# Dokumentation des Projekts „Quiz zur Topographie Sachsens“

Maximilian Kautzsch
Geschwister-Scholl-Gymnasium Freiberg
Klasse 10/3

18. April 2024

## 1. Dokumentation der verwendeten Methoden

In dieser JavaScript-Datei wurden insgesamt sechs Funktionen geschrieben. Um einen Überblick zu gewinnen, soll im Nachfolgenden jede Methode kurz beschrieben werden.

### 1.1. `showQuetions(index)`

Zeigt die Fragen und Optionen basierend auf dem angegebenen Index an.

#### Syntax

```
1  showQuetions(index);
```

#### Parameter

- *index*: Index der Frage, die angezeigt werden soll

### 1.2. `optionSelected(answer)`

Verarbeitet die vom Benutzer ausgewählte Option und aktualisiert das Spiel entsprechend. Die im Quiz verwendeten Bilder wurden mit Hilfe einer physischen stummen Karte Sachsens erstellt [Die].

#### Syntax

```
1  optionSelected(answer);
```

#### Parameter

- *answer*: Die vom Benutzer ausgewählte Antwortoption

### 1.3. `showResult()`

Zeigt das Endergebnis des Quiz an, einschließlich der Anzahl der richtigen Antworten.

#### Detaillierte Beschreibung

Die Funktion `showResult()` wird aufgerufen, wenn das Quiz beendet ist und zeigt das Endergebnis des Quiz an. Zuerst werden die verschiedenen HTML-Boxen angepasst, um die Ergebnisbox anzuzeigen und die Quiz-Box auszublenden. Dann wird überprüft, wie viele Fragen der Benutzer richtig beantwortet hat, und entsprechend wird eine Nachricht generiert und angezeigt. Je nach Punktzahl wird dem Benutzer entweder gratuliert oder ermutigt, es beim nächsten Mal besser zu machen:

- Mehr als 3 richtige Antworten: Es wird eine sehr gute Nachricht angezeigt, was auf eine hervorragende Leistung hinweist.
- Score zwischen 1 und 3: Eine gute Nachricht wird angezeigt, was bedeutet, dass die Leistung positiv war, aber noch Raum für Verbesserung besteht.
- Score von weniger als 1: Es wird eine eher schlechte Nachricht angezeigt, was darauf hindeutet, dass die Leistung verbessert werden muss.

Bei der Ausgabe des Punktestands wird auf die Länge des `questions[]`-Arrays zugegriffen, um einen Anteil der richtigen Antworten an der Anzahl an Fragen zu bilden.

#### Syntax

```
1  showResult();
```

### 1.4. `startTimer(time)`

Startet den Timer für das Quiz.

#### Syntax

```
1  startTimer(time);
```

#### Parameter

- *time*: Die Zeit, die für den Timer festgelegt werden soll

### 1.5. `startTimerLine(time)`

Startet die Timerleiste, um den Fortschritt des Timers visuell darzustellen.

#### Syntax

```
1  startTimerLine(time);
```

#### Parameter

- *time*: Der aktuelle Zeitwert für die Timerleiste

### 1.6. `queCounter(index)`

Aktualisiert den Fragenzähler, um die aktuelle Fragenummer anzuzeigen.

#### Syntax

```
1  queCounter(index);
```

#### Parameter

- *index*: Die Nummer der aktuellen Frage

# 2. Beschreibung der Logik des Programms

## 2.1. Hauptalgorithmus

In der digitalen Welt, wo Interaktivität und Benutzerfreundlichkeit im Vordergrund stehen, ist die klare Darstellung von Programmabläufen unerlässlich. Ein solches Beispiel ist das Flussdiagramm eines Quiz-Programms, das nicht nur die technische Logik, sondern auch die Benutzererfahrung widerspiegelt. In diesem Text werde ich die Logik dieses Programms anhand des bereitgestellten Flussdiagramms (Abbildung 3) detailliert erläutern.

Das Programm beginnt mit einem simplen Startpunkt, der als Einladung an den Benutzer dient, sich auf das bevorstehende Quiz einzulassen. Unmittelbar darauf folgt die Anzeige von Informationen, die als eine Art Begrüßungsbildschirm fungieren kann, auf dem die Spielregeln oder interessante Fakten zum Quiz präsentiert werden. Diese Phase ist entscheidend, da sie den ersten Eindruck des Benutzers prägt und das Interesse weckt.

