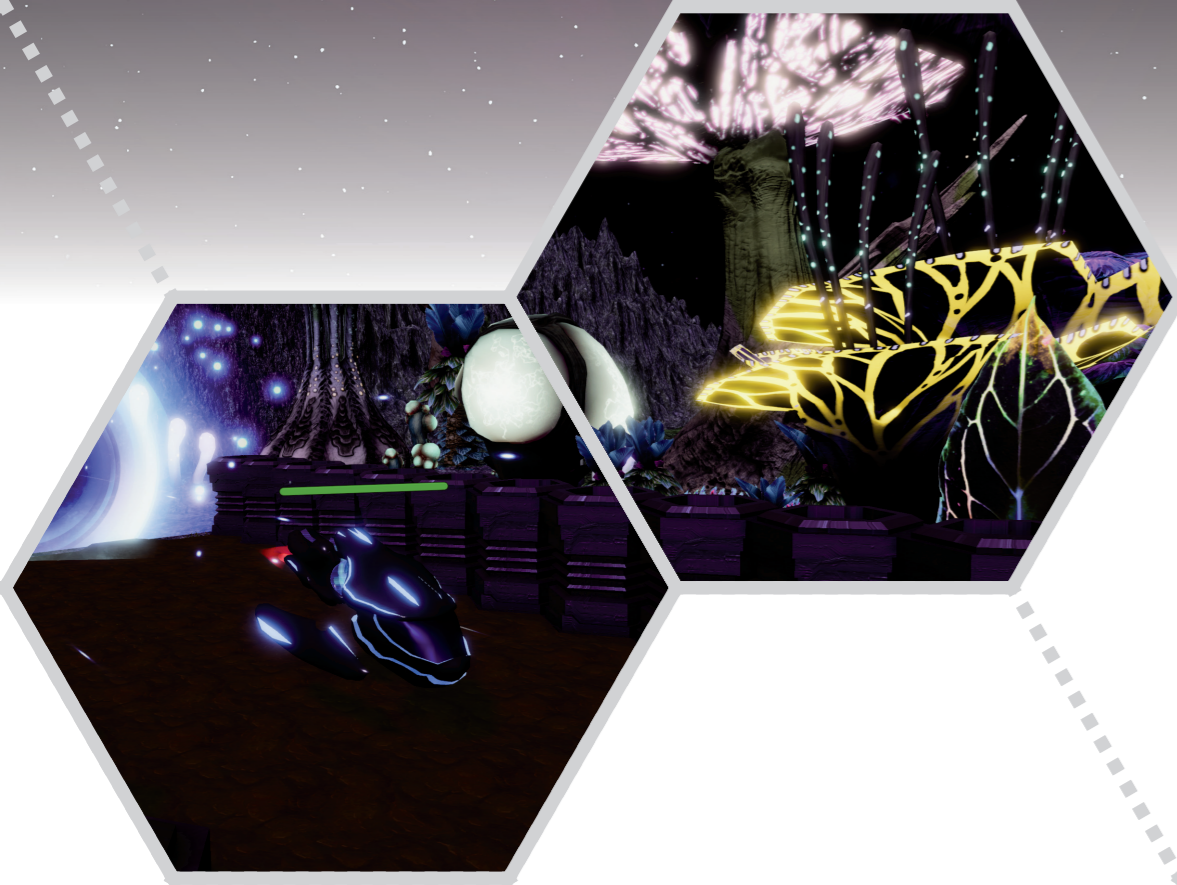




VIRTUAL REALITY



ARBEITSWEISE



Taiga.io

Als Projektmanagementmethode wurde das für agile Softwareentwicklung bekannte Vorgehensmodell „Scrum“ gewählt. Die Erstellung und Verwaltung der Sprints, User Stories und Tasks erfolgt über das webbasierte Projektmanagementtool Taiga.io. Hierüber wird gleichzeitig auch das Bug-Tracking abgebildet.



