

# NSC-GENERATOR FÜR PEN & PAPER- SYSTEME

## INHALT

Kurzbeschreibung .....	2
Warum ein NSC-Generator? .....	2
Wie funktioniert das? .....	2
Was verbirgt sich hinter dem Begriff "Fragenpool"? .....	2
Was passiert mit dem Grundfragenpool im Laufe der Generierung? .....	3
Beispiel .....	3
Installationsanweisung .....	3
Bedienungsanleitung .....	4
Systemarchitektur .....	4
Beschreibung eines technischen Teilaspekts .....	5
Programmablauf .....	5
Wichtige Funktionen .....	6
Fazit und Auswertung .....	7
Planung und Aufgabenverteilung .....	7
Programmierung .....	7
Geplante Features .....	7
Werteberechnung und OnChange-Ausgabe .....	7
Kooperationsmodus .....	8
Account-Verwaltung und Bibliothek .....	8

# KURZBESCHREIBUNG

## Warum ein NSC-Generator?

Als Spielleiter von Rollenspielsystemen (wie bspw. D&D oder DSA) ist es für fast jedes eigene Abenteuer nötig, sich Nichtspielercharaktere auszudenken. Diese spielen häufig nur in einem Spieldurchgang mit und haben oft einzig den Sinn die Welt lebendiger erscheinen zu lassen. Die Erstellung einer stimmigen Hintergrundgeschichte für eine solche Figur ist oft mit einem hohen Zeitaufwand verbunden.

Das Programm soll dem Nutzer bei der Erstellung der erzählerischen Anteile helfen und den kreativen Prozess, durch zur Figur passende Vorschläge und Fragestellungen, vereinfachen.

## Wie funktioniert das?

- der Nutzer macht Angaben zu seiner Wunschfigur in der Vorauswahl
- diese Angaben werden in Form von **Tags** abgespeichert und bestimmen den **individuell-generierten Fragenpool (igFP)** und den Detailgrad der Fragen

## Was verbirgt sich hinter dem Begriff "Fragenpool"?

- ein Fragenpool besteht aus **Gruppierungen von Fragen/Antwort-Objekten**
- der **Grundfragenpool** enthält **alle** Fragen/Antwort-Objekte
- der **individuell-generierten Fragenpool** basiert auf diesem, streicht aber alle Fragen/Antwort-Objekte raus, die den aktuellen **Tags** widersprechen