Nach der Informationsanzeige steht der Spieler vor einer Entscheidung: Soll das Quiz gestartet werden? Diese Entscheidung ist interaktiv gestaltet, sodass der Spieler aktiv in den Prozess eingebunden wird. Entscheidet sich der Spieler gegen den Start, wird er erneut zur Informationsanzeige geleitet – ein Kreislauf, der Geduld und Flexibilität seitens des Programms zeigt.

Entscheidet sich der Spieler jedoch für "Ja", beginnt das eigentliche Herzstück des Programms – das Quiz. Eine Frage wird angezeigt, und der Spieler muss innerhalb einer vorgegebenen Zeit eine Antwort auswählen. Diese Phase ist geprägt von Spannung und Schnelligkeit, da der Spieler unter Zeitdruck steht. Läuft die Zeit ab, ohne dass eine Antwort ausgewählt wurde, endet die Frage mit der Anzeige des Resultats. Wurde jedoch eine Antwort ausgewählt, prüft das Programm, ob es sich um die letzte Frage handelt oder ob weitere Fragen folgen.

Ist die aktuelle Frage die letzte, wird das Resultat des gesamten Quiz angezeigt. Sind jedoch weitere Fragen vorhanden, wird die nächste Frage präsentiert. Dieser Zyklus aus Frage und Antwort setzt sich fort, bis alle Fragen beantwortet sind. Das Resultat bietet dem Spieler eine Rückmeldung über seine Leistung und kann als Motivation dienen, sich zu verbessern oder das Gelernte zu festigen.

Nach Abschluss des Quiz steht der Spieler vor einer weiteren Entscheidung: Soll das Quiz wiederholt oder beendet werden? Die Option, das Quiz zu wiederholen, ermöglicht es dem Spieler, aus Fehlern zu lernen und sich zu verbessern. Die Option, das Quiz zu beenden, führt zum Ende des Programms. Dieser letzte Schritt symbolisiert den Abschluss eines Lernzyklus und gibt dem Spieler die Kontrolle über seine Lernerfahrung.

## 2.2. Zählen der richtigen Antworten

Zu Beginn erfolgt die Initialisierungsphase, in der alle notwendigen Variablen und Zustände für das Quiz festgelegt werden. Diese umfasst beispielsweise das Zurücksetzen des Zählers für die richtigen Antworten und das Abrufen der ausgewählten Option des Users sowie der richtigen Antwort aus dem `questions[]`-Array (Abb. 1).

Sobald die Initialisierung abgeschlossen ist, stellt das Programm die zentrale Frage: Wurde eine Frage beantwortet? Diese Entscheidungsschleife ist das Herzstück des Programms, da sie bestimmt, wie das Programm auf Benutzereingaben reagiert. Wenn keine Antwort gegeben wurde, kehrt das Programm zur Initialisierungsphase zurück, um auf die nächste Aktion des Benutzers zu warten. Dies gewährleistet, dass das Programm in einem bereiten Zustand bleibt und auf die Interaktion des Benutzers reagieren kann.

Wurde jedoch eine Antwort gegeben, prüft das Programm, ob diese Antwort richtig ist. Diese Überprüfung ist entscheidend, da sie die Genauigkeit des Quiz sicherstellt. Bei einer richtigen Antwort wird der Zähler für die richtigen Ergebnisse erhöht und die Box der korrekten Antwort erhält einen grünen Hintergrund. Dieser Schritt ist ein wesentlicher Bestandteil des Feedback-Mechanismus, der dem Benutzer eine direkte Rückmeldung über seine Leistung gibt [Ren00].

Ist die Antwort jedoch falsch oder wurde der Zähler bereits erhöht, wird zunächst die richtige Antwort grün markiert und das Programm geht weiter zur nächsten Frage. Dieser Übergang ist nahtlos und stellt sicher, dass das Quiz ohne Unterbrechung fortgesetzt wird. Die Schleife, die von der Überprüfung der Antwort zurück zur zentralen Frage führt, ermöglicht es dem Programm, durch das gesamte Quiz zu navigieren, bis alle Fragen beantwortet wurden.

