Objektorientiertes Programmieren

Übungsbeispiele

1 Oasencrawler

Implementieren Sie eine zweidimensionale Spielwelt, die von einem benutzergesteuerten Charakter erforscht wird. Der Charakter gewinnt das Spiel, wenn er alle Relikte, die in der Welt verteilt sind, findet und verliert, wenn er durch die Gefahren der Welt stirbt.
Folgende Punkte müssen dabei beachtet werden:

- 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

Folgende Elemente sollen als Klassen implementiert werden:

- Die Spielwelt: Verwenden Sie ein zweidimensionales Array mit den Dimensionen 5x5 um die einzelnen Spielfelder darzustellen. Je nachdem, welcher Wert gespeichert wird, handelt es sich um ein anderes Feld. Die Spielwelt soll zu Beginn zufällig generiert werden, verwenden Sie dazu die rand()-Funktion um zufällige Werte zu generieren und damit bestimmen zu können, welche Art von Feld erstellt wird. Es gibt verschiedene Typen von Feldern mit ihren jeweiligen Wahrscheinlichkeiten bei der Erstellung:
  - Leere Felder $\frac{4}{10}$: Betritt ein Charakter ein leeres Feld passiert nichts.
  - Gefahren $\frac{4}{10}$: Betritt ein Charakter Gefahrenfelder, wird er mit einer Wahrscheinlichkeit von $\frac{1}{6}$ verletzt und verliert einen Lebenspunkt, hat er dann keine Lebenspunkte mehr, verliert er das Spiel.
  - Brunnen $\frac{1}{10}$: Der Charakter kann sich ausruhen und erhält einen Lebenspunkt.
  - Relikte $\frac{1}{10}$: Der Charakter erhält einen Reliktpunkt, gibt es dann keine Relikte mehr, gewinnt er das Spiel. Bei der Erstellung der Welt muss sichergestellt werden, dass zumindest ein Relikt vorhanden ist.

Alle Felder werden zu leeren Feldern, nachdem der Charakter sie betreten hat.

- Der Charakter: Der Charakter verfügt über einen Lebenspunkte- und Reliktpunktewert. Weiters soll seine Position in der Spielwelt durch eine x- und y-Koordinate angegeben werden. Zu Beginn hat der Charakter 5 Lebens- und 0 Reliktpunkte. Er startet bei den Koordinaten $(0|0)$

Nachdem die Spielwelt erstellt wurde, soll es möglich sein, den Charakter durch Benutzereingaben zu steuern. Nach jeder Bewegung soll das Feld, auf das sich der Charakter bewegt hat, abgehandelt werden, solange bis eine der Abbruchbedingungen für das Spiel erreicht wurde.

Stufe 2

Erweitern Sie das Spiel um einen oder mehrere Gegner, die den Charakter jagen und das Spiel so erschweren. 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.

Stufe 3

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 Unsichtbarkeit strank finden, der ihn eine fehlgeschlagene Attributsprobe ignorieren lässt und dann abgelegt wird.

Anforderungen

- Fehlerüberprüfung der Inputs
- Erstellung eines UML-Diagramms, welches die Klassenstruktur darstellt
- Stufe 1
  - Befüllung der Spielwelt (2D-Array) mit zufälligen Werten (Leere Felder, Gefahrenfelder, Brunnen, Relikte)
  - Implementierung des Charakters (Lebenspunkte, Reliktpunkte) inkl. Steuerung
  - Visualisierung der Spielwelt und des Charakters als Konsolenausgabe
  - Endlosschleife für das Spiel
- Stufe 2

  - Implementierung von mindestens einen Gegner mit Hindernislogik
  - Neugenerierung der Spielwelt nach dem Finden aller Relikte
  - Erhöhung des Schwierigkeitsgrads nach dem Finden aller Relikte
- Stufe 3
  - Implementierung von 3 Attributwerten für den Charakter
  - Überprüfung der Attribute bei Interaktionen mit Gefahrenfeldern
  - Implementierung von mindestens 3 Gegenständen (mindestens 1 je Attribut)

Bewertung

Aspekt	**Bewertung**
Fehlerprüfung	10%
Klassendiagramm	10%
Stufe 1	30%
Stufe 2	30%
Stufe 3	20%