
Gruppe Jenga

Patrick Otte,
Maximilian Fornaçon,
Johann Gerhardt,
Marian Geißler

VR Jenga

27. März 2020

Motivation

Ziel des Projekts VR-Jenga mit dem Arbeitsnamen “VRCKELTURM” ist es, ein klassisches Geschicklichkeitsspiel in einer virtuellen Realität zu simulieren. Die Simulation basiert auf dem simplen Spielkonzept, 60 quaderförmige Bausteine möglichst hoch aufeinander zu stapeln und dabei zu vermeiden, dass der resultierende Turm umgestoßen wird oder einstürzt.

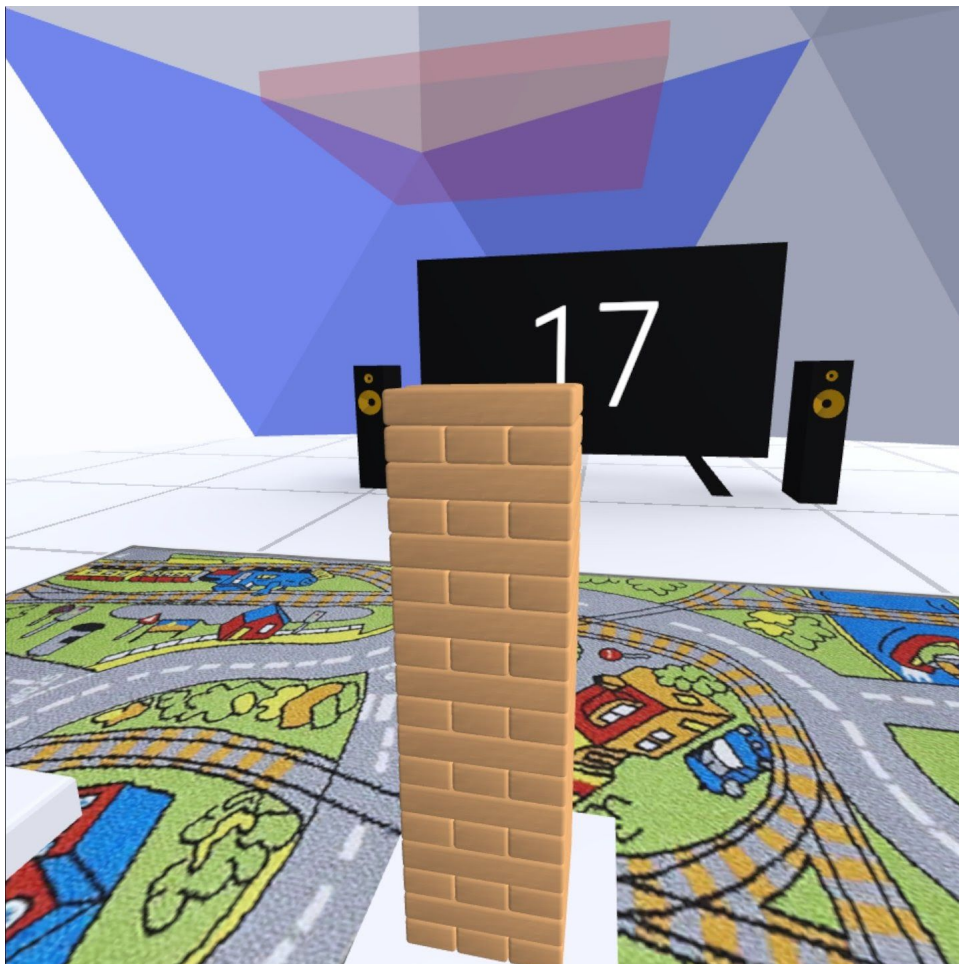
Erweitert wurde dieses Konzept von der Projektgruppe, durch mehrere “Sandbox”-Level, die zum experimentieren und spielen einladen. Zum einen wurde mit unterschiedlichen Materialeigenschaften der Blöcke gearbeitet, darüber hinaus kann mit unterschiedlichen Gegenständen hantiert werden, um Blöcke aus dem Turm zu befördern.

