

## DOKUMENTATION (KEIN-)VR PROJEKT

### 26.02.2020

Zuerst wurden mittels Brainstorming verschiedene Begriffe und Ideen gesammelt, bis schlussendlich „Kochen“ als finales Thema gewählt wurde. Da Modellieren kein Teil der Aufgabe war, wurde der Asset Store von Unity durchsucht um sowohl eine passende Umgebung, als auch Zutaten zu finden. Basierend auf den gefundenen Assets wurde eine Wüstenlandschaft erstellt, in der Nahrungsmittel und Edelsteine verteilt lagen – somit war die neue Idee eines Zaubertranks geboren. Aufgrund des eigentlichen Konzepts, eine Suppe oder ähnliches zu kochen, wurden diese Veränderungen jedoch nicht gepusht.

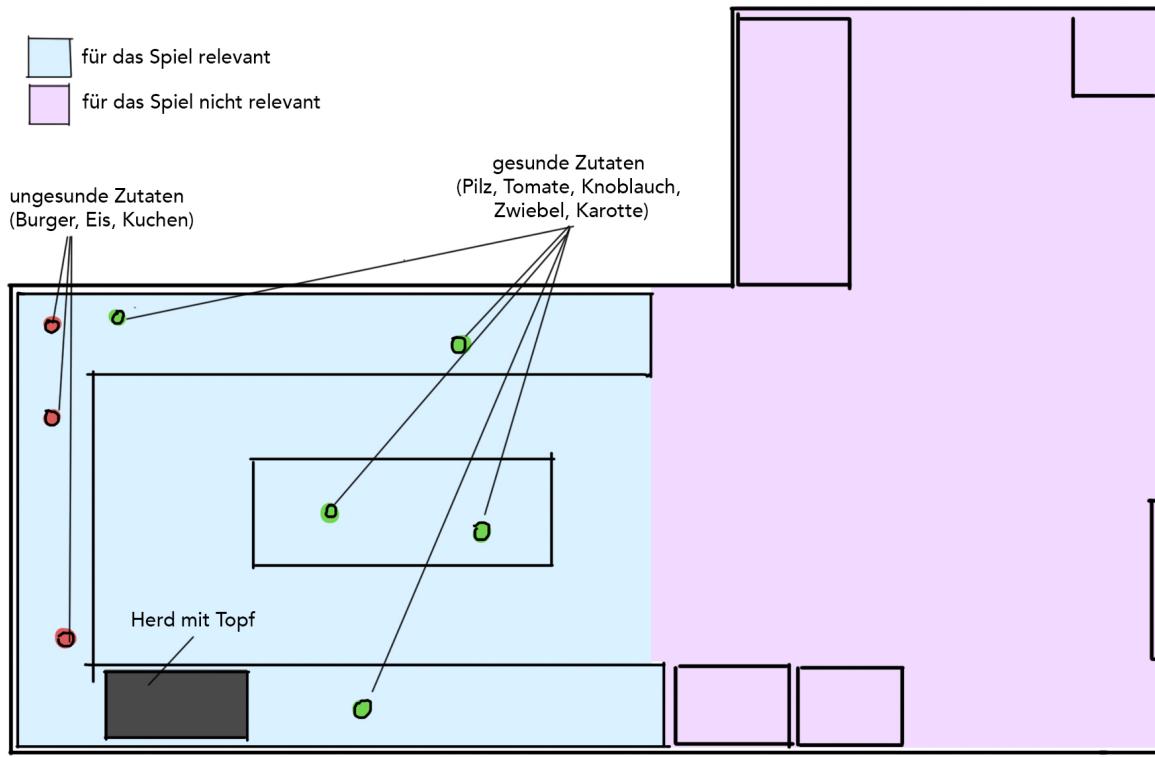
Zeitaufwand: 3 Schulstunden

### 04.03.2020

Da die Wüstenlandschaft zwar ästhetisch perfekt gepasst hätte, jedoch das Konzept vom Suppenkochen nicht aufgegeben werden wollte, wurde alles gelöscht und nochmal der Asset Store nach passenden Models durchsucht. Eine Kücheneinrichtung, die einigermaßen den Vorstellungen entsprach, und einige richtige/falsche Zutaten waren kostenlos verfügbar und wurden in einer neuen Szene platziert.

