

Dokumentation zur Web-Applikation „Dice“

Author: Britta Vieth

Inhalt

Kurzbeschreibung der Funktionalität	2
Technik	2
Bestandteile.....	2
Model	3
Ablauf – Bild	4
Ablauf – Beschreibung.....	5
Funktionen.....	6
highscore	6
game	6
Template dice.html und Filter	7
Template.....	7
Filter.....	7

Kurzbeschreibung der Funktionalität

Dice simuliert ein Würfelspiel, das mit fünf Würfeln gespielt wird und in dem es darum geht, in dreizehn Kategorien jeweils möglichst viele Punkte zu erzielen, um auf eine hohe Gesamtpunktzahl zu kommen. Pro Runde darf ein Spieler drei Mal würfeln, jeweils mit beliebig vielen der fünf Würfel. Nach dem dritten Wurf muss er in einer der dreizehn Kategorien Punkte eintragen oder eine Kategorie „streichen“, d. h. 0 Punkte eintragen. Das Spiel ist beendet, wenn in allen Kategorien Punkte eingetragen wurden.

Außerdem werden die fünf besten Spieler in einer Bestenliste angezeigt.

Technik

Python 3.8

Django 3.1.4

PostgreSQL

Bestandteile

Das Programm „dice“ enthält die Apps „highscore“ und „game“;
„highscore“ ist der Einstiegspunkt, potenziell auch für weitere Apps, und zeigt eine Bestenliste der User mit den höchsten Punktzahlen an,
„game“ enthält die eigentlichen Funktionalitäten des Spiels.

Funktionen sind in den jeweiligen views.py-Dateien zu finden.

Der Ordner templates enthält die Templates beider Apps.

Das Datenmodell ist in game/models.py definiert.

Die Datei game/templatetags/event_tags.py enthält zwei Filter, die im Template dice.html genutzt werden.

Der Ordner static enthält die Unterordner css, img und js mit entsprechenden Dateien.

Model

Vereinfachte Darstellung der Datenstruktur

User	
name	Charfield(max_length=30, alphanumeric)
score	IntegerField(default=0)

Game	
user	ForeignKey(User, on_delete=CASCADE)
score_sheet	Dictionary als JSONField
dice	Array als JSONField
round	IntegerField(default=0)

Anmerkungen zu den Attributen:

score_sheet: Im Dictionary score_sheet werden den Kategorien des Spiels (,triple', ,fullhouse' etc.) die jeweiligen Punkte des aktuellen Spiels zugeordnet, dabei gilt:

- 1000 Punkte => noch keine Punkte für diese Kategorie eingetragen
- negative Zahl => bereits Punkte gespeichert, der Wert ergibt sich dabei aus dem eigentlichen Punktwert abzüglich einer festgelegten Zahl (,that_number'), aktuell 400

Diese Speicherung dient im Spielverlauf der Unterscheidung zwischen möglichen Punkten und bereits gespeicherten Punkten (siehe Funktion roll_dice)

dice: Ein Array der Größe 5, in dem die aktuellen Werte der fünf Würfel gespeichert werden, z. B. [3, 3, 3, 3, 3], wenn ausschließlich Dreien gewürfelt wurden

