*Dieses Dokument enthält die detaillierten Informationen zu Charakteren, Skills, Gegnern und den grundlegenden Spielmechaniken, basierend auf den* ***JSON5****-Dateien, Konfigurationsdateien (src/config/settings.json5) und dem aktuellen Stand des Codes im Projekt. Es dient als primäre Referenz für die Spielinhalte und deren Funktionsweise.*

***Wichtig: JSON5 Format und Kommentare****Wir verwenden das* ***JSON5****-Format für unsere Datendateien (typischerweise mit der Dateiendung .json5). JSON5 ist eine Erweiterung von JSON, die zusätzliche Features für menschliche Lesbarkeit erlaubt. Das wichtigste für uns ist die Unterstützung für* ***Kommentare****.*

* ***Single-line Kommentare:*** *Beginnen mit //*
* ***Multi-line Kommentare:*** *Eingeschlossen in /\* ... \*/*

{

// Dies ist ein einzelner Kommentar

"name": "Beispiel-Charakter",

"level": 1, // Level des Charakters

/\*

Dies ist ein mehrzeiliger Kommentar

und beschreibt die Attribute.

\*/

"attributes": {

"STR": 10,

"DEX": 12,

"INT": 5,

"CON": 8,

"WIS": 7, // Weisheit

}, // Nach dem letzten Element ist ein Komma erlaubt (JSON5 Feature)

"skills": [

"basic\_attack", // Standardangriff

],

} // Ein Komma am Ende des Top-Level-Objekts ist erlaubt (JSON5 Feature)

Du musst sicherstellen, dass du dieses **JSON5-Format** für Datendateien verwendest und verstehst, dass im Python-Code die **-Bibliothek** (anstelle der Standard json-Bibliothek) zum Laden dieser Dateien benötigt wird.

***Umfassende Zusammenfassung der Spiel-Definitionen und Mechaniken (Stand: Basierend auf bereitgestellten Dokumenten und Q&A)***

***1. Datenformat und Primäre Datenquellen:***

* ***Format:*** *JSON5 (erlaubt Kommentare, flexiblere Syntax). Benötigt json5-Bibliothek in Python.*
* ***Konfigurationsdatei:*** *src/config/settings.json5 (globale Spielparameter).*
* ***Definitionsdateien:***
  + *src/definitions/json\_data/characters.json5 (Charakterklassen-Startwerte)*
  + *src/definitions/json\_data/skills.json5 (Skill-Definitionen)*
  + *src/definitions/json\_data/opponents.json5 (Gegner-Definitionen)*
  + *(Implizit: primary\_attributes.json, combat\_stats.json aus Teil 1 sind nun in diesen detaillierteren Dateien aufgegangen oder konzeptionell vorhanden).*

***2. Globale Konfigurationen (aus***

* *game\_settings.min\_damage:* ***1***
* *game\_settings.base\_weapon\_damage:* ***5*** *(genutzt für Skills mit base\_damage: null)*
* *game\_settings.hit\_chance\_base:* ***90***
* *game\_settings.hit\_chance\_accuracy\_factor:* ***3***
* *game\_settings.hit\_chance\_evasion\_factor:* ***2***
* *game\_settings.hit\_chance\_min:* ***5***
* *game\_settings.hit\_chance\_max:* ***95***
* *game\_settings.xp\_level\_base:* ***100***
* *game\_settings.xp\_level\_factor:* ***1.5***
* *(Logging- und RL-spezifische Settings sind ebenfalls vorhanden).*

***3. Primärattribute:***

* ***STR (Stärke):*** *Physischer Schaden, Tragkraft (Tragkraft-Mechanik TBD).*
* ***DEX (Geschicklichkeit):*** *Angriffsgeschwindigkeit (nicht explizit implementiert), Ausweichen, Fernkampfschaden, Genauigkeit.*
* ***INT (Intelligenz):*** *Magischer Schaden, Beeinflusst Mana (aktuell nur base\_mana als Max-Wert, keine weitere Skalierung der Max-Menge durch INT).*
* ***CON (Konstitution):*** *Maximale Lebenspunkte (HP).*
* ***WIS (Weisheit):*** *Willenskraft, Magieresistenz, Effektivität von Heilzaubern.*

