

Text Adventure:

- Avatar zw. Räumen und Plätzen navigieren
 - ↳ `north(n)`, `east(e)`, `west(w)`, `south(s)`
 - über Karte eingeben
- Nach jeder Navigation: kurze Beschreibung des aktuell umgebenden Raumes + Liste der darin befindlichen aufnehmbaren Gegenstände
 - ↳ `look(l)` → Ausgabe wird wiederholt
 - `take(t) <items>` → fügt den als angegebenen Gegenstand dem Inventar der Spielern an + entfernt ihn aus dem Raum
 - `inventory(i)` → Liste der aufgenommenen Gegenstände wird angezeigt
 - `drop(d) <items>` → wird ein mit angegebener Gegenstand umgekehrt aus dem Inventar entfernt + in den Raum gelegt
 - ↳ An einem Ort abgelegte Gegenstände verbleiben dort und können auch nur dort wieder aufgenommen werden.
- `command(c)` → eine Liste der Interaktionsmöglichkeiten wird dargestellt
 - `quit(q)` → beendet das Programm
 - `attack(a) <character>` → Angriffe durch die Spielern
- Gesundheitszustand der Charaktere
- Nicht-Spieler-Charaktere wie Monster
- Intelligenter Charaktere → eigenes Inventory, verfügen über Gegenstände + Infos + können eigenständig zw. Räumen & Plätzen navigieren

Konzept:

1. Karte mit einem Adventure mit wenigen Räumen (5), Plätzen, Gegenständen und Charakteren
2. Use-Case-Diagramme entwerfen
3. Klassendiagramme für die unterschiedlichen Entitäten im Spiel
4. Aktivitätsdiagramme
5. Strukturen für die Pakete entwerfen, welche den Aufbau eines Adventures und dessen vollständigen Startzustand beschreiben.

Charaktere → Character.cs

Räume → Location.cs

Gegenstände → Items.cs

Befehle → Controls.cs

Programm
(Main-Methode) → Program.cs

Konsolenausgabe → ConsoleOutput.cs