

VR-Projekt

Wintersemester 2024



Bericht Gruppe 08

Burger Maker

Stefan Procik minf104111 (TB087)

Christelle Maaß minf104420 (TB087)

Jonathan El Jusup cgt104707 (TB087)

Prince Lare-Lantone cgt104645 (TB087)

INHALTSVERZEICHNIS

1	Einleitung	1
2	Spielbeschreibung	1
2.1	Gameloop	1
2.2	Zubereitung des Burgers	1
2.3	Spielmodi	2
3	Projektorganisation	3
3.1	Tutorial & Affordanz	3
3.2	Stapeln von Zutaten	4
3.3	Anbraten des Patties	4
3.4	Bewertungssystem	5
3.5	Modellierung	5
3.6	Spielmodi	5
3.7	Optionale Features	7
4	Arbeitsteilung	9
4.1	Stefan Procik	9
4.1.1	Überblick	9
4.1.2	Beschreibung	9
4.2	Christelle Maaß	15
4.2.1	Überblick	15
4.2.2	Beschreibung	15
4.3	Jonathan El Jusup	21
4.3.1	Überblick	21
4.3.2	Beschreibung	21
4.4	Prince Lare-Lantone	29
4.4.1	Überblick	29
4.4.2	Beschreibung	29
5	Playtesting	34
6	Kompilieren	35
7	Externe Assets & Referenzen	36
8	Bewertung	37
8.1	Stefan Procik	37
8.2	Christelle Maaß	37
8.3	Jonathan El Jusup	38
8.4	Prince Lare-Lantone	38

1 EINLEITUNG

Stack & Flip: Burger Maker ist eine VR-Kochsimulation, in der der Spieler Burger nach Bestellung zubereitet. Hierzu muss er zum einen Patties anbraten und weitere Zutaten aufeinanderstapeln. Je nach Bestellung variieren die Zutaten. Zusammen mit der Bestellung wird der zubereitete Burger am Ende nach speziellen Kriterien bewertet. Es gibt 2 Spielmodi: Der Standard-Spielmodus simuliert einen einfachen Arbeitstag mit einer bestimmten Anzahl von Bestellungen, welche erfüllt werden müssen. Im endlosen Hell's-Kitchen-Spielmodus kommen kontinuierlich neue Bestellungen rein. Diesmal steht der Spieler unter Zeitdruck durch einen Timer. Ziel ist es, so viele Bestellungen wie möglich zu erfüllen.

2 SPIELBESCHREIBUNG

2.1 GAMELOOP

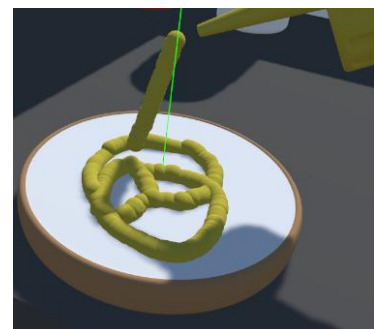
Der Gameloop besteht grundlegend aus dem Nehmen einer Bestellung, das Zubereiten eines Burgers nach dieser Bestellung und die Bewertung. Zu Beginn des Spiels kommen neue Bestellungen rein, welche abgearbeitet werden müssen. Wurde ein Burger zubereitet, legt der Spieler diesen und die korrespondierende Bestellung auf einer die dafür vorgesehene Fläche und betätigt die Klingel, um bewertet zu werden. Dies wiederholt sich für jede Bestellung. Dieser Gameloop findet sich in jedem Spielmodi wieder.

2.2 ZUBEREITUNG DES BURGERS

Der Kern des Spiels ist das Stapeln von Zutaten aufeinander, um einen Burger zuzubereiten. Hierfür startet jeder Burger mit einem „Bottom Bun“, welcher als sog. „Stack“ fungiert. Auf diesen können jegliche Zutaten gestapelt werden. Von einzelnen Zutaten wie Salat oder Käse, bis hin zu Toppings wie Tomatenscheiben, Zwiebelringen, aber auch komplexere Zutaten wie Soßen oder einen Patty, dessen Hälften zunächst angebraten werden müssen. Hierbei sind viele Kombinationen und Variationen denkbar und in den selbst kreierten Bestellungen festgehalten, welche der Spieler erfüllen muss.

Zum Stapeln der Zutaten nimmt der Spieler Zutaten und hält sie über dem Stack. Schwebt eine Zutat über dem Stack, wird eine sog. „Distance Guide“ sichtbar, welche dem Spieler anzeigt, wie mittig die Zutat über dem Stack liegt. Ist alles im grünen Bereich, kann der Spieler seine Zutat loslassen. Diese fällt dann auf dem Stack und wird Teil dessen.

Ein wichtiger Bestandteil der Zubereitung des Burgers ist das Anbraten des Patties. Hierfür nimmt sich der Spieler einen Rohen Patty und legt ihn auf die Kochfläche. Der Patty besteht aus 2 Hälften, welche separat angebraten werden müssen. Liegt eine Hälfte auf der Kochfläche beginnt diese zu braten. Dies wird dem Spieler visuell und auditiv mitgeteilt. Ziel ist es, beide Hälften des Patties perfekt anzubraten.



Neben Zutaten, die aufgehoben und auf dem Stack gelegt werden können gibt es auch eine Reihe von verschiedenen Soßen, welche aus Tuben

gequetscht werden. Hierfür nimmt sich der Spieler eine Soßentube und drückt diese. Abhängig davon, wie stark der Spieler die Soßentube drückt, kommen größere oder kleinere Soßenkugeln raus.

2.3 SPIELMODI

Lobby

Die Lobby ist kein expliziter Spielmodus, kann jedoch als eine Art Tutorial und Sandbox betrachtet werden. Hier wird dem Spieler das Spielprinzip und die Steuerung erklärt. Es kommen kontinuierlich neue Bestellungen rein, an denen sich der Spieler setzen kann. Ihm ist aber freie Wahl gelassen, auch einen Burger nach seinen eigenen Vorstellungen zuzubereiten. In der Lobby kann sich der Spieler ohne Stress mit der Steuerung und dem Spielprinzip vertraut machen, bevor er in den Standard- oder Hell's-Kitchen-Spielmodus geht.



Die Lobby ist mit vielen Hinweisen versehen wo Zutaten und Bestellungen erscheinen, wo man den Patty anbrät und was der generelle Ablauf ist; Vom Nehmen der Bestellung, der Zubereitung des Burgers, bis hin zur Bewertung. Außerdem befindet sich in der Lobby ein zusätzliches Menü, welches den Standard- und Hell's-Kitchen-Spielmodus erklärt. Es kommen fortlaufend neue Bestellungen rein und der Spieler kann so viele Burger zubereiten, wie er möchte.

Standard

Im Standard-Spielmodus wird eine typische Arbeitsschicht simuliert. Dafür muss der Spieler eine bestimmte Anzahl an Bestellungen erfüllen. Jeder Burger wird separat bewertet. Wurden alle Bestellungen erfüllt, werden alle Bewertungen miteinander verrechnet und der Spieler bekommt eine Endpunktzahl. Hinzu kommt die Sauberkeit der Küche, welche in die Endbewertung mit einfließt. Hat nämlich der Spieler zu viele Zutaten während seiner Schicht auf den Boden fallen gelassen oder gar Zutaten in den Essbereich geschmissen, wirkt sich dies negativ auf die Bewertung aus. Der Fokus liegt hierbei auf Qualitäts-Maximierung.

Hell's Kitchen

Der Hell's-Kitchen-Spielmodus ist ein endlos Spielmodus, in welchem Bestellungen kontinuierlich reinkommen. Der Spieler steht hier unter Zeitstress. Während des ganzen Spiels läuft ein Timer, der nur zurückgesetzt wird, wenn eine Bestellung erfüllt wurde. Hierbei gilt es zu beachten, dass dieser bei jedem Zurücksetzen, etwas schneller wird. In diesem Spielmodus steht weniger Qualität, sondern Anzahl von erfüllten



Bestellungen im Fokus. Ziel ist es, so viele Bestellungen wie möglich zu erfüllen, bevor die Zeit abläuft. Sobald der Timer abgelaufen ist, endet das Spiel und der Spieler bekommt eine Anzeige mit der Anzahl der Bestellungen, die er „überlebt“ hat.

Damit der Spieler nicht ausnutzen kann, unfertige Burger bzw. leere Stacks abzuliefern, um den Timer zurückzusetzen und seinen Punktestand zu erhöhen, wurde folgende Maßnahme entwickelt, um diesen davon abzuhalten. Liefert der Spieler einen Burger ab, der nicht einem gewissen Qualitätsmaß entspricht, wird der Timer nur teilweise zurückgesetzt und der Spieler bekommt weniger Zeit. Diese Bestrafung akkumuliert sich, wenn der Spieler solches wiederholt. Diese Bestrafung des Spielers wird durch mehrere visuelle und auditive Elemente untermalt.

3 PROJEKTORGANISATION

Bei der Planung des Projektes haben wir zunächst den Fokus auf die Implementierung der Stacking Mechanik gelegt, um zu verifizieren, dass dies gut funktioniert und Spaß macht. Dies konnten wir zum Proof-of-Concept Meilenstein bestätigen und bot uns eine solide Basis für weitere Schritte. Aufbauend darauf folgten das Anbraten des Patties, verschiedene Zutaten und das Bewertungssystem. Wir haben uns bewusst nicht zu viel vorgenommen, dafür zunächst viel optional gelassen. Dies hat sich während der Entwicklungsphase als vorteilhaft herausgestellt. Dies bot uns Flexibilität und wir konnten uns dynamischer Aufgaben zuteilen und auch etwas abschweifen, ohne das Grundkonzept aus den Augen zu verlieren.

Rückblickend können wir bestätigen, dass wir unserer Vision treu geblieben sind und es uns leisten konnten, viele optionale Arbeitspakete zu implementieren und einige davon auch tiefer in die Hauptbestandteile des Projektes zu integrieren. Zusätzlich zum Basispaket konnten wir dieses noch weiter abrunden und unsere eigenen persönlichen Noten hinzugeben.

3.1 TUTORIAL & AFFORDANZ

Da VR noch ein relatives Nischenprodukt ist, können wir nicht erwarten, dass jeder mit dieser Technologie vertraut ist. Deswegen war es umso wichtiger, dem Spieler das Spielprinzip und die Steuerung so einfach wie möglich zu vermitteln. Diese Notwendigkeit stellte sich schon früh bei den ersten Playtests heraus. Die Spieler wollen sich im Klaren sein, was sie tun sollen und ein gewisses Feedback auf ihre Aktionen bekommen. Anfangs haben wir alles mit einer rudimentären README-Datei erklärt, waren uns aber bewusst, dass dies keine Endlösung bieten würde.

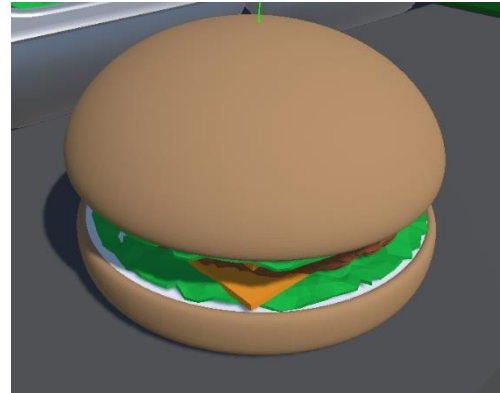
Schon früh war es unser Ziel, dem Spieler so viele Informationen zu geben, wie er braucht. Von der Erklärung des Spielprinzips und der Steuerung bis hin zu visuellen und auditiven Hinweisen, welche wir nach und nach eingeführt haben. Beispielsweise haben wir es dem Spieler durch einen „Distance Guide“ auf einfache Weise deutlich gemacht, wie mittig er eine Zutat über dem Stack hält. Der Fortschritt von Patties, die gerade angebraten werden, wurde durch Fortschrittsbalken deutlich gemacht. Viele Elemente wurden noch mit weiteren visuellen Hinweisen wie Animationen, Änderung der Farbe oder Soundeffekten deutlich gemacht.

Zuletzt haben wir uns ganz im Sinne eines Tutorials für eine Lobby entschieden, in welcher dem Spieler das Spielprinzip und die Steuerung erklärt werden. Mit der Einführung der Lobby

und allen zuvor genannten Elementen sollte dem Spieler alles gegeben sein, um zu verstehen, wie das Spiel funktioniert.

3.2 STAPELN VON ZUTATEN

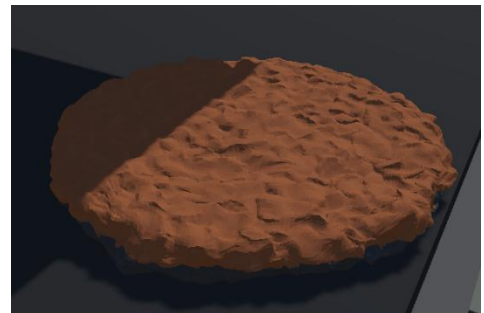
Das Stapeln von Zutaten ist die Hauptmechanik des ganzen Spiels. Demnach war es eine der ersten Funktionalitäten, welche wir implementiert haben. Diese haben wir zum Proof-of-Concept zunächst rudimentär implementiert, um zu zeigen, dass diese Mechanik funktioniert und Spaß macht. Im Verlauf der Entwicklung wurde diese Mechanik weiter ausgereift und verbessert.



Zum Stapeln von Zutaten nimmt der Spieler initial einen „Bottom Bun“, welcher als sog. „Stack“ fungiert. Es ist die Basis, auf der neue Zutaten gestapelt werden können. Zum Stapeln wird vorausgesetzt, dass der Spieler die zu stapelnde Zutat relativ mittig über dem Stack hält. Zur Hilfe wurde eine sog. „Distance Guide“ implementiert, die dem Spieler zeigt, wie mittig er die Zutat über dem Stack hält. Diese Platzierung der Zutat ist eine von mehreren Metriken, nach denen ein Burger bewertet wird. Es gibt verschiedene Zutaten, die gestapelt werden können. So ist es beispielsweise für mehrere Toppings nicht so wichtig, diese mittig auf dem Burger zu platzieren. Oder beim Auftragen von Saucen ist die Position egal, solange die Soße auf dem Stack liegt. Eine weitere Herausforderung war die Orientierung des Stacks, welcher flach liegen muss. Dies wurde erst gegen Ende der Entwicklung implementiert. Alles in allem haben wir die Stacking-Mechanik zufriedenstellend implementieren und abrunden können. Es gibt nichts, was der Mechanik noch fehlen würde.

3.3 ANBRATEN DES PATTIES

Für die Zubereitung eines Burgers darf das Anbraten von Patties auf keinen Fall fehlen. Diese und die Stacking-Mechanik gingen einher, konnten jedoch auch unabhängig voneinander und parallel implementiert werden. Dieses Arbeitspaket eignete sich sehr gut für die Gruppenmitglieder, welche noch nicht viel mit der Unity-Engine gearbeitet haben und galt als eine Art Einführung in die Engine. So konnten wir uns mit der Implementierung mehr Zeit lassen, da das Spiel theoretisch auch erst ohne diese Mechanik funktionieren würde.



Das Anbraten eines Patties sollte über 2 Hälften separat geschehen. Der Spieler sollte sich dabei Mühe geben, beide Hälften perfekt und gleich lang anzubraten. Hierfür muss der Patty auf halben Wege gewendet werden, damit die andere Hälfte angebraten werden kann. Diese Interaktion mit dem Patty sollte für mehr Tiefe sorgen, damit sich der Patty von den anderen Zutaten besser abgrenzt.

Ursprünglich wollten wir das Anbraten von Patties komplexer gestalten und verschiedene Stufen implementieren (Rare, Medium, Well-done). Es hat sich jedoch während des Entwicklungsprozesses herausgestellt, dass diese steigende Komplexität den Spieler doch sehr

überfordern würde. Für Einsteiger in das VR-Genre, sollte Interaktion mit Spielelementen eher im Vordergrund stehen als Mikro-Management. Rein objektiv betrachtet wäre die Implementierung verschiedener „Roasting-Stufen“ nicht zu komplex gewesen. Dies würde doch sehr stark auf Kosten des Spielspaßes gehen und würde das Bewertungssystem umso komplexer machen. Dennoch denken wir, dass wir eine gute Mitte gefunden haben, dass diese Mechanik einzigartig ist, das Anbraten des Patties Spaß macht und dies den Spieler dennoch nicht überfordert.

3.4 BEWERTUNGSSYSTEM

Das Bewertungssystem war eine der herausforderndsten Funktionalitäten, die wir implementiert haben. Dafür mussten alle Zutaten kategorisiert und mit Metadaten versehen werden, welche eine Bewertung ermöglichen sollten. Wir mussten uns für gewisse Bewertungs-Metriken entscheiden, diese entsprechend gewichten und diese dem Spieler transparent mitteilen.

Zum ersten Proof-of-Concept Meilenstein war das Bewertungssystem noch nicht notwendig, doch haben wir uns schon früh Gedanken gemacht, ein geeignetes Grundgerüst zu erstellen, auf dem das Bewertungssystem dann aufgebaut werden sollte. Zudem sollte sichergestellt werden, dass die Einführung neuer Zutaten schnell und ohne große Mühe in das Bewertungssystem integriert werden kann. Dies hat uns in langer Hinsicht viel Arbeit erspart und das Bewertungssystem relativ flexibel und erweiterbar gemacht.

Für die Bewertung war aber auch eine angemessene Gewichtung aller Metriken notwendig. Durch viel Testen und Feineinstellen der Gewichte konnten wir am Ende eine angemessene Bewertung erzielen, welche den Spieler nicht zu sehr bestraft und Platz für Fehler erlaubt. Dieses Bewertungssystem wurde zum zugrundeliegenden System, welches alle Spielmodi verwenden und den Spielfluss beeinflusst. Dementsprechend haben wir auch signifikant viel Zeit und Ressourcen darin investiert, um sicherzustellen, dass das Bewertungssystem richtig funktioniert.

3.5 MODELLIERUNG

Schon von Anfang an hatten wir uns vorgenommen, die meisten Assets selbst zu modellieren. Dies sollte u. a. das Ungleichgewicht der Erfahrungen mit der Arbeit mit Game-Engines abfedern, sodass die Medieninformatiker sich mehr auf die Modellierung fokussieren konnten und einen leichteren Einstieg in die verwendete Engine bekommen. Dies hat sich gegen Ende des Entwicklungsprozesses bewährt und wir können von uns behaupten, dass der Großteil unserer Assets selbst modelliert wurde. Dies hatte zudem auch den Vorteil, dass wir den Stil kohärenter und konsistenter gestalten konnten. So wurden jegliche Elemente der Küche, Zutaten, der Essbereich und der Außenbereich komplett selbst modelliert, mit Ausnahme von ein paar Texturen.

3.6 SPIELMODI

Ursprünglich war es vorgesehen, 3 Spielmodi zu implementieren: Standard, Hell's-Kitchen und Sandbox. Schon während des Exposés wurde uns schnell klar, dass der Sandbox-Spielmodus nur einen kurzen Spielspaß bieten würde, weshalb wir diesen zunächst verworfen haben. Hauptfokus der Entwicklung lag im Standard-Spielmodus, welcher später dann auch als Basis des Hell's-Kitchen-Spielmodus' dienen sollte. Während des Entwicklungsprozesses lag der

größte Fokus auf diesem Spielmodus. Deshalb wurde anfangs die Implementierung des Hell's-Kitchen-Spielmodus aufgeschoben.

Nach des MVP-Meilensteins begann die tatsächliche Entwicklung des Hell's-Kitchen-Spielmodus, welcher auf dem Standard-Spielmodus aufbaute. Dabei sollte jedoch die Zeitkomponente in den Vordergrund rücken. Uns war wichtig, diesen Spielmodus von den anderen klar abzugrenzen. Die Einführung eines Timers und eines anderen Ziels allein sollten dafür nicht ausreichen. Uns war wichtig, dem Spieler ein Gefühl von Zeitdruck zu geben. Wir fingen zunächst damit an, dem Timer mehr Bedeutung zuzuweisen und kamen schnell auf die Idee, unser optionales Arbeitspaket von Easter Eggs und kleinen Referenzen als Bestandteil des Spielmodus mit einzubauen. Dies manifestierte sich durch ein 2D-Gordon-Ramsay-Billboard, welches sich dem Spieler nähert, während der Timer abläuft. Uns gefiel die Idee so sehr, dass wir uns leicht in die Absurdität mitreißen ließen. Das Endprodukt ist ein Gordon-Ramsay-Billboard mit roten Augen, welches sich dem Spieler nähert, Anmerkungen macht, wenn der Spieler Burger abliefert, die nicht einem gewissen Maß an Qualität entsprechen und diese Burger mit seinen roten Augen durch Laserstrahlen vernichtet.



Dennoch hat jedes Element seinen Sinn und soll eine gewisse Atmosphäre erzeugen; Alles mit dem Ziel, den Spieler unter Druck zu setzen. Trotz allem soll diese Absurdität das Spiel als solches relativieren. Denn es handelt sich immer noch um ein Spiel und nicht um die Realität. Der Spieler muss sich nicht fürchten oder abgewertet fühlen.

Zusammen mit dem Hell's-Kitchen-Spielmodus kamen wir doch dazu, den Sandbox-Spielmodus einzubauen. Wir haben uns aber dazu entschieden, diesen nicht als expliziten Spielmodus zu verpacken, sondern als Teil der Lobby zu integrieren. Dort wird dem Spieler die Steuerung und das Spielprinzip erklärt. Der Spieler hat freie Wahl, was er machen möchte. Ob er nun Burger nach Bestellungen zubereitet oder einen eigenen Burger nach seinen Vorstellungen kreiert, ist ihm überlassen. Hier kann der Spieler ohne Zeitdruck und Stress sich mit dem Spiel vertraut machen und alles ausprobieren. Wir denken, dies war ein guter Kompromiss, die Sandbox in dieser Form eingebunden zu haben.

3.7 OPTIONALE FEATURES

Wir hatten eine Reihe von optionalen Arbeitspaketen, bei denen wir uns noch nicht sicher waren, ob wir diese in das Projekt aufnehmen sollten. Eine der ersten Funktionalitäten, welche wir ins Projekt aufgenommen haben, war die Sauberkeit der Küche, welche auch bei der Proof-of-Concept-Vorstellung vorgeschlagen wurde.

Auch für Easter Eggs haben wir uns entschieden, diese zunächst versteckt einzubauen, wie beispielsweise ein Plankton aus SpongeBob, der versteckt hinter einem Schrank in der Ecke der Küche zu finden ist. Während der Entwicklung des Hell's-Kitchen-Spielmodus sind wir aber dann zu dem Schluss gekommen, diese Easter Eggs nicht mehr zu verstecken, sondern diese zu zelebrieren. So wollten wir anfangs ein paar Sprüche von Gordon Ramsay ins Spiel packen, haben uns dann dafür entschieden, Gordon Ramsay als Billboard in die Spielszene zu setzen, welcher den Spieler beleidigt, wenn er schlechte Burger abliefert. Auch wenn solche Easter Eggs das Ganze humorvoller machen sollten, haben sie dennoch ihren Zweck, den sie erfüllen. So nähert sich Gordon Ramsay dem Spieler, ausgehend von der Zeit, die ihm noch bleibt. Er verschießt Laser auf einen Burger, der schlecht zubereitet worden ist und merkt das dem Spieler an. Egal wie chaotisch diese Elemente auch erscheinen mögen, helfen sie, die zugrundeliegende Logik und Konsequenzen zu veranschaulichen.

Ein weiterer Punkt waren die verschiedenen Küchen-Umgebungen. Anfangs wollten wir verschiedene Umgebungen bieten, zwischen denen der Spieler wechseln kann. Natürlich wäre das ein wunderbarer Zeitfüller gewesen, machte doch am Ende wenig Sinn, wenn diese funktional keinen Unterschied machen. Am Ende haben wir uns für einen Kompromiss entschieden, die Küche im Hell's-Kitchen-Spielmodus umzugestalten, um eine passende Atmosphäre zu gestalten. Dies beschränkte sich nicht nur auf die Küche, sondern auch auf den Essbereich und den Außenbereich.

Wie schon in der Spielbeschreibung erwähnt, haben wir den Burger-Creator- bzw. Sandbox-Spielmodus in neuer Form implementiert. Da befürchtet wurde, dass dieser nur einen kurzen Spielspaß gewähren würde, haben wir ihn in der Lobby neu verpackt. Während die Lobby als Tutorial-Szene fungiert, kann der Spieler dort auch Burger zubereiten, ob nach Bestellung oder nach eigenen Vorstellungen. Ihm sind in dieser Hinsicht keine Grenzen gesetzt. Auf die Funktionalität, eigene Kreationen zu speichern, haben wir verzichtet. Doch es gab auch optionale Features, die wir nicht weiterverfolgt haben. Darunter zählt die komplexere Zubereitung des Burgers. Wir hatten uns gedacht, den Zubereitungsprozess tiefer zu gestalten, indem z. B. die Brötchen getoastet werden oder Käse geschmolzen werden sollte. Wir haben am Ende nur einige dieser Ideen umgesetzt, um den Prozess nicht zu komplex zu gestalten, denn jede weitere Komplexitätsstufe müsste auch im Bewertungssystem berücksichtigt werden. Wir haben uns für Features entschieden, welche für Tiefe sorgen würden, ohne die Komplexität zu sehr zu erhöhen. Darunter zählt das Anbraten beider Patty-Hälften. In diesem Fall wurde in der Bewertung berücksichtigt, wie lange beide Seiten angebraten wurden und wie stark deren Differenz ist. Auch haben wir uns für die Klingel entschieden, die den ganzen Prozess von der Zubereitung des Burgers und dessen Bewertung abschließt und dem Spieler ein gewisses Taktgefühl gibt.

Auch für die verschiedenen Schwierigkeitsstufen haben wir uns am Ende nicht entschieden, da die Standard- und Hell's-Kitchen-Spielmodi bereits jeweils einer Schwierigkeitsstufe entsprechen. Dennoch können wir sagen, dass wir viele optionale Features implementieren konnten, darunter einige neu verpackt und vorgestellt, andere sogar als Teil von

Hauptmechaniken. Diese einzelnen Features in ihrer Menge und in Verbindung mit den Hauptelementen des Spiels unterstützen sich gegenseitig und sorgen für ein abgerundetes Produkt.

4 ARBEITSTEILUNG

4.1 STEFAN PROCIK

4.1.1 Überblick

Gesamtzeit: 116 Stunden

1. Projekteinrichtung	5 Stunden
2. Meetings und Besprechungen	18 Stunden
3. PoC	4 Stunden
4. Playtesting	2 Stunden
5. Playtests Auswerten	2 Stunden
6. MVP	3 Stunden
7. Burger-Rezept	25 Stunden
8. Blender/Umgebung Modellierung	44 Stunden
9. Dokumentation und Kommentierung	13 Stunden

4.1.2 Beschreibung

1. Projekteinrichtung

Die Projekteinrichtung umfasste eine Diskussion aller Gruppenmitglieder bezüglich der zu nutzenden Technologien und der endgültigen Entwicklungsumgebung. Trotz erhaltener Eingrenzung waren trotzdem viele Möglichkeiten offen, unsere Idee eines „Burger Makers“ zu realisieren. Darüber hinaus brachten alle Gruppenmitglieder unterschiedlichste Erfahrungen mit verschiedenen Technologien ein, wodurch jeder seine Einschätzung und Präferenz hinsichtlich der am besten geeigneten Technologien darlegen konnte.

Letztendlich entschieden wir uns für die Nutzung einer Long-Term-Support (LTS)-Version von Unity und des Versionsverwaltungssystems Git. Die Wahl der Unity-Engine beruhte auf der Übereinstimmung positiver Erfahrungen und ihrer benutzerfreundlichen Handhabung. Das Git-Versioning-Tool hingegen war mit einer einstimmigen Meinung das passendste Tool für unser Projekt gewesen, insbesondere aufgrund der zahlreichen kostenlos verfügbaren GUI-Git-Clients.

2. Meetings und Besprechungen

Zur Planung und Standbesprechung haben wir uns als Gruppe mehrfach online und vor Ort getroffen, um weitere Schritte zu planen und den aktuellen Stand zu besprechen.

3. PoC

Wir trafen uns als Projektgruppe im VR-Labor, um den Proof-of-Concept (PoC) abzuschließen und auf kritische Fehler zu testen. Ziel dieser Tests war es, sicherzustellen, dass der PoC vorführbereit ist.

4. Playtesting

Testung der Projekte „Whack-a-mole“ und „Sword and Shield“ im VR-Labor und dessen Aufzeichnung.

5. Playtests auswerten

Die Aufsicht während der Playtests vor Ort von den Kommilitonen war ein sehr aufschlussreicher Schritt in der Entwicklung unseres VR-Projekts. Die Perspektive von Spielern zu erhalten, die nicht am Spiel mitgearbeitet haben, ermöglichte es uns, Elemente und Ansätze zu identifizieren, die wir zu diesem Zeitpunkt noch nicht berücksichtigt hatten.