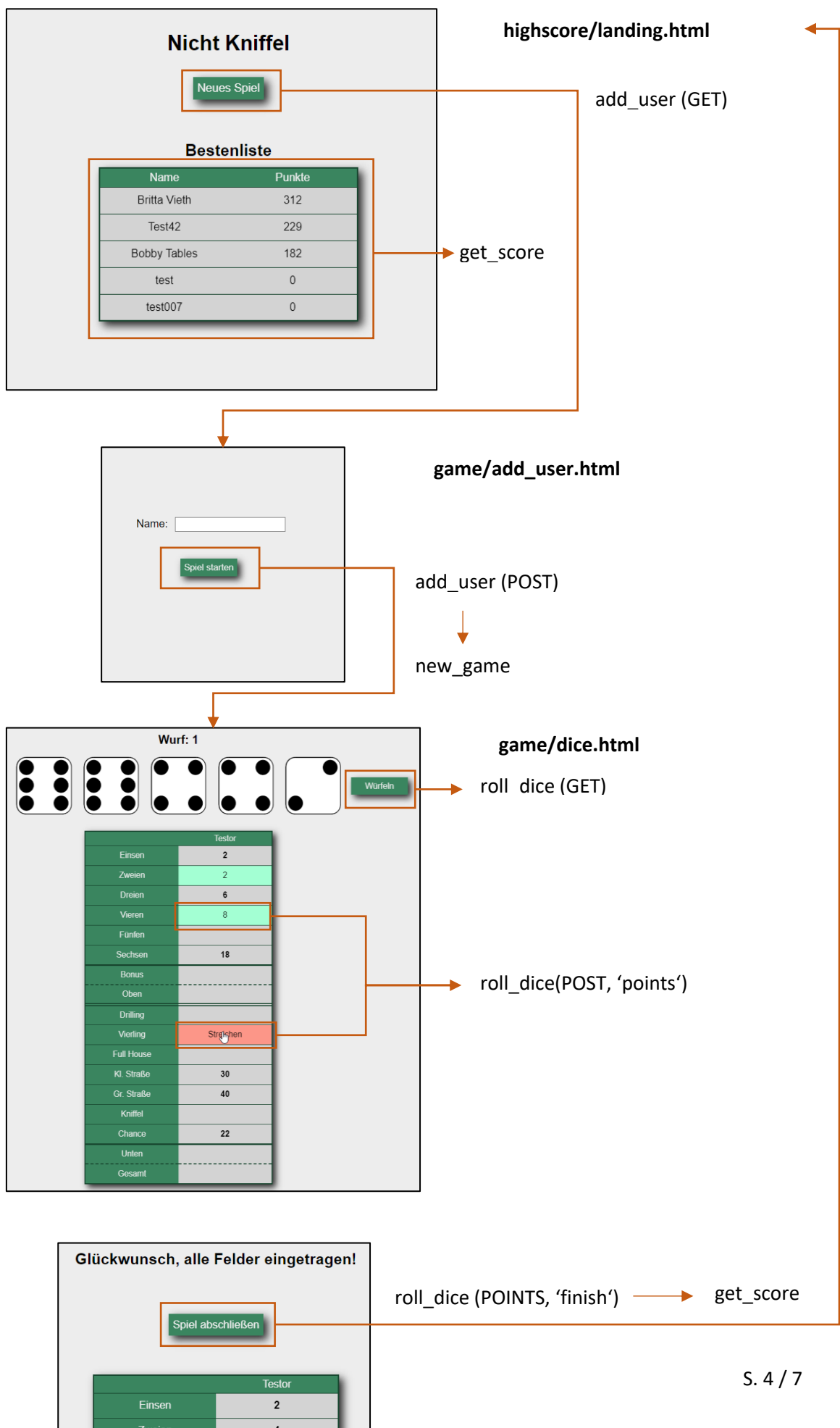
round: Die Rundenzahl wird nach jedem Würfeln oder Eintragen von Punkten um 1 erhöht. Für Funktionen und Templates ist jedoch nicht die tatsächliche Rundenzahl wichtig, sondern ,round mod 4', es gibt also die Runden

- 1: vor dem ersten Wurf -> nur würfeln möglich,
- 2: nach dem ersten Wurf -> Würfel anklicken, Punkte eintragen oder würfeln möglich,
- 3: nach dem zweiten Wurf -> Würfel anklicken, Punkte eintragen oder würfeln möglich,
- 0: nach dem dritten Wurf -> nur Punkte eintragen möglich.

Näheres zur Bedeutung in den Beschreibungen zu roll_dice und dice.html.

Ablauf – Bild

(textliche Beschreibung auf der folgenden Seite)



Ablauf – Beschreibung

Erklärung des Spielablaufs (bildlich auf der vorherigen Seite dargestellt)

Einstieg (URL: `/`, Template: `landing.html`) Die Landing Page in der App „highscore“

→ Die Funktion `get_score` wird aufgerufen und liefert die fünf Spieler mit der höchsten Punktzahl zurück, diese werden in einer Tabelle als Bestenliste dargestellt

→ Der Button „Neues Spiel“ leitet mit einem GET-Request auf die Funktion `add_user` in der App `game` weiter

User hinzufügen (URL: `/game/add_user/`, Template: `add_user.html`)

→ `UsersForm` wird geladen und ein Feld zum Eintragen des Namens eines neuen Nutzers dargestellt

