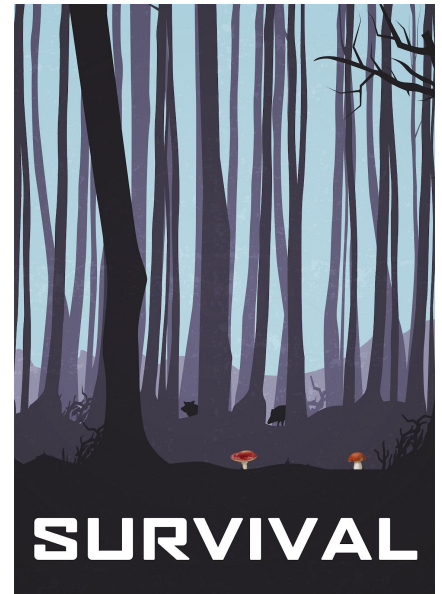


SURVIVAL

Martin Crepaz 11776187
Philipp Lutz 01503299



Überblick

Gameplay

Ziel des Spiels ist es, Nahrung zu sammeln, um so das eigene Überleben zu sichern. Auf der Map finden sich neben Pilzen auch Schweine, welche beide gesammelt werden können. Die unterschiedlichen Nahrungsangebote liefern auch unterschiedlich viel Energie. Jedoch ist Vorsicht geboten, da nicht alle Pilze auch genießbar sind. So kann der Verzehr eines giftigen Pilzes schnell zum Tod führen.

Der Spieler hat gewonnen, wenn seine Energie auf 100% angestiegen ist. Über die Zeit verliert der Spieler an Energie und sollte das Energielevel auf 0% gesunken sein, hat der Spieler verloren.

Steuerung

W,A,S,D	Charakter steuern
Maus bewegen	innerhalb der Spielwelt umsehen
linke Maustaste	Schwammerl/Schweine sammeln
Mausrad	Zoomen
Space	Springen
T	Taschenlampe ein-, ausschalten
E	Stein werfen
ESC	Fenster schließen
F6	HUD aktivieren/deaktivieren
F8	View-Frustum Culling ein-, ausschalten

Änderbare Parameter

Über die settings.ini-Datei ist es möglich, die Bildschirmgröße und die Aktualisierungsrate einzustellen. Weiters kann auch die Option für Fullscreen aktiviert und der Fenstertitel festgelegt werden. Über "brightness" ist es auch möglich die Helligkeit einzustellen.

Zusätzlich kann über diese Datei auch die Größe des Sichtfeldes und der Wert für die vordere und hintere Clippingebene eingestellt werden.

3D Geometrie

Die Spielwelt beinhaltet einen Wald mit Bäumen und Baumstümpfen, sowie verschiedene Arten von Pilzen, die zufällig in der Nähe von Bäumen wachsen. Im Mittelpunkt der Karte befindet sich eine Waldhütte vor der ein Lagerfeuer brennt und die nähere Umgebung beleuchtet. Zusätzlich gibt es noch Schweine die sich in der Spielwelt bewegen und als Nahrungsquelle dienen. Der See, der aus einer Vertiefung in der Karte besteht, besitzt eine Wasseroberfläche welche sich bewegt und so Wellen simuliert. Begrenzt wird die Spielwelt durch eine Berglandschaft welche teilweise begehbar ist.

Features

Physik/Kollisionserkennung

Das Spiel verfügt über eine mit Physx implementierte Physik die eine Schwerkraft und Kollisionserkennung bietet. Schwerkraft erlaubt es zwar im Spiel zu Springen jedoch verhindert sie eine freie Bewegung in der 3. Dimension. Sämtliche Objekte (mit Ausnahme des Wassers und des Adlers) verfügen über eine Kollisionserkennung welche das Fallen durch den Boden oder das Laufen durch Objekte verhindert. Jedoch erlaubt die Kollisionserkennung das Begehen von niedrigen Stufen und Erhebungen im Gelände, sowie das Abprallen geworfener Steine. Die Limitation der Bewegung der Kamera erfolgt durch einen Physx Character Controller welcher anhand von Kollisionserkennung bestimmt welche Bewegungsrichtung zur derzeitigen Position zulässig ist. Geworfene Steine werden als dynamische Akteure generiert und in Blickrichtung beschleunigt.

