



XCLIMB: WER RUNTERSCHAUT VERLIERT!

„The Casual-Contest for everyone! “

Game Design Document
Copyright: © 2021 GE-Lab
Erstellt von: Funsi
Version: 27.10.2021

Inhalt

Überblick.....	4
Essenz XClimb	4
Winning Condition.....	4
Kurzfassung.....	4
Geplantes Spielerlebnis	4
Zielgruppe.....	4
Slogan	4
Ansätze zur Umsetzung.....	5
Die Kletter Mechanik.....	5
Self-Balancing.....	5
Konkretes Self-Balancing	6
Aufbau der Kletterwand	6
Spieler Interaktionen	7
GUI	7
GUI für Bewegungen.....	7
GUI für das Level	7
HUD	8
Herunterfallen //Update	8
Griffe.....	8
Der Splitscreen //new.....	9
Modi //New	10
Events //New	11
Arten von Events	11
Timing von events	12
Power-Ups //NEW	13
Power-Ups aufsammeln	13
Verschiedene Power-Ups.....	13
Power-Up Distribution	14
Andere Level-Elemente //TO-DO	15

XClimb: Wer runterschaut verliert!

Verschiedene Kletterer	//TO-DO.....	15
Kletter-Animation	//TO-DO.....	15
Stages	//TO-DO.....	15
Lobbys	//TO-DO.....	15
Streak	//TO-DO.....	15
Setting	//TO-DO.....	15

ÜBERBLICK

ESSENZ XCLIMB

XClimb: Wer runterschaut verliert!

WINNING CONDITION

Das Ziel ist denkbar einfach: Der Erste **ganz oben** zu sein.

KURZFASSUNG

In XClimb tritt man im direkten **Rennen** gegen einen anderen Spieler an, um als Erster ganz oben an der Kletterwand zu sein.

Die Spieler greifen nach den nächsten „Griffen“ um weiter zu klettern. Dieser simple Ablauf wird **ständig unterbrochen** und verändert aus einer Mischung von Spielerinteraktion, Events, Mini-Spielen und Power-Ups.

GEPLANNTES SPIELERLEBNIS

Obwohl XClimb im Kern ein Kompetitives Spiel ist, soll beim spielen **nicht** das Gewinnen/Verlieren **im Vordergrund** stehen sondern nur als Motivation und Zielsetzung für das Spielerlebnis dienen.

Dafür soll der „reguläre“ Spielablauf ständig durch Events, Mini-Spiele und Power-Ups verändert und unterbrochen werden. Ähnlich wie bei Mario-Kart oder Smash-Bros (mit Arcade-Einstellungen) soll das Erlebnis sehr chaotisch sein und dadurch einen zu ernsthaften Charakter verhindern. Gleichzeitig wird durch das **Chaos zu navigieren** der eigentliche Sinn des Spiels.

Bei jedem Durchlauf soll das sich das Erlebnis stark von den vorangegangenen Durchläufen unterscheiden um den Spielern keine Möglichkeit zu geben eine dominante Strategie zu entwickeln. Dafür kann man beispielsweise zufällig generierte Level, Events und Power-Ups einsetzen. Möglichkeiten, wie die Länge des Levels oder die Art der Events einzustellen können zusammen mit unterschiedlichen Modi ebenfalls zum **Abwechslungsreichen** Spielerlebnis beitragen.

Zusammengefasst ist das wichtigste, dass die Spieler immer auf Trab gehalten werden und sich ständig auf neue Situationen und Ereignisse einstellen müssen.

ZIELGRUPPE

XClimb soll durch seinen Casual-Charakter für ein sehr breites Publikum zugänglich sein.

Die Steuerung ist sehr Intuitiv, auch für nicht gaming-affine Spieler und lässt sich durch einfaches Spielen vermitteln.

Bewegungsspiele sind im Allgemeinen bei so gut wie allen Altersgruppen beliebt und bieten die Möglichkeit Gruppen die sonst kaum miteinander spielen ein gemeinsames Erlebnis zu bieten. Dafür ist auch sehr entscheidend, dass sich die Intensität des Spiels sehr leicht anpassen lässt und somit für jeden Spieler eine faszinierende Herausforderung bereithält.

SLOGAN

Der Slogan ist eine Anspielung an „Wer bremst verliert!“, was sehr passend ist, da es sich bei XClimb um ein Rennen handelt, aber eben nicht nach „vorne“ sondern nach „oben“.

