VIVE DEFENDER

VIRTUAL REALITY

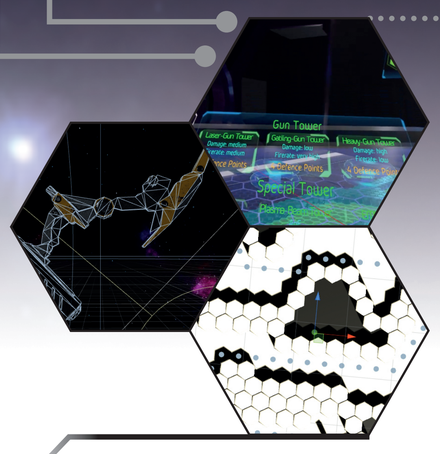
STEAM VR

unity

HOCHSCHULE DER MEDIEN

Institut Für Games

HOW WE WORKED



ABOUT

Vive Defender ist eine Mischung aus Tower Defense und First Person Shooter, umgesetzt in Virtual Reality (VR). Der Spieler nimmt dabei in der sogenannten Egoperspektive am Spielgeschehen teil. Einfluss übt er dabei über die Controller der HTC Vive aus.

Der eingebaute Mehrspielermodus gibt dem Spieler über das Internet die Möglichkeit mit Freunden und Gleichgesinnten das Spiel zu genießen - europaweit. Dieser Modus wurde erstmalig in einem VR-Titel der Hochschule der Medien umgesetzt.

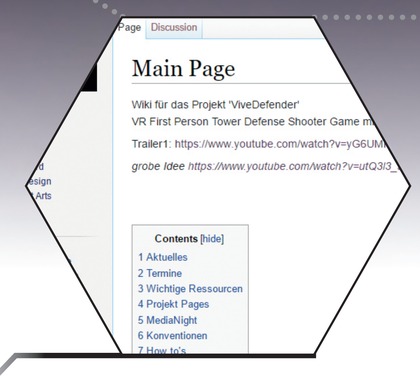
Sowohl einzeln als auch im Team, finden sich die Spieler auf einem futuristischen Raumschiff wieder, in welchem sie ihre Strategie planen und Gegenstände kaufen können. Auf einem holografischen Tisch findet der Spieler eine Ansicht des aktuellen Levels wieder, auf der er seine Verteidigung aufbauen und erweitern kann.

Nach Abschluss der Taktikphase wird der Spieler in das eigentliche Spiellevel teleportiert und ist nun in der Lage mit verschiedenen Waffen, wie beispielweise einem Bogen oder Granaten, die Gegnerwellen selbst zu eliminieren. Gerade hier dient die Steuerung als wichtiges Element für unser Gameplay. Ist der Spieler erfolgreich, so wird er dafür mit Münzen für den Ingame-Shop belohnt.

Bereits im letzten Semester starteten wir mit der Planung. Ideen für Spiele aus verschiedenen Genres wurden vorgeschlagen und abgewogen. Letzten Endes entschieden wir uns für das bekannte Konzept und verfeinerten dies zunehmend. Wir entwarfen ein Gamedesign-Wiki in dem wir alle wichtigen Entscheidungen festhielten und überlegten uns zudem eine Architektur, sowohl für die Software als auch für den Arbeitsablauf.

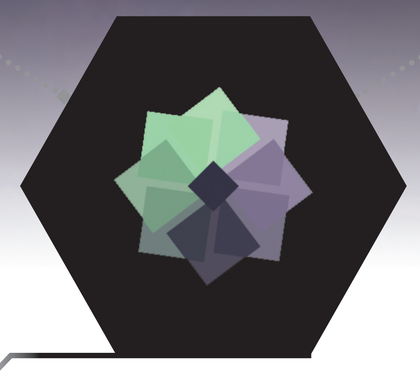
Anfänglich rechneten wir mit deutlich mehr Zulauf im Artist-Bereich, kalkulierten aber eine gewisse Flexibilität mit ein. Somit war es uns möglich den Aufwand ohne großen Schaden für das Gesamtprojekt zu reduzieren.

