

Java 3D: Das verrückte Labyrinth

Dokumentation zur Implementierung eines Brettspiels mit Java 3D.

Modul: B-201, Computergrafik
Modulleitung: Dr. Rafael Radkowski / Florian Klompmaker
Projektgruppe: Claudio Diaspero (221944)
Daniel Grosche (241691)
Joschka Zimdars (255702)
Semester: Sommersemester 2010
Stand: Montag, 28. Juni 2010

Inhaltsverzeichnis

Inhaltsverzeichnis	i
1 Beschreibung des Spiels	2
1.1 Grundidee	2
1.2 Ziel	2
1.3 Handlung	2
Spielzug Phase 1: Einschieben des freien Gangfeldes	2
Spielzug Phase 2: Setzen der Spielfigur	2
2 Aufbau des Modells	3
Spielwelt	3
Transformationen	3
Gangfelder und Randfelder	3
Objekte (Spielfiguren und Schätze)	4
3 Kamera und Kameraführung	4
3.1 Kameraführung (Intro)	4
3.2 Kamera (während des Spiels)	4
4 Beleuchtung, Materialien und Texturen	4
Beleuchtung (im Speziellen)	4
Materialien	5
Transparenz	5
Texturen	5
5 Animation und Interaktion	5
5.1 Animation	5
Einschieben des Gangfeldes	5
Verschieben der Spielfigur	6
Rotation der Schätze	6
Fokus aktuelle Spielfigur und aktueller Schatz	6
5.2 Interaktion	6
6 Aufbau des Szenegraphen	6

1 Beschreibung des Spiels

Das Softwareprojekt (praktischer Prüfungsteil) im Modul Computergrafik umfasste die Konzeption und Implementierung eines beliebigen Brettspiels mit Hilfe der Java 3D Klassenbibliothek. Die Java 3D Klassenbibliothek bietet Entwicklern die Möglichkeit Software mit virtuellen, dreidimensionalen Szenen zu ergänzen. Als Brettspiel bezeichnet wird ein Gesellschaftsspiel für zwei oder mehr Mitspieler, dessen zentrales Element ein Spielbrett ist. Auf diesem agieren Spieler mit Figuren, Steinen und anderen Materialien. In folgender Dokumentation ist die Implementierung des Brettspiels *Das verrückte Labyrinth* von Ravensburger Spiele beschrieben.

1.1 Grundidee

Das verrückte Labyrinth ist eine Mischung aus Brettspiel und Legespiel für bis zu vier Spieler, zu dessen Bewältigung räumliches Denken und Konzentration erforderlich ist. Das Spielbrett ist mit 49 Feldern (Gangfelder) bestückt. Zusätzlich gibt es ein überschüssiges (freies) Gangfeld, das im Spielzug seinen Einsatz findet. Es gibt drei unterschiedliche Typen an Gangfeldern: Kurve, Gerade und T-Stück. Begonnen bei den vier unbeweglichen Eckfeldern folgt alternierend eine bewegliche auf eine unbewegliche Reihe / Spalte an Gangfeldern.

1.2 Ziel

Ziel des Spiels ist es als erster Spieler eine Reihe unterschiedlicher Zwischenziele / Objekte (z.B. Buch, Ring oder Schatzkiste) zu erreichen und anschließend zum Ausgangspunkt (Startpunkt) zurückzukehren.

1.3 Handlung

Die Spielhandlung bzw. Spielrunden der einzelnen Spieler sind in ihrem Ablauf über den Gesamtspielverlauf gleichbleibend und wiederkehrend. In einer Spielrunde wird nacheinander von jedem Spieler ein vollständiger, in zwei Phasen unterteilter Spielzug durchlaufen.

Spielzug Phase 1: Einschieben des freien Gangfeldes

In eine beliebige bewegliche Reihen / Spalten schiebt der am Zug befindliche Spieler das aktuell freie Gangfeld (um eine Position) ein. Die Ausrichtung des eingeschobenen Feldes steht dem Spieler frei. Das auf der gegenüberliegenden Seite heraus geschobene Gangfeld wird zum neuen freien Gangfeld des nächsten Spielers. Beim Einschieben des Gangfeldes verfolgt jeder Spieler die Absicht eine für ihn möglichst optimale Wegstellung auf dem Spielbrett zu erreichen (siehe Spielzug Phase 2). Hierbei gilt das übergeordnete Ziel das nächste Schatz (Zwischenziel) möglichst schnell und direkt erreichen zu können. Wechsel zu Spielzug Phase 2.

Spielzug Phase 2: Setzen der Spielfigur

Abgeschlossen wird der Spielzug durch das Verschieben der Spielfigur. Hierbei kann der am Zug befindliche Spieler jede, durch die Gangfelder als Weg symbolisierte (siehe Abschnitt 4, Transparenz), erreichbare Stelle des Spielbretts als neuen Ort für seine Spielfigur wählen (Ziehen der Spielfigur). Das Ziehen der Spielfigur ist optional. Erreicht der Spieler durch das Setzen der Spielfigur den zu erreichenden Schatz gilt dieser als „erledigt“. In diesem Fall wird dem Spieler ein zufällig gewählter

Schatz als nächstes Zwischenziel zugeteilt. Der Spielzug des Spielers endet – das Spiel wechselt zurück in die Phase 1 des nächsten Spielers.

2 Aufbau des Modells

Bei der Implementierung des Spiels entschieden wir uns für eine dem Brettspiel analoge Umsetzung. Der Nutzer der Software sollte das Spiel *Das verrückte Labyrinth* direkt wiedererkennen und ohne zusätzliche Anleitung / Erklärung, vorausgesetzt der Nutzer ist mit dem Brettspiel vertraut, spielen können. Das Modell (die Software) ist nach dem Model View Controller (MVC) Entwurfsmuster umgesetzt worden. Dadurch konnte eine strikte und klare Trennung der Schichten Daten (-haltung), Präsentation (in 2D und 3D) und (Spiel-) Steuerung eingehalten werden.

Spielwelt

Als Spielwelt ist der dreidimensionale Raum (*Universum*) zu sehen, der Spielbrett und Objekte enthält. Das Spielbrett besitzt eine 9 x 9 Matrix¹ in der Gang- und Randfelder „platziert“ sind. Die in der Spielwelt enthaltenen Elemente (Gangfelder, Spielfiguren und Schätze) setzen sich entweder aus einfachen Grafikprimitiven oder aus einer Kombination mit *UnBBoolean* manipulierter Primitiven zusammen.

Transformationen

Transformationen (Translation und Rotation) wurden verwendet, um die graphischen Objekte an gewünschter Stelle im Universum zu positionieren. Die Gangfelder, Spielfiguren und Schätze wurden vom Mittelpunkt der Szene aus transformiert. Für die interne Zusammensetzung der Objekte wurden lokale Transformationen verwendet. Ausgangspunkt der lokalen Transformationen ist jeweils der Mittelpunkt des jeweiligen Objektes. Für die Spielfigur wurde beispielsweise ein Kegel erstellt und so transformiert (Translation), dass er auf dem Weg (der Grundfläche) des Gangfeldes steht. Anschließend wurde eine Kugel auf die Spitze des Kegels verschoben. Rotationen sind im Folgenden am Beispiel des Aufbaus eines Gangfeldes erläutert.

Gangfelder und Randfelder

Die Grundfläche und Höhe der Wände gilt für jeden Typ des Gangfeldes (Kurve, Grade, T-Stück) gleichermaßen. Boden und Wände sind jeweils durch die Primitive Quader `Box()` realisiert. Durch Transformationen (Rotation und Verschiebung) werden die Wände an, für jeden Typ passender, Stelle platziert. Die Position der Wände ihrerseits bestimmt den Typ des Gangfeldes. Jedes Gangfeld wird in einer Basisorientierung erzeugt; z.B. für den Typ Kurve von links nach oben. Entsprechend der Basisorientierung sind die Wände auf dem Gangfeld platziert. Eine im Szenegraphen dem Gangfeld übergeordnete TransformGroup erlaubt die Rotierung des Gangfeldes in 90° Schritten (siehe auch Abschnitt 5.2). Die Grundfläche des Gangfeldes ist mit einer transparenten `Box()` versehen, die zur Hervorhebung des möglichen Spielzuges der Figur genutzt wird (siehe Abschnitt 4).

¹ 7 x 7 Gangfelder auf dem Spielbrett zzgl. einreihiges Randfelder um gesamtes Spielbrett (Randfelder auswählbar zur Auswahl des Einschubs).

Abweichend werden Randfelder zwar ebenfalls aus Gangfeldern erzeugt, so enthalten diese jedoch keine Wände. Eine Änderung des transparenten Bodens wird zur Markierung der vom Nutzer gewählten Einschubposition genutzt (siehe Abschnitt 4).

Objekte (Spielfiguren und Schätze)

Spielfiguren setzen sich aus den zwei Grafikprimitiven Kegel `Cone()` und Kugel `Sphere()` und deren passende Transformation zueinander zusammen. Zudem sind die Spielfiguren mit einem Material belegt (siehe Abschnitt 4) und in einer, den Spieler repräsentierenden Farbe (rot, gelb, blau oder grün) gefärbt.

Schätze wurden auf unterschiedliche Weisen modelliert. So ist beispielsweise die Schatzkiste aus den zwei Primitiven Quader `Box()` und Zylinder `Cylinder()` zusammengefügt, und mit einem passenden Material versehen. Der Ring hingegen setzt sich aus mehreren, mit *UnBBoolean* manipulierten Primitiven `BooleanModeller()` zusammen (siehe detaillierter Szenegraph im Anhang). Ergänzend wurde für das komplexe Grafikobjekt Krone eine fertiges *OBJ Modell* eingebunden `loadObject()`.

3 Kamera und Kameraführung

3.1 Kameraführung (Intro)

Das Spiel beginnt nach dem Laden mit einer Kamerafahrt. Aus deutlicher Entfernung „nähert sich der Betrachter“ dem Spielbrett. Bei Erreichen einer nur noch leicht distanzierten Ansicht wird das Bild um 360° um das Spielbrett geführt. Das gesamte Spielbrett bleibt während dieser Kameraführung weiterhin sichtbar. Durch die Kameraführung erhalten alle Nutzer einen Einblick in die Anordnung der Gangfelder, Spielfiguren und der auf dem Spielbrett verteilten Schätze. Nach dem Beenden der Kamerafahrt wechselt der Kameramodus in die Kameraansicht während des Spiels.

3.2 Kamera (während des Spiels)

Gestartet wird der Spielablauf in einer Kameraeinstellung in der Vogelperspektive (schräge, leicht distanzierte Aufsicht auf das Spielbrett). So sind auf einen Blick das gesamte Spielbrett, alle Spielfiguren und die verschiedenen Schätze sichtbar. Per Klick und Bewegung mit der Maus `MouseRotate()` kann der Nutzer die Spielwelt frei rotieren und aus jedem gewünschten Blickwinkel betrachten. Die Betätigung des Mausekads `MouseWheelZoom()` erlaubt dem Nutzer freie Kontrolle über das stufenlose Zoomen in die Spielwelt hinein bzw. aus der Spielwelt heraus.

4 Beleuchtung, Materialien und Texturen

Die gewählte Mischung (Grundbeleuchtung) aus ambienter `AmbientLight()` und direktonaler Beleuchtung `DirectionalLight()` ist so getroffen, dass alle in der Spielwelt enthaltenen Elemente (Spielbrett, Spielfiguren und Schätze), Texturen, Materialien und Transparenzen gut sichtbar sind.

Beleuchtung (im Speziellen)

Beleuchtungsakzente in Form von Lichtkegeln bzw. Spots `SpotLight()` helfen dem Nutzer Spielfigur und Schatz schneller aufzufinden. Zwei Spots sind dabei jeweils zum Beginn jeder Spielrunde auf den

aktuellen Spieler und dessen aktuell zu erreichenden Schatz gerichtet und eingeschaltet. Eine detaillierte Beschreibung des Einsatzes der Lichtkegel findet sich im Abschnitt 5.1.

Materialien

Materialien wurden für nahezu alle graphischen Objekte der Spielwelt verwendet. Die Materialien beeinflussen ihrerseits das Erscheinungsbild (z.B. Reflexion) der Objektoberfläche im Licht der zuvor beschriebenen Grundbeleuchtung. Spielfiguren wurden mit dem Material Kunststoff überzogen `setMaterial()`. Sie erscheinen somit plastisch und heben sich von den mit Texturen belegten Gangfeldern deutlicher ab. Die Schatzkiste, als Beispiel aus der Reihe der Schätze, ist mit dem Material Kupfer versehen.

Transparenz

Transparenz `TransparencyAttributes()` wurde unter anderem im Bereich der Randfelder eingesetzt. Diese, ansonsten nicht sichtbaren Felder, werden bei Auswahl durch den Nutzer als Einschub für das freie Gangfeld hervorgehoben. In Spielzug Phase 2 „Setzen der Spielfigur“ werden alle zum Versetzen der Spielfigur möglichen Gangfelder entsprechend der Farbe des aktuellen Spielers eingefärbt.

Texturen

Texturen wurden eingesetzt, um ausgewählten Spielelementen ein realistischeres Erscheinungsbild zu verleihen. Der Einsatz von Texturen wurde auf die Gangfeld Elemente beschränkt. Die Grundfläche (der Boden) des Gangfelds ist einer Textur aus Stein belegt `setTexture()`. Die Wände sind mit einer Textur einer Backsteinmauer versehen. Um die Spielwelt lebhafter und weniger monoton erscheinen zu lassen, ist die Spielwelt mit einem Umgebungsbild belegt. Die Rand- und Unterseite des Spielbretts spiegelt das Umgebungsbild wieder (*Environment Mapping*).

5 Animation und Interaktion

5.1 Animation

Durch den Einsatz von Animationen werden Objekte der Szene mit der Zeit verändert. Animationen wurden eingesetzt, um Spielhandlungen und das Erscheinungsbild von Objekten realistischer wirken zu lassen.

Einschieben des Gangfeldes

Eine der Kernhandlungen des Spielablaufs ist das Einschieben des freien Gangfeldes in die gewünschte Reihe / Spalte. Zur realistischen Abbildung dieses Schritts wurde eine Pfadanimation genutzt. Beginnend am Ausgangspunkt des Einschiebens wird die Reihe / Spalte an Gangfeldern um eine Position in das Spielfeld hinein verschoben. Das dem Ausgangspunkt gegenüberliegend heraus geschobene Gangfeld wird vom Spielbrett entfernt (verblasst durch Zunahme der Transparenz). Es wird zum neuen freien Gangfeld des nächsten Spielzugs. Eine eventuell mit heraus geschobene Spielfigur wird auf dem eingeschobenen Gangfeld platziert (erscheint durch Abnahme der Transparenz).

Verschieben der Spielfigur

Das Verschieben der Spielfigur ist ebenfalls ein animierter Vorgang. Die Figur wird hierbei von ihrer Ausgangsposition zu der vom aktuellen Spieler gewählten Zielposition verschoben. Die Verschiebung erfolgt als kontinuierliche Bewegung der Spielfigur entlang eines zur Spielzeit errechneten Pfades.

Rotation der Schätze

Das eher triste Erscheinungsbild der unbewegten, statischen Schätze wurde durch den Einsatz von Animationen aufgewertet. Die rotierenden Schätze reflektieren Licht in Abhängigkeit des Einfallswinkels. Die entstehenden Glanzlichter lassen die mit Materialien belegten Oberflächen der Schätze wirklichkeitsnaher aussehen.

Fokus aktuelle Spielfigur und aktueller Schatz

Um dem aktuellen Spieler zu Beginn seiner Spielrunde das Auffinden der Spielfigur und des als nächstes zu erreichenden Schatzes auf dem Spielbrett zu erleichtern, werden Spotlights eingesetzt. Die Animation durchläuft die nachfolgend beschriebenen Schritte. Dimmen (verdunkeln) der Grundbeleuchtung. Weiches Einschalten der Spotlights über der Spielfigur und des aktuellen Schatzes. Weiches Ausschalten der Spotlights. Dimmen (aufhellen) der Grundbeleuchtung.

5.2 Interaktion

Vom Nutzer ausgeführte Interaktionen haben Einfluss auf die visualisierte Szene. Interaktionen bzw. Nutzerhandlungen bestimmen und steuern den Spielablauf, lösen Animationen aus und lassen die erschaffene Computerszenarie zum Spiel werden. Dem Nutzer wurde die Möglichkeit gegeben das Spiel mit der Maus zu „steuern“, d.h. Spielaktionen mit der Maus zu initiieren. Hierzu liefert eine Kollisionserkennung Aufschluss über das vom Nutzer angeklickte Objekt. Eine situationsabhängige Aktion (z.B. Animation) kann daraufhin ausgeführt werden – z.B. das Verschieben der Spielfigur zur angeklickten Zielposition.

6 Aufbau des Szenegraphen

Zur Modellierung dreidimensionaler Szenen werden die in der Szene enthaltenen Objekte hierarchisch in einer Baumstruktur angeordnet (siehe Abbildung 1, nächste Seite). Ausgehend vom Wurzelknoten *scene* beschreibt der Szenegraph die virtuelle Spielwelt. Auf oberster Hierarchieebene sind die Verhaltenssensoren für die Spielinteraktion (Mausaktionen) und die Basis-Transformationsgruppe der Spielszene *sceneTG* angeordnet. Dem inneren Knoten *sceneTG* sind die Grundbeleuchtung *AmbientLight* und *DirectionalLight* sowie die Transformgruppe *rootTG* angefügt. Diese trägt ihrerseits die Kindelemente des Hintergrundes *Background* und die, das *Spielbrett* (im Sinne der Zusammenfassung als Branchgroup) beinhaltende, Transformgroup *tmpTG*.

Hinweis: Zwei ausführliche Abbildungen des Szenegraphen finden sich im Anhang an diese Dokumentation und wurden aus Platzgründen hier nicht aufgenommen.

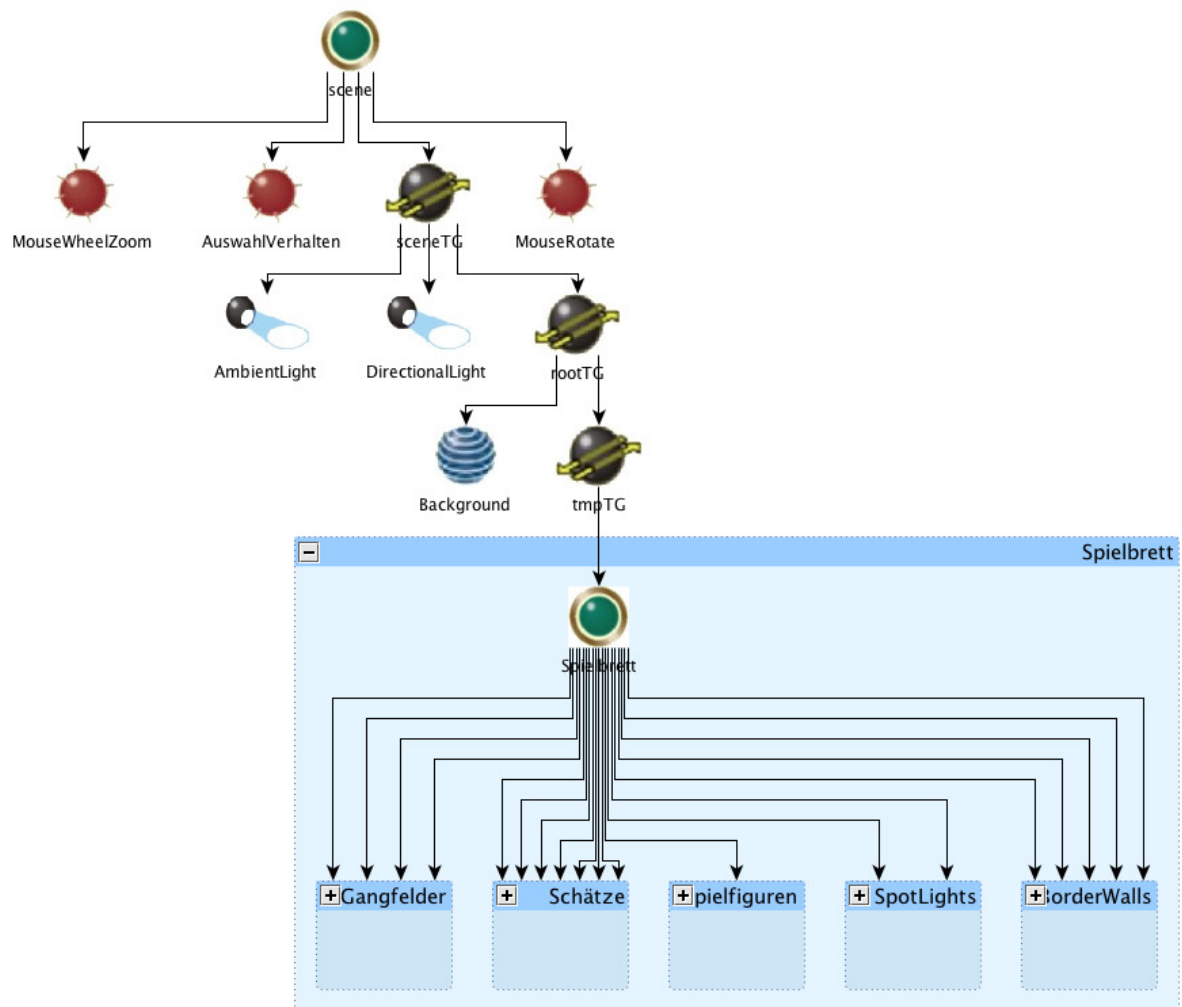


Abbildung 1 Szenegraph des Spiels *Das verrückte Labyrinth*. Der Szenegraph zeigt lediglich die obere Hierarchie im Detail. Die Hierarchiestufen unterhalb des Elternelements *tmpTG*, d.h. alle Kindelemente der Branchgroup *Spielbrett*, sind den anliegenden Szenegraphen zu entnehmen. Die Symbole des Szenegraphen sind entsprechend der Vorlesungsfolien „Kapitel 1 / Szenegraph“ übernommen und hier nicht erläutert.