1125099 Johannes Wawerda, 1227109 Florian Steinschorn

RPS Tanks

Controls

Mapped für Xbox Controller bzw. andere mit gleichem Mapping
(getestet mit „MotioninJoy Gamepad tool“ setting: „Xinput-Default“)

- rechter Analogstick: Zielen/Kamerasteuerung
- linker Analogstick: Fahren
- rechte Schultertaste: Schuss
- linke Schultertaste: Selbstzerstörung
- „A“ -Taste: Invertierung der Kamera-Y-Achse

Das Spiel erkennt die Anzahl der Controller automatisch. Wenn keine Controller angesteckt sind, wird ein Einzelspieler Testmodus aktiv, in dem man sich mit WASD und Maus bewegen kann, mit 1-4 Type wechseln und mit „k“ Selbstzerstörung einleiten kann.

Implementation

- ein SceneGraph zum Updaten und Zeichnen der Welt
- eine Heightmap die aus einem Grayscale-Image den Boden ausliest
- Einladen von beliebigen .obj Models sowie Texturen
- Physic Simulation mittels bullet Engine: einmal mit TriangleMeshShape für den Boden und ein BoxShape für den Panzer, sowie SphereShape GhostObjects für die Pickup Items

- Ein mittels bullet erstelltes Raycastvehicle
- Input Handling mittels glfw
- Zielen mittels ständigen Raycast in der Mitte des Bildschirms und nachziehen der Panzerrohr Drehung/Kippung
- Vollständig selbst aufgenommene Sounds
- View Frustum Culling via BoundingBoxes

Features

- Physik basierte Panzersteuerung
- Schießen mit Raycast-Treffererkennung
- Pick-Ups um den Panzertyp zu wechseln
- Minenfeld als Kampfbereichsbegerenzung
- Splitscreen Multiplayermodus
- Punktezähler
- Kill-Streak Announcer

Objects

Alle Objects werden von der Sonne (directional light) und den Scheinwerfern der Panzer (spotlights) beleuchtet. Der Specular Anteil ist besonders bei den Panzern sichtbar, da diese mehr glänzen als die restlichen Objekte.

- Heightmap:
  Die Heightmap wird von einem schwarzweiß Bild eingelesen

- Panzer:
  Models erstellt in Blender, Texturen in Gimp. Haben vorne 2 Spotlights und hinten 2 „fake“ Lichter (siehe „texture illumination“).

- Schilder:
  Die Schilder markieren das Minenfeld, wenn man ihnen zu nahe kommt, explodiert man.

- pick-ups:
  Mit Hilfe der Pickups kann man den Type des Panzers wechseln (Schere/Stein/Papier/Ente). Die Ente ist ein Joker und kann alle anderen abschießen, wirkt aber nur eine begrenzte Zeit lang.

Effects

- Spotlights:
  Jedes Licht hat eine Leuchtrichtung und einen Leuchtwinkel, ist dieser Leuchtwinkel nicht 360° entsteht ein Spotlight. Im Fragment-Shader wird hierzu der Winkel zwischen der Leuchtrichtung und der Verbindung Licht-Object mit dem halben Leuchtwinkel verglichen.
- shadow maps:
  Vor dem Hauptrendering wird die Szene aus der Sicht der Sonne gerendert um eine Tiefentextur zu erstellen, die dann später abgefragt werden kann um zu entscheiden ob sich der Punkt im Schatten befindet oder nicht.
  Spotlights haben aus Performancegründen keinen Schatten.
- cel shading:

Jeder Farbkanal wird auf 5 diskrete Werte gemapped. Ausnahmen siehe „texture illumination“

- „texture illumination“:
  Offizieller Name dazu nicht gefunden, ähnlich Borderlands Kisten. Gewisse Farben werden bei der Beleuchtungsberechnung ignoriert und direkt weitergeleitet. Dadurch sieht es so aus, als würden diese Teile der Textur leuchten, ohne wirklich Lichtquellen zu sein. (Farbhighlights auf den Panzern, Scheinwerfer, Hover-Thingys am Panzerboden)

- edge detection:
  Anstatt auf den Bildschim wird das Bild in einen Framebuffer gerendert, zusammen mit einer Normalen-Textur. Bei einem 2ten Shading Durchlauf kann nun ein Sobel-Filter über die Normalen-Textur laufen gelassen werden um Kanten zu finden und im endgültigen Bild hervozuheben.
  Wir sind mit diesem Effekt nicht wirklich zufrieden und überlegen ihn gegen depth of field auszutauschen.
- Gpu particle effects (Transform Feedback):
  Die (halbtransparenten) Partikel werden von einem Quellpartikel ausgehend erzeugt. Dabei kann man ihnen Farbe, Richtung, Kegelwinkel, Geschwindigkeitsbereich usw. als Parameter geben. Die Partikel die beim Schießen aus dem Rohr kommen werden außerdem mit der Zeit langsamer und ändern ihre Farbe (werden dunkler).

Command Line Help

F1 - Help
F2 - Frame Time on / off
F3 - Wire Frame on / off
F4 - Textur - Sampling - Quality: Nearest Neighbor / Bilinear
F5 - Mip Mapping - Quality : Off / Nearest Neighbor / Linear
F6 - Shadows on / off
F7 - Edge Detection on / off
F8 - Viewfrustum - Culling on / off
F9 - Transparency on / off
F10 - View Frustum Counter on / off
F11 - Texture Illumination on / off

1125099 Johannes Wawerda, 1227109 Florian Steinschorn

Referencen und Libraries

In der Entwicklung haben wir einige Online Tutorials verwendet:

http://www.opengl-tutorial.org/
http://www.tomdalling.com/blog/category/modern-opengl/
http://www.bulletphysics.org/Bullet/phpBB3/viewtopic.php?f=17&t=8086&view=next
http://www.mbsoftworks.sk/index.php?page=tutorials&series=1&tutorial=24
http://bulletphysics.org/mediawiki-1.5.8/index.php/Tutorial_Articles
http://bulletphysics.org/mediawiki-1.5.8/index.php/Getting_Started
http://www.lighthouse3d.com/tutorials/view-frustum-culling/
http://www.informatik-forum.at/forumdisplay.php?188-Computergraphik-UE
http://www.mbsoftworks.sk/index.php?page=tutorials&series=1&tutorial=26

Verwendete Bibliotheken:

bullet http://bulletphysics.org/
assimp http://assimp.sourceforge.net/
FreeImage http://freeimage.sourceforge.net/
openGl, glew, glm https://www.opengl.org/sdk/libs/
glfw http://www.glfw.org/download.html
irrklang http://www.ambiera.com/irrklang/

Tools

- Blender
- Gimp
- Audacity