

One Game A Week

Design Journey

Semester - SoSe2021

Kurs - One Game A Week

Dozentin - Lena Falkenhagen

Alessa Troglauer
Student Number: 82985621

Contents

Einleitung	3
Themes und Spiele	3
One Button – Ralf	3
Destroy – Second Skin	7
End of the world – Sue The Sun	10
Jump – Galaxy Jump	13
In (my) mind - That´s why	16
Physics – Shape Stack	19
Gift A Game – Samaras Offspring	22
Sequel – Point of View 2.5	25
Fazit	28

Einleitung

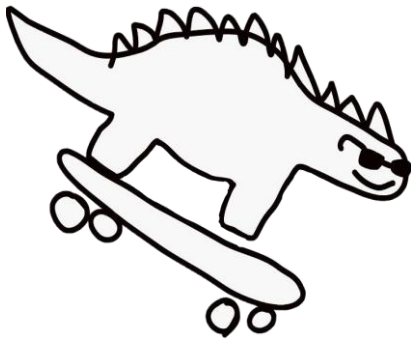
Über den Verlauf des Kurses sind acht Spiele zu vorgegebenen Themen entstanden. Im Folgenden werde ich rückblickend meine Gedanken zur Ideenfindung, dem daraus entstandenen Resultat und den Prozess zu eben diesen darstellen. Zudem werde ich versuchen möglichst reflektiert zu begutachten, was ich aus jedem der einzelnen Projekte für mich mitgenommen habe, womit ich Schwierigkeiten hatte und was verbessert werden muss.

Themen und Spiele

One Button – Ralf

Ideenfindung

Da ich dem Kurs später beigetreten bin, hatte ich gerade für das erste Thema anfangs besondere Schwierigkeiten eine Spielidee auszuarbeiten, weil ich keine der bereits erstellten Spiele meiner Kommilitonen nachahmen wollte. Dementsprechend begann ich erstmal mir weitere Inspirationen im Internet zu suchen, da scheinbar bereits recht viele Game Jams zu dem Thema



stattgefunden haben. Nachdem zwei grundlegende Mechaniken mein Interesse geweckt hatten, kam mir dann jedoch recht spontan die Idee ein Spiel zu einem Skateboard fahrenden Dino zu machen, den ich ein paar Tage zuvor einer Freundin tätowiert hatte. Dieser Charakter, der vorher beim Tätowieren bereits mit dem Namen Ralf versehen wurde, gab mir durch das Skateboard und den einfach zu adaptierenden

Art Style eine gewisse Art von Spiel vor und so entschied ich mich einen 2D Endless Runner anzufertigen. Um dem Spiel ein Ziel bzw. Ansporn und etwas mehr Schwierigkeit hinzuzufügen, beschloss ich einen Highscore für den längsten Run und Hindernisse einzubauen, die die Leben des Spielenden reduzieren. Aufgrund des bereits vorhandenen persönlichen Hintergrunds des Spiels, entschloss ich mich die Hindernisse als Gegenstände darzustellen, die einen Bezug zu uns beiden haben.

Spielkonzept

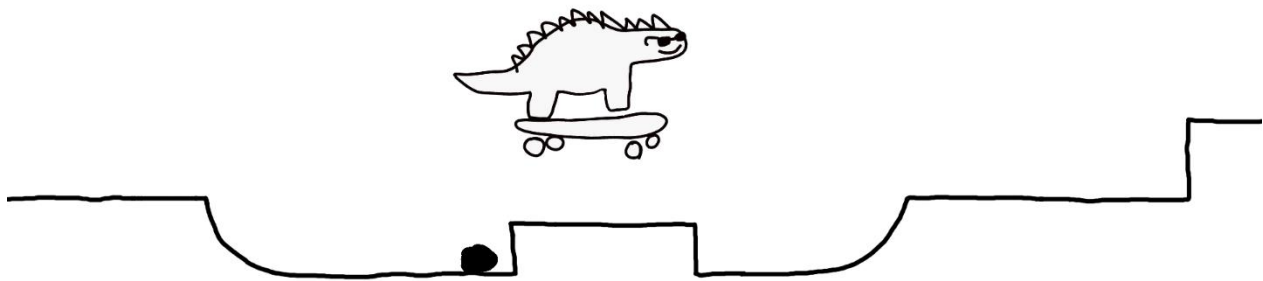
In *Ralf* muss der Spielende als Skateboard fahrender Dino Ralf so lange wie möglich in einem Skatepark Hindernissen ausweichen, um den bestehenden Highscore zu überbieten.

Das Spiel besteht aus einem endlosen zufällig generierten Level, in dem der Spielende den Spielcharakter mit der Space Taste steuern kann. Es stehen hierbei ein Double Jump, sowie verschiedene Jump Höhen, abhängig von der Dauer des Tasten Drucks zur Verfügung.



Your Time: 13,00

HighScore: 25,56



Arbeitsprozess

Für *Ralf* hatte ich aufgrund verschiedener anderer parallel laufender Projekte lediglich einen Arbeitstag eingeplant. Nach dem Aufsetzen des Projekts fing ich zunächst an mich um die Art Assets zu kümmern, da ich als Coder nur schwierig einschätzen konnte, wie viel Zeit diese letztendlich benötigen würden. Ich brauchte für das Spiel verschiedene Enemy Sprites, sowie eine Anzeige für die verbleibenden Leben und vor allem verschiedene Komponenten für die Level Layouts, die am meisten Zeit in Anspruch nahmen, da ich diese immer wieder austesten und anpassen musste und sie deshalb in Unity zusammengesetzt werden mussten. Zudem entschied ich mich auch für die Buttons die Schriften selber zu gestalten, um einen durchgängigen Art Style zu haben und diese nicht ungewollt herausstechen zu lassen. Danach habe ich mich an den Code gesetzt und hierbei mit dem Movement und dem Health System angefangen, da ich dies bereits bei einigen vorherigen Projekten gecodet hatte und somit den

Arbeitsaufwand abschätzen konnte. Danach habe ich mich an den Code für das zufällige Generieren der Level gesetzt, was einfacher und besser funktioniert hat, als ich anfangs gedacht hätte. Dasselbe gilt für den implementierten Highscore, vor welchem ich am meisten Respekt hatte, da ich nur einmal zuvor mit Saving und Loading und somit mit PlayerPrefs gearbeitet hatte. Ich entschied mich aus Gründen der Einfachheit die Zeit eines Timers als Highscore zu benutzen. Ursprünglich war angedacht diesen in eine gefahrene Strecke umzuwandeln, was sich jedoch als recht kompliziert erwies. Als letztes habe ich dem Start Menu noch eine kleine Animation hinzugefügt, damit sich dieses etwas interessanter gestaltet, da es ansonsten eher recht einfach gehalten ist. Für das gesamte Spiel habe ich insgesamt an die 12 Stunden gebraucht.

