Ein Bild, das Grafiken, Kreis, Symbol, Schrift enthält.

Automatisch generierte Beschreibung

Facharbeit

im Leistungskurs Informatik

Schuljahr 2023/2024

Modellierung und Implementation eines eigenen Programms zur Zusammensetzung einer grafischen Oberfläche (GUI)

Verfasser: Peer Prigge

Kursleiterin: Frau Dr. Stehe

Inhaltsverzeichnis

[1. Einleitung 2](#_Toc160773013)

[Hintergrund 2](#_Toc160773014)

[Motivation 2](#_Toc160773015)

[Zielsetzung 2](#_Toc160773016)

[Nützlichkeit 3](#_Toc160773017)

[Anwendungsfälle 3](#_Toc160773018)

[2. Hauptteil 4](#_Toc160773019)

[Anforderungen und Möglichkeiten 4](#_Toc160773020)

[Umsetzung 5](#_Toc160773021)

[Begrenztheit und Hindernisse 6](#_Toc160773022)

[Schwierigkeiten und Probleme 6](#_Toc160773023)

[Unterschiedliche Ansätze anhand von Problemen 7](#_Toc160773024)

[3. Ende 8](#_Toc160773025)

[Zusammenfassung der Ergebnisse 8](#_Toc160773026)

[Bedeutung für die Zukunft inklusive Ausblick 8](#_Toc160773027)

[Literatur- und Quellenverzeichnis 9](#_Toc160773028)

[Literaturverzeichnis 9](#_Toc160773029)

[Quellenverzeichnis 9](#_Toc160773030)

# Einleitung

## Hintergrund

Die Entwicklung von Softwareanwendungen mit einer benutzerfreundlichen grafischen Benutzeroberfläche (GUI) ist ein wesentlicher Aspekt in der Informatik und wird in allen Bereichen, die einem im Alltag begegnen, verwendet. Eine effektive GUI ermöglicht es Benutzern, auf einfache Weise mit der Anwendung zu interagieren, ihre Funktionen zu nutzen oder zu bearbeiten. In diesem Zusammenhang hat sich das Konzept des Drag & Drop (auf Deutsch: Ziehen & Loslassen) in der Entwicklung von GUIs als äußerst nützlich erwiesen.

Diese Technik ermöglicht es Benutzern, Elemente innerhalb der Benutzeroberfläche zu verschieben und anzuordnen, indem sie sie einfach mit der Maus ziehen und ablegen. Durch diese Nutzerinteraktion können wesentliche Bestandteile eines GUIs erstellt oder verändert werden.

## Motivation

Ein GUI erfordert häufig die manuelle Codierung von Benutzeroberflächenelementen wie Buttons, Textfeldern, Slider und Schalter. Dieser Prozess ist sehr zeitaufwändig und erforderte umfangreiche Kenntnisse in der Programmierung, wobei es jedoch oftmals zu Fehlern kommen kann, welche nicht offensichtlich sind und sich erst in der Programmausführung zeigen. Mit dem Aufkommen von Drag & Drop-Editoren ist es jedoch möglich, GUIs visuell zu gestalten, indem man Elemente aus einer Bibliothek auswählt und sie an die gewünschten Positionen zieht. Dies vereinfach und beschleunigt die GUI-Entwicklung stark.

## Zielsetzung

Das Hauptziel meines Programms besteht darin, eine benutzerfreundliche Drag & Drop-Umgebung bereitzustellen, in der Benutzer GUIs für ihre Anwendungszwecke erstellen können. Das Programm soll einige vorgefertigten GUI-Elementen wie Buttons, Textfelder, Schieberegler und Schalter anbieten, aus denen Benutzer die benötigten Elemente auswählen und durch einfaches Ziehen und Ablegen dieser Elemente in ihr eigenes GUI einbinden können. Außerdem ist es Programmierern möglich, die vorhandenen Elemente durch weitere zu ergänzen, da die Grundstruktur vorliegt.

## Nützlichkeit

Die Nutzung eines solchen Programms kann für verschiedene Personengruppen von Vorteil sein. Zum einen können Entwickler von Computerspielen, insbesondere solche ohne umfangreiche Programmierkenntnisse, von der einfachen Erstellung von GUIs profitieren. Zum anderen können auch Hobby-Programmierer, darunter auch Personen ohne Programmierkenntnisse, die ihre eigenen kleinen GUIs erstellen möchten, von diesem Programm Gebrauch machen und Prototypen ihres GUIs bauen.

