

Feature-Ideen für Best of Games

1. Kuratoren-Listen (Curator Lists)

Status: Geplant

Geschätzter Aufwand: ~4-5 Stunden

Kuratierte Sammlungen wie "Top 10 Cozy Games", "Beste Roguelikes unter 15€", "Hidden Gems 2024".

Warum dieses Feature?


Listen sind **Traffic-Magnete** weil sie Long-Tail Keywords abdecken, die einzelne Spieleseiten nicht erreichen:

- "beste roguelikes unter 15 euro"
 - "cozy games 2024"
 - "hidden gems indie spiele"
-

Architektur



Dynamisch vs. Statisch

Ansatz	Speichert	Aktualisierung	Empfohlen
Statisch	game_ids	Nur bei Cron-Lauf	Nein
Dynamisch	criteria (Regeln)	Bei jedem Request	Ja 

Dynamisch bedeutet:

- Neues Roguelike wird hinzugefügt → erscheint sofort in "Top Roguelikes"
- "Beste Spiele dieser Woche" zeigt immer die letzten 7 Tage
- Kein Cron-Job nötig für Listen-Updates

Datenbank-Schema

```
// src/lib/schema.ts

export const curatorLists = pgTable('curator_lists', {
  id: uuid('id').defaultRandom().primaryKey(),
  slug: text('slug').notNull().unique(),
  title: jsonb('title').$type<LocalizedField>().notNull(),
  description: jsonb('description').$type<LocalizedField>(),
  criteria: jsonb('criteria').$type<ListCriteria>().notNull(),
  coverImage: text('cover_image'), // Optional: Hero-Bild für Liste
  sortOrder: integer('sort_order').default(0),
  isActive: boolean('is_active').default(true),
  isFeatured: boolean('is_featured').default(false), // Für Homepage
  createdAt: timestamp('created_at', { withTimezone: true }).defaultNow(),
  updatedAt: timestamp('updated_at', { withTimezone: true }).defaultNow(),
});
```

Kriterien-Typ

```
// src/lib/list-criteria.ts

export type ListCriteria = {
  // Tag-Filter (OR-Verknüpfung)
  tags?: string[];

  // Score-Filter
  minScore?: number;
  maxScore?: number;

  // Preis-Filter (in Cents)
  maxPrice?: number;
  minDiscount?: number;
  onSale?: boolean;

  // Zeit-Filter (RELATIV – immer aktuell!)
```

```
publishedWithinDays?: number; // Review veröffentlicht innerhalb X
Tage
releasedWithinDays?: number; // Spiel released innerhalb X Tage

// Zeit-Filter (ABSOLUT – fixiert)
year?: number;

// Sortierung
orderBy?: 'score' | 'publishedAt' | 'releaseDate' | 'title';
orderDirection?: 'asc' | 'desc';

// Limit
limit?: number;
};
```

Beispiel-Listen

Slug	Titel (DE)	Kriterien
top-roguelike-games	Top 10 Roguelikes	{ tags: ["Roguelike"], orderBy: "score", limit: 10 }
best-this-week	Beste Spiele der Woche	{ publishedWithinDays: 7, orderBy: "score", limit: 10 }
new-this-month	Neu diesen Monat	{ publishedWithinDays: 30, orderBy: "publishedAt" }
best-cozy-games	Top 10 Cozy Games	{ tags: ["Cozy"], orderBy: "score", limit: 10 }
games-under-10-euros	Beste Spiele unter 10€	{ maxPrice: 1000, minScore: 7.0 }
games-under-15-euros	Beste Spiele unter 15€	{ maxPrice: 1500, minScore: 7.0 }
best-of-2024	Beste Indie-Spiele 2024	{ year: 2024, orderBy: "score", limit: 20 }
hidden-gems	Geheimtipps	{ minScore: 7.5, maxScore: 8.5, limit: 15 }
on-sale-now	Aktuell im Sale	{ onSale: true, minScore: 7.0 }

Zeitbasierte Listen - So funktioniert's

Montag, 9. Dezember – User besucht "Beste Spiele der Woche":

Beste Spiele der Woche
(Reviews vom 2.–9. Dezember)

- | |
|---------------------------------|
| 1. Game A (Score: 9.2) – 5. Dez |
| 2. Game B (Score: 8.8) – 7. Dez |
| 3. Game C (Score: 8.5) – 3. Dez |

Eine Woche später, Montag 16. Dezember:

Beste Spiele der Woche (Reviews vom 9.–16. Dezember)

← Automatisch verschoben!

