

Projekt „404 – Name Not Found“

Programmieren 3

Team:

Theresa Artus
Torsten Bosecker
Marika Friedel
Peter Weigand

Spielprinzip

Das Spielprinzip ist das eines Point-and-Click Adventures (Zeigen und Klicken), mit der Maus wird ein Objekt in der Spielwelt angesteuert und mit einem Klick die Aktion ausgelöst.

Schwerpunkt des Spiels ist es im Spielverlauf verschiedene Gegenstands-, Kombinations-, Dialogrätsel zu lösen. Die Spielfigur wird mit der Maus, durch das Spielgeschehen, gesteuert und per Mausklick werden Anweisungen gegeben. Diese sind bspw. schau, benutze, nimm, öffne, schließe, gib, rede.

In der Spielwelt findet man unter anderem verschiedene Gegenstände, Items oder Charaktere, die man Verwenden oder mit welchen man zusammenarbeiten muss um im Spiel voran zu kommen. Ziel ist es am Ende alle Rätsel zu lösen und die Handlung damit erfolgreich zu beenden.

Aufbau des Programms:

Das Spiel kann aus einer oder mehrerer Teilwelten bestehen, repräsentiert durch die Klasse „World“. Diese Klasse verwaltet alle Elemente der Spielwelt, diese wären einmal der menschliche Spieler (Klasse „Player“), alle weiteren Personen im Spiel mit denen der Spieler reden und Gegenstände austauschen kann (Klasse „Character“), die Gegenstände die der Spieler aufnehmen und manchmal miteinander kombinieren kann (Klasse „Item“), und die Vordergrund-Objekte (Klasse „Object“) wie z.B. Zäune, Häuser, Bäume. Gegenstände können zum Teil mit Objekten benutzt werden, z.B. ein Schlüssel (Item) mit einer Tür (Object) um diese zu öffnen, oder ein Schlüssel (Item) und eine Truhe (Object) um sie zu öffnen und ihren Inhalt zu bekommen. Für sämtliche Objekte in der Spielwelt wurde eine Kollisionsabfrage implementiert.

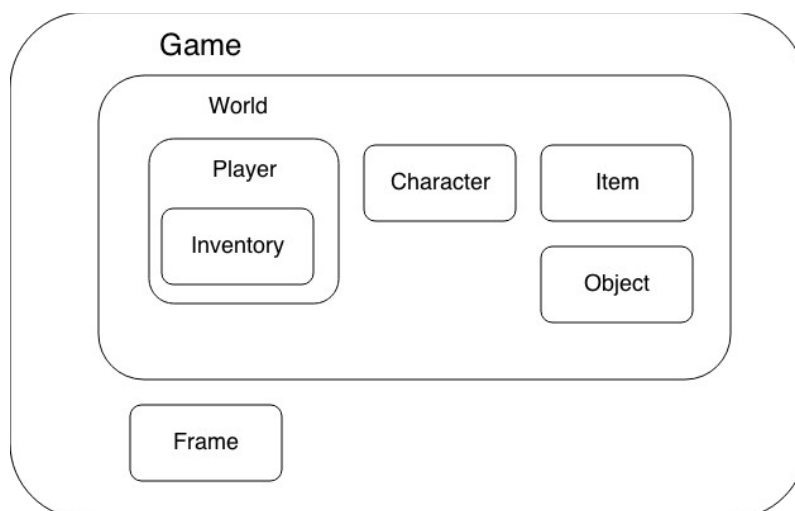
Die Klasse „World“ verwaltet die Elemente der Spielwelt in mehreren Listen, jeweils eine für Character, Items und Objects. Dies hat den Vorteil, dass z.B. beim Zeichnen der Spielwelt einfach nur sämtliche Listen durchgegangen werden müssen und für jedes Objekt darin die jeweilige Draw()-Methode aufgerufen werden muss (jedes Objekt im Spiel besitzt seine eigene Draw()_Methode und kann sich so selbst zeichnen). Ähnlich bei der Kollisionsabfrage: man muss einfach nur durch sämtliche Listen der Klasse „World“ iterieren und mit den Gettern der Objekte die Position und Größe der Objekte abfragen, durch einen Vergleich mit der Position des Spielers lässt sich so leicht eine Kollision feststellen.

Der Spieler besitzt ein Inventar (Klasse „Inventory“), in diesem werden alle Gegenstände abgelegt die der Spieler in der Spielwelt aufgehoben hat, oder von einer anderen Person erhalten hat.

Umschlossen wird die Klasse „World“ von der Klasse „Game“, in der das eigentliche Spiel dann gestartet und am Laufen gehalten wird. Im Moment dient die Klasse „Game“ nur zum Ausprobieren des Zusammenspiels der anderen Klassen, und um zu zeigen wie diese später eventuell einmal verwendet werden können.

Die Klasse „Frame“ wird später für das Darstellen der Spielwelt auf dem Bildschirm zuständig sein, wobei diese dann einfach nur die Draw()-Methoden sämtlicher Elemente der Spielwelt aufruft. Bis jetzt sind die eigentlichen Draw()-Methoden noch nicht implementiert.

Übersicht:



Vollständiges Klassendiagramm:

