



## Learning by Questing

André Matutat, Birgit Christina George, Carsten Gips



- Ausgangssituation

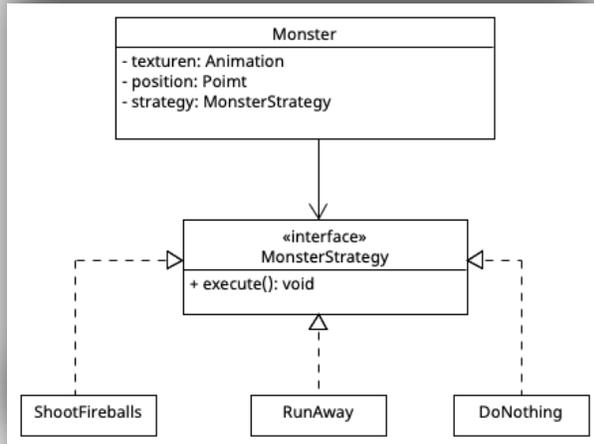
- Klassische Vorlesung
- Wöchentliche Praktikumsaufgaben mit Vorstellung und Bewertung
- Klausur am Ende der Vorlesungszeit

- Probleme

- Große Heterogenität der Studierenden
- Motivation bei wöchentlichen kleinen Aufgaben niedrig
- Kontaktzeit vor allem für Bewertung von Aufgaben

KW	Vorlesung (Sprechstunde)	Praktikum (Rücksprache)
15	<a href="#">Orga (Zoom)</a>    <a href="#">Git 1: Intro</a>   <a href="#">Git 2: Basics</a>	- (extra Zoom für alle 10-11 Uhr)
16	<a href="#">Git 3: Branches</a>   <a href="#">Git 4: Branching-Strategien</a>    <a href="#">Logging</a>    <a href="#">Javadoc</a>	B1: PM-Dungeon und Held
17	<a href="#">Git 5: Remote</a>    <a href="#">Visitor-Pattern</a>    <a href="#">Generics 1: Klassen &amp; Methoden</a>   <a href="#">Generics 2: Bounds &amp; Wildcards</a>	B2: Logging, Monster und Kampf
18	<a href="#">Generics 3: Type Erasure</a>   <a href="#">Generics 4: Polymorphie</a>    <a href="#">Observer-Pattern</a>    <a href="#">Defaultmethoden</a>	B3: Loot, Taschen und Kisten
19	<b>E1</b>    <a href="#">JUnit 1: Testen mit JUnit</a>   <a href="#">JUnit 2: Testfallerstellung</a>	B4: HUD, Fallen, Erfahrung und Skills
20	<a href="#">Bad Smells</a>   <a href="#">Refactoring</a>    <a href="#">Git 7: Bisect</a>    <a href="#">Strategy-Pattern</a>	B5: Quests, JUnit
21	-	-
22	<a href="#">Funktions-Interfaces &amp; Lambdas</a>   <a href="#">Methodenreferenzen</a>    <a href="#">RegExp</a>    <a href="#">Git 6: Workflows</a>	B6: Fernkampf, schlaue Monster und Refactoring
23	<a href="#">Annotationen</a>    <a href="#">Reflection</a>	Pitch Projekt
24	<b>E2</b>    <a href="#">CLI/Konfiguration</a>    <a href="#">Swing1</a>   <a href="#">Swing2</a>   <a href="#">Swing3</a>   <a href="#">Swing4</a>	Meilenstein 1
25	<a href="#">Build 1: Ant</a>   <a href="#">Build 2: Gradle</a>   <a href="#">Build 3: CI</a>   <a href="#">Docker</a>	Meilenstein 2
26	<a href="#">Threads 1: Intro</a>   <a href="#">Threads 2: Synchronisation</a>   <a href="#">Threads 3: High Level Concurrency</a>    <a href="#">Java2D</a>	Präsentation Projekt
27	<a href="#">Build 4: Maven</a>    <a href="#">Enumerationen</a>    <a href="#">Rückblick</a>    <a href="#">Prüfungsvorbereitung</a>	-
28	-	-
29	<b>E3, E-Klausur Performanz</b>	-

**Aufgabe**  
 Implementiere verschiedene Monster, mit verschiedenen Verhaltensweisen.



```

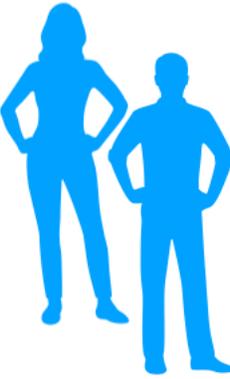
1 usage
private Animation texture;
1 usage
private Point position;
1 usage
private MonsterStrategy strategy;

no usages new *
public Monster(Animation texture, Point position, MonsterStrategy strategy) {
    this.texture = texture;
    this.position = position;
    this.strategy = strategy;
}
    
```



# DUNGEON: LEARNING BY QUESTING

Aufgabe

Lehrende

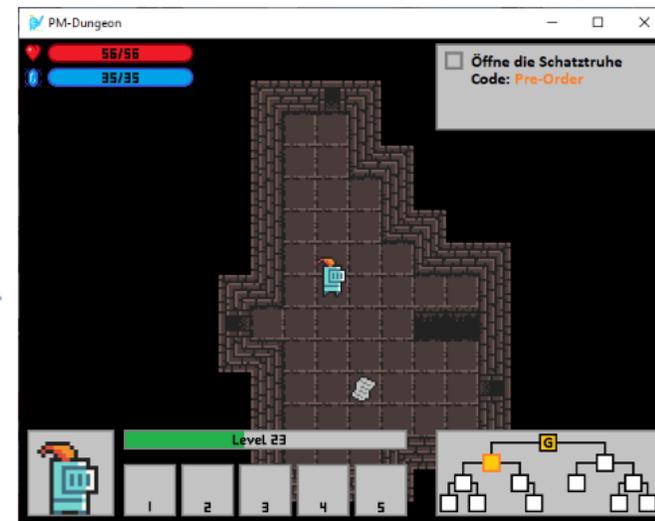
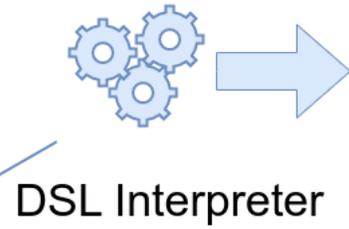
```
graph g {
  G--T--Q--D
  Q--F
  T--X--S
  G--W--E
  W--C--U
  C--N
}

task my_task {
  text: "Öffne die Schatztruhe, Code: Post-Order",
  task_type: TEXT,
  correct_answer: postorder(g)
}

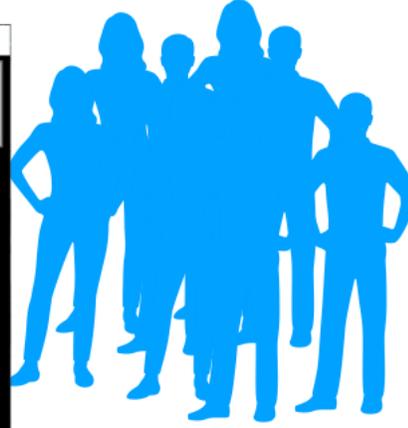
fn postorder(graph g) -> string[] {
  retval: string;
  if g.left
    retval += postorder(g.left)
  if g.right
    retval += postorder(g.right)
  retval += g.node_name
  return retval;
}

level_config config {
  task: my_task,
  // task-builder Funktion registrieren, sodass sie
  task_builder_functions: [build_chest_quiz]
}
```

DSL Eingabe



Generiertes Dungeon-Level und Quest



Studierende

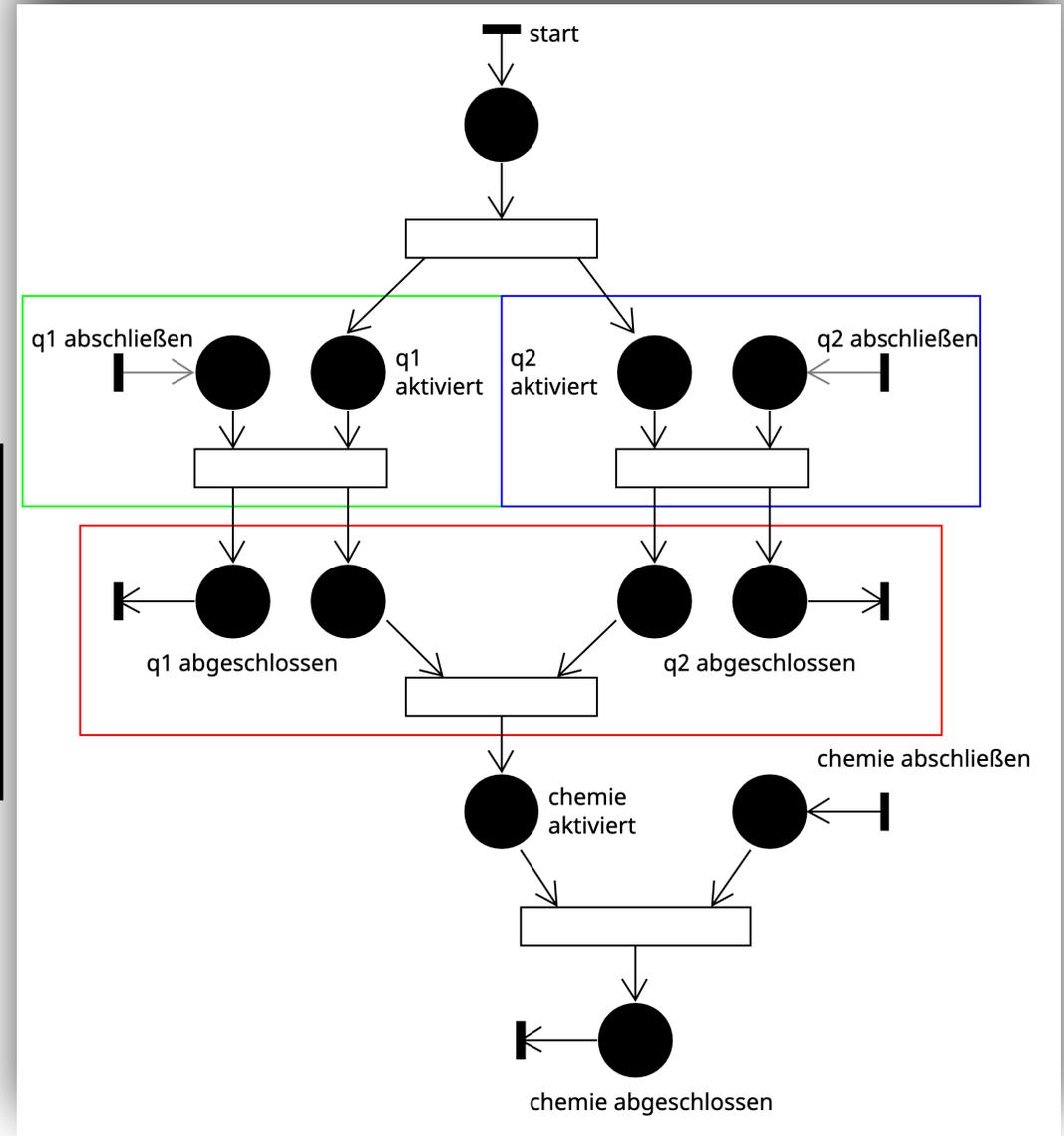
# BEISPIEL-AUFGABE: KOCHEN MIT FOTOSYNTHESE



```
single_choice_task q1 {
  description: "Was ergibt 3+3?",
  answers: ["6", "3", "12", "0"],
  correct_answer_index: 0
}
```

```
multiple_choice_task q2 {
  description: „Welche Elemente gehören zur
  Fotosynthese?“,
  answers: ["C", "AU", "O2", "F"],
  correct_answer_indices: [0, 2]
}
```

```
task_dependency t {
  q1 → chemie [type=„seq“]
  q2 → chemie [type=„seq“]
}
```



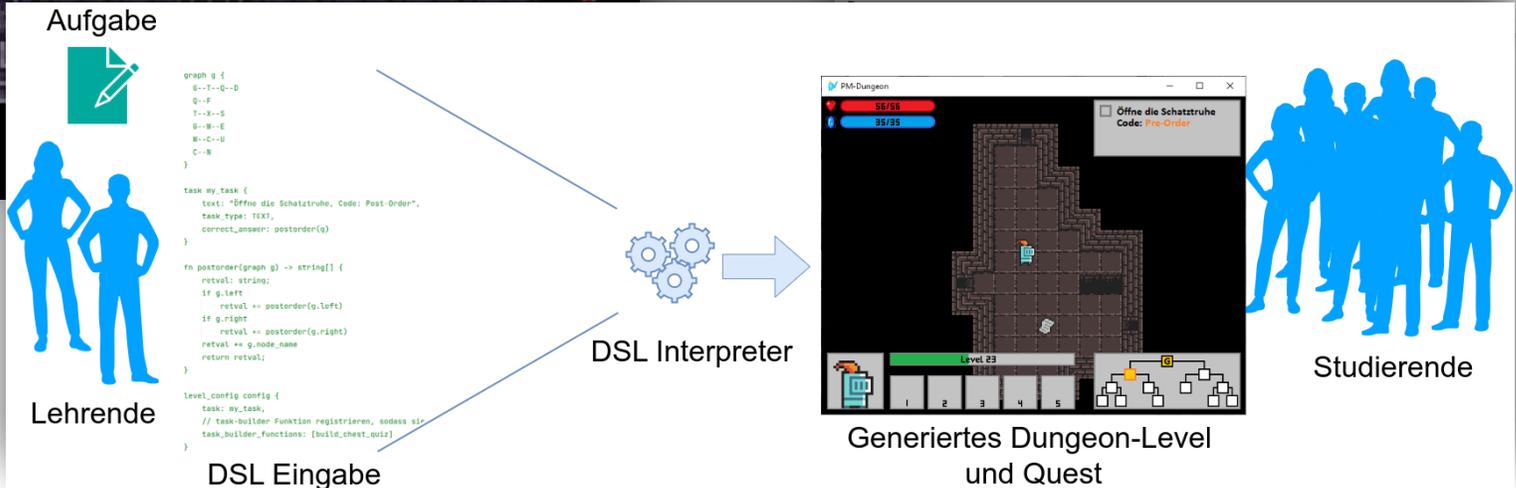
- Einsetzbar in unterschiedlichen Fachbereichen und für verschiedene Aufgabentypen
- Kein Entwickler Know-How notwendig
- Beschränkt auf "generische" Aufgabentypen
- Geringere Komplexität der Aufgaben im Vergleich zu spezifischen Anwendungen
- Unser Ansatz ergänzt das normale Lernprogramm und soll es nicht ersetzen.

- Usability: Entwicklung einer Benutzeroberfläche für die einfache Erstellung von Aufgaben
- Usability: Implementierung eines dynamischen Hilfesystems für Studierende
- Bewertung: Integration von LLM
- Erweiterung: Analyse von Spielertypen und Erstellung typenabhängiger Spielszenarien
- Verbreitung: Erforschung der Einsatzmöglichkeiten außerhalb von Hochschulen

# ZUSAMMENFASSUNG



```
replacement_task·chemie·{
  ·description: "Führen Sie Fotosynthese durch!"
  ·initial_element_set: {10x"C", 10x"O2", 10x"H2O"}
  ·elements: {
    ·n1: ("C", "O2"),
    ·n2: {"CO2"},
    ·n3: {6x"CO2", 6x"H2O"},
    ·n4: {"C6H12O6", "6O2"}
  },
  ·rules: {
    ·r1: n1 -> n2,
    ·r2: n3 -> n4
  },
  ·answer_configuration: ["C6H12O6", "6O2"],
}
```





This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).