

# Titel

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—TODO

### I. EINLEITUNG

Das erste Videospiel *Tennis for Two* aus dem Jahr 1958 ebnete den Weg für die Spieleindustrie. Über Jahrzehnten hinaus kamen immer mehr verschiedene Spiele auf den Markt, die sich in ihrer Art und Weise unterscheiden. Mit den heutigen Videospiele ist es möglich, sich mit Spielern aus der ganzen Welt zu messen oder gemeinsam zu spielen. Aus den unzähligen veröffentlichten Videospiele ist eine breite Palette an Genres entstanden. Dabei wird in der Spieleindustrie grob in sechs Kategorien unterschieden, hierbei zählen die Strategie- und Action-Spiele zu den beliebtesten Genres [1]. Zu den anderen vier Kategorien zählen hier die Rollen-, Adventure-, Simulationen- und Sonstigespiele. Jedes dieser Kategorien besitzt weitere Unterkategorien, die sich in ihrer Art und Weise unterscheiden.

Um einen besseren Überblick zu erhalten, präsentieren wir hier eine plattformunabhängige Anwendung, welche die Spiele in den jeweiligen Genres in eine übersichtliche Struktur zusammenfasst. In Abschnitt II des Konzeptpapiers werden bereits bestehende Anwendungen vorgestellt, die sich mit der Kategorisierung von Spielen beschäftigen. In Abschnitt III beschreibt die Anforderungen in Form von User Stories. Zum Schluss wird auf die Methoden eingegangen, welche in Abschnitt IV näher beschrieben werden.

### II. VERWANDTE ARBEITEN

Die Website igdb.com gibt einen Eindruck, was der Nutzer von einer Spielesuche erwarten kann. Der Funktionsumfang und die Spieleanzahl ist groß, sodass das Suchergebnis meist zufriedenstellend ist. Des Weiteren besitzt die Website eine gut dokumentierte Entwickler-API. Dies ermöglicht es Entwickler, Daten in Form von JSON, REST oder ProtoBuf zu erhalten. Jedoch setzt IGDB eine relationale Datenbank ein (kann man das irgendwo sehen?). Hier setzt das Projekt von Team Rot an. Es wird eine Web-Applikation entworfen, welche eine vergleichbare Oberfläche bietet, jedoch mit dem Zusatz, dass die Daten in die LOD-Cloud überführt werden.

### III. ANFORDERUNGEN

In der Anforderungsanalyse wurden zwei Stakeholder identifiziert: Benutzer und Entwickler. Deren Anforderungen werden in diesem Abschnitt in Form von User Stories beschrieben.

#### A. Selection von Genre

Als Benutzer möchte ich mir ein Überblick über die unterschiedlichen Genres verschaffen. Akzeptanzkriterien sind:

- Button zum Auswählen der Kategorie
- Unterkategorien werden in einer Liste angezeigt

#### B. Auflistung von Spielen

Als Benutzer möchte ich meine Genre-Favoriten eingeben und einen Vorschlag an Spielen erhalten. Akzeptanzkriterien sind:

- Eingabefeld für Favoriten
- Spiele absteigen in einer Liste anzeigen
- Button zum Suchen der Spiele

#### C. Twitter-Cards für das jeweilige Spiel

Als Benutzer möchte ich Twitter-Cards zu dem Spiel sehen, welches aktuell betrachtet wird. Akzeptanzkriterien sind:

- Aktuelle Twitter-Cards geordnet anzeigen
- Twitter-Cards fügen sich formschön in die Website-Oberfläche ein
- Es existiert eine Verlinkung zum jeweiligen Twitter Beitrag

#### D. (Optional) Abonieren von Spielen

Als Benutzer möchte ich Spiele abonieren können, damit ich über die Twitter-Cards immer auf dem neusten Stand bin. Akzeptanzkriterien sind:

- Twitter-Cards zu den abonierten Spielen werden angezeigt
- Twitter-Cards sind aktuell
- Eigener Bereich auf der Website bei dem die Twitter-Cards angezeigt werden

#### E. (Optional) Anforderung

Als (...) möchte ich (...), damit (...). Akzeptanzkriterien sind:

- Item 1
- Item 2
- Item 3

#### F. Testabdeckung

Als Entwickler möchte ich eine ausreichend hohe Testabdeckung, um Fehler frühzeitig zu erkennen. Akzeptanzkriterien sind:

- Testabdeckung von mindestens 50
- Sowohl Frontend- als auch Backend sind mit den Tests abgedeckt

#### G. (Optional) API-Nutzung

Als Entwickler möchte ich erkennen können, in welchen Umfang meine API genutzt wird, damit ich Rückschlüsse auf die Auslastung und Interessen erhalte. Akzeptanzkriterien sind:

- Wie häufig wird die API angesprochen
- Welche API-Endpunkte werden am häufigsten genutzt

#### H. (Optional) Einsehen meist aufgerufener Spiele

Als Benutzer möchte ich, dass mir die meist aufgerufenen Spiele angezeigt werden, damit ich wissen kann, welche Spiele in der Spielebranche momentan gefragt sind. Akzeptanzkriterien sind:

- Ein eigener Bereich für zuletzt aufgerufene Spiele auf der Startseite
- Spiele werden jeweils nur einmal angezeigt
- Hierarchische Anordnung nach Anzahl der Aufrufe in bestimmter Zeit

#### I. (Optional) Einsehen von Bewertungen

Als Benutzer möchte ich Bewertungen zu dem jeweiligen Spiel einsehen können, damit ich mir ein Bild über die Qualität des Spiels verschaffen kann. Akzeptanzkriterien sind:

- Eine Anzeige in Form von Sternen oder eines Prozentsatzes
- Eine Möglichkeit Spiele anhand der Bewertungen zu sortieren

## IV. METHODEN

Ziel bei der Entwicklung und bei der späteren Verwendung ist eine plattformunabhängige Anwendung, die eine Schnittstelle für die *Linked Open Data Cloud* als auch der Bedienoberfläche der Anwendung bereitstellt. Daraus ergeben sich folgende technische Schlüsselbausteine:

Für die Repräsentation der semantischen Daten im Frontend wird das Framework Svelte verwendet. Die Daten werden voraussichtlich in einer Ontotext GraphDB Instanz persistiert. Im Backend wird FastAPI für eine REST-Schnittstelle und RDFLib zum arbeiten mit RDF verwendet. Die Kommunikation zwischen Frontend und Backend wird über eine RESTful-API abgewickelt. Um eine fehlerfreie Anwendung zu entwickeln, werden geeignete Test-Frameworks für Frontend als auch Backend selektiert und verwendet. Die einzelnen Komponenten der Anwendung werden mittels Docker *containerisiert*.

## LITERATUR

- [1] Statistica. “Statistica.” [Online]. (2022), Adresse: <https://de.statista.com/prognosen/999755/deutschland-beliebteste-genres-bei-videospielen>.