**Projektdokumentation**

Elefant / Arn, Lalic und Lehner

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Version** | **Änderung** | **Autor** |
| 27.10.2021 | 0.0.1 | Erste Version | Jan Lehner |
| 03.11.2021 | 0.0.2 | Programm ist fast fertig | Andrej Lalic |
| 10.11.2021 | 1.0.0 | Finale Version | Andrej Lalic |

# Informieren

1.1 Ihr Projekt

Wir haben das Projekt A "Vokabelabfragesystem" gewählt.

Ein Programm mit dem man Vokabeln einfach und unkompliziert lernen kann.

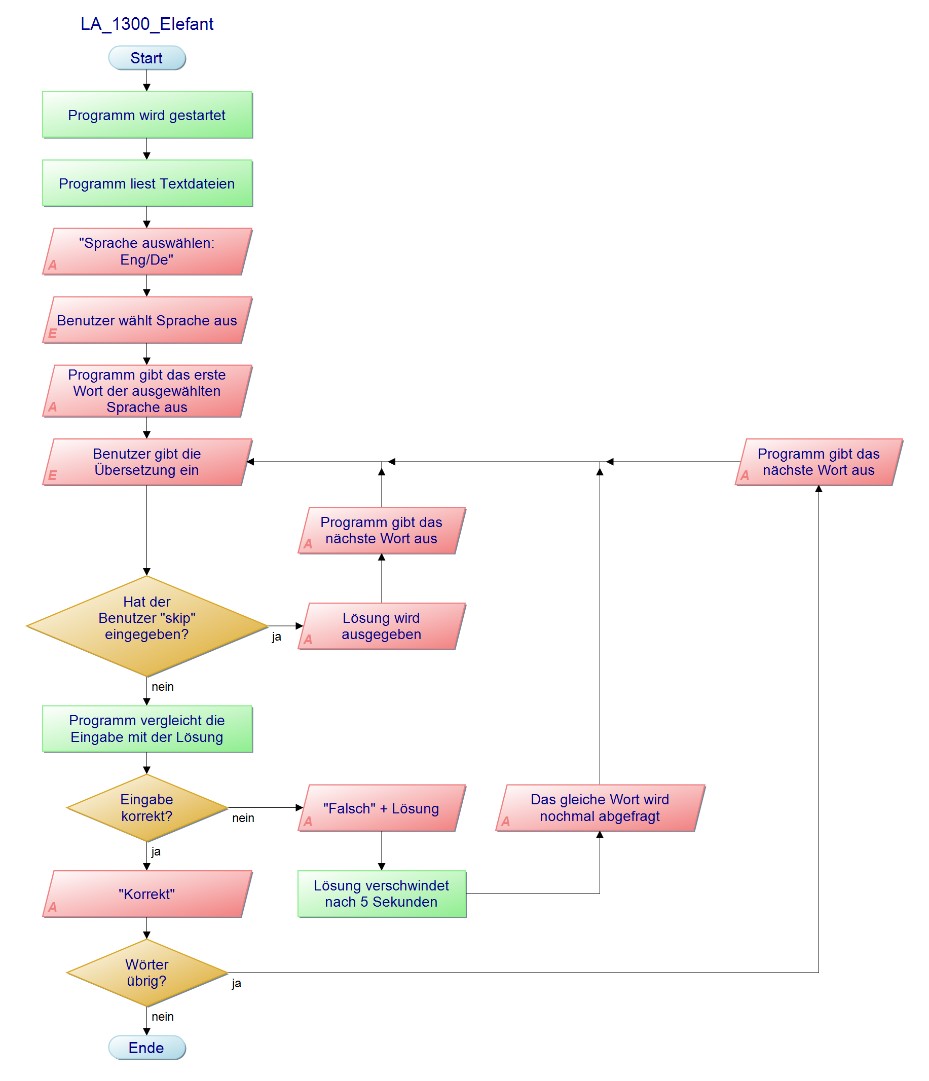
1.2 Quellen

Speichern der Wörter in Dateien --> https://docs.microsoft.com/de-de/troubleshoot/dotnet/csharp/read-write-text-file