Ergänzungen

Um das Spiel vollständig über einen Button laufen lassen zu können, war anfangs angedacht auch das Start und Pause Menu über den Space Button zu steuern. Hierfür hätte ich Ralf automatisch von links nach rechts und wieder zurückfahren lassen und ihn dann, wie im Spiel, über die Space Taste in die jeweiligen Buttons springen lassen. Zudem benötigt das Spiel dringend ein Tutorial, da der Spielende momentan einfach direkt in das Level startet. Aus Zeitgründen hatte ich lediglich durch Sprites im Start Menu versucht das Spielprinzip zu erklären, dies erwies sich jedoch nicht als ausreichend. Des Weiteren bin ich mir sehr unsicher, was die Schwierigkeit des Spiels betrifft und würde es dementsprechend gegebenenfalls weiter balancieren.

Learnings

Für *Ralf* hab ich zum ersten Mal mit Random Level Generation in einem meiner Spiele gearbeitet. Wir haben zwar im Rahmen von einem von Da Wilkens' Kursen Zufälligkeit als Thema behandelt und auch prinzipielle Sachen nachprogrammiert, jedoch hat dies vorher noch keine Verwendung in einem meiner Spiele gezeigt, sich jetzt aber als durchaus nützlich erwiesen. Des Weiteren habe ich auch das Erste Mal PlayerPrefs benutzt um einen Highscore zu sichern. Zuvor hab ich immer versucht Saven und Loaden aufgrund von Schwierigkeiten in einem anderen Projekt zu vermeiden.

Für *One Game A Week* habe ich gelernt wie viel Zeit ein Projekt in etwa in Anspruch nehmen kann und wie viel Zeit ich mindestens einplanen sollte. Zudem habe ich gelernt, dass ich prinzipiell dazu in der Lage bin ein Spiel alleine zu erstellen, was für mich persönlich eine gute Erfahrung war.

Schwierigkeiten

Das größte Problem in diesem Projekt stellte für mich die zu knapp bemessene Arbeitszeit da, die ich mir selbst gesetzt hatte, da ich an zwei Projekten parallel arbeitete. Dies führte zu viel Stress, da ich mich um die Art und hier vor allem um ausreichend unterschiedliche Level Layouts kümmern musste, die nicht zu repetitiv sind, damit sich die Wiederholung nicht zu auffällig und somit als zu langweilig für den Spielenden gestaltet.

Bei den eben benannten Level Layouts hatte ich vor allem Schwierigkeiten die Übergänge der einzelnen Layouts flüssig zu gestalten, damit der Spielcharakter nicht hängen bleibt. Das Benutzen der 2D Collider war auch bei den Rampen eine Herausforderung, da ich auch hier darauf achten musste das Ralf dazu in der Lage ist diese zu befahren.

Itch.io: <https://hassavocado.itch.io/ralf>

Destroy – Second Skin

Ideenfindung

Für dieses Theme war meine erste Assoziation ein Spiel zu machen, dass in Richtung Klimawandel oder Umweltzerstörung geht. Jedoch war ich gedanklich sehr eingeschränkt und hatte Schwierigkeiten eine Idee auszuarbeiten, beziehungsweise vor Allem eine Idee auszuarbeiten, die zeitlich machbar ist und das ohne einen Artist. Dann hörte ich allerdings ein Lied der Band *VNV Nation* und beschloss zu versuchen dieses Spiel in Form eines Walking Simulators zu visualisieren. Hierbei diente besonders das Musikvideo derselben Band von dem Lied *Into the Future* als Inspiration, in dem der Sänger zu sehen ist wie er durch verschiedene Stadtteile Tokyos läuft. Also beschloss ich zu versuchen ein Ähnliches Setting mit den Assets, die mir zur Verfügung standen aufzubauen. Während des Erstellens des Spiels kam ich auf die Idee, den Anfang des Liedes zu nutzen, um den Spielenden spielerisch die Mechaniken beizubringen. Um dem gesamten Spiel zwangsläufig ein Ende zu setzen, auch falls der Spielende nicht bis ans Ender der Stadt laufen sollte, implementierte ich einen fahrenden Zug, der auf den Spieler zufährt und zwangsläufig mit diesem kollidiert.

VNV Nation, Into the Future: <https://www.youtube.com/watch?v=SF5mf4LV7Jw>

VNV Nation, Second Skin: <https://www.youtube.com/watch?v=3zFr6XAaKKc>



Spielkonzept

In dem Walking Simulator *Second Skin* erkundet der Spieler, begleitet von dem Lied *Second Skin* der Band VNV Nation eine futuristisch angehauchte und verlassene Stadt.

Das gesamte Spiel beruht auf der Bewegung des Spielenden durch die beiden Level. Das Lied gibt hierbei teilweise den Fortschritt, beziehungsweise die Bewegungen und das Visuelle vor.

Arbeitsprozess

Für dieses Spiel habe ich mich dazu entschlossen zuerst die Walking Mechanik zu programmieren, da der gesamte Fortschritt des Spiels darauf beruht und weiteres Down-Scoping kaum möglich gewesen wäre. Danach habe ich daran gearbeitet das Tutorial für die ersten beiden Verse zu erstellen, was deutlich aufwendiger war als anfangs gedacht und auch deutlich mehr Schwierigkeiten aufwies als zunächst eingeplant. Als dies endlich funktionierte versuchte ich die Belichtung für das Spiel richtig einzurichten und implementierte die zuvor herausgesuchten 3D Assets. Zuletzt baute ich die Stadt fürs zweite Level und fügte die Animation und den Zug für das Beenden des Spiels hinzu. Insgesamt habe ich über einen längeren Zeitraum hinweg immer wieder für kürzere Zeit daran gearbeitet, da ich doch recht unentschlossen und unzufrieden mit dem gesamten Konzept war, was wiederum das Benennen der akkuraten Arbeitszeit als schwierig gestaltet.

Ergänzungen

Wie bereits erwähnt bin ich allgemein sehr unzufrieden mit dem Spiel und würde es wahrscheinlich einmal komplett verändern, wenn es zu einer Weiterentwicklung käme. Prinzipiell würde ich allerdings sagen, dass das gesamte Spiel deutlich interessanter und spannender für den Spielenden gestaltet werden müsste, da es momentan sehr repetitiv ist. Des Weiteren würde ich versuchen es stärker in eine Story einzubinden, beziehungsweise das Gesagte in dem Lied auch in dem Spiel darzustellen und widerzuspiegeln, was bisher nur in ganz geringen Teilen der Fall ist.

Learnings

Ich habe vorher das Audio System von Unity verhältnismäßig wenig benutzt. Zwar habe ich mal einen Sound Manager programmiert, jedoch hierbei keinen direkten Einfluss auf das Spielen und Pausieren der Sound Dateien genommen. Zudem habe ich zum ersten Mal Particle Systems benutzt um Regen darzustellen und konnte hierdurch meine Kenntnisse dazu etwas erweitern.

Abseits davon habe ich gemerkt, dass Walking Simulatoren nicht wirklich ein Genre darstellen bei dem ich Spaß an der Arbeit habe und dass weder Coding- noch Designtechnisch, da mir Storytelling wenig liegt und ich auch wenig Freude daran habe. Des Weiteren habe ich wieder einmal mehr über die Grenzen und Probleme von Version Control erfahren, worüber ich aber sehr froh bin, da es sich um ein Projekt ohne Teampartner handelte und somit keinen Einfluss auf andere Personen oder den Workflow hatte.