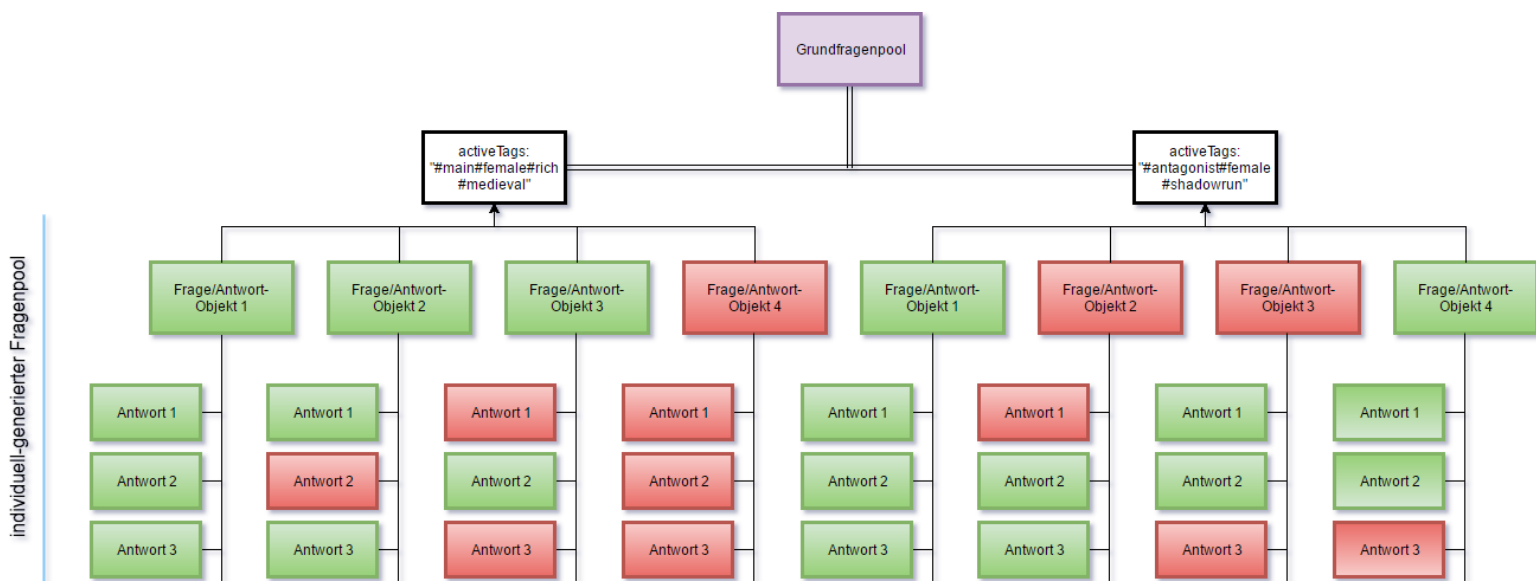


Abbildung 1: Beispiel der Fragenverwaltung

## *Was passiert mit dem Grundfragenpool im Laufe der Generierung?*

- während der **Frage/Antwort-Phase** - also während der Nutzer passende Antworten für seine Figur auswählen kann - bearbeitet die Anwendung kontinuierlich die aktiven **Tags** und somit den igFP
- das führt zu einem zunehmend kleiner-werdenden, personalisierten Pool
- Fragen, die nicht zur Figur passen (weil bspw. angegeben wurde, dass die Figur adelig ist) werden nicht angezeigt; dasselbe gilt für Antworten, sodass nur relevante Optionen ausgegeben werden
- bis zum Ende hin verkleinert sich der Fragenpool also gravierend und wird tatsächlich individuell auf die Figur abgestimmt; je nach Figur werden also unterschiedliche Fragen/Antwort-Objekte durchlaufen

## *Beispiel*

Rolle und Typ bestimmen die Art des Grundfragenpools und den Detailgrad

- *Hauptfigur (Typ: Antagonist) erhält andere und detailliertere Fragen als eine neutral-gesinnte Nebenfigur*

Aufgrund der Vorauswahl von z.B. Rasse und Spielsystem fallen einige unpassende Fragen direkt zu Anfang weg

- *Wird NICHT Sci-Fi gewählt, fallen alle Fragen/Antworten weg, die ausschließlich in einen Sci-Fi-Hintergrund passen*

## INSTALLATIONSANWEISUNG

Die Anwendung muss nicht installiert werden. In ihrem aktuellen Zustand ist sie am einfachsten über den [JSFiddle-Link](#) aufzurufen. Bitte nutzen Sie dafür Google Chrome oder Mozilla Firefox. Dort können Funktionen überprüft werden. Mit F12 lässt sich die Browserkonsole öffnen, um Probeausgaben einsehen zu können.