## Anwendungsfälle

Die Anwendungsfälle für dieses Programm sind vielfältig. Einige Beispiele umfassen die Erstellung von GUIs für Spielemenüs, Einstellungsoptionen, Charakterinventare, Level-Editor-Tools und vieles mehr. Das Programm ermöglicht es den Benutzern, die visuelle Gestaltung der Benutzeroberfläche an ihre spezifischen Anforderungen anzupassen und gleichzeitig eine einfache Interaktion mit den Programmelementen zu gewährleisten.

Mit diesem Programm zur Drag & Drop-Erstellung von GUIs können Entwickler und Hobby-Programmierer ansprechende Benutzeroberflächen für Anwendungen erstellen, ohne dabei auf umfangreiche Programmierkenntnisse angewiesen zu sein.

Ein Bild, das Text, Screenshot, Design enthält.

Automatisch generierte Beschreibung

Quelle: <https://dribbble.com/tags/game-menu>

# Hauptteil

## Anforderungen und Möglichkeiten

Der Hauptteil meiner Facharbeit befasst sich mit der Entwicklung eines eigenen Programms zur Drag & Drop-Erstellung einer grafischen Benutzeroberfläche (GUI) für einfache Anwendungen.

Eine der wichtigsten Anforderungen an das Programm ist eine intuitive Benutzeroberfläche. Ein solches Programm sollte den Nutzern eine benutzerfreundliche und leicht verständliche Oberfläche bieten. Die Nutzer sollten in der Lage sein, die GUI-Elemente einfach durch Drag & Drop auszuwählen, zu ziehen und abzulegen, um ihre gewünschten Elemente am Ende als groben Code Aufbau zu erhalten.

Ein weiteres wichtiges Merkmal des Programms ist die Auswahl der am häufigsten GUI-Elementen. Es sollte eine Palette von Elementen geben, die für alle möglichen Szenarien geeignet sind. Dazu gehören Buttons, Textfelder, Schalter und andere interaktive Elemente. Die Nutzer sollten in der Lage sein, diese Elemente in ihrer Benutzeroberfläche zu platzieren, um somit die relevanten Codestücke zu erhalten. Die am häufigsten genutzten GUI-Elementen ermöglichen es den Nutzern, ihre Benutzeroberfläche individuell anzupassen und an die spezifischen Anforderungen ihrer Idee anzupassen.

Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibung

Quelle: Eigenes Programm

Desweiterem steht es dem Nutzer zur Verfügung die Codestücke, welche als Standard zur Verfügung stehen komplett zu bearbeiten. Durch diese Möglichkeit hat der Nutzer die Einschränkung das Programm nur für eine Programmiersprache zu nutzen überwunden. Man kann selbst entscheiden, was für Funktionen in den Standardelementen vorhanden sind und wie sie sich verhalten.

Ein Bild, das Text, Screenshot, Software, Design enthält.

Automatisch generierte Beschreibung

Quelle: Eigenes Programm

## Umsetzung

Die Umsetzung eines solchen Programms erfordert den Einsatz geeigneter Programmiersprachen und Frameworks. In der Informatik gibt es verschiedene Ansätze und Technologien, die für die Entwicklung von GUIs und Drag & Drop-Funktionalitäten verwendet werden können.

Eine Möglichkeit besteht darin, Programmiersprachen wie C++, Java oder Python in Kombination mit GUI-Bibliotheken wie Qt, JavaFX, Tkinter oder Customtkinter zu verwenden. Diese Bibliotheken bieten Funktionen zur Erstellung von GUI-Elementen und unterstützen Drag & Drop-Operationen. Sie ermöglichen es den Entwicklern, die gewünschten Funktionen in das Programm zu integrieren und eine robuste und leistungsfähige Benutzeroberfläche zu erstellen.

Ein alternativer Ansatz besteht darin, webbasierte Technologien wie HTML, CSS und JavaScript zu verwenden. In Kombination mit JavaScript-Bibliotheken oder Frameworks wie React, Angular oder Vue.js kann man damit plattformübergreifende GUIs erstellen, die in einem Webbrowser ausgeführt werden können. Dies ermöglicht eine größere Flexibilität bei der Bereitstellung und Nutzung des Programms.