ANSÄTZE ZUR UMSETZUNG

DIE KLETTER MECHANIK

Im Herzen des Spiels liegt natürlich die **Kletter-Mechanik**, die an eine Bewegung des Spielers gebunden ist und durch die Mov-X Sensoren erfasst wird. Dabei wird das Klettern im gesamten Spiel ständig vorkommen und sollte sich deswegen gut anfühlen und mit der Kletter-Animation zusammenpassen. Das Klettern **alleine sollte schon Spaß** machen, auch wenn es natürlich nicht nur dabei bleiben wird. Genauso wie jeder in Smash-Bros ohne Ziel trotzdem anfangen würde rum zu springen und schlagen, weil es sich gut anfühlt und Spaß macht, sollte das Klettern selber sich schon gut anfühlen.

Der Bewegungsablauf:

Die Bewegung lässt sich in **2 Teile** aufteilen: Im ersten Teil greift der Spieler nach dem Griff, im zweiten zieht er sich zu dem Griff.

1. Ausgangsposition ist immer, dass der Spieler beide Hände **vor seinem Körper** auf Brusthöhe hat. Wenn in der Nähe seines Kletterers nun ein Griff eingeblendet wird, streckt er seinen Arm in die entsprechende Richtung, bis das UI ihm zeigt, dass er seine Hand nun am Griff hat.
2. Wenn er die Hand am Griff hat muss er sie nur wieder zum Körper **zurückziehen** und der Kletterer wird sich zum nächsten Griff ziehen.

Da es den Spielfluss erheblich stören würde, wenn der Spieler es mehrmals nicht schafft weiter zu klettern, sollte das System den Spieler ein bisschen schummeln lassen und auch wenn der Griff nicht genau getroffen wurde ihn weiter lassen. So vermeiden wir Frustration und machen es uns ein bisschen einfacher die Sensor Bewegung in Spiel-Ereignisse umzusetzen. Wir sollten allerdings vermeiden, dass zwei Griffe zu nah beieinander liegen, dass macht es uns nur schwieriger festzustellen, welchen davon der Spieler tatsächlich greifen wollte. Es ist hier auch möglich dieses „Greif-Hilfe“ für schlechtere Spieler stärker ausfallen zu lassen, also auch unpräzisere Bewegungen zu akzeptieren. Mehr dazu siehe „Self-Balancing“.

Abschließend ist noch festzuhalten, dass die Kletter-Bewegung sich nur in 2-D abspielt, der Spieler streckt nämlich seinen Arm immer parallel zum Körper aus. Mit dieser Konvention zu brechen kann ein interessanter Ansatz für Events oder Power-Ups sein.

SELF-BALANCING

Mit „Self-Balancing“ ist ein Spiel gemeint, dass auf natürliche Art und Weise dafür sorgt, dass Spieler die hinter ihren Kontrahenten zurückfallen wieder aufholen können. Das bekannteste Beispiel hierfür sind wohl die ?-Boxen in Mario-Kart, die abgeschlagenen Spielern deutlich bessere Power-Ups geben als führenden.

Das ist für Spiele, die von Aktion und Wettkampf leben besonders wichtig, da sie dem Spiel eine gewisse Unberechenbarkeit geben und somit nie ein Punkt an erreicht wird, an dem das Ergebnis schon klar ist und es nur noch ausgespielt werden muss. Außerdem erlaubt es Spielern, mit unterschiedlichem Können zusammen zu spielen.

Natürlich leiden unter diesem Prinzip immer die Fairness und das Gefühl der Spieler direkten Einfluss auf das Ergebnis der Spiels zu haben. Solange diese Nachteile sich in einem gewissen Rahmen halten kann unser Spiel aber viel durch dieses Prinzip dazugewinnen.

KONKRETES SELF-BALANCING

Für XClimb ist die naheliegende Anwendung für „Self-Balancing“ ein Power-Up System, das den Spieler, der aufholen muss einen Vorteil verschafft. Das hat den Vorteil, dass die Beeinflussung für die Spieler nicht direkt ersichtlich ist (die Power-Ups werden ja schließlich zufällig vergeben) und sich sehr einfach umsetzen lässt.

Möglich wäre es auch, dass der führende Spieler öfter von negativen Events betroffen ist, das darf sich dann aber nicht zu unfair anfühlen.

Es ist auch denkbar, das Ganze noch subtiler zu gestalten und die Animationen der Spieler zu beschleunigen oder verlangsamen, je nach Abstand.