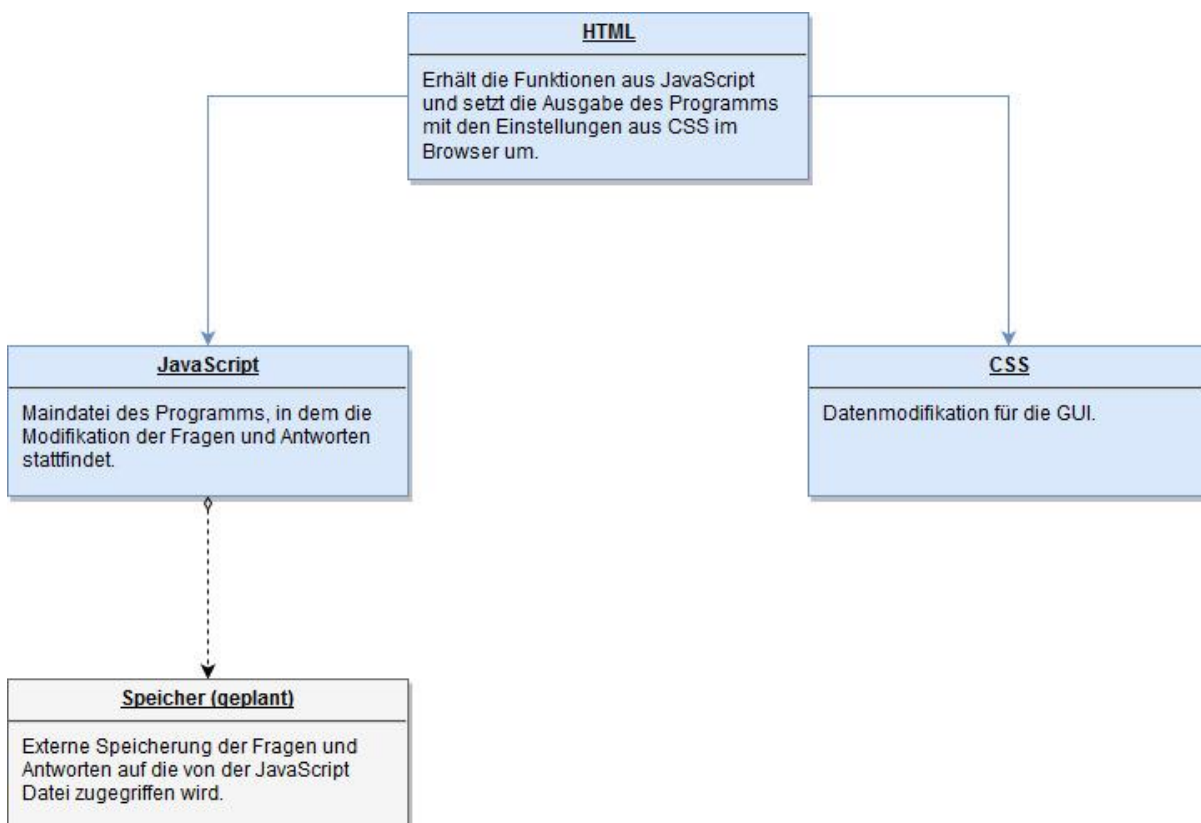
## BEDIENUNGSANLEITUNG

In der Vorauswahl müssen Angaben zur Figur gemacht werden. Anhand dieser Angaben wird der individuelle Fragenpool generiert, der **Fragen sowie Antworten** bestimmt. Nach dem Bestätigen öffnet sich das Fragenfenster.

Der Generator gibt die erste Frage der ersten Kategorie aus. Eine Antwort wird ausgewählt und bestätigt. Es erfolgt die Ausgabe der nächsten Frage. Der individuelle Fragenpool passt sich der Auswahl an und wird um jede weitere Antwort kleiner und personalisierter als zuvor.

Der Button "Neu / Zurücksetzen" löscht den aktuellen Stand, setzt die Kategorie zurück und lässt Änderungen an der Vorauswahl zu.

## SYSTEMARCHITEKTUR



# BESCHREIBUNG EINES TECHNISCHEN TEILASPEKTS

## Programmablauf

### 1. Nutzer startet die Erstellung und beendet die Vorauswahl

- a. `newCharacter()` sorgt dafür, dass der Kategorienzähler und der Fragenzähler zurückgesetzt werden
- b. Es startet die Vorauswahl:
  - i. `confirmPreselection()` nimmt die Auswahl des Users an
  - ii. `modifyTags()` = `activeTags+=preSelection` (mit Anti-Dopplungsfunktion)
- c. das Array `allQuestions` bzw. die nächste Kategorie wird gemischt

### 2. Die Gültigkeit wird angepasst

- a. `updateValidity()` durchsucht `allQuestions` bzw. das Kategorienarray mit `checkForMatch`:
  - i. wenn eine Übereinstimmung besteht, wird `valid` auf **false** gesetzt
  - ii. wenn keine Übereinstimmung besteht, passiert **nichts**

### 3. Kategorie wird mit `checkIfCatGoesUp()` überprüft

- a. Grundwert für `cat` = 1;
  - i. wenn eine vorher definierte Anzahl von Fragen aus Kategorie 1 ausgegeben- und beantwortet wurden (Zähler `questionCount++`) **oder** wenn es **keine** Fragen aus Kategorie 1 mehr gibt, schaltet `cat++`;
  - ii. sobald die zuvor definierte Endkategorie erreicht wurde, erfolgt die **Werteausgabe**

### 4. Ausgabe der ersten/nächsten Frage und der Antwortmöglichkeiten mit `nextQuestion()`:

- a. Frage wird ausgegeben
- b. Antworten, die im Fragen-Objekt gespeichert sind, werden ausgegeben
  - i. Ermittlung der Anzahl der Antworten, übrige radio-Felder müssen auf `hidden` geschaltet werden und nicht anklickbar gemacht werden

### 5. User kann jetzt eine Antwort auswählen

- a. User wählt eine Antwort per Klick aus, `showModifier()`;
  - i. visuelle Ausgabe für den Fall einer Modifikator-Änderung erfolgt
- b. User bestätigt die Antwort mit einem anderen Knopf `setAnswer()`;
  - i. `modifyTags()` wird ausgeführt, um `activeTags` um die Tags der Antwort zu erweitern
  - ii. `Valid` der Frage wird auf "false" gesetzt, sodass sie nicht noch einmal vorkommen kann
  - iii. Fragencounter wird um eins hochgezählt

### 6. zurück zu Nummer 2)

## Wichtige Funktionen

### startGenerator()

- Setzt currentCategory, AnswerQuestionCounter und das historyArray zurück; führt dann nextQuestion() aus und gibt die erste Frage aus

### confirmPreselection()

- Funktion, die die Vorauswahl-variablen zuordnet und sie den aktiven Tags hinzufügt

### nextQuestion()

- gibt die nächste gültige Frage aus und ändert die RadioButtons entsprechend ab

### addQuestion(category, txt, exclusionTags)

- erstellt ein Fragenobjekt und pusht es in allQuestions
- `new addQuestion(1, "Wie sind die Eltern deiner Figur ums Leben gekommen?", "parentsAlive#statist#parentsUnknown");`