Auch im kommenden Sommersemester 2017 arbeiten wir an diesem Projekt weiter. Dabei werden wir den Fokus auf zusätzliche Funktionen und ein neues Design legen.



WIKI

Alle wichtigen Daten zu Design-Entscheidungen, Terminen, Storyboard und Architektur werden hier niedergeschrieben. Zusätzlichen dient das Wiki als zentrale Datenbank für nützliche Inhalte von Drittanbietern, wie beispielsweise passende Tutorials für die Skripter.



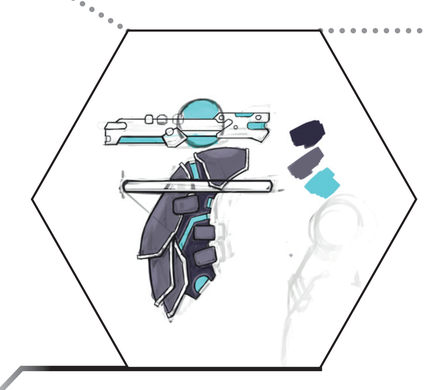
SCRUM

Als Vorgehensmodell auf Projektmanagement-Ebene haben wir uns für Scrum entschieden. Über die Webplattform Taiga (taiga.io) definierten wir unsere Sprints, User Stories und Tasks, die sich die Projektmitglieder als Aufgabenpakete selektieren konnten. Unter Abweichung der Standardmethode nutzten wir wöchentliche Scrum-Meetings als Ersatz für das Daily Scrum Meeting.

UNITY

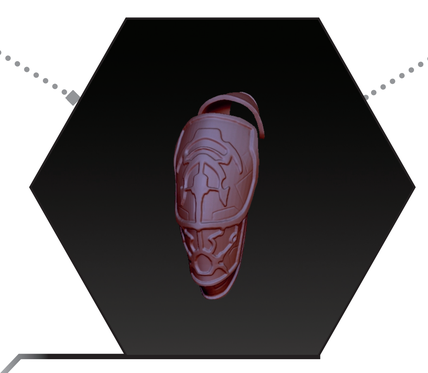
Innerhalb der Umgebung der Unity-Engine vereinten wir jeglichen Inhalt, vom 3D-Mesh über Sound bis hin zur Programmlogik des Games.

3D-CONTENT



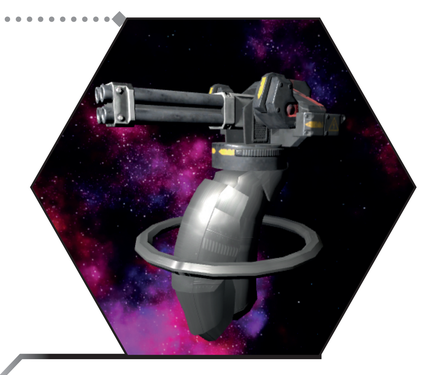
CONCEPT ARTS

Vor jedem greifbaren Inhalt steht ein Konzept. Bereits früh in der Planungsphase entwerfen wir unsere Ideen auf (digitalem) Papier. Auf Basis dieser Konzepte wurden Besprechungen innerhalb des Management-Teams durchgeführt. Bestand ein Konzept diese Prüfung, so wurde es zur Modellierung freigegeben, ansonsten wurde nachgearbeitet.



MODEL

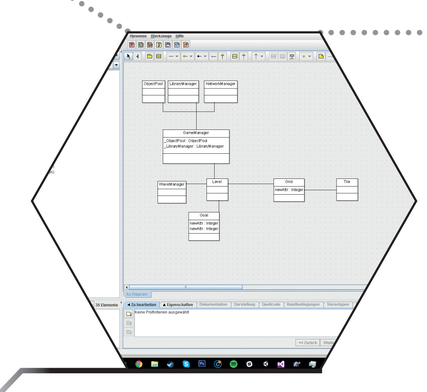
Auf Basis der Concept Art wurde von unserem 3D-Artist ein passenden Modell mit Hilfe von Blender entwickelt. Dieses Modell wird anschließend mit einer Textur versehen und in Unity im fbx Format exportiert.