Der Umgang der Kommilitonen mit unserem Projekt war dabei sehr unterschiedlich. Während einige vorsichtig mit der Umgebung interagierten, um nichts zum Abstürzen zu bringen und sich auf das Testen der Kernmechaniken konzentrierten, setzten andere wiederum stark auf sog. „Gamebreaking“. Diese gezielte Suche nach Spielfehlern zeigte, dass viele Szenarien von Entwicklern nicht vorhergesehen werden konnten, wie beispielsweise das Wegwerfen von Spielobjekten und das Fehlen einer Methode, diese Objekte respawnen zu lassen. Viele der entdeckten „Gamebreaking“-Bugs standen in Zusammenhang mit der VR-Umgebung und deren umfangreichen Interaktionsmöglichkeiten. Je immersiver die virtuelle Welt um den Spieler konzipiert wird, desto wahrscheinlicher ist es, dass die Spieler auf unerwartete Weise mit ihr interagieren. Solche Interaktionen können nur durch umfangreiches Playtesting identifiziert werden. Aus diesen Gründen waren die Playtests sehr aufschlussreich und von großem Wert für uns. Sie halfen uns nicht nur, als perfekt angesehene Konzepte weiter zu verbessern, sondern auch, die Spielerführung zu optimieren und eine präzisere Anleitung für die Spieler zu entwickeln.

6. MVP

Treffen vor der Präsentation des MVPs und der anschließenden Vorträge.

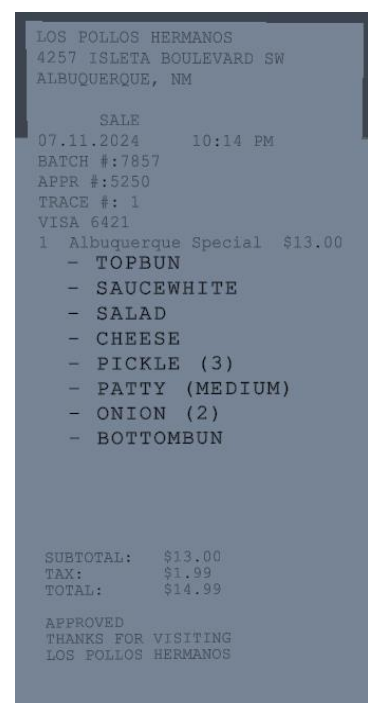
7. Burger-Rezept

Das Burger-Rezept stellt eines der zentralen Elemente des Spiels dar, da es als wesentliche Anleitung dient, nach der sich der Spieler richten muss, um das Spiel erfolgreich zu spielen. Es dient dazu, alle benötigten Informationen zur Erstellung eines Burgers dem Spieler leserlich und eindeutig darzustellen.

Nach eingehender Recherche stellte sich heraus, dass in Restaurants üblicherweise nur der Name und die Anzahl der bestellten Burger angegeben werden, nicht jedoch das Rezept per se oder die Zusammenstellung. Dies führte zur Überlegung, wie genau dem Spieler die Anleitungen der Burger vermittelt werden sollen.

Eine Möglichkeit wäre es, durch eine fliegende Infobox dem Spieler das Rezept darzustellen, doch diene dies nicht der Immersion des Spielers als Burgerbrater. Die zweite Möglichkeit wäre es, dem Spieler eine Art von Rezeptbuch anzubieten, welches aber wiederum den Nachteil hätte, dass der Spieler je nach endgültiger Anzahl an Rezepten eine Menge blättern müsste, was als anstrengend, wenn nicht schon nervend, empfunden werden kann.

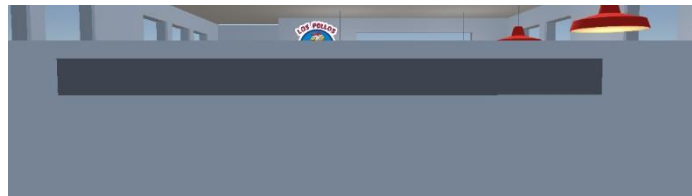
Schließlich kamen wir zu dem Entschluss, die Anleitung der Burger direkt auf die Bestellung darzustellen. Dies hat den Vorteil, dass der Spieler immer direkt die Anleitung für die



Bestellung vor sich hat und dabei die Immersion beibehält, da die Anleitung nicht unrealistisch erscheint wie eine fliegende Infobox.

Darüber hinaus galt es, die Bestellung als solche auch kenntlich zu machen. Da unser Restaurant das Thema des Restaurants aus der Serie „Breaking Bad“ besaß, empfanden wir es als unabdinglich, dies auch in der Bestellung zu repräsentieren. Dazu haben wir recherchiert, was die reale Adresse des Drehorts des Restaurants „Los Pollos Hermanos“ war, und dieses in Albuquerque, New Mexico, gefunden und somit in die Bestellungen übernommen, als ein sog. „Easter Egg“ für die Kenner der Serie. Außerdem haben wir das Datum und die Uhrzeit in Echtzeit sowie die BATCH, APPR, VISA, SUBTOTAL, TAX und TOTAL bei jeder Bestellung integriert, um die bestmögliche Immersion zu realisieren.

Die Bestellungen werden als JSON-Objekte im Ordner „Streaming-Assets“ gespeichert und sind dem Parser somit zur Laufzeit zugreifbar. Der Parser konvertiert die JSON-Objekte in String-Objekte, um diese anschließend in „Order“-Objekte zu überführen, welche die Informationen zur Bestellung und zur Rezeptur des Burgers enthalten. Diese „Order“-Objekte können dann einem Bestellungs-Objekt zugewiesen werden, welches gespawnt und dem Spieler zur Interaktion bereitgestellt wird.



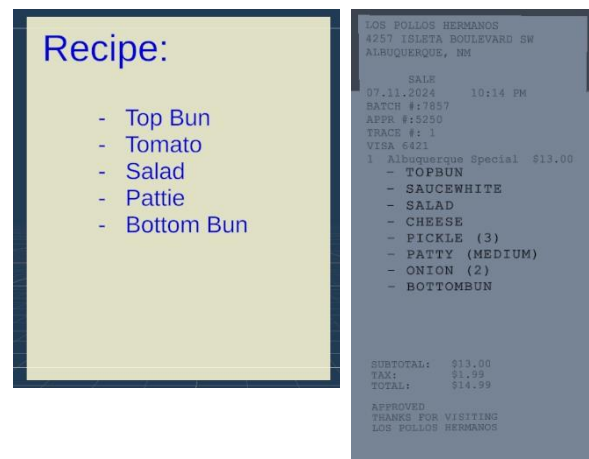
Die Bestellungen werden auf einer Metallleiste mit fünf festen Spawnpositionen generiert, wo sie vom Spieler abgenommen werden können. Sobald eine Bestellung entnommen wurde, wird, sofern noch Bestellungen ausstehen, eine neue Bestellung an der freigewordenen Position generiert.

8. Blender/Umgebung Modellierung

Die Modellierung von Spielobjekten und Umgebung war, neben der Konzeptionierung und Implementierung der Burger-Bestellungen, eine der zeitintensiven Aufgaben für mich in diesem VR-Projekt.

8.1 Rezept / Bestellung

Eines der essenziellsten Modelle mit der höchsten Interaktionsrate des Spielers war das Rezept bzw. die Bestellung. Zunächst war es wichtig, dem Spieler das Rezept für die Bestellung einfach und leicht leserlich zu übermitteln. Dies wurde durch das gelbe Modell realisiert. Anschließend galt jedoch, das Rezept dem Spieler immersiv zu übergeben. Hierbei eignete sich ein Kassenbon als Bestellung mit Rezept des gewünschten Burgers. Diese Lösung fand in der Projektgruppe breiten Konsens und wurde als finale Entscheidung umgesetzt. Der Kassenbon orientiert sich an üblichen Formatierungen von



Restaurantbestellungen, was sowohl an der Font als auch an den zusätzlichen, für unser Spiel nicht unbedingt notwendigen Informationen erkennbar ist. Zur weiteren Verbesserung galt es, die Informationen auf dem Bon realistisch zu gestalten, worauf bereits in Abschnitt 2 näher eingegangen wurde.

8.2 Restaurant



Das virtuelle Restaurant wurde dem realen Restaurant „Los Pollos Hermanos“ aus der Serie „Breaking Bad“ nachempfunden. Bei der Konzeptionierung des Projekts einigten wir uns auf dieses Thema und strebten an, das Restaurant so originalgetreu wie möglich zu gestalten. Aufgrund der signifikanten Unterschiede in der benötigten Größe übernahmen wir die ikonischsten Aspekte des Restaurants, um diese trotz des abweichenden Aufbaus in unserem Design wiederzuerkennen. Klassische Merkmale, die integriert wurden, umfassen die Inneneinrichtung, die Wandgestaltung sowie das berühmte Farbschema Weiß-Rot-Blau. Auch die Fensterplatzierung und das große Logo an der Wand wurden nachgebildet. Eines der markantesten Merkmale jedoch ist der Eingang, welcher ein Teil der Gebäudestruktur ist, anstelle einer einfachen Glastür. Die Platzierung der ikonischen Sitzbänke und Tische wurde so gewählt, dass der Spieler vom Küchenbereich aus das gesamte Restaurant überblicken kann, ohne dass Trennwände die Sicht versperren.

8.3 Umgebung



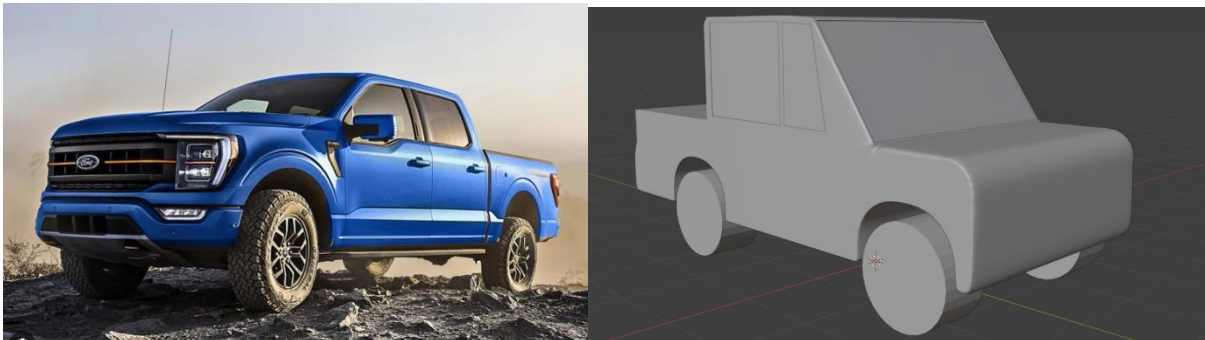
Die Umgebung des Restaurants sollte der eines Fast-Food-Restaurants entsprechen und dies visuell widerspiegeln. Des Weiteren sollte der Standort in den USA angedeutet werden, wo die meisten Menschen mit dem Auto unterwegs sind. Dies erfordert zahlreiche Parkplätze, selbst für ein vergleichsweise kleines Restaurant. Die Flora wurde ebenfalls entsprechend des realen Standorts des Restaurants in der Serie, in Albuquerque, New Mexico, gestaltet, wo die Natur eher karg und spärlich ist.

8.4 Strommasten



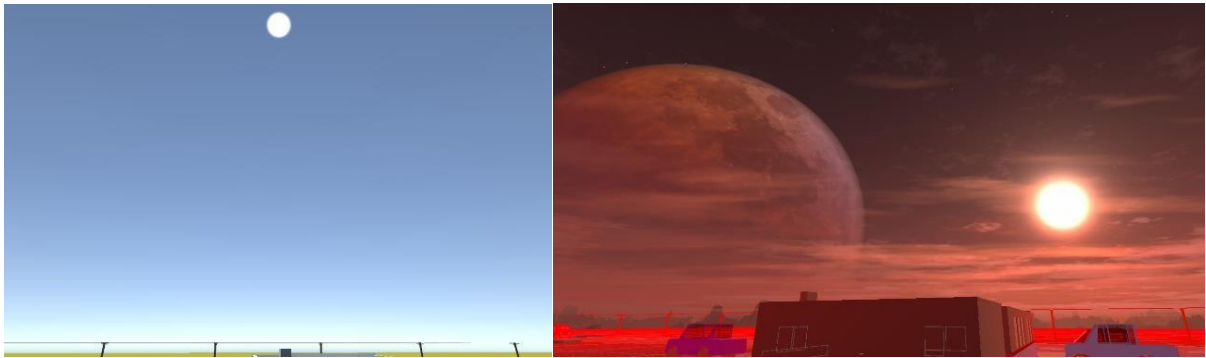
Um der virtuellen Umgebung den authentischen Charme einer Ödlandstraße in Albuquerque, New Mexico, zu verleihen, haben wir Strommasten erstellt, die den realen Strommasten in der Nähe des tatsächlichen Restaurants aus der Serie nachempfunden sind. In ländlichen Gebieten der Vereinigten Staaten verlaufen Stromleitungen typischerweise oberirdisch, statt im Boden verlegt zu werden. Diese Masten bestehen aus Holz und tragen drei Stromkabel, die von einem zum nächsten Mast verlaufen. Unser Modell wurde des Weiteren so gestaltet, dass es nahtlos mehrfach platziert und miteinander verbunden werden kann.

8.5 Ford F150



Um das Restaurant nicht verlassen wirken zu lassen, mussten die Parkplätze mit Fahrzeugen ausgestattet werden. In den Vereinigten Staaten ist der Ford-F150-Truck besonders in ländlichen Gebieten mit schwierigen Bodenverhältnissen weit verbreitet und beliebt. Dieser Truck hat einen derartigen Status, dass er oft als das klischeehafte amerikanische Auto betrachtet wird. Um den Ford F150 in unser auf Poly-Grafik basierendes Projekt einzufügen, haben wir ein Modell erstellt, das dem realen Truck möglichst genau ähnelt. Durch die finale Platzierung der Trucks in der Szene gewann das Restaurant deutlich an Charakter. Die Szene wirkte lebendiger und authentisch amerikanisch, was erheblich zur Immersion beitrug.

8.6 Skybox



In unserem VR-Projekt nutzen wir zwei Skyboxes. Im Standardmodus nutzten wir eine Skybox, die schlicht und einfach gehalten ist, mit blauem Himmel und Sonne, was einen realen, klaren Himmel repräsentiert. Dies hilft dem Spieler, ein Gefühl von Ruhe zu vermitteln. Beim Hell's-Kitchen-Spielmodus hingegen nutzen wir eine Skybox, die stark im Kontrast zur ersten steht. Diese ist düster und surreal, mit ihrem riesigen Mond und rotem Himmel. Durch diese gezielte Differenzierung der visuellen Elemente können wir die emotionalen und atmosphärischen Aspekte des Spiels intensivieren und die Spielerfahrung erheblich verbessern.

9. Dokumentation und Kommentierung

Die Dokumentierung und Kommentierung im Code benötigten ebenfalls Zeit. Dazu kommt die Erstellung dieses Dokuments und die Einbindung relevanter Grafiken und deren Recherche.

