

Modul 150

# Jenga Ultimate

Dokumentation

by

Simon Krieger

&

David Zweili

## Inhalt des Dokumentes

1	Management Summary (Kurzfassung) .....	3
1.1	Ausgangssituation .....	3
1.2	Umsetzung .....	3
1.3	Ergebnis .....	3
2	Bericht Teil 1: Umfeld und Ablauf.....	4
2.1	Einleitung .....	4
2.2	Abgrenzungen .....	4
2.3	Zeitplan .....	5
2.4	Arbeitsjournal.....	6
2.4.1	12.11.2020 – 18.11.2020.....	6
2.4.2	19.11.2020 – 25.11.2020.....	6
2.4.3	26.11.2020 – 01.12.2020.....	7
2.4.4	03.12.2020 – 09.12.2020.....	7
2.4.5	10.12.2020 – 16.12.2020.....	8
2.4.6	17.12.2020 – 23.12.2020.....	8
2.4.7	24.12.2020 – 30.12.2020.....	9
2.4.8	31.12.2020 – 06.01.2021.....	9
3	Bericht Teil 2: Projekt .....	10
3.1	Produkt .....	10
3.2	Aufbau .....	10
3.2.1	GameManager .....	10
3.2.2	UIController .....	10
3.2.3	PlayerMovement .....	10
3.2.4	AudioManager .....	11
3.3	Projektvorgehen.....	11
3.3.1	Scrum .....	11
3.3.2	Konzept .....	11
3.4	Testfälle.....	12
3.5	Testprotokoll .....	14
4	Schlussreflexion .....	14

# 1 Management Summary (Kurzfassung)

## 1.1 Ausgangssituation

Das Ziel des Projektes war ein Game, welches die zwei Komponenten Jenga und Ego-Shooter miteinander kombiniert und mit einem einfachen Punktesystem den Erfolg (vom Tisch katapultierte Klötze) darstellt.

Kernpunkte des Projektes sind die Implementation der Physik insbesondere der des Jenga-Turms, Programmierung des Players, Design und Animation der Spielwelt, Implementierung eines Userinterfaces mit Pause-Funktion und einer Continuous-Deployment Pipeline.

## 1.2 Umsetzung

Als Game-Engine wurde die Unity3D-Engine verwendet. Als Physikgrundlage wird die von Unity zur Verfügung gestellte Physik angewendet.

Bis auf die Jengasteine wurden hauptsächlich Assets aus dem Web verwendet. Trotzdem gab es auch da einige Herausforderungen, wie z.B. das saubere mappen der Playeranimationen auf den Player-Controller. Die Animationen werden über einen Animation-Controller gesteuert, welcher fließende Übergänge von dem einem in den anderen Animation-State ermöglicht.

Um das User-Interface umzusetzen entschlossen wir uns eine neue Technologie zu benutzen. Sie ist sehr von HTML/CSS/JS inspiriert und ist somit sehr flexibel. Allerdings handelt es sich um ein Preview-Package, welches noch nicht für die Produktion empfohlen wird. Ein paar Kinderkrankheiten wie z.B. unlösbare Warnungen mussten wir somit in Kauf nehmen.

Die Continuous-Deployment Pipeline wurde mit GitHub-Actions und einem vorkonfigurierten Docker-Container umgesetzt, auf welchem unsere Unity Version bereits installiert ist. Wird ein Commit getaggt, wird vom Stand dieses Commits ein Build getriggert und auf GitHub zum Download bereitgestellt.

## 1.3 Ergebnis

Als Ergebnis dieses Projektes liegt ein Egoshooter Spiel vor, indem mehrere Komponenten aus der Spieleentwicklung entwickelt und zusammengefügt wurden. Es handelt sich nicht um ein fertiges Spiel, sondern eher um ein Prototyp um das Gameplay und das Zusammenspiel der verschiedenen Komponenten zu testen.