## Begrenztheit und Hindernisse

Im Verlauf der Programmierung sind einige Hindernisse aufgetreten, die die schnelle Entwicklung des Programms erschwert haben. Insbesondere stellten sich bestimmte Funktionen im Laufe der Programmierung als umfangreicher und seltener bearbeitet heraus.   
Beispielsweise gab es keine vorhandenen Ansätze im Internet für Funktionen wie das Aktivieren eines Buttons erst ab einer bestimmten Position oder das Klickbar-Machen von Seitenleisten-Frames basierend auf Benutzeraktionen.

Aufgrund dieser Schwierigkeiten habe ich mich dazu entschieden, von meiner ursprünglichen Idee, das GUI mit Tkinter zu programmieren, abzuweichen und auf Customtkinter umzusteigen.

Customtkinter ist eine moderne Python-Bibliothek, die regelmäßig aktualisiert wird und mir ermöglicht, das Erscheinungsbild meines GUIs und seiner Elemente anzupassen. Obwohl es in der Customtkinter-Dokumentation möglicherweise nicht alle erforderlichen Informationen gab, konnte ich mithilfe der Tkinter-Dokumentation und Online-Foren eine Lösung finden. Dieser Ansatz eröffnete mir neue Möglichkeiten, da ich die Gestaltung und das Verhalten der GUI-Elemente besser kontrollieren konnte. Indem ich auf Customtkinter umgestiegen bin, konnte ich die gewünschten Funktionen implementieren und das Benutzererlebnis verbessern.

Trotz der Herausforderungen während der Programmierung war es wichtig, alternative Lösungen zu finden und auf bestehende Ressourcen zurückzugreifen, um das gewünschte Ergebnis zu erzielen.

Durch die Kombination von verschiedenen Quellen und dem Einsatz von Customtkinter konnte ich letztendlich die erforderlichen Anpassungen vornehmen und das GUI gemäß meinen Vorstellungen gestalten.

## Schwierigkeiten und Probleme

Während der Programmierung traten verschiedene Schwierigkeiten und Probleme auf, die den Fortschritt des Projekts beeinträchtigten.

Eine der Hauptprobleme war die Umsetzung einiger komplexer Funktionen, die sich im Laufe der Programmierung als anspruchsvoller erwiesen als zunächst angenommen.   
Ein Beispiel dafür war die Schwierigkeit, eine Funktion zu implementieren, die es ermöglicht, einen Button erst ab einer bestimmten Position auf dem Bildschirm anzuzeigen. Dies erforderte eine Positionsberechnung und die Anpassung des GUI-Layouts. Ein weiteres Hindernis bestand darin, Seitenleisten-Frames basierend auf bestimmten Benutzeraktionen klickbar zu machen. Auch hier gab es keine vorgefertigten Lösungen oder ausführliche Beispiele in der verfügbaren Literatur. Um dieses Problem zu lösen, habe ich meine Kenntnisse über die Tkinter-Bibliothek erweitert und spezifische Ereignisbehandlungen implementiert. Dies erforderte experimentelles Vorgehen und das Testen verschiedener Ansätze, um die gewünschte Funktionalität zu erreichen.   
Ein weiteres Problem war die begrenzte Dokumentation und Verfügbarkeit von Informationen über die verwendete Bibliothek Customtkinter.

