

# Ball Burst Saga

---

*Ein Match-3-Spiel mit Adaptiver KI*

**Kurs:** Game Programming & AI

**Engine:** Godot 4.5 • GDScript



# Inhaltsverzeichnis

01

Projektübersicht & Spielidee

02

Spielmechanik & Architektur

03

Künstliche Intelligenz – Übersicht

04

Adaptive Schwierigkeit (DDA)

05

KI-Hinweissystem (AIAdvisor)

06

Animationen & Visuelles Feedback

07

Spielerprofil & Personalisierung

08

Spielstand & Datenpersistenz

09

Fazit & Ausblick

# 01 Projektübersicht & Spielidee

## Was ist Ball Burst Saga?

**Ball Burst Saga** ist ein buntes Match-3-Puzzlespiel, entwickelt in [Godot 4.5](#) mit GDScript als Skriptsprache.

### Projektziel:

Ein vollständiges Spiel mit moderner KI-Integration entwickeln – inklusive adaptiver Schwierigkeit, automatischem Hinweissystem und personalisiertem Spielerprofil.

	<a href="#">Godot 4.5</a>	Open-Source Game-Engine
	<a href="#">GDScript</a>	Python-ähnliche Skriptsprache
	<a href="#">KI-Module</a>	DDA, AIAdvisor, EMA-Profil
	<a href="#">SaveSystem</a>	JSON + ConfigFile Persistenz

## Spielfeatures

-  10 progressive Level
-  8x8 Spielfeld
-  6 Kugelfarben
-  Combo-Kaskaden
-  Candy-Crush-Animationen
-  Adaptive KI-Schwierigkeit
-  Automatische Hinweise
-  Personalisiertes Profil
-  Spielstand-Speicherung

# 02 Spielmechanik & Architektur

## Spielablauf (Zustandsmaschine / FSM)



### Punkte-Formel

Punkte = Anzahl × 100 × Combo-Multiplikator

Combo-Bonus: +50% pro Kaskade

### Level-Übersicht (Basis-Werte, vor KI-Anpassung)

Level	Züge	Ziel
1–2	20–22	4.000–6.000
3–5	18–20	8.000–12.000
6–8	15–16	14.000–18.000
9–10	13–14	20.000–24.000

### Skript-Architektur

- Main.gd Szenensteuerung, UI, Level-Start
- Board.gd Spielfeld, FSM, Animationen
- GameState.gd Singleton: Level, Profil, Speichern
- AdaptiveDifficulty.gd DDA-KI: EMA-basierte Anpassung
- AIAdvisor.gd Heuristisches Hinweissystem
- SaveSystem.gd JSON-Datenpersistenz

# 03 Künstliche Intelligenz - Übersicht

*Das Spiel integriert zwei KI-Systeme, die unterschiedliche Aspekte der Spielerfahrung optimieren.*



## Adaptive Schwierigkeit

*Dynamic Difficulty Adjustment (DDA)*

- ▶ Analysiert Win-Rate, Zug-Effizienz, Hint-Nutzung
- ▶ Berechnet Schwierigkeits-Score  $D \in [0.0, 1.0]$
- ▶ Passt Züge, Ziel-Score & Hint-Delay an
- ▶ Algorithmus: Exponential Moving Average (EMA)
- ▶ Akademische Referenz: Hunicke (2005)



## AIAdvisor Hinweissystem

*Heuristik-basierter Game-Solver*

- ▶ Analysiert alle möglichen Züge auf dem Spielfeld
- ▶ Bewertet Züge nach Match-Größe und Position
- ▶ Bonus für zentrale Züge (bessere Kaskadenposition)
- ▶ Automatischer Hint nach X Sekunden Inaktivität
- ▶ Verwendet `_virtual_swap()` für sichere Simulation

# 04 Adaptive Schwierigkeit (DDA)

EMA-Formel:

$$\text{neuer\_Wert} = \alpha \times \text{Letztes\_Ergebnis} + (1 - \alpha) \times \text{Bisheriger\_Wert} \quad \text{mit } \alpha = 0.3$$

Eingabe-Faktoren für Schwierigkeits-Score  $D \in [0, 1]$ :

40%



Win-Rate

Gewinnt der Spieler oft?