1.3 Anforderungen

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Muss / Kann?** | **Funktional? Qualität? Rand?** | **Beschreibung** |
| 1 | Muss | Funktional | Die Vokabel, welche im Programm verwendet werden müssen in einer Datei abgespeichert werden. |
| 2 | Muss | Funktional | Das Programm muss aus einer Datei Wörter entnehmen können und diese Anzeigen können. |
| 3 | Muss | Funktional | Der Benutzer muss eine Eingabe für Wörter machen können. |
| 4 | Muss | Funktional | Alle Wörter entweder aus der 1. Datei oder der 2. Datei werden abgefragt. (Die Lösungen dazu befinden sich jeweils in der anderen Datei.) |
| 5 | Muss | Funktional | Die zu übersetzenden Wörter werden auf Deutsch angezeigt und es wird darauf hingewiesen, dass die Übersetzung auf Englisch erfolgen soll. |
| 6 | Muss | Funktional | Der Benutzer wird alle Wörter abgefragt. |
| 7 | Muss | Funktional | Die Eingabe des Benutzers muss mit der Übersetzung des angezeigten Worts verglichen werden. Je nach dem, ob Sie identisch sind oder nicht gibt es eine Verschiedene Ausgabe |
| 8 | Kann | Funktional | Wenn die Eingabe des Benutzers falsch war und der jeweilige Text ausgegeben wurde wird zusätzlich die richtige Übersetzung angezeigt welche nach 5sek. verschwindet. |
| 9 | Muss | Funktonal | Wenn die Übersetzung falsch ist, wird das gleiche Wort erneut abgefragt. |
| 10 | Muss | Funktional | Wenn die Eingabe des Benutzers mit der Lösung übereinstimmt, muss ein Text ausgegeben werden, welcher sagt, dass die Eingabe korrekt ist. |
| 11 | Muss | Funktional | Wenn das Wort richtig war, wird das nächste Wort mit dem gleichen Schema abgefragt. |
| 12 | Kann | Funktional | Wenn der User bei der Eingabe "skip" eingibt, wird die Lösung angezeigt und danach kommt das nächste  Wort |
| 13 | Kann | Qualität | Wenn man das Programm öffnet, wird der Titel mit einer ASCII Art angezeigt. |
| 14 | Kann | Qualität | Die Schriftfarbe der Konsole sollte in verschiedenen Farben ausgegeben werden. |
| 15 | Kann | Funktional | Jedes Wort welches richtig eigegeben wurde sollte nur einmal abgefragt werden |
| 17 | Kann | Funktional | Es wird gefragt, ob man nochmals spielen will und bei ja wird das Programm nochmals ablaufen. |
| 18 | Kann | Funktional | Das Programm beendet sich, wenn «stop» eingegeben wird. |

* 1. Diagramme



* 1. Testfälle

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Vorbereitung** | **Eingabe** | **Erwartete Ausgabe** |
| 1.1 | Der Computer ist gestartet | Sucht nach dem Ordner, in dem die Wörter abgespeichert sind und öffnet ihn. | Die Wörter werden gefunden. |
| 2.1 | \*Der Computer ist gestartet und das Programm läuft. | Keine Eingabe. | Beliebiges Wort. |
| 3.1 | \* | Der Benutzer gibt die richtige Übersetzung für ein Wort ein | Herzlichen Glückwunsch, Sie haben das Wort richtig. |
| 4.1 | \* | Der Benutzer gibt die Übersetzung für ein Wort ein | Herzlichen Glückwunsch, Sie haben das Wort richtig. |
| 5.1 | \* | Keine Eingabe. | Deutsches Wort. |
| 6.1 | \* | Die jeweils korrekte Übersetzung wird eingegeben. | Herzlichen Glückwunsch, Sie haben das Wort richtig. Danach wird das nächste Wort zum Übersetzen ausgegeben. |
| 7.1 | \* | Der Benutzer gibt ein Wort ein, welches nicht identisch mit dem aus der Datei ist. | Es gibt die Ausgabe: "Leider haben Sie das Wort nicht richtig geschrieben." |
| 7.2 | \* | Der Benutzer gibt ein Wort ein, welches identisch mit dem Wort aus der Datei ist. | Es gibt die Ausgabe: "Herzlichen Glückwunsch, Sie haben das Wort richtig." |
| 8.1 | \* | Eine falsche Übersetzung | Ausgabe von 7.1, dann die richtige Übersetzung, welche nach 5 Sekunden verschwindet. |
| 9.1 | \* | Eine falsche Übersetzung | Ausgabe von 8.1, dann das gleiche Wort auf Deutsch. |
| 10.1 | \* | Eine korrekte Übersetzung | «Herzlichen Glückwunsch, Sie haben das Wort richtig.» |
| 11.1 | \* | Eine korrekte Übersetzung | 10.1 und dann das nächste deutsche Wort. |
| 12.1 | \* | «skip» | Lösung wird angezeigt, dann das nächste deutsche Wort. |
| 13.1 | \* | Keine Eingabe | Titel sieht nach einer ASCII Art aus. |
| 14.1 | \* | Keine Eingabe | Die Texte haben verschiedene Farben. |
| 15.1 | \* | Jeweils richtige Übersetzung | Es wird nie das gleiche Wort abgefragt. |
| 17.1 | Computer gestartet, Programm läuft, Benutzer hat 50-mal «skip» eingegeben. | Keine Eingabe | Möchten Sie nochmals spielen? |
| 17.2 | Gleich wie 17.1 | «Ja» | Die Wörter werden nochmals abgefragt. |
| 18.1 | \* | «stop» | Das Programm beendet sich. |