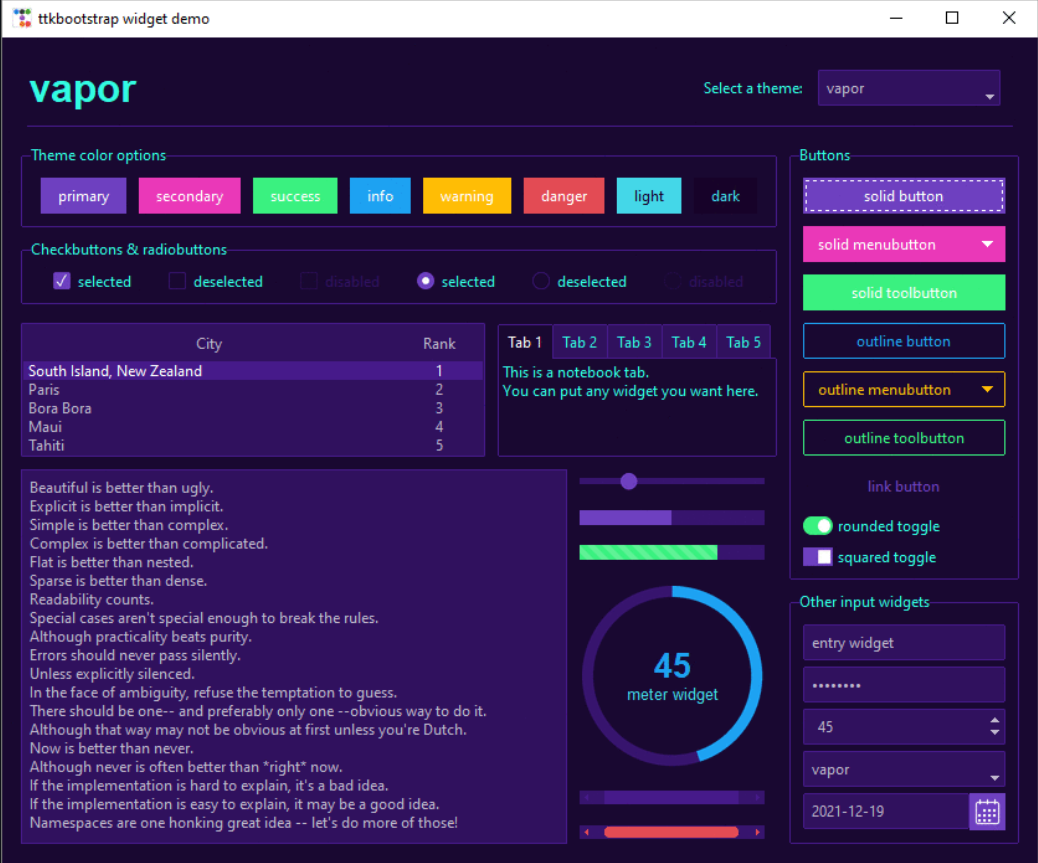
Obwohl Customtkinter eine vielversprechende und moderne Bibliothek ist, stellte ich fest, dass einige spezifische Funktionen oder Anpassungsmöglichkeiten nicht ausführlich dokumentiert waren. In solchen Fällen musste ich auf mein Verständnis von Tkinter zurückgreifen und mit Hilfe von Online-Foren nach Lösungen suchen. Dies erforderte Kreativität und die Fähigkeit, vorhandene Ressourcen optimal zu nutzen und nach Bedürfnissen passend auf mein Programm abzuändern.

Durch eine sorgfältige Planung, gründliche Recherche und die Bereitschaft, neue Ansätze auszuprobieren, konnte ich letztendlich den Großteil der gewünschten Funktionen implementieren und das GUI erfolgreich entwickeln.

## Unterschiedliche Ansätze anhand von Problemen

Es gab im Verlauf der Entwicklung einige Lösungsansätze, welche ich aufgrund von Problemen abändern musste. Zum Beispiel war mein eigentlicher Ansatz zur Entwicklung des GUIs die Nutzung der Python Bibliothek Tkinter. Diesen Versuch brach ich jedoch nach einigem Testen ab, da einige Funktionen nicht verfügbar waren.

Aus diesem Grund fiel meine zweite Wahl auf die GUI Bibliothek Customtkinter, welche mich bei dem erstmaligem testen sehr überraschte. Es gab wesentlich mehr Funktionen und eine auf dem Stand gehaltene Dokumentation. Desweiterem waren die Elemente in einem moderneren Stil gehalten, was an der Aktualität liegt.

1. 2.Ein Bild, das Text, Software, Computersymbol, Webseite enthält.