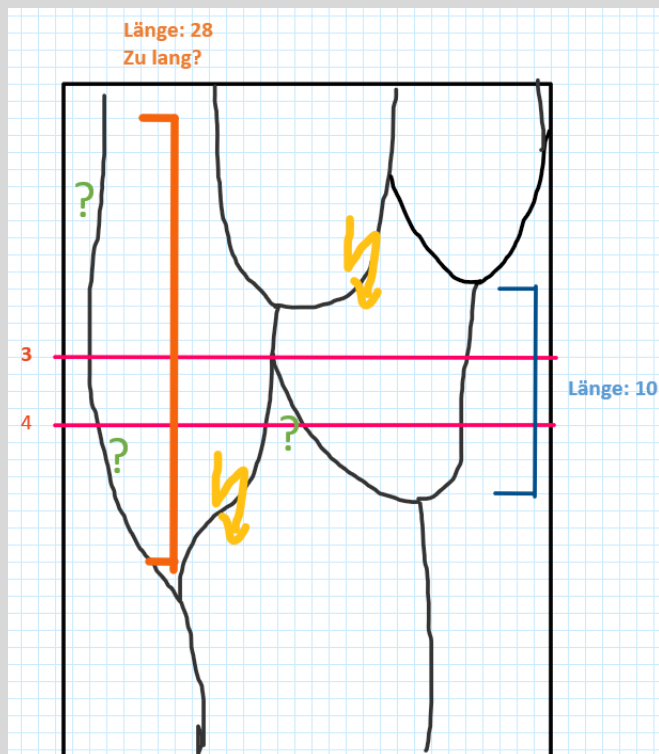
AUFBAU DER KLETTERWAND

Die Kletterwand soll den Spielern die Möglichkeit geben, sich zwischen verschiedenen Routen zu **entscheiden**, die absehbare Vor- und Nachteile haben. Beispielsweise kann man sich entscheiden sich in die Nähe des Mitspielers zu bewegen oder weiter von ihm weg. Außerdem können auch bestimmte Risiken und Belohnungen, wie Power-Ups oder schwierigere Griffe auf die verschiedenen Routen verteilt sein.

Der Aufbau der Wand soll **zufällig generiert** sein. Das beinhaltet den Aufbau der Routen, die Art der Griffe, Verteilung der Power-Ups und andere Besonderheiten. Damit müssen Spieler sich auf ständig neue Begebenheiten einstellen.

Bei der Verteilung der Routen muss auf einige Dinge geachtet werden:

1. An jeder Stelle in der Kletterwand dürfen nicht zu viele Routen gleichzeitig sein um das Level übersichtlich zu halten und den Spielern klare Entscheidungsmöglichkeiten zu bieten. (Siehe rote Striche)
2. Routen sollten sich nicht zu oft verzweigen, dh. nach eine Abzweigung sollte nicht zu schnell wieder eine neue Abzweigung kommen damit Spieler genug Zeit haben sich zu entscheiden. Es sollte also eine Mindestzahl an Griffen zwischen zwei Abzweigungen geben. (Siehe blaue Markierung)
3. Keine Route darf eine Sackgasse sein oder dem Spieler einen nicht mehr aufholbaren Nachteil geben. Dann würde das Spiel sofort frustrierend und langweilig werden.
4. Keine Route sollte zu lange ohne Abzweigung verlaufen, da ein Spieler sonst zu lange keine wichtigen Entscheidungen zu treffen hat. (Siehe Orange Markierung)



Die Werte in der Zeichnung haben natürlich keine Bedeutung und müssen durch ausprobieren ermittelt werden.

SPIELER INTERAKTIONEN

Für ein Spiel, das auf dem Wettkampf basiert sind die Möglichkeiten mit seinem **Gegner** zu **interagieren** essentiell für das Spielerlebnis. Außerdem bieten sie für uns eine weitere Möglichkeit das Geschehen chaotischer werden zu lassen.

Die erste Art der Interaktion, die festgelegt werden muss, ist der Fall dass beide Spieler die gleiche Route genommen haben und einer nun versucht auf einen Griff zu klettern, der schon vom anderen Spieler besetzt ist. Mein Vorschlag diesen Konflikt zu lösen wäre, dass der von unten kommende Spieler seinen Gegner ein kleines Stück runterwirft, wenn er seinen Kontrahenten einholt. Dadurch wird dieser für sein Aufholen **belohnt** und eine **neue Spielsituation** entsteht. Außerdem entsteht so ein intensiver Moment, indem der führende gejagt wird und Versuchen muss wieder Abstand zu gewinnen. Diese hektische Situation ist genau was wir bei unseren Spieler hervorrufen wollen und kann zu einem kleinen Höhepunkt des Spielgeschehens werden.

