
Game Hardware Checker

Leistungsbeurteilung LB2 - Dokumentation

Modul 122

Eingereicht von: Abdulrahman Rasul
Hristijan Kocev

Eingereicht bei: Alexis Winiger

Abgabedatum: 2021. Dezember 2024

Inhaltsverzeichnis

1	Ziele und Anforderungen	2
1.1	Einleitung	2
1.2	Zweck des Skriptes	2
1.3	Ziele	3
1.4	Anforderungen.....	3
1.4.1	Allgemeine Anforderungen.....	3
1.4.2	Eingaben und Ausgaben.....	3
1.4.3	Programmtechnische Anforderungen	3
2	Ablaufdiagramm	4
2.1	PAP / Flussdiagramm / Struktogramm.....	4
2.2	Kommentar / Beschreibung	4
3	Skript/Programm (Realisierung).....	5
3.1	Technologie Bash / Python.....	5
3.2	Ein- und Ausgabe	5
3.2.1	Eingaben:.....	5
3.2.2	Ausgaben:.....	5
3.3	Kontrollstrukturen	5
3.4	Regex.....	6
3.5	Datenbank.....	6
3.6	Sicherheitsmassnahmen	6
4	Usecases und Testfälle	6
4.1	Usecase	6
4.1.1	Beschreibung:.....	6
4.1.2	Ablauf:.....	6
4.1.3	Beispiel:	6
4.2	Testfall.....	7
4.3	Hardwareerkennung:	7
4.3.1	Leistungsanalyse:	7
4.3.2	Funktionalität der Eingabe.....	7
4.3.3	Ergebnisse	8
5	Reflexion	8
5.1	Journal	9
6	Fazit	9

1 Ziele und Anforderungen

1.1 Einleitung

Das Projekt "Hardware-Kompatibilitätscheck" wurde ursprünglich entwickelt, um Nutzern eine einfache und intuitive Möglichkeit zu bieten, die Hardware-Kompatibilität ihres PCs für spezifische Spiele oder Anwendungen zu überprüfen. Das Skript sollte ursprünglich in PowerShell umgesetzt werden, musste jedoch aufgrund begrenzter Möglichkeiten auf Python umgestellt werden. Python erwies sich als flexibler und besser geeignet für die Anforderungen des Projekts.

Unsere ursprüngliche Idee, die Hardware unseres Geräts auszulesen, die Anforderungen aus einer Datenbank zu vergleichen und die Ergebnisse in einer GUI darzustellen, war schwierig umzusetzen. Der Hauptgrund dafür war das Fehlen einer geeigneten API, die wir als Anfänger nicht selbst entwickeln oder finden konnten. Daher wurde die Projektidee angepasst.

Jetzt liest das Skript die Hardware des Geräts aus und gibt eine Schätzung der Spieleleistung in FPS basierend auf Benchmark-Punkten für die ausgewählte Hardware (CPU, GPU). Hierfür wurden zwei separate Datenbanken implementiert: eine für die Hardware und eine für die Spiele. Falls eine Hardware oder ein Spiel in der Datenbank nicht vorhanden ist, können die benötigten Benchmark-Punkte manuell eingegeben und dauerhaft gespeichert werden, um in Zukunft darauf zugreifen zu können.

1.2 Zweck des Skriptes

Der Zweck des Skriptes ist es, Nutzern eine Schätzung der Spieleleistung (in FPS) basierend auf ihrer aktuellen Hardware-Konfiguration zu bieten. Dies hilft dabei, die Leistungsfähigkeit des Systems besser einzuschätzen und informiert darüber, ob ein Upgrade erforderlich sein könnte. Das Skript ermöglicht es ausserdem, neue Benchmark-Daten einzugeben und diese der Datenbank hinzuzufügen, um die Datenbank Aktuell halten zu könne

1.3 Ziele

Entwicklung des Skripts: Ein funktionales Skript, das Hardwaredaten ausliest und eine FPS-Schätzung liefert.

Datenbankmanagement: Speicherung und Verwaltung von Hardware-Benchmarks und Spieledaten.

Erweiterbare Funktionalität: Möglichkeit zur Eingabe und Speicherung neuer Hardware- oder Spieledaten durch den Nutzer.

Benutzerfreundliche Ausgabe: Darstellung der FPS-Schätzung und Empfehlungen für Optimierungen.