### addQuestion.prototype.addAnswer(answerTxt, answerExclusionTags, tagsToAdd)

- Funktion innerhalb von addQuestion, die es erlaubt neue Fragen in das entsprechende answerArray des addQuestions-Objekts zu speichern
- `allQuestions[1].addAnswer("Dorf", "#", "#OriginVillage");`

### modifyTags(tagsToBeAdded)

- Funktion, die activeTags um weitere Tags erweitert
- `modifyTags("#er#zwo#uno#mah");`

### checkForMatch(exclusionTags, activeTags)

- Funktion, die überprüft, ob es eine Übereinstimmung zwischen den ausschließenden- und den aktiven Tags gibt; wenn ja, wird true ausgegeben

### updateValidity(arrayToLookThrough, activeTagsToCheck)

- Funktion, die mithilfe von checkForMatch die Gültigkeit ändert
- `out(updateValidity(allQuestions, activeTags));`

### checkValidity(arrayToLookThrough)

- Funktion, die nur die derzeit gültigen Fragen ausspuckt
- `out(checkValidity(allQuestions));`

### checkIfCatGoesUp()

- Funktion, die überprüft ob ein Wechsel in die nächste Kategorie erfolgt ist oder ob die letzte Kategorie abgeschlossen wurde; wenn ja: Ausgabe erfolgt mit showResults();

### applyToAll(categoryArray, functionName, optionalActiveTags)

- Führt eine Funktion für alle bestehenden Kategorien aus
- `applyToAll(categoryArray, checkValidity);`

## FAZIT UND AUSWERTUNG

### *Planung und Aufgabenverteilung*

Allgemein verlief das Projekt wie von uns erhofft. Wir konnten uns schnell auf ein Ziel für unser Programm einigen und auf Essenzielles fokussieren. Vorerst haben wir die ursprüngliche Idee für das Projekt 1 von Tim übernommen.

Nach einer effizienten Konzeption haben wir uns für ein Tagsystem zur Filterung des Fragenpools entschieden und die Aufgaben strikt in Programmierung und Texterstellung sowie Redaktion getrennt. An fixen Terminen haben wir den aktuellen Stand besprochen, gegenseitig kritisiert und zusammen bearbeitet.

### *Programmierung*

Die umfangreiche Planungsphase, samt Konzeptionierung, war für die spätere Programmierarbeit wichtig. In Anbetracht dessen, dass dies unser erster Schritt zu einem umfassenden, professionellen Programm war und wir zuvor keine Erfahrung mit Javascript, HTML und CSS hatten, sind wir mit dem aktuellen Stand des Projekts sehr zufrieden. Das Hauptaugenmerk liegt auf dem Javascriptteil. In Bezug auf die Deadline haben wir die Zeit gut eingehalten. Alle elementaren Funktionen, die geplant waren, haben es in das Programm geschafft.

### *Geplante Features*

Einige Ideen, die uns bei der Konzeptionierung in den Sinn gekommen sind, sowie die Vereinigung beider Projektideen, haben wir bewusst nicht in das Programm aufgenommen oder zunächst nach hinten verschoben. Anfangs hatte Leonard den Einfall, es dem Nutzer zu ermöglichen Fragen und Antworten zu erstellen und dem Grundfragenpool hinzuzufügen, also eine Speicherbearbeitung. Allerdings haben wir entschieden, dass der Zeitaufwand und fehlende Erfahrung diesen Aspekt zu komplex für uns macht.

Sollten wir am Projekt weiterarbeiten, würden wir uns mit JQuery auseinandersetzen und eine OnChange-Funktion für die Werteberechnung einbauen. Des Weiteren könnten wir uns folgende Möglichkeiten vorstellen:

### *Werteberechnung und OnChange-Ausgabe*

- erfolgt nach der Generierung der erzählerischen Anteile
- bezieht sich direkt auf die Antworten der Fragen/Antwort-Phase; also muss hier mit Countern o.Ä. gearbeitet werden

*Beispiel: Figur stammt vom Land, geht einem körperlichen Beruf nach und treibt viel Sport oder Ähnliches – hier würde der Counter für das Attribut Stärke/Gewandtheit hochzählen und sich am Ende als positiver Modifikator auf den Würfelwurf auswirken.*

### *Kooperationsmodus*

- für beliebig viele Nutzer
- richtet sich an Spieler und Spielleiter
- gemeinsam können
  - Kontakte von Spielerfiguren erstellt werden
  - neue Figuren angelegt werden

### *Account-Verwaltung und Bibliothek*

- erstellte Figuren können in der Bibliothek gespeichert, geteilt und eingesehen werden
- Suchfunktion für öffentliche Inhalte
- Figuren sind bewertbar, kommentierbar und importierbar