Planung:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nummer** | **Frist** | **Beschreibung** | **Zeit (geplant)** |  |  |
| 1.1 | Vor  2.1 | Vokabeln zusammensuchen und in einer Datei abspeichern, die aufgerufen werden kann. | 45 Minuten |
| 2.1 | Vor  5.1 | Das Programm muss aus einer Datei Wörter entnehmen und benutzen können. | 45 Minuten |
| 3.1 | Vor  5.1 | Der Benutzer muss eine Eingabe für Wörter machen können. | 45 Minuten |
| 4.1 |  | Alle Wörter entweder aus der 1. Datei oder der 2. Datei werden abgefragt. (Die Lösungen dazu befinden sich jeweils in der anderen Datei.) | 45 Minuten |
| 5.1 | Vor  6.1 | Die zu übersetzenden Wörter werden auf Deutsch angezeigt und es wird darauf hingewiesen, dass die Übersetzung auf Englisch verfolgen soll. | 45 Minuten |
| 6.1 | Vor  7.1 | Der Benutzer wird alle Wörter abgefragt. | 45 Minuten |
| 7.1 |  | Die Eingabe des Benutzers wird mit der tatsächlichen Übersetzung des Wortes verglichen. Es sollen verschiedene Ausgaben für verschiedene Fälle gemacht werden: Identisch/nicht identisch/»skip» | 45 Minuten |
| 8.1 |  | Jedes falsch übersetzte Wort wird nochmals abgefragt nachdem die Lösung angezeigt wurde, die nach 5 Sekunden verschwindet ist. | 45 Minuten |
| 9.1 | Vor 13.1 | Wenn die Eingabe des Benutzers falsch ist, wird das gleiche Wort erneut abgefragt. | 45 Minuten |
| 10.1 |  | Wenn die Eingabe des Benutzers korrekt ist, wird gesagt, dass sie korrekt ist. | 45 Minuten |
| 11.1 |  | 10.1 + das nächste Wort wird abgefragt | 45 Minuten |
| 12.1 |  | Bei der Eingabe «skip» wird die Lösung angezeigt und danach das nächste Wort. | 45 Minuten |
| 13.1 |  | Bei Öffnung des Programms wird der Titel mit einer ASCII Art angezeigt. | 45 Minuten |
| 14.1 |  | Der Text soll in verschiedenen Farben ausgegeben werden. | 45 Minuten |
| 15.1 |  | Wenn ein Wort richtig eingeben worden ist, wird es nur einmal abgefragt (angenommen, 17.1 existiert nicht) | 45 Minuten |
| 17.1 | 10.11.2021 | Es wird gefragt, ob der Nutzer nochmals spielen möchte. Bei «ja» wird das Programm nochmals ablaufen. | 45 Minuten |
| 18.1 | 10.11.2021 | Wenn der Benutzer «stop» eingibt, beendet das Programm sich selbst. | 45 Minuten |

# Entscheiden

Wir haben uns dazu entschieden, ein paar zusätzliche, nicht essenzielle Anforderungen zu stellen, die wir finden, würden die Qualität des Programmes sowie die Nutzerfreundlichkeit verbessern.