Für *One Game A Week* habe ich mitgenommen, dass ich mich, wenn ich in kurzer Zeit alleine an einem Projekt arbeite, lieber auf 2D Spiele beschränken sollte, da die Zeit für die Modellierung für mich persönlich zu knapp ist und die Auswahl im Internet beschränkt ist.

Schwierigkeiten

Als besonders kompliziert stellte sich das Timen des Songs heraus. Abgesehen vom Bemessen der Länge des Levels und das Finden eines Endes, an das der Spielende bestenfalls auch gelangt, hatte ich Schwierigkeiten damit den passenden Code zu finden, um Einfluss auf den Sound zu nehmen und diesen an die Bewegungen des Spielers anzupassen, beziehungsweise diesen von ihm abhängig zu machen. Des Weiteren hatte ich Schwierigkeiten mit dem Belichten der einzelnen Szenen. Vor allem die erste Szene, in der der Spielende ins Licht läuft gestaltete sich als schwieriger als gedacht.

Im Allgemeinen hatte ich Probleme damit Assets zu finden, die mir zusagten und auch mit den jetzt verwendeten bin ich nicht komplett zufrieden. Zudem konnte ich aufgrund der eben benannten 3D Assets, die offenbar eine relativ große Dateigröße aufweisen, meinen Fortschritt nicht mehr auf Gitkraken pushen. Dies führte dazu, sich die Meisten meiner Commits nun auf lokale Ebene beschränken. Zudem stellte sich die Größe meines Spiels dann auch als zu groß *itch.io* heraus, da die Uploads dort auf 1GB beschränkt sind, was mir vorher nicht bewusst war.

End of the world – Sue The Sun

Ideenfindung

Für dieses Spiel stand fest, dass Karam Hamadeh und ich gerne zusammen arbeiten würden. Da wir beide erstmal etwas Zeit für die Ideenfindung benötigten beschlossen wir uns ein paar Tage später nochmal zusammzusetzen. Mein erster Gedanke war *End of the world* wortwörtlich zu nehmen und das Ende der Welt in Form der Flat-Earth-Theorie darzustellen. Allerdings haben Karam und ich diese Idee bei einem Meeting als zu aufwendig und kompliziert empfunden, da wir hierfür wahrscheinlich mehr Kenntnisse in Unity, sowie 3D Assets aus dem Unity Asset Store benötigt hätten, die wiederum nicht unserer Vorstellung entsprachen. Karam kam auf die Idee eine Slingshot Mechanik, im Stile von *Angry Birds*, mit Schwerkraft zu verbinden und diese in eine Galaxie Umgebung einzubinden. Das Planeten Thema assoziierte ich mit einem Interview von Beatrix von Storch, in dem sie erläuterte, wie sie die Sonne verklagen möchte. Daher kombinierten wir beide Eingebungen und beschlossen einen Gravity Shooter mit einem etwas politischeren Thema zu verknüpfen und das Gameplay somit in eine Backstory einzubetten.

Jung & Naiv, Klimaexpertin Beatrix von Storch (AfD) will die Sonne verklagen:

https://www.youtube.com/watch?v=IV8UzT_9bXg&t=15s



Spielkonzept

In *Sue The Sun* muss der Spieler, in Form einer Beatrix von Storch Erde, die Sonne überzeugen weniger zu scheinen, um die Klimaerwärmung zu stoppen. Da diese sich weigert ist Beatrix dazu gezwungen die Sonne zu verklagen und sie so lange mit Briefen zu bombardieren bis diese aufgibt. Dies geschieht mit Hilfe einer Slingshot Mechanik, bei der der Spielende Stärke und Höhe abschätzen muss, um die Sonne zu treffen und ihre Leben zu reduzieren. Das Spiel besteht aus drei Leveln, welche alle unterschiedliche Schwerkraft Eigenschaften aufweisen, an die der Wurf angepasst werden muss. Schafft der Spielende es alle Level erfolgreich zu absolvieren spielt eine Animation ab, bei der die Sonne explodiert und die Erde daraufhin ebenfalls untergeht, um die Abhängigkeit der Erde von der Sonne zu verdeutlichen.

Arbeitsprozess

Zuerst habe ich wie immer das Projekt erstellt und eingerichtet. Die Aufgabenverteilung von mir und Karam sah vor, dass er sich um die Slingshot Mechanik kümmert, während ich den Rest übernehme. Dementsprechend habe ich daraufhin angefangen alle Art Assets herauszusuchen und zu erstellen, was unter anderem das Anfertigen der beiden Planeten umfasste, da dies maßgeblich für unser Spielkonzept war. Anschließend habe ich die bereits erstellten Scripts vom Start und Pause Menu implementiert und die entsprechenden Menüs erstellt und designt. Danach habe ich das eben benannte Interview eingebaut und den Code für den Dialog erstellt. Um diesen etwas ansprechender zu gestalten, habe ich ein paar kleine Animationen hinzugefügt, wie zum Beispiel einen Crossfade beim Laden einer Szene, einen Text Writing Effekt und das Wechseln des Gesichtsausdruck der Sonne sowie eine leichte Bewegung der Erde. Auf Nachfrage Karams habe ich auch am Ende des Spiels eine Animation zu Erde hinzugefügt, sowie mehrere Particle Systems erstellt, die eine Explosion der Sonne ergeben. Zudem habe ich das Health System für die Sonne erstellt und Karams Scripts hinzugefügt. An diesem Spiel habe ich in etwa 10 - 12 Stunden gearbeitet.



Ergänzungen

Was wir gerne noch ergänzt hätten wäre vor allem ein Tutorial gewesen, um den Spielenden besser in das Spiel einzuführen und Verwirrungen zu vermeiden. Des Weiteren hätte ich gerne mehr an den Art Assets, den Animationen und dem Particle System gearbeitet um hier mehr Variation einzubringen und den Allgemeinen Stil zu verbessern. Wovon das Spiel meiner Meinung nach auch noch hätte profitieren können wäre Balancing gewesen, da wir den Eindruck gewonnen haben, dass das Spiel in Teilen zu einfach und in Teilen zu schwierig ist, was eventuell auch auf die fehlende Player Guidance zurückzuführen sein könnte.

Learnings

Sue The Sun war das erste Spiel, in dem ich Unitys Videoplayer benutzt habe. Dies erwies sich als relativ einfach und wird sicherlich Verwendung in zukünftigen Projekten finden.

Des Weiteren habe ich zum ersten Mal in anderer Form mit Particle Systems gearbeitet. Zuvor

habe ich mich im letzten Semester zwar recht intensiv mit eben diesen auseinandergesetzt, allerdings nicht in Form einer Explosion, was sich in der Praxis als recht unterschiedlich erwies.