→ Der Button „Spiel starten“ sendet einen POST-Request an die Funktion `add_user`, diese speichert den User in der Datenbank mit dem `score`-Defaultwert 0 und dem eingegebenen Nutzernamen (nur alphanumerische Zeichen und Leerzeichen möglich) und leitet, sofern der Nutzername gültig ist, an `new_game` weiter mit der User-ID als Argument; andernfalls wird eine Error-Message ausgegeben

→ `new_game` erstellt ein neues Spiel und leitet unter Angabe der Game-ID an `roll_dice` weiter

Das Spiel (URL: `/game/<int:game_id>/`, Template: `dice.html`)

→ Je nach Runde zeigt das Template Folgendes an

→ Runde 1:

- Images von neutralen Würfeln (ohne Zahlen), nicht klickbar
- Tabelle mit den bisher gespeicherten Punkten
- Button „Würfeln“ → sendet GET-Request an `roll_dice`
 - `roll_dice` erzeugt für alle nicht angeklickten Würfel (in dieser Runde also alle) einen Zufallswert zwischen 1 und 6 (inklusive)
 - für die so erzeugten Werte werden anschließend für alle Tabellenfelder, in denen bisher kein Wert gespeichert ist, die jeweils mögliche Punktzahl berechnet
 - alle möglichen und bereits gespeicherten Punkte werden in der Variablen `scores` an das Template übergeben

→ Runden 2 und 3:

- Labels, die Images enthalten, die die gewürfelten Zahlen darstellen und anklickbar sind, das Anklicken kreuzt eine nicht sichtbare Input-Checkbox an
- Tabelle mit den bisher gespeicherten Punkten und den möglichen Punkten
 - auf Feldern mit möglichen Punkten und ohne Punkte liegt jeweils ein nicht sichtbarer Button, der einen POST-Request (enthält Wert `points`) an `roll_dice` sendet, sodass die entsprechenden Punkte (bzw. 0 Punkte) eingetragen werden; der Runden-Counter wird dann auf Runde 1 zurückgesetzt
- Button „Würfeln“ → sendet GET-Request an `roll_dice`

→ Runde 4:

- Images von Würfeln, die die gewürfelten Zahlen darstellen, nicht klickbar
 - Tabelle mit den bisher gespeicherten Punkten und den möglichen Punkten
 - kein Button, in dieser Runde muss ein Wert eingetragen werden
- Wenn in allen Feldern Werte eingetragen wurden, gilt das Spiel als beendet:

- Tabelle mit allen eingetragenen Werten
- Button „Spiel abschließen“, der einen POST-Request an `roll_dice` sendet
 - der Request enthält den Wert `finish`, daher wird der Endpunktestand mit der aktuellen Bestenliste verglichen; ist der Wert höher als der niedrigste Score in der Bestenliste, wird der Endpunktestand im User-Objekt gespeichert, andernfalls wird der User gelöscht;
 - anschließend wird das Game-Objekt gelöscht und auf die Landingpage (`get_score`) weitergeleitet

Funktionen

highscore

`get_score(request)`: liefert die fünf User-Objekte mit der höchsten Punktzahl zurück

game

`add_user(request)`:

- GET: Leitet auf das Template `game/add_user.html` und übergibt das `UsersForm` mit einer leeren User-Instanz
- POST: überprüft Validität des Forms, legt ggf. einen neuen User an und leitet an die URL mit dem Namen `,new_game'` weiter mit der ID des neuen Users als Argument

`new_game(request, user_id)`: Legt ein neues Game-Objekt an mit der übergebenen User-ID als Foreign-Key und leitet an die URL mit dem Namen `,roll'` weiter mit der ID des neuen Games als Argument

`roll_dice(request, game_id)`:

- POST und Request enthält `,points'`: Im `score_sheet` des Game-Objekts mit der übergebenen ID wird für eine Kategorie der Punktwert eingetragen. Die Kategorie wird aus dem Value des geklickten Buttons ausgelesen und der Punktwert aus dem zugehörigen Input-Feld. Der eingetragene Wert ergibt sich aus dem ausgelesenen Punktwert abzüglich der Zahl, die in `,that_number'` gespeichert ist.
- POST und Request enthält `,finish'`: Der Endpunktestand wird mit der aktuellen Bestenliste verglichen; ist der Wert höher als der niedrigste Score in der Bestenliste, wird der Endpunktestand im User-Objekt gespeichert, andernfalls wird der User gelöscht; anschließend wird das Game-Objekt gelöscht und auf die Landingpage (`get_score`) weitergeleitet.
- GET: in Runde 0, also vor dem ersten Wurf, werden nur die aktuell gespeicherten Punkte an das Template weitergegeben, in den anderen Runden wird ermittelt, welche Würfel angeklickt wurden, und für alle nicht angeklickten Würfel ein neuer Zufallswert zwischen 1 und 6 erzeugt. Die gewürfelten Werte werden ermittelt und anschließend wird für alle Kategorien, in denen bisher kein Wert gespeichert ist, ausgewertet, welche Punkte mit diesen Würfeln möglich sind. Gespeicherte und mögliche Punkte werden in der Variablen `score` an das Template `dice.html` weitergegeben.

Template dice.html und Filter

Template

Das Template besteht aus drei Hauptabschnitten, die je nach Rundenzahl und Spielzustand variieren können:

- h3-Überschrift, die dem Spieler Hinweise geben, z. B. „Spieler1, bitte würfeln“
- Interaktionsteil, der Würfel und/oder einen Button enthält
- Tabelle, die die gespeicherten Punkte und ggf. mögliche Punkte anzeigt und je nach Runde ebenfalls interaktiv sein kann

Argumente

- page_title: im base.html als <title> eingebunden
- user: Name des aktuellen Users, für die Anrede im Seitentext und als Tabellenüberschrift
- dice: enthält die aktuellen Werte der fünf Würfel, zur Darstellung der korrekten Images
- scores: enthält für alle Kategorien die gespeicherten und ggf. möglichen Punkte, die in der Tabelle angezeigt werden, dabei gilt:
 - o Punkte = 1000 => keine Punkte gespeichert, keine Punkte möglich
 - o $0 < \text{Punkte} < 1000$ => keine Punkte gespeichert, aber Punkte möglich
 - o Punkte < 0 => bereits Punkte gespeichert
- round: hat den Wert 0, 1, 2 oder 3, je nach Runde wird die Darstellung entsprechend angepasst:
 - o 1: vor dem ersten Wurf, Würfel ohne Zahlenwerte, nicht klickbar, Button zum Würfeln, Tabelle zeigt gespeicherte Werte und ist nicht interaktiv
 - o 2 und 3: nach dem ersten bzw. zweiten Wurf, Würfel zeigen gewürfelte Werte, lassen sich markieren, Button zum Würfeln, Tabelle zeigt gespeicherte und mögliche Punkte und Felder sind klickbar, um die möglichen Punkte einzutragen oder eine Kategorie zu streichen (also 0 Punkte einzutragen)
 - o 0: nach dem dritten Wurf, Würfel zeigen gewürfelte Werte, sind nicht mehr markierbar, kein Button zum Würfeln, interaktive Tabelle (wie in den beiden vorherigen Runden)
- number: Zahl, die zu den gespeicherten (negativen) Werten addiert werden muss, um die tatsächliche Punktzahl zu erhalten
- images: Dateinamen der Bilder zur Darstellung der Würfel
- upper: eine Liste von Tupeln, ein Tupel besteht aus einer Kategoriebezeichnung (z. B. „Drilling“) und der zugehörigen Backend-Kategorie (z. B. „triple“), dient der Darstellung der oberen Tabellenhälfte
- lower: wie upper, dient der Darstellung der unteren Tabellenhälfte

Filter

get_index(array, index):

Gibt array[index] zurück und dient dazu, einen Wert aus einem Array als Index für ein anderes Array zu nutzen, was Templates so nicht unterstützen, z. B.

images|get_index:dice.2

Der dritte Würfel (dice.2) hat den Wert 1, daher soll das Bild „die_1.png“ geladen werden; dieses liegt im Array „images“ am Index 1. Der Filter liefert in diesem Beispiel also images[1] zurück.

get_score(scores, pair):

Gibt scores[pair[1]] zurück. scores ist ein Dictionary, das einer Kategorie (z. B. „triple“) einen Punktwert zuordnet (z. B. 20). pair ist ein Tupel (z. B. („Drilling“, „triple“)), an dessen Index 1 die Bezeichnung der Kategorie gespeichert ist. Der Filter liefert in diesem Beispiel scores[„triple“] zurück.