

SGDb: Semantic Video Game Database

Technical Report

Anastasia Chernysheva
a.chernysheva@oth-aw.de

Jakob Götz
j.goetz@oth-aw.de

Ardian Imeraj
a.imeraj@oth-aw.de

Patrice Korinth
p.korinth@oth-aw.de

Philipp Stangl
p.stangl1@oth-aw.de

Zusammenfassung—Dieser Technical Report beschreibt die Architektur von SGDb – eine webbasierte Anwendung mit einer Graphen-basierten Suche von Videospiele.

I. EINFÜHRUNG UND ZIELE

In den weiteren Abschnitten des Technical Reports wird zuerst auf die Lösungsstrategie in Abschnitt II eingegangen. Im Nächsten Abschnitt III wird das Gesamtsystem aus Bausteinsicht beschrieben. Anschließend wird in Abschnitt IV die Verteilungssicht der Anwendung beschrieben. In Abschnitt V werden die angewandten Werkzeuge zur Entwicklung der Anwendung vorgestellt. Abschließend wird ein Fazit und Ausblick in Abschnitt VI gegeben.

II. LÖSUNGSSTRATEGIE

Dieser Abschnitt enthält einen stark verdichten Architekturüberblick. Eine Gegenüberstellung der wichtigsten Ziele und Lösungsansätze.

III. BAUSTEINSICHT

Diese Sicht zeigt die statische Zerlegung des Systems in Bausteine sowie deren Beziehungen.

A. Gesamtsystem

B. Backend

C. Frontend

Das Frontend ist für die Benutzerschnittstelle verantwortlich. Hauptbestandteile des Frontends sind die Suchmaske, der Graph und die Detailseite zu einem Videospiel. Für die Darstellung des Graphen und die Interaktion mit diesem wird Sigma.js [1] verwendet. Sigma.js rendert Graphen mit WebGL. Damit lassen sich größere Graphen schneller zeichnen als mit Canvas- oder SVG-basierten Lösungen. Das Graphenmodell wird in einer separaten Bibliothek namens Graphology [2] verwaltet. Dies ist eine Standardbibliothek mit Algorithmen aus der Graphentheorie und allgemeinen Hilfsprogrammen wie z.B. Graphengeneratoren.

Generierung Für das Graphen Layout wird der ForceAtlas2 Algorithmus [3] verwendet. Vor dem Start von ForceAtlas 2 Layout muss die Startposition jedes Knotens festgelegt werden. Daher müssen zwei Attribute namens x und y für alle Knoten des Diagramms definiert werden.

Dazu wird ein Graph Objekt mit dem Random layout erzeugt. Positionierung jedes Knotens, indem die Koordinaten gleichmäßig nach dem Zufallsprinzip auf dem Intervall [0, 1) auswählt.

Die Interaktion mit dem Backend erfolgt über die REST-Schnittstelle.

IV. VERTEILUNGSSICHT

V. ENTWICKLUNGWERKZEUGE

VI. FAZIT UND AUSBLICK

Im Rahmen der Arbeit wurde mit einem begrenzten Datensatz von 500 Spielen eine erste Version der Anwendung entwickelt.

LITERATUR

- [1] Sigma. (2022). “Sigma.js.” [Online], Adresse: <https://www.sigmaj.s.org/>.
- [2] Graphology. (2022). “Graphology.” [Online], Adresse: <https://graphology.github.io/>.
- [3] M. Jacomy, T. Venturini, S. Heymann und M. Bastian, “ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software,” *PloS one*, Jg. 9, Nr. 6, e98679, 2014.