***4. Kampfwerte:***

* ***HP (Lebenspunkte):*** *Gesundheit.*
* ***MANA:*** *Ressource für Magier, Kleriker.*
* ***STAMINA:*** *Ressource für Krieger.*
* ***ENERGY:*** *Ressource für Schurken.*
* ***ARMOR (Rüstung):*** *Reduziert physischen Schaden.*
* ***MAGIC\_RESIST (Magieresistenz):*** *Reduziert magischen Schaden.*
* ***HIT\_CHANCE (Trefferchance):*** *Basis-Wahrscheinlichkeit zu treffen.*
* ***EVASION (Ausweichen):*** *Reduziert Trefferchance des Angreifers.*
* ***ACCURACY (Genauigkeit):*** *Erhöht eigene Trefferchance.*
* ***INITIATIVE:*** *Bestimmt Zugreihenfolge (Kernmechanik TBD).*

***5. Charakterklassen (aus****Alle Klassen haben base\_hp: 50 (vor CON-Bonus).  
\** ***Krieger ("krieger"):****\* Primärattribute: STR:14, DEX:10, INT:8, CON:12, WIS:8  
\* Kampfwerte: base\_stamina:100, armor:5, magic\_resist:1. (Mana/Energy: 0)  
\* Primärressource: STAMINA  
\* Skills: basic\_strike\_phys, power\_strike, shield\_bash, cleave.  
\** ***Magier ("magier"):****\* Primärattribute: STR:8, DEX:10, INT:14, CON:9, WIS:11  
\* Kampfwerte: base\_mana:120, armor:1, magic\_resist:4. (Stamina/Energy: 0)  
\* Primärressource: MANA  
\* Skills: basic\_magic\_bolt, fireball, frostbolt, arcane\_barrier.  
\** ***Schurke ("schurke"):****\* Primärattribute: STR:10, DEX:14, INT:8, CON:10, WIS:8  
\* Kampfwerte: base\_energy:100, armor:3, magic\_resist:2. (Mana/Stamina: 0)  
\* Primärressource: ENERGY  
\* Skills: basic\_strike\_finesse, precise\_stab, distraction, sprint.  
\** ***Kleriker ("kleriker"):****\* Primärattribute: STR:10, DEX:8, INT:10, CON:11, WIS:13  
\* Kampfwerte: base\_mana:110, armor:4, magic\_resist:3. (Stamina/Energy: 0)  
\* Primärressource: MANA  
\* Skills: basic\_holy\_spark, heal, holy\_light, protective\_ward.*

***6. Skills (aus****Jeder Skill hat ID, Kosten (Wert & Typ), Effekte (Basis-Schaden, Attributskalierung, Multiplikator) und ggf. applies\_effects (Status-Effekte).  
\** ***Schadens-Skills:****\* base\_damage: null nutzt game\_settings.base\_weapon\_damage (5).  
\* Beispiele: power\_strike (STR, x1.5), shield\_bash (STR, Basis 2 + STUNNED 1 Rd.), fireball (INT, Basis 10 x2 + BURNING 3DMG/2Rd.), frostbolt (INT, Basis 8 x2 + SLOWED 1Rd.), precise\_stab (DEX, 25% Rüstungs-Ignoranz).  
\** ***Heil-Skills:*** *heal (Heilt 15 + WIS5).5 für 3Rd.), distraction (ACCURACY\_DOWN -3 für 2Rd.), sprint (INITIATIVE\_UP +20 für 1Rd.), protective\_ward (DEFENSE\_UP +3 für 3Rd.).  
\** ***Gegner-Skills:*** *basic\_shot\_phys (DEX), weakening\_curse (WEAKENED STR -3 für 3Rd.), heal\_lesser (Heilt 10 + WIS\*3).  
\** ***Status-Effekt-Anwendung:*** *Aktuell 100% Chance bei Treffer. Geplant: variables Feld application\_chance.*

