
Programmieren – Wintersemester 2021/22

Abschlussaufgabe 2 20 Punkte

Version 1.2

Ausgabe: 28.02.2022, ca. 06:00 Uhr
Abgabe: 14.03.2022, 13:00 Uhr
Abgabefrist: 28.03.2022, 06:00 Uhr



Geschlechtergerechte Sprache

Wenn das generische Maskulinum gewählt wurde, geschieht dies zur besseren Lesbarkeit und zum einfachen Verständnis der Aufgabenstellung. Sofern nicht anders angegeben, beziehen sich Angaben im Sinne der Gleichbehandlung auf Vertretende aller Geschlechter.



Bei Fragen

In unseren *FAQs* finden Sie einen Überblick über häufig gestellte Fragen und die entsprechenden Antworten zum Modul „Programmieren“. Bitte lesen Sie diese sorgfältig durch, noch bevor Sie Fragen stellen, und überprüfen Sie diese regelmäßig und eigenverantwortlich auf Änderungen.

<https://sdqweb.ipd.kit.edu/wiki/Programmieren/FAQ>

Abgabehinweise

Bitte beachten Sie, dass das erfolgreiche Bestehen der verpflichtenden Tests für eine erfolgreiche Abgabe von Abschlussaufgabe 2 notwendig ist. Ihre Abgabe wird automatisch mit null Punkten bewertet, falls eine der nachfolgenden Regeln verletzt ist. Sie müssen zuerst die verpflichtenden Tests bestehen, bevor die anderen Tests ausgewertet werden können. Planen Sie entsprechend Zeit für Ihren ersten Abgabeversuch ein.

- Achten Sie auf fehlerfrei kompilierenden Programmcode.
- Verwenden Sie keine Elemente der Java-Bibliotheken. Ausgenommen sind Elemente der Pakete `java.lang`, `java.io`, `java.util`, `java.util.regex`, `java.util.function` und `java.util.stream`.
- Achten Sie darauf, nicht zu lange Zeilen, Methoden und Dateien zu erstellen. Sie müssen bei Ihren Lösungen eine maximale Zeilenbreite von 120 Zeichen einhalten.

- Halten Sie alle Whitespace-Regeln ein.
- Halten Sie alle Regeln zu Variablen-, Methoden- und Paketbenennung ein.
- Wählen Sie geeignete Sichtbarkeiten für Ihre Klassen, Methoden und Attribute.
- Nutzen Sie nicht das `default`-Package.
- `System.exit()` und `Runtime.exit()` dürfen nicht verwendet werden.
- Halten Sie die Regeln zur JavaDoc-Dokumentation ein.
- Halten Sie auch alle anderen Checkstyle-Regeln ein.

Bearbeitungshinweise

Diese Bearbeitungshinweise sind relevant für die Bewertung Ihrer Abgabe. Dennoch wird Ihre Abgabe durch das Abgabesystem **nicht** automatisch mit null Punkten bewertet, falls eine der nachfolgenden Regeln verletzt ist.

- Fügen Sie außer Ihrem u-Kürzel keine weiteren persönlichen Daten zu Ihren Abgaben hinzu.
- Beachten Sie, dass Ihre Abgaben sowohl in Bezug auf objektorientierte Modellierung als auch Funktionalität bewertet werden. Halten Sie die Hinweise zur Modellierung im ILIAS-Wiki ein.
- Programmcode muss in englischer Sprache verfasst sein.
- Kommentieren Sie Ihren Code angemessen: So viel wie nötig, so wenig wie möglich.
- Die Kommentare sollen einheitlich in englischer oder deutscher Sprache verfasst werden.
- Geben Sie im JavaDoc-Autoren-Tag nur Ihr u-Kürzel an.
- Wählen Sie aussagekräftige Namen für alle Ihre Bezeichner.

Plagiat

Es werden nur selbstständig angefertigte Lösungen akzeptiert. Das Einreichen fremder Lösungen, seien es auch nur teilweise Lösungen von Dritten, aus Büchern, dem Internet oder anderen Quellen, ist ein Täuschungsversuch und führt zur Bewertung „nicht bestanden“. Ausdrücklich ausgenommen hiervon sind Quelltextschnipsel von den Vorlesungsfolien und aus den Lösungsvorschlägen des Übungsbetriebes in diesem Semester. Alle benutzten Hilfsmittel müssen vollständig und genau angegeben werden. Alles, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde, muss deutlich kenntlich gemacht werden. Ebenso stellt die Weitergabe einer Lösung oder von Teilen davon eine Störung des ordnungsgemäßen Ablaufs der Erfolgskontrolle dar. Dieser Ordnungsverstoß kann ebenfalls zum Ausschluss von der Erfolgskontrolle führen.

✓ Checkstyle

Das Online-Einreichungssystem überprüft Ihre Quelltexte während der Abgabe automatisiert auf die Einhaltung der Checkstyle-Regeln. Es gibt speziell markierte Regeln, bei denen das Online-Einreichungssystem die Abgabe mit null Punkten bewertet, da diese Regeln verpflichtend einzuhalten sind. Andere Regelverletzungen können zu Punktabzug führen. Sie können und sollten Ihre Quelltexte bereits während der Entwicklung auf die Regeleinhaltung überprüfen. Das Programmieren-Wiki im ILIAS beschreibt, wie Checkstyle verwendet werden kann.

👥 Interaktive Benutzerschnittstelle

Da wir automatische Tests Ihrer interaktiven Benutzerschnittstelle durchführen, müssen die Ausgaben exakt den Vorgaben entsprechen. Insbesondere sollen sowohl Klein- und Großbuchstaben als auch die Leerzeichen und Zeilenumbrüche genau übereinstimmen. Setzen Sie nur die in der Aufgabenstellung angegebenen Informationen um. Geben Sie auch keine zusätzlichen Informationen aus. Bei Fehlermeldungen dürfen Sie den englischsprachigen Text frei wählen, er sollte jedoch sinnvoll sein. Jede Fehlermeldung muss aber mit **Error**, beginnen und darf keine Sonderzeichen, wie beispielsweise Zeilenumbrüche oder Umlaute, enthalten.

⚠️ Prüfungsmodus in Artemis

Wenn Sie mit einer Abschlusssaufgabe fertig sind oder die Zeit abläuft, müssen Sie in Artemis explizit zum zugehörigen Endbildschirm navigieren. Entweder geschieht dies durch Anklicken der Schaltfläche „Vorzeitig abgeben“ oder automatisch, wenn die Bearbeitungszeit abgelaufen ist. *Ihre Abschlusssaufgabe wird mit null Punkten bewertet, wenn Sie diese nicht explizit selbstständig abschließen!* In dem entsprechenden Endbildschirm sollten Sie bestätigen, dass Sie alle Regeln eingehalten haben und mit Ihrem vollen Namen unterschreiben. Sobald Sie so Ihre Abschlusssaufgabe abgeschlossen haben, können keine weiteren Änderungen an dieser vorgenommen werden.

Abgabemodalitäten

Die Abgabe im Online-Einreichungssystem¹ wird am **Montag, den 14. März 2022 um 13:00 Uhr**, freigeschaltet. Achten Sie unbedingt darauf, Ihre Dateien im Einreichungssystem bei der richtigen Aufgabe vor Ablauf der Abgabefrist hochzuladen. Beginnen Sie frühzeitig mit dem Einreichen, um Ihre Lösung dahingehend zu testen, und verwenden Sie das Forum, um eventuelle Unklarheiten zu klären.