HUD

Das Heads-up-display gibt Information über Steuerung und den aktuellen Energielevel, sowie ein Kreuz zum Anvisieren der Tiere/Pilze und wie viele davon jeweils gesammelt wurden. In der rechten oberen Ecke finden sich auch die Informationen, wie viele Pilze schon gesammelt und Schweine gejagt wurden.

Dynamische Objekte

Zufällig generierte Tiere und Pilze sorgen dafür, dass das Spiel jedes Mal etwas anders ist und von Spielenden nicht einfach der selbe Weg zum Sieg führt.

Es gibt 3 verschiedene Arten von Pilzen:

Essbar: kleine braune Pilze die bei Verzehr (Mausklick links) 10% Energie wiederherstellen.

Speedboost: violett leuchtende Pilze welche das Lauftempo, nach Verzehr, permanent erhöhen.

Giftig: rote und grüne Pilze welche bei Verzehr sofort zu einer Niederlage führen.

Es gibt eine Art von Tieren:

Schweine: spawnen zu zufälligen Zeitpunkten und laufen über die Map. Bei Verzehr stellen sie 20% Energie wieder her.

Beleuchtung

Zu Spielbeginn ist der Wald noch hell erleuchtet, jedoch geht die Sonne schnell unter und die anderen Lichtquellen werden die einzige Möglichkeit sich im Wald zurecht zu finden.

Es gibt 5 verschiedene Arten von Lichtern:

Tageslicht: Ein Directionallight, welches als Sonne dient und der Karte am Tag Licht spendet.

Mond: Ein Pointlight, welches in der Nacht etwas Licht spendet, jedoch auch nicht ausreichend und auch nicht auf der gesamten Karte.

Taschenlampe: Ein Spotlight, welches helfen soll in der Nacht die Orientierung zu behalten. Kann mit der Taste T ein- und ausgeschaltet werden. Wird im Laufe des Spiels immer wichtiger um Nahrung zu finden. Sie leuchtet immer in Blickrichtung.

Lagerfeuer: Ein Pointlight, welches die nähere Umgebung erleuchtet. Es befindet sich direkt beim Spawnpunkt und dient als Orientierung.

Leuchtpilze: Die bereits erwähnten lila Pilze sind ebenfalls Pointlights und spenden auch ein wenig Licht jedoch auch nur genug um sich anhand dessen zu Orientieren, sie schneller zu finden und potenziell umliegende Pilze sehen zu können.

Sounds

Während des Spiels läuft im Hintergrund ein Sound, der die Umgebungsgeräusche in einem Wald simulieren soll. Zusätzlich hört der Spieler auch das Feuer knistern, wenn er sich diesem nähert. Und auch beim See ist ein plätschern wahrnehmbar, wenn man sich in der Nähe des Sees befindet. Für die positionsabhängige Geräuschkulisse wurde mit 3D Sounds gearbeitet, die je nach Position des Spielers auf der Karte das Geräusch lauter oder leiser abspielt.

View-Frustum-Culling

Mit F8 kann während des Spiels view-frustum-culling ein- und ausgeschaltet werden.

Dadurch werden Objekte die sich nicht im Sichtfeld befinden nicht gerendert.

Derzeit berücksichtigt nur das Rendern von See, Schildern, Feuerstelle und der Hütte das view-frustum.

Effekte

CPU Particle System

Dieser Effekt wurde benutzt um die Funken bei der Feuerstelle zu erstellen. Dazu werden kleine Partikel erstellt und deren Position etwas zufällig nach oben und zur Seite verändert wird, um einen realistischen Funkenflug zu erzeugen. Über den Shader verändert sich auch die Farbe der Partikel während des Flugs. Ein Partikel hat dabei auch eine gewisse Lebensdauer die zu Beginn festgelegt wird. Hierzu gibt es 4 verschiedene Zeiten von Lebensdauer um einen realistischeren Eindruck zu vermitteln und damit auch einzelne Partikel höher in die Luft steigen als andere und trotzdem nicht zu viele Partikel gleichzeitig vorhanden sind.

Vertex Shader Animation

Wird verwendet um die Wellen im See zu erzeugen. Mit Sinus und Kosinus Berechnungen werden Wellenbewegungen simuliert. Damit sich die Welle bewegt musste dem Vertex Shader außerdem eine Zeitvariable übergeben werden. Diese wird während der Laufzeit des Programms immer um einen bestimmten Wert erhöht. Die Oberfläche des Wassers besteht

dabei aus einer Textur. Über Variablen und Parameter kann die Wellenhöhe sowie die Wellenlänge variiert werden. Für die Berechnung der Helligkeit und Farbe wird ein Fragment Shader verwendet welcher eine korrekte Belichtung durch das einbeziehen der vorhandenen gerichteten Lichter, Punktlichter und Spotlichter vornimmt.