## 2 Bericht Teil 1: Umfeld und Ablauf

### 2.1 Einleitung

Dieses Dokument ist der Bericht zur Probe-PA. Es wird im Laufe der Arbeit erstellt und ergänzt, um die Arbeit zu planen und zu dokumentieren. Da diese Probe-PA, anders als die richtige PA über mehrere Wochen verteilt und nicht 10 Tage am Stück ist, haben wir im Zeitplan und dem Arbeitsjournal jeweils den Zeitraum einer Woche zusammengefasst, da wir auch ausserhalb des jeweiligen Schultages am Projekt gearbeitet haben.

### 2.2 Abgrenzungen

Das Spiel wird mit der bereits vorhandenen Game-Engine Unity3D entwickelt. Die in Unity eingebaute Physikengine wird zum Berechnen der Physik der Spielwelt angewendet und wird nicht selbst entwickelt. Auch können Assets aus dem Web wie z.B. Animationen oder geriggte Gameobjekte verwendet werden, um die Spielwelt lebendiger zu gestalten.

## 2.3 Zeitplan

Geplanter Aufwand:

Effektiver Aufwand:

Aktivität	Geplant (h)	Effektiv (h)	12.11 – 18.11.20	19.11 – 25.11.20	26.11 – 02.12.20	03.12 – 09.12.20	10.12 – 16.12.20	17.12 – 23.12.20	24.12 – 30.12.20	31.12 – 06.01.21
Dokumentation / Arbeitsjournal	10	12								
Projekt initialisieren	3	3								
Continuous Deployment konfigurieren	3	12								
FPS Controller entwickeln	4	4								
Jenga Turm generieren	4	5								
Schiesslogik entwickeln	5	10								
User Interface entwickeln	6	7								
Jetpack entwickeln	3	3								
Texturen implementieren	3	3								
Player-Animationen implementieren	5	5								
Welt erstellen	4	10								
Audio implementieren	3	2								
Punktesystem entwickeln	5	6								
Mainmenu erstellen	3	3								
Finetuning / Bugfix	8	6								

## 2.4 Arbeitsjournal

### 2.4.1 12.11.2020 – 18.11.2020

Tätigkeiten	<ul style="list-style-type: none"> <li>• Definieren der Arbeit</li> <li>• Github-Repository mit neuem Unity-Projekt initialisieren.</li> <li>• Projektaufgaben definieren</li> </ul>
Erreichte Ziele	<ul style="list-style-type: none"> <li>• Grundidee des Projektes ist vorhanden</li> <li>• Aufgaben sind grundsätzlich definiert</li> <li>• Repository ist bereit für den Beginn zum Programmieren</li> </ul>
Probleme	Die Umsetzung des Projektes ist noch nicht klar definiert, da wir von der Gravity-Engine von Unity abhängig sind. Diese muss zuerst noch für unseren Anwendungsfall getestet werden.
Reflexion	Der erste Tag lief wie gewünscht, wir haben alle grundsätzlichen Vorbereitungen getroffen, um das Projekt zu starten.

### 2.4.2 19.11.2020 – 25.11.2020

Tätigkeiten	<ul style="list-style-type: none"> <li>• Continuous Deployment konfigurieren</li> <li>• Jenga Turm generieren</li> </ul>
Erreichte Ziele	<ul style="list-style-type: none"> <li>• Grundkenntnisse wie CD mit Github-Actions umgesetzt.</li> <li>• Jenga Turm kann aus einer Vorlage von 3 Jenga Blöcken Programmatisch aufgebaut werden. Per Knopfdruck kann ein gefallener Turm gelöscht und neu aufgebaut werden.</li> </ul>
Probleme	<p>Das Konfigurieren der CD-Pipeline ergab sich schwieriger als erwartet. Ein Build für Windows funktionierte, aber für den Web-Build schien der vorkonfigurierte Dockercontainer nicht optimiert zu sein.</p> <p>Auch die Physik im Zusammenhang des Jenga Turmes ergibt sich als problematisch. Der Turm ist sehr instabil und fällt bereits bei kleinen Krafteinflüssen um.</p>
Reflexion	Für die CD-Pipeline wurde bereits viel mehr Zeit beansprucht als geplant war. Hinzu haben wir erfahren, dass Unity allgemein den Web-Build in den nächsten Versionen nicht mehr supportet. Also haben wir uns entschieden uns auf den Windows-Build zu beschränken

