Coding Lernspiel

Kurzfassung

Wir haben ein Spiel programmiert das das Learning by Doing Prinzip auf Java anwendet. Das Spiel wird dabei nicht wie normale Spiele mit Tastatur, Maus oder ähnlichem gesteuert, sondern mit Java-Befehlen. Dies führt spielerisch zum Programmieren hin ohne langweilig zu wirken. Dieses Spiel haben wir in Java mithilfe der Programmierumgebung Eclipse programmiert.

Inhaltsverzeichnis

1. Einleitung Seite 1

2. Vorgehensweise Seite 1

3. Ergebnisse Seite 1

4. Ergebnisdiskussion Seite 1

1.Einleitung

Direkt nachdem wir das letzten Wettbewerb haben wir beschlossen, das alte Projekt beiseitezulegen und nach neuen Ideen zu suchen. Dabei sind wir im Internet auf mehrere Programmierspiele gestoßen, die wir verbesserungswürdig fanden. Daraufhin haben wir uns zum Ziel gesetzt, ein eigenes zu programmieren. Und da wir in der 9+ Klasse im Informatikunterricht Java lernen, haben wir beschlossen das Spiel in Java, für Java zu schreiben. Da wir den Spielspaß hoch halten wollten, haben wir uns für den Spieletyp des RPG‘s (Role Player Game) entschieden. Das Ziel eines RPG‘s ist meistens das erfüllen einer Hauptstory, die sich wie ein roter Faden durch das ganze Spiel zieht. Viele RPG‘s basieren auf einer großen Welt, in der sich der Spieler meist uneingeschränkt bewegen kann und Aufgaben erledigt muss. So muss man beispielsweise Gegenstände sammeln, die Welt erkunden oder Monster bekämpfen. Dieser Spieletyp passt auch zu unserem Ziel, da Java – wie viele RPG‘s – eine Hauptstory (Grundlagen in Java) sowie viele Nebenstorys (Erweiterungen bzw. Packages) hat.

2. Vorgehensweise

Zunächst haben wir mit der Programmierung der Welt und des Spielers des RPG‘s begonnen, da der Rest nur Erweiterungen dieser Grundelemente ist. Die Welt besteht aus gleichgroßen, rechteckigen Feldern. Diese Felder sind in einem Dreidimensionalen Array gespeichert, das die Informationen über die einzelnen Felder enthält. Zunächst bestand die Welt nur aus Gras und Stein und einem Kreis, der den Spieler darstellen sollte. Nachdem wir die Bewegung des Spielers nach mehreren Versuchen mit den Pfeiltasten möglich gemacht hatten, führten wir ein, dass man manche Felder nicht betreten kann. Kurz danach haben wir das Inventar programmiert. Um dies auch direkt im Spiel auszuprobieren, haben wir als nächstes ein aufhebbares Objekt hinzugefügt, das in das Inventar aufgenommen wird. Dieses Objekt wird als extra Bild über ein anderes Feld gezeichnet, damit wir nicht jedes aufhebbare Objekt auf jeden Untergrund zeichnen müssen. Dieses extra Bild wird dann nach dem Aufheben entfernt. Mit einem zusätzlichen Stück Code haben wir dann dafür gesorgt dass das aufhebbare Objekt nach einer gegebenen Zeit wieder erscheint. Um nun die Handlungen des Spielers besser nachvollziehen zu können, haben wir nun einen in-game Log hinzugefügt. In diesem werden Tätigkeiten wie das Aufheben eines Objektes oder das Inventar (wenn vom Spieler angefordert) dokumentiert. Als nächstes haben wir eine weitere Welt hinzugefügt, damit wir mehrere verschiedene Orte leichter darstellen können. Daraufhin – um weitere Aspekte eines RPG‘s hinzuzufügen – haben wir Gegner einprogrammiert.

3.Ergebnisse

4. Ergebnisdiskussion

was es werden soll

inventar:

anzeige durch: inventar, aber auf 32\*32 , benutzung durch commands mit nummer (0, 1, 2 …)

player:

stats: health, level, (speed)

monster :

pathfinding

stats: health, level, (speed)

Quests:

werden durch levelaufstieg freigeschalten, geben erfahrung und geld,

Der Start sollte sein, dass man aus einem Wagen gefallen ist / dieser zerstört wurde und dass man dann die erste stadt aufsucht. Da am Anfang sollen dann die ganzen nicht-programmierten Steuerungselemente gezeigt werden.

Mit der Zeit sollte die Benutzung der anfänglichen Steuerungshilfe wegfallen, sodass der Spieler mit dem Code “weiterspielen” muss. An der Rechten Seite ist ein Log zu sehen.