FINAL MESH

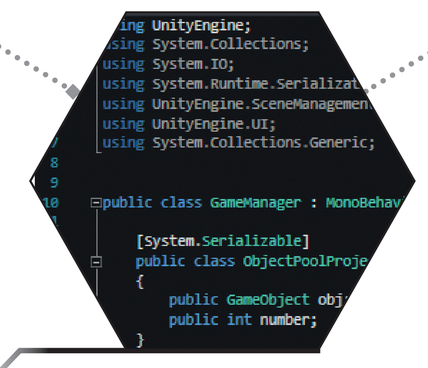
Das fertige 3D-Modell wird im Anschluss in das Spiellevel eingebaut und von den Programmierern zum Leben erweckt. Wichtig war uns hierbei auf humanoide Eigenschaften zu verzichten, um einen möglichst mechanischen Gesamtlook zu gewährleisten und aufwändige Animationen zu verhindern.

SCRIPTING



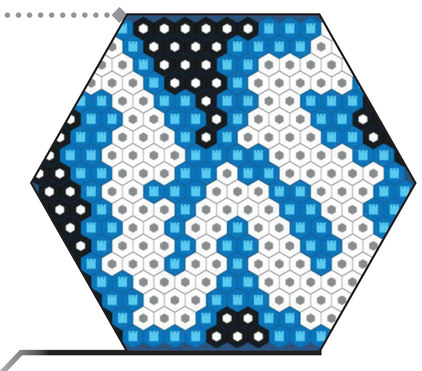
UML

Am Anfang steht die Architektur. Hierbei überlegten wir uns, welche Strukturen und Kommunikationswege zwischen verschiedenen Programmklassen notwendig sind, um einen sauberen und performanten Spielfluss zu gewährleisten. Diese bildeten wir über sogenannte UML Diagramme ab.



C#

Als Programmierumgebung haben wir uns für Microsofts Visual Studio 2015 entschlossen. Mit Hilfe der Programmiersprache C# entstanden so alle Skripte im Spiel, und somit auch die Funktionalität.



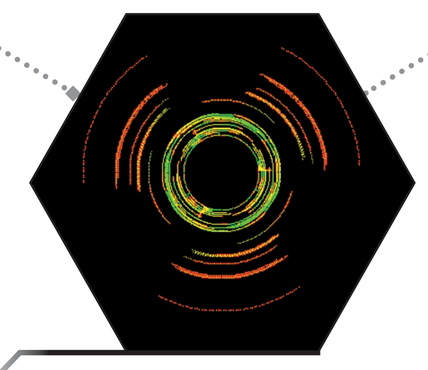
XML

Um das Level Design, Gegnerwellen und zukünftige Kampagnen möglichst einfach zu gestalten, programmierten wir uns eigene Tools. Mit deren Hilfe ist es uns möglich ohne großen Aufwand z. B. neue Karten zu erstellen, diese als Vorlage abzuspeichern und später automatisch eine 3D-Welt daraus zu generieren. Der Datenimport und -export findet hierbei über XML statt.

GAMEPLAY

CONTROLLS

Die Interaktion des Spielers basiert auf den Controllern der HTC Vive. Diese ermöglichen eine möglichst interessante Steuerung, wie das Spannen des Bogens. Durch haptisches und visuelles Feedback unterstützen wir den Spieler in der Steuerung und Koordination in der Spielwelt.



SOUND

Für die passende Atmosphäre sorgen Sounds aus der HdM-Soundbibliothek für die wir die Lizenz erhalten haben. Gewählt wurden futuristische und elektrische Klänge, die die Spielumgebung stark unterstützen können.

VR

Immer mehr Spiele erscheinen in VR oder werden dafür zurechtgeschnitten. Durch diese Umgebung erhält der Spieler eine ganz neue Perspektive auf das Geschehen im Spiel und schafft dadurch eine einzigartige Atmosphäre. Aus unserer Entwicklersicht stellt ein VR-Game zusätzliche Anforderungen an das gesamte Team, da eine deutlich höhere Performance notwendig ist.

VIVE DEFENDER

VIRTUAL REALITY

WWW.VIVEDEFENDER.COM

https://www.youtube.com/watch?v=yG6UMkcCD3k

https://www.facebook.com/vivedefender