Alleinstellungsmerkmale

1. Vollständige Jenga simulation

- a. Hauptmerkmal der Anwendung ist die simulation eines Jenga-Spiels in einer virtuellen Realität. Dazu gehören neben der Interaktion mit den Steinen durch die Controller, auch physikalische Eigenschaften, wie das kippen des Turmes oder unterschiedliche Reibungswiderstände beim Entfernen einzelner Blöcke.

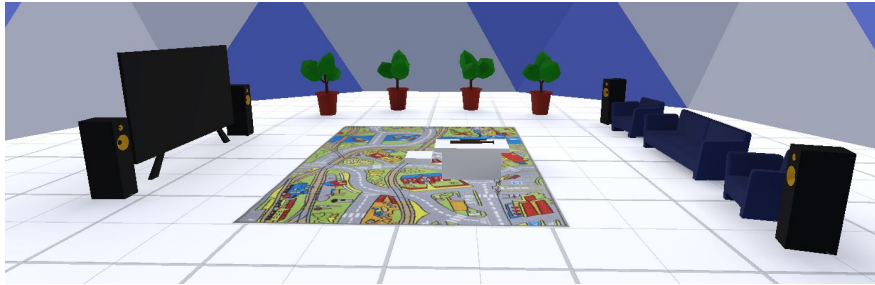
Die Steine haben genau wie im echten Leben nicht zu 100% die gleichen Größen. Dadurch wird erreicht, dass einige Steine sehr leicht herauszuziehen sind und andere völlig feststecken.



2. Spielumgebung “Virtuelles Wohnzimmer”

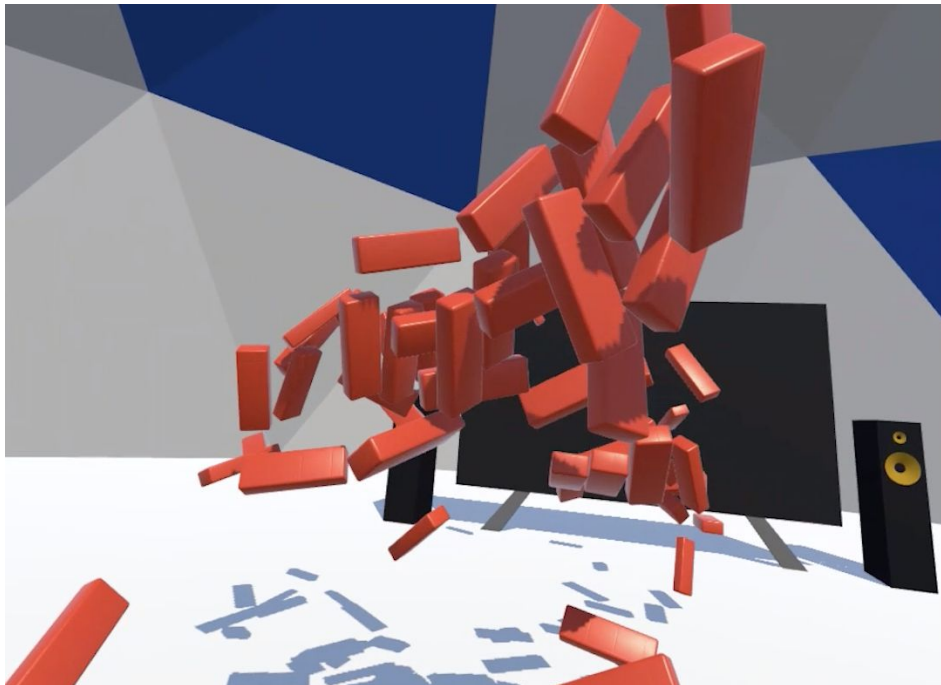
- a. Um das Interesse des Nutzers zu wecken und gleichzeitig “Motionsickness” vorzubeugen, wurden in der VR-Umgebung einige Gegenstände platziert, die eine “Wohnzimmer”-Atmosphäre erzeugen. Orientiert hat sich die Projektgruppe dabei an dem Hauptmenü der Oculus Quest.