Abseits meiner Learnings in Bezug auf technische Kenntnisse, habe ich auch gelernt wie es ist mit einem Coder ein Projekt von Grund auf aufzubauen. Davor habe ich zwar mit anderen Codern zusammen an Projekten gearbeitet, jedoch nur an bereits bestehenden. Zudem habe ich wieder einmal erfahren, wie wichtig es ist seine Erwartungen an ein Projekt, dessen MVP und die gemeinsame Zusammenarbeit zu kommunizieren und abzustimmen, um Stress und Frustration zu reduzieren oder gar zu vermeiden. Dazu gehört auch das Bugfixing, wenn möglich, nicht auf den Morgen der Abgabe zu legen und den Workflow dementsprechend anzupassen und eben auch zu kommunizieren.

Für den *One Game A Week* Kurs habe ich mitgenommen, dass es sich als deutlich entspannter gestaltet, wenn man weiß, dass man nicht alleine für ein Projekt verantwortlich ist und es notfalls noch eine Person als Backup gibt. Zudem konnte ich auch deutlich von der Teamarbeit bei der Ideenfindung profitieren, da mir vor allem die Kommunikation mit einer anderen Person sehr geholfen hat.

Schwierigkeiten

Bei diesem Spiel war ich besonders mit dem Pausieren des Video Players in Kombination mit dem Pause Menu am kämpfen. Zudem bin ich nicht sonderlich geübt mit der Arbeit in Photoshop, weshalb ich hier für ein relativ geringes Resultat, verhältnismäßig lang gebraucht habe.

Itch.io: <https://hassavocado.itch.io/sue-the-sun>

Jump – Galaxy Jump

Ideenfindung

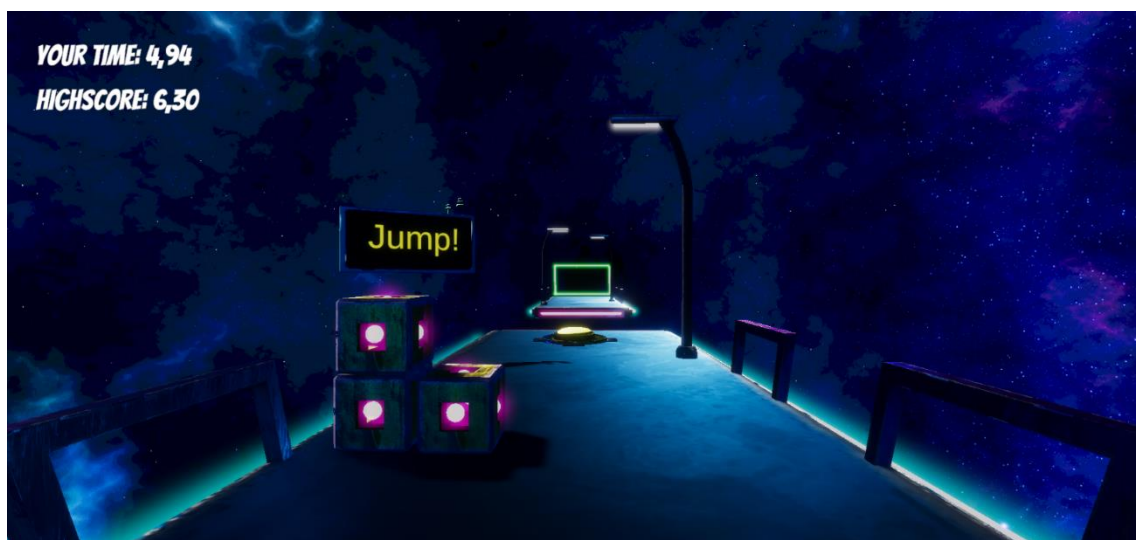
Für dieses Game haben Merve Gümüs und ich zusammengearbeitet. Jedoch waren wir beide nach der Theme Announcement noch relativ plan- und ideenlos. Dementsprechend beschlossen wir zunächst nach weiterer Inspiration im Internet zu suchen. Dabei stießen wir auf ein Video von einem Spiel, in dem der Spielende mit Hilfe eines Jump Pads eine Art Parcours absolvieren und dabei Kollisionen mit dem Boden vermeiden musste. Wir beschlossen die Jump Pad Mechanik zu adaptieren und in ein anderes Setting einzubinden um das Ganze räumlich etwas freier zu gestalten. Bei der Wahl des Settings ließ ich Merve freie Wahl, da ich mich mit 3D Art verhältnismäßig wenig auskenne und ihr somit den Scope selbst überlassen wollte. Sie entschied sich für eine futuristisch angehauchte Galaxie Umgebung, in der die Plattformen in der Luft schweben und der Spielende somit das eigene Abstürzen vermeiden muss. Als noch etwas Zeit übrig war, fügten wir eine kleine Backstory hinzu, um den Player weiteren Ansporn zu geben.

Inspiration: https://www.youtube.com/watch?v=z_XWgeqwSds

Spielkonzept

In *Galaxy Jump* muss der Spielende der Protagonisten helfen bei einem Rennen durch die Galaxie den ersten Platz zu gewinnen.

Ziel des Spiels ist es möglichst schnell durch Jump Pads von Plattform zu Plattform springen, ohne herunterzufallen.



Arbeitsprozess

Die Arbeitsaufteilung von Merve und mir sah vor, dass sie sich um das Heraussuchen und Erstellen der 3D Assets kümmert, während ich das Projekt einrichte und den Code erstelle. Da ich zuvor noch nie mit Merve zusammen gearbeitet hatte, sprach ich mit ihr ab, dass ich versuchen werde schnellstmöglich das Movement und die Jump Pad Mechanik zu erstellen, damit wir für den Fall, dass dies nicht funktionieren sollte eine Alternative erstellen können. Um Zeit zu sparen verwendete ich die First Person Movement Scripts aus meinem Walking Simulator und fügte lediglich das Jumping hinzu, sowie die Funktion für das Jump Pad, was sich als deutlich einfacher herausstellte als zunächst gedacht. Zudem benötigten wir einen Highscore, für die einzelnen Level, was ebenfalls schneller und einfacher funktionierte als ich erwartet hatte. Nachdem Merve alle Art Assets implementiert hatte, übertrug ich die Funktionalität von der Test Szene und erstellte das Start und Pause Menü, während Merve das Level Design für alle Level kreierte und passende Musik heraussuchte. Für dieses Spiel habe ich ungefähr 8 Stunden an Arbeit aufgewendet.

Ergänzungen

Um *Galaxy Jump* weiter zu verbessern, würde ich vor allem das Level Design weiter ausbauen und balancieren, was leider aufgrund fehlender Zeit zu kurz gekommen ist. Des Weiteren würde ich 3D Assets in Form von Hindernissen und allgemein Dekorationen hinzufügen, um mehr Variation und Abwechslung einzubringen. Zudem wäre eventuell ein spielbares Tutorial sinnvoll, um die Spielenden besser in die Mechaniken einzuführen. Obendrein denke ich, dass das Spiel davon profitieren könnte mehr Abwechslung bei der Musikwahl einzubringen, damit sich die Level auch abseits von der Komplexität stärker unterscheiden.

Learnings

In diesem Projekt habe ich zum ersten Mal die HDRP benutzt, um das Volumetric Lighting für das Neon Licht implementieren zu können, was in meinen vorherigen Projekten leider nie funktioniert hat.

Zudem habe ich speziell bei diesem Projekt gemerkt wie sinnvoll und zeitsparend es sein kann, alten Code wiederzuverwenden. Dies mag allerdings auch mit der recht geringen Coding Arbeit in Verbindung stehen. Jedoch fiel mir dadurch auf, wie viel Zeit das Einrichten eines Projekts und dessen Formalitäten, wie ein Start und Pause Menu eigentlich benötigen. Besonders interessant und lehrreich war für mich die Zusammenarbeit mit einer Person mit der ich vorher

noch nie gearbeitet und mit der ich auch vorher keinen bis wenig persönlichen Kontakt hatte. Dies erwies sich jedoch als äußerst positive Erfahrung, vor allem weil ich aufgrund von Merves Spezifikation auf den künstlerischen Bereich nun ein Spiel für *One Game A Week* erstellen konnte, dass mich auch visuell mehr anspricht.

Schwierigkeiten

Das Erstellen des Spiels lief erstaunlich reibungslos und positiv ab. Es gab meinerseits weder große Schwierigkeiten was den Code betrifft, noch was die Kommunikation oder Gruppenarbeit im Allgemeinen betraf.

Einzig das Einrichten der HDRP ergab einige Schwierigkeiten, da hierbei scheinbar ein Bug seitens Unity vorliegt, der verhindert dass das Projekt in einem Ordner mit Unterstrichen, Leerzeichen oder ähnlichen residieren darf. Zusätzlich gab es ein paar Schwierigkeiten bei Merves Version Control, die sich aber nach einiger Zeit lösen ließen und glücklicherweise auch nicht zu einem Verlust unsers Fortschritts führten.

Itch.io: <https://hassavocado.itch.io/galaxy-jump>

In (my) mind - That's why Ideenfindung

Auch für dieses Projekt beschlossen Karam und ich zusammenzuarbeiten. Meine erste Eingebung zu dem Thema beruhte auf einer der Folien der Power Point Präsentationen, in der der Surrealismus erwähnt wurde. Ich musste vor Allem an einige der Werke von René Magritte denken, in denen er mit Proportionen verschiedener Gegenstände spielt und Objekte miteinander vereint, die scheinbar nicht zusammen gehören. Karam hingegen brachte das Thema mit einem Spiel in Verbindung, das einem umgekehrten Mario ähnelte. Sprich Items haben den gegenteiligen Effekt, Plattformen sind tatsächlich keine usw. Dementsprechend beschlossen wir beide Ideen von uns miteinander zu kombinieren und die Mechanik von dem umgekehrten Mario in eine etwas surrealistischere Umgebung einzubauen. Ursprünglich hatte ich ein 3D Setting als passender empfunden, da man so vor allem mit verschiedenen Größenverhältnissen hätte spielen können, aufgrund der begrenzten Zeit und den vergangenen Erfahrungen entschlossen wir uns jedoch ein 2D Platformer zu entwickeln.



<https://comedonchisciotte.org/forum/opinioni/il-distacco-come-arte-superiore-metaphora/>

Spielkonzept

In *That's Why* muss der Spielende verschiedene Level absolvieren, in denen die klassischen Elemente eines Platformers genau den gegenteiligen Funktionen entsprechen.

Das Spiel beinhaltet ein Tutorial Level, sowie ein vollständiges erstes Level. Teil des Spiels ist, dass der Spielende herausfindet welche Plattformen wirklich welche sind und lernt mit den umgekehrten Begebenheiten zu arbeiten.

Arbeitsprozess

Nach dem Einrichten des Repos und des Unity Projekts habe ich mein 2D Movement Script aus dem letzten Semester, an dem ich kontinuierlich gearbeitet habe, hinzugefügt und mit dem Placeholder für die Spielfigur eingerichtet. Danach habe ich einen Teil der Art erstellt, unter anderem den Background, für den ich auch einen Parallax Effekt entwickelte und die URP implementierte, um 2D Lights für die Sonnenstrahlen verwenden zu können. Danach habe ich das restliche Setup fürs Projekt erstellt, damit Karam am Level Design arbeiten konnte. Dies beinhaltete das Health System für den Player, die Menüs, das Erstellen der Items, Enemies und der Platform Prefabs. Danach designte Karam das Tutorial und das erste Level und versuchte das Health System anzupassen. Dies führte leider zu einem Bug beim Laden der Level, da diese abhängig von den verbleibenden Spielerleben sind. Leider schaffte ich es nicht mehr eben diesen am Morgen vollständig zu beheben. Für dieses Projekt habe ich insgesamt an die 8 bis 10 Stunden benötigt.



Ergänzungen

Natürlich würde ich bei weiterer Arbeit am Projekt gerne den bestehenden Bug fixen und wie auch schon in den Spielen zuvor, hätte ich gerne weiter am Level Design gearbeitet, vor allem weil wir hier nur zwei verhältnismäßig kurze Level haben. Obendrein würde ich auch gerne die Sprites überarbeiten, da diese recht zusammengewürfelt aussehen. Hierbei könnte man dann auch versuchen den Hyperrealismus stärker in Kontrast zu der Pixel Art zu setzen. Zudem könnte das

Spiel meiner Meinung nach davon profitieren weitere umgekehrte Mechaniken hinzuzufügen, um das Ganze etwas diverser zu gestalten.

Learnings

Während der Arbeit an *That's Why* habe ich wieder etwas mehr über die Arbeit mit und in Photoshop gelernt und konnte etwas meinen Workflow damit verbessern. Zudem konnte ich nochmal mein bisher gesammeltes Wissen über 2D Lights auffrischen und festigen.

Was sich wieder als besonders nützlich erwiesen hat, war das Wiederverwenden meines Movement Scripts aus dem dritten Semester. Vor allem weil sich dadurch auch die vergangene Arbeit und die kontinuierlichen Verbesserungen an dem Script ausgezahlt haben, da dieses beliebig für meine jeweilige Verwendung anpassbar ist.

Abseits davon habe ich wieder erfahren, wie wichtig Teamabsprachen, regelmäßige Meetings und Kommunikation im Allgemeinen sind. Besonders der unerwartete Versuch des Bugfixings am Morgen sorgte für besonders viel Stress meinerseits und hätte durch bessere Kommunikation vermieden werden können.

Schwierigkeiten

Codetechnisch gab es meinerseits keine Schwierigkeiten, was vor allem an der Wiederverwendung bereits vorhandener Scripts lag. Es gab lediglich eine kleinere Schwierigkeit bei der Benutzung des Parallax Backgrounds, da sich dieser nicht korrekt wiederholte und von einer Stelle zur anderen sprang. Dies ließ sich auch nicht komplett beheben, machte aber aufgrund der relativ kurz gehaltenen Level keinen merkbaren Unterschied.

Itch.io: <https://hassavocado.itch.io/thats-why>

Physics – Shape Stack

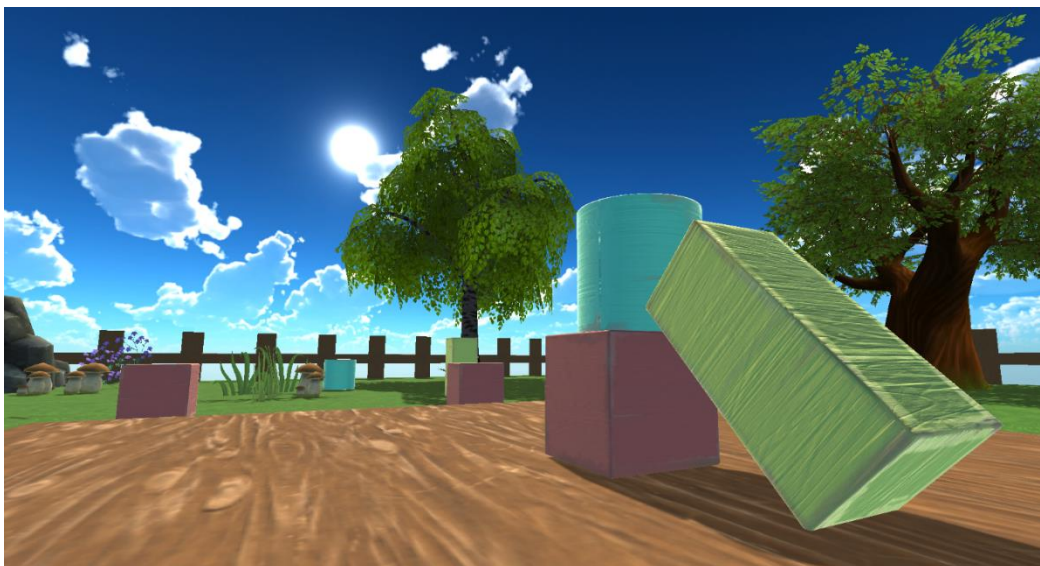
Ideenfindung

Für *Shape Stack* haben Merve und ich beschlossen nochmal zusammen zu arbeiten. Wir beide beschlossen zunächst im Internet nach weiterer Inspiration zu suchen, um uns nötigenfalls daran orientieren zu können. Merve stellte mir dann jedoch eine Art 2D Puzzle Spiel für Kinder vor, in dem es darum ging verschiedene Formen übereinander zu stapeln, um eine gewisse Höhe zu erreichen. Die Idee sagte uns beiden zu und wir entschieden uns dazu ein ähnliches Spiel in 3D und in einem anderen Setting zu gestalten. Hierbei ließ ich Merve wieder freie Wahl und sie kam mit dem jetzigen Design auf, welches mir persönlich ebenfalls für diese Art von Spiel sehr zusagte. Anfangs bestand noch die Überlegung die Bauteile in einem Inventar, zusammen mit einer Nummer für die Anzahl, zu platzieren. Wir entschieden uns aber einerseits aus Komplexitätsgründen und andererseits aus ästhetischen Gründen dagegen, da wir ein spielerisches Aussehen und einen besseren Überblick als sinnvoller erachteten.

Spielkonzept

In *Shape Stack* muss der Spielende versuchen einen Turm aus verschiedenen Klötzen hoch genug zu stapeln, um bis zur angegebenen Markierung und ins nächste Level zu gelangen.

Hierfür liegen Blocks in verschiedenen Formen herum, die auf einer Plattform als Turm aufgebaut werden müssen. Von Level zu Level wird der Schwierigkeitsgrad durch abändern der zu erreichenden Höhe und durch ein anderes Verhältnis der zur Verfügung gestellten Blöcke erhöht.



Arbeitsprozess

Die Arbeitsverteilung verhielt sich ähnlich wie bei *Galaxy Jump*. Merve war für die 3D Assets und für das Level Design zuständig, während ich mich um den Code und das Upsetten des Projekts kümmerte. Ich fing zunächst an die grundlegenden Mechaniken für das Player Movement und für das Dragging und Dropping der Objects zu coden. Danach fügte ich das Kontrollieren der Höhe für die Vollendung der Level hinzu und verbrachte, wie anschließend beschrieben, die restliche Zeit größtenteils mit Bugfixing.

Ergänzungen

Aufgrund des Feedbacks wissen wir, dass das Umwerfen des Turms, einerseits durch den Spieler und andererseits durch die Blöcke selbst, deutlich verstärkt, beziehungsweise überhaupt erst hinzugefügt werden muss. Des Weiteren wurde gesagt, dass die Kameraperspektive teilweise als störend empfunden wird, weshalb wir hier wahrscheinlich nochmal an der Sicht arbeiten müssen. Ich persönlich würde gerne ausprobieren, wie sich das ganze Spiel mit einer dritten Dimension beim Draggen und Droppen verhalten würde und es gegebenenfalls dementsprechend anpassen.

Learnings

Codetechnisch habe ich bei *Shape Stack* zum ersten Mal mit Draggen und Droppen gearbeitet, was zwar nicht zu meiner vollen Zufriedenheit funktioniert hat, aber dennoch lehrreich war. Ansonsten habe ich noch festgestellt, dass gar nicht unbedingt Art für das Start Menu benötigt wird, sondern dass es auch ausreichend sein kann dieses in die Spielwelt einzubinden, was vor allem für zukünftige Projekte ohne Artist nützlich sein könnte.

Besonders bei diesem Spiel ist mir aufgefallen, wie nützlich das Feedback und Testen eines Spieles sein kann. Einige Aspekte waren mir zwar bekannt, andere wiederum sind mir überhaupt nicht aufgefallen, weil ich zu sehr auf die mich störenden Punkte geachtet habe.

Schwierigkeiten

Probleme hatte ich vor allem beim Erstellen des Scripts für das Draggen und Droppen von Objects. Besonders ärgerlich ist hierbei für mich, dass sich das Draggen und Droppen nur auf zwei Achsen beschränkt und die Tiefe nicht mit einbezieht. Zudem haben die Objekte aus mir bis heute nicht bekannten Gründen angefangen zu rotieren, sobald man sie festhielt. Dies führte dazu, dass das saubere Ablegen dieser Gegenstände kaum noch möglich war und somit eben auch nicht das aufeinander Stapeln. Daher entschloss ich mich dazu die Rotation von Objekten

auszustellen, was wiederum dazu führte, dass der Turm in den meisten Fällen auch nicht mehr umkippte. Hinzu kam auch noch, dass einige Schwierigkeiten beim Feststellen, ob der Turm tatsächlich hoch genug ist, auftraten. Zum einen konnte der Spieler die Blöcke zuerst einfach in den Collider halten, um das Level abzuschließen und andererseits fing der Timer nur manchmal an herunterzuzählen. Diese beiden Bugs ließen sich aber glücklicherweise nach etwas Arbeit beheben.

Itch.io: <https://hassavocado.itch.io/shape-stack>

Gift A Game – Samaras Offspring

Ideenfindung

Für Samara´s Offspring haben Kai Heide zusammen gearbeitet. Uns beiden ist jedoch anfangs keine spezifische Person eingefallen, der wir gerne ein Game schenken würden. Ich hatte vor allem Schwierigkeiten damit, da ich bereits Ralf einer Person „geschenkt“ hatte und ich daher gedanklich noch sehr an diesem Spiel festhielt. Kai schlug dann jedoch vor ein Spiel für seine Mutter zu entwickeln, in dem der Spielende Samara aus dem Film *The Ring* bei der Geburt assistiert. Hierfür sollte der Player eine gewisse Zeit überstehen und parallel dazu Aufgaben vollenden. Das Setting war durch die Hauptfigur somit auch schon weitestgehend vorgegeben.