4.2 CHRISTELLE MAAß

4.2.1 Überblick

Gesamtzeit: 114 Stunden

1. Projektorganisation	19 Stunden
2. PoC- und MVP-Präsentation	4 Stunden
3. PoC Abgabe vorbereiten und testen	5 Stunden
4. Playtesting und Auswertung	7 Stunden
5. Modellierung der Küche	23 Stunden
6. Modellierung von Zutaten	7 Stunden
7. Modellierung von Utensilien	12 Stunden
8. Modellierung und Animation der Uhr	10 Stunden
9. Implementation des Soßenpools	10 Stunden
10. Anbraten von Patties	4 Stunden
11. Erstellung von Bestellungen	1 Stunden
12. Dokumentation	12 Stunden

4.2.2 Beschreibung

1. Projektorganisation

Es gab regelmäßig Termine zur Projektorganisation. Diese Meetings dienten der Planung des Projekts, der Diskussion von Ideen und der Definition und Verteilung von Aufgaben. Die Projektorganisation fand größtenteils wöchentlich statt und bestand darin, dass wir den aktuellen Stand des Projekts mit dem Rest der Gruppe kommuniziert haben und jeder kurz vorgestellt hat, was er seit dem letzten Treffen umgesetzt hat. Es wurde das weitere Vorgehen besprochen, neue Aufgaben wurden unter den Gruppenmitgliedern verteilt und es wurden gemeinsam Prioritäten gesetzt. Dabei war es nicht schlimm, wenn jemand innerhalb einer Woche gar keine Zeit gefunden hat, an dem Projekt weiterzuarbeiten. Es ging vielmehr darum, den Faden nicht zu verlieren, alle Gruppenmitglieder mitzunehmen und kontinuierlich an dem Projekt zu arbeiten. In die Projektorganisation zählen ebenfalls die Vorbereitungen für Abgaben, wie z. B. das Vorbereiten und Schreiben des Exposés.

2. PoC- und MVP-Präsentationen

Bei der PoC- und MVP-Präsentation haben wir unser Projekt vorgestellt und die Präsentationen der anderen Gruppen angesehen.

3. PoC-Abgabe vorbereiten und testen

Bei dem Proof of Concept ging es darum, erstmal nur die zentralen Mechaniken des Spiels zu implementieren und zu testen, ob diese so funktionieren, wie wir es erwarten. Dieser Schritt ist sehr wichtig, damit man schon am Anfang des Projekts Schwierigkeiten identifizieren und evaluieren kann. Die Kernmechaniken des Burger Makers liegen darin, Zutaten aufeinanderzustapeln und Burger-Patties anzubraten. Wir haben uns gemeinsam im VR-Lab getroffen, um die Kernmechaniken des Spiels zu testen, ein paar Bugs zu beheben und unser Proof-of-Concept abgabebereit zu machen.

4. Playtesting und Auswertung

Ich habe Playtests für die Spiele „Pirate Island“ und „Whack-a-mole“ durchgeführt. Außerdem habe ich zusammen mit weiteren Mitgliedern meiner Gruppe die Playtests für unser Spiel im

VR-Lab begleitet. Ich habe einen Teil der Playtest-Videos zu unserem Spiel gesichtet und ausgewertet.

5. Modellierung der Küche

Meine Aufgabe war es, die Küche in unserem Spiel zu modellieren. Dafür habe ich in Blender zuerst die Küchenmöbel, bestehend aus den Küchenschränken, dem Kühlschrank, dem Herd sowie dem Spülbecken und den Behältern für die Zutaten, modelliert. Da ich zuvor noch keine Erfahrung mit 3D-Modellierung hatte, war dies ein guter Einstieg, um mich damit vertraut zu machen. Schnell hatte ich Spaß an der Modellierung und habe noch weitere Modelle zur Vervollständigung der Küche modelliert. Dazu zählen zum Beispiel die Tür sowie das EXIT-Schild und die Uhr, auf die ich in einem separaten Abschnitt genauer eingehe.



6. Modellierung von Zutaten

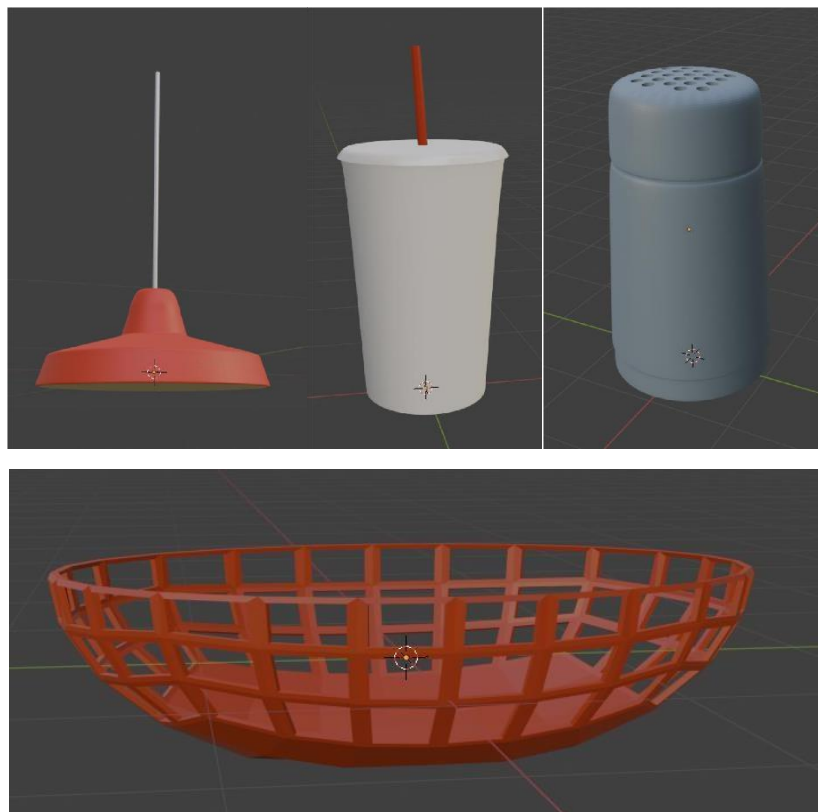
Zusammen mit Jonathan war ich an der Modellierung der Zutaten beteiligt. Ich habe die Tomatenscheibe und die Zwiebelringe modelliert. Bei der Modellierung der Zutaten war es wichtig, einen einheitlichen Stil beizubehalten. Da wir uns für einen Low-Poly-Stil entschieden haben, war uns trotzdem wichtig, dass der Nutzer die Zutaten richtig erkennt und die Modelle nicht zu sehr vereinfacht werden. Das Modellieren der Tomatenscheibe gestaltete sich nicht ganz einfach, da eine Tomate von innen eine eher komplexe Form hat, die entsprechend unserem Stil vereinfacht und trotzdem erkenntlich dargestellt werden musste. Bis das endgültige Modell der Tomatenscheibe entstanden ist, hat es mehrere Überarbeitungen

gebraucht. Zwischenergebnisse wurden dabei im wöchentlichen Meeting mit der Gruppe besprochen und evaluiert, bis schließlich das endgültige Modell der Tomatenscheibe entstanden ist.

7. Modellierung von Utensilien

Wir haben uns zusammen dafür entschieden, ein Restaurant im Stil des Restaurants „Los Pollos Hermanos“ zu modellieren, welches den meisten aus der Serie „Breaking Bad“ bekannt sein sollte.

Dafür hat Sam die typischen rot-blauen Möbel des Restaurants nachmodelliert. Wir haben neben den für das Restaurant essenziellen Möbeln auch ein großes Augenmerk auf die Details gelegt. Dafür habe ich viele kleinere Dekorationen für das Restaurant modelliert. Durch die Dekorationen wirkt das Restaurant trotzdem belebt und wohnlich, obwohl sich keine Personen darin befinden. Zu den Dekorationen, die ich im Restaurant modelliert habe, gehören die Lampe sowie die für Los Pollos Hermanos typischen roten Brotkörbe, die Becher sowie Salz- und Pfefferstreuer, die auf den Tischen verteilt sind.



8. Modellierung und Animation der Uhr

Um die Küchenumgebung noch etwas realer zu gestalten, kam die Idee auf, eine Uhr mit Echtzeit-Integration in die Küche zu implementieren. Dazu habe ich die Uhr zuerst in Blender modelliert und anschließend mit einem Skript animiert. Das Modell der Uhr besteht aus mehreren Objekten, der Uhr selbst sowie den drei unterschiedlichen Zeigern für Stunden, Minuten und Sekunden.

Das Modellieren der Uhr hat mehr Zeit in Anspruch genommen, als ursprünglich erwartet, was unter anderem an der Integration des Ziffernblattes lag. Ich wollte gerne eine fertige Textur als Ziffernblatt verwenden, allerdings hatte ich Probleme, das Ziffernblatt in Blender richtig auf meinem Modell auszurichten und anzuzeigen. Das Anzeigen ist mir lediglich mit einer Textur gelungen, die ein Wasserzeichen enthielt, mit anderen Texturen allerdings nicht. Leider war die Auswahl an kostenlosen Zifferblatttexturen auch nicht gerade groß, sodass ich nur wenige Zifferblätter zur Auswahl hatte, die auch zum Stil unseres Spiels gepasst haben.



Dementsprechend habe ich mich umentschieden und das Ziffernblatt selbst modelliert. Dabei war besonders wichtig, dass die Markierungen auf dem Zifferblatt exakt sind, da der Sekundenzeiger in der Animation die Markierungen auf dem Zifferblatt genau treffen musste und nicht zwischen den Markierungen landen sollte. Dazu habe ich mir einige Tutorials angesehen, wie man Objekte in Blender kreisförmig anordnen kann und habe die Markierungen auf dem Zifferblatt schlussendlich selbst modelliert.

Im nächsten Schritt habe ich die Uhrzeiger modelliert. Der Stundenzeiger und der Minutenzeiger sind bis auf ihre Skalierung identisch. Der Sekundenzeiger hat allerdings eine sehr spezielle Form. Für das Design habe ich mir zuvor verschiedene Uhren und deren Sekundenzeiger angesehen und mich daran orientiert. Bei der Modellierung der Zeiger war außerdem wichtig zu beachten, dass der Pivot-Punkt richtig platziert wird, weil die Zeiger später in Unity um den Pivot-Punkt rotieren. Das Exportieren der Objekte von Blender nach Unity gestaltete sich auch nicht ganz einfach. Beim Exportieren in bestimmten Rotationen trat in Unity ein Gimbal-Lock auf, sodass sich die Zeiger nur um zwei Achsen rotieren ließen.

Nachdem alle Objekte korrekt in Unity importiert waren, habe ich das Skript geschrieben, welches die Zeiger entsprechend der aktuellen Uhrzeit rotiert. Die Rotation der Zeiger sowie die Echtzeit-Integration mit einem Skript zu implementieren, war simpel, nachdem alle zuvor genannten Probleme behoben waren.

9. Implementation des Soßenpools

Beim Verwenden der Soßen hatten wir immer wieder einige Probleme festgestellt. Die Soßen bestehen aus vielen kleinen Kugeln, die beim Drücken der Soßentube nacheinander instanziiert werden. Wenn man die Soßentube im Spiel drückt, sieht es so aus, als würde ein Strahl Soße aus der Tube kommen. Wenn man die Tube schnell bewegt, kann man sehen, dass es sich bei der Soße um einzelne Kugeln handelt. Leider gab es einige Bugs, dadurch dass beim Drücken der Soßentube sehr viele Soßenkugeln hintereinander instanziiert wurden. Diese

traten vor allem auf, wenn man die Soßentube über längere Zeit gedrückt hielt. Dann konnten die Kugeln irgendwann nicht mehr schnell genug instanziiert werden und einige Eigenschaften der Kugeln gingen dadurch vermutlich bei der Instanziierung verloren. Daraus resultierten weitere Probleme, z. B. konnten die Soßenkugeln nicht mehr von dem Burger-Stack aufgenommen werden und sind wieder heruntergerollt.

Um dieses Problem zu lösen, habe ich mich mit Jonathan über den Ursprung des Problems ausgetauscht und anschließend mit Hilfe eines Skriptes einen Soßenpool implementiert, der zum Start des Spiels eine ausreichende Anzahl von Soßenkugeln vorinstanziiert und diese anschließend deaktiviert. Wenn die Soßentube nun gedrückt wird, werden die vorgehaltenen Kugeln aus dem Pool genommen und aktiviert. Der Pool selbst sorgt immer dafür, dass eine vordefinierte Anzahl an Soßenkugeln vorgehalten wird. Das bedeutet, sobald die Soßentube gedrückt wird und Kugeln aus dem Pool genommen werden, instanziiert der Pool neue Soßenkugeln nach. Durch den Soßenpool müssen nun nicht mehr Unmengen an Soßenkugeln auf einmal instanziiert werden, was die Bugs mit der Soße deutlich reduziert hat.

Da ich vor dem VR-Projekt noch nicht mit Unity oder anderen Game-Engines gearbeitet habe, habe ich etwas Zeit benötigt, um mich mit den von Unity bereitgestellten Funktionen vertraut zu machen und die betroffenen Komponenten im Spiel entsprechend anzupassen sowie Verbindungen zwischen den Objekten herzustellen.

10. Anbraten von Patties

Eine der Kernmechaniken, die wir ziemlich früh schon für unser Proof-of-Concept implementiert haben, war das Anbraten des Burger-Patties. Dafür habe ich mich zusammen mit Stefan und Sam im VR-Lab getroffen. Gemeinsam haben wir zwei Skripte geschrieben, die den Bratvorgang eines Burger-Patties kontrollieren: den Roasting Controller und den Roasting Trigger.

Der Roasting-Controller definiert den gesamten Bratprozess. Er enthält einen Timer, der misst, wie lange der Patty schon gebraten wird. Im Roasting-Controller ist außerdem definiert, in welchem Zeitfenster das Patty perfekt gebraten oder verbrannt ist. Abhängig von der Bratzeit wird im Roasting-Controller auch der Farbton des Patties errechnet und in jedem Frame aktualisiert.

Im Roasting-Trigger wird lediglich der Bratvorgang gestartet. Um den Bratvorgang zu starten, muss der Collider des Patties mit einem Objekt kollidieren, welches den Tag „Pan“ hat. In unserem Fall ist dies die Grillfläche.

Ein Patty besteht aus zwei Hälften, welche separat angebraten werden müssen. Es wird immer nur die Seite des Patties gebraten, mit der das Patty auf der Grillfläche aufliegt. Um die jeweils andere Seite des Patties zu braten, muss das Patty zwischendurch gewendet werden. Im späteren Verlauf des Projekts wurden dem Roasting-Controller durch andere Gruppenmitglieder weitere Funktionen hinzugefügt. Es wurde eine Progress-Bar ergänzt, die den Fortschritt des Bratvorgangs für beide Hälften des Patties visualisiert. Soundeffekte für das Braten und ein Signalton, dass das Patty nun perfekt gebraten ist, wurden ebenfalls ergänzt.

11. Erstellen von Bestellungen

Zusammen mit Stefan und Sam war ich daran beteiligt, mir Burgerbestellungen auszudenken, die vom Nutzer im Spiel zubereitet werden müssen.

12. Dokumentation

Die Dokumentation umfasst die Dokumentation von Quellcode sowie die des Projektes in Form des Projektberichtes.