Eine weitere Art der Interaktion ist möglich, wenn die Spieler sich auf unterschiedlichen Routen befinden aber ihre Kletterer sich sehr nahe beieinander an der Wand befinden. Dann können die Spieler versuchen den anderen von der Wand zu schlagen, sodass dieser ein kleines Stück herunterfällt. Solche direkte Interaktionen verstärken nochmal das Gefühl sich in einem **Kampf/Rennen** zu befinden und wird den Spielern auch im Gedächtnis bleiben, genauso wie sich jeder an den blauen Panzer erinnern kann, mit dem ein Freund einem den Sieg genommen hat.

Ansonsten wird es natürlich Power-Ups geben, die in irgendeiner Art und Weise den Gegner beeinflussen, dazu aber mehr unter „Power-Ups“.

GUI

GUI FÜR BEWEGUNGEN

Es werden im Laufe des Spiels immer wieder **neue Mechaniken** und damit **Bewegungen** auftauchen. Beispiele wären ein Power-Up, mit dem man Springen kann oder ein Event, dass erfordert mit beiden Händen zuzugreifen. Diese neue Bewegungen müssen den Spieler intuitiv und schnell vermittelt werden um den Spielfluss nicht zu stören und den Einstieg für neue Spieler einfach zu gestalten.

Dafür braucht es eine Art GUI das mit Pfeilen oder anderen Mitteln die Art der Bewegungen vermittelt.

GUI FÜR DAS LEVEL

An der Wand selber wird es auch viele Informationen geben, die den Spielern ständig kommuniziert werden müssen. Am wichtigsten ist dabei wohl die Art der Griffe. Das lässt sich wohl am besten durch eine farbliche Markierung umsetzen.

Essentiell ist auch dem Spieler zu zeigen, in welche Richtung er Greifen muss um weiter zu klettern. Dafür sollten die nächsten Griffe sich stark vor allem anderen hervorheben und der Spieler muss sofort sehen, wo sich der Griff relativ zu seinem Kletterer befindet um dort hinzugreifen. Bedeutet wenn der Griff rechts über dem Kletterer aufleuchtet greift der Spieler nach rechts-oben.

Zusätzlich sollen um den Spieler einen **Überblick** über den Aufbau des Levels zu verschaffen die verschiedenen Routen mit ihren wichtigsten Stellen erkennbar sein. So können die Spieler einfacher und schneller Entscheidungen treffen.

HUD

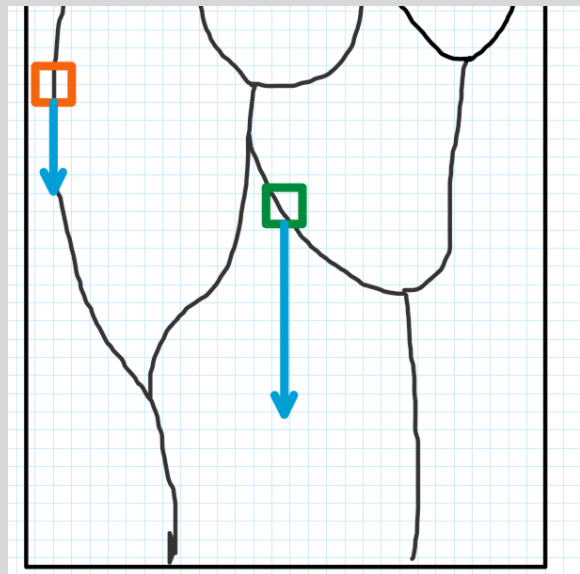
Da die meisten wichtigen Informationen schon im Level selber dargestellt werden, muss das HUD nicht allzu kompliziert sein. Da der Stand des Spiels schon durch die Position der Kletterer klar ersichtlich ist, braucht es keinen Punktestand (außer man spielt einen BO3/BO5 Modus). Sehr hilfreich für das richtige Spielgefühl ist es, einen Timer einzublenden, um zu verdeutlichen, dass es um Schnelligkeit geht. Bei einem Zeitbegrenzten Spielmodus könnte dieser sogar eine tatsächliche Bedeutung bekommen. Des Weiteren kann das HUD einen Slot für aufgesammelte Power-Ups haben.