1.4 Anforderungen

1.4.1 Allgemeine Anforderungen

- Alle Teammitglieder beteiligen sich gleichmässig an der Entwicklung, Dokumentation und dem Testen des Projekts.
- Das Projekt soll sowohl für Anfänger als auch für fortgeschrittene Nutzer leicht verständlich und zugänglich sein.
- Fertigstellung und Abgabe des Projekts vor den Weihnachtsferien.

1.4.2 Eingaben und Ausgaben

Eingaben:

- Systemkomponenten wie RAM, GPU und CPU werden automatisiert ausgelesen.
- Benutzer können Programme oder Anwendungen angeben/auswählen, für die sie den Kompatibilitätscheck durchführen möchten.

Ausgaben:

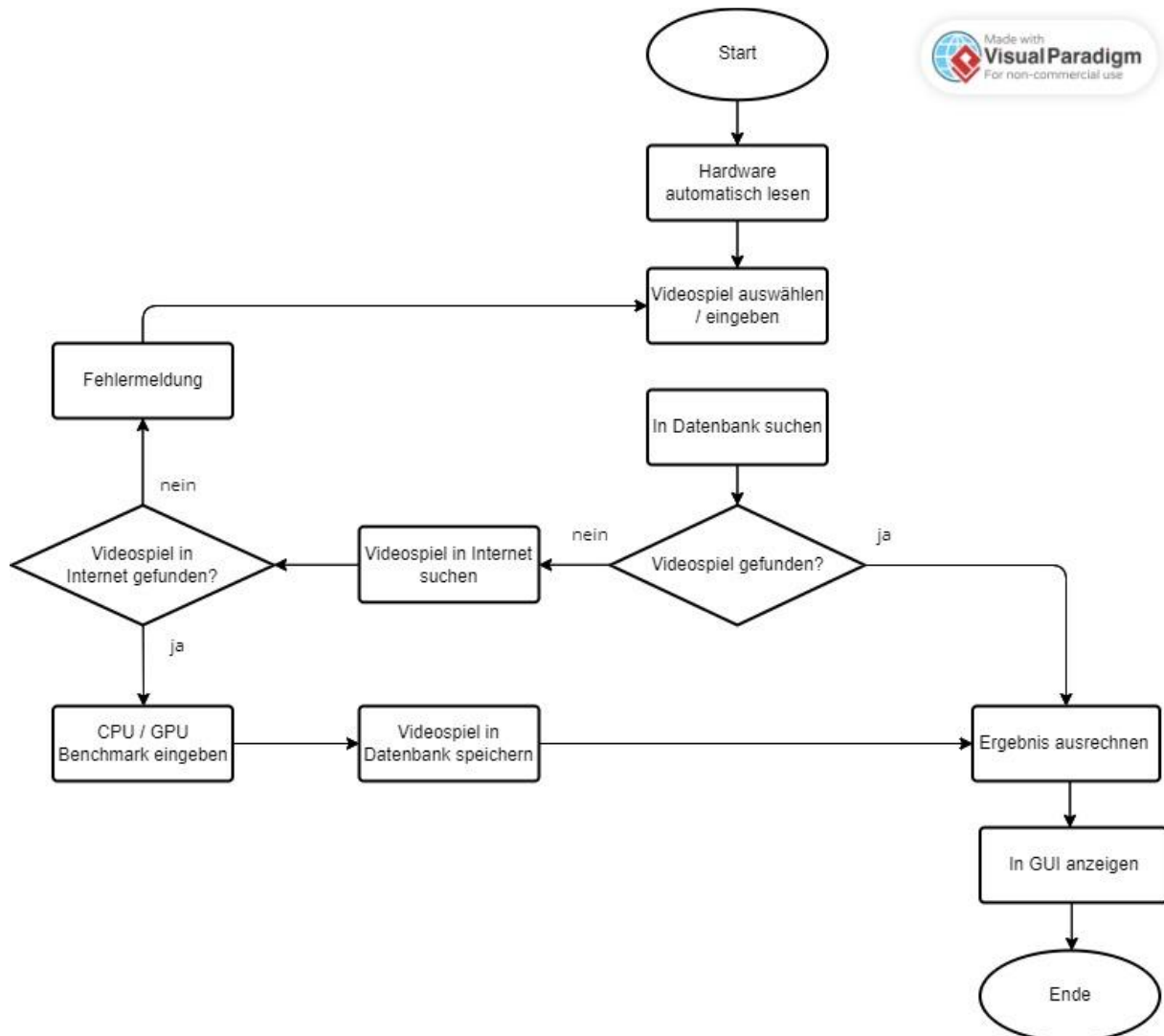
- Berechnung der Leistung basierend auf den Systemkomponenten und den Anforderungen des gewählten Spiels.
- Ergebnisse werden grafisch und benutzerfreundlich in der GUI dargestellt.