Diese beinhalten unter anderem das Zeigen der richtigen Lösung bei einer falschen Eingabe, die Möglichkeit, Wörter durch «skip» zu überspringen, eine hellgrüne Farbe, ein Titel, der durch eine ASCII Art angezeigt wird, das einmalige Abfragen von korrekt übersetzten Begriffen sowie ein Punkte System.

# Realisieren

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nummer** | **Datum** | **Beschreibung** | **Zeit (geplant in Minuten)** | **Zeit (effektiv)** |
| 1 | 03.11.2021 | Siehe die Planung\* | 45 | 45 (teilweise erfüllt) |
| 2 | 03.11.2021 | \* | 45 | 20 |
| 3 | 03.11.2021 | \* | 45 | 10 |
| 4 | 03.11.2021 | \* | 45 | 5 |
| 5 | 03.11.2021 | \* | 45 | 5 |
| 6 | 03.11.2021 | \* | 45 | 5 |
| 7 | 03.11.2021 | \* | 45 | 30 |
| 8 | 03.11.2021 | \* | 45 | 10 |
| 9 | 03.11.2021 | \* | 45 | 20 |
| 10 | 03.11.2021 | \* | 45 | 5 |
| 11 | 03.11.2021 | \* | 45 | 5 |
| 12 | 03.11.2021 | \* | 45 | 3 |
| 13 | 03.11.2021 | \* | 45 | 1 |
| 14 | 03.11.2021 | \* | 45 | 1 |
| 15 | 03.11.2021 | \* | 45 | Nebenbei |
| 16.1 | 10.11.2021 | \* | 45 | 5 |
| 16.2 | 10.11.2021 | \* | 45 | 20 |
| 16.3 | 10.11.2021 | \* | 45 | 5 |
| 17.1 | 10.11.2021 | \* | 45 | 5 |
| 18.1 | 10.11.2021 |  | 45 | 5 |

# Kontrollieren

**5.1 Testprotokoll**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nummer** | **Datum** | **Resultat** | **Durchgeführt** |
| **1.1** | **10.11.2021** | **OK** | Lalic |
| **2.1** | **10.11.2021** | **OK** | Lalic |
| **3.1** | **10.11.2021** | **OK** | Lalic |
| **4.1** | **10.11.2021** | **OK** | Lalic |
| **5.1** | **10.11.2021** | **OK** | Lalic |
| **6.1** | **10.11.2021** | **OK** | Lalic |
| **7.1** | **17.11.2021** | **OK** | elephant |
| **7.2** | **17.11.2021** | **OK** | elephant |
| **8.1** | **17.11.2021** | **OK** | elephant |
| **9.1** | **17.11.2021** | **OK** | elephant |
| **10.1** | **17.11.2021** | **OK** | elephant |
| **11.1** | **17.11.2021** | **OK** | elephant |
| **12.1** | **17.11.2021** | **OK** | elephant |
| **13.1** | **17.11.2021** | **OK** | elephant |
| **14.1** | **17.11.2021** | **OK** | elephant |
| **15.1** | **17.11.2021** | **OK** | elephant |
| **17.1** | **17.11.2021** | **NOK** | elephant |
| **17.2** | **17.11.2021** | **NOK** | elephant |
| **18.1** | **17.11.2021** | **OK** | elephant |

**Fazit: Das Programm kann so abgegeben werden, nur eine kann Anforderung wurde nicht erfüllt.**

# Auswerten

Gut gelaufen: Codierung ist gut gelaufen, konnten uns das nötige Wissen schnell aneignen.

Schlecht gelaufen: Teamkommunikation, konnten uns nicht gut entscheiden, welches Projekt wir machen, weshalb wir zusätzlich Zeit verbrauchen mussten, um sicher zu sein, dass wir dieses Projekt wirklich machen wollen.