***7. Gegner (aus src/definitions/json\_data/opponents.json5):****Jeder Gegner hat Primärattribute, combat\_values (base\_hp, base\_mana etc., armor, magic\_resist), eine Liste von Skill-IDs und einen xp\_reward.  
\* Beispiele:  
\* goblin\_lv1: STR:8, DEX:12, CON:9, base\_hp:50, armor:2, xp\_reward:50. Skills: basic\_strike\_phys. Tags: ["GOBLINOID", "HUMANOID"].  
\* goblin\_schamane\_lv3: INT:14, WIS:12, base\_hp:50, base\_mana:100, magic\_resist:3, xp\_reward:100. Skills: basic\_magic\_bolt, weakening\_curse, heal\_lesser.  
\** ***Gegner-Tags:*** *Definiert im tags-Feld (z.B. ["UNDEAD"]). Logik für Boni wie holy\_lights bonus\_vs\_type: "UNDEAD" muss in Schadensberechnung implementiert werden.*

***8. Kern-Spielmechaniken & Berechnungen:***

* ***HP-Berechnung (Start):*** *MaxHP = base\_hp (aus JSON) + (CON \* 5).*
* ***Maximal-Ressourcen (Mana, Stamina, Energy):*** *Aktuell die base\_... Werte aus den JSON-Dateien. Keine weitere Skalierung durch Attribute für die Max-Werte.*
* ***Attribut-Bonus:*** *(Attributwert - 10) // 2 (Ganzzahlige Division).*
* ***Schadensberechnung (Basis-Skill):*** *floor((Basis-Schaden + Attribut-Bonus) \* Multiplikator).*
  + *Basis-Schaden: Aus Skill-Definition oder game\_settings.base\_weapon\_damage (5).*
* ***Schadensreduktion:*** *max(game\_settings.min\_damage (1), Eingehender\_Schaden - (Rüstung oder Magieresistenz)).*
* ***Trefferchance:*** *max(hit\_chance\_min (5), min(hit\_chance\_max (95), hit\_chance\_base (90) + (Angreifer\_GenauigkeitsMod \* hit\_chance\_accuracy\_factor (3)) - (Ziel\_AusweichMod \* hit\_chance\_evasion\_factor (2)))).*
  + *Genauigkeits-/Ausweich-Modifikator: Basis (DEX - 10) // 2 + Effekte.*
* ***Status-Effekte (Interaktion):***
  + *Bei Neuanwendung desselben Effekts: Dauer wird auf max(alte\_Dauer, neue\_Dauer) gesetzt (Refresh/Verlängerung).*
  + *Potency wird von der neuen Anwendung überschrieben (kein Stacken der Stärke).*
* ***XP-Vergabe:*** *Fester Wert pro Gegner (xp\_reward).*
* ***XP für Levelaufstieg:*** *ceil(xp\_level\_base (100) \* (xp\_level\_factor (1.5) ^ (Aktuelles\_Level - 1))).*
* ***Level-Up-Boni:*** *Aktuell: Volle Heilung/Ressourcenwiederherstellung. Weitere Boni (Attributpunkte etc.) sind TODO. Es gibt kein festes Maximallevel im Spieldesign.*

***9. Wichtige Offene Punkte / Zukünftige Features (Design & Implementierung):***

* ***Initiative & Zugreihenfolge:*** *Die Kernmechanik zur Bestimmung der Basis-Initiative und der Kampfreihenfolge fehlt. Der INITIATIVE\_UP-Effekt existiert, aber seine Einbindung ist offen. (Aktuell feste Reihenfolge in cli\_main\_loop.py).*
* ***Items & Waffen:*** *Geplant, insbesondere Waffen, die den base\_weapon\_damage ersetzen/modifizieren. Struktur für Item-Definitionen (z.B. items.json5) fehlt.*
* ***Tragkraft:*** *Konzeptionell durch STR beeinflusst, aber Mechanik und Formel (z.B. Basis + STR \* Faktor) sowie Konsequenzen bei Überschreitung sind TBD.*
* ***Multi-Target Skill Logik:*** *Auswahl von Sekundärzielen für AREA-Skills (z.B. cleaves zweites Ziel) muss implementiert werden.*
* ***Implementierung von bonus\_vs\_type:*** *Die Logik, um den Schadensbonus gegen bestimmte Gegner-Tags (z.B. UNDEAD) anzuwenden, muss in die Schadensberechnung integriert werden.*
* ***Detaillierte Level-Up-Boni:*** *Konkrete numerische Boni (Attributpunkte, Skillpunkte, etc.) sind noch zu definieren.*
* ***Chance für Status-Effekte:*** *Implementierung eines application\_chance-Feldes für Skills, um Effekte mit <100% Wahrscheinlichkeit zu ermöglichen.*