Simple Normal Mapping

Ein paar der 3D Modelle werden mit einem Shader gerendert der sich Normal Mapping zu nutze macht um so einen Struktureffekt zu geben. Dies erfolgt über die Normal Maps dieser Modelle deren rgb werte als ein Vektor verwendet werden um die Lichtreflexion so zu berechnen als wäre dieser Vektor die Flächennormale. Die Modelle die mit diesem Shader gerendert werden sind: Hütte, Lagerfeuer

Bloom/Glow

Der Bloom Effekt wird verwendet um einen Leuchteffekt zu erzeugen. Zu sehen ist dieser Effekt bei den leuchtenden lila Pilzkappen deren Ränder geblurt werden um den Eindruck zu machen, dass sie leuchten. Dies erfolgt über das Rendern in 2 Framebuffers einen für das reguläre Rendern der andere für die gewünschten bzw besonders hellen Bereiche. Anschließend werden diese Kanäle übereinander gelegt und mit einem Gauß Filter an den überschneidenden Stellen geblurt. Weiters wird dieser Effekt unterstützt indem eine Lichtquelle an die Position dieser Pilze gesetzt wird die die umliegenden Bereiche farbigem Licht beleuchtet.

Specular Map

Das Rendering von Feuerstelle und Hütte nutzt Specular Maps um die specular Werte der Texturen an der korrespondierenden Stelle zur Farbberechnung zu verwenden, anstatt einen fixen wert für die gesamte Textur zu nehmen.

Environment Map

Basis ist der Environment Map ist eine Skybox welche den Nachthimmel darstellen soll. Eingesetzt wird die Environment Map beim See, welcher auf der Wasseroberfläche eine Reflexion der Skybox simuliert. Um keine vollständige Reflexion auf der Wasseroberfläche zu erhalten wurde diese mit der zuvor verwendeten Texture des Wassers verrechnet, wodurch diese nur noch ganz leicht zu sehen ist. Dies hat auch den Vorteil, dass die Wellenbewegungen und Wellenspitzen besser zu erkennen sind.

Bibliotheken

GLFW <https://www.glfw.org/>

ASSIMP <https://www.assimp.org/>

FreeType <https://www.freetype.org/>

PhysX https://gameworksdocs.nvidia.com/PhysX/4.1/release_notes.html

GLM <https://glm.g-truc.net/0.9.9/index.html>

Glad <https://glad.dav1d.de/>

Referenzen

OpenGL

<https://learnopengl.com/>

CPU Particle System

<https://learnopengl.com/In-Practice/2D-Game/Particles>

<http://www.opengl-tutorial.org/intermediate-tutorials/billboards-particles/billboards/>

HUD

<https://learnopengl.com/In-Practice/Text-Rendering>

Vertex Shader Animation

<https://www.youtube.com/watch?v=5yhDb9dzJ58>

Physx

<https://gameworksdocs.nvidia.com/PhysX/4.0/documentation/PhysXGuide/Index.html>

https://github.com/NVIDIAGameWorks/PhysX-3.4/tree/master/PhysX_3.4/Samples

https://github.com/NVIDIAGameWorks/PhysX-3.4/tree/master/PhysX_3.4/Snippets

Code snippets aus den kompilierten Binaries.

Bloom

<https://learnopengl.com/Advanced-Lighting/Bloom>

<https://learnopengl.com/Advanced-OpenGL/Framebuffers>

Normal Mapping

<https://learnopengl.com/Advanced-Lighting/Normal-Mapping>

Environment Map

<https://learnopengl.com/Advanced-OpenGL/Cubemaps>

3D-Modelle

<https://www.turbosquid.com>

<https://free3d.com>

Skybox-Textur

https://www.reddit.com/r/KerbalSpaceProgram/comments/341lid/check_out_this_skybox_i_made_more_info_download/

Gras-Textur

<https://www.pinterest.at/pin/367676757052083137/>

INIReader

Framework (ECG)

View-Frustum Culling

<http://www.lighthouse3d.com/tutorials/view-frustum-culling/>

Audio/Sounds

<https://learnopengl.com/In-Practice/2D-Game/Audio>

<https://www.ambiera.com/irrklang/tutorials.html>

<https://noises.online/>