1.4.3 Programmtechnische Anforderungen

- Das Skript muss vollständig umgesetzt sein und auf gängigen Windows-Versionen lauffähig sein.
- Das Projekt wird auf GitHub gehostet, um die Versionskontrolle und Teamarbeit zu erleichtern.
- Modularer Aufbau des Skripts, um zukünftige Erweiterungen und Anpassungen zu ermöglichen.
- Ein README-datei mit Instruktionen zu der Nutzung des Skripts.
- Kommentierung des Codes, um ihn leichter verständlich zu machen.

2 Ablaufdiagramm

2.1 PAP / Flussdiagramm / Struktogramm



2.2 Kommentar / Beschreibung

Das Flussdiagramm zeigt den Ablauf des Skripts zur Berechnung der FPS-Leistung für Videospiele basierend auf der vorhandenen Hardware. Der Ablauf beginnt mit dem automatischen Auslesen der Systemdaten, wie CPU, GPU und RAM. Danach kann der Benutzer ein Spiel auswählen oder ein neues Spiel eingeben.

Wenn das Spiel in der lokalen Datenbank gefunden wird, werden die Leistungsdaten direkt abgerufen. Falls das Spiel nicht gefunden wird, fordert das Skript den Benutzer dazu auf, die minimalen CPU- und GPU-Benchmarks für das Spiel einzugeben. Anschliessend sucht das Skript online nach zusätzlichen Daten. Falls das Spiel auch im Internet nicht gefunden werden kann, wird eine Fehlermeldung ausgegeben.

Wenn die notwendigen Daten vorliegen, berechnet das Skript die FPS-Leistung für das Spiel und zeigt die Ergebnisse in einer grafischen Benutzeroberfläche (GUI) an. Der Ablauf endet mit der Darstellung der berechneten Ergebnisse.

3 Skript/Programm (Realisierung)

3.1 Technologie Bash / Python

Das Projekt wurde vollständig in Python umgesetzt. Python wurde aufgrund seiner Flexibilität und der breiten Verfügbarkeit von Bibliotheken gewählt. Bibliotheken wie «psutil» wurden für die Hardwareabfrage genutzt.

3.2 Ein- und Ausgabe

3.2.1 Eingaben:

- Automatisches Auslesen der Hardwaredaten.
- Benutzer können Spiele auswählen und Benchmark-Daten manuell eingeben.

3.2.2 Ausgaben:

- Schätzung der FPS-Leistung für ein ausgewähltes Spiel basierend auf Benchmark-Daten.
- Optional: Ausgabe der Ergebnisse in einer Textdatei.
- Es gibt schriftlich einen Kommentar dazu, wie «You can play this game at a decent performance level».

3.3 Kontrollstrukturen

Das Skript verwendet grundlegende und erweiterte Kontrollstrukturen, wie:

- **If-Else-Bedingungen:** Zur Überprüfung der Hardwareanforderungen.
- **Schleifen:** Um mehrere Spiele oder Anwendungen in einer Sitzung zu prüfen.
- **Try-Catch-Blöcke:** Zur Fehlerbehandlung, z. B. wenn Hardwaredaten nicht ausgelesen werden können

3.4 Regex

Reguläre Ausdrücke (Regex) werden verwendet, um:

- Hardwareinformationen aus komplexen Zeichenketten zu extrahieren.
- Benutzereingaben wie Spielnamen auf Korrektheit zu prüfen.

3.5 Datenbank

Zwei Datenbanken wurden implementiert. Sie enthält:

- **Hardware-Datenbank:** Speichert Benchmark-Punkte für CPUs und GPUs.
- **Spiele-Datenbank:** Enthält Informationen über die Systemanforderungen von Spielen.
- Funktionen zur Aktualisierung und Speicherung neuer Einträge sind integriert.