Automatisch generierte Beschreibung

Quelle:   
1. <https://raw.githubusercontent.com/israel-dryer/ttkbootstrap/master/docs/assets/themes/themes.gif>  
2. <https://github.com/TomSchimansky/CustomTkinter/blob/master/documentation_images/complex_example_light_macOS.png>

# Ende

## Zusammenfassung der Ergebnisse

Als Resultat durch meine Entscheidungen, darunter die Entscheidung, die GUI-Bibliothek Customtkinter anstelle von Tkinter zu verwenden, konnte ich eine moderne Benutzeroberfläche erstellen, die den Anforderungen und Erwartungen von mir entspricht. Durch die Implementierung verschiedener Funktionen und die Anpassung der GUI-Elemente konnte ich eine benutzerfreundliche und effiziente Drag & Drop-Umgebung schaffen. Die Flexibilität von Customtkinter ermöglichte es mir, das Erscheinungsbild und das Verhalten der GUI-Elemente zu optimieren und somit ein nahtloses Benutzererlebnis zu gewährleisten.

## Bedeutung für die Zukunft inklusive Ausblick

Die Bedeutung für die Zukunft meines Programms liegt in der kontinuierlichen Verbesserung und Anpassung an die sich ändernden Anforderungen und Technologien.   
Durch die Integration von neuen GUI-Elementen, Funktionen und Technologien wie Machine Learning können zukünftige Versionen des Programms noch leistungsfähiger und benutzerfreundlicher werden. Darüber hinaus könnte das Programm auch in verschiedenen Branchen und Anwendungsbereichen eingesetzt werden, wie beispielsweise im Bildungsbereich, um Schülern und Studenten eine einfache und intuitive Möglichkeit zu bieten, ihre eigenen GUI-Anwendungen zu erstellen und zu entwickeln.   
Diese Integration könnte auch dazu beitragen, dass das Programm noch ausgebauter und benutzerfreundlicher wird, indem es in der Lage ist, das Verhalten und die Präferenzen der Benutzer zu analysieren und darauf basierend Anpassungen an den GUI-Elementen vorzunehmen.   
Insgesamt wird die Entwicklung von GUI-Erstellungswerkzeugen wie meinem Programm in Zukunft weiter an Bedeutung gewinnen, da die Nachfrage nach benutzerfreundlichen und leistungsfähigen Anwendungen mit ansprechenden und effizienten Benutzeroberflächen weiter zunehmen wird.

# Literatur- und Quellenverzeichnis

## Literaturverzeichnis

* <https://dribbble.com/tags/game-menu>
* [Python GUI Programming with Tkinter, 2nd edition by Alan D. Moore](https://www.packtpub.com/product/python-gui-programming-with-tkinter-2nd-edition-second-edition/9781801815925)

## Quellenverzeichnis

* <https://docs.python.org/3/library/tkinter.html>
* <https://github.com/dawid-scripts/Fluent/blob/master/Example.lua>
* <https://github.com/topics/tkinter-python>
* <https://stackoverflow.com/questions/31699447/how-to-move-a-tkinter-button>
* <https://github.com/Akascape/CTkMessagebox>
* <https://coderslegacy.com/python/customtkinter-messagebox-using-ctkmessagebox/>
* <https://www.askpython.com/python-modules/top-best-python-gui-libraries>
* <https://realpython.com/pysimplegui-python/>
* <https://github.com/TomSchimansky/Customtkinter>
* <https://stackoverflow.com/questions/14928146/python-moving-an-object-label-button-and-changing-its-attribute>
* <https://stackoverflow.com/questions/69720181/how-to-make-a-tkinter-button-clickable-or-not-on-a-given-condition>
* <https://www.w3schools.com/python/>

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst und keine anderen als die im Literaturverzeichnis angegebenen Hilfsmittel verwendet habe. Auch alle genutzten Internetquellen wurden unter Angabe des Entnahmedatums kenntlich gemacht.

Da Textteile, die mit Hilfe textgenerierender KI erstellt wurden, grundsätzlich keine selbstständige Leistung im Sinne der Aufgabe darstellen, wurden diese unter Angabe der genutzten Generierungsimpulse ebenfalls kenntlich gemacht.

Sofern sich – auch zu einem späteren Zeitpunkt – herausstellen sollte, dass die Arbeit oder Teile davon nicht selbstständig verfasst wurden, die Zitationshinweise fehlen oder aus dem Internet entnommene bzw. entlehnte bzw. KI-unterstützt erstellte Textteile nicht gekennzeichnet sind, so wird die Arbeit auch nachträglich mit null Punkten gewertet.