

Dokumentation – Entwicklungsprozess

Projekt: Videospiel-Zeitreise: Verkaufstrends 1980 - 2016

Kurs: Informationsvisualisierung SS 2025, FH Salzburg

Team: Sarah Haböck, Christina Kollnig, Dina Alhusaini

1 Zielsetzung & Zielgruppe

Ziel	Konkretisierung
Langfristige Genre-Trends sichtbar machen	Animiertes Bar-Chart-Race der Top-Genres von 1985 – 2016
Regionale Besonderheiten aufdecken	Stacked-Area-Chart vergleicht NA - EU - JP
Interaktive Exploration fördern	Jahr-Slider, Replay-Button, Checkbox-Filter, Fun-Fact-Cards

Primäre Zielgruppen

- **Gaming-Community** – erkennt, wann ihr Lieblingsgenre boomte.
- **Branchen-Analyst:innen** – erhalten eine schnelle, visuelle Übersicht zu Umsatzspitzen & Regionen.

2 Workflow

Phase	Schritte	Tooling
Datenbeschaffung	Kaggle-CSV <i>Video Game Sales</i> laden	Manuell → data/vgsales.csv im Repo
Cleaning & Feature Engineering	Zeilen ohne Sales entfernt, Genre-Synonyme zusammengeführt, Global_Sales vorab addiert	Excel / Google Sheets
Parsing im Browser	CSV einlesen & Typen konvertieren	d3.csv(..., d3.autoType) in main.js
Aggregation	Umsätze pro Jahr × Genre, Regionensummen (NA/EU/JP)	d3.rollup / d3.rollups in main.js
Frontend-Build	Bar-Chart-Race-Komponente, Stacked-Area-Chart, Section-Scroll	Vanilla JS + D3 v7; Modul-Bundling via Vite
Deployment	GitHub Pages	GitHub Actions

3 Komponenten & Design

	Komponente (Datei / Ordner)	Typ / Diagramm	Haupt-Interaktion	Visuelle Variablen
1	scripts/barChartRace.js	Bar-Chart-Race - Top-Genres (1985 - 2016)	<i>Replay</i> -Button (#replayGenres) startet Animation erneut	Länge der Balken = Global Sales Farbe = Genre-Color aus colors.js
2	scripts/barChartPlatform.js	Bar-Chart-Race - Top-Plattformen (2010 - 2016)	<i>Replay</i> -Button (#replayPlatforms)	Länge = Plattform-Sales Farbe = Plattform-Palette
3	scripts/stackedArea.js	Stacked-Area-Chart NA/EU/JP	input[type=range] (Jahr-Slider) + Checkboxes (#chkNA, #chkEU, #chkJP)	Flächenhöhe = Sum Sales Farbe = Region (Blau / Grün / Rot)
4	components/FunFactCard.html + cards.css	Fun-Facts-Cards	CSS-Flipseffekt, Wikipedialink hinterlegt	Cover-Bild als Icon Badges: Jahr (pink), Genre (violett), Sales (orange)
5	components/navbar.js	Top-Navbar (Kapitel-Anchor)	Smooth-Scroll + active-Highlight via intersectionObserver	Farbe = Menu-Item pink, aktiv → weiß
6	index.html (Section #hero)	Hero-Section (Titel + CTA)	Button „Los geht's ↓“ scrollt zur Section #1980	Headline-Pink #FF2E7E auf Night-Blue #0B0B1A (Kontrast 7.0 : 1)

Designsystem

- **Palette**
 - Primär-Akzent Pink #FF2E63 für Buttons & Hover
 - Balken/Flächen: d3.schemeTableau10 (10 farbenblinden-freundliche Töne)
- **Typografie**
 - Orbitron 700 für Überschriften & CTA-Buttons
 - Inter 400 / 500 für Fließtext (*per Google-Fonts-Import*)
- **Spacing & Formen**
 - Basis-8-pt-Grid; Element-Abstände meist 0.75 rem