3.6 Sicherheitsmassnahmen

- Validierung der Benutzereingaben.
- Vermeidung von Fehlern bei der Datenbankabfrage durch vordefinierte Strukturprüfungen.

4 Usecases und Testfälle

4.1 Usecase

4.1.1 Beschreibung:

Ein Nutzer möchte wissen, wie gut ein bestimmtes Spiel auf seinem Computer läuft. Das Ziel ist es, basierend auf der Hardware des Nutzers und den hinterlegten Benchmark-Daten, eine FPS-Schätzung zu liefern.

4.1.2 Ablauf:

1. Der Nutzer startet das Programm und gibt den Namen des Spiels ein, das er testen möchte.
2. Das Programm prüft, ob das Spiel und die Hardware des Nutzers in den Datenbanken vorhanden sind.
Wenn ja: Es werden die Benchmark-Daten verwendet, um die FPS-Leistung zu berechnen.
Wenn nein: Der Nutzer wird aufgefordert, die fehlenden Benchmark-Daten einzugeben.
3. Die FPS-Schätzung wird berechnet und dem Nutzer angezeigt.

4.1.3 Beispiel:

Eingabe: Der Nutzer wählt "GTA V" aus.

System: CPU: Intel i5-9600K

GPU: NVIDIA GTX 1660 Super

Datenbank: GTA V; GPU- und CPU-Benchmark-Daten vorhanden.

Berechnung: Das Skript schätzt basierend auf den Benchmark-Punkten, dass das Spiel auf mittleren Einstellungen bei 1080p mit 45 FPS läuft.

Ausgabe: "Geschätzte Leistung für GTA V: 45 FPS (1080p, mittlere Einstellungen)."

Erweiterung: Falls der Nutzer möchte, kann er die Ergebnisse speichern oder eine weitere Analyse mit anderen Spielen durchführen.

4.2 Testfall

4.3 Hardwareerkennung:

Das Programm liest die Hardwaredaten zuverlässig aus. Falls die Hardware oder das Spiel nicht in der Datenbank vorhanden ist, wird der Nutzer aufgefordert, die Benchmark-Punkte manuell einzugeben. Diese können dann dauerhaft gespeichert werden.

```
Welcome to the Game Benchmark Checker!  
Detected CPU: AMD Ryzen 7 5800U with Radeon Graphics  
Detected GPU: Integrated GPU  
  
Finalized Hardware: CPU = AMD Ryzen 7 5800U with Radeon Graphics, GPU = Integrated GPU, RAM = 31GB
```

4.3.1 Leistungsanalyse:

Eingabe: Auswahl eines Spiels mit hinterlegten Anforderungen.

Erwartetes Ergebnis: Schätzung der FPS-Leistung und Empfehlungen für Optimierungen.

```
Choose a game (number): 5  
  
You selected: League of Legends  
  
Estimated FPS for League of Legends: 46.02 FPS  
You can play this game at a decent performance level.  
  
Would you like to search again or exit? (search/exit):
```

4.3.2 Funktionalität der Eingabe

- Manuelle Eingabe von Benchmark-Daten für Hardware oder Spiele erforderlich, wenn sie bereits nicht vorhanden sind.
- Erfolgreiche Speicherung und Abrufbarkeit der eingegebenen Daten bei zukünftigen Abfragen.

4.3.3 Ergebnisse

Die Tests zeigten, dass das Skript in allen Szenarien stabil arbeitet. Die Datenbanken reagierten performant auf Abfragen und das Speichern neuer Daten funktionierte fehlerfrei.

5 Reflexion

Die Umstellung von PowerShell auf Python war eine wichtige Entscheidung, die die Umsetzung des Projekts erheblich erleichtert hat. Python bietet eine benutzerfreundlichere und vielseitigere Entwicklungsumgebung, die es ermöglichte, die verschiedenen Anforderungen des Projekts effizient umzusetzen. Besonders hilfreich war die Verfügbarkeit zahlreicher Bibliotheken wie psutil, GPUtil und cpuinfo, die eine präzise und unkomplizierte Erfassung von Systeminformationen ermöglichten. Diese Flexibilität und die einfache Handhabung von Python trugen wesentlich dazu bei, dass wir das Projekt schneller und fehlerfreier realisieren konnten.