- | |
|----------------------------------|
| 1. Game D (Score: 9.5) – 12. Dez |
| 2. Game E (Score: 9.0) – 10. Dez |
| 3. Game A (Score: 9.2) – 5. Dez |

← Neue Spiele!

← Fällt nächste Woche raus

Query-Funktionen

```
// src/lib/queries.ts

// Alle aktiven Listen (für /lists Übersicht)
export async function getCuratorLists(): Promise<CuratorList[]>

// Featured Listen (für Homepage)
export async function getFeaturedLists(): Promise<CuratorList[]>

// Einzelne Liste laden
export async function getCuratorListBySlug(slug: string):
Promise<CuratorList | null>

// Spiele basierend auf Kriterien – DYNAMISCH!
export async function getGamesForList(criteria: ListCriteria):
Promise<ReviewListItem[]>
```

getGamesForList baut dynamisch eine SQL-Query:

```
export async function getGamesForList(criteria: ListCriteria):
Promise<ReviewListItem[]> {
  const conditions = [eq(reviews.isPublished, true)];

  // Relativer Zeitfilter – wird bei JEDEM Request neu berechnet!
  if (criteria.publishedWithinDays) {
    conditions.push(
      sql`${reviews.publishedAt} >= NOW() - INTERVAL
'${criteria.publishedWithinDays} days'`
    );
  }
}
```

```
if (criteria.minScore) {
  conditions.push(sql`CAST(${reviews.score} AS DECIMAL) >=
${criteria.minScore}`);
}

if (criteria.tags?.length) {
  // Join mit tags Tabelle
}

if (criteria.maxPrice) {
  // Join mit price_snapshots (latest)
}

// ... weitere Bedingungen

return db
  .select({ /* ... */ })
  .from(reviews)
  .innerJoin(games, eq(reviews.gameId, games.id))
  .where(and(...conditions))
  .orderBy(/* basierend auf criteria.orderBy */)
  .limit(criteria.limit || 20);
}
```

Frontend-Seiten

1. Listen-Übersicht: `/[lang]/lists/page.tsx`

- Zeigt alle aktiven Listen als Karten
- Jede Karte: Titel, Beschreibung, Anzahl Spiele, Cover-Bild
- Link zu `/lists/[slug]`

2. Listen-Detail: `/[lang]/lists/[slug]/page.tsx`

- Lädt Liste via `getCuratorListBySlug`
- Lädt Spiele via `getGamesForList(list.criteria)`
- Rendert Spiele mit `ReviewCard` Komponente
- SEO: Meta-Tags, Structured Data (ItemList Schema)

3. Homepage-Integration: `/[lang]/page.tsx`

- Neue Sektion "Featured Lists" nach Recent Reviews
- Zeigt 3-4 Listen mit `isFeatured=true`
- Link zu `/lists` für alle Listen

Neue Komponente: `ListCard.tsx`

```
// src/components/ListCard.tsx

type ListCardProps = {
  slug: string;
  title: string;
  description?: string;
  coverImage?: string;
  gameCount: number;
  locale: Locale;
};

// Zeigt:
// - Cover-Bild oder generiertes Grid aus Spiele-Bildern
// - Titel
// - Beschreibung (gekürzt)
// - Badge mit Anzahl Spiele
// - Link zu /lists/[slug]
```

SEO

Structured Data (ItemList Schema)

```
// src/lib/structured-data.ts

export function generateCuratorListStructuredData(
  list: CuratorList,
  games: ReviewListItem[]
) {
  return {
    '@context': 'https://schema.org',
    '@type': 'ItemList',
    name: list.title,
    description: list.description,
    url: `${SITE_URL}/lists/${list.slug}`,
    numberOfItems: games.length,
    itemListElement: games.map((game, index) => ({
      '@type': 'ListItem',
      position: index + 1,
      url: `${SITE_URL}/games/${game.slug}`,
      name: game.title,
    })),
  };
}
```