## 2.4.3 26.11.2020 – 01.12.2020

Tätigkeiten	<ul style="list-style-type: none"> <li>• FPS-Controller entwickeln</li> <li>• Schiesslogik entwickeln</li> </ul>
Erreichte Ziele	<ul style="list-style-type: none"> <li>• Man kann sich dank dem FPS-Controller in der Ego-Perspektive über das Spielfeld bewegen. Der Prototyp eines Gewehres folgt dem Blick des Spielers</li> <li>• Über Mausklick können Projektile aus dem Gewehr abgefeuert werden, um Klötze des Turmes zu Beschießen.</li> </ul>
Probleme	Das Zusammenspiel der Projektile und des Turmes ist nicht ganz optimal. Es kann beispielsweise vorkommen, dass ein Projektil sich zwischen zwei Klötzen hindurchquetscht, was sehr unrealistisch wirkt und ein unfaires Spielgefühl vermittelt.
Reflexion	Uns war bewusst, dass es zu Problemen mit der Umsetzung des Jenga-Turmes in Zusammenhang mit der Physik kommen könnte. Fakt ist, dass die aktuelle Umsetzung mit den Projektilen kein robustes Ergebnis bietet. Wichtig ist, dass wir in solchen Situationen flexibel sind und alternative Herangehensweisen erkennen.

## 2.4.4 03.12.2020 – 09.12.2020

Tätigkeiten	<ul style="list-style-type: none"> <li>• CD-Pipeline finalisieren</li> <li>• Schiesslogik umbauen</li> </ul>
Erreichte Ziele	<ul style="list-style-type: none"> <li>• CD-Pipeline läuft nun stabil und generiert einen Windows-Build als output. Wird ein Commit getaggt, triggert dies den Build.</li> <li>• Die Schiesslogik wurde überarbeitet. Es gibt nun keine Projektile mehr. Beim vorherigen Schuss, wird neu über einen Raycast den geradeaus der Blickrichtung entlang liegende JengaBlock detektiert. Die Kraft wird dann direkt anhand der Blickrichtung des Players auf den Jengaklotz übertragen.</li> </ul>
Probleme	-
Reflexion	Die neue Lösung der Schiesslogik war ein voller Erfolg. Einzig der sichtbare Schuss fehlt jetzt. Aus Zeitgründen beschlossen wir aber dies erst gegen Ende, falls noch Zeit übrig ist zu implementieren, da es die Spielmechanik nicht direkt beeinflusst.

## 2.4.5 10.12.2020 – 16.12.2020

Tätigkeiten	<ul style="list-style-type: none"> <li>• User Interface entwickeln</li> <li>• Jetpack entwickeln</li> </ul>
Erreichte Ziele	<ul style="list-style-type: none"> <li>• Ein Simple User-Interface, welches das Game Pausieren lässt und verschiedene Seiten enthält wurde entwickelt.</li> <li>• Der FPS-Controller wurde mit einem Jetpack erweitert, um auch höhere Bereiche des Turmes gezielt zu erreichen.</li> </ul>
Probleme	-
Reflexion	Die Implementierung des User Interfaces verlief überraschend gut. Die Struktur des Interfaces ähnelt sehr stark der Struktur von HTML/CSS/JS. Da wir bereits Erfahrung mit Webentwicklung haben, konnten wir uns damit ziemlich schnell anfreunden.

## 2.4.6 17.12.2020 – 23.12.2020

Tätigkeiten	<ul style="list-style-type: none"> <li>• Texturen implementieren</li> <li>• Player-Animationen implementieren</li> </ul>
Erreichte Ziele	<ul style="list-style-type: none"> <li>• Durch ein Open-Source Projekt (Materialize) konnten wir Texturen anhand von Bildern erstellen. Dies haben wir gebraucht, um die Textur vom Jenga-Turm zu erstellen</li> <li>• Die Animationen stammen alle von Mixamo, dadurch konnten wir sehr schnell Animationen für unseren Charakter implementieren.</li> <li>• Ein weiteres Tool (Animation Rigging) benutzen wir für die Animationen zu verbessern. Dies ermöglichte uns die Hände an der Waffe zu befestigen, ohne die Animation gross zu stören.</li> </ul>
Probleme	Die Animationen waren leider nicht immer komplett passend aber für den Moment reicht die vollkommen aus. Wir werden die Animationen, falls noch Zeit übrig ist am Ende noch verbessern. Da die Animationen die Spielmechanik nicht gross beeinflusst.
Reflexion	Mixamo und Materialize haben es sehr einfach gemacht Animationen und Texturen für unser Spiel zu erstellen und Implementieren. Dadurch haben wir sehr viel Zeit gewonnen. Aber die Animationen brauchen noch ein wenig Finetuning.



## 2.4.7 24.12.2020 – 30.12.2020

Tätigkeiten	<ul style="list-style-type: none"> <li>• Welt erstellen</li> <li>• Audio implementieren</li> </ul>
Erreichte Ziele	<ul style="list-style-type: none"> <li>• Durch ein Asset-Pack mussten wir nicht selber 3D-Objekte erstellen und konnten direkt mit der Erstellung der Welt beginnen. Die Welt haben wir komplett selber erstellt mithilfe von den 3D-Objekten aus dem Asset-Pack.</li> <li>• Wir haben den Sound beim Schießen der Waffe hinzugefügt.</li> </ul>
Probleme	Wir haben Das Platzieren der 3D-Objekte sehr unterschätzt da diese Arbeit sehr Zeitintensiv war. Aber am Ende hat sich der Zeitaufwand sehr gelohnt.
Reflexion	Durch das Einsetzen eines Asset-Packs mussten wir keine 3D-Objekte selber erstellen und das hat sehr viel Zeit gespart. Aber wir haben das Platzieren der 3D-Objekte sehr unterschätzt. Nächstes Mal müssen wir mehr Zeit für die Welt-Erstellung berechnen.

## 2.4.8 31.12.2020 – 06.01.2021

Tätigkeiten	<ul style="list-style-type: none"> <li>• Dokumentation Finalisieren</li> <li>• Score-System / Save-System</li> <li>• Audiomanager</li> <li>• UI-Update</li> <li>• Bugfixing und Finetuning</li> </ul>
Erreichte Ziele	<ul style="list-style-type: none"> <li>• Dokumentation fertiggestellt</li> <li>• Score-System / Save-System implementiert. Highscore und die Einstellung werden nun gespeichert in einem Binary-File</li> <li>• Audiomanager, durch den Audiomanager haben wir alle Sounds zentral abgelegt und können jeden Sound einzeln ansteuern.</li> <li>• Wir haben das UI überarbeitet und haben entsprechende Elemente für die Audio-Einstellungen und für das Score-System erstellt.</li> <li>• Bugfixing und Finetuning</li> </ul>
Probleme	-
Reflexion	Die Dokumentation haben wir bisher ziemlich vernachlässigt, was uns nun ein wenig zum Verhängnis wurde. Immerhin haben wir genug Zeit eingeplant, um nicht groß unter Zeitdruck zu kommen. Es war uns aber auf jeden Fall eine Lehre, dass wir bei der Offiziellen PA die Dokumentation immer aktuell halten sollen. Durch das Save-System und dem Audiomanager haben wir auch automatisch die Code-Qualität um einiges verbessert. Dazu haben wir immer wieder ein kurzes Refactoring gemacht damit der Code immer besser optimiert wird.

## 3 Bericht Teil 2: Projekt

### 3.1 Produkt

Als Produkt haben wir ein Spielbares Game entwickelt. Der Spieler wird aus der Egoperspektive gesteuert. Ziel ist es, möglichst viele Punkte zu erzielen. Punkte werden erzielt, wenn man es schafft, durch Beschuss der Jenga Klötze, diese aus dem Spielfeld zu befördern. Das Spiel ist dann beendet, wenn der Turm in sich zu instabil wird und somit umkippt.

### 3.2 Aufbau

Wie bereits erwähnt, wurde das Projekt mit der Unity3D Engine entwickelt. In den nächsten Unterpunkten dieses Kapitels, werden wir die wichtigsten Komponenten des Spieles erläutern.

#### 3.2.1 GameManager

Der GameManager ist ein GameObjekt, welches mit einem Skript verknüpft ist. Durch ihn können diverse Mechanismen, welche in der Scene ausgeführt werden, gesteuert. Momentan ist seine Hauptaufgabe den Jengaturm zu Bauen. Für die Entwicklung bauten wir ein Feature ein, mit welchem sich der Turm per Knopfdruck neu aufbauen lässt.

Im Inspector kann über die BlockCount Variable noch angepasst werden, wie hoch der Turm gebaut werden soll. Dies benötigten wir vor allem, um die ideale Höhe des Turmes herauszufinden, in welcher er noch eine ausreichende Stabilität verfügte.

Zudem wird der GameManger für das Speichern des Scores und um die Verwaltung der Szenen benutzt.

#### 3.2.2 UIController

Wir haben jeweils einen UIController für die Menu-Szene und einen UiController für die Spiel-Szene.

Wie der Name schon vermuten lässt, wird über dieses GameObject das User Interface gesteuert. Auf diesem Objekt befindet sich der Komponent „UI Document“. Über diesen Komponenten werden einerseits die PanelSettings definiert. An diesen Settings mussten wir allerdings keine Anpassungen vornehmen und konnten alles auf Default belassen. Interessanter ist der Slot „Source Asset“ im selben Komponenten. Dies ist der HauptContainer, welcher die verschiedenen UI-Elemente zusammenfasst, welche in ähnlicher Form wie HTML/CSS. Aufgebaut sind.

Die Logik des UI's befindet sich auf dem UIController in einem Separaten C# Script. In der Start() Methode welche beim Start der Scene aufgerufen wird, werden als erstes alle Buttons über Querys als Variable initialisiert. In der Update() Methode wird dann in jedem Frame geprüft, ob der Pause Knopf gedrückt wurde. Trifft dies zu wird das Spiel pausiert und die PausePage wird eingeblendet.

#### 3.2.3 PlayerMovement

Die Steuerung des Spielers wird über das PlayerMovement.cs Script umgesetzt welches dem GameObject „First Person Player“ attached ist. In der Variable isGrounded wird festgehalten, ob der Spieler am Boden steht, und somit ein Sprung, bzw. das Aktivieren des Hovermodes möglich ist.



**Hinweis:** Das Property Time.deltatime wird bei allen Berechnungen multipliziert, welche mit Bewegung zu tun haben. Es beschreibt die Zeit, welche seit dem letzten Frame vergangen ist. Wenn diese Multiplikation nicht berücksichtigt würde, wären die Geschwindigkeiten abhängig von der Framerate des Spiels.

### 3.2.4 AudioManager

Im AudioManager sind die beiden Audiodateien hinterlegt, welche im Spiel verwendet werden. Der LaserShot und die BackgroundMusic. Das AudioManager.cs Script bietet Methoden an, um diese Sounds abzuspielen, bzw. die Lautstärke zu regeln. Die Methoden, um die Lautstärke zu regeln werden dann aus dem UIController aufgerufen.

## 3.3 Projektvorgehen

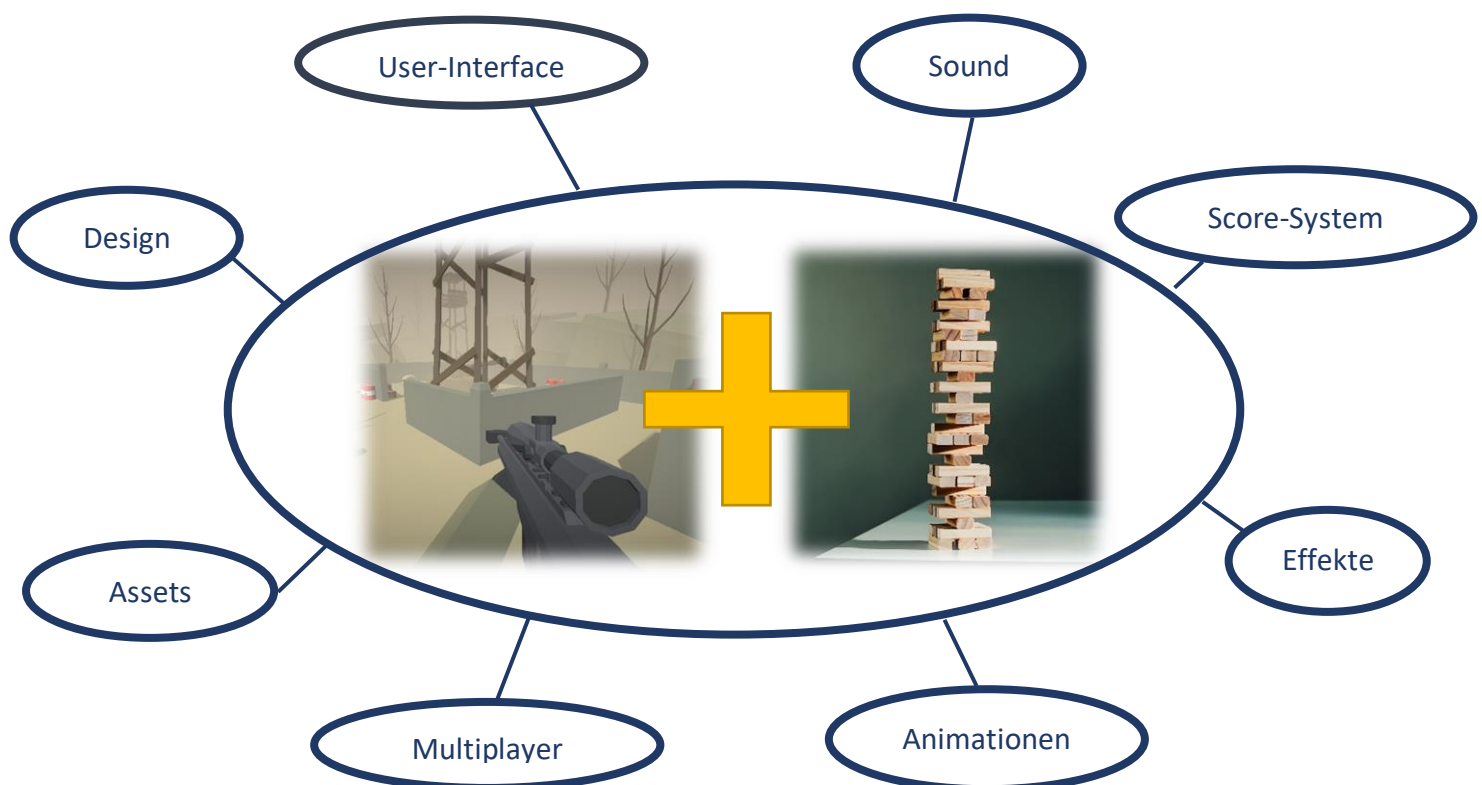
### 3.3.1 Scrum

Da wir bei der Vorgehensweise in unserem Projekt sehr agil handeln können müssen, haben wir uns bei der Projektmanagementmethode für die agile Scrum Methode entschieden. Somit können wir auf unerwartete Probleme mit Kreativität und grossem Spielraum reagieren. Um unsere Tasks zu managen haben wir mit dem CartTool auf Github gearbeitet.



### 3.3.2 Konzept

Das Konzept ist simpel. Die Grundidee des Projektes war, einen Jenga Turm mit Physik und einem Egoshooter Feeling zu kombinieren. Sobald dieser Grundstein gelegt ist, können wir das Game dann um beliebige Komponenten erweitern. Wir haben beide schon ein wenig mit Unity3D gearbeitet, aber möchten diese Möglichkeit nun nutzen, um neues zu lernen.