Geben Sie online Ihre *.java-Dateien zur Aufgabe A in Einzelarbeit mit der entsprechenden Ordnerstruktur im zugehörigen src-Verzeichnis ab.

¹<https://artemis.praktomat.cs.kit.edu/>

Aufgabe A: Runa's Strive (20 Punkte)

Runa's Strive ist ein rundenbasiertes, Rogue-like² Kartenspiel, bei dem die Heldin *Runa* mit Fähigkeiten gegen Monster in Verliesen und Wäldern kämpft, um neue Fähigkeiten zu erhalten und schließlich die Boss-Monster über verschiedene Spiel-Level hinweg zu besiegen.

Implementieren Sie *Runa's Strive* wie nachfolgend beschrieben. Abschnitt A.1 führt die zentralen Spielelemente ein. Anschließend folgt in Abschnitt A.2 eine generelle Übersicht über den Spielablauf. In Abschnitt A.3 werden die Fähigkeitskarten und Monsterkarten eingeführt und aufgelistet. Abschließend wird in Abschnitt A.4 die zu implementierende Schnittstelle erläutert.

A.1 Spielelemente

In diesem Abschnitt soll es zunächst um die Elemente und das Konzept des Spiels gehen. Beachten Sie, dass der detaillierte Spielablauf in Abschnitt A.2 beschrieben wird.

Sie spielen als Heldin *Runa*, die, um das Spiel zu gewinnen zwei Boss-Monster besiegen muss. Das Spiel hat zwei *Level*, die sich jeweils in vier *Räume* aufteilen. In jedem Raum muss Runa ein oder zwei Monster in runden-basierten Kämpfen besiegen. Im ersten Raum muss Runa genau ein Monster besiegen. Im zweiten und dritten Raum folgen jeweils zwei Monster. Im vierten Raum kämpft Runa gegen das Boss-Monster des Spiel-Levels.

A.1.1 Karten

Im Spiel gibt es zwei Arten von Karten. Zum einen Fähigkeitskarten, welche Runa spielen kann, um gegen Monster zu kämpfen. Fähigkeitskarten definieren die Fähigkeiten, welche Runa im Kampf verwenden kann. Die Karten und deren Fähigkeiten werden in A.3.1 eingeführt.

Zum anderen gibt es Monster-Karten. Diese definieren die Abfolge von Fähigkeiten, die die Monster spielen, um Runa zu besiegen. Im Gegensatz zu den Fähigkeiten von Runa können die Fähigkeiten der Monster Vorbedingungen haben, die erfüllt sein müssen, damit ein Monster eine Fähigkeit nutzen kann. Die Details zu den Monster-Karten, den Vorbedingungen und den Fähigkeiten finden Sie in A.3.1.4.

A.1.2 Würfel

Runa benutzt einen Würfel, um einige ihrer Angriffe zu verstärken. Außerdem entspricht die höchste Augenzahl ihres Würfels ihren maximalen *Fokus-Punkten*. Die aktuellen Fokus-Punkte von Runa haben Auswirkungen auf die Stärke einiger Fähigkeitskarten. Gleichzeitig erzeugen oder verbrauchen einige Fähigkeitskarten auch Fokus-Punkte. Im Spiel startet Runa mit dem vierseitigen Würfel $d4$. Im Verlauf des Spiels kann sie ihren Würfel durch andere Würfel ersetzen. Diese sind $d6$, $d8$, $d10$, $d12$. Die Zahl im Namen des Würfels gibt hierbei die Anzahl der Seiten des Würfels an. Ein Würfel dn hat die Zahlen $[1, n]$ (beispielsweise ist $d6$ der sechsseitige Würfel mit den Zahlen 1, 2, 3, 4, 5, 6).

²<https://de.wikipedia.org/wiki/Rogue-like>

A.1.3 Runas Klasse

Runa muss zu Beginn des Spiels ihre *Klasse* festlegen. Sie kann zwischen den Klassen *Mage*, *Warrior* und *Paladin* wählen. Ihre Klasse definiert, welche Fähigkeitskarten sie zu Beginn jedes Spiel-Levels erhält. Das Fähigkeitslevel der Karten, die sie zu Beginn eines Spiel-Levels erhält, entspricht hierbei diesem Spiel-Level. In der Tabelle A.1 werden die möglichen Klassen von Runa aufgelistet. Beispielsweise Runa zu Beginn des zweiten Spiel-Levels als *Mage* die Fähigkeitskarten *Focus(2)* und *Water(2)* erhalten. Die Details zu den Fähigkeitskarten werden später in Abschnitt A.3.1 erläutert.

Klasse	Fähigkeiten, die Runa zu Beginn eines Spiel-Levels n erhält
Mage	Focus(n), Water(n)
Warrior	Thrust(n), Parry(n)
Paladin	Slash(n), Reflect(n)

Tabelle A.1: Runas mögliche Klassen

A.2 Übersicht des Spielablaufs

In Abbildung A.1 wird der generelle Spielablauf zusammengefasst, der im Folgenden genauer erläutert wird.

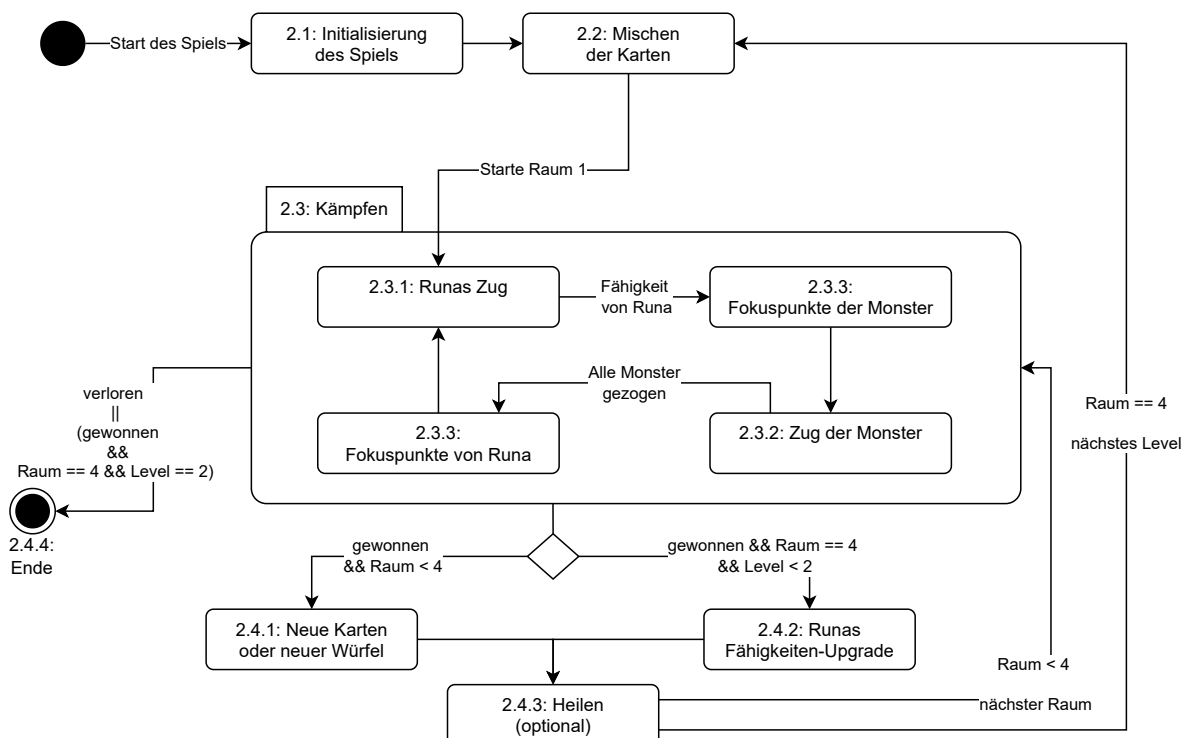


Abbildung A.1: Genereller Spielablauf (Zahlen verweisen auf die folgenden Kapitel)

