HELLDIVERS II Ausrüstungshelfer

# Inhaltsverzeichnis

[1. Einführung über das Projekt S.3](#_Toc198571440)

[1.1 Überblick über das Projekt S.3](#_Toc198571441)

[1.2 Zielsetzung des Projekts S.3](#_Toc198571442)

[1.3 Zielgruppe der Dokumentation S.3](#_Toc198571443)

[1.4 Funktionalität des Programms S.3](#_Toc198571444)

[1.5 Anwendungsbeispiele S.3](#_Toc198571445)

[1.6 Verwendete Technologien und Bibliotheken S.3](#_Toc198571446)

[2. Installation und Einrichtung S.4](#_Toc198571447)

[2.1 Voraussetzungen S.4](#_Toc198571448)

[2.2 Installationsanleitung S.4](#_Toc198571449)

[3. Benutzerhandbuch S.4](#_Toc198571450)

[3.1 Screenshots des Frontend S.4](#_Toc198571451)

[4. Technische Details S.6](#_Toc198571452)

[4.1 Grundlegender Programmaufbau S.6](#_Toc198571453)

[4.2 Erklärung wichtiger Code-Abschnitte S.6](#_Toc198571454)

[4.3 Nassi-Shneiderman-Diagramm S.7](#_Toc198571455)

[5. Herausforderungen und Lösungen S.8](#_Toc198571456)

[5.1 Schwierigkeiten während der Entwicklung S.8](#_Toc198571457)

[5.2 Lösungsansätze und getroffene Entscheidungen S.8](#_Toc198571458)

[6. Ausblick und Weiterentwicklung S.8](#_Toc198571459)

[Quellen S.9](#_Toc198571460)

# Einführung über das Projekt

Hier stehe warum wir dieses Programm entwickeln.

## 1.1 Überblick über das Projekt

Dieses Projekt ist ein Werkzeug für Helldivers 2-Spieler, um die optimale Ausrüstung auszuwählen und das Spiel einfacher zu spielen.

## 1.2 Zielsetzung des Projekts

Dieses Projekt soll dem Benutzer dabei helfen, die beste Ausrüstungslösung für den Kampf gegen den Feind zu finden und ihm basierend auf der Art und Schwierigkeit des Feindes die optimale Ausrüstungsoption zu bieten.

## 1.3 Zielgruppe der Dokumentation

Dieses Projekt richtet sich an Helldivers 2-Spieler, die Schwierigkeiten haben, die Mission abzuschließen.

## 1.4 Funktionalität des Programms

Dieses Tool analysiert den ausgewählten Schwierigkeitsgrad und die gegnerische Fraktion und gibt die optimale Ausrüstung vor.

## 1.5 Anwendungsbeispiele

Der Benutzer kann dieses Tool verwenden, bevor er eine neue Mission in Helldivers 2 startet. Anschließend kann der Benutzer im Spiel die optimale Ausrüstung auswählen.

## 1.6 Verwendete Technologien und Bibliotheken

Diese Python Programm braucht:

* Tkinter
* Pillow
* time
* os
* logging
* yaml

# 2. Installation und Einrichtung

Hire steht wie kann man dieses Programm installieren.

## 2.1 Voraussetzungen

Es braucht:

* python ==3.10.11
* pillow == 10.4.0
* PyYAML == 6.0.1

## 2.2 Installationsanleitung

Alle Voraussetzungen kann durch pip und req.txt Datei automatisch installieren.

pip install -r req.txt

Achtung! Für „pillow“ muss man es manuell installieren mit:

pip install pillow

# 3. Benutzerhandbuch

Hier steht wie man dieses Programm benutzen können.

## 3.1 Screenshots des Frontend

A screenshot of a video game with Paramount Pictures in the background

AI-generated content may be incorrect.

Abb. 1

Hier klicken, dann das Hauptprogramm gelaufen wird.

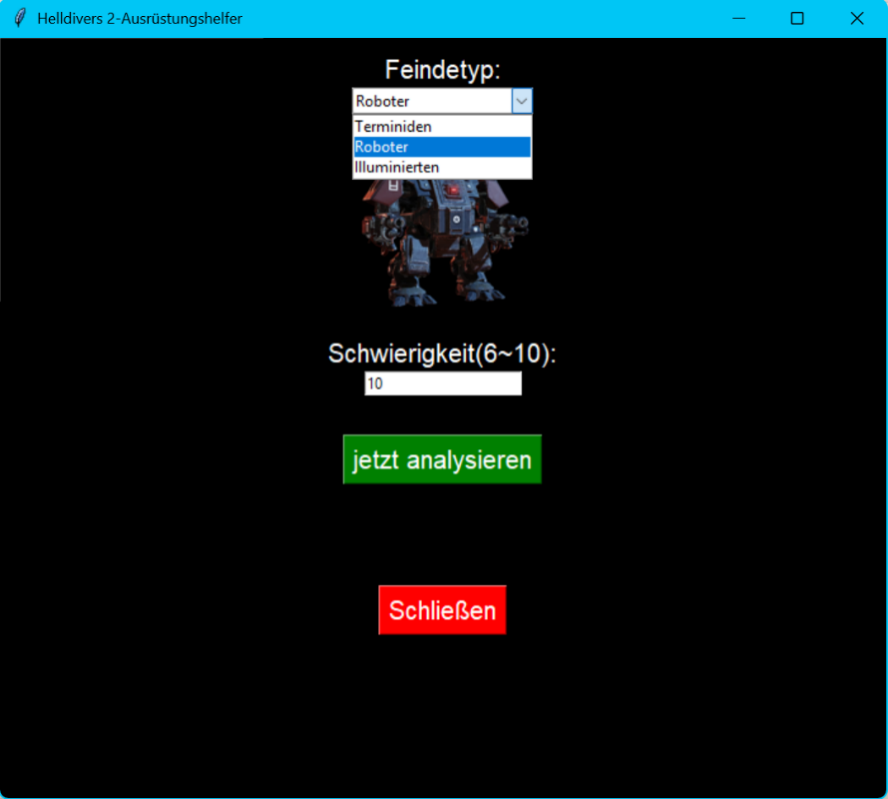


Abb. 2

Hier kann man die Schwierigkeit und der Feindetyp wählen.