4.3 JONATHAN EL JUSUP

4.3.1 Überblick

Gesamtzeit: 139 Stunden

1. Projektorganisation	17 Stunden
2. Modellierung	8 Stunden
3. Zutaten Implementierung	14 Stunden
4. Stacking Mechanik	12 Stunden
5. Soßen Mechanik	8 Stunden
6. Bewertungssystem	20 Stunden
7. Lobby	4 Stunden
8. Hell's Kitchen	19 Stunden
9. Level Design	8 Stunden
10. Sonstiges	10 Stunden
11. Dokumentation	19 Stunden

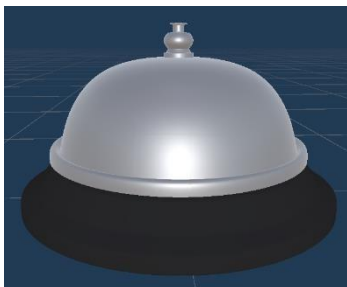
4.3.2 Beschreibung

1. Projektorganisation

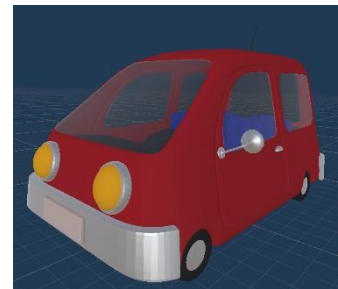
Wir haben uns von Anfang an stets über Aufgabenverteilungen in dazu vorgesehenen Kanälen ausgetauscht. Hierfür wurde hauptsächlich Discord verwendet. Bei Gelegenheit trafen wir uns auch vor Ort im VR-Lab. Hin und wieder haben wir zusammen an bestimmten Funktionalitäten gearbeitet und Aufgaben so verteilt, dass diese gut parallelisiert werden konnten. Dies hat sich als sinnvoll erwiesen, da sich während der Entwicklungsphase keine großen Merge-Konflikte ergaben.

2. Modellierung

Wie meine Gruppenmitglieder habe ich auch einige Modellierungsaufgaben übernommen, um meinen Coding-Anteil einigermaßen ausgleichen zu können. Dafür habe ich einige Zutaten wie den Patty und seine 2 Hälften, beide Buns, das Salatblatt und die Gurkenscheibe modelliert. Zudem konnte ich

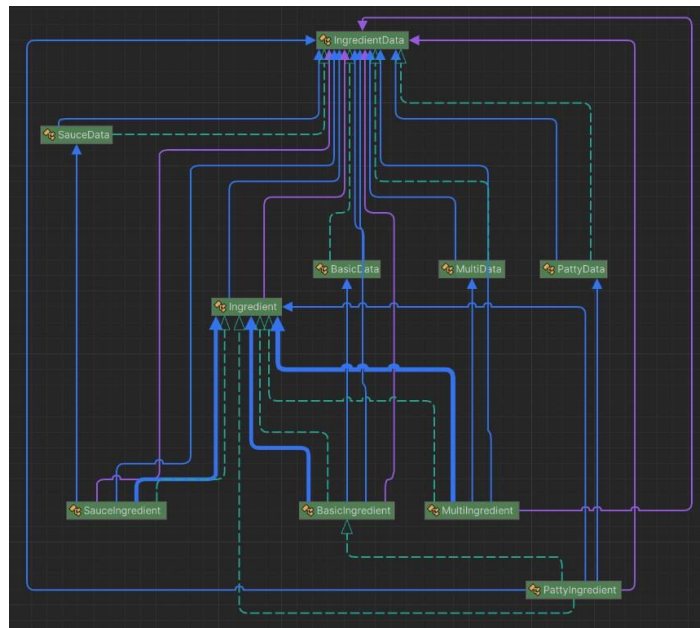


mein Auto-Modellierungsprojekt aus der Vergangenheit importieren und auf dem Parkplatz und der Straße wiederverwenden. Hier und da habe ich noch kleine Modellierungsaufgaben übernommen, wie die Teller auf dem Regal in der Küche, die Mülltonne oder die Klingel, die der Spieler zur Bewertung betätigen kann. Auch die Soßen-Tuben habe ich in ihren Varianten selbst modelliert.



3. Zutaten Implementierung

Von Anfang an war mir klar, dass für die Implementierung der Zutaten eine geeignete Hierarchie gefunden werden musste. Spätestens nach dem PoC spielte nun auch der Typ der Zutat eine Rolle, damit der Stack und somit der Burger analysiert und bewertet werden können. Hierfür habe ich mir erstmal grundlegende Gedanken darüber gemacht, welche Zutaten-Typen es geben soll und welche Eigenschaften sie ggf. teilen. Ich habe mich für 4 grundlegende Zutaten-Typen entschieden: **Basic Ingredient**, **Multi Ingredient**, **Patty Ingredient** und **Sauce Ingredient**.



Diese sind jeweils eine Spezialisierung der Basisklasse Ingredient, wobei Patty Ingredient ein Untertyp von Basic Ingredient ist. Alle Klassen haben ihre eigenen Eigenschaften, die ihren Zutatentyp ausmachen. Darunter zählen beispielsweise verschiedene Metriken, nach denen sie bewertet werden, wie die Platzierung, Anzahl, Anbratzeit etc. Aber auch das Verhalten, wie sie mit dem Stack reagieren, wird dort speziell definiert.

Ein **Basic Ingredient** ist eine einzelne große Zutat, wie ein Top Bun, Salat, Käse, etc. Ein **Multi Ingredient** ist ein Topping, von denen es mehrere gibt. Dazu zählen u.a. Tomatenscheiben, Zwiebelscheiben und Gurkenscheiben. Zu dem **Patty Ingredient** zählt nur der Patty und zum **Sauce Ingredient** zählen die verschiedenen Soßen.

Alle 4 Zutaten Typen besitzen parallel eine Datenklasse, welche für die Bewertung der Zutat verwendet wird, ohne die Zutat selbst adressieren zu müssen. Diese Datenklassen beinhalten abhängig vom Zutaten Typ den Typen selbst, aber auch spezielle Metriken, die zur Bewertung herangezogen werden.

Mir wurde früh klar, dass ich die Toppings nicht wie normale Zutaten behandeln sollte, denn neben der Platzierung auf dem Stack war die Anzahl der Toppings relevant. Natürlich könnte ich jedes Topping einzeln auf dem Stack hinzufügen und bewerten. Dies würde aber die Komplexität und Zutaten-Hierarchie in der Szene explodieren lassen. Dies wäre noch viel schlimmer mit den Soßen, da mehrere 100 Soßenpartikel auf dem Stack liegen würden. Um dies zu lösen, sollten diese Zutaten-Typen die Kollision mit dem Stack anders behandeln. Dafür schaue ich auf die erste neue Zutat, die auf dem Stack gelandet ist und betrachte das folgende Element. Wenn dies den gleichen Typ hat, haben wir einfach mehr von der gleichen Zutat auf dem Stack liegen. Dies kann entsprechend hochgezählt werden und muss nicht die Hierarchie unnötig verschmutzen oder komplexer machen. So muss auch nicht mehr jede Zutat bei der Bewertung gezählt werden, da dies schon am Anfang gemacht worden ist. Dies ist nur eine von vielen speziellen Funktionsweisen der Zutaten, wie sie mit dem Stack interagieren und wie sie bewertet werden.



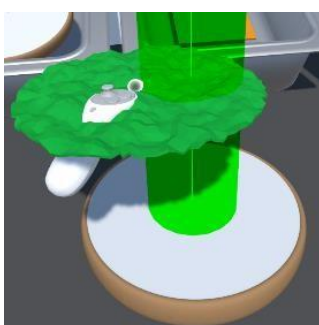
Jede Zutat besitzt einen eigenen Object-Pool. Dieser instanziiert eine Reihe von Zutaten im Voraus und stellt sicher, dass immer eine neue Zutat zur Verfügung steht, wenn man sie braucht. Hierfür habe ich einen bereits modellierten Container zu einem Pool gemacht, aus dem die Zutat entnommen werden kann. Da Object-Pools die Instanziierung am Programmstart schon vornehmen, sollte dies auch zugunsten der Laufzeit kommen.

Eine besondere Zutat ist der Patty. Dieser wurde bereits implementiert, doch habe ich zusätzlich Affordanz-Elemente eingeführt, die es dem Spieler einfacher machen, zu sehen, wie lange eine Patty-Hälfte angebraten wurde. Dies war einer der größten Feedback-Punkte aus den vorangegangenen Playtests. So ändert sich nicht nur die Farbe, sondern ein Ton erklingt auch, wenn die Patty-Hälfte den perfekten Zustand erreicht hat. Dazu habe ich 2 UI-Fortschrittsbalken implementiert, die dem Spieler exakt anzeigen, wie lange die jeweilige Hälfte angebraten wurde.



4. Stacking Mechanik

Das Stapeln von Zutaten auf dem Stack war die wichtigste Mechanik des ganzen Projekts. Dementsprechend floss auch signifikant viel Zeit dort hinein, um sicherzustellen, dass diese Mechanik gut funktioniert, Spaß macht und robust gegenüber nicht vorhergesehenen Interaktionen ist. Das Stapeln von Zutaten auf dem Stack muss demnach in gewisser Hinsicht eingeschränkt werden, sodass dies nur unter bestimmten Umständen möglich ist. Hierzu habe ich eine Linie definiert, welche von der Mitte des Stacks ausgeht. Jede Zutat, die auf dem Stack gelegt werden soll,



muss eine gewisse Distanz unterschreiten, damit diese nicht zu weit vom Stack fallen gelassen wird. Hierzu wird der Abstand der Zutat von der Mittellinie durch Projektion berechnet.

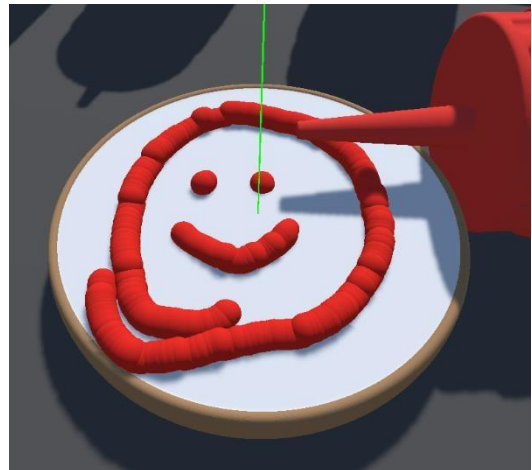
Damit der Spieler eine Zutat auf den Stack packen kann, muss dieser die Zutat über dem Stack halten. Es erscheint eine sog. „Distance-Guide“, welche dem Spieler zeigt, wie mittig er gerade die Zutat über dem Stack hält. Ist diese im grünen Bereich, kann der Spieler die Zutat loslassen. Die Zutat fällt auf den Stack und wird ein Teil

von ihr. Die Distance-Guide bewegt sich immer entlang der Stack-Mittellinie, verschiebt und skaliert sich immer entsprechend der Position der Zutat relativ zur Mittellinie.

Es gibt aber auch weitere Einschränkungen, welche später hinzugekommen sind. Beispielsweise soll es nicht möglich sein, eine Zutat unter dem Stack zu platzieren. Auch die Orientierung des Stacks ist entscheidend. Dieser darf nicht zu schief gehalten werden, sonst fallen Zutaten im falschen Winkel auf den Stack. Damit die Orientierung der Zutaten immer korrekt ist, habe ich die Orientierung gesperrt, sobald die Zutat über dem Stack gehalten wird. So wird sichergestellt, dass die Zutat nicht schief auf dem Stack landet. Ebenso darf aber auch nicht der Stack selbst schief sein. Diese sind einige von mehreren Bedingungen, die erfüllt werden müssen, damit eine Zutat erfolgreich auf dem Stack gelegt werden kann. Wie schon bei der Implementierung der Zutaten erwähnt worden ist, verhält sich jeder Zutatentyp anders, wenn dieser mit dem Stack kollidiert. So ist der Radius, in dem ein Topping auf dem Stack gelegt werden kann, viel größer als bei einer normalen Zutat.

5. Soßen Mechanik

Wir wollten mehr VR-Interaktion bieten. Dabei hat sich die Soße sehr gut angeboten. Dafür haben wir uns die Trigger der VR-Controller zunutze gemacht, um zu steuern, wie groß die Soßenpartikel sind, wenn sie aus der Tube gequetscht werden. Die Interaktion der Soßenpartikel mit dem Stack konnte hierbei ähnlich gehandhabt werden wie die Interaktion der Toppings mit dem Stack. Grundlegend wird die erste kollidierte Kugel zu einem Counter, welcher bei jedem weiteren Soßenpartikel hochzählt. Damit wollte ich erzielen, dass die Zutaten-Hierarchie nicht durch 1000+ Soßenpartikel explodiert, sondern übersichtlich bleibt.



Die eigentliche Herausforderung war, wie man an die Trigger-Werte der Controller drankommt und wie man den linken und rechten Controller voneinander unterscheiden kann. Anfangs habe ich das Drücken der Soßentube durch den Trigger des rechten VR-Controllers gesteuert, war mir jedoch bewusst, dass das keine Endlösung sei. Dieses Problem konnten Sam und ich am Ende doch noch lösen. Dies erlaubt nun auch die gleichzeitige Bedienung von 2 Soßentuben, abhängig von der Hand, die sie hält. Die Soßentube wurde von Christelle später noch weiter optimiert. Hierfür wurde ein Soßen-Object-Pool implementiert, welcher die Soßenpartikel im Voraus instanziiert und bereithält. Dieser Pool wird dynamisch wieder aufgefüllt. Die Implementierung der Soßen-Mechanik war also eine große Teamarbeit.

6. Bewertungssystem

Für das Bewertungssystem muss eine geeignete Datenstruktur für die einzelnen Zutatentypen bereitgestellt werden. Auf diesen Punkt bin ich bereits in Punkt 3 bei der Zutaten Implementierung eingegangen. Demnach besitzt jeder Zutatentyp seine eigene Datenklasse, welche den Typen und sämtliche Metriken beinhaltet. Dies allein reicht aber nicht aus. Das Bewertungssystem muss die Bestellungen und die Zutaten verbinden. Demnach brauchen wir ebenfalls korrespondierende Datenklassen von der Bestellungsseite aus. Wenn beide gegeben sind, können diese verglichen und bewertet werden. Wenn also beispielsweise in einer Bestellung genau 10 Tomatenscheiben gefordert werden, wird das in einer Datenklasse für **Multi-Ingredients** und **Order-Multi-Ingredients** gespeichert. Bei der Bewertung kann dann verglichen werden, wie viele Tomatenscheiben gefordert wurden und wie viele tatsächlich auf dem Burger vorhanden sind. Es gibt weitere Metriken neben der Anzahl der Toppings. Darunter zählen z. B. die Platzierung der Zutat auf dem Stack relativ zu seiner Mitte, der Zutatentyp, die Zeit, die ein Patty angebraten wurde und die Differenz beider Patty-Hälften. All diese Metriken werden miteinander verrechnet, gewichtet und ergeben eine finale Punktzahl.