Sitemap erweitern

```
// src/app/sitemap.ts

// Listen hinzufügen
const lists = await db
  .select({ slug: curatorLists.slug, updatedAt: curatorLists.updatedAt })
  .from(curatorLists)
  .where(eq(curatorLists.isActive, true));

for (const list of lists) {
  sitemapEntries.push({
    url: `${base}/${locale}/lists/${list.slug}`,
    lastModified: list.updatedAt || new Date(),
    changeFrequency: 'weekly' as const,
    priority: 0.8, // Hoch! Listen sind wertvoller Content
  });
}
```

Interne Verlinkung

Homepage

- "Entdecke unsere Listen" Sektion
 - Links zu Featured Lists
- Footer: Beliebte Listen

Game Detail Page

- "Diese Listen enthalten {Game}" (optional, später)

/lists Übersichtsseite

- Alle Listen mit Vorschau

Benötigte Dateien

Aktion	Datei	Beschreibung
Ändern	src/lib/schema.ts	curatorLists Tabelle hinzufügen
Neu	src/drizzle/0005_*.sql	Migration
Neu	src/lib/list-criteria.ts	ListCriteria Type
Ändern	src/lib/queries.ts	4 neue Query-Funktionen
Ändern	src/lib/structured-data.ts	ItemList für Listen
Neu	src/app/[lang]/lists/page.tsx	Listen-Übersicht
Neu	src/app/[lang]/lists/[slug]/page.tsx	Listen-Detail
Ändern	src/app/[lang]/page.tsx	Featured Lists Sektion

Aktion	Datei	Beschreibung
Neu	<code>src/components/ListCard.tsx</code>	Vorschau-Karte
Ändern	<code>src/app/sitemap.ts</code>	Listen-URLs hinzufügen
Ändern	<code>src/dictionaries/en.json</code>	Übersetzungen
Ändern	<code>src/dictionaries/de.json</code>	Übersetzungen

Reihenfolge der Implementation

1. Schema + Migration erstellen
2. ListCriteria Type definieren
3. Query-Funktionen implementieren
4. Listen-Detail-Seite (`/lists/[slug]`)
5. Listen-Übersicht (`/lists`)
6. ListCard Komponente
7. Homepage-Integration (Featured Lists)
8. SEO (Structured Data, Sitemap)
9. Seed-Daten: Erste Listen via SQL einfügen

Listen via SQL erstellen (Beispiele)

```
-- Top Roguelikes
INSERT INTO curator_lists (slug, title, description, criteria,
is_featured, sort_order)
VALUES (
  'top-roguelike-games',
  '{"en": "Top 10 Roguelike Games", "de": "Top 10 Roguelike-Spiele"}',
  '{"en": "The best roguelike games, ranked by our review scores.", "de":
"Die besten Roguelike-Spiele, sortiert nach unserer Bewertung."}',
  '{"tags": ["Roguelike"], "orderBy": "score", "limit": 10}',
  true,
  1
);

-- Beste Spiele der Woche
INSERT INTO curator_lists (slug, title, description, criteria,
is_featured, sort_order)
VALUES (
  'best-this-week',
  '{"en": "Best Games This Week", "de": "Beste Spiele der Woche"}',
  '{"en": "Top-rated games from the past 7 days.", "de": "Top-bewertete
Spiele der letzten 7 Tage."}',
  '{"publishedWithinDays": 7, "orderBy": "score", "limit": 10}',
  true,
  0
);
```



```
-- Beste Spiele unter 10€
INSERT INTO curator_lists (slug, title, description, criteria,
is_featured, sort_order)
VALUES (
  'games-under-10-euros',
  '{"en": "Best Games Under €10", "de": "Beste Spiele unter 10€"}',
  '{"en": "Great games that won\'t break the bank.", "de": "Tolle
Spiele, die das Budget schonen."}',
  '{"maxPrice": 1000, "minScore": 7.0, "orderBy": "score"}',
  false,
  10
);
```

Optionale Erweiterungen (später)

Idee	Aufwand	Nutzen
Manuelles Override - einzelne Spiele hinzufügen/entfernen	Mittel	Für "Editor's Picks"
List Cover Auto-Generate - Grid aus ersten 4 Spiele-Bildern	Gering	Schönere Vorschau
"Diese Listen enthalten {Game}" auf Game-Detail-Seite	Gering	Bessere Verlinkung
Admin-UI zum Erstellen von Listen	Hoch	Benutzerfreundlicher

Weitere Ideen

(Platz für zukünftige Feature-Ideen)