

Stack Till You Drop - Dokumentation

Alois Ott (01121383), Jakob Kogler (01127020)

19. Juni 2017

”Stack Till You Drop” ist ein Puzzle-Plattformer. Der Spieler muss in verschiedenen Level mit Hilfe von Kisten Hindernisse überwinden und zu einem Portal gelangen.

1 Spielprinzip

Der Spieler befindet sich auf einer Plattform. Von dieser aus muss ein Parkour überwunden werden. Als Hilfsmaterial liegen Kisten verstreut, die er (jeweils einzeln) nehmen und beliebig versetzen kann. So ist dieser in der Lage Treppen oder dergleichen zu formen, die ihm helfen Wände zu überwinden. Die Anzahl der Kisten ist knapp bemessen, weshalb auf die Bauweise geachtet werden muss. Sollte der Spieler von der Plattform abkommen, fällt er ins Void und das Level wird neu gestartet. Die Kisten können ebenfalls (durch schlechte Platzierung) von der Plattform fallen, wodurch sich die Überwindung des Parkours deutlich erschwert.

2 Auflösung und Fullscreen

Mittels Kommandozeilenargumente kann man die Auflösung des Fensters einstellen. Ebenfalls kann damit ein Fullscreen-Modus aktiviert werden und die Frequenz eingestellt werden. Der Aufruf des Programmes erfolgt mit *StackTillYouDrop.exe 1920 1080 1 60*.

3 Steuerung

Der Spieler bewegt seine Position mit den W,A,S,D Tasten und der Leertaste. Die Blickrichtung steuert man mit der Maus. Kisten können mit der linken Maustaste aufgehoben und losgelassen werden.

Taste	Effekt
W,A,S,D	Bewegen der Figur
Leertaste	Springen
Maus	Blickrichtung
Linke Maustaste	Aufheben und Loslassen von Blöcken
ESC	Beenden

Wir greifen auf die Tastatureingaben, Mausbewegungen und Mausklicks via drei Callback-Funktionen zu, die wir der GLFW-Library übergeben.

Zurzeit ist es möglich in der Luft zu springen, wenn man länger auf die Leertaste drückt. Dies ermöglicht, dass man auch ohne Kisten über die Plattform gelangen kann. Dies wird bis zur endgültigen Abgabe noch geändert.

4 Kamera

Die Sicht ist aus der Egoperspektive. Mit F5 lässt sich (für Testzwecke) auf eine dritte Person Perspektive wechseln. Der Spieler steuert die Blickrichtung. Die Drehung nach oben und unten ist beschränkt. Die für Links und Rechts sind durchgehend.

5 Objekte

Die komplexeren verwendeten Objekte sind von <https://www.turbosquid.com>. Die Objekte stehen unter der Royalty Free License. Die einfacheren Objekte (Mauern, Würfel, Ziel, Plattform) wurden mit Blender erzeugt.

Alle Objekte werden aus .obj und .mtl Dateien eingelesen. Dazu wird Assimp verwendet.

Einige der Objekte sind fest (nicht beweglich). Dazu gehören die Plattform, die Mauer, und die Blumen und die Lichtquelle. Alle anderen Objekte sind beweglich. Das sind die Kisten, die Person, und der Spieler. Alle Bewegungen und Kollisionen der Objekte werden mithilfe der Bullet-Physik-Library simuliert.

Der Spieler (Kamera) kann alle beweglichen Objekte anstoßen, aufheben und wieder loslassen. Alle Kollisionen werden erkannt und die Bewegungen gestoppt. Ebenfalls simuliert die Bullet-Library die Gravitation. Für aufgehobene Objekte wird die Gravitation temporär ausgeschaltet.

6 Texturen

Jedes verwendete Objekt mit Ausnahme der Lichtquelle besitzt eine Textur. Auch die komplexeren Objekte wie die Blumen oder der Nanosuite. Die Texturen wurden teils mit CrazyBump generiert und mit Blender aufgezogen. Für die meisten Objekte wurden neben der Diffuse-Texture auch eine Normal-Textur und eine Height-Textur eingelesen/erzeugt.

7 Licht und Materialien

Es gibt eine einzige globale Lichtquelle, die Sonne. Die Helligkeit eines Objekts ist über drei Lichtberechnungen gegeben. Ambient Lighting bestimmt die Grundhelligkeit eines Objekts. Selbst die von der Sonne abgewandten Seiten sollen gering sichtbar sein. Diffuse Lighting erhellt Flächen stärker, die direkt zur punktwise Lichtquelle weisen. Specular Lighting hebt schließlich noch Reflexionen der Sonne an der Oberfläche hervor. Über die Materialeigenschaften lässt sich die Farbe der Objekte (nur bei nichttexturierten Objekten) unter den drei Lichteinflüssen festlegen. Ebenfalls kann man auch die Größe des Reflexionskreises festlegen.

8 Verwendete Libraries

- GLFW (Version 3.2.1, <http://www.glfw.org>): Erzeugen des Fensters, Lesen von Input
- GLEW (Version 2.0.0, <http://glew.sourceforge.net>): OpenGL Erweiterungen
- GLM (Version 0.9.8.4, <http://glm.g-truc.net>): Vector- und Matrizenberechnungen
- Assimp (Version 3.3.1, <http://assimp.sourceforge.net>): Laden von 3D Objekten
- SOIL (Stand 7. Juli 2008, <http://www.lonesock.net/soil.html>): Laden von Texturen
- Bullet (Version 2.86, <http://bulletphysics.org>): Physikberechnungen (z.B. Kollisionen)
- irrKlang (Version 1.5, <http://www.ambiera.com/irrclang>): Audio

9 Effekte

9.1 Shadowmaps (mit PCF)

Schatten wurden mit Shadowmaps erzeugt. Artefakte wie Shadow acne, Peter Panning wurden beseitigt. Und für das Aliasing wurde PCF mit mehreren Samples verwendet.

9.2 Normal Mapping

Kann mit F6 ein und ausgeschaltet werden. Der Effekt ist besonders gut bei den Boden und den Kisten erkennbar. Aber auch für die anderen Objekte wird Normalmapping verwendet. Beim Nanobot ist der Effekt nicht erkennbar, da die Texturen leider gespiegelt sind.

9.3 Parallax Occlusion Mapping

Kann mit F7 ein und ausgeschaltet werden. Der Effekt kann bei der Mauer und bei der Plattform besichtigt werden.

10 View-Frustum-Culling

View-Frustum-Culling wurde mit einem geometrischen Ansatz gelöst. Dabei wird festgestellt auf welcher Seite der View-Frustum-Ebenen sich die Eckpunkte der Bounding-Boxen von Objekten befinden. Sind alle Eckpunkte eines Objekts auf der falschen Seite einer Ebene, wird das Objekt nicht gezeichnet. Das Ergebnis kann man besonders gut beobachten, wenn man mit F5 in die Dritten-Personen-Perspektive wechselt.

11 Animierte Objekte

Wir haben das Ziel Objekt animiert. Um den Ziel-Block fliegt ein kleinerer Block, der wiederum von einem noch kleineren Block umkreist wird.

12 Verwendete Tutorials

Neben den Unterlagen aus dem viertägigen OpenGL-Einführungskurs zu Beginn des Semesters, wurden Tutorials von den Seiten <https://learnopengl.com> und <http://www.bulletphysics.org> verwendet. Dazu zählen die Steuerung der Kamera, das Einlesen der Modelle mittels Assimp, Verknüpfung der Modelle mit Bullet, und die Beleuchtung der Objekte.

Für View-Frustum-Culling wurde <http://www.lighthouse3d.com/tutorials/view-frustum-culling> verwendet.