Spielkonzept

In dem Spiel *Samaras Offspring* muss der Spielende Samara helfen ihren Nachkommen zu gebären und die Geburt zu überstehen, indem verschiedene Aufgaben erfüllt werden.

Während der Timer zwei Minuten lang herunterzählt muss der Balken auf der linken Seite durch Drücken der Leertaste in einem bestimmten Bereich gehalten werden, um die Geburt so schmerzlos wie möglich zu gestalten. Geschieht dies nicht reduzieren sich Samara´s Leben. Jedoch schlägt dieser Balken aufgrund der Wehen regelmäßig aus und muss von dem Spielenden wieder eingepegelt werden. Zur Beruhigung verlangt Samara in gewissen Abständen nach ihrer Kreissäge, die ihr gereicht werden muss. Abhängig von Erfolg oder Misserfolg dieser Aufgabe wird die Toleranz des Balkens angepasst. Übersteht Samara die zwei Minuten wurde das Spiel, beziehungsweise die Geburt, erfolgreich abgeschlossen.



Arbeitsprozess

Die Aufgabenteilung von Kai und mir sah vor, dass er sich um die künstlerische Gestaltung des Projekts kümmert, sprich 3D Assets Animationen, Sound etc. und ich mich um das Codetechnische. Dieser war für mich bei diesem Projekt wahrscheinlich einer der Aufwendigsten des Kurses, da ich abseits meiner Menu Scripts und meines Dragging und Dropping Scripts keinerlei Code wiederverwenden konnte und dementsprechend alle Mechaniken von Grund auf neu programmieren musste. Ich fing damit an die Scripts für Thresholds zu erstellen, welches auch das zufällige Ausschlagen der Bar umfasste. Danach arbeitete ich an der Aufgabe für das Reichen der Kreissäge, welche mit Abstand am meisten Zeit benötigte und für am meisten Bugs sorgte. Zudem stattete ich für Kai das Projekt mit der URP aus, um schönere Grafiken zu ermöglichen und implementierte seine zuvor aufgesetzten Animationen in den Code. An *Samaras Offspring* habe ich an die 14-15 Stunden gearbeitet.

Ergänzungen

Leider haben wir während der kompletten Entwicklung des Projekts vergessen ein Tutorial zu implementieren, weshalb ich lediglich ein paar Sätze auf der itch.io Seite hinzugefügt habe, um wenigstens die grundlegenden Mechaniken zu erklären. Bei einer Weiterentwicklung des Spiels, wäre ein Tutorial aber wahrscheinlich eine der ersten und wichtigsten Ergänzungen, die ich hinzufügen würde. Zudem würde ich gerne die Soundeffekte, die Kai mir herausgesucht hat einbinden, da ich das aus Zeitgründen nicht mehr geschafft habe. Was mir persönlich auch noch wichtig wäre ist die Ausarbeitung des Codes, da ich hier viele Sachen eher unschön lösen musste und ich denke, dass das auch deutlich einfacher zu handhaben wäre.

Learnings

In diesem Projekt habe ich zum ersten Mal mit 3D Animationen gearbeitet und mit Animationen, die nicht von mir selbst erstellt wurden, da bisher immer ich das Animieren in Projekten übernommen habe. Zudem habe ich durch Kais Arbeitsweise gelernt, wie gut regelmäßige Meetings und Absprachen tatsächlich sind und auch wie hilfreich es ist, wenn man weiß wie weit die andere Person mit der Arbeit ist und wie gut oder schlecht sie vorankommt. Daher denke ich, dass ich auch in Zukunft mein Verhalten anpassen werde und meinen Fortschritt deutlicher kommuniziere, da ich das Gefühl habe, dass dies in diesem Semester eher weniger der Fall war.

Schwierigkeiten

Leider mussten wir kurz vor der Abgabe des Projekts feststellen, dass einige Animationen von Kai nicht richtig dargestellt wurden. Wir haben uns dann jedoch letztendlich dazu entschieden es dabei zu belassen, da dies beim Spielen kaum auffällt. Problematisch war auch der Coding Umfang des Projekts, da wir, wie eben beschrieben, sehr viele verschiedene Mechaniken im Spiel implementiert haben, die größtenteils alle neu gecodet werden mussten. Vor allem das Reichen der Kreissäge führte zu immer weiteren Problemen, unter anderem auch da wir zwei Runden implementieren wollten und es dies so zu Problematiken mit bereits gesetzten Variablen führte.

Itch.io: <https://hassavocado.itch.io/samaras-offspring>

Sequel – Point of View 2.5

Ideenfindung

Bei diesem Theme entschloss ich mich dazu das Spiel von Karam Hamadeh aus der Physics Woche zu nehmen und weiterzuentwickeln, beziehungsweise abzuändern. Meine ersten Ideen hierfür waren vor allem an der Grafik des Spiels zu arbeiten und dem ganzen auch hier ein neues Theme zu geben. Zudem wollte ich das Design der Menus aufbauen und erweitern und dem ganzen Spiel Sounds hinzufügen. Des Weiteren nahm ich mir vor einen bestehenden Bug zu fixen, der verursachte, dass der Jump nach dem Wechseln der Schwerkraft nicht der Stärke des Jumps vor dem Wechsel entsprach.

Niklas Tarnowski beschloss sich diesem Projekt anzuschließen und wir legten fest, dass wir das Spiel in ein Sci-Fi Setting einbetten wollen. Zunächst wollten wir *Point of View* in 2D lassen, nach einen Blick in den Asset Store stellten wir aber fest, dass es verhältnismäßig viel Auswahl für 3D Sci-Fi Assets gibt und wir entschieden uns dazu das Spiel in 2.5D zu gestalten, um einen deutlicheren Unterschied zu der erstellten Basis von Karam herzustellen.

Spielkonzept

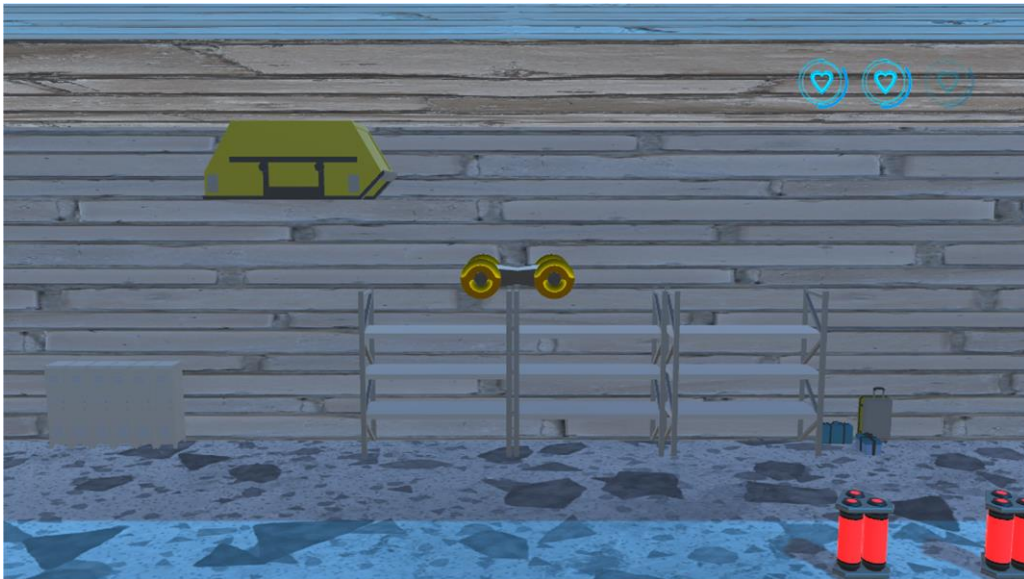
Point of View 2.5 ist ein 2.5D Sequel des Spiels *Point of View* von Karam Hamadeh, in dem der Spielende durch das Switchen der eigenen Gravity Hindernissen ausweichen muss, um das Ende des Levels zu erreichen.

Das Spiel beinhaltet hauptsächlich drei Komponenten, auf denen das Spiel aufbaut. Den Gravity Switcher, der die Schwerkraft des Playercharakters umdreht und ihn an die Decke oder Boden befördert, Stachel ähnliche Objekte, die die Leben des Players bei Kollisionen reduzieren und Power-ups, die die Leben wieder regenerieren.

Arbeitsprozess

Die Aufgabenverteilung von Niklas und mir sah vor, dass ich mich um das Programmieren und Einrichten des Spiels kümmere, während Niklas das Level- und Sound Design übernimmt. Da wir uns dazu entschieden hatten aus einem 2D ein 3D Spiel zu machen, war ich dazu gezwungen das gesamte Movement für den Spielcharakter neu zu coden. Dies war auch das Erste was ich machte, da ich vorher noch kein 2.5D spiel programmiert hatte und ich mir dementsprechend unsicher war wie komplex dies ist. Dazu habe ich auch den Code für das Umkehren der Schwerkraft gemacht, da dies selbstverständlich, neben dem Movement, das Wichtigste Element

des Spiels ist. Danach fügte ich die Cinemachine, das Health System und die bereits bestehenden Scripts für die Menus hinzu. Daraufhin erstellte ich auch schon mal das Design des Start Menus, sowie des Pause Menus. Damit Niklas sich einfach im Projekt zurechtfinden und schnell mit dem Level Design beginnen kann, importierte ich auch schon mal Packages aus dem Asset Store und erstellte Prefabs für alle wichtigen Komponenten des Spiels, wie den Spikes und den Power-ups. Für die final hochgeladene Version habe ich in etwa 9 Stunden benötigt.



Ergänzungen

Abseits des Umstellens der Dimension fiel es mir tatsächlich schwer weitere Ergänzungen für die Mechaniken zu finden, da mir die bereits bestehenden als sehr sinnvoll und für die Größe auch als ausreichend erschienen. In der finalen Version, die auf itch.io hochgeladen wurde, fehlt jedoch das Start Menu, welches zwar theoretisch vorhanden wäre, es jedoch nicht mehr in den Build geschafft hat. Zudem könnte das Spiel bei weiterer Ausarbeit noch weitere Level vertragen und die Bewegung der Kamera müsste weiter angepasst werden.

Learnings

Point of View 2.5 war das erste Spiel für mich, in dem ich mit dieser Dimension gearbeitet hab. Dementsprechend konnte ich auch beim Coden des Movements einiges neues Lernen, vor allem da ich meine Scripts für verschiedene Arten von Movements soweit bereits zusammengestellt

habe, hierfür aber noch nicht. Des Weiteren habe ich auch zum ersten Mal das Cinemachine Package von Unity benutzt, dessen Setup sich als einfacher erwiesen hat als zunächst angenommen, obwohl ich definitiv noch einige Zeit brauchen werde, um hiermit ein vernünftiges Ergebnis erzielen zu können.

Weiterhin habe ich gelernt, wie es sein kann in bereits bestehenden Projekten von anderen Personen zu arbeiten und auf diesen aufzubauen. Dies stellte sich allerdings nicht als allzu große Herausforderung heraus, da mir die meisten Scripts bereits bekannt waren. Letztendlich habe ich wieder gelernt wie wichtig die Kommunikation unter Teammitgliedern ist, vor allem wenn es um den Workflow geht. Dadurch hätte zum Beispiel das Fehlen des Start Menus verhindert werden können und das Spiel wäre als insgesamt abgerundeter erschienen.

Schwierigkeiten

Technische Schwierigkeiten hatte ich vor allem bei dem Verwenden der Cinemachine, da ich nicht herausfinden konnte, wie ich das Hochschweifen beim Umkehren der Schwerkraft vermeiden kann, was dazu führte, dass der Spielcharakter nur noch schwer zu sehen war.

Ansonsten gab es kaum Schwierigkeiten was das technische betrifft, da es sich, abgesehen von dem Movement Script, wieder um Code handelte, den ich bereits mehrfach verwendet habe.

Fazit

Über den Verlauf des Kurses habe ich besonders das schnelle Implementieren grundlegender Mechaniken erlernt, um gerade anfangs schnell voranzuschreiten. Zuvor habe ich immer relativ viel Zeit damit verbracht eher unwichtige Sachen zuerst einzufügen und mich damit eher selbst aufgehalten.

Designtechnisch habe ich gelernt anfangs weniger komplex zu denken und mich auf die Sachen zu konzentrieren die ich innerhalb einer bestimmten Zeit erreichen kann, anstelle mich in bestimmte Ideen zu verstricken. Ich habe zudem einen Weg für mich gefunden, um mit Spielkonzepten aufzukommen und habe gemerkt, dass mir hier vor allem der Austausch mit anderen Leuten hilft. Ich habe vor allem auch gelernt meine eigenen Fähigkeiten und Kenntnisse besser einzuschätzen und kann nun deutlich besser beurteilen, was ich innerhalb einer gewissen Zeitspanne erreichen kann und was nicht.

Aufgrund dieses Kurses und den Vorteilen der Teamarbeit, habe zum ersten Mal mit Kommilitonen gearbeitet, mit denen ich vorher keinen Kontakt hatte. Hieraus konnte ich vor allem einen Einblick in die verschiedenen Arbeitsweisen von Personen gewinnen und auch für mich Dinge mitnehmen um meine eigene Arbeit in Gruppen zu verbessern.

Des Weiteren konnte ich für mich persönlich feststellen, welche Art von Spielen mir Spaß machen und an welchen Projekten ich gerne arbeite und habe ein weitergehendes Verständnis für Bereiche der Spielentwicklung auch abseits des Programmierens erlangt.

Insgesamt war der Kurs zwar sehr anstrengend und zeitaufwendig, aber aufgrund der zuvor genannten Aspekte auch definitiv einer der wertvollsten des Semesters.