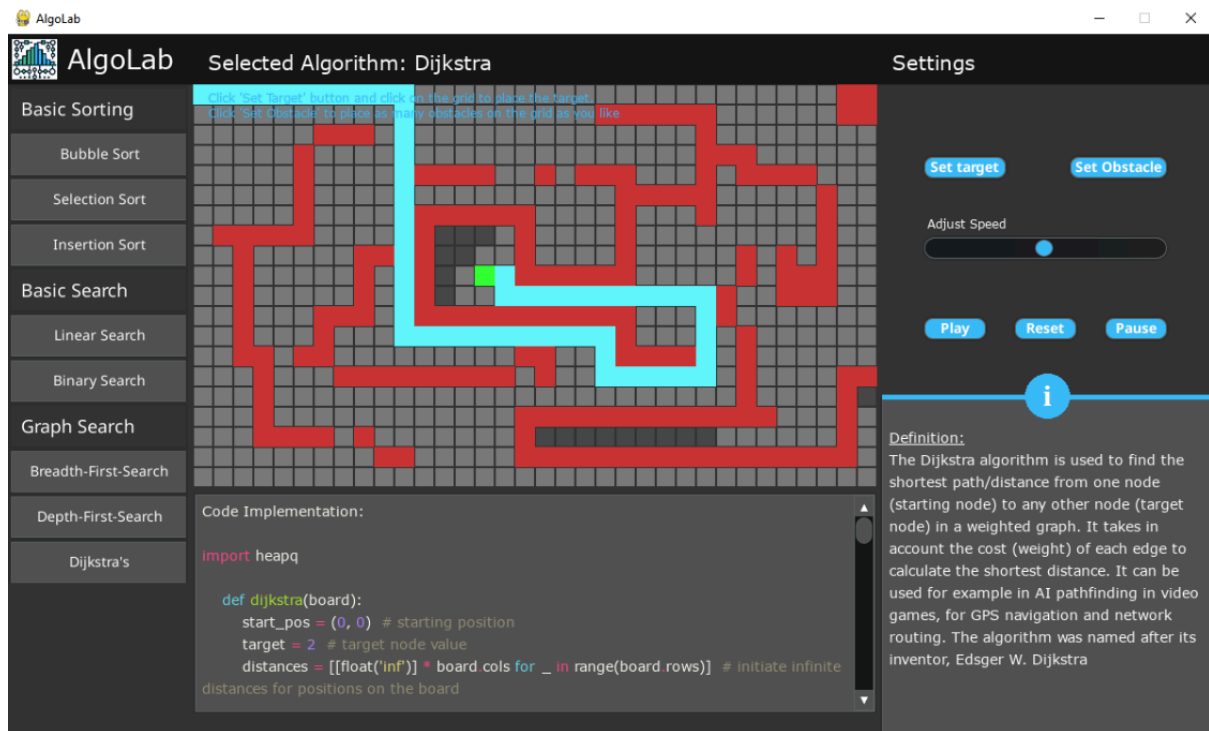


Abbildung 8: Screenshot aus dem Dijkstra-Fenster von “AlgoLab” nach erfolgreicher Suche des Pfads.

Durch Klicken auf das Kreuz im oberen rechten Eck kann die Anwendung jederzeit geschlossen werden.

## 6. Verwendete Bibliotheken

- **Pygame Community Edition Team:**  
Pygame-ce - Community Fork der Pygame-Bibliothek. Version 2.5.5. URL: <https://github.com/pygame-community/pygame-ce>
- **Myre, D. et al.:** Pygame\_gui - GUI Bibliothek für Pygame. Version 0.6.13. URL: <https://pygame-gui.readthedocs.io/>
- **Pygments Project:** Pygments - Syntax Highlighter für Python. Version 2.19.1. URL: <https://pygments.org/>
- **Kirill Simonov et al.:** PyYAML - YAML Parser für Python. Version 6.0.2. URL: <https://pyyaml.org/>