A screenshot of a computer

AI-generated content may be incorrect.

Abb. 3

Die Empfohlene Waffen werden schauen, wenn man „jetzt analysieren“.

# 4. Technische Details

Hier steht was passiert unter unser Programm.

## 4.1 Grundlegender Programmaufbau

## In diesem Programm haben wir drei Kernmodule. Das erste ist das Backend, das Benutzereingaben verarbeitet und Ratschläge gibt. Dieses Kernmodul lädt die Konfiguration aus einer YAML-Datei, ist also skalierbar. Das zweite ist das GUI-Modul, das mithilfe der Tkinter-Bibliothek eine grundlegende GUI bereitstellt. Das letzte Modul ist der Launcher, auch bekannt als main.py, der für ein grundlegendes CLI-Frontend und das Starten der GUI verantwortlich ist.

## 4.2 Erklärung wichtiger Code-Abschnitte

Die Funktion `show\_main\_ui()` dient als Haupteinstiegspunkt für die Helldivers 2-Ausrüstungshelfer-Software und initialisiert die Benutzeroberfläche. Zuerst wird eine Titelleiste mit einem Startknopf angezeigt, der bei Betätigung zur Hauptansicht überleitet. Diese Hauptansicht beinhaltet das Laden von Bildern, die Einrichtung einer Beschriftung zur Auswahl des Gegnertyps und des Schwierigkeitsgrades sowie Schaltflächen zum Starten der Analyse. Weiterhin gibt es ein Eingabefeld zur Auswahl des Gegnertyps ("Terminiden", "Roboter" oder "Illuminierten"), und die Auswahl wird in einem Label auf der Hauptseite angezeigt. Ein "Jetzt analysieren"-Button löst bei Klick die Funktion `on\_submit()` aus, welche die Variablen `weapon\_output`, `weapon\_imgs` und `result\_label` aufruft. Zudem wird ein Rahmen zur Anzeige von Waffenbildern eingerichtet, und der Gegnertyp sowie der Schwierigkeitsgrad werden basierend auf den Eingabewerten aktualisiert. Die Bilder der verschiedenen Waffen sind in separaten Variablen gespeichert. Abschließend wird das Hauptfenster mit dem festgelegten Titel und der Größe erstellt. Wenn das Skript direkt ausgeführt wird, erscheint zunächst der Startbildschirm, bevor die Hauptbenutzeroberfläche angezeigt wird.

## 4.3 Nassi-Shneiderman-Diagramm

Hier steht unser Nassi-Shneiderman-Diagramm

A white sheet of paper with black text

AI-generated content may be incorrect.

Abb. 4

# 5. Herausforderungen und Lösungen

Hier stehen welche Herausforderungen wir standen.

## 5.1 Schwierigkeiten während der Entwicklung

Es gibt viele Probleme während des Entwicklungsprozesses, und zwei waren jedoch besonders schwerwiegend.

Zum Beispiel, Bilder wurden nicht aus dem richtigen Ordner geladen. Es funktioniert nicht, wenn das aktuelle Arbeitsverzeichnis nicht gleich mit dem Speicherort des Skripts ist.

Anderes Problem ist das Fenster immer zu groß oder klein sein. Wenn es zu groß oder klein ist, werde das Bild nicht richtig zeigen können.

## Das größte Problem ist die chaotische Code-Versionierung. Wir haben während der Entwicklungsphase zu viele Änderungen an der Codebasis vorgenommen. Dies führte zu einer verwirrenden Versionsverwaltung und erschwerte die Suche nach der richtigen Version sowie die Zusammenarbeit.

## 5.2 Lösungsansätze und getroffene Entscheidungen

Für das Problem mit Bildern, benutzen wir eines relativen Pfads statt eines absoluten Pfads. Im Programm haben wir os.path verwendet. Es wird bei jedem Skriptlauf der entsprechende relative Pfad erstellt.

Für die Größe des Fensters haben wir die Fenstergröße für den Titelbildschirm festgelegt, um sicherzustellen, dass sich die Schaltflächen und das Bild immer an derselben Position befinden.

# Bei chaotischer Code-Versionierung verwenden wir Git, um die Versionen zu verwalten. Außerdem laden wir die gesamte Codebasis auf Github hoch, um die Zusammenarbeit zu verbessern.

# 6. Ausblick und Weiterentwicklung

Für die weitere Entwicklung können wir die Unterstützung der Verwendung von API hinzufügen, um Generalbefehle und Planeteninformationen aus dem Spiel abzurufen. Außerdem können wir je nach Planeten weitere unterschiedliche Ausrüstungen hinzufügen. Bei kaltem Wetter empfehlen wir beispielsweise die Verwendung einer Laserwaffe, da diese länger schießen kann.

# Quellen

Abbildung 4 Shun Yiu, Yeung

Abbildung 1 Shun Yiu, Yeung

Abbildung 2 Shun Yiu, Yeung

Abbildung 3 Shun Yiu, Yeung