25%



Zug-Effizienz

Wenig Züge gebraucht = stark

20%



Hint-Nutzung

Selten Hints = unabhängig

15%



Sieges-Serie

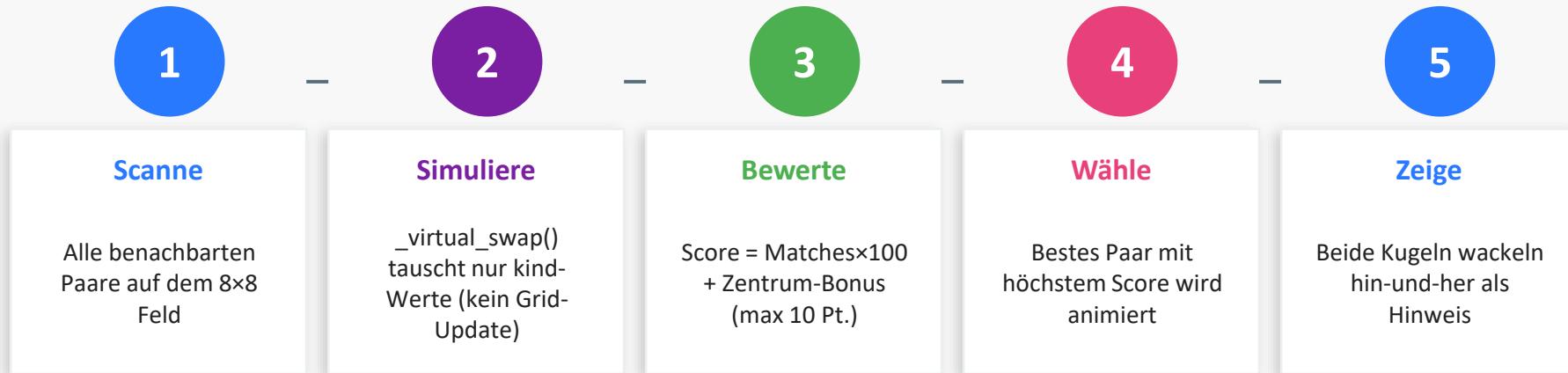
5 Siege = Maximum

Ausgabe-Parameter (je nach Schwierigkeits-Score D):

Parameter	Schwach ( $D \approx 0$ )	Stark ( $D \approx 1$ )
Züge	$D=0 \rightarrow +4$ Züge	$D=1 \rightarrow -3$ Züge
Ziel-Score	$D=0 \rightarrow \times 0.82$	$D=1 \rightarrow \times 1.18$

# 05 KI-Hinweissystem (AIAdvisor)

## Algorithmus - Greedy Heuristic Search:



### func Scoring\_Funktion (AIAdvisor.gd):

```
func _score_swap(board, a, b):
    board._virtual_swap(a, b)
    var matches = board._find_all_matches()    # ← Simulation
    var base    = matches.size()
    board._virtual_swap(a, b)                  # ← Zurücksetzen
    var center_bonus = 10 - (abs(a.grid_x - cx) + abs(a.grid_y - cy))
    return base * 100 + center bonus
```

# 06 Animationen & Visuelles Feedback

## 3er-Match



- ▶ Stern schwebt nach oben
- ▶ Fade-Out-Effekt
- ▶ Zufälliger X-Versatz

## 4+ Match Burst



- ▶ 8 Sterne im Kreisring
- ▶ Jede Kugel schießt Icon
- ▶ Kombo-Label erscheint

## Screen Flash



- ▶ 4-Match → Gelb
- ▶ 5-Match → Cyan
- ▶ 6 → Orange, 7+ → Pink

## Combo-Kaskade



- ▶ +50% Bonus pro Kaskade
- ▶ 'Combo x2!' Label
- ▶ Punkte multipliziert

# 07 Spielerprofil & Personalisierung

## Willkommens-Bildschirm

- Zeigt Spieltitle & Untertitel
- Spitzname-Eingabe (LineEdit)
- Vollständige Spielanleitung
- 'Los geht's!' Button
- Name wird dauerhaft gespeichert
- Beim Neustart: Name wird gelöscht, Welcome-Screen erscheint neu

## Personalisierung im Spiel

- HUD:** Name unten-links angezeigt
- Level-Start:** 'Viel Glück, [Name]! 🎉'
- Sieg:** '🎉 Super, [Name]!'
- Niederlage:** '😢 Nicht aufgeben, [Name]!'
- Score:** '[Name]'s Punkte: XXXXX'
- Restart:** Bestätigung → alles wird komplett gelöscht

# 08 Spielstand & Datenpersistenz

## GameState → save.cfg

*Godot ConfigFile (Key-Value)*

- current\_level\_index
- player\_name
- win\_rate (EMA)
- avg\_moves\_ratio (EMA)
- consecutive\_wins / losses
- hint\_usage\_rate (EMA)
- total\_sessions

## SaveSystem → save\_game.json

*JSON via FileAccess*

- current\_level (Index)
- total\_score (Sitzungs-Score)
- Einfach erweiterbar
- Menschenlesbar / debugbar
- Plattformübergreifend

⟳ Neustart löscht BEIDE Dateien und setzt den kompletten GameState auf Standardwerte zurück.

# 09 Fazit & Ausblick

## Erreichte Ziele

- ✓ Vollständiges Match-3-Spiel (10 Level)
- ✓ Adaptive Schwierigkeit via EMA-Algorithmus
- ✓ Heuristisches KI-Hinweissystem
- ✓ Candy-Crush-Animationen (3er & 4+ Matches)
- ✓ Personalisiertes Spielerprofil mit Name
- ✓ Vollständiger Neustart mit Datenlöschung
- ✓ Duale Datenpersistenz (cfg + JSON)

## Mögliche Erweiterungen

- Q-Learning Reinforcement Agent
- Online Highscore-System
- Spezial-Kugeln (Bombe, Regenbogen)
- Zeitbasierter Spielmodus
- Sound-reaktive Animationen
- Mehr Level mit Story-Elementen
- Mobile-Portierung (Android/iOS)

*Ball Burst Saga zeigt, wie KI-Techniken spielerisch eingesetzt werden können, um eine personalisierte und dynamische Spielerfahrung zu schaffen – ganz ohne maschinelles Lernen im klassischen Sinne.*



# Vielen Dank!

---

*Ball Burst Saga • Game Programming & AI • 2026*



Live-Demo  
verfügbar



Fragen &  
Diskussion



KI-Parameter live  
anpassen