

Monsterkampf-Simulator (K1, S1, S4)

Fällig 17. Nov bis 23:59 Punkte 0

Monsterkampf-Simulator (K1, S1, S4)

Erstelle dir vor Beginn der Arbeit einen Zeitplan in Form einer Tabelle mit 3 Spalten. Auf diesem schreibst du dir die einzelnen Meilensteine deiner geplanten Umsetzung in die erste Spalte auf. In die zweite Spalte schreibst du auf wie lange deiner Meinung nach du für das abschließen des Meilensteins voraussichtlich brauchst. Bei deiner Arbeit misst du deine Zeit. Wenn der Meilenstein abgeschlossen ist, schreibst du in die dritte Spalte, wie lange du tatsächlich dafür gebraucht hast.

Schreibe ein Programm, das den Kampf zwischen zwei Monstern simuliert. Hierzu sollen zwei Monster mit Hilfe von objektorientierter Programmierung erstellt werden.

Erstelle eine Klasse die ein Monster repräsentiert. Dieses Monster muss mindestens folgende Attribute besitzen: Lebenspunkte (HP), Angriffsstärke (AP), Abwehrrpunkte (DP), Geschwindigkeit (S) und Rasse. Mögliche Rassen sind Ork, Troll und Goblin. Alle Monster besitzen eine Methode zum attackieren, die "Attack" heißen soll.

Zu Beginn soll das Programm nach einer Eingabe fragen, um festzulegen welche zwei Monster gegeneinander antreten sollen. Dies wird durch Integer-Werte realisiert (1 = Ork, 2 = Troll, 3 = Goblin) gefolgt von 4 Float-Werten zur Bestimmung der oben genannten Attribute, welche der Benutzer selber eingeben kann. Die Reihenfolge HP - AP - DP - S ist dabei einzuhalten. Das Programm darf keinen Kampf zwischen Monstern der gleichen Rasse erlauben.

Im Anschluss berechnet das Programm wer den Kampf gewinnt. Die Formel dazu lautet: Schaden = Angriffsstärke - Abwehrrpunkte. Sollte sich für den Schaden ein negativer Wert ergeben wird dieser auf 0 gesetzt. Der Schaden wird von den Lebenspunkten des jeweiligen Monsters abgezogen. Sollte das Monster nach einer Attacke noch Lebenspunkte übrig haben führt dieses nun einen Angriff aus. Dies wiederholt sich bis ein Monster stirbt.

Die Angriffsmechanik wird von der Attack-Methode der Monster-Klasse realisiert. Das Monster mit dem höheren Geschwindigkeitswert führt den ersten Angriff aus. Nachdem ein Monster besiegt wurde gibt das Programm aus wer gewonnen hat und wie viele Runden der Kampf dauerte.

Tipps:

- Wende dich bei Problemen an dein Fachbereichs-Team
- Lass andere aus deinem Kurs deinen Code testen um mögliche Fehler zu finden.
- Fange unzulässigen User-Input ab (z.B. Wertebereich der Eingangsdaten, Eingabe nicht-numerischer Werte, Datenvalidierung)
- Mache Dir Gedanken zu Extrembedingungen (unendlicher Kampf, Attributwertkonflikte)
- Achte auf Datenkapselung
- Vermeide Goto und Magic-Numbers

Lernziele:

K1 - Erklärung der Prinzipien des objektorientierten Programmierens

S1 - Erstellen einer stabilen Codebasis mit Hilfe objektorientierter Programmierung

S4 - Planung von Terminen und Ressourcen, um Projektziele und -ergebnisse erfolgreich zu erreichen.

Feedbackelemente:

Achte auf folgende Elemente und notiere dir dazu Feedback von deiner Fachbetreuung:

Proficiency:

- Läuft das Programm stabil und fehlerfrei?
- Ist die User-Eingabe verständlich gestaltet?
- Wird der Kampf zwischen den Monstern korrekt simuliert?

Process:

- Wurden eine einheitliche Coding-Convention verwendet?
- Wurde die Vererbung von Klassen sinnvoll verwendet?
- Ist die Spielelogik in sich schlüssig?

Person:

- Wie gut ist der erstellte Code verständlich und lesbar?
- Wie kreativ wurde mit der Aufgabenstellung umgegangen?