- Cards: 12 px ($\approx 2 \times l$) Corner-Radius + Soft-Shadow 0 8 20 rgba(0,0,0,.35) bei Hover
- **Barrierefreiheit**
 - Text-zu-Hintergrund-Kontrast $\geq 4.5 : 1 \rightarrow$ WCAG AA erfüllt
 - Alt-Texte für alle Cover-Bilder
 - Keyboard-Focus klar sichtbar (outline: 2px solid var(--accent))

4 Probleme & Lösungen

Kategorie	Problem	Lösung
CORS	Browser blockierte das lokale CSV beim d3.csv-Laden	Projekt lokal über HTTP-Server starten (z. B. python -m http.server)
Inkonsistente Genres	Mehrfachschreibweisen (<i>FPS</i> , <i>Shooter</i> ...) führten zu Doppelzählungen	Einmalige Mapping-Tabelle erstellt & Genres zusammengeführt
Leere Jahre im Area-Chart	Fehlende Jahreswerte ließen die Fläche einbrechen	Vor dem Rendern Null-Zeilen für fehlende Jahre eingefügt
FPS-Jitter bei Race	Animation lief nicht stabil	Positions-Update über requestAnimationFrame und konstanter Throttle-Takt
Touch-Scroll	Kapitel-Links setzten sich unter die fixe Navbar	Für alle Sections scroll-margin-top gesetzt \rightarrow sauberes Ankern

5 Evaluation & Resultate

5.1 Methodik

- Think-Aloud-Mini-Test - 3 Personen \times 15 min
- Kurz-SUS (10 Items)
- Heuristische Checkliste (eigener 1-Seiter)

5.2 Kernerkenntnisse

Beobachtung (3-Personen-Mini-Test)	Aktion
2 / 3 störten sich an Lücken im Stacked-Area-Chart	Fehlende Jahre mit 0 gefüllt \rightarrow Grafik jetzt kontinuierlich
3 / 3 fanden das Bar-Chart-Race sofort verständlich	Flow unverändert beibehalten
1 / 3 wünschte Genre-Detailansicht	Als Ausblick notiert

5.3 Checkliste

Abschnitt	Kriterium	Erfüllt ✓/✗
Allgemeines Design	Titel & Kontext klar verständlich	
	Zielgruppe explizit adressiert	
	Datenquelle & Autor:innen genannt	
	Farb- & Schriftschema konsistent	
	Farbschema passend zum Datentyp (qualitativ / sequenziell / divergent)	
	Text-zu-Hintergrund-Kontrast $\geq 4.5 : 1$ (WCAG AA)	
Visualisierung & Technik	Achsentitel + Einheiten vorhanden	
	Skalen logisch & (falls sinnvoll) nullbasiert	
	Farben gut unterscheidbar & farbenblindfreundlich	
	Mind. 2 „Advanced“-Diagramme integriert	
Interaktion & Usability	Tooltip / Hover zeigt Detaildaten	
	Zeit-Slider oder Play/Pause vorhanden	
	Filter / Checkboxes für Nutzersteuerung	
	Tastatur-Navigation & Fokus-Outline sichtbar	
Barrierefreiheit	Kontrast AA erfüllt ($\geq 4.5 : 1$)	
	Alt-Texte für alle Bilder	
	WCAG-Prinzipien <i>wahrnehmbar • bedienbar • verständlich • robust</i> beachtet	
Storytelling & Narration	Einleitung → Verlauf → Fazit logisch	
	Perspektiven (z. B. Trends über Zeit) klar erkennbar	
	Begleittext erklärt jedes Visual	
	Klare Nutzerhinweise („Klicken ...“, „Bewegen ...“)	
Qualitätssicherung & Evaluation	Kurztest / Peer-Review durchgeführt	
	SUS-Fragebogen oder ähnliches ausgefüllt	
	Fehlerprüfung (Labels, Werte, Grammatik) erledigt	
	Diese Checkliste vollständig durchgegangen	