Zusätzlich zu den verschiedenen Assets wurde das Steam VR Plugin installiert und ein erster Probegang im 2D-Debug durchgeführt. Damit alle Gegenstände am richtigen Ort bleiben, mussten Collider (Box Collider, Sphere Collider) hinzugefügt und angepasst werden.

Allen Objekten, die man hochheben können soll, wurde ein Throwaway Script zugewiesen und im Rigidbody die Masse angepasst.



Skizze Küche

Zeitaufwand: 3 Schulstunden

### 11.03.2020

Da die verschiedenen Zutaten in den Topf geworfen werden sollen, musste dieser ebenfalls einen Collider haben, was sich bei dieser Form als schwierig herausstellte. Als Lösung wurden mehrere Cubes angeordnet, sodass in der Mitte ein Hohlraum ist – der Mesh Renderer wurde deaktiviert.

Um zu testen, ob der Collider funktioniert, wurde mittels Debug.Log der Trigger getestet – hat einwandfrei funktioniert. Damit zwischen „guten“ und „schlechten“ Zutaten unterschieden werden kann, wurden neue Tags gemacht, zugewiesen und mittels if Statement im ColliderPot Script ausgewählt. Die Selektion funktionierte richtig, jedoch gab es bei der Färbung des Topfes Probleme, da es trotz Anwendung verschiedener Script-vorschläge aus dem Internet nicht funktioniert hat. Schlussendlich musste auf den Renderer zugegriffen werden, um die Farbe zu ändern.

Zeitaufwand: 3 Schulstunden

### 18.03.2020

Nach der Umstellung von „VR-Projekt“ auf „Nicht-VR-Projekt“ wurde überprüft ob alles aus dem Konzept noch immer umsetzbar ist. Da das der Fall war, wurde kurz nochmal der 2D-Debug Modus ausprobiert, um zu checken ob wirklich alles bisher Gemachte funktioniert.

Weil die Dokumentation 2/5 der Benotung ausmacht, wurde alles schriftlich festgehalten, das bis jetzt gemacht wurde. Außerdem wurde versucht, dass die Färbung des Topfs nur eine Sekunde anhält, wenn etwas hineingeworfen wird.



Topf nach richtigem Item

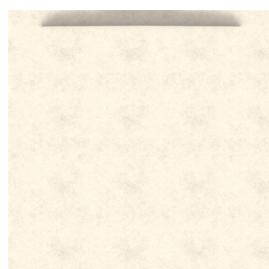


Topf nach falschem Item

Zeitaufwand: 3 Schulstunden

### 25.03.2020

Damit der Spieler weiß was er in den Topf geben muss, war es nötig ein Rezept in der Küche zu platzieren. Mittels Veränderung der Textur eines im Assets Store gefundenen Clipboard wurde das umgesetzt und in Unity eingebaut. Man sollte das Rezept auch aufheben und bewegen können, aber um zu verhindern, dass es der Spieler im Kochtopf versenkt und nicht mehr weiß, welche Zutaten er wieder an den Ursprungsort zurückgesetzt wird.



Ursprungstextur



Textur mit Rezept

Zeitaufwand: 3 Schulstunden

## 01.04.2020

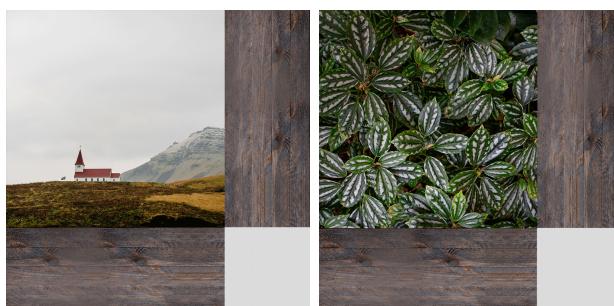
Damit der Topf nicht leer ist und der Spieler mehr das Gefühl vom Kochen hat, wurde ein Zylinder hinzugefügt und eine passende Textur ausgewählt. Diese Suppe soll je nachdem welche Zutat hinzugefügt wurde die Farbe ändern. Um das umzusetzen, bekam jedes Gemüse einen eigenen Tag, welcher im Script angesprochen wurde. Zuerst wurde versucht wie beim Topf die Farbe zu ändern, jedoch war die Methode (mit beispielsweise Color.green) zu ungenau, weshalb mit RGB Farbmischung (0-255) gearbeitet wurde. Das Objekt war dann nur noch weiß. Das Problem dabei war, dass die RGB Werte nur mit 0-1 funktionieren und alle Farben dazwischenliegen mussten.

Zeitaufwand: 2 Schulstunden

## 13.04.2020

Um die Szene lebendiger zu gestalten, wurden neue Assets (Pflanzen, Bilder, Müsli, ...) in die Szene geladen, ein Material für die Wände und die Decke erstellt und verschiedene Lichter gesetzt. Zuerst war der Plan ein Areal Light zu verwenden, aber weil das nicht funktionierte, wurden drei Point Lights verwendet. Weil manche Assets farblich nicht ganz in die Küche gepasst haben, wurden die störenden Texturen in Photoshop überarbeitet und anders gestaltet. Da verschiedene Objekte in das Regal gestellt wurden, musste dieses einen Collider bekommen. Dafür wurden verschiedene Cubes verwendet und der Mesh Renderer ausgeschaltet. Die meisten neu importierten Assets bekamen einen eigenen Collider und ein Throwaway Script. Um sicherzugehen, wurde auch noch getestet, ob diese Assets den richtigen Trigger beim Topf auslösen – was der Fall war.

Damit das Spiel ein Ende hat, wurden if Statements verwendet, die erkennen, ob alle richtigen Zutaten im Topf sind. Sobald das der Fall ist, wird das Spotlight über dem Topf grün. Wenn man nun das Clipboard betrachtet, hat sich die Seite geändert und eine Zeichnung von einem Löffel ist zu sehen. Wenn man den Löffel in den Topf wirft, ist das Spiel komplett vorbei. Alles wird schwarz und man kann mit nichts mehr interagieren.



Vorher (links) – Nachher (rechts)



Vorher (open) – Nachher (unten)

Zeitaufwand: 4,5 Schulstunden

## 15.04.2020

Weil die Interactable Items trotz „abschalten“ des Lichts sichtbar bleiben, versteht man als Spieler nicht, dass das Spiel vorbei ist. Deshalb wurde ein Canvas mit schwarzem Hintergrund und Text erstellt, jedoch funktionierte das zuerst nicht, da es an die Mainkamera gebunden wurde. Als Workaround wurde eine zweite Kamera erstellt und zu „FallbackObjects“ als Child angehängt. Im Script konnte der Canvas wieder ausgeblendet werden und erst wenn am Ende der Löffel im Topf landet, wird der Text und die Farbfläche sichtbar. Dadurch erspart man sich das Abdrehen der Lichter.

Ein Problem, welches auftreten könnte, ist, dass der Spieler den Löffel zuvor schon versenkt. Deshalb wurde im Script definiert, dass dieser wieder an der Originalposition auftaucht.

Zeitaufwand: 2 Schulstunden

### **19.04.2020**

Nach Ändern des Endscreens wurde zusätzlich noch ein ParticleSystem erstellt. Das Ziel war Dapf über dem Kochtopf zu gestalten, doch trotz einiger Tutorials wurde es nicht so wie gewollt, weshalb dann schlussendlich ein ParticleSystem aus dem Assetstore angepasst und verwendet wurde. Als nächstes wurden zwei Audioclips von freesound.org runtergeladen und in das Unityprojekt importiert. Die Sounds sollten erst abgespielt werden, wenn bestimmte Trigger ausgelöst wurden, was im Script festgelegt wurde. Problem dabei war, dass man die Audioclips in Unity nicht in die dafür vorgesehenen public Fields ziehen konnte – Lösung dafür war die Sounds in die Szene als Objekt hinzuzufügen. Ein weiteres Hindernis war, dass die Audios zu Beginn des Spiels sofort abgespielt wurden – um das zu verhindern musste einfach der Haken bei „Play on Awake“ entfernt werden.

Zeitaufwand: 1,5 Schulstunden