Die Anpassung der ursprünglichen Projektidee hin zu einer FPS-Schätzung war ebenfalls sinnvoll und führte zu einem praktikablen Ergebnis. Durch die Fokussierung auf die Berechnung der FPS in Relation zur Hardwarekonfiguration konnten wir eine konkrete, messbare Auswertung bieten, die dem Benutzer klare Informationen über die Spielbarkeit eines Spiels liefert. Zudem gab uns diese Anpassung die Möglichkeit, die Hardware des Benutzers interaktiv in die Datenbank aufzunehmen, was nicht nur den praktischen Nutzen erhöhte, sondern auch das Projekt flexibler und anpassungsfähiger gestaltete.

Für künftige Projekte könnten wir versuchen, die GUI-Komponente wieder zu integrieren, um die Benutzerfreundlichkeit weiter zu verbessern. Eine grafische Benutzeroberfläche würde nicht nur die Interaktivität erhöhen, sondern auch die Bedienung für weniger technisch versierte Benutzer deutlich erleichtern. Indem wir visuelle Elemente wie Schaltflächen, Menüs und Diagramme einfügen, könnte die Nutzung des Programms intuitiver und ansprechender gestaltet werden. Zudem könnte eine GUI auch dazu beitragen, die Ergebnisse der FPS-Schätzung übersichtlicher darzustellen und dem Benutzer eine klarere Vorstellung von der Leistung seines Systems zu geben. Insgesamt würde dies das Projekt zugänglicher machen und den praktischen Wert weiter steigern.

5.1 Journal

Tag	Tätigkeit	Person	Bemerkungen
KW 48	Projektidee entwickelt, Anforderungen definiert	AR HK	Erste Idee war ambitioniert, erfordert Umstrukturierung.
KW 49	Umstellung auf Python beschlossen. Mit dem Code angefangen.	AR HK	Python bietet mehr Möglichkeiten und Flexibilität, mit PowerShell können wir unser Projekt so nicht umsetzen.
KW 49	An dem Code gearbeitet. Dokumentation erstellt.	AR HK	Weiter recherchiert und Code angepasst. Dokumentation und Datenstruktur erstellt.
KW 50	An dem Code gearbeitet. Testfälle definiert und durchgeführt	AR HK	Tests liefen stabil, kleinere Fehler in der Eingabevalidierung behoben.
KW 51	Dokumentation bearbeitet und finalisiert Abgabe.	AR HK	Dokumentation deckt alle Punkte ab, Übersicht könnte noch verbessert werden.

6 Fazit

Das "Game Hardware Checker" in seiner neuen Form ist eine praktische Lösung, die es ermöglicht, die Hardware-Leistung für Spiele zu schätzen. Das Projekt zeigt, wie Python mit Datenbanktechnologien kombiniert werden kann, um eine flexible und erweiterbare Anwendung zu entwickeln. Durch die Implementierung einer automatischen Hardware-Erkennung, die Integration einer Datenbank und die Möglichkeit, Benchmarks manuell hinzuzufügen, bietet das Skript eine vielseitige Grundlage für die FPS-Schätzung.

Besonders hervorzuheben ist die Benutzerfreundlichkeit des Systems: Benutzer können entweder Spiele aus einer Datenbank auswählen oder neue Daten eingeben, die anschliessend gespeichert und für zukünftige Abfragen genutzt werden. Dies macht die Anwendung sowohl für private Nutzer als auch für technische Analysen geeignet.

Die modulare Struktur des Skripts erlaubt es, zukünftige Funktionen, wie detailliertere FPS-Prognosen oder eine erweiterte grafische Benutzeroberfläche, leicht zu integrieren. Darüber hinaus zeigt das Projekt, wie Python als leistungsstarkes Werkzeug zur Lösung praktischer Probleme dienen kann, indem es vielseitige Bibliotheken und Technologien nutzt.

Das Projekt hat nicht nur die technischen Anforderungen erfüllt, sondern auch den Teammitgliedern die Möglichkeit gegeben, ihre Fähigkeiten in Python, Datenbankdesign und kollaborativer Entwicklung zu erweitern. Insgesamt stellt das "Game Hardware Checker"-Projekt eine gelungene Kombination aus Innovation, Funktionalität und technischer Umsetzung dar, die sowohl für die beteiligten Entwickler als auch für die Endnutzer einen hohen Mehrwert bietet.