Bild von VR-Wohnzimmer einfügen ToDo:MAX



3. Experimentelle Modi

- a. Die Simulation soll zum Ausprobieren einladen, besonders dafür geeignet ist der Modus, in dem die roten Spielsteine verwendet werden und diese wie schwerelos in der Szene umherfliegen.



4. Zeitmessung

- a. Um dem Nutzer eine befriedigende Spielerfahrung zu bieten, hat sich die Projektgruppe für die Integration eines Timers in der Szene entschieden. Dieser zählt die Zeit ab Spielbeginn und wird bei Einsturz des Turms oder bei Erreichen einer bestimmten Höhe angehalten und ist permanent sichtbar.



Meilensteine

Herausforderungen und Schwierigkeiten bei der Realisierung

- **Einbinden der Controller in die Szene**

Zu Beginn hatten wir uns als Ziel gesetzt für VR-Interaktion die VRTK Bibliothek zu verwenden, da diese einen VR-Simulator mitbringt, wodurch theoretisch auch die Entwicklung für diejenigen Teammitglieder, die gerade die Oculus Quest nicht vor Ort hatten, ermöglicht werden sollte. Außerdem ist VRTK mit verschiedenen VR-Brillen Herstellern kompatibel und wir so eine Cross-Plattform Anwendung erstellen wollten.

Leider hat sich herausgestellt, dass VRTK in der Version 3 sowohl von der Codebase als auch von der Dokumentation her sehr unübersichtlich ist und nicht mehr gepflegt wird, da gerade Version 4 in der Entwicklung ist. Version 4 war aber noch nicht ausgereift genug um sie hierfür zu verwenden.

Letztendlich haben wir beschlossen uns komplett von VRTK zu trennen und allein auf die Oculus Integration zu setzen, was vieles erleichtert hat. Unser Spiel ist dadurch aber nur auf der Oculus Quest spielbar.

- **Towerbuilder**

Das Towerbuilderscript generiert den Turm variabel aus Eingabedaten und Zufallsparametern. So ist es möglich einfach neue Level zu bauen, in denen Schwierigkeit und Spielgefühl stark variieren.

- **Tracking der Spielmenüs an das Headset (nicht in Finaler Version)**

Ziel ist es, die Menüs dem Spieler als Vollbild anzuzeigen, unabhängig von der aktuellen Position des Spielerheadsets, so dass entsprechende Menü-Interaktionen zu jeder Zeit möglich sind. Im ersten Versuch wurde dies über das VR-Toolkit versucht einzubinden, in dem die Kamera der Szene dem "Headsetfollower" des Toolkits zugeordnet wird. Leider führte dies zu Problemen bei der Interaktion (es wurde nicht der komplette Sichtbereich mit dem Menü bedeckt, sondern nur ein Teil) und der Erweiterung der Menüs (zahlreiche Neuordnungen der "Canvas"). Letzten Endes wurde dies dann über die direkte Schnittstelle der Oculus Brille gelöst (<https://developer.oculus.com/documentation/unity/>). In der finalen Version wurde diese Funktion jedoch verworfen, da die Inhalte der Menüs nun auf dem Fernseher in der Szene angezeigt werden, wesentlich intuitiver zu bedienen sind und besser in das Gesamtkonzept passen.