## 2.3. Auswertung der Ergebnisse

Zunächst startet die Funktion mit einer Initialisierungsphase, in der das System vorbereitet wird, um die Ergebnisse des Benutzers zu empfangen und zu verarbeiten. Diese Phase beinhaltet das Ausblenden der Info-Box und Quiz-Box sowie das Anzeigen der Ergebnis-Box (Abb. 2).

Nach der Initialisierung tritt das Programm in die entscheidende Phase der Score-Überprüfung ein. Hier wird der vom Benutzer erzielte Score mit festgelegten Schwellenwerten verglichen, um die Qualität der Leistung zu bestimmen. Abhängig vom Ergebnis dieses Vergleichs wird eine von drei Nachrichten angezeigt: eine gute, eine mittlere oder eine schlechte Nachricht.

Wenn der Score des Benutzers größer als drei ist, zeigt das Programm eine gute Nachricht an. Diese Nachricht könnte eine Gratulation oder eine positive Rückmeldung enthalten, die den Benutzer für seine hervorragende Leistung lobt. Es ist ein Moment der Anerkennung und des Erfolgs, der den Benutzer motivieren soll, weiterhin gute Leistungen zu erbringen.

Liegt der Score zwischen einschließlich zwei und drei, wird eine mittlere Nachricht angezeigt. Diese Nachricht ist wahrscheinlich konstruktiv und ermutigend, weist aber darauf hin, dass noch Raum für Verbesserungen besteht. Sie dient dazu, den Benutzer anzuspornen, sich weiter anzustrengen und seine Fähigkeiten zu verbessern.

Ist der Score jedoch gleich oder kleiner als eins, wird eine schlechte Nachricht präsentiert. Diese Nachricht könnte Hinweise auf Bereiche enthalten, die einer Verbesserung bedürfen, und bietet möglicherweise Ratschläge oder Ressourcen zur Unterstützung des Benutzers bei der Verbesserung seiner Leistung.

Unabhängig von der Art der Nachricht, die angezeigt wird, führen alle Wege zum Ende der Funktion.

# 3. Testdurchlauf

Testdurchlauf durch: _____

Name, Vorname

Notiere deine Eindrücke und Feststellungen in maximal drei Stichpunkten:

- _____

  _____

- _____

  _____

- _____

  _____

# Literatur

[Die] Diercke. *Sachsen Physisch*. https://media.diercke.net/omeda/88814__Sachsen_physisch.pdf. Zuletzt abgerufen am 2024-04-18.

[Ren00] Karen Renaud. *Feedback in Human-Computer Interaction: Characteristics and Recommendations*. https://www.researchgate.net/profile/Karen-Renaud/publication/220102904_Feedback_in_human-computer_interaction_-_characteristics_and_recommendations/links/0c960533bef7033d7d000000/Feedback-in-human-computer-interaction-characteristics-and-recommendations.pdf. Zuletzt abgerufen am 2024-04-18. 2000.

# A. Appendix

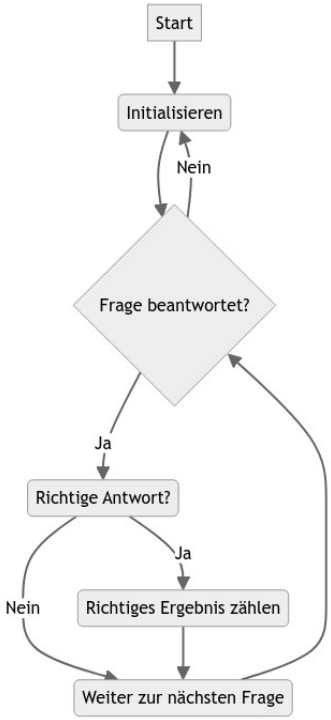


Abbildung 1: Präzisionsmechanismus zur Auswertung von Quizantworten: Ein systematischer Ansatz zur Erfassung und Bewertung der Benutzerleistung im interaktiven Quizkontext.

