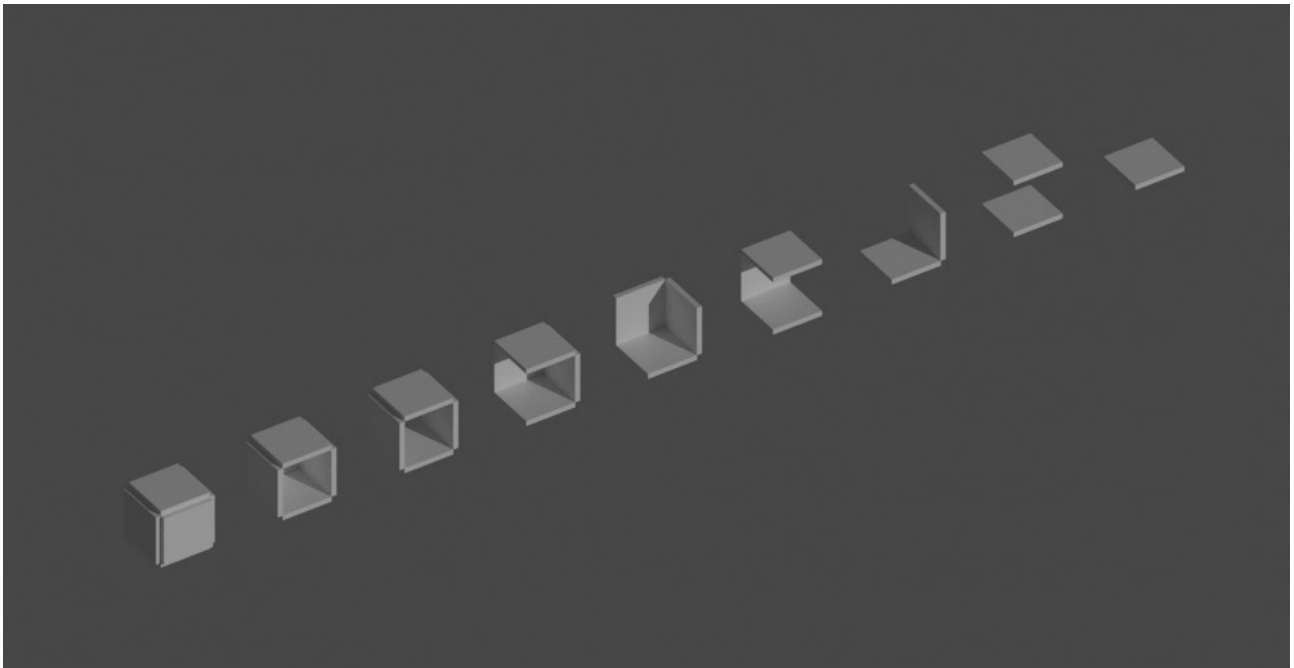


Designdokument

PRIMA

Prototyping interactive media-applications and games

Discombobulated Space Station – 3D FUDGE Game



Calvin Dell'Oro
Sommerbergstraße 25
78120 Furtwangen
calvin.delloro@hs-furtwangen.de
Mat.-Nr. 263179

Betreuer: Jirka Dell'Oro-Friedl
23.7.2021

Idee

Story

Das 3D-Videospiel “Discombobulated Space Station” handelt von dem Astronauten Felix, der sich versehentlich über Nacht in einer niedrig erdumlaufenden Raumstation eingeschlossen hat.

Aufgrund der vertrackten Komplexität der Raumstation kann Felix die steilen Winkel nicht ohne Manipulation der Umgebung überwinden, dennoch versucht er schnellstmöglich die goldene Kabine zu erreichen, und dabei nicht aus der Raumstation in die Leere zu fallen.

Doch zu seinem Glück findet er einen Controller, der es ihm ermöglicht, die Raumstation um sich selbst zu drehen, oder die Gravitation zu ändern – er konnte den Unterschied aufgrund höchster Dringlichkeit nicht erkennen.

Um sich in dem Labyrinth aus unzähligen Gängen zurechtzufinden, versucht er Gegenstände zu verteilen, und somit den Weg zu markieren. Da alle Objekte in der Raumstation fest montiert sind, müssen die von seiner Großmutter Kristy Hopkins nach originalem Rezept (<https://de.tagrezepte.com/dinkelbeeren-kochen---einfach-R2877061>) frisch gekochten Dinkelbeeren herreichen.

Seine Lage verschlimmert sich am nächsten eingeschlossenen Tag, als er plötzlich mit gebrochenen Beinen aufwacht...

GameDesign und Pädagogik

Das Spiel wird in der ersten Person gespielt und gesteuert. Der Charakter kann sich auf dem planaren Boden fortbewegen, jedoch nicht klettern. Sprünge sind nur begrenzt möglich, da die Grundmechanik auf der Rotation der Raumstation um den Charakter beruhen soll, und dienen lediglich der Umgehung von fehlplatzierten Kanten.

Das Spiel ist ein Logik-Rätsel mit verstärktem Blick auf räumliches Orientierungsvermögen, Mustererkennung und die Relativitätstheorie.

Das Spiel bietet eine “volle” 3D-Labyrinth-Erfahrung, inklusive Abzweigungen, Überlappungen und Schleifen in allen drei kartesischen räumlichen Dimensionen. Darüber hinaus unterstützt “Discombobulated Space Station” (im folgenden “DSS” genannt) durch seine Rotationsmechanik freie Fälle resp. Flugbahnen zwischen beliebig weit voneinander entfernten Gängen.

Besonders der letzte Aspekt kann Spieler mit einer außerordentlich hohen Komplexität trotz simpler Bedienelemente auf kleinstem Raum, sowie der Herausforderung, sich völlig von unserer menschlichen Orientierung zu trennen, bereichern.

Das Legen von steinähnlichen Objekten dient weniger als Unterstützung der Orientierung; vielmehr ist es eine komedianische Repräsentation der Sinnfreiheit des losen Platzierens von Objekten in einem vollkommen inkonsistentem System, diene es der Orientierung.

Gefahr droht dem Spieler, wenn er eine Öffnung übersieht und falsch rotiert, oder sogar einfach hinausläuft. Die Labyrinth-Öffnungen werden optimal so gelegt, dass der Spieler bei Austritt in die Leere, nicht auf die Außenseite der Raumstation fällt. Da der Spieler allerdings nur rotieren kann, wenn er sich in, und nicht auf einem Element befindet, hätte dieser Außenbereich auch GameDesign-technische Vorteile und neue Möglichkeiten (Mehr in Ausblick).

Ausblick

Ursprünglich geplant war ein Station-Editor, der es dem Spieler ermöglicht, eine Raumstation mit den Elementen seiner Wahl zu erbauen, oder eine Raumstation auszubauen.

Die Geräuschkulisse und Soundeffekte eines solchen Spiels könnten durch “wooooosh”-Effekte bei Passieren eines Ganges, die bei freiem Fall resp. Flug zu einer akkustischen Vermittlung des Gefühls von Geschwindigkeit führen, und der Implentierung weiterer Umgebungsgeräusche bereichert werden.

Auch eine Idee sind runde oder andere Formen, statt lediglich quadratischer. Bei einem Pentagondodekaeder wären beispielsweise zwölf Abzweigungen pro Element, sowie zwölf verschiedener Rotationsstufen, anstatt sechs. Ebenfalls spannend wäre die Kombination mehrerer rotationsfähiger Systeme, die sich durch offene Tunnel als Schnittstellen verbinden lassen. Dies kann geschehen, indem sie schlichtweg “dorthin rotiert” werden. Durch Kombination von Fortbewegung im Labyrinth und Rotation um ein bestimmtes Element wird die absolute Position verschoben.

Im Multiplayer-Einsatz hätte dies besonders große Wirkung:

- Spieler könnten sich in einer Spielwelt mit ihren Raumstationen treffen
- Ungewollte Multiplayer-Eindringlinge, die auf der Außenseite der Raumstation laufen, können “abgeschüttelt” werden
- Team-Work ermöglicht schnellere Fortbewegung durch Positionierung zweier oder mehrerer Multiplayer in einer Raumstation, die sich Koordinieren und die Raumstation drehen
- Der Station-Editor ermöglicht jedem Spieler seine eigenen Individuellen Raumstationen zu bauen, mit Freiheitsgraden ähnlich zu “Minecraft”
- Budget-Bedingungen für das Bauen von Raumstation-Elementen kann in diesem Fall einen vollkommen generischen und rein strategischen Ressourcenkrieg zwischen den Spielern führen
- Serverbetreiber finden allgemein großen Spaß darin, ein simples Spielprinzip mit Serverregeln und anderen Spielmodi zu erweitern

Umsetzung

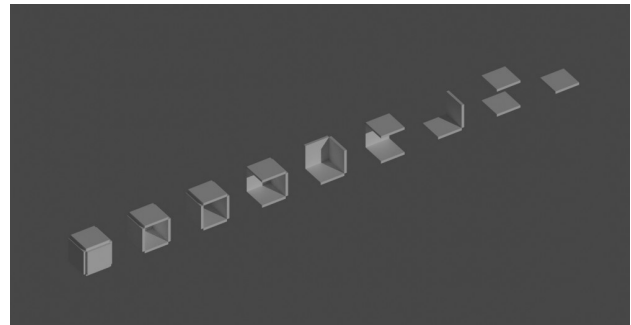
Allgemein

Die Einheitseins ist im DSS als die Seitenlänge der Element-Würfel definiert.

Labyrinth-Elemente

Für die Umsetzung aller Möglichkeiten, die ein 3D-Labyrinth bietet, wurden 9 Grundbausteine entwickelt, die nach Rotation und Translation alle beliebigen Abzweigungsarten darstellen:

0. DeadEnd (Sackgasse)
 1. Goal (Subtype)
1. Tunnel
2. Turn
3. L (Abgänge ähneln in Grundstellung einem "L")
4. T (Alle Gänge bilden ein "T")
5. Tripod (Abg. Ähn. in Grundst. einem Dreibein)
6. X (Alle G. bild. ein Kreuz)
7. OneBlocker (nur 1 Richtung blockiert)
8. Void (Alle Richtungen erlaubt, nichts beschränkt)



9 Elements of "Discombobulated Space Station"

Von Links unten nach Rechts oben aufsteigende Indizes.

In Fudge werden diese Elemente einmalig per Hand zusammengebaut und die GraphID aufbewahrt.

Blender .dae -> DSS Importer

Um den Aufbau eines Labyrinth-Levels für einen Entwickler einfacher zu gestalten und durch die üblichen Modelling-Tools zu unterstützen, wurde ein Importer, zu finden in der Klasse "ElementLoader" geschrieben.

Dieser liest aus einer .dae Datei alle Transformationen auf oberster Ebene aus, prüft sie auf Namensgleichheit mit den 9 Elementen, wählt ein Element aus den Fudge-internal-ressources und platziert sowie rotiert es. Da der Blender-Workflow standardmäßig Würfel mit der Seitenlänge 2 ausstattet, CSS allerdings mit 1, werden alle Positionen beim Import halbiert.

Maze Rotator

Befindet sich der Spieler in einem Element, und fordert eine Rotation an, so werden alle aus Blender importierten Elemente um dieses Element gedreht.

Ausblick - Umsetzung

Um den Blender importer weiter zu vereinfachen, könnte ein einzelnes Objekt importiert werden, welches als Mesh alle benötigten Informationen enthält.

Hierzu müssten keine Namens-Kodizes mehr eingehalten werden, ebenfalls spielten Rotationen keine Rolle mehr.

Der Importer würde zu jedem Vertex im Mesh die Anzahl und Richtungen der über Kanten verbundenen umliegenden Vertices auswerten, und daraus das Passende Element, die passende Rotation und Translation erreichen.