Slack

Zur Kommunikation innerhalb des Teams wird die Kollaborationsplattform Slack eingesetzt. Abweichend vom Standard der Projektmanagementmethode „Scrum“ dient dort ein gesonderter Channel als Ersatz für die Daily Scrum Meetings, worüber jedes Teammitglied seinen aktuellen Arbeitsstand mitteilt.



Wiki

Zur Organisation des Wissensmanagements innerhalb des Teams wurde ein zentraler Informationspool auf Basis der webbasierten Plattform MediaWiki bereitgestellt. Die ausführliche Dokumentation des Projekts ermöglicht es so neuen Teammitgliedern, sich schneller einzuarbeiten.

KURZINFORMATION

In Virtual Reality vereint AIRA die strategischen Elemente des Tower Defense Genres mit der packenden Action eines First Person Shooters. Das Spielerlebnis unterteilt sich somit grundsätzlich in zwei Phasen – eine **Strategiephase** und eine **Actionphase**. Diese finden dabei immer im Wechsel statt.

Zu Beginn einer jeden Runde befindet sich der Spieler in der Raumstation. Während der Strategiephase kann er dort Türme auf einer holografischen Darstellung der jeweiligen Map individuell platzieren. Die zur Verfügung stehenden „Defense Points“ geben dabei vor, wie viele Türme während einer Runde gebaut werden können. Über einen Ingame-Shop können neue Turmtypen, Waffen sowie andere Tools gegen „Credits“ gekauft werden. Bereits gebaute Türme können ausgewählt und anschließend ebenfalls gegen „Credits“ verbessert oder bei Bedarf wieder entfernt werden.

Um die Strategiephase abzuschließen, muss der Spieler zunächst einen Teleporter betreten. Daraufhin wird er direkt in das Level des jeweiligen Planeten transportiert. In der nun folgenden Actionphase kann der Spieler die zuvor gebauten Türme in voller Größe betrachten. Zudem hat er die Möglichkeit, sich auf den Mauern, welche den Weg der Gegner auf der Map eingrenzen, frei zu bewegen. Seine Aufgabe ist es nun, alle Gegner daran zu hindern, das Ziel am Ende der Levels zu erreichen. Nach jeder Runde erhält der Spieler weitere „Credits“, abhängig davon wie viele Gegner besiegt wurden und wie viel Schaden durch die Türme bzw. ihn selbst verursacht wurden.

Nach einer erfolgreichen Runde folgt die nächste Strategiephase. Der Spieler erhält hierzu weitere „Defense Points“ und kann nun seine bisherige Verteidigungsstrategie erweitern bzw. überarbeiten. Nach jeweils drei erfolgreich überstandenen Runden folgt das nächste Level.

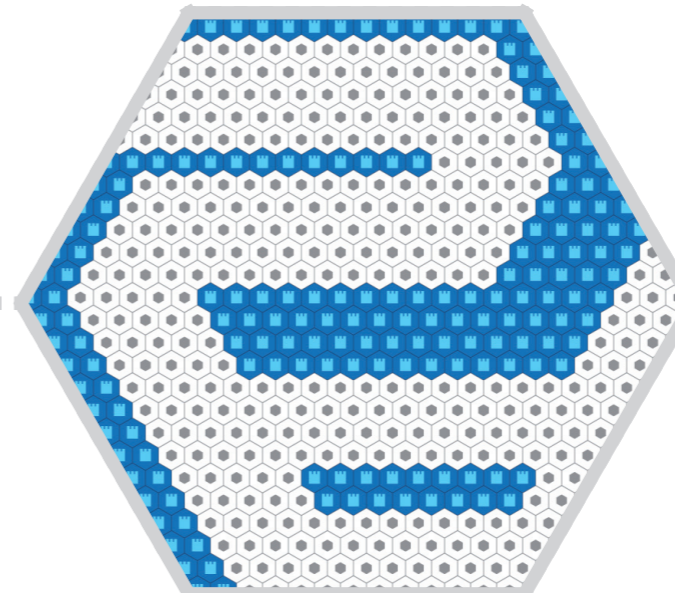
Ursprünglich entstand dieses VR-Spielprojekt bereits im Sommersemester 2016 unter dem Arbeitstitel „Vive Defender“. Seither wird die Weiterentwicklung primär durch das Kern-Team vorangetrieben, um fehlende Elemente im Spiel zu ergänzen und dieses so auf eine mögliche Veröffentlichung vorzubereiten.