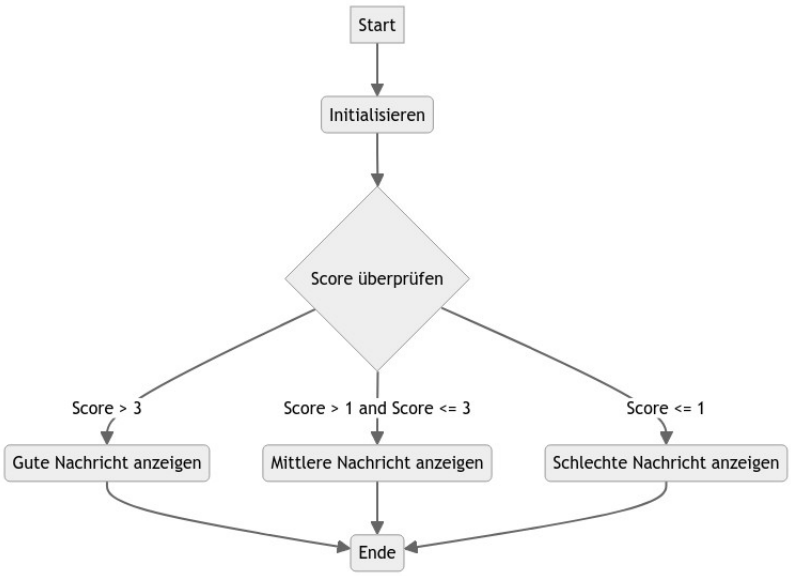


Abbildung 2: Stufenweise Bewertung und differenziertes Feedback: Ein Flussdiagramm, das den Prozess der Leistungsauswertung und die darauf basierende individuelle Rückmeldung im Bildungskontext visualisiert.

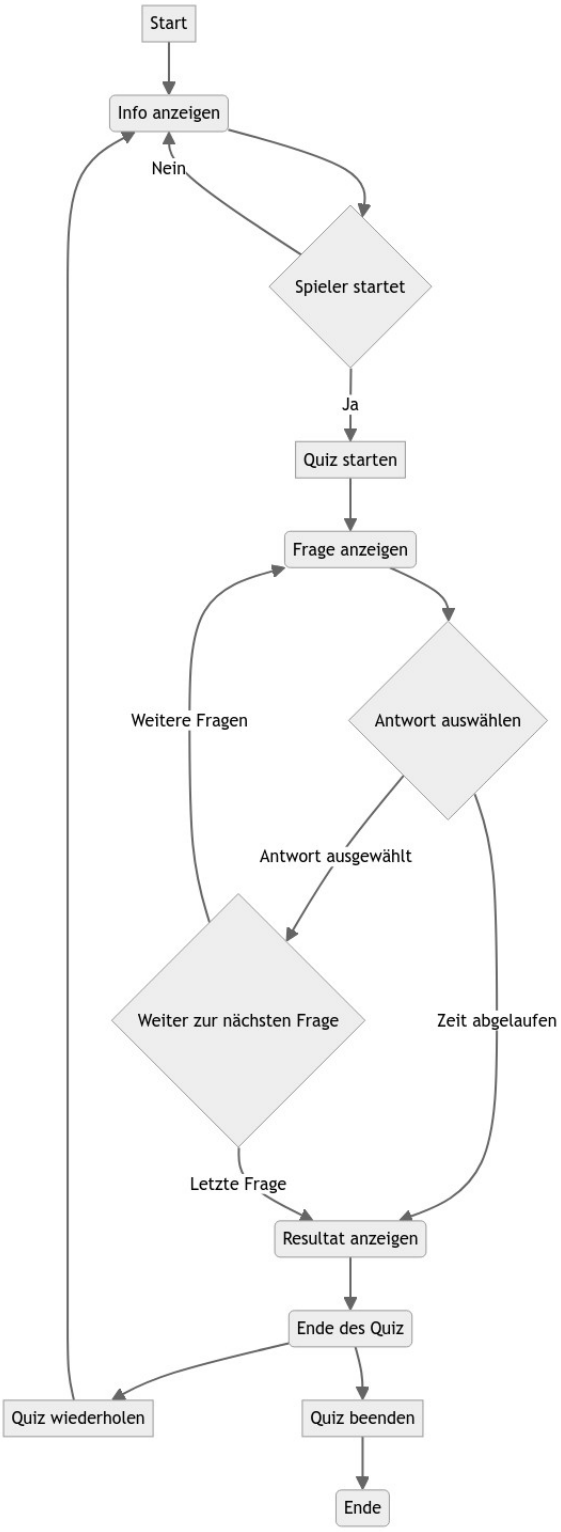


Abbildung 3: Interaktiver Ablauf des Quiz-Programms: Von der Benutzerführung bis zur Ergebnispräsentation – ein Wegweiser durch die digitale Lernerfahrung