### 3.4 Testfälle

<b>Testfall 1: Horizontales Movement und Kamera</b>	
<b>Ausgangslage</b>	Das Spiel ist gestartet
<b>Schritte</b>	<ul style="list-style-type: none"> <li>▪ Mit den Knöpfen W/A/S/D in den verschiedenen Richtungen laufen.</li> <li>▪ Mit der Maus die Blickrichtung ändern.</li> <li>▪ Erneut in die verschiedenen Richtungen laufen.</li> </ul>
<b>Erwartetes Ergebnis</b>	<ul style="list-style-type: none"> <li>▪ Der Player bewegt sich beim Tastendruck in die gewünschte Richtung, wobei der Knopf W gerade aus der Blickrichtung ist.</li> <li>▪ Wird die Blickrichtung geändert, wird durch den Knopf W immer noch in die Blickrichtung navigiert.</li> <li>▪ Eine unsichtbare Wand verhindert das Herunterfallen vom Tisch</li> </ul>

<b>Testfall 2: Vertikales Movement mit Jetpack</b>	
<b>Ausgangslage</b>	Das Spiel ist gestartet.
<b>Schritte</b>	<ul style="list-style-type: none"> <li>▪ Mit dem Space-Knopf springen.</li> <li>▪ Dann in der Luft SHIFT gedrückt halten um mit dem Jetpack an Höhe zu gewinnen.</li> <li>▪ Während der Benutzung des Jetpacks die rechte Maustaste drücken um den Hover-Mode zu aktivieren und somit die aktuelle Höhe zu fixieren.</li> </ul>
<b>Erwartetes Ergebnis</b>	<ul style="list-style-type: none"> <li>▪ Während einem Sprung kann mit dem Space-Knopf nicht erneut gesprungen werden.</li> <li>▪ Wenn man sich am Boden befindet kann das Jetpack nicht aktiviert werden.</li> <li>▪ Im Hover-Mode ist die vertikale Bewegung deaktiviert.</li> </ul>

<b>Testfall 3: Pause Menu</b>	
<b>Ausgangslage</b>	Das Spiel ist gestartet.
<b>Schritte</b>	<ul style="list-style-type: none"> <li>▪ Über P oder ESC das Spiel pausieren.</li> </ul>
<b>Erwartetes Ergebnis</b>	<ul style="list-style-type: none"> <li>▪ Wenn das Spiel pausiert ist, kann der Spieler nicht bewegt werden.</li> <li>▪ Auch die Physik des Jenga Turmes ist pausiert.</li> </ul>

<b>Testfall 4: Score</b>	
<b>Ausgangslage</b>	Das Spiel ist gestartet.
<b>Schritte</b>	<ul style="list-style-type: none"> <li>▪ Einen Block durch Schiessen aus dem Spielfeld katapultieren.</li> </ul>
<b>Erwartetes Ergebnis</b>	<ul style="list-style-type: none"> <li>▪ Ein Block, welcher unter das Niveau vom Spielfeld fällt, wird aus der Scene entfernt und der Score steigt um 10 Punkte</li> </ul>

<b>Testfall 5:</b>	<b>Schiessen</b>
<b>Ausgangslage</b>	Das Spiel ist gestartet.
<b>Schritte</b>	<ul style="list-style-type: none"> <li>▪ Auf einen Jenga Block zielen</li> <li>▪ Mit linker Maustaste schießen.</li> </ul>
<b>Erwartetes Ergebnis</b>	<ul style="list-style-type: none"> <li>▪ Der Block wird in die gewünschte Richtung wegkatapultiert.</li> <li>▪ Die Obersten drei Klotzreihen und die Untersten können nicht durch Schiessen bewegt werden</li> </ul>