Zuordnung der Aufgaben

Patrick Otte

Vorkenntnisse

- keine Erfahrung in der Spieleentwicklung
- Programmiererfahrungen in verschiedenen Sprachen
- Git/Jira Erfahrung
- Erfahrung im Video- / Audioschnitt

Lernerfahrungen:

- Unity Grundlagen (Aufbau der IDE, Prefabs nutzen, Erstellen und Nutzen von Scripts,...)
- C# Grundlagen
- Genereller Einblick in den Erstellungsworkflow einer (Unity) VR Anwendung

Aufgabenbereich: Scripting & Video

- generelles Einarbeiten in Unity **(10h)**
- First Person Controller (Siehe PlayerLook.cs, PlayerMove.cs) **(8h)**
 - Obsolet da VRTK Integration
 - Auch obsolet da doppelter Entwickleraufwand
- Soundkonzept **(4h inkl. Erstellung eigener Sounds)**
 - Konzept, Auswahl und teilweise eigenes Erstellen von Sounds
- Highlighting der Bricks bei Auswahl **(4h)**
- Trailer und Live-Preview **(6h)**
 - Live-Preview via scrpcy
 - github.com/Genymobile/scrpcy

Maximilian Fornaçon

Aufgabenbereich: Projektleitung, Koordination und Entwicklung

Vorkenntnisse:

- keinerlei Erfahrung mit Entwicklung von Spielen
- Programmiererfahrungen in verschiedenen Sprachen
- Erfahrung mit Git

Lernerfahrungen:

- Zusammenarbeit im Team mit Ticketsystem und Git Repository
 - Lösen von dadurch und von Unity bedingter Mergekonflikte
- Arbeiten mit der Unity Physikengine
- Erstellen von Assets, Prefabs, C#-Scripten
- Einfaches Anfassen der Spielsteine mit den Händen macht das Spielen sehr schwierig, da Steine keine Masse für den Spieler besitzen und so bei der kleinsten Bewegung der Turm zerstört wird
 - Lösung: Zauberstab = fehlertolleranteres Interaktionselement

Aufgaben im Projekt

- Zentrale Versionskontrolle **(5h)**
 - Aufsicht über das Git Repository und das Ticketsystem: Jira
 - Anlaufstelle für Probleme mit Mergeconflicts
- Integration VR-Libraries in Unity **(10h)**
 - Oculus Unity Integration
 - Controller und Hände einbinden
 - Grabbing
 - VRTK (später wieder entfernt)
- Pointer für die Interaktion mit den Menüs **(6h)**
 - Pointer auf Basis von RayCasting implementiert
- Implementierung eines GameManagers, der für die Spielzustände (Gewonnen, Verloren) zuständig ist und das User Interface updatet **(5h)**
- Implementierung der Lose- und Win-Conditions **(4h)**
- Interaktionselement: "Der Zauberstab" **(7h)**
 - ermöglicht es einfacher Steine aus dem Turm herauszuziehen

Johann Gerhardt

Aufgabenbereich: Spielsteine, Modelling & Unity

- Einarbeitung in Unity **(5h)**
- Brick Konzepte (verschiedene Bricks) **(4h)**
 - Holz (Standard), Gold (schwer, etwas glatter), Eis (rutschig), Wasserstoff (schwerelos), Gummi (Reibung, „bouncy“)
- Physik: **(3h)**
 - Performance optimiert und daher unrealistisch und instabil

-
- Anpassung der Einstellungen
 - Testing
 - Towerbuilder **(5h)**
 - Funktion, die ggb. Parameter zur Turmgenerierung verarbeitet – Randomisierung
 - Testing
 - Aufbau der Spielszene (Modellierung von Raum mit Gegenständen und Anpassung der Beleuchtung) **(8h)**
 - Implementierung des Bolzenschussgeräts **(8h)**
 - Wurde wegen der Ausgangssperre nicht fertiggestellt und befindet sich jetzt im Ordner “Unused Code”

Wissensstand vor dem Projekt:

- kleine Modellierungs- und Animationserfahrungen mit Blender
- Programmierkenntnisse
- keinerlei Erfahrung mit Unity oder anderen Spiele-Engines

Lernerfahrungen:

- Umgang mit Unity (grundlegender Aufbau, Prefabs erstellen und nutzen, Beleuchtungs- und Physicseinstellungen, Einbinden von Scripts, Imports von Blender zu Unity,...)
- Vertiefung von Modellierungsfähigkeiten und Programmierkenntnissen

Marian Geißler

Wissensstand vor dem Projekt:

- Programmierkenntnisse
- keinerlei Erfahrung mit Unity oder anderen Spiele-Engines

Aufgabenbereich: Menü & Nutzereingabe

- Einarbeitung in Unity **(8h)**
- Menü Main **(8h)**
 - Unity „TextMash Pro“ – Plug in
 - Tracking der Menüs an Sichtfeld
 - Trigger für Buttons (Parameter „TowerBuilder“)
 - Import von Fonts
- Towerbuilder Ansteuern über Menü **(3h)**

-
- Menü In-Game **(6h)**
 - timeScale = 0
 - OVRInput, da VRTK fehlerhaft
 - VRTK Pointer **(4h)**
 - Dokumentation **(1h)**

Bedienung

Zum Starten des Spiels unter Bibliothek → unbekannte Quellen → VRCKELTURM öffnen. Nach einer kurzen Ladezeit befindet man sich in einem Zimmer mit einem Fernseher auf dem das Hauptmenü zu sehen ist. Mit dem linken oder rechten Zeigefingertrigger kann der Pointer gewechselt werden und das Menü bedient werden. Hat man ein Level ausgewählt, kann man das Menü auf dem Fernseher wieder aufrufen (und damit das Spiel pausieren), in dem man die Menütaste drückt. Diese ist die Taste mit dem Hamburgermenü-Icon auf dem linken Controller.

Levelauswahl:

- **Easy:** Enthält einen Turm bestehend aus Holzsteinen. Mit diesem kann man gut üben, da er verglichen mit den anderen relativ stabil ist und keine unbekannten Steinarten enthalten sind.
- **Mixed:** Dieser Turm besteht zum Großteil aus Holz, enthält aber auch Steine aller anderen Sorten. Dadurch kommt Abwechslung ins Spiel und die Schwierigkeit wird erhöht. Die Sorten sind:
 - Wood: klassischer, normaler Stein
 - Ice: sehr rutschig
 - Rubber: sehr klebrig und bouncy
 - Helium: geringe Masse und schwerelos
 - Gold: sehr schwer
- **Zero Gravity:** Ohne Ziel. Ein Turm aus Helium Steinen. Der Turm ist nur zum Zerstören und Bestaunen da.
- **Slippery:** Der Turm enthält Holz-, Gold- und Ice-Steine. Dadurch wird der Turm sehr rutschig, wodurch das Herausziehen von Blöcken leichter ist, der Turm allerdings auch instabiler wird.
- **Minigolf:** Ohne Ziel. Die Steine können mit dem Schläger durch die "Tore" geschossen werden.

Spielmechaniken:

Wenn die oberste, vollständige Ebene eine bestimmte Höhe (angedeutete rote Plattform) erreicht, gewinnt man.

Sobald zwei Steine, gleichzeitig, auf dem Boden liegen, hat man verloren.

Sollte ein Stein runterfallen hat man noch die Möglichkeit diesen wieder aufzuheben.

Man hat mehrere Möglichkeiten mit dem Turm zu interagieren:

- Mit den Händen: Im Normalzustand haben die Hände keine Hitbox, daher kann man durch den Turm durchfassen. Formt man allerdings einen Zeigefinger, oder eine Faust, kollidieren die Steine mit der Hand und die Steine lassen sich bewegen. Wenn man die Mittelfinger-Taste drückt, kann man die Steine greifen und mit loslassen der Taste, auch die Steine wieder loslassen.
- Baseballschläger und Golfschläger: Diese kann man greifen um dann mit ihnen Blöcke weg zu schlagen.
- Zauberstab: Diesen kann man greifen und einen Stein auswählen, indem man einen Stein mit der Zauberstabspitze berührt. Berührte Steine erscheinen grün umrandet. Wenn man nun die Zeigefingertaste gedrückt hält, wird der Stein an den Zauberstab (mit einer Kraft, die bei großer Entfernung immer größer wird) gezogen. Daher kann man sich mit dem Zauberstab entfernen und der Stein wird hinterhergezogen. Wenn nichts passiert, steckt der Stein eventuell zu sehr fest.
- Bolzenschussgerät: diesen kann man auch greifen und mit dem Zeigefingertrigger auslösen. Hiermit lassen sich einzelne Steine aus dem Turm schießen. Zum Nachladen mit der einer Faust den Bolzen wieder reindrücken. (Nicht fertiggestellt)

Tipp: Die Steine haben genau wie im echten Leben nicht zu 100% die gleichen Größen. Es kann also gut sein, dass sich nicht jeder Stein leicht herausziehen lässt, und man sich lieber einen anderen Stein suchen sollte.