Die Abbildung dient der Veranschaulichung der verschiedenen Abschnitte des Spiels. Die Rechtecke stellen Abschnitte des Spiels dar. Die Aktionen, die in den einzelnen Abschnitten des Spiels durchgeführt werden, sind in den zugehörigen Abschnitten beschrieben. Die Zahlen in den Rechtecken verweisen auf die entsprechenden Abschnitte des Textes. Die Pfeile zwischen den Rechtecken stellen Übergänge zwischen den Abschnitten dar. Beachten Sie, dass die Zusammenhänge zwischen den verschiedenen Abschnitten und Übergängen des Spiels im Folgenden auch textuell erklärt werden.

A.2.1 Initialisierung des Spiels

Zu Beginn des Spiels muss dieses initialisiert werden. Hierzu wird die Klasse von Runa gewählt (vgl. Abschnitt A.1.3). Abhängig von ihrer Klasse erhält Runa die entsprechenden Fähigkeitskarten mit Fähigkeitslevel 1. Runa startet mit ihren maximalen 50 Lebenspunkten (50 *HP*) das Spiel.

A.2.2 Mischen der Karten

Zu Beginn jedes Spiel-Levels müssen zunächst die Fähigkeitskarten gemischt werden. Alle Fähigkeitskarten werden in Abschnitt A.3.1 in den Tabellen Tabelle A.2 und Tabelle A.3 sortiert aufgelistet. Für das Spiel-Level n werden alle Fähigkeitskarten mit dem Fähigkeitslevel n gemischt. Zu beachten ist hier, dass die zwei Karten, die Runa aufgrund Ihrer Klasse erhalten hat, nicht in der Menge der zu mischenden Karten enthalten ist. Nachdem die Fähigkeitskarten gemischt worden sind, müssen danach die Monsterkarten des Spiel-Levels gemischt werden. Hierbei gilt, dass nur die normalen Monsterkarten und nicht die Karte des Boss-Monsters gemischt werden. Die Karte des Boss-Monsters ist beim Mischen damit nicht in der Menge der zu mischenden Monsterkarten enthalten. Die Details zu den Monsterkarten der verschiedenen Spiel-Level werden in A.3.1.4 beschrieben.

A.2.3 Kämpfen

Wenn Runa einen Raum betritt, werden je nach Nummer des Raums eine oder zwei Monsterkarten vom gemischten Monsterkarten-Stapel gezogen. Die gezogenen Karten definieren gegen welche Monster Runa in dem Raum kämpfen muss. Während des Kampfes wechseln sich Runa und die Monster ab, wobei Runa mit ihrem Zug beginnt. Beachten Sie, dass es offensive und defensive Fähigkeiten gibt. Offensive Fähigkeiten fügen dem Ziel physischen oder magischen Schaden zu, während defensive Fähigkeiten den physischen oder magischen Schaden einer Fähigkeit reduzieren. Offensive Fähigkeiten unterteilen sich in physische und magische Angriffe. Wobei gilt, dass physische Angriffe physischen Schaden und magische Angriffe magischen Schaden verursachen. Defensive Fähigkeiten unterteilen sich analog in physische und magische defensive Fähigkeiten. Bei ihnen gilt analog, dass sie entweder den physischen oder den magischen Schaden von Angriffen reduzieren. Die Details zu den verschiedenen Fähigkeiten von Runa und den Monstern finden Sie in Abschnitt A.3.

Ein Monster ist *besiegt* und aus dem Kampf ausgeschieden, wenn es keine Lebenspunkte mehr hat ($HP \leq 0$). Sobald es besiegt ist, kann es keine Aktionen mehr ausführen. Falls alle Monster eines Raums besiegt wurden, so hat Runa den Kampf *gewonnen*. Genauso gilt, dass wenn Runas Lebenspunkte null erreichen (≤ 0), sie stirbt und den Kampf und das Spiel *verloren* hat.

A.2.3.1 Runas Zug Wenn Runa an der Reihe ist, kann sie genau eine ihrer Fähigkeitskarten spielen. Bei allen offensiven Fähigkeiten muss Runa eines der Monster als Ziel auswählen. Während die offensiven Fähigkeiten sofort ausgelöst werden, reduzieren defensive Fähigkeiten (Für Runa: *Parry* oder *Reflect*) den Schaden durch Angriffe der Monster im folgenden Zug der Monster.

Die Angriffe von mehreren Monstern eines Zuges werden getrennt voneinander ausgewertet. Zum Verständnis folgt ein Beispiel mit der defensiven Fähigkeit *Parry(1)* (Parry mit Fähigkeitslevel 1). Die Fähigkeit *Parry(1)* hat den Effekt, dass sie den physischen Schaden der Angriffe der Monster im nächsten Zug der Monster um 7 reduziert (vgl. Abschnitt A.3.1). Falls Runa gegen zwei Monster kämpft, reduziert Runa den Angriff des ersten Monsters um 7 physischen Schaden und auch den Angriff des zweiten Monsters um 7 physischen Schaden.

A.2.3.2 Zug der Monster Wenn die Monster am Zug sind, führen diese jeweils eine ihrer Fähigkeiten aus. Sollten zwei Monster im Raum sein, spielen sie in einem Zug nacheinander je eine Fähigkeit aus. Hierbei beginnt das Monster, das zuerst vom Kartenstapel gezogen worden ist. Die Fähigkeit, die ein Monster ausführt, wird durch die Fähigkeiten auf der Monsterkarte bestimmt. Die Details zu den Monstern eines Spiel-Levels und zu deren Fähigkeiten finden Sie in A.3.1.4. Im ersten Zug wird die erste Fähigkeit eingesetzt, im zweiten Zug die zweite und so weiter. Sobald die letzte Fähigkeit benutzt wurde, nutzt das Monster wieder die erste Fähigkeit. Ist die Bedingung (vgl. Kosten in Tabelle A.4) einer Fähigkeit eines Monsters nicht erfüllt, überspringt das Monster diese Fähigkeit und verwendet die nächste Fähigkeit die möglich ist, sodass jedes Monster in einem Kampf immer eine Aktion ausführt. Beachten Sie, dass Monster immer mindestens eine Fähigkeit besitzen, die ohne Bedingung ausgeführt werden kann, sodass ein Monster immer eine Fähigkeit verwenden kann. Die offensiven Fähigkeiten (beispielsweise *Claw* oder *Smash*) eines Monsters richten sich immer gegen Runa. Beachten Sie bitte, dass die defensiven Fähigkeiten eines Monsters (beispielsweise *Block* oder *Deflect*) nur während des nächsten Zugs von Runa wirken.

A.2.3.3 Fokus-Punkte Fokus-Punkte sind essenziell für die magischen Angriffe von Runa und auch der Monster. Die magischen Fähigkeiten brauchen als Vorbereitung eine gewisse Anzahl an Fokus-Punkten (vgl. Kosten in Abschnitt A.3). Durch die *Focus*-Fähigkeit können die Fokus-Punkte von Runa oder der Monster erhöht werden. Da der Effekt einer *Focus*-Karte aber durch spezielle Angriffe gebrochen werden kann, wirkt der Effekt einer *Focus*-Karte erst direkt vor dem nächsten Zug von Runa bzw. der Monster. Es gilt insbesondere, dass der Einsatz der *Focus*-Fähigkeit nicht direkt nach dem Einsatz die Fokus-Punkte des spielenden Charakters (Runa oder Monster) erhöht. Das Brechen des *Focus*-Effekts hat nur zur Folge, dass die zuletzt gewirkte *Focus*-Fähigkeit keine Wirkung hat. Auf bereits vorhandene Fokus-Punkte hat dies keinen Einfluss. Beachten Sie außerdem, dass Runa ihre aktuellen Fokus-Punkte auch nach einem Kampf behält und diese nicht zurückgesetzt werden.

A.2.4 Nach dem Kampf gegen Monster

Ein Kampf gegen Monster kann auf zwei Arten beendet werden. Falls Runas Lebenspunkte null erreichen ($HP \leq 0$), so stirbt sie und hat das gesamte Spiel *verloren*. Falls alle Monster besiegt worden sind, so hat sie den Kampf *gewonnen*. Nach dem Ende eines Kampfes gibt es drei mögliche

nächste Schritte im Spiel. Diese sind abhängig von der Nummer des Spiel-Levels, von der Nummer des Raums und dem Ausgang des Kampfes.

A.2.4.1 (a) Neue Karten oder neuer Würfel Falls Runa den Kampf gewonnen hat und es sich nicht um den Raum eines Boss-Monsters gehandelt hat, darf Runa wählen: Entweder sie verbessert ihren Würfel oder sie darf Fähigkeitskarten ziehen. Wählt sie die Verbesserung des Würfels, so bekommt sie den nächstgrößeren Würfel (vgl. Abschnitt A.1.2). Sollte Runa bereits den Würfel mit den meisten Seiten haben, entfällt diese Option und Runa muss Karten ziehen.

Entscheidet sich Runa zum Ziehen von Fähigkeitskarten, so werden ihr $2 * m$ Karten angeboten, von denen sie m Karten wählen darf. Hierbei bezeichnet m die Anzahl der von Runa besiegten Monster des gerade bespielten Raums. Die Karten, die Runa nicht wählt, werden zur Seite gelegt und sind aus dem Spiel. Ein Randfall: Sollte sich Runa im Spiel dazu entschieden haben nach Raum 1, Raum 2 und Raum 3 eines Spiel-Levels Karten zu ziehen, so können ihr nach Raum 3 nicht $2 * m = 4$, sondern nur 3 Karten angeboten werden, aus denen sie 2 Karten ziehen darf.

A.2.4.2 (b) Runas Fähigkeiten-Upgrade Falls Runa im letzten Raum das Boss-Monster des ersten Spiel-Levels besiegt hat, erhält sie ein Fähigkeiten-Upgrade. Das heißt, dass Sie die Fähigkeitskarten ihrer Klasse (vgl. Abschnitt A.1.3) für das nächste Spiel-Level erhält. Diese haben das Fähigkeitslevel 2.

A.2.4.3 Heilen Unabhängig davon, ob Runa gerade neue Karten gewählt, ihren Würfel verbessert oder ein Fähigkeiten-Upgrade erhalten hat darf sie sich, sofern möglich, heilen, indem sie Fähigkeitskarten ablegt. Runa kann sich heilen, sofern sie weniger Lebenspunkte als ihre maximalen 50 Lebenspunkte hat ($HP < 50$) und sie noch mehr als eine Fähigkeitskarte besitzt ($|Karten| > 1$). Sollte es unter den oben genannten Bedingungen möglich sein sich zu heilen, kann Runa Karten wählen, die sie ablegen möchte. Für jede Karte, die sie welegt, erhält sie 10 Lebenspunkte. Beachten Sie, dass Runa nach dem Ablegen noch mindestens eine Karte besitzen muss und ihre Lebenspunkte die 50 nicht überschreiten können ($HP \leq 50$). Sollte Runa also nach einem Kampf beispielsweise 45 Lebenspunkte haben und eine Karte ablegen, so hat sie danach 50 Lebenspunkte.

A.2.4.4 (c) Ende des Spiels Das Spiel hat genau zwei Enden: Falls Runa durch den Angriff eines Monsters stirbt, hat Runa das Spiel *verloren*. Sollte Runa das Boss-Monster des zweiten Spiel-Levels besiegt haben, so hat Runa das Spiel *gewonnen*.

A.3 Fähigkeitskarten & Monsterkarten

Als nächstes werden die Fähigkeitskarten von Runa und auch die Monsterkarten genauer erläutert. In Tabelle A.2 werden die möglichen physischen Fähigkeiten (Fähigkeiten mit Bezug zu physischem Schaden) und in Tabelle A.3 die möglichen magischen Fähigkeiten (Fähigkeiten mit Bezug zu magischem Schaden) von Runa aufgelistet. In A.3.1.4 werden die Monsterkarten genauer beschrieben und die Monster der verschiedenen Spiel-Level eingeführt. Die folgenden Paragraphen beschreiben zunächst die Effekte der Fähigkeiten und wie diese wirken.

A.3.1 Fähigkeiten

Die Effekte von Runas Fähigkeitskarten sind abhängig vom Fähigkeitslevel n der Karten. Zusätzlich können sie von einem Würfelwurf w oder Runas aktuellen Fokus-Punkten f abhängig sein. Runa hat generell immer mindestens einen Fokus-Punkt. Im Gegensatz zu physischen Fähigkeiten haben magische Fähigkeiten die Voraussetzung, dass eine gewisse Anzahl an Fokus-Punkten (vgl. *Kosten* in Tabelle A.3 und Tabelle A.4) erreicht sein muss, damit die entsprechende Fähigkeit verwendet werden kann. Im Allgemeinen verbraucht eine magische Fähigkeit bei ihrer Verwendung die Menge an Fokus-Punkten, die in der entsprechenden Kosten-Spalte der Tabellen angegeben ist. Für Runas magischen Fähigkeiten gilt jedoch auch, dass sie nur dann einen Fokus-Punkt verbrauchen, wenn Runa mehr als einen Fokus-Punkt angehäuft hat ($> 1FP$). Somit reduziert das Verwenden einer magischen Fähigkeit Runas Fokus-Punkte nie auf null. Sie könnte somit in jedem Zug eine Fähigkeit mit magischem Schaden zum Spielen wählen. Im Gegensatz dazu beginnen alle Monster mit null Fokus-Punkten und können durch das Verwenden von magischen Fähigkeiten auch wieder null Fokus-Punkte erreichen.

A.3.1.1 Offensive Fähigkeiten Bei physischen Angriffen muss Runa ihren Würfel einmal werfen, um den Schaden zu bestimmen. Im Falle von *Thrust* oder *Pierce* wird der Schaden zusätzlich erhöht, wenn der Würfelwurf ≥ 6 war. Bei magischen Angriffen kann je nach Monster-Typ der Schaden noch einmal verstärkt werden. Es gibt Wasser-, Eis-, Blitz- und Feuer-Monster. Im Falle von *Water* wird der Schaden beispielsweise um $2n$ vergrößert, falls die Fähigkeit gegen ein Blitz-Monster gespielt wird. Bevor jedoch der physische oder magische Schaden zugefügt wird, wird der Schaden reduziert, wenn das anvisierte Monster in seinem vorherigen Zug *Block* bzw. *Deflect* benutzt hat. Runas offensive Fähigkeiten sind: Physische Angriffe, d.h. *Slash*, *Swing*, *Thrust* und *Pierce* und magische Angriffe, d.h. *Water*, *Ice*, *Fire* und *Lightning*.

A.3.1.2 Defensive Fähigkeiten Sowohl Runa als auch die Monster können defensive Fähigkeiten einsetzen, wie beispielsweise *Parry*, *Block* oder *Reflect*. Wenn Runa eine defensive Fähigkeit wie *Parry* in ihrem Zug einsetzt, wirkt sich das nur auf den direkt folgenden Zug der Monster aus und reduziert den eingehenden Schaden um den Betrag auf der Karte. Wenn sie sich zwei Monstern gegenüber sieht und beide im nächsten Zug angreifen, wird jeder einzelne Angriff abgeschwächt, d.h. der Schaden jedes einzelnen wird reduziert. Das Gleiche gilt für die defensiven Fähigkeiten der Monster, nur, dass sie sich auf den direkt folgenden Zug von Runa auswirken (vgl. Abschnitt A.2.3). Auch *Reflect* wirkt nur auf den direkt folgenden Zug der Monster, reflektiert aber den eintreffenden magischen Schaden bis zur auf der Karte angegebenen Höhe zurück auf das angreifende Monster. Wenn der eingehende magische Schaden kleiner oder gleich dem angegebenen Betrag ist, wird der eingehende Schaden vollständig reflektiert. Andernfalls wird nur der angegebene Betrag reflektiert und Runa erhält den restlichen Schaden (d.h. den eingehenden Schaden minus den angegebenen Betrag). Beachten Sie, dass in diesem Fall Runa zuerst den Schaden erhält und dann das Monster.

A.3.1.3 Focus *Focus* ist eine Fähigkeit, die Runa oder auch Monster verwenden können, um ihre Fokus-Punkte zu erhöhen. Die Fähigkeit hat im Gegensatz zu den anderen Fähigkeiten erst direkt vor dem nächsten Zug des jeweiligen Charakters (Runa oder Monster) einen Effekt. Somit gilt für Runa, falls sie *Focus* spielt, dass der Effekt nach dem Zug der Monster in Kraft tritt. Falls

ein Monster *Focus* spielt, gilt dies analog für den Zeitpunkt direkt nach dem nächsten Zug von Runa. Wichtig ist, dass der Focus-Effekt durch *Slash*, *Swing*, *Scratch* oder *Claw* gebrochen werden kann. Falls ein gegnerischer Charakter (Runa oder Monster) eine dieser Fähigkeiten anschließend verwendet, werden die Fokus-Punkte somit nicht erhöht. Beachten Sie, dass Runas Fokus-Punkte niemals ihre maximalen Fokus-Punkte (bestimmt durch ihren aktuellen Würfel) überschreiten können ($FP \leq FP_{MAX}$) und Runa ihre Fokus-Punkte zwischen *Kämpfen* (vgl. A.2.3) behält.

Name	Art	Effekt
Slash(n)	offensiv	$4n + w$ physischer Schaden Bricht <i>Focus</i> des Ziel-Monsters
Swing(n)	offensiv	$5n + w$ physischer Schaden Bricht <i>Focus</i> des Ziel-Monsters
Thrust(n)	offensiv	$6n + w$ physischer Schaden + $4n$ physischer Schaden, falls $w \geq 6$
Pierce(n)	offensiv	$7n + w$ physischer Schaden + $5n$ physischer Schaden, falls $w \geq 6$
Parry(n)	defensiv	Reduziere den physischen Schaden des nächsten Angriffs eines Monsters auf Runa um $7n$

Tabelle A.2: Runas Physische Fähigkeiten (Legende: Fähigkeitslevel n , Würfelwurf w)

Name	Kosten	Art	Effekt
Focus(n)	—	—	Erhöhe Runas Fokus-Punkte vor ihrem nächsten Zug um n , außer durch nächsten Zug der Monster gebrochen
Reflect(n)	—	defensiv	Reflektiere bis zu $10n$ magischen Schaden. Der restliche Schaden trifft Runa.
Water(n)	1 FP	offensiv	$(2n + 4) * f$ magischer Schaden + $2n$ magischer Schaden, falls Ziel ein Blitz-Monster
Ice(n)	1 FP	offensiv	$(2n + 4) * f + 2$ magischer Schaden + $2n$ magischer Schaden, falls Ziel ein Wasser-Monster
Fire(n)	1 FP	offensiv	$(2n + 5) * f$ magischer Schaden + $2n$ magischer Schaden, falls Ziel ein Eis-Monster
Lightning(n)	1 FP	offensiv	$(2n + 5) * f + 2$ magischer Schaden + $2n$ magischer Schaden, falls Ziel ein Feuer-Monster

Tabelle A.3: Runas Magische Fähigkeiten (Legende: Fähigkeitslevel n , aktuelle Fokus-Punkte f)

A.3.1.4 Monster Die Fähigkeiten der Monster sind nur abhängig von dem Level der Fähigkeiten n . Die magischen Fähigkeiten der Monster benötigen und verbrauchen immer n Fokus-Punkte. Eine Übersicht über die möglichen Fähigkeiten der Monster finden Sie in der Tabelle A.4.

Name	Kosten	Art	Effekt
Scratch(n)	—	offensiv	$5n$ physischer Schaden, Bricht Runas <i>Focus</i>
Claw(n)	—	offensiv	$6n$ physischer Schaden, Bricht Runas <i>Focus</i>
Smash(n)	—	offensiv	$8n$ physischer Schaden
Bite(n)	—	offensiv	$10n$ physischer Schaden
Focus(n)	—	—	Erhöhe die Fokus-Punkte des Monsters vor dessen nächsten Zug um n , außer Fähigkeit wird gebrochen
Block(n)	—	defensiv	Reduziere nächsten Angriff um $7n$ physischen Schaden
Deflect(n)	—	defensiv	Reduziere nächsten Angriff um $11n+2$ magischen Schaden
Water(n)	n FP	offensiv	$8n + 2$ magischer Schaden
Ice(n)	n FP	offensiv	$10n + 2$ magischer Schaden
Fire(n)	n FP	offensiv	$12n + 2$ magischer Schaden
Lightning(n)	n FP	offensiv	$14n + 2$ magischer Schaden

 Tabelle A.4: Fähigkeiten der Monster (Legende: Fähigkeitslevel n)

A.3.2 Monster der Spiel-Level 1 und 2

In Spiel-Level 1 des Spiels haben alle Monster-Fähigkeiten das Fähigkeitslevel 1. In Spiel-Level 2 des Spiels haben alle Monster-Fähigkeiten das Fähigkeitslevel 1 oder 2. Die Monster sind sortiert in Tabelle A.5 und Tabelle A.6 dargestellt.

Name	Typ	Leben	geordnete Fähigkeiten mit deren Level
Spider King	Blitz, Boss	50	Bite(1), Block(1), Focus(1), Lightning(1)
Frog	Wasser	16	Focus(1), Water(1)
Ghost	Eis	15	Focus(1), Ice(1)
Gorgon	Feuer	13	Focus(1), Fire(1)
Skeleton	Blitz	14	Focus(1), Lightning(1)
Spider	—	15	Bite(1), Block(1)
Goblin	—	12	Smash(1), Deflect(1)
Rat	—	14	Block(1), Claw(1)
Mushroomlin	—	20	Deflect(1), Scratch(1)

Tabelle A.5: Monster des Spiel-Levels 1

Name	Typ	Leben	geordnete Fähigkeiten mit deren Level
Mega Saurus	Boss	100	Bite(2), Block(2), Focus(2), Fire(1), Lightning(1)
Snake	Eis	31	Bite(2), Focus(2), Ice(2)
Dark Elf	—	34	Focus(2), Water(1), Lightning(1)
Shadow Blade	Blitz	27	Scratch(2), Focus(2), Lightning(2)
Hornet	Feuer	32	Scratch(2), Focus(2), Fire(1), Fire(2)
Tarantula	—	33	Bite(2), Block(2), Scratch(2)
Bear	—	40	Claw(2), Deflect(2), Block(2)
Mushroomlon	—	50	Deflect(2), Scratch(2), Block(2)
Wild Boar	—	27	Scratch(2), Deflect(2), Scratch(2)

Tabelle A.6: Monster des Spiel-Levels 2



Cover Art zu *Runa's Strive* (Copyright © Delta_Lightning 2022)

A.4 Implementierung von Runa's Strive

Ihre Aufgabe ist es das Spiel mittels einer Text-basierten Schnittstelle spielbar zu machen. Die dafür nötigen Interaktionen werden im Folgenden pro Phase dargestellt. Beachten Sie, dass die Regeln, die in den vorherigen Abschnitten beschrieben worden sind, immer einzuhalten sind.

A.4.1 Darstellung der Beispielinteraktionen

In Beispielinteraktionen stellt das Symbol `%>` (Prozent-Zeichen und Größer-Zeichen gefolgt von einem Leerzeichen) die Kommandozeile dar. Der Programmname ist frei gewählt und muss bei Ihnen nicht *Runa* lauten. Das Symbol `>` (Größer-Zeichen gefolgt von einem Leerzeichen) stellt eine Benutzereingabe dar und ist selbst nicht Teil der Eingabe. Sollte in einer Interaktion [...] geschrieben werden, so bedeutet dies, dass weitere Interaktionen ausgelassen worden sind, um auf eine spezielle Stelle der Interaktion einzugehen. [...] ist keine Ausgabe Ihres Spiels.

A.4.1.1 Aufforderungen zur Eingabe Aufforderungen zur Eingabe werden durch das Spiel immer mit einer der folgenden Ausgaben angezeigt. Es gilt hierbei, dass der Wertebereich der Eingabe immer abgeschlossen ist, als inklusive 1 bis inklusive n , d bzw. 2147483647.

Für das Wählen genau einer Option aus n Optionen	Enter number [1--<n>]:
Für das Wählen mehr als einer Option aus n verschiedenen Optionen	Enter numbers [1--<n>] separated by comma:
Für das Eingeben eines Würfelwurfs eines Würfels mit d Seiten	Enter dice roll [1--<d>]:
Für das Eingeben von genau zwei Seeds für das Mischen	Enter seeds [1--2147483647] separated by comma:

Beachten Sie, dass die Reihenfolge der eingegebenen Zahlen bei mehreren Optionen zu berücksichtigen ist. Somit wäre bei zwei Optionen, die aus vier Werten (1-4) gewählt werden müssen, 1,3 eine andere Auswahl als 3,1. Details zu den einzelnen Aufforderungen werden im Folgenden erläutert.

A.4.1.2 Platzhalter Im Folgenden werden einige Platzhalter definiert, die in der Beschreibung der erwarteten Ausgaben verwendet werden.

<Name>	Runa oder der Name eines Monsters
<Aktueller Wert>	Aktueller Wert der Lebens- bzw. Fokus-Punkte
<Maximaler Wert>	Maximaler Wert der Lebens- bzw. Fokus-Punkte
<Fähigkeitsname>	Name einer Fähigkeit ohne Fähigkeitslevel wie in Tabellen oben, beispielsweise Focus
<Fähigkeitslevel>	Fähigkeitslevel einer spezifischen Fähigkeit, beispielsweise 2
<Schaden>	n phy. damage oder n mag. damage für n Schaden
<FP>	Eine Anzahl von Fokus-Punkten
<DN>	Die Darstellung eines Würfels, beispielsweise d6
<HP>	Die Darstellung einer Menge von Lebenspunkten, beispielsweise 12

A.4.2 Quit-Befehl

Generell muss Ihr Programm jederzeit mit der Eingabe des Befehls `quit` beendet werden können. Beachten Sie, dass für das Testen Ihrer Abgabe dieser Befehl essenziell ist, da viele der Tests `quit` am Ende einer Testsequenz verwenden, um Ihr Programm zu beenden. Stellen Sie daher sicher, dass der Befehl in jeder Situation funktioniert. Beachten Sie bitte nochmals, dass Sie hierfür nicht `System.exit()` oder andere ausgeschlossene Funktionen verwenden dürfen.

A.4.3 Initialisierung des Spiels

Sie starten das Spiel mit keinem Kommandozeilen-Parameter. Sollten ein oder mehrere Parameter angegeben werden, wird eine aussagekräftige Fehlermeldung ausgegeben und das Programm wird beendet.

Ansonsten beginnt das Spiel mit der Begrüßung `Welcome to Runa's Strive`. Zum Spielstart muss Runa zunächst ihre *Klasse* wählen. Hierbei kann sie drei Möglichkeiten (siehe Abschnitt A.1.3). Im Folgenden wird die Interaktion zur Auswahl von Runas Klasse dargestellt.

➤ Beispielinteraktion

```
1  %> java Runa
2  Welcome to Runa's Strive
3  Select Runa's character class
4  1) Warrior
5  2) Mage
6  3) Paladin
7  Enter number [1--3]:
8  [...]
```

Generell gilt für das gesamte Spiel, dass die Interaktion mit dem Nutzer über Auswahldialoge umgesetzt wird (vgl. A.4.1.1). Einen solchen sehen Sie exemplarisch für die Auswahl der Klasse von Runa. Sollte eine Auswahl ungültig sein, wird kein Fehler ausgegeben, sondern die Aufforderung zur Eingabe wiederholt. Beachten Sie hier nochmals, dass der Befehl `quit` das Programm jeder Zeit beenden muss.

➤ Beispielinteraktion

```
1  %> java Runa
2  Welcome to Runa's Strive
3  Select Runa's character class
4  1) Warrior
5  2) Mage
6  3) Paladin
7  Enter number [1--3]:
8  > 42
9  Enter number [1--3]:
10 > quit
```

A.4.4 Mischen der Karten

Bevor ein Spiel-Level gestartet werden kann, müssen sowohl die Fähigkeitskarten von Runa für das entsprechende Spiel-Level als auch die Monsterkarten für das Spiel-Level gemischt werden. Hierfür benötigt das Spiel zwei *Seeds* für die Zufallsgeneratoren.

Zuerst werden die Fähigkeitskarten von Runa gemischt. Hierfür müssen Sie zuerst die Fähigkeiten aus Tabelle A.2 und Tabelle A.3, die noch zu mischen sind, in einer `List`³ (z. B. einer `ArrayList`⁴) abspeichern. Beachten Sie hierbei, dass Sie sich exakt an die vorgegebene Reihenfolge der Tabellen halten. Für das Spiel-Level n ist das potentiell erste Listen-Element *Slash*(n) und das letzte Element *Lightning*(n). Um nun diese Liste nach dem Zufallsprinzip unter Verwendung der angegebenen Zufallsquelle zu mischen, muss diese der `Collections::shuffle`-Methode⁵ übergeben werden.

Der zweite Seed wird im Anschluss dafür benutzt, um die Monsterkarten des aktuellen Spiel-Levels zu mischen. Als Reihenfolge wird hierbei die der Tabellen Tabelle A.5 und Tabelle A.6 verwendet. Da der Boss eines Spiel-Levels bereits für den letzten Raum gesetzt ist, ist dieser nicht Teil der zu mischenden Monsterkarten. Für die Vergabe der Monster- und Fähigkeitskarten für Runa werden aus den gemischten Listen immer Elemente beginnend bei Index 0 entnommen.

Im Folgenden ist die erwartete Interaktion zum Definieren der Seeds beispielhaft dargestellt. Beachten Sie, dass die Interaktion mit der Erklärung der Seeds beginnt und anschließend die Aufforderung zum Eingeben der Seeds ausgegeben wird. Der mögliche Wertebereich eines Seeds umfasst in dieser Aufgabe das Intervall [1, 2147483647].

► Beispielinteraktion

```
1 | [...]
2 | To shuffle ability cards and monsters, enter two seeds
3 | Enter seeds [1--2147483647] separated by comma:
4 | 23,32
5 | [...]
```

A.4.5 Kämpfen

Sobald Runa einen Raum betritt, beginnt die Kampf-Phase des Spiels. Diese läuft unabhängig von Spiel-Level und Raum immer gleich ab. Beim Start des Raums gibt das Spiel einen Hinweis in welchem Spiel-Level (*Level*) und welchem Raum (*Stage*) sich Runa befindet. Beispielhaft für das erste Spiel-Level und den ersten Raum wäre dies *Runa enters Stage 1 of Level 1*. Anschließend startet Runas Zug. Die folgende Beispielinteraktion und die nötigen Ausgaben werden im Folgenden detailliert erklärt.

³<https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/List.html>

⁴<https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/ArrayList.html>

⁵[https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/Collections.html#shuffle\(java.util.List,java.util.Random\)](https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/Collections.html#shuffle(java.util.List,java.util.Random))

➤ Beispielinteraktion

```

1  [...]
2  Runa enters Stage 2 of Level 1
3  -----
4  Runa (50/50 HP, 1/6 FP)
5  vs.
6  Spider (15 HP, 0 FP): attempts Bite(1) next
7  Gorgon (13 HP, 0 FP): attempts Focus(1) next
8  -----
9  Select card to play
10  1) Slash(1)
11  2) Reflect(1)
12  Enter number [1--2]:
13  > 1
14  Select Runa's target.
15  1) Spider
16  2) Gorgon
17  Enter number [1--2]:
18  > 1
19  Runa uses Slash(1)
20  Enter dice roll [1--6]:
21  > 6
22  Spider takes 10 phy. damage
23  Spider uses Bite(1)
24  Runa takes 10 phy. damage
25  Gorgon uses Focus(1)
26  -----
27  Runa (40/50 HP, 1/6 FP)
28  vs.
29  Spider (5 HP, 0 FP): attempts Block(1) next
30  Gorgon (13 HP, 0 FP): attempts Fire(1) next
31  -----
32  Select card to play
33  1) Slash(1)
34  2) Reflect(1)
35  Enter number [1--2]:
36  > 2
37  Runa uses Reflect(1)
38  Gorgon gains 1 focus
39  Spider uses Block(1)
40  Gorgon uses Fire(1)
41  Runa takes 4 mag. damage
42  Gorgon takes 10 mag. damage
43  [...]

```


A.4.5.1 Runas Zug Runas Zug beginnt mit einer Ausgabe des Spiels, die die aktuelle Situation im Raum beschreibt: Zur besseren Lesbarkeit trennt das Spiel die Ausgabe mithilfe von 40 Minuszeichen in den Zeilen der Übersicht. Zunächst wird angezeigt, wie der Zustand von Runa ist und gegen wen sie kämpft. Ihr Zustand umfasst ihre Lebenspunkte (<HP>) und ihre aktuellen Fokus-Punkte (<FP>). Beide werden angegeben als <Aktueller Wert>/<Maximaler Wert>.

Die Monster, gegen die sie kämpft, werden im Anschluss aufgelistet. Die Reihenfolge der Auflistung der Monster ergibt sich hierbei aus der Reihenfolge, in der die Monster gezogen wurden, wobei das zuerst gezogene Monster oben steht. Zusätzlich zu Informationen über die aktuellen Lebenspunkte und die aktuellen Fokus-Punkte wird pro Monster auch angegeben, welche Fähigkeit das Monster als Nächstes nutzen wird. Die Fähigkeit wird hierbei als <Fähigkeitsname>(<Fähigkeitslevel>) dargestellt. Hierbei ist wichtig, dass die Fähigkeit, die hier angezeigt wird, der nächsten Fähigkeit in der Liste der Fähigkeiten der jeweiligen Monster entspricht. Sie müssen insbesondere nicht prüfen, ob diese Fähigkeit im nächsten Zug der Monster überhaupt verwendbar ist oder nicht. Die Auflistung der Raumsituation endet mit einer Zeile bestehend aus 40 Minuszeichen.

Im Anschluss an diese Auflistung fordert das Programm auf, zu wählen, welche Fähigkeit Runa spielen soll. Die Reihenfolge, in der die Fähigkeiten aufgelistet werden, ergibt sich daraus, wann Runa die Fähigkeitskarte erhalten hat. Neue Fähigkeiten werden immer unten angehängt. Beachten Sie, dass das Spiel auch dann nachfragt, welche Karte Runa spielen möchte, wenn Runa nur noch eine Karte besitzt. Sollte Runa gegen mehrere (> 1) Monster kämpfen fragt das Spiel nach, gegen welches Monster die Fähigkeit gespielt werden soll. (Bei nur einem Monster oder einer defensiven Fähigkeit entfällt dieser Dialog). Die Reihenfolge der Monster entspricht auch hier der Reihenfolge, in der diese aus der sortierten Liste (vgl. Abschnitt A.4.4) entnommen worden sind. Falls die Fähigkeit und das mögliche Ziel bekannt sind, erfolgt die Bestätigung des Spiels, dass Runa eine Fähigkeit spielt. Dies wird mittels einer Ausgabe dargestellt: <Name> uses <Fähigkeitsname>(<Fähigkeitslevel>). Spielt Runa beispielsweise die Fähigkeitskarte *Water(1)*, so gibt das Spiel Runa uses Water(1) aus. Wählt Runa eine Fähigkeit, bei der gewürfelt werden muss, wird als nächstes die Augenzahl, die gewürfelt wurde, durch einen Dialog abgefragt. Anschließend wird der Schaden des Monsters ausgegeben. Physischer Schaden wird hierbei mit **phy. damage** und magischer Schaden mit **mag. damage** abgekürzt. Beachten Sie, dass das Spiel nur den endgültigen Schaden nach möglichen Abzügen durch defensive Fähigkeiten angibt. Sollte beispielsweise ein Monster oder Runa mittels *Block* allen Schaden abhalten, wird kein Text ausgegeben. Eine Ausgabe von Schaden erfolgt immer nach folgendem Schema: <Name> takes <Schaden>. Stirbt ein Charakter (Runa oder Monster) quittiert das Spiel das mit der Ausgabe <Name> dies.

Im Anschluss an den Zug von Runa erfolgt die Auswertung der Fokus-Punkte der Monster (vgl. A.4.5.3).

A.4.5.2 Zug der Monster Im Zug der Monster wenden diese nacheinander die Fähigkeit an, die als Nächstes für sie möglich ist. Jede Aktion eines Monsters wird hierbei zunächst vollständig ausgewertet. Sollte beispielsweise Runa durch den Angriff eines ersten Monsters sterben, so wird der nachfolgende Zug eines zweiten Monsters nicht mehr ausgewertet. Das Protokollieren der Aktionen erfolgt analog zu den Aktionen von Runa. D.h. es wird zunächst ausgegeben, dass ein Monster eine Fähigkeit ausspielt. Anschließend wird der Schaden berechnet und ausgegeben, wie viel Schaden Runa bzw. das Monster erleiden. Falls ein Charakter stirbt, wird dies analog zu Runas Zug ausgegeben als <Name> dies. Falls Runa nach dem Angriff des ersten Monsters noch lebt und

ein zweites Monster existiert, wird dessen Zug ausgewertet. Anschließend erfolgt die Auswertung der Fokus-Punkte von Runa (vgl. A.4.5.3).

A.4.5.3 Fokus-Punkte Zwischen dem Zug von Runa und dem Zug der Monster werden die Effekte der *Focus*-Fähigkeit ausgewertet. Sollten Runa oder ein Monster Fokus-Punkte erhalten ($\Delta FP > 0$), so wird dies vom Spiel mit `<Name> gains <FP> focus` quittiert. Sollte die *Focus*-Fähigkeit gebrochen worden sein, findet keine Ausgabe statt. Beachten Sie, dass für die Ausgabe die Differenz der ursprünglichen zu den neuen Fokus-Punkten (ΔFP) entscheidend ist.

A.4.6 Nach dem Kampf gegen Monster

Es gibt drei mögliche Enden eines Kampfes (vgl. Abschnitt A.2.4). Diese werden im Folgenden beschrieben.

A.4.6.1 Neue Karten oder neuer Würfel Diese Phase beginnt mit der Aufforderung, dass Runas Belohnung gewählt werden soll: `Choose Runa's reward`. Wie für diese Phase des Spiels bereits geschrieben, hat Runa die Wahl, ob sie einen besseren Würfel erhalten oder Karten ziehen möchte. Hierfür wird (sofern es eine Wahl für Runa gibt) ein Auswahldialog verwendet. Die Reihenfolge der Karten ergibt sich auch hier durch das Entnehmen der Karten aus der gemischten Liste beginnend bei Index 0. Falls Runa Karten erhält, gibt das Spiel dies aus, indem für jede erhaltene Karte `Runa gets <Fähigkeitsname>(<Fähigkeitslevel>)` ausgegeben wird. Sollte Runa einen neuen Würfel erhalten, so quittiert das Spiel dies mit der Ausgabe `Runa upgrades her die to a <DN>`.

➤ Beispielinteraktion

```

1  [...]
2  Choose Runa's reward
3  1) new ability cards
4  2) next player dice
5  Enter number [1--2]:
6  > 1
7  Pick 1 card(s) as loot
8  1) Fire(2)
9  2) Slash(2)
10 Enter number [1--2]:
11 > 2
12 Runa gets Slash(2)
13 [...]
14 Pick 2 card(s) as loot
15 1) Lightning(1)
16 2) Slash(1)
17 3) Swing(1)
18 4) Ice(1)
19 Enter numbers [1--4] separated by comma:
20 > 2,3
21 [...]
```

➤ Beispielinteraktion

```

1  [...]
2  Choose Runa's reward
3  1) new ability cards
4  2) next player dice
5  Enter number [1--2]:
6  > 2
7  Runa upgrades her die to a d6
8  [...]
```

A.4.6.2 Runas Fähigkeiten-Upgrade Falls Runa gerade das Monster *Spider King* in Level 1 besiegt hat, so erhält Runa bessere Fähigkeitskarten. Diese sind abhängig von Ihrer Klasse. Das Spiel gibt in diesem Fall diejenigen Fähigkeitskarten für Level 2 aus, die Runa erhält. Das Format hierfür ist das gleiche wie in A.4.6.1. Im Folgenden sehen Sie ein Beispiel für die Ausgabe des Spiels beginnend mit der Todesnachricht des *Spider King* und eine Runa, die die Klasse *Warrior* gewählt hat.

➤ Beispielinteraktion

```

1  [...]
2  Spider King dies
3  Runa gets Thrust(2)
4  Runa gets Parry(2)
5  [...]
```

A.4.6.3 Heilen Sollte Runa sich nicht heilen können oder nicht heilen müssen, wird diese Phase übersprungen. Falls Runa sich heilen kann, wird mittels eines Auswahldialogs bestimmt, welche Karten sie abwerfen will. Im Falle, dass das Heilen möglich ist, beginnt das Spiel mit der Ausgabe Runa (<Aktueller Wert>/<Maximaler Wert> HP) can discard ability cards for healing (or none). Falls kein Wert eingegeben und nur die *Enter*-Taste gedrückt wird, so akzeptiert das Spiel dies, Runa wird nicht geheilt und keine Karte wird abgeworfen. Sollte Runa sich heilen, so quittiert das Spiel dies mit der Ausgabe Runa gains <HP> health. Beachten Sie hierbei, dass das Spiel genau ausgibt, wie viele Lebenspunkte Runa regeneriert hat.

➤ Beispielinteraktion

```

1  [...]
2  Runa (38/50 HP) can discard ability cards for healing (or none)
3  1) Focus(1)
4  2) Water(1)
5  Enter number [1--2]:
6  > 1
7  Runa gains 10 health
8  [...]
```

➤ Beispielinteraktion

```

8 | [...]
9 | Runa (8/50 HP) can discard ability cards for healing (or none)
10 | 1) Thrust(2)
11 | 2) Parry(2)
12 | 3) Slash(2)
13 | 4) Lightning(2)
14 | 5) Ice(2)
15 | Enter numbers [1--5] separated by comma:
16 | > 4,5
17 | Runa gains 20 health
18 | [...]
```

A.4.6.4 Ende des Spiels Für das Ende des Spiels gibt es zwei mögliche Ausgaben. Falls Runa gestorben ist, wird das Spiel mit der Meldung **Runa dies** beendet. Falls Runa das Boss-Monster des zweiten Levels besiegt hat, endet das Programm mit der Ausgabe, dass dieses Monster gestorben ist und zusätzlich mit der Zeile **Runa won!**. Ein mögliches Ende wird im Folgenden dargestellt.

➤ Beispielinteraktion

```

1 | [...]
2 | Runa uses Thrust(2)
3 | Enter dice roll [1--10]:
4 | > 10
5 | Mega Saurus takes 30 phy. damage
6 | Mega Saurus dies
7 | Runa won!
```

A.5 Beispielinteraktionen

Sie finden zwei vollständige Beispielinteraktionen des Spiels in ILIAS.