<b>Testfall 6:</b>	<b>Game Over</b>
<b>Ausgangslage</b>	Das Spiel ist gestartet.
<b>Schritte</b>	<ul style="list-style-type: none"> <li>▪ Durch Schiessen den ganzen Turm zu Fall bringen</li> </ul>
<b>Erwartetes Ergebnis</b>	<ul style="list-style-type: none"> <li>▪ Sobald die obersten Klötze zu sehr von Ihrer Startposition abweichen, wird dies als fallen des Turmes interpretiert und der Game Over Screen taucht auf.</li> </ul>

<b>Testfall 7:</b>	<b>Spiel Beenden</b>
<b>Ausgangslage</b>	Das Spiel ist gestartet.
<b>Schritte</b>	<ul style="list-style-type: none"> <li>▪ Durch Drücken auf P oder ESC zum Pause-Menü navigieren</li> <li>▪ Auf Main Menü drücken</li> <li>▪ Auf Exit Game drücken</li> </ul>
<b>Erwartetes Ergebnis</b>	<ul style="list-style-type: none"> <li>▪ Wird auf Exit Game gedrückt, wird das Spiel beendet.</li> </ul>

<b>Testfall 8:</b>	<b>Main Menu</b>
<b>Ausgangslage</b>	Spiel ist nicht gestartet
<b>Schritte</b>	<ul style="list-style-type: none"> <li>▪ Spiel starten.</li> <li>▪ Auf Play Drücken</li> <li>▪ P und dann auf MainMenu drücken.</li> </ul>
<b>Erwartetes Ergebnis</b>	<ul style="list-style-type: none"> <li>▪ Beim Starten des Spiels ist das MainMenu geöffnet.</li> <li>▪ Durch Drücken auf Play wird die neue Scene dargestellt, in welcher das Spiel gespielt werden kann.</li> <li>▪ Über das PauseMenu kann wieder zum MainMenu zurückgewechselt werden.</li> </ul>

### 3.5 Testprotokoll

Hinweis: Die hier referenzierten Testfälle sind im vorangehenden Kapitel ausformuliert.

Testfall #	Datum	Tester	Kommentar	Erfüllt	Massnahmen
1	06.01.2020	David Zweili	Das erwartete Ergebnis ist eingetroffen.	✓	Keine Massnahmen notwendig
2	06.01.2020	David Zweili	Das erwartete Ergebnis ist eingetroffen.	✓	Keine Massnahmen notwendig
3	06.01.2020	David Zweili	Das erwartete Ergebnis ist eingetroffen.	✓	Keine Massnahmen notwendig
4	06.01.2020	David Zweili	Das erwartete Ergebnis ist eingetroffen.	✓	Keine Massnahmen notwendig
5	06.01.2020	David Zweili	Das erwartete Ergebnis ist eingetroffen.	✓	Keine Massnahmen notwendig
6	06.01.2020	David Zweili	Das erwartete Ergebnis ist eingetroffen.	✓	Keine Massnahmen notwendig
7	06.01.2020	David Zweili	Das erwartete Ergebnis ist eingetroffen.	✓	Keine Massnahmen notwendig
8	06.01.2020	David Zweili	Das erwartete Ergebnis ist eingetroffen.	✓	Keine Massnahmen notwendig

## 4 Schlussreflexion

Ich denke mit dem Endprodukt können wir sehr zufrieden sein. Wir haben viele neue Aspekte im Arbeiten mit Unity kennengelernt, und bekannte Aspekte gefestigt. Durch unsere Agile Methode konnten wir immer ein Funktionsfähiges Produkt ausliefern und als Demo zeigen. Somit konnten wir unseren Fortschritt sehr gut sehen und waren immer wieder überrascht was wir erreicht hatten. Was uns auf jeden Fall eine Lehre war, ist dass wir die Doku nicht zu lange herausschieben dürfen. Der Reiz ist gross, einfach drauflos zu programmieren, besonders wenn man eine klare Vision vor Augen hat. Auch die projektspezifischen Bewertungskriterien haben wir ziemlich lange gar nicht angeschaut, da wir in unserer eigenen Vision vertieft waren. Somit haben wir uns zum Teil auf Aspekte fokussiert welche gar nicht unbedingt relevant für die Bewertungskriterien waren.