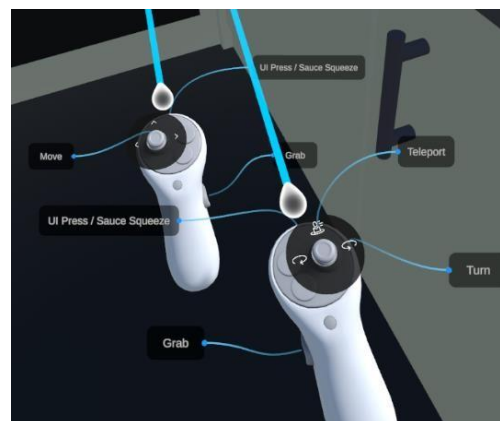


Um einen Burger bewerten zu lassen, muss der Spieler diesen zunächst zubereiten. Diesen Burger zusammen mit der passenden Bestellung legt er dann auf einer dafür vorgesehenen Fläche und betätigt die Glocke. Dies leitet die Bewertung ein. Das Ergebnis wird auf einem Menü angezeigt. Hierbei wird die Bewertung heruntergebrochen auf einzelne Subkategorien. Darunter zählen die Anzahl der Zutaten des Burgers, dessen Reihenfolge und die Bewertung der einzelnen Zutaten entsprechend ihrer Metriken.

Die große Herausforderung war, die einzelnen Metriken angemessen zu gewichten. Beispielsweise soll es nicht so wichtig sein, wie mittig ein Topping auf dem Stack liegt im Vergleich zu einer großen Standard-Zutat. Auch sollten nicht mehr Zutaten bewertet werden, die erst gar nicht in der Bestellung gefordert waren. Diese und viele weitere Spezialfälle werden im Bewertungssystem berücksichtigt. Erst durch viel Testen und durch Anpassung der Gewichtungen konnten wir sicherstellen, dass das Bewertungssystem fair ist.

7. Lobby

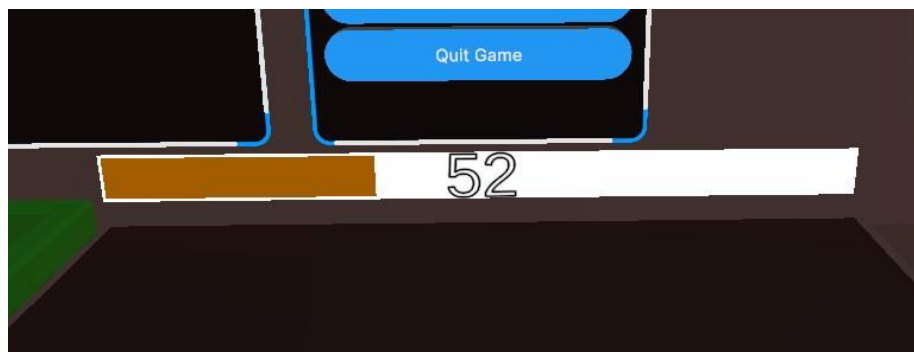
Nach dem MVP habe ich mich an die Lobby gesetzt. Diese sollte die Szene sein, die der Spieler zuerst zu Gesicht bekommt, wenn er das Spiel startet. Ich dachte mir aber, dass die Lobby wunderbar mit unserem (verworfenen) Sandbox-Spielmodus verknüpft werden könnte. So habe ich mich entschieden, die Lobby so zu gestalten, dass diese dem Spieler ermöglicht, alles auszuprobieren. In der Lobby kommen fortlaufend neue Bestellungen rein, die der Spieler ohne Stress abarbeiten kann, um sich mit der Steuerung vertraut zu machen. Er kann aber auch einen eigenen Burger zusammenbasteln, wie er möchte. Neben der Sandbox fungiert die



Lobby auch als Tutorial. Dort werden dem Spieler durch zahlreiche Hinweise das Spielprinzip und die Steuerung erklärt. Außerdem werden die anderen Spielmodi in einem extra Menü beschrieben. Des Weiteren erscheint die Steuerung der Controller, wenn der Spieler auf diese schaut. Somit soll dem Spieler in der Lobby alles Nötige vermittelt werden, was er braucht, um das Spiel zu spielen. Im Vergleich zur ursprünglichen README-Datei ist dies viel intuitiver und angenehmer, um das Spiel zu lernen, indem es in einer sicheren Umgebung direkt getestet werden kann.

8. Hell's Kitchen

Der Hell's-Kitchen-Spielmodus ist ein endloser Spielmodus, in dem Bestellungen fortlaufend reinkommen, während ein Timer abläuft. Ich habe mich dabei um die technische Implementierung des Spielmodus gekümmert. Ziel ist es, so viele Bestellungen wie möglich abzuliefern, bevor der Timer abläuft. Jede erfüllte Bestellung setzt den Timer zurück. Wichtig hierbei ist aber, dass der Timer jedes Mal ein bisschen schneller wird. Der Spielmodus endet also irgendwann. Am Ende wird dem Spieler die Anzahl an Bestellungen angezeigt, die er geschafft hat.



Dies allein reichte uns aber nicht aus. Wir wollten dem Namen gerecht werden und haben uns zusammen dazu entschieden, Gordon Ramsay als Billboard hinzuzufügen, welcher sich im Essbereich aufhält und sich basierend auf dem Timer dem Spieler nähert. Das wollten wir natürlich so ominös wie möglich machen. Und dies war der Punkt, als das Ganze ein wenig in Absurdität ausschweifte. Ich habe mich dazu entschieden, Gordon Ramsay rote leuchtende Augen zu geben, welche die Umgebung beleuchten. In Kombination mit der düsteren Umgebung passte dies perfekt dazu.

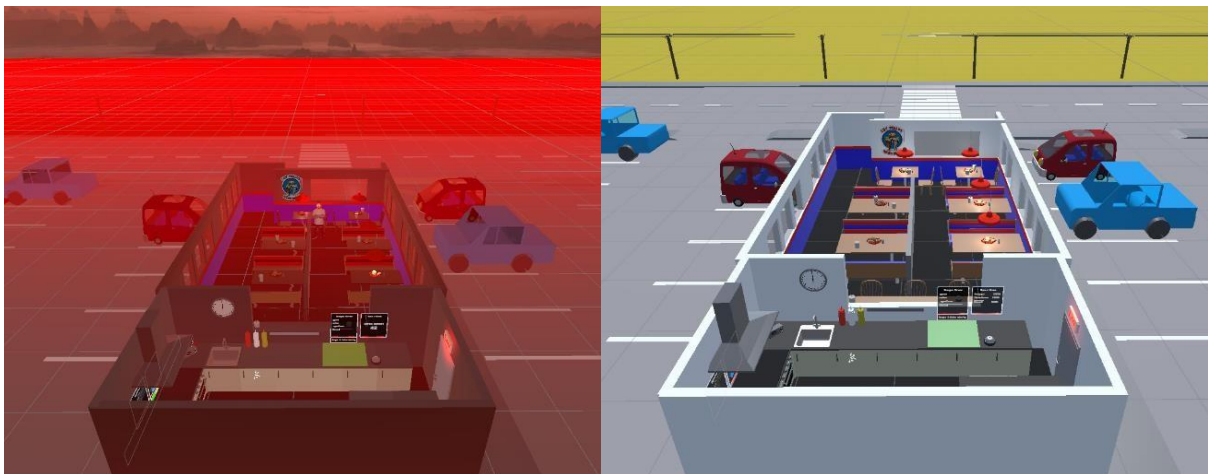
Was die Timer-Mechanik anging, gab es noch einen Punkt, der ausgenutzt werden konnte. Denn zurzeit war es nicht wichtig, wie gut ein Burger abgeliefert wurde. Demnach könnte ein Spieler einen schlechten Burger, bestehend lediglich aus einem Bottom Bun, einfach abliefern und der Timer würde sich zurücksetzen. Dies wollte ich unbedingt vermeiden und kam auf die Idee, einen Bestrafungsfaktor einzuführen. Jedes Mal, wenn der Spieler einen Burger abliefern, dessen Qualität aber einen gewissen Schwellenwert unterschreitet, wird der Timer nicht komplett zurückgesetzt. Wiederholt sich dies, wird dieser Faktor noch größer und der Spieler bekommt immer weniger Zeit. Damit dies dem Spieler nicht nur im



Timer klargemacht wird, habe ich mich wieder Gordon Ramsay gewidmet. Ich wollte, dass er dies dem Spieler auch vermittelt. So habe ich mich kurzerhand dafür entschieden, dass aus seinen Augen Laserstrahlen abgefeuert werden und den schlechten Burger zerstören. Hinzu kommen zufällige Zitate von Gordon Ramsay, die dem Spieler klarmachen, dass der Burger nicht gut genug war. Dafür hat Sam bereits mehrere Zitate von Gordon Ramsay in Form von Audioclips bereitgestellt, welche ich dann nutzen konnte.

So wollte ich Gordon Ramsay zum Kernbestandteil des Hell's-Kitchen-Spielmodus machen. Außerdem warnt er den Spieler, wenn er nicht mehr viel Zeit hat. Im Großen und Ganzen rundet Gordon Ramsay den Spielmodus sehr gut ab. Er dient als Mittel, dem Spieler visuelles und auditives Feedback zu geben. In Sachen Affordanz leistet er einen großen Beitrag. Und gerade wegen dieser Absurdität soll der Spieler daran erinnert werden, dass das Ganze nur ein Spiel ist und ihm nichts droht. Ich finde, dass der Hell's Kitchen Spielmodus sich damit sehr gut vom Standard-Spielmodus abgrenzt, echten Wiederspielwert bietet und schon zum Highlight des Projektes wurde, gerade gegen Ende der Entwicklungsphase hin.

9. Level Design



Ich war ebenfalls am Level-Design der Szene beteiligt. Während es Stefans Aufgabe war, den Essbereich und den Außenbereich samt Parkplatz zu entwickeln, habe ich die Küche so weit zusammengestellt und diese mit der restlichen bereitgestellten Umgebung verbunden. Hin und wieder lieferten meine Gruppenmitglieder weitere Modelle, welche ich dann in die finalen Szenen eingebaut habe. Es war besonders wichtig, alle finalen Szenen von den Inhalten her konsistent zu halten, damit nicht einige Elemente in anderen Szenen fehlten. Dies war ein Grund, warum wir die Umgebung in bestimmte Bereiche gegliedert haben, welche erweiterbar und austauschbar sind.

10. Sonstiges

In diesem Abschnitt möchte ich auf Arbeitspakete eingehen, die nicht wirklich eingeordnet werden konnten und als separate Punkte überflüssig wären. Als ursprünglicher Visionsträger der Projektidee lag es mir besonders am Herzen, dass alle Elemente ein stimmiges Gesamtpaket liefern. Hierfür brauchte ich den Überblick über alle Funktionalitäten und habe dementsprechend überall mitgewirkt. Grob lässt sich dieser Abschnitt in 2 Bereiche gliedern: Extra-Funktionalitäten und Optimierung bzw. Korrektur von bereits bestehenden Funktionalitäten.

Während der Entwicklungsphase bin ich oft umhergesprungen und habe an Teilen von Gruppenmitgliedern gearbeitet, um diese zu optimieren oder auch zu vereinfachen. Dabei handelt es sich sowohl um Funktionalitäten als auch um Kleinigkeiten und Bugfixes. Nur so konnte ich sicherstellen, dass alles so funktioniert, wie wir es uns vorgestellt haben. Darunter habe ich beispielsweise den Soundmanager, welchen Sam implementiert hat, um weitere Funktionen erweitert, dass beispielsweise die Lautstärke über UI-Slider angepasst werden kann oder dass jegliche Soundeffekte räumlich platziert und geortet werden können.

Ich habe aber auch an weiteren extra Funktionalitäten und Elementen gearbeitet. Dazu zählen u. a. ein Küchenfenster-Collider, welcher verhindert, dass der Spieler Bestellungen oder Soßentuben aus der Küche werfen kann und das Spiel somit nicht mehr weitergespielt werden kann. Währenddessen konnten alle sonstigen Zutaten aus dem Fenster geworfen werden, um dem Spieler diese Freiheit zu geben. Außerdem habe ich eine Mülltonne modelliert und funktional implementiert, damit der Spieler falsch zubereitete Burger oder verbrannte Patties wegwerfen kann, ohne diese in der Küche liegen lassen zu müssen. Ein weiteres optionales Arbeitspaket war der Boden, welcher Zutaten aufnimmt, die fallengelassen wurden. Dieser Boden analysiert diese und liefert einen Verschmutzungsgrad der Küche, der in die Bewertung im Standard-Spielmodus mit einfließt. Eine weitere nennenswerte Kleinigkeit waren die beweglichen Autos im Hintergrund der Szene. Wenn der Spieler aus dem Fenster schaut, sieht er Autos in unregelmäßigen Zeitabständen auf der Straße entlangfahren. Dies sollte die Szene dynamischer und lebhafter machen, welche andererseits sehr statisch ist.

11. Dokumentation

Dieser Bereich umfasst jegliche Dokumentation des Projektes und des Codes. Die Dokumentation ist besonders wichtig, da wir so unsere Gedankengänge detailliert darlegen und Designentscheidungen hervorheben können. Des Weiteren haben die Dokumentation und die Lesbarkeit des geschriebenen Codes einen hohen Stellenwert und haben dementsprechend signifikant viel Zeit in Anspruch genommen. Dennoch möchte ich einmal anmerken, dass dieser Bericht die tatsächliche Arbeit und Zeit, die in das Projekt eingeflossen sind, nicht komplett wiedergeben kann. Hierzu muss insbesondere auch auf die Implementierung der einzelnen Funktionalitäten geschaut werden.