PROGRAMMIERUNG



C# / .NET

Zur Erstellung von Programmcode für die Unity-Engine wurde die moderne Programmiersprache C# gewählt. Hierfür wurden anfangs entsprechende Codekonventionen definiert. Aufgrund der zahlreichen unterstützenden Funktionen kommt als Entwicklungsumgebung Microsoft Visual Studio zum Einsatz.



XML

Um das Leveldesign von Beginn an so einfach und flexibel wie möglich zu gestalten, wurde hierfür ein grafischer Level-Map-Editor implementiert. Die damit generierten Informationen werden anschließend im XML-Format gespeichert. Auch bestehende Level-Maps können erneut geöffnet und bearbeitet werden.



Photon

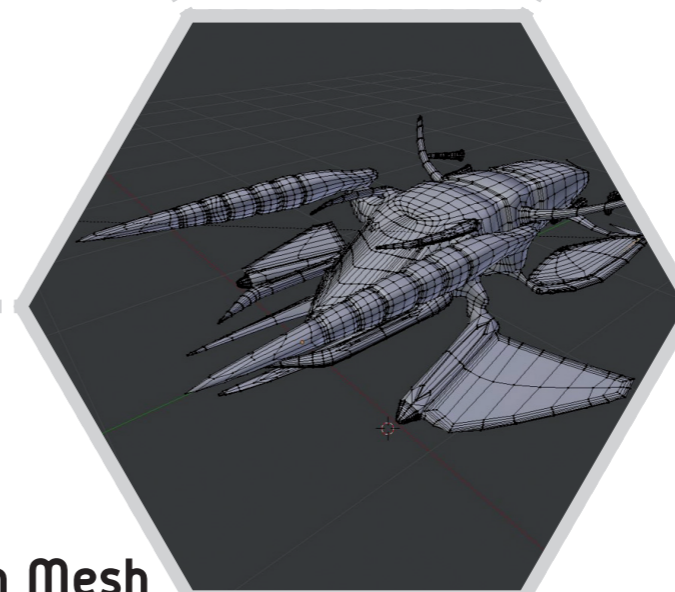
Bereits in einer frühen Entwicklungsphase wurde die Entscheidung getroffen, einen Multiplayer-Modus zu implementieren. Die Umsetzung erfolgte dann auf Basis der Photon-Engine, worüber alle relevanten Statusinformationen via Internet zwischen den einzelnen Clients ausgetauscht werden können.

3D-MODELLIERUNG



Concept Art

Am Anfang eines greifbaren Objekts im Spiel steht die Konzeptzeichnung. Sie dient dazu einen groben Eindruck zu erhalten. Während der Planungsphase wurden so erste Ideen digital erstellt. Daran anknüpfend konnten dann im Team gemeinsam Designentscheidungen für die weitere Umsetzung getroffen werden.



Polygon Mesh

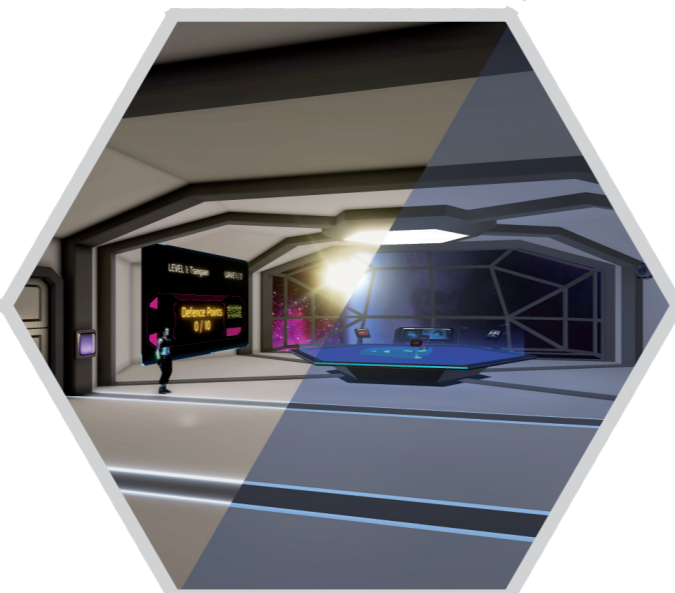
Auf Basis der zuvor erstellten Konzeptzeichnung wurde dann ein konkretes dreidimensionales Drahtgittermodell in Blender umgesetzt. In einem weiteren Schritt wurden die fertigen Flächen dann entsprechend vorbereitet, um das Modell am Ende korrekt mit einer Textur belegen zu können.



3D-Asset

Das fertige 3D-Modell wurde nun abschließend in Unity importiert und mit aufeinander abgestimmten Texturen sowie Shadern versehen. Bei der Erstellung sämtlicher grafischer Objekte lag stets ein besonderes Augenmerk auf der harmonischen Gesamterscheinung des Spiels.

GAMEPLAY



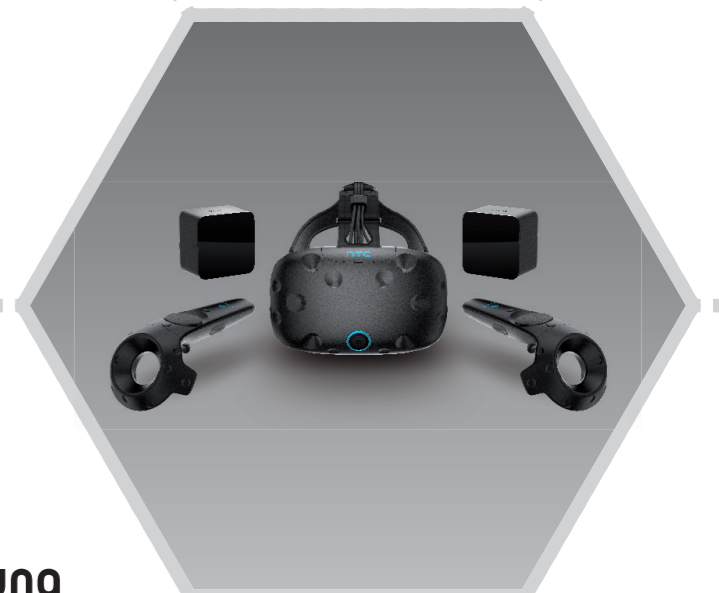
Grafik

Insbesondere bei der Entwicklung in VR ist die optische Darstellung im Spiel eine permanente Abwägung zwischen ansprechender Gestaltung und Systemperformance. Mit einem speziell für VR angepassten Post-Processing-Workflow ist es letzten Endes gelungen, ein immersives Spielerlebnis zu erzeugen.



Sound

Um auch akustisch für ein atmosphärisches Spielerlebnis zu sorgen, wurden verschiedene Soundtracks eigens für das Spiel komponiert. Gleichzeitig wurden alle Sprachausgaben der intelligenten Assistentin AIRA von einer erfahrenen Synchronsprecherin vertont und abschließend digital nachbearbeitet.



Steuerung

Bei der Entwicklung der Spielsteuerung wurde stets auf eine einfache und intuitive Handhabung geachtet. Insbesondere durch die Umsetzung in VR mussten an verschiedenen Stellen zusätzliche Überlegungen getroffen werden, da konventionelle Steuerungsmechaniken zu meist nicht übernommen werden können.



INFO@AIRA-VR.COM
WWW.AIRA-VR.COM