HERUNTERFALLEN

//UPDATE

Das Herunterfallen, das in Bezug auf das Rennen bedeutet zurückgesetzt zu werden, ist gar keine so triviale Mechanik wie es auf den ersten Blick scheint. Der erste Gedanke ist, dass der Kletterer einfach runterfällt, bis er beim Fallen auf eine Route kommt und sich dann an einem Griff festhält.

Wenn man sich aber die Abbildung rechts anschaut, sieht man, dass in manchen Fällen, wie bei dem orangenen Kletterer, die Route senkrecht verläuft und dieser somit sofort wieder auf einer Route ist und sich an einem Griff festhalten kann. Bei dem grünen Kletterer hingegen ist gar keine Route unter ihm und er fällt „Getting Over It“-Style nach ganz unten. Wir achten zwar nicht immer ganz genau auf Fairness, aber das ist natürlich zu viel des Guten. Um solche Situationen zu verhindern, braucht es zwei Dinge:



1. Ein Minimum an Distanz, die der Kletterer mindestens runterfallen muss, bevor er sich fangen kann.
2. Ein Maximum an Distanz, die der Kletterer höchstens runterfallen kann. Wenn der Kletterer dann fernab von allen Routen ist, müssen wir ihn wahrscheinlich einfach zum nächsten Griff springen lassen. Das ist zwar keine schöne Lösung, aber die einzige, die mir einfällt.

Es bleibt auch noch festzulegen, ob der Spieler etwas tun muss, um sich festzuhalten, oder ob dies automatisch passiert. Ich würde vorschlagen, dass wenn der Kletterer genau über einen Griff fällt, hält er sich automatisch fest. Wenn er knapp vorbei fällt, kriegt der Spieler die Möglichkeit danach zu greifen. So werden schlechtere Spieler nicht zu unfair behandelt und bessere Spieler belohnt.

Update 19.10: Wie Niko vorgeschlagen hat, ist wohl die einfachste und fairste Methode, den Kletterer einfach aus dem Bild fallen zu lassen und danach eine feste Anzahl an Griffen zurückzusetzen. Es gab auch die Idee, den Kletterer nicht einfach wieder an die Wand zu setzen, sondern ihn hochspringen zu lassen wie von einem Trampolin.

GRIFFE

Die Griffe sind als Teil der Grund-Mechanik zentraler Bestandteil des Erlebnisses und sind die beste Möglichkeit **Abwechslung** in das elementare Spielprinzip zu bringen.

Griffe können Eigenschaften haben, die sie gefährlicher machen oder eine andere Art der Bewegung erfordern.

Beispiele für „**Gefährliche**“ Griffe:

1. Ein bröckeliger Griff, von dem der Spieler runterfällt wenn er nicht schnell genug weiterklettert. Dieser eignet sich auch gut dazu dem schlechteren Spieler einen versteckten Vorteil zu verschaffen, indem man ihm mehr Zeit gibt.
2. Eine „Attrappe“ die nur da ist um den Spieler zu täuschen und denjenigen bestraft, der danach greift.
3. Ein wackeliger Griff, der erfordert dass man sich sofort nochmal festhält wenn man zu ihm gewechselt hat.

Beispiele für „**Andere**“ Griffe

1. Ein Griff, der mit beiden Händen gepackt werden muss.
2. Ein heißer Griff, den man erst abkühlen lassen muss um ihn greifen zu können. Auch geeignet für Self-Balancing.
3. Zwei Griffe, die man gleichzeitig greifen muss um weiterzukommen.

Die unterschiedlichen Griffe müssen durch das UI für die Spieler klar unterscheidbar sein. Es sollte am besten auch schon absehbar sein, welche Griffe als nächstes kommen werden, um den Spielern die Möglichkeit zu geben ihren Weg zu planen. Aber nicht alle Besonderheiten müssen klar zu sehen, manche können auch überraschend auftreten, wenn der Spieler sie erreicht.

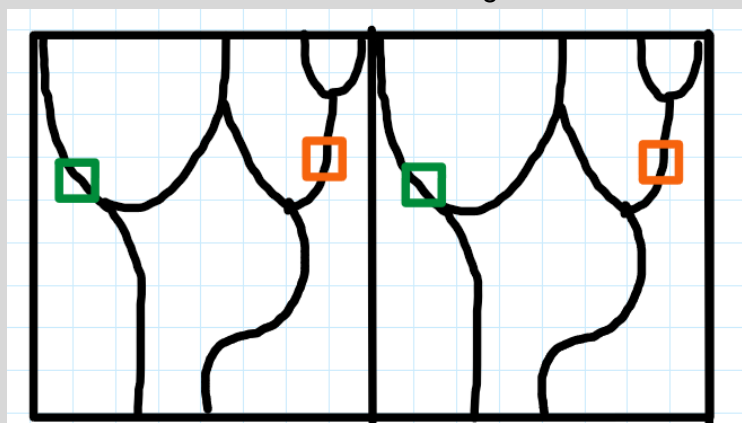
DER SPLITSCREEN

//NEW

Da XClimb von zwei Spielern im lokalen Co-op gespielt werden soll, muss der Bildschirm als Split-Screen aufgeteilt werden. Dabei stellen sich zwei entscheidende Frage:

- Wird das Bild **vertikal oder horizontal** geteilt?
- Wird das Bild **zusammengeführt**, wenn die Spieler nahe genug sind?

Wenn das Bild dauerhaft im Split-Screen bleibt dann ist das wohl erstens einfacher umzusetzen, weil man nicht überprüfen muss ob beide Kletterer in einen Frame passen. Zweitens bietet sich es dann an das ganze vertikal zu trennen, da so jeder Spieler am meisten von dem sieht was als nächstes kommt, nämlich das was über dem Kletterer ist.

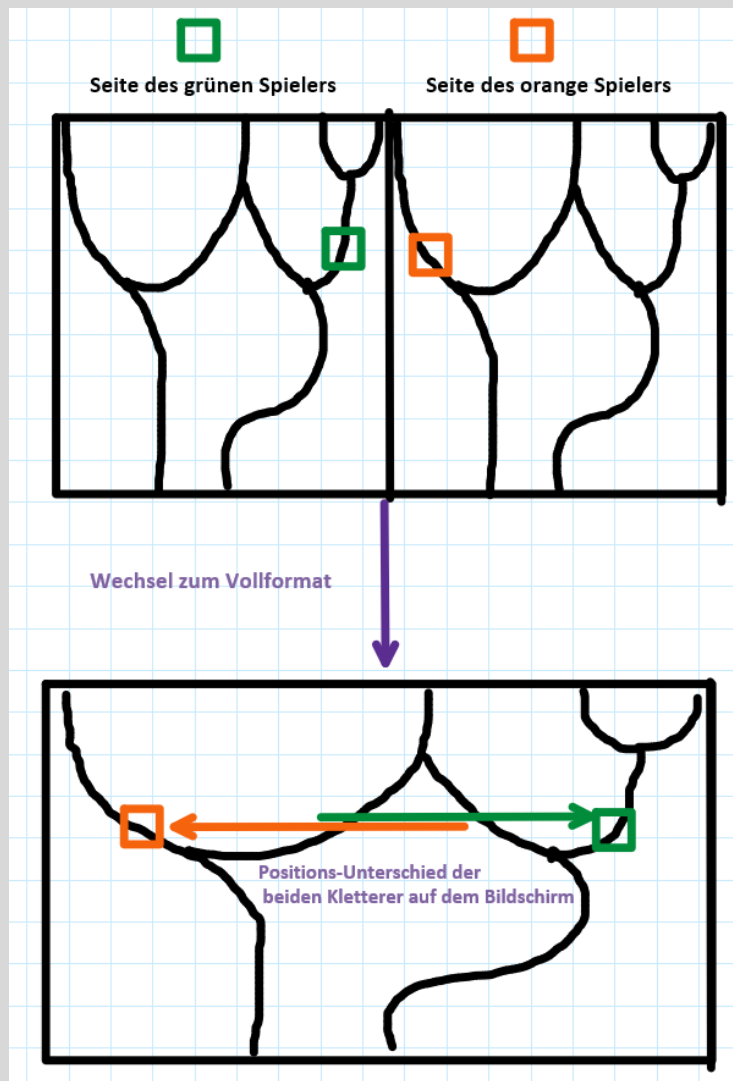


Der vertikale-permanente Split-Screen ist eine einfache und natürliche Variante.

Es gibt aber auch noch eine andere Variante, die vielleicht sogar besser ist als diese. Wenn wir den Split-Screen dynamisch gestalten, also dass wenn die Kletterer nah genug zusammen sind um sie auf einen Frame zu bringen, dann können wir den Split Screen auflösen und den Spielern das Level auf dem ganzen Bildschirm zeigen. Das hätte auch den schönen Effekt, dass den Spielern klar wird, dass der Abstand kleiner geworden ist und würde die Spannung in diesem Moment erhöhen. In diesem Fall muss man dann aber den Bildschirm horizontal teilen, weil es sonst passieren kann, dass ein Spieler der gerade noch die rechte Hälfte des Bildes hatte jetzt auf einmal ganz links ist, was den Spielfluss stark beeinträchtigen würde. Wenn man das Bild allerdings horizontal teilt, kann man immer dem Spieler, der weiter oben ist, die obere Hälfte geben und wenn das Bild wieder zusammengeführt wird, befindet er sich immer noch in der oberen Hälfte des Bildes. (Siehe Abbildung).

Der größte Positionsunterschied, der möglich ist, ist bei einem horizontalen Split ist viel geringer.

Trotz allem bleibt wohl die Frage ob es eine gute Idee ist unser vertikales Level in einem Querformat darzustellen.



MODI

//NEW

XClimb hat viele Parameter, die sich verändern lassen um ein neues Spielerlebnis zu schaffen. Die so entstehenden Spiel-Modi können Abwechslung in das Geschehen bringen und ständig neue Anforderungen an die Spieler stellen. Folgende Parameter sollten sich einfach ändern lassen:

1. Die **Länge** des Levels: Da die Level generiert werden sollen, sollte einfach ein kürzeres oder längeres Level zu generieren keine großen Umstände mehr machen.
2. Art und Häufigkeit von **Events**.
3. Art und Vorkommen der **Griffe**.
4. Art und Häufigkeit von Power-Ups

Und vieles mehr.

Indem man diese Parameter auf verschiedene Arten manipuliert und kombiniert entstehen neue Modi. Beispiele dafür sind:

1. Endless-Mode: In diesem Modus hat das Level kein Ende und alle 10 Griffe bekommen die Spieler 5 Sekunden dazu. Die Spieler spielen gemeinsam gegen die Zeit.
2. Sprint: Extrem kurzes Level, ohne Events und ähnliches, dass nach ungefähr 20 Sekunden vorbei ist. Das ist sehr nahe am echten Speed-Climbing.
3. Apokalypse: In diesem Modus spielt die Welt verrückt es treten ständig Events auf.

Natürlich ist es auch sinnvoll im Default-Modus den Spieler einfach selber die Möglichkeit zu geben die Parameter für ihre eigenen Wünsche anzupassen (wie bei Smash-Bros). Ansonsten sind der Kreativität beim Erstellen von weiteren Modi keine Grenzen gesetzt.

EVENTS

//NEW

Wie bereits erwähnt, sollen Events einen großen Teil des Spielgeschehens darstellen. Mit Events sind **zufällig** auftretende Ereignisse gemeint, die **beide** Spieler betreffen und meistens eine zusätzliche Herausforderung darstellen oder ein elementares Spielprinzip ändern. Ziel ist es mit den Events den Spielfluss spannend und unvorhersehbar zu gestalten. Außerdem lassen die Events die Spieler immer wieder neue Inhalte entdecken und geben damit einen Anreiz eine weitere Runde zu spielen.

Bei Events sind zwei Dinge im Fokus:

Wann und **wie oft** treten **welche** Events auf?

ARTEN VON EVENTS

Events sollen beide Spieler betreffen und eine interessante Herausforderung darstellen. Dafür sollen sie entweder ein grundlegendes Spielprinzip ändern oder für ihre Dauer eine Mechanik hinzufügen. Events sollen nach einer bestimmten Zeit einfach aufhören. Ein gutes Zeitfenster wäre wohl etwas um die 5 bis 10 Sekunden, das wird sich durch ausprobieren wohl am besten eingrenzen lassen. Nach dem Event soll das Spiel wie davor weitergehen, abgesehen von Veränderungen der Positionen der beiden Spieler.

Vorläufige Beispiele für **neue Mechaniken**:

1. Steine (oder andere Gegenstände) die von oben herunterfallen und Spieler, die von ihnen getroffen werden herunterfallen lassen.
2. Disko-Modus: Spieler dürfen nur im Rhythmus zur Musik weiterklettern, wenn sie zur falschen Zeit klettern fallen sie herunter. So hätten wir auch Rhythm-Games in unser Spiel aufgenommen.
3. 4 Drohnen fliegen auf den Spieler zu, umkreisen ihn und halten ihn mit einer Art Lähmungsstrahl fest. Um weiter klettern zu können müssen die Drohnen zerschlagen werden.

Vorläufige Beispiele für **zusätzliche Herausforderungen**:

1. Sicht-Behinderung: Ob einfach irgendein Objekt, wie eine Wolke, vor ein Teil des Frames zieht, oder einfach das Bild so stark wackelt, dass es schwer wird noch alles genau zu sehen, beides sind einfache Möglichkeiten den Spieler eine besondere Herausforderung stellen. Visuelle Informationen einzuschränken ist besonders interessant, weil man ohne das eigentliche Spiel zu verändern ein ganz neues Erlebnis schafft.
2. Erdbeben: Mit kurzer Vorwarnung rütteln Erdbeben die Wand und wer gerade versucht weiter zu klettern fällt runter.
3. Ein Schwarm an Vögeln (oder Drohnen) greift die Spieler an. Um sich zu verteidigen müssen sie die Vögel aus der Luft schlagen bevor sie den Kletterer erreichen. Wenn

ein Vogel den Kletterer erreicht wird dieser kurzzeitig betäubt. In diesem Sinne sollte man sich immer überlegen wenn ein Spieler bestraft wird, wie hart diese Strafe ausfallen soll. Man könnte den Spieler auch runterfallen lassen, das wäre wahrscheinlich aber zu hart.

Für den Bereich „Events“ lohnt es sich noch ein richtige Brainstorming Session einzulegen, da es noch sehr viel Potential für Event Ideen gibt und es am besten ist, wenn Events sich nicht zu ähnlich sind und die Spieler überraschen. Außerdem gilt bei Events tatsächlich mehr ist besser um das Spiel möglichst abwechslungsreich zu gestalten. Meine Beispiele sind nur Rudimentäre Vorschläge und es macht definitiv Sinn, wenn sich jeder selber noch ein paar Events überlegt um eine möglichst große Auswahl zu haben.

TIMING VON EVENTS

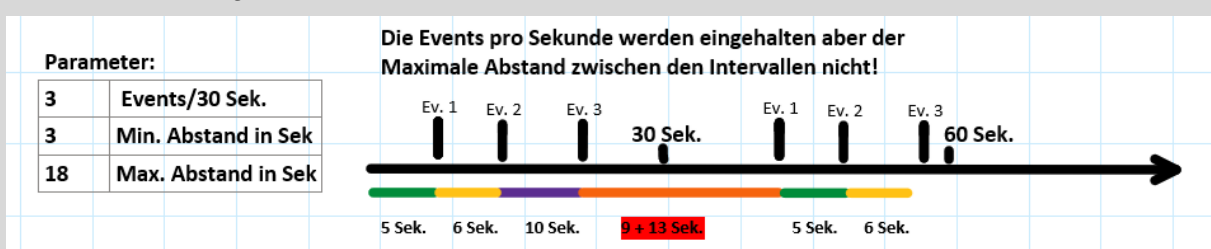
Event-Timer: Um in regelmäßigen Abständen verschiedene Events auszulösen werden wir eine Event-Timer Klasse brauchen, die genau dafür zuständig ist. Diese Klasse wird besonders praktisch, weil Events eine der besten Möglichkeiten sind, die Schwierigkeit des Spiels dynamisch anzupassen. Soll heißen je nach gewähltem Modi, Schwierigkeitsgrad oder Custom-Einstellung soll die Häufigkeit und Art des Events anders sein.

Bei der ersten Funktion des Timers, dem **scheduling von Events**, müssen wir erstmal überlegen ob die Spieler wissen sollen wann das nächste Event ausbricht. Ich würde vorschlagen, dass die Spieler von den Events überrascht werden sollen, aber um nicht zu unfair zu sein, sollten wir kurz vor jedem Event eine Vorwarnung geben. Beispielsweise kann das Bild schon mal anfangen zu wackeln, kurz bevor Steine von oben runterfallen. Die Warnung sollte so **1,5-3 Sekunden** vor dem Event erfolgen, sodass die Spieler sich gerade noch wappnen können.

Zum scheduling gehören auch **Abstände** und **Häufigkeit**. Die Häufigkeit lässt sich gut durch einen einzigen Parameter festlegen. Ich würde vorschlagen die Häufigkeit in **Events pro 30 Sekunden** messen. Den Abstand zwischen