Soundkonzept

Benötigte Sounds

Menüs

Bestätigung/Ok

Abbrechen/Cancel

Hover

In-Game

Start Sound bei Beginn des Spiels

Hintergrundmusik

Fail Sound

Success Sound

Implementierung

Menüs

Unser Soundkonzept sieht vor, bei jedem Hover über einen Button ein leises Klickgeräusch, erzeugt und aufgenommen mit Jenga-Steinen, zu verwenden. Alle weiteren Menüpunkte wie der Start, die Bestätigung (Ok/Confirm Button) sowie das Abbrechen von Aktionen werden mit dem vom Rapper Lil Wayne typischen Ad-libs belegt.

Diese sind hier zu sehen:

<https://www.youtube.com/watch?v=sSkRdO9K1og>

In-Game

Mit dem Starten des Spiels wird ein "Game-Start" Sound abgespielt. Dieser ist ein "Jenga" Ruf aus einer alten Jenga Werbung.

Nachdem dieser Sound abgespielt wurde wird die Hintergrund-Musik eingefaded. Hierbei haben wir darauf geachtet einen Sound zu finden, der auch nach Minuten nicht zu sehr störend ist. Dies hat sich als sehr schwierig herausgestellt, da auch hier jeder seinen eigenen Geschmack hat und wir verhindern wollten, dass die Musik schnell nervig wird. Wir haben uns letzten Endes darauf geeinigt den Song Buttercup zu nehmen:

<https://www.youtube.com/watch?v=NYhl4poDFJ4>

Sollte bei einem Zug der Turm einstürzen, wird ein Fail-Sound abgespielt.

Sollte man gewinnen wird ein Success Sound eingespielt.

Update:

Da am Ende der Entwicklung nicht mehr Zeit für das Einbinden einiger Sounds war, haben wir sie weggelassen. Das betrifft hier folgende Sounds: Bestätigung/Ok, Abbrechen/Cancel, Hover. Außerdem gab es die Idee eine randomisierte Playlist für Hintergrundmusiken zu verwenden, da nicht jeder Musikgeschmack getroffen werden konnte, doch auch hier fehlte ein wenig die Zeit.

Ressourcen

Folgende Ressourcen wurden im Projekt verwendet.

TextMesh Pro

Plug-In zum Bearbeiten und Integrieren von Schriften in Unity. Das Tool wurde für die Gestaltung der Menüs verwendet.

<https://assetstore.unity.com/packages/essentials/beta-projects/textmesh-pro-84126>

Roboto Condensed

Font von Google, unter Verwendung der Apache License Version 2.0.

<https://fonts.google.com/specimen/Roboto+Condensed>

Sounds

Jenga Steine und Aufnahmegerät

<https://www.youtube.com/watch?v=sSkRdO9K1og>

Quellen und Inspirationen

Unity Dokumentation

docs.unity3d.com/

VR Toolkit Dokumentation

<https://vrtoolkit.readme.io/docs>

Dokumentation der Oculus Integration

<https://developer.oculus.com/downloads/package/unity-integration/>

Blender Dokumentation

<https://docs.blender.org/manual/en/latest/index.html>

100 Days of Unity VR Development

<https://codingchronicles.com/100-days-of-unity-vr-development>

First Person Controller

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

<https://www.youtube.com/watch?v=n-KX8AeGK7E>

Inspiration

<https://www.youtube.com/watch?v=DsNd62ime98>