

Hopsi Endabgabe

Gameplay

In diesem Spiel ist Hopsi der Hase unser Haupcharakter. Er versucht während dem Spiel, dem Weihnachtsmann seine Mütze wieder zu bringen. Diese hat er im Dorf verloren als die Schneekönigin den Weihnachtsmann entführt hat. Die Schneemänner, welche Untertanen der Schneekönigin sind, versuchen Hopsi von seiner Mission aufzuhalten, indem sie ihn mit Schneebällen bewerfen. Hat Hopsi keine Leben mehr, ist das Spiel vorbei.

Steuerung für Hopsi:

- Pfeiltaste nach oben oder W um nach vorne zu gehen
- Pfeiltaste nach unten oder S um nach hinten zu gehen
- Pfeiltaste nach links oder A um nach links zu gehen
- Pfeiltaste nach rechts oder D um nach rechts zu gehen
- Leertaste zum Springen
- Maus gedrückt halten, um die Kamera zu drehen

Effekte

- Toon-Shading: kann im gesamten Spiel gesehen werden
- Konturen: Die meisten der im Spiel vorhandenen Objekte wurden schwarz umrandet
- Partikelsystem: Das Partikelsystem kann an den Schneeflocken erkannt werden
- Skeletal-Animation: Der Hauptcharakter Hopsi wurde dadurch animiert.

Toon-Shading

Alle Objekte werden toon-geshadet. Dazu wird einfach in zwei Stufen diskret statt kontinuierlich geshadet. (0.5P)

(<http://www.sunandblackcat.com/tipFullView.php?l=eng&topicid=15>.)

Konturen

Die Konturen der Objekte wurden erzeugt, indem einfach vor der richtigen Geometrie die Backfaces mit dickeren Linien gezeichnet werden. (0.5P)

(<http://www.sunandblackcat.com/tipFullView.php?l=eng&topicid=15>.)

Partikelsystem

Das Partikelsystem läuft auf der GPU und funktioniert in zwei Schritten. In einem Update-Shader werden zunächst von einem Spawnpunkt weg die Partikelpunkte erzeugt und upgedatet, der Render-Shader erzeugt dann Quads mit Texturen aus den Punkten und zeichnet sie. (1P)

(<http://www.mbsoftworks.sk/tutorials/opengl3/23-particle-system/>)

Skeletal-Animation

Da wir eine Skeletal-Animation verwendet haben, haben wir bei Hopsi auf die hierarchischen Transformationen verzichtet. Dadurch bewegt sich Hopsi noch flüssiger. Hier wurden ganz klassisch die Skelettdaten an die GPU übergeben, sodass mit den über Vektoren und Quaternionen interpolierten Gelenkposen das Mesh die Bewegung darstellen kann. (2P)

(<http://ogldev.atspace.co.uk/www/tutorial38/tutorial38.html>)

Komplexe 3D Modelle

Die Objekte in diesem Spiel wurden mit Hilfe von einem Assimp-Loader in das Spiel geladen.

Physik

Wurde bereits in der letzten Abgabe mit PhysX implementiert.

Lightmaps

Die Ebene, auf der der Weihnachtsmann gerettet werden möchte, zeigt den Schatten des Turms der Eiskönigin. Dies wurde in Blender erstellt.

HUD

Text wurde mit Freetype gerendert, für die Herzen wurde ein eigener Sprite-Shader geschrieben.

View-Frustum Culling

Das Frustum Culling kann mit Hilfe von der F8 Taste ein und ausgeschaltet werden. Das Frustum wird aus den Kameradaten berechnet. Jede Geometrie wird mit einer Kugel angenähert, um zu Cullen.

Debug Options

- F2 - Frame Time on/off
- F3 - Wire Frame on/off
- F4 – Backface Culling on/off
- F5 – New Game
- F6 – Debug Camera on/off
- F8 - View-frustum Culling on/off

Die Debug Camera kann wie folgt verwendet werden:

- Pfeiltaste nach oben oder W um nach vorne zu gehen
- Pfeiltaste nach unten oder S um nach hinten zu gehen
- Pfeiltaste nach links oder A um nach links zu gehen
- Pfeiltaste nach rechts oder D um nach rechts zu gehen
- 9 um die Kamera nach oben zu bewegen
- 0 um die Kamera nach unten zu bewegen

Der Spieler bekommt durch geschriebenen Text im Spiel Feedback. Anfangs wird die Aufgabe, welche zu erledigen ist, angezeigt. Wenn der Spieler verliert, erscheint „GameOver“, wenn er gewinnt erscheint dies jedenfalls im Spiel.

Features

Die Bewegungssteuerung erfolgt relativ zur Kameraansicht, sodass vorne stets vorne ist. Der Hase und der Weihnachtsmann können die Mütze aufsetzen!

Belichtung

In diesem Spiel wird die Sonne als globale (gerichtete) Lichtquelle verwendet.

Bibliotheken

Folgende Bibliotheken wurden verwendet:

- PhysX: als Physik Engine (<https://github.com/NVIDIAGameWorks/PhysX-3.4>)
- assimp.lib: zum Laden von Objekten (http://assimp.sourceforge.net/main_downloads.html)
Version 3.3.1
- ECG Framework (mit glew32s.lib, glfw3.lib)
- Freetype: für Schrift (<https://www.freetype.org/download.html>)
- DevIL für Bilder (<http://openil.sourceforge.net/>)

Tools für Modelle

Die meisten Modelle wurden aus dem Internet geladen. Die anderen Modelle wurden in Blender erstellt.