4.4 PRINCE LARE-LANTONE

4.4.1 Überblick

Gesamtzeit: 114 Stunden

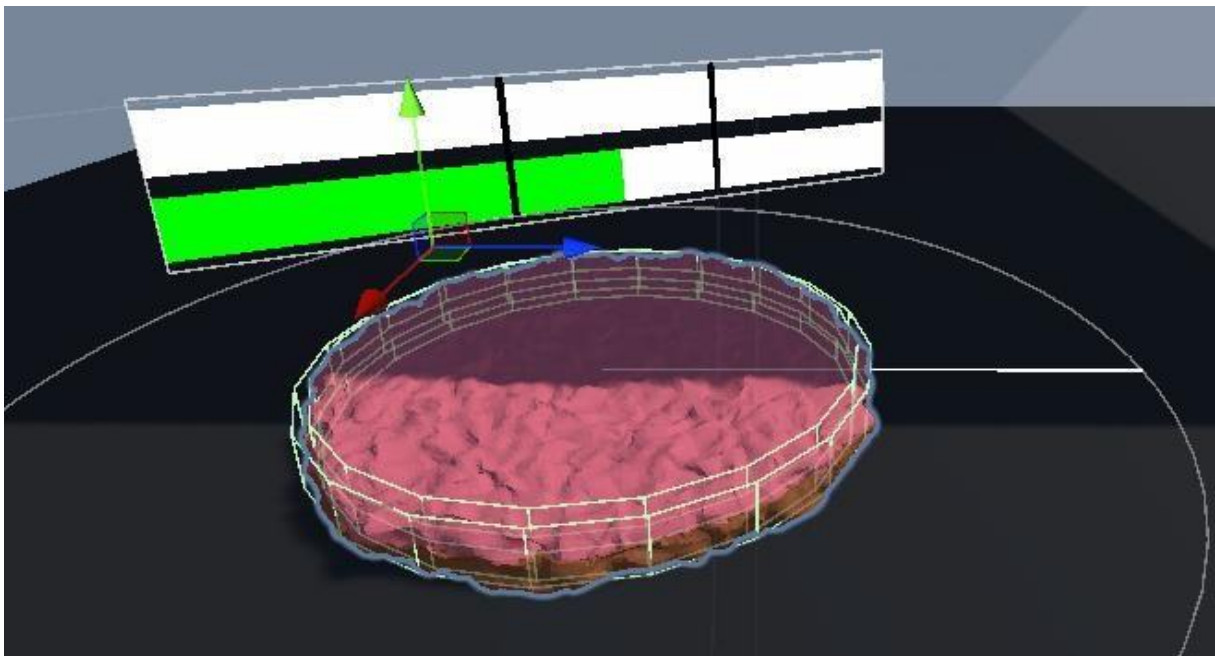
1. Projektorganisation	17 Stunden
2. Braten des Patties	3 Stunden
3. PoC review und Testen	5 Stunden
4. Playtesting	2 Stunden
5. Playtests Auswertung	2 Stunden
6. MVP und PoC Vorträge	4 Stunden
7. Controller Input	14 Stunden
8. Audio	24 Stunden
9. Blender Modellierung	34 Stunden
10. Dokumentation	9 Stunden

4.4.2 Beschreibung

1. Projektorganisation

Im Laufe unseres Projektes haben wir sehr regelmäßig Meetings gehalten, in denen wir unseren Fortschritt, die Aufgabenverteilung, Meilensteine und alles Weitere besprochen haben. Hierbei hat jedes Gruppenmitglied vorgestellt, welche Aufgaben erledigt wurden und wie diese implementiert wurden. Zudem wurden Pläne erstellt, was von wem bis zu welchem Zeitraum erledigt wird und wir haben uns generell über das Projekt ausgetauscht.

2. Braten des Patties



Das Braten des Patties ist eine absolut essenzielle Mechanik unseres Burgermakers und war aus diesem Grund eine der ersten Dinge, die wir für unseren Proof-of-Concept implementieren mussten. Hierzu haben sich Christelle, Stefan und ich zusammengefunden und im VR-Lab ein relativ simples, aber effektives Skript geschrieben. Das Skript besteht aus zwei Hauptkomponenten: Dem Roasting-Controller und dem Roasting-Trigger. Der Roasting-Controller steuert den gesamten Bratvorgang des Burger-Patties. Es enthält mehrere

Funktionen, die den Bratprozess umfassend regeln. Direkt am Anfang haben wir einen Timer, der die Bratzeit vorgibt und entsprechend die Farbe und damit den Zustand des Patties verändert, eingebaut. Sollte der Timer die optimale Bratzeit überschreiten, so brennt das Patty an und wird schwarz. Ansonsten haben wir noch den Roasting-Trigger. Dieser ist dafür zuständig, den Bratvorgang zu starten, sobald der Collider des Patties mit einem Objekt mit dem Tag „Pan“, also Pfanne, kollidiert. Insgesamt haben wir auch direkt darauf geachtet, dass beide Seiten des Patties separat angebraten werden müssen. Diese Funktionalität wurde im Verlauf des Projektes ein wenig überarbeitet, da die ursprüngliche Implementierung nicht vollständig geklappt hat.

Zusätzlich wurden im späteren Verlauf sowohl Soundeffekte für den Bratprozess als auch ein visueller Balken, der den Bratzustand leichter anzeigt, sowie eine Animation für das Braten und eine Interpolation, die zwischen den Farbzuständen interpoliert, hinzugefügt. An diesen Erweiterungen war ich persönlich nur am Sound beteiligt.

3. PoC review und Testen

Zum PoC haben wir uns im VR-Lab getroffen und dort zusammen unseren Proof-of-Concept-Build getestet, Bugs behoben und den Build vervollständigt und abgabebereit gemacht.

4. Playtesting

Das vorgeschriebene Playtesting bei den Gruppen „Zombie Boogey“ und „Pirate Island“.

5. Playtests Auswertung

Ich habe einen Teil der Playtests, die bei uns eingereicht wurden, ausgewertet.

6. MVP und PoC Vorträge

Die Vortragstermine, an denen jede Gruppe ihren PoC bzw. MVP vorgestellt hat.

7. Controller Input

Relativ früh in der Entwicklung trat bei uns ein Problem auf, welches zunächst eine niedrige Priorität hatte, aber später sehr wichtig für das Endprodukt wurde. Durch die erste Implementierung unserer Mechanik, eine Sauce zu greifen und dann mithilfe des Triggers Sauce aus der Tube zu quetschen, war es möglich, mit dem rechten Trigger und nur dem rechten Trigger die Sauce zu aktivieren. Dies war völlig unabhängig davon, in welcher Hand sich die Saucentube überhaupt befand. Das große Problem hierbei war, dass man so nicht zwei Saucen gleichzeitig halten und unabhängig voneinander nutzen konnte. Außerdem ist es extrem unintuitiv, wenn man die Sauce mit der linken Hand hält, diese aber mit der rechten Hand aktiviert wird.

Ich hatte mir vorgenommen, dieses Problem zu beheben, indem ich auslese, welcher Controller die Sauce hält und welcher Trigger eigentlich betätigt wird. Allerdings hatte ich nur eine vage Vorstellung, wie dies zu erreichen sei. Ich hätte nicht gedacht, so lange Zeit verschiedene Ansätze auszuprobieren, bis ich endlich eine Lösung finde. Aber dieses Problem hat mich weit länger beschäftigt, als ich erwartet hatte. Dies lag an verschiedensten Problemen. Zum einen ist die Dokumentation für XRI-Interactables meiner Meinung nach ziemlich schlecht. Es ist sehr schwer, genau die Informationen zu finden, die man sucht und es gibt einige Unterklassen, die dennoch so grundlegend unterschiedlich sind, dass viele Inkompatibilitäten bestehen. Dazu gibt es in Unity auch mehrere Wege, wie man mit den

Controller-Objekten interagiert. Und ich wusste nicht genau, welchen ich nehmen sollte. Mein erster Ansatz war es, via Input Devices die Controller jeweils auszulesen und darüber dann den gedrückten Trigger etc. zu erfahren. Hierbei habe ich ein Tutorial angeschaut, mit dem Ergebnis, dass ich zwei Skripte geschrieben habe, mit denen ich Trigger Values, Velocity und andere Values der Controller zwar auslesen könnte, aber im Endeffekt nicht meinem Ziel, nur den Controller, der die Sauce hält, auch aktivieren zu lassen, näher gekommen bin. Daraufhin habe ich dann gelesen, dass man in Unity den Actionmanager nutzen kann, um Aktionen auszulesen, zu erstellen und weiteres. Hierzu bin ich auf eine Tutorial-Reihe gestoßen, die genau das versprach, was ich gesucht habe. Das Problem war allerdings, dass ich nach ca. 1 h Videoschauen feststellen musste, dass die Tutorial-Reihe unvollständig ist und genau dort aufhört, wo mein Problem behandelt werden würde. Ich weiß, dass ich vielleicht vorher auf Vollständigkeit hätte prüfen sollen.

Im Endeffekt bin ich dann über mehrere Ecken und Wege an dem Punkt gelandet, dieses Problem von der Sauce selbst ausgehend zu lösen. Man kann auf einem XR-Interactable-Objekt relativ leicht auslesen, ob es gerade gehalten wird. Daraufhin kann man sich dann den `firstInteractorSelecting` holen (das ist der Interactor, der das Objekt gegriffen hat) und wenn man diesen (und dieser Schritt hat auch lange gedauert, bis ich endlich diese Lösung gefunden habe) in einen `XRBaseInteractor` castet, dann kann man dessen Namen auslesen und diesen Namen nutzen, um zu vergleichen, ob es der linke oder der rechte Controller ist der das Objekt greift. Den Trigger-Value kann man jeweils recht einfach mit direktem Arrayzugriff auf dem Controller auslesen. Mithilfe dieser Infos war es dann möglich, das Problem zu lösen.

8. Audio

Dies ist mein zweites größeres Spiele-Projekt nach dem Games-Projekt. Allerdings habe ich im Games-Projekt rein gar nichts mit Sound oder Audio zu tun gehabt, deshalb war ich sehr daran interessiert, hier im VR-Projekt die Implementation und Verwaltung des Audios zu übernehmen. Dies stellte einen der Hauptanteile dar, für die ich zuständig war. Generell war das Erste, was ich hierzu implementiert habe, ein Sound-Manager. Dieser enthält zwei Audiosources: **Sound Object** & **Music Object**. Davon ist das **Sound Object** für das Abspielen von Soundeffekten und das **Music Object** für die Hintergrundmusik zuständig. Diese Trennung ist vor allem notwendig, damit man ganz leicht über den Audio-Mixer die Lautstärken separat voneinander steuern kann. Der Sound-Manager besitzt unterschiedliche Methoden, jeweils um einen AudioClip einmal abzuspielen, als Loop abzuspielen, zu stoppen oder die BGM zu starten oder zu stoppen. Außerdem gibt es noch Methoden, die mit dem Lautstärkeregler-UI interagieren, um dort die Einstellungen anzupassen. Sehr früh habe ich Sounds für das Braten der Patties, die BGM (diese wird über den Gamemanager gesteuert) und das Fallenlassen von Ingredients auf den Boden oder den Burger hinzugefügt. Später wurde folgendes ergänzt:

- das Ringen der Klingel
- ein Signalton, wenn der Burger perfekt gebraten ist
- ein extra Ton, wenn man den Hell's-Kitchen-Spielmodus auswählt (in der letzten Version entfernt)
- Hell's-Kitchen-spezifisches BGM
- Das Ausdrücken von Soße aus der Tube
- Laser, der aus Gordon Ramsays Augen schießt, wenn eine Bestellung zu schlecht abgegeben wurde
- Eine Vielzahl an berühmten Gordon-Ramsay-Quotes, die je nach Ereignis in Hell's Kitchen abgespielt werden können

- ein Game-Over-Sound

Hierbei habe ich vor allem die Seite [Freesound.org](https://freesound.org) genutzt, auf der Nutzer selbst aufgenommene Sounds zur freien Verfügung stellen. Oft war es schwer, genau das zu finden, was bei uns passt, denn es ist oftmals gar nicht so genau klar, wie man einen Sound beschreiben würde. Demnach habe ich mir Unmengen von Effekten angehört und das passendste ausgewählt. In vielen Fällen habe ich diese nachträglich noch bearbeitet.

Für unsere Hintergrundmusik haben wir das Breaking-Bad-Lied „Los Cuates De Sinaloa – Negro Y Azul: The Ballad of Heisenberg“ als Karaoke-Version verwendet. Da unser Restaurant ohnehin sehr an das Los Pollos Hermanos aus genau dieser Serie angelehnt ist (mit dem Twist, dass es diesmal ein Burger-Restaurant ist), war dies sehr passend. Das Thema ist außerdem relativ entspannend im Standard-Spielmodus. Für den Hell's-Kitchen-Spielmodus habe ich mir diesen Soundtrack ein wenig vorgenommen und ihn sowohl um ca. 30 % schneller gemacht als auch die Tonlage leicht tiefer geschaltet. Hierdurch wird der Song hektischer und weniger fröhlich, was zu unserem stressigen Setting des Hell's-Kitchen-Spielmodus besser passt.

Besonders gut gelungen finde ich auch die Addition von Gordon-Ramsay-Quotes im Hell's-Kitchen-Spielmodus. Hierzu habe ich eine Compilation mit einigen der bekanntesten Quotes zurechtgeschnitten und diese als Sound-Clips exportiert. Diese werden im Hell's-Kitchen-Spielmodus je nach Inhalt an verschiedenen Stellen gespielt. Um nicht zu sehr repetitiv zu sein, werden diese je nach Aufrufort random aus einer Liste ausgewählt. Natürlich nur jeweils zum Ereignis passend. Das Auswählen, Bearbeiten und Implementieren der Sounds war teilweise relativ aufwändig, aber es macht um einiges mehr Spaß, wenn die auditive Komponente des Spiels ansprechend ist.

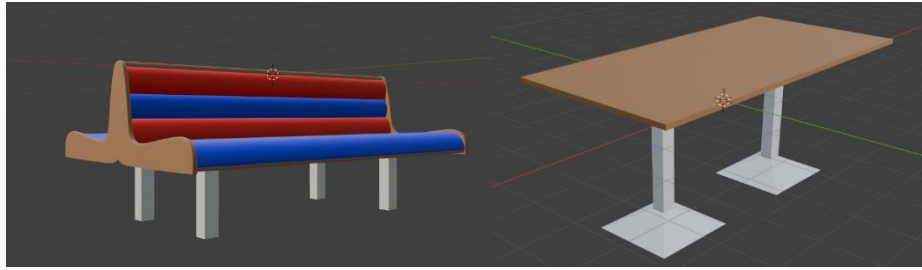
9. Blender Modellierung

Wir hatten uns von Anfang an vorgenommen, die Modellierung in unserem Burger Maker selber zu übernehmen. Deshalb hat Blender bei jedem von uns einen relativ hohen Stellenwert eingenommen. Unsere Idee war es, das Los Pollos Hermanos aus Breaking Bad nachzubauen.



Demnach haben sich unsere Designs für den Standard-Spielmodus hieran orientiert.

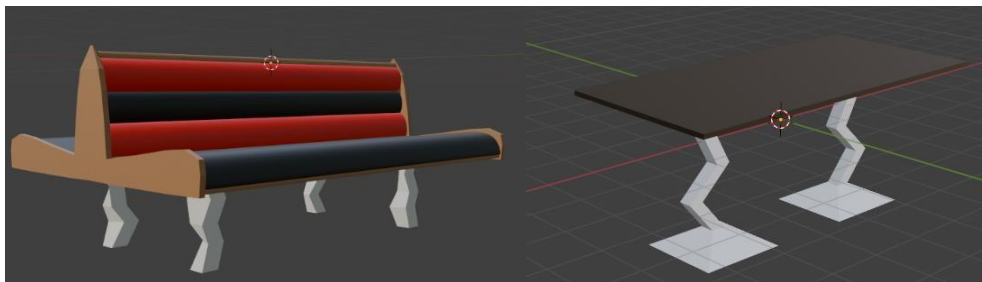
Ich habe primär das Mobiliar innerhalb des Restaurants modelliert. Dazu jeweils verschiedene Arten von Tischen, Bänken und Stühlen. Gerade am Anfang habe ich einige Zeit gebraucht, um wieder in Blender hereinzukommen, und musste mir ein paar Tutorials anschauen, da ich seit Jahren nicht mehr modelliert hatte. Allerdings hat das Modellieren sehr viel Spaß gemacht. Verschiedene Techniken auszuprobieren und wie welches Design funktioniert.



