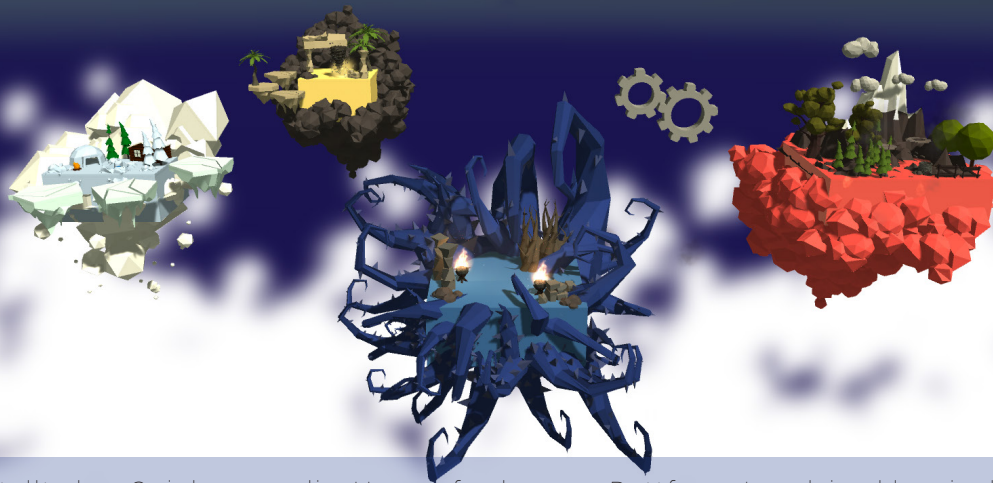




FLYING ISLES



Flying Isles stellt den Spieler vor die Herausforderung, Plattform-Level im klassischen 3D-Jump 'n' Run-Genre vor dem Hintergrund eines klaren, abstrakten Grafikstils zu bewältigen. Die in der Schwierigkeit steigenden Level sind dabei auf vier schwebende Inseln verteilt, die sich thematisch alle stark unterscheiden und zu deren Erreichen der vorige erfolgreich abgeschlossen werden muss.

Gamedesign:

Im Kern ist Flying Isles ein Plattform/Jump 'n' Run-Spiel, bei dem der Spieler einen Charakter durch verschiedene Level führen muss. Jeder Level ist dabei auf einer fliegenden Insel angesiedelt und birgt dadurch die Gefahr in sich, von dieser hinunter in den Tod zu fallen. Typisch für einen Plattformers muss der Spieler auch in Flying Isles von Objekt zu Objekt springen und dabei teilweise anderen Hindernissen ausweichen, um am Ende einen vorgegebenen Punkt zu erreichen. Dabei ist der Spieler in seinen Möglichkeiten auf seine Kernfähigkeiten Laufen und Springen beschränkt. Flying Isles richtet sich an Spieler jeglicher Altersklasse, die die Herausforderung in Geschicklichkeitsspielen suchen. Der Schwierigkeitsgrad ist so angesetzt, dass das Spiel auch für jüngere und unerfahrene Nutzer zu bewältigen ist, obwohl sich der Schwierigkeitsgrad mit zunehmendem Freischalten der Level stetig erhöht. Diese Mechanik soll die sich steigernden Fähigkeiten des Spielers kompensieren und so eine anhaltende Herausforderung bieten. Die Level sind dennoch eher kurz gehalten, vermeiden langatmige Passagen und bieten somit kurzweiligen Spaß. Zusätzlich kann das Spiel im klassischen Sinne nicht verloren werden, da der Spieler beim Tod wieder auf den letzten Speicherpunkt gesetzt wird und die Anzahl der Versuche nicht eingeschränkt ist. Durch das Festhalten der Zeit, die benötigt wurde, um einen Level zu bewältigen, erhöht sich der Wiederspielwert und zusätzlich wird die Wettkampfmentalität der Spieler angesprochen, die so untereinander Bestzeiten vergleichen können.

Technical Design:

Empfohlen wird, das Spiel mit einem Gamepad zu spielen und somit den Avatar mit einem Steuerkreuz oder Analog-Stick sowie mit den Tasten des Gamecontrollers (Standard: X-BOX-Controller, alternativ: PS2-Controller oder Maus und Tastatur) zu steuern. Die Kamera kann frei vom Spieler bewegt und rotiert werden und ist lediglich in der Entfernung eingeschränkt. Dadurch kann man sich in unübersichtlichen Situationen einen Überblick verschaffen und Gefahren frühzeitig erkennen. Der optionale Eye-Tracker kommt in besonderen Situationen zum Einsatz und ermöglicht die Interaktion mit entfernt gelegenen Objekten. So können beispielsweise Objekte und Plattformen bewegt und rotiert werden, um Passagen im Spiel bewältigen zu können. Alternativ kann hierzu auch die Maus eingesetzt werden.

Style:

Bei Flying Isles handelt es sich um ein 3D-Spiel, das im sogenannten LowPoly-Stil gehalten ist, was bedeutet, dass alle Modelle über eine verhältnismäßig geringe Polygonzahl verfügen. So werden mehrere Vorteile erzielt. Durch diesen abstrakten und modernen Stil mit Wiedererkennungswert geht zusammen mit dem unaufdringlichen, aber konsistent gehaltenen Grafikstil der Fokus auf die zentralen Spielmechaniken nicht verloren, während trotzdem eine angenehme Atmosphäre geschaffen wird, die durch Details wie Umgebungsanimationen wie z.B. Wind zusätzlich bereichert wird.



Detailed Design Document:

Spielelemente:

- Plattformen sind das Grundelement, aus dem alle Level im Wesentlichen aufgebaut sind. Hierbei gibt es große Plattformen, die sogenannten Flying Isles, die die zentralen Strukturen in jedem Level darstellen und auf denen dann weitere Spielelemente platziert werden können. Zusätzlich gibt es kleinere Plattformen, die um sich selbst rotieren oder sich in eine bestimmte Richtung bewegen können, sobald sie aktiviert werden.
- Eine besondere Form der Plattform, sind die auseinanderbrechenden Plattformen, die nach kurzer Zeit zerfallen, wenn der Spieler sie zum ersten Mal berührt. Diese setzen die Spieler unter Zeitdruck und sorgen für ein gewisses Maß an Dynamik.
- Mit den Einsatz von trampolinen kann der Spieler große Höhenunterschiede überwinden. Zusätzlich muss der Zielpunkt während des Sprungs anvisiert werden, was eine weitere Herausforderung darstellt. Trampoline benötigen keine externe Aktivierung.
- Mit Stacheln versehene Kugeln werden dazu eingesetzt, bestimmte Bereiche eines Levels zu versperren und dem Spieler einen bestimmten Weg vorzugeben. Berührt der Spieler eine solche Kugel, wird er entweder direkt an den letzten Speicherpunkt zurückgesetzt oder von der Kugel in eine Richtung abgestoßen.
- Pendel schwingen von einem festgelegten Punkt zum anderen und blockieren so immer für kurze Zeit das Vorankommen des Spielers. Der Spieler muss die Bewegung des Pendels nun richtig einschätzen und ihm so ausweichen.
- In regelmäßigen Abständen finden sich in jedem Level Speicherpunkte (Checkpoints), die den Spielfortschritt festhalten. Wenn der Spieler von einer Plattform stürzt, wird der immer zum letzten Speicherpunkt zurückgesetzt. Speicherpunkte sind visuell so dargestellt, dass dem Spieler eindeutig klar wird, wann er einen Speicherpunkt erreicht hat.
- Gegner wie zum Beispiel Wirbelstürme werden auf den größeren Plattformen eingesetzt, um den Spieler weiter zu fordern. Sie bewegen sich zufällig oder auf festgelegten Pfaden, sodass der Spieler immer aufpassen muss, nicht plötzlich einen Gegner zu berühren. Sollte dies passieren, so wird der Spieler in eine Richtung gestoßen, was meistens dazu führt, dass er von der Plattform stürzt.
- Sogenannte Gaze Kristalle können durch den Einsatz des Eyetrackers dazu benutzt werden, bestimmte Plattformen oder andere Spielelemente zu manipulieren. So können bewegliche oder rotierende Objekte gestoppt oder in Bewegung versetzt werden.
- Eine Besonderheit des Dornenlevels ist der Einsatz einer Lichtquelle, die bestimmte Objekte sichtbar macht, wenn der Spieler, der die Lichtquelle trägt, in ihre Nähe kommt. Diese Spielmechanik sorgt dafür, dass der Spieler besonders vorsichtig vorgehen muss.

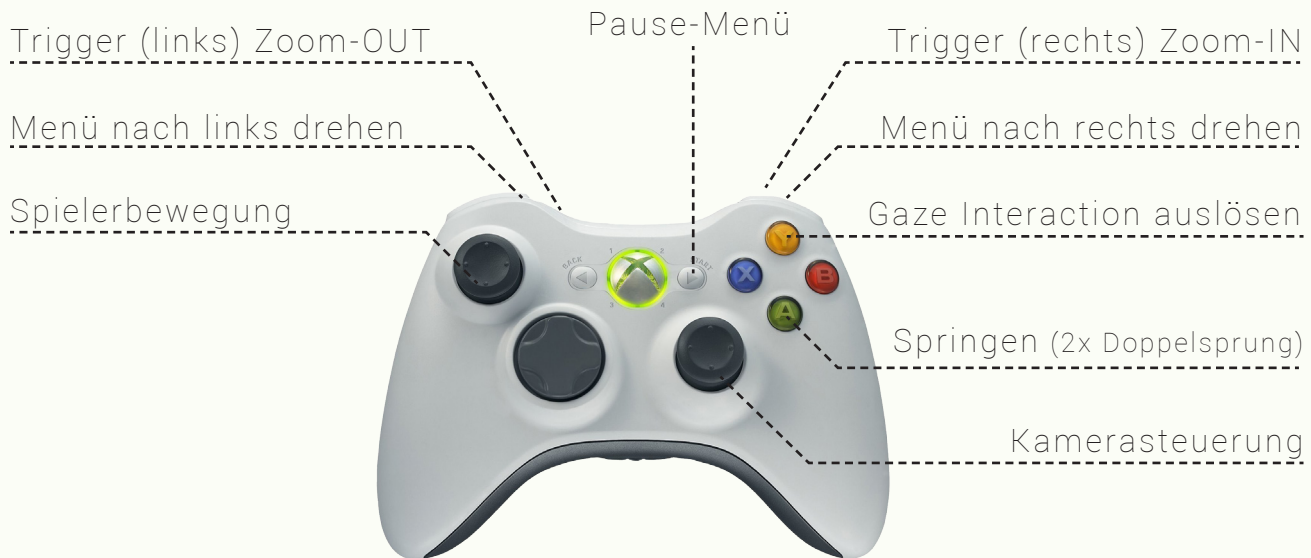
Spielregeln:

- Der Spieler kann sich frei auf den Plattformen in alle Himmelsrichtungen bewegen.
- Durch einmaliges Antippen der Sprungtaste kann der Spieler einen einfachen Sprung ausführen. Bei zweifacher Betätigung wird ein sogenannter Doppelsprung ausgeführt, der den Spieler besonders hoch und weit springen lässt.
- Manche Objekte können nur über den Eyetracker oder die Maus aktiviert werden.
- Fällt der Spieler von einer Plattform hinunter und ist keine andere Plattform unter ihm, wird er zum letzten Speicherpunkt zurückgesetzt.
- Wird der Spieler von einem tödlichen Objekt oder einem Gegner getroffen, stirbt er und wird somit auch zum letzten Speicherpunkt zurückgesetzt.
- Falls der Spieler zurückgesetzt wird, wird die ursprüngliche Levelstruktur wiederhergestellt. Somit können Spielpassagen, die vom Spieler zuvor bewältigt und verändert wurden, erneut gespielt werden und das erfolgreiche Bewältigen des Levels ist weiterhin möglich.
- Ein Level kann erst dann erfolgreich abgeschlossen werden, wenn der Zielpunkt erreicht ist. Wird ein Level erfolgreich beendet, so erhält der Spieler Zugang zum nächsten Spielabschnitt. Innerhalb eines Levels gibt es mehrere Speicherpunkte, die als Zwischenziele dienen und eine Erleichterung für den Spieler darstellen.

Tastenbelegung: (X-BOX-Controller)



FLYING
ISLES



Aufgabenbereiche der Teammitglieder: *

David Englmeier:

- + 50 Modelle in Blender einschließlich Playermodell und Gegner.
- Animationen, Rigging, Animations-Controller (Player, Gegner, Umgebung)
- Entwicklung der vier Themen: Forest, Desert, Thorns und Ice.
- Grafikstil, optische Gestaltung, Titel und Logo, Typographie.
- Skyboxen für alle Levels.
- Partikeleffekte (Nebel und Schneefall).
- Soundeffekte.
- Programmierung (Animation, Sprung-Schatten, Sound)
- Levelbau und Konzeption: IceLevel.
- Erstellung von Gamedesign Dokument, Storyboard, Moodboard.
- Testen und Bugfixing im Desert-, Forest- und Ice-Level.

Patrik Hagen:

- Controller-Input.
- Spielerbewegung bzw. -physik.
- Kameraführung.
- Pushback und Trampolinsprung.
- CheckPoints, Finish und Deadzone.
- LevelController.
- Zeit stoppen, speichern und anzeigen.
- Gaze-Integration in ActivatingObject.
- Menüarbeiten (nicht Grundkonzept).
- Laden von Leveln und Menüs.
- Speichern von Einstellungen, freigeschalteten Leveln und Bestzeiten.
- Testen und Bugfixing im Desert-, Forest- und Ice-Level.
- Bugfixing allgemein.

Elias Englmeier:

- Levelbau und Konzeption: Desert-Level (komplett)
- Levelbau und Konzeption: Ice-Level
- Implementierung und Testen der Gaze-Interaktion
- Konzeption und Umsetzung der Gaze-Interaktion durch das zentrale Gameobject "Gaze-Crystal".
- Skript zur Bewegungsübertragung auf den Spieler.
- Testen und Bugfixing im Desert-, Forest- und Ice-Level.
- Erstellung Gamedesign Dokument.

Linh Nguyen Vo:

- ca. 20 Modelle in Blender hauptsächlich für Ice- und Thorn-Level
- Partikelsystem (Firewalls) fürThorn-Level.
- Optische Ausgestaltung und Dekoration des Forest-Levels.
- Contenterstellung für Dornenlevel.
- Testen und Bugfixing: Dornenlevel.

Stefan Schätz:

- Levelbau und Konzeption: Forest-Level (einschließlich Gaze-Rätsel).
- Prefabs für komplexere Level-Elemente: Rotierende SpikeBars, rotierenden Plattformen u.a.
- Skripte für interaktive und sich bewegende Elemente in den Levels.
- Diverse Skripte für zusätzliche Interaktionsmöglichkeiten des Spielers.
- Komposition und Produktion der Hintergrundmusik in den Levels ("Forest Theme" und "Thorn Theme").

Thomas Wimmer:

- Gestaltung und Animation des Menüs.
- Konzept und Umsetzung sämtlicher Objekte mit denen der Spieler interagieren kann (außer Trampolin) und wie sie ausgelöst werden.
- Implementierung und Bugfixing beweglicher, rotierender, zerbrechlicher Objekte.
- alle zusätzlichen Skripte im Thorn-Level Grundlegende Struktur des Projekts und Utility-Skripts.
- Levelbau und Konzeption: Thorn-Level.
- Testen und Bugfixing: Thorn-Level, Forest-Level.

* falls gewünscht kann noch weitere Information zu nicht in der finalen Version enthaltenen Features geliefert werden.