



Lesen Sie: Kapitel 7.

Das Spiel

Ihre Aufgabe ist es, ein "Adventure"-Spiel zu implementieren. Sie haben ein einfaches Projekt als Anfangspunkt (zuul-better). Erweitern Sie dies für Ihr Projekt.

Die Gruppe

Dieses Projekt ist ein Gruppenprojekt. Es muss in einer Gruppe bearbeitet werden. Alleine zu arbeiten ist keine ausreichende Bearbeitung dieses Projektes.

Gruppen bestehen aus vier Studenten. (Ausnahmen müssen vom Tutor genehmigt werden.)

Alle Gruppenmitglieder müssen einen signifikanten Beitrag zum Projekt leisten. (Mehr dazu unten.) Sie müssen Ihre Gruppe selber organisieren – dies ist eine der Herausforderungen dieses Projektes.

1 Code lesen

Lernen, Quelltext zu lesen, ist wichtig. Ihre erste Aufgabe ist es, sich den existierenden Quelltext anzugucken und zu verstehen, wie er funktioniert. Am Ende dieses Projektes werden Sie das meiste davon verstehen.

2 Kleine Erweiterungen

Als Aufwärmübung, nehmen Sie kleine Änderungen am Code vor. Zum Beispiel:

- ändern Sie die Beschreibung eines Raums.
- ändern Sie die Ausgänge – nehmen Sie einen Raum, der im Moment im Westen eines anderen Raums liegt, und tun Sie ihn in den Norden.
- fügen Sie einen Raum (oder zwei, oder drei Räume, ...) hinzu.

Durch diese und andere Übungen werden Sie den Code kennenlernen.

3 Spiel-Design

Zuerst sollten Sie entscheiden, was das Ziel Ihres Spieles ist. Es sollte vielleicht folgende Elemente enthalten: Die Spielerin muss einige Gegenstände finden, und diese in einen bestimmten Raum (oder zu einer Person?) bringen. Dann bekommt man einen anderen Gegenstand. Wenn man den in einen anderen Raum bringt, gewinnt man.

Zum Beispiel:

Sie sind an der Freien Universität Berlin. Sie müssen herausfinden, wo der Programmierkurs stattfindet. Dazu müssen Sie erst die Sekretärin finden, und fragen. Am Ende müssen Sie den richtigen Raum für die Klausur finden. Wenn Sie dort rechtzeitig ankommen, und Sie haben unterwegs irgendwo das Textbuch gefunden, und Sie waren auch im Tutorium, dann gewinnen Sie. Falls Sie mehr als fünfmal in der Studentenkneipe waren, halbiert sich Ihre Examensnote.

Oder:

Sie haben sich in einem Verließ verirrt. Sie treffen auf einen Zwerg. Wenn Sie etwas zu essen finden, das Sie dem Zwerg geben können, dann gibt der Zwerg Ihnen einen magischen Stein. Wenn Sie den magischen Stein in der großen Höhle benutzen, dann öffnet sich der Ausgang, und Sie haben gewonnen.

Es könnte alles mögliche sein. Überlegen Sie sich die Szene, die Sie benutzen wollen (ein Verließ, eine Stadt, ein Gebäude, ...) und entscheiden Sie, was Ihre "Räume" sind. Machen Sie es interessant, aber nicht zu kompliziert. (Ich würde vorschlagen, nicht mehr als 10 Räume.)

Erfinden Sie Objekte in den Szenen, vielleicht Menschen, Monster, usw. Entscheiden Sie, was für eine Aufgabe der Spieler lösen muss.

4 Implementierung

Entscheiden Sie, was für Klassen Sie brauchen, um das Spiel zu implementieren; dann implementieren und testen Sie.

5 Grundfunktionalität

Die Grundfunktionalität, die Sie implementieren müssen, ist:

- Ein Spiel hat mehrere "Räume". (Es müssen nicht Räume im wörtlichen Sinne sein.)
- Der Spieler kann durch die Räume gehen. (Dies ist schon implementiert.)
- Ein Raum kann Gegenstände enthalten. Jeder Raum kann eine beliebige Anzahl von Gegenständen halten. Einige Gegenstände können vom Spieler aufgehoben und mitgenommen werden, andere nicht.
- Die Spielerin kann einige Gegenstände mit sich tragen. Jeder Gegenstand hat ein Gewicht. Die Spielerin kann nur Gegenstände bis zu einem bestimmten Gesamtgewicht tragen.

- Der Spieler kann gewinnen. Es muss eine Situation geben die als Spielende erkannt wird, wo der Spieler informiert wird, dass er gewonnen hat.
- Implementieren Sie einen Befehl "back" der Sie in den Raum zurückbringt, in dem Sie zuletzt gewesen sind.
- Fügen Sie mindestens vier neue Befehle hinzu.

6 Weitere Funktionen

Wenn Sie wollen und Zeit haben, können Sie noch weitere Funktionen hinzufügen. Zum Beispiel:

- Fügen Sie Figuren in Ihr Spiel ein. Figuren sind Menschen oder Tiere oder Monster – alles was sich bewegt. Figuren sind auch in Räumen (so wie der Spieler und die Gegenstände). Figuren können sich selber bewegen.
- Erweitern Sie den Parser, so dass er drei-Wort-Befehle versteht. Zum Beispiel

```
give bread dwarf
```

und dem Zwerg etwas Brot zu geben (das Sie gerade mit sich tragen).

- Fügen Sie einen Transporterraum ein – jedes Mal, wenn Sie dort reingehen, werden Sie in einen zufälligen anderen Raum transportiert.
- Alles mögliche andere. Sie können selber weitere Funktionen erfinden. Diskutieren Sie Ihre Ideen mit Ihrem Tutor, so dass Sie Rat bekommen können, damit Sie sich nicht etwas allzu Schwieriges ausdenken. Einige weitere Ideen sind auch im Textbuch vorgeschlagen.

7 Abgabe und Bewertung

Sie müssen zwei Dinge einreichen: Ihr Projekt und einen Bericht.

Das Projekt muss vor Abgabeschluss als zip-Datei als Emailanhang an Ihren Tutor geschickt werden. Der Code muss professionell geschrieben sein (Kommentare und Formatierung!) und wird bewertet nach:

- Korrektheit
- angemessener Benutzung von Sprachkonstrukten
- Stil (Kommentare, Formatierung, Variablennamen, usw.)
- Design (Kohäsion, Kopplung, Wartbarkeit)
- Schwierigkeitsgrad

Der Bericht

Sie müssen ausserdem einen kurzen Bericht einreichen mit:

- dem Namen und einer kurzen Beschreibung Ihres Spiels;
- einer kurzen Beschreibung der weiteren Funktionen Ihres Spiels;
- bekannte Fehler im Programm;
- den Namen aller Gruppenmitglieder;
- den Beiträgen aller Gruppenmitglieder (wer hat was getan?).

Der Bericht sollte nicht länger als drei Seiten sein. Er muss als PDF-Datei per Email an Ihren Tutor geschickt werden.

Bewertung

Dieses Projekt ist ein Gruppenprojekt, aber die Bewertung ist individuell. Um das Projekt zu bestehen, müssen Sie aktiv und ausreichend an der Projektarbeit teilgenommen haben. Das bezieht sich auf die Implementierung und den Bericht. Autoren jeder Klasse müssen schriftlich im Klassenkommentar genannt werden. Jede(r) Student(in) muss mindestens eine nicht-triviale Klasse alleine implementieren (d.h., Sie müssen eine Klasse vorweisen können, von der Sie alleinige(r) Autor(in) sind).

Eine gute Implementierung der Grundfunktionalität ist ausreichend, um das Projekt zu bestehen.

Sie müssen ausserdem das Programm am Ende im Tutorium demonstrieren – dies ist auch Teil der formalen Anforderungen.

Abgabefrist

Dienstag, 31 August 2010, 23:59