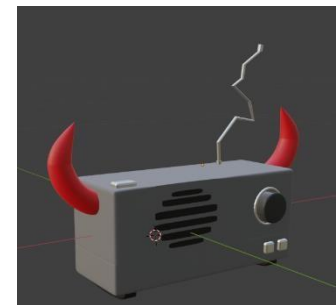
Später wollten wir dann die BGM von einem Radio abspielen lassen. Dieses habe ich dann auch modelliert. Außerdem habe ich das Modell eines Autos, welches ich vor Jahren mal erstellt habe, in abgewandelter Form implementiert, um dieses auf den Parkplatz außerhalb des Restaurants zu stellen.



Mit Blick auf den Hell's-Kitchen-Spielmodus kam dann die Idee auf, nicht nur den Spielmodus zu ändern, sondern auch die gesamte Atmosphäre des Restaurants. Hierzu wollten wir das generelle Layout zwar beibehalten, aber Hell-Mode-Versionen des Mobiliars erstellen. Hierbei habe ich zu jedem meiner Modelle eine weitere Version mit anderer Farbgebung und weiteren Designänderungen erstellt.



Blau wird schwarz, alles ein wenig düsterer und bizarrer. Das Los Pollos Hermanos wird in einer verzerrten Höllenversion dargestellt, die den Stress des Spielmodus darstellen soll. Aufgrund unseres Stils sehen ein paar Dinge sehr niedlich „höllisch“ aus, was zum Witz und Spaß des Spielmodus beiträgt. Besonders diese große Differenz, auch im Visuellen, sollte bei den Spielern zu Überraschung führen und die gesamte Kombination aus Absurdität im Design des Hell's-Kitchen-Spielmodus könnte dem einen oder anderen einen Lacher entlocken. Ein sehr gutes Beispiel, finde ich, ist das Höllenradio.



10. Dokumentation

Das Schreiben an diesem Bericht und das Kommentieren von Skripten haben natürlich auch ein wenig Zeit in Anspruch genommen.

5 PLAYTESTING

Wir haben unser Projekt in verschiedenen Phasen getestet. Neben den obligatorischen Playtests zwischen dem PoC und dem MVP wurden auch danach zwischen dem MVP und der Abgabe weitere Playtests durchgeführt. Die meisten Playtests konnten wir vor Ort im VR-Lab durchführen und konnten unmittelbar während des Playtestens dabei sein. Diese Playtests wurden aufgenommen und extra ausgewertet. Wir möchten uns hier nochmal bei allen Playtestern bedanken:

- Bernhard Hansen
- Felix Türk
- Jonathan Pultz
- Leon Faßheber
- Nadine Jokisch
- Florian Kern
- Valentin Gundlach

Im Folgenden erläutern wir eine kompilierte Liste von Feedback, welches uns in den verschiedenen Phasen der Entwicklung mitgegeben worden ist. Die meisten dieser Verbesserungsvorschläge konnten wir auch zeitgerecht umsetzen. Dies ist nicht das komplette Feedback, sondern eine reduzierte Ansicht, der wichtigsten und nennenswertesten Punkte. Individuelle Auswertungen wurden bereits vorgenommen und verarbeitet.

Einer der größten Punkte war das Spieler-Feedback. Viele haben sich z. B. gewünscht, zu sehen, wie weit der Patty gerade angebraten ist. Außerdem haben sich viele eine Erklärung des Spielprinzips gewünscht, ein Tutorial, was man genau tun soll. Auch wollten die Spieler eine fertige Umgebung sehen und die Soßentuben individuell bedienen können.

Aber auch technische Themen wurden angesprochen, wie die Bugs, die noch bei der Soße aufgetreten sind, und dass man eine Bestellung aus dem Fenster werfen kann und sich damit „soft-lockt“. Ein weiterer Punkt war das Stapeln der Zutaten auf einem schiefen Stack, welches große Probleme bereitete. Ansonsten konnten wir uns aber über relativ wenige Beschwerden über Bugs freuen, deren wir uns nicht schon ohnehin bewusst waren.

Das meiste Feedback war aber überaus positiv. Es hat den Playtestern viel Spaß gemacht und sie haben sich besonders über das Greifen von Objekten aus der Ferne gefreut, sodass man semi-stationär bleiben konnte und sich nicht großartig bewegen musste. Auch die mittige Platzierung von Zutaten wurde als positiv und intuitiv empfunden. Dieses Feedback hat uns bekräftigt und bestätigt, was gut funktioniert und wo es noch Mängel gibt. Zum Ende der Entwicklungsphase konnten wir auch die meisten Verbesserungsvorschläge umsetzen und können zufrieden darauf zurückblicken.

6 KOMPILIEREN

Zum Ausführen der VR-Anwendung wird ein geeignetes VR-Setup mit Headset und Controllern vorausgesetzt. Wir empfehlen primär die Meta Quest 2/3 oder Valve Index. Aber auch die HTC Vive sollte problemlos funktionieren. Die Wahl der VR-Controller ist aufgrund der Interaktion mit Spielelementen signifikant. Am besten eignen sich die Meta-Quest-2/3-Controller, da diese über seitliche Trigger verfügen. Die Controller der Valve Index sollten auch funktionieren, aber das Greifen von Objekten über Kontaktflächen der Controller ist gewöhnungsbedürftig. Die Controller der HTC Vive sind nicht zu empfehlen, da deren D-Pad als Touchpad fungiert und die Steuerung erschwert. Neben dem VR-Setup wird mindestens das Windows-10-Betriebssystem benötigt. Zum Öffnen des Projektes in der Unity-Engine wird die Version 2022.3.21f1 benötigt.

7 EXTERNE ASSETS & REFERENZEN

Soundeffekte & Musik

- Roasting: <https://freesound.org/people/rivernile7/sounds/240546/>
- Bell: <https://freesound.org/people/JohnsonBrandEditing/sounds/173932/>
- RoastingSucces: <https://freesound.org/people/FunWithSound/sounds/456965/>
- Ingredient drop: <https://freesound.org/people/rombart/sounds/186748/>
- Hell's Kitchen violin: <https://www.youtube.com/watch?v=pIDBAh2hkuM>
- BGM: https://www.youtube.com/watch?v=eP7YnUIJ7NE&ab_channel=ObsKureKaraoke
- Gordon Ramsay Quotes: <https://www.youtube.com/watch?v=mdqb3fVqZgM&t=9s>
- Game over: https://freesound.org/people/deleted_user_877451/sounds/76376/
- Laser: <https://freesound.org/people/Corruptinator/sounds/705727/>
- SauceTube: https://freesound.org/people/maarten_wez/sounds/43749/
-

Bilder

- Los Pollos Hermanos Referenz 1: [https://cdn.vox-cdn.com/thumbor/l_tnN7KFGLTenjz0z2QloCUhZDU=/0x0:1800x2700/1920x0/filters:focal\(0x0:1800x2700\):no_upscale\(\)/cdn.vox-cdn.com/uploads/chorus_asset/file/8147793/IMG_9921.jpg](https://cdn.vox-cdn.com/thumbor/l_tnN7KFGLTenjz0z2QloCUhZDU=/0x0:1800x2700/1920x0/filters:focal(0x0:1800x2700):no_upscale()/cdn.vox-cdn.com/uploads/chorus_asset/file/8147793/IMG_9921.jpg)
- Los Pollos Hermanos Referenz 2: <https://th.bing.com/th/id/OIP.kKAGCxxH3R8CITMh5qMNoQHaE8?pid=ImgDet&w=474&h=316&rs=1>
- Ford F150 Truck: https://api.ford-koegler.de/fileadmin/Modellreihe/F150/Lightning/Text-Bild-Ford_F-150_Lariat_Willkommen_in_der_Koenigsklasse-Desktop-2x.jpg
- Los Pollos Hermanos Gus standing: <https://www.videobuster.de/dvd-bluray-verleih/214488/better-call-saul-staffel-3>
- Gordon Ramsay: <https://www.celebrity-cutouts.com/wp-content/uploads/2019/12/gordon-ramsay-white-jacket.jpg>

Referenzen

- Controller Input Tutorial: https://www.youtube.com/watch?v=Kh_94glqO-0

8 BEWERTUNG

8.1 STEFAN PROCIK

Die Entwicklung des VR-Games war eine schöne und bereichernde Erfahrung für mich. Dank der großen Freiheit in diesem Projekt konnte ich lang gehegte Ideen und Träume eines eigenen Spiels in die Realität umsetzen. Schon lange hatte ich den Wunsch, mich intensiver mit dem Bereich der Virtual Reality auseinanderzusetzen, und dieses Projekt bot mir die perfekte Gelegenheit dazu.

Ich habe tiefgehende Einblicke in die Programmierung mittels eines VR-Headsets gewinnen, aber auch die großen Unterschiede zum konventionellen Programmieren von Computergames erkennen können. Im Gegensatz zu traditionellen Spielen, bei denen der Spieler sich mittels Controller oder Maus bewegt, muss bei einem VR-Spiel zu jeder Zeit die vollständige Bewegungsfreiheit des Spielers bedacht werden, was mir als signifikanten Unterschied klar wurde.

Zusammenfassend hat mir die Arbeit an dem VR-Spiel „Burger Maker“ mit meinem Team sehr gefallen. In der Gruppe war stets eine angenehme Arbeitsatmosphäre, in der man sich frei und ohne große Kontroversen bewegen konnte. Dies ermöglichte es uns, bei Unklarheiten jederzeit nachzufragen, ohne dass Streitigkeiten entstanden.

8.2 CHRISTELLE MAAß

Insgesamt hat mir das VR-Projekt gut gefallen. Die Zusammenarbeit mit meiner Gruppe war sehr harmonisch und hat mir viel Spaß gemacht. Alle Gruppenmitglieder waren sehr offen und jeder konnte seine Ideen mit in das Projekt einfließen lassen. Besonders hilfreich fand ich dabei unsere meist wöchentlichen Meetings, in denen wir uns zum aktuellen Stand des Projekts ausgetauscht haben. Dadurch haben wir unser Projekt während des laufenden Semesters nicht aus den Augen verloren.

Zum Start des Projekts hatte ich weder Erfahrung mit Unity noch hatte ich Erfahrungen mit 3D-Modellierung. Das Praktikum „Interaktive geometrische Modellierung“ habe ich parallel zum VR-Projekt belegt. Zu Anfang war das Modellieren noch schwierig für mich, allerdings habe ich recht schnell gelernt. Vor allem, weil ich die Inhalte aus dem anderen Praktikum direkt im VR-Projekt anwenden konnte. Durch den regelmäßigen Austausch mit meiner Gruppe und die Erfahrung meiner Gruppenmitglieder war das VR-Projekt trotz weniger Vorkenntnisse gut zu bewältigen. Allerdings kann ich mir vorstellen, dass das VR-Projekt für eine nur aus Medieninformatik-Studenten bestehende Gruppe etwas schwieriger ist, da man im Studiengang Medieninformatik keine anderweitigen Berührungen mit Game-Engines hat. Privat begeistere ich mich kaum für Computerspiele, trotzdem hat mir das Projekt viel Spaß gemacht und ich bin froh, einen Einblick in die Entwicklung von Spielen bekommen zu haben.

Die Organisation des Projekts hat mir ebenfalls gut gefallen. Ich fand es gut, dass es mehrere Zwischenpräsentationen wie z. B. das PoC und das MVP gab, diese aber nicht bewertet wurden. Dadurch konnte man sich ein Feedback zum aktuellen Stand des Projekts abholen, hatte aber trotzdem keinen großen Druck, dass das Projekt unbedingt einen bestimmten Stand erreicht haben muss.

8.3 JONATHAN EL JUSUP

Mir persönlich hat das VR-Projekt viel Spaß gemacht. Von der Ausarbeitung einer Idee bis hin zur Umsetzung mit all ihren Details. Wir konnten uns auf den VR-Aspekt fokussieren und unsere eigenen Ideen verwirklichen. Auch konnten wir es uns erlauben, ein bisschen abzuschweifen, unserer Kreativität freien Lauf zu lassen. Gerade mit Gruppenmitgliedern, die noch nicht viel Erfahrung mit der Unity-Engine hatten, fand ich es eine gute Gelegenheit, den Fokus auch auf Modellierung zu setzen.

Wir konnten unsere Arbeit einigermaßen gut aufteilen und konnten alles, was wir uns vorgenommen haben, auch umsetzen. Dazu konnten wir auch zahlreiche optionale Arbeitspakete umsetzen, die das Projekt abgerundet haben. Ich hatte besonders Spaß an der Implementierung des Hell's Kitchen Spielmodus'. Auch wenn das erst gegen Ende hin gemacht wurde, konnten wir ein wenig in Absurdität mit Gordon Ramsay abschweifen, ohne den Kern unseres Projektes aus den Augen zu verlieren. Dennoch war die Implementierung der Funktionalitäten sehr fordernd. Wir haben rund 35 Skripte selbst geschrieben, von denen ich die meisten implementiert habe. Dies hat mich schon relativ belastet, doch hatte ich den kompletten Überblick über alles und konnte weitere Aufgaben dementsprechend verteilen. Ich denke, dass es eine gute Erfahrung gewesen ist, auf die ich positiv zurückblicken kann.

8.4 PRINCE LARE-LANTONE

Die Arbeit an diesem VR-Projekt hat mir insgesamt sehr viel Spaß bereitet. Innerhalb unserer Gruppe haben wir uns regelmäßig ausgetauscht und es wurden von jedem konstruktive Ideen eingebracht, die dann in der Gruppe umgesetzt wurden. Der Umgang miteinander war sehr respektvoll und von gegenseitigem Helfen geprägt. Die Arbeitsteilung hat meiner Meinung nach auch sehr gut funktioniert und es gab keinerlei Unstimmigkeiten darüber, wer welchen Teil des Projektes bearbeitet.

Nach dem Games-Projekt hat mir das VR-Projekt die Möglichkeit gegeben, Aufgabenteile, welche ich vorher noch nicht so intensiv behandelt habe, hier umzusetzen. Darunter vor allem viel Modellierung und das Soundmanagement. Durch meine Unerfahrenheit in einigen Bereichen habe ich zu Anfang allerdings des Öfteren eine gewisse Lernkurve gehabt, der Progress wurde aber mit gefestigtem Wissen sehr deutlich.

Mit der Aufgabenstellung, den Fristen und weiteren war ich auch völlig zufrieden. Selbst mit einem vollen Stundenplan konnte man gut an diesem Projekt vorankommen. Zudem war das Feedback der Organisatoren immer sehr hilfreich und konstruktiv, um das Projekt weiter zu verbessern. Man hat auf jeden Fall die Hilfe bekommen, die man benötigt hat.