Submission 2 – Dokumentation
Battle Pilots

Gameplay

Das Ziel des Spiel ist es, innerhalb einer gewissen Zeit (Countdown links unten) ein Target (blinkend animiert) mit einer Rakete zu treffen. Schafft man es, kommt man ins nächste Level, das Target wird random neupositioniert, die Zeit wird reduziert, und man muss das Target wieder finden und treffen. Schafft man es nicht, oder kollidiert mit irgendeinem Objekt in der Szene, verliert man. (Anm.: Aufgrund der Form der Flügel der Jets haben die Bounding Spheres dieses Components einen großen Radius, weswegen es zur Kollision kommen kann, auch wenn sich die Objekte nicht direkt berühren. Man sollte zu den drei Enemy Jets in der Szene Abstand halten.)

Instructions

Den Jet steuert man mit den Up and Down, A und D Tasten. Mit W und S beschleunigt/bremst man. Mit der Leertaste schießt man (alle 5 Sekunden). Das Ziel ist es, das blinkende Target in der Szene zu finden, und mit einer Rakete abzuschießen. Kollidiert man mit irgendeinem Objekt in der Szene (auch Boden/Skybox) oder lässt sie zu nahe in den Luftraum des Jets kommen, ist das Spiel vorbei. Mit F6 als Cheatcode verhindert man den Game Over Screen (und damit greift auch die Collision Detection nicht mehr).
In der config.js Datei kann die Screengröße und Fullscreen ja/nein eingestellt werden. (Anm.: Screentext ist tlw. an 1600x800 als Auflösung gekoppelt.)

Effects

Lightmap – Statische Objekte in der Szene sind durch eine Lightmap beleuchtet/beschattet. → http://www.opengl-tutorial.org/intermediate-tutorials/tutorial-15-lightmaps/ (Als Erklärung, Objekte selbst modelliert)
+Separate Texture – Statt die Lightmap in der Textur des Objekts abzuspeichern, sind die beiden separate Texturen und werden erst zur Laufzeit durch einen Fragmentshader kombiniert (jeweils 50% der beiden Texturen). Die Lightmap wird dabei erst zur Laufzeit ins Spiel geladen. → Zusätzliches Textureloading und Shaderkombination selbst implementiert, als Erklärung des Blendings: http://learnopengl.com/#!Getting-started/Textures, http://stackoverflow.com/questions/2525251 für das korrekte Binden der Texturen
Es sind alle statischen Objekte (Heightmap, die einzelnen Komponenten des Lagers) durch separate Lightmaps beleuchtet. Die beweglichen Objekte (Jets, Targets, Cargo) sind durch ein Pointlight dynamisch beleuchtet.
Animated Textures – Um das Target in der Welt besser aufzufinden, bekommt es eine blinkend animierte Textur. Dafür werden alle Frames des Videos zur Laufzeit ins Spiel geladen und nacheinander mit kurzem Abschnitt an das Objekt gebunden, um das Video abzuspielen. → Als Erklärung der Idee dahinter: http://www.swiftless.com/tutorials/opengl/texture_animation.html

Complex Objects

Es gibt einige komplexe Objekte in der Szene, wie die Jets und die Heightmap, und alle Objekte werden durch einen Model Loader (Assimp) importiert. Es werden keine Objekte im Code erzeugt. Jet, Heightmap und Rakete sind aus fremder Quelle, alle anderen Objekte wurden von mir in Blender erstellt. Der Modelloader kann verschiedene Datenformate importieren.

Animated Objects

Unterhalb des Jets befindet sich eine Box, die Cargo an einer fixen Eisenstange darstellt. Diese Box ist immer relativ zum Boden des Fliegers, hat aber zusätzlich eine eigene Rotation (stellt Schwanken im Wind dar). Box und Flieger sind separate Meshes und Objekte, aber die Position der Box ist immer abhängig von der des Fliegers.

View Frustum Culling
Damit Mesh Components, die im Frustum nicht sichtbar sind, gar nicht erst gezeichnet werden, gibt es eine Frustum Klasse, die sich aus den Matrizen der aktiven Kamera berechnet. Es wird der Clipping Plane Approach angewandt. Das Frustum hat also 6 Clipping Planes. Sobald eine Bounding Sphere innerhalb einer der sechs Planes ist oder sie schneidet, wird das Mesh Component gezeichnet, sonst wird es nicht gezeichnet. Diese Funktion wird über F8 getoggelt. Im Performance Output sieht man den Unterschied in der Anzahl der gezeichneten Objekte, vor allem, wenn man nach oben in den Himmel schaut, und über '1' die unabhängige Kamera aktiviert.
→ http://www.lighthouse3d.com/tutorials/view-frustum-culling/,
https://wiki.delphigl.com/index.php/Tutorial_Frustum_Culling

Transparency
Mit F9 kann man die Transparency toggeln. Wird sie eingeschaltet, werden alle Objekte außer dem Text transparent gezeichnet.

Experimenting with OpenGL
Es werden VBOs und VAOs beim Handeln der Meshes verwendet. Mit den vorgesehenen Tasten kann zwischen den verschiedenen MipMapping (3) und Texture Sampling Quality (2) Stufen geschaltet werden (die beiden werden korrekt kombiniert). Die Auswirkung sieht man am besten in der Textur der Heightmap.
Mit F1 kommt man zu einem Help Screen, wo alle belegten Tasten aufgelistet und erklärt sind. Währenddessen pausiert das Spiel. Mit F2 toggelt man den Performance Output on und off. F3 toggelt Wireframe Modus. F4 & F5 schaltet zwischen den Textur Stufen (s.o.). F8 schaltet Culling ein und aus. F9 toggelt die Transparenz. Man sieht im Text im Spiel, welche Einstellungen aktiv sind (= Feedback Message).
F6 ist eine Cheat-Taste, mit der der Game Over Screen disabled wird. Dadurch kann man sich komplett frei im Raum bewegen, denn die Collision Detection hat keine Auswirkung mehr.
→ Text Rendering: http://learnopengl.com/#!In-Practice/Text-Rendering

Illuminated Objects
Der Boden, die Flieger, Cargo, das Target sind durch das Pointlight beleuchtet, da sie keine statischen Objekte sind.
Die statischen Objekte, wie Heightmap und das Lager, sind durch eine Lightmap beleuchtet, wobei die Lightmap-Textur erst zur Laufzeit geladen und auf die Objekte gemappt wird.

Additional Libraries
Durch Assimp werden die Modelle geladen. → www.assimp.org
Durch IrrKlang werden Soundeffekte und Musik abgespielt. → www.ambiera.com/irrklang
Durch FreeType wird Text dargestellt. → https://www.freetype.org
Durch DevIL werden Texturen geladen. → openil.sourceforge.net
GLM für Mathematik. → glm.g-truc.net

Special Features
Man kann zwischen verschiedenen Kameras und Ansichten schalten. Taste '1' wechselt zwischen einer Kamera, die an den Jet gekoppelt ist, und einer unabhängigen, die frei im Raum bewegbar ist. Taste '2' wechselt zwischen 3 Ansichten, wenn die Fliegerkamera aktiv ist. Mit der Leertaste schießt man eine Rakete ab (nur alle 5 Sekunden), dabei gibt es einen Soundeffekt. Sobald man sich irgendeinem Objekt zu sehr nähert, ist das Spiel vorbei, da der Flieger einen gewissen Luftraum frei halten muss, der nicht unterschritten werden darf. Schießt man das Target ab, kommt man ins nächste Level und das Target wird zufällig neu im Raum positioniert.

Tools for Models

Außer Heightmap und Flieger/Rakete alle Modelle selbst in Blender erstellt. Heightmap & Flieger/Rakete sind von tf3dm.com.