

Objektorientiertes Programmieren

Übungsbeispiele

1 Oasencrawler mit Klassen

Verwenden Sie das erste Beispiel als Basis und erweitern Sie die structs zu Klassen. Folgende Punkte müssen dabei beachtet werden:

- Verwenden Sie nun statt structs Klassen. Beachten Sie dabei die richtige Verwendung von Zugriffsmodifikatoren, Gettern/Settern, den verschiedenen Konstruktoren und Destruktoren sowie der sinnvollen Strukturierung von Funktionen und Daten.
- Bilden Sie Ihre Klassen in einem entsprechenden Diagramm ab.
- Überprüfen Sie alle Parameterübergaben an Funktionen und Benutzereingaben auf Fehler und verhindern Sie so, dass Ihr Programm bei ungültigen Eingaben nicht mehr richtig funktioniert, Eingaben sollen so lange wiederholt werden, bis sie korrekt sind und der Spielfluss erst dann fortgesetzt werden.
- Testen Sie Ihren Code ausgiebig und berücksichtigen Sie Randbedingungen.

Stufe 1

Siehe Beispiel 1, verwenden Sie Klassen statt structs.

Stufe 2

Der Charakter soll nun zusätzlich drei verschiedene Attribute besitzen. Die Gefahrenfelder sollen einem dieser Attribute entsprechen. Betritt der Charakter nun ein Gefahrenfeld, muss er mit seinem entsprechenden Attribut eine Probe bestehen, wie diese aussieht, können Sie selbst gestalten. Besteht er diese Probe, verliert er keinen Lebenspunkt.

Zusätzlich soll der Charakter bei Brunnen und Relikten Gegenstände finden können. Diese Gegenstände entsprechen ebenfalls einem der drei Attribute und haben einen Effekt, der ausgelöst wird, wenn der Charakter auf ein Gefahrenfeld des entsprechenden Attributs geht, auch hier sollen Sie den Effekt selbst gestalten. Zum Beispiel könnte er einen Unsichtbarkeitstrank finden, der ihn eine fehlgeschlagene Attributprobe ignorieren lässt und dann abgelegt wird.

Stufe 3

Das Spiel soll nun nicht mehr aufhören, wenn der Charakter alle Relikte findet, stattdessen soll die Welt mit neuen Feldern befüllt und das Spiel dann fortgesetzt werden. Immer wenn der Charakter alle Relikte gefunden hat, soll weiters der Schwierigkeitsgrad des Spiels ansteigen, wie genau Sie das ausgestalten wollen, bleibt Ihnen überlassen.

Bewertung

Aspekt	Bewertung
Klassen	25%
Diagramm	10%
Fehlerprüfung	5%
Stufe 1	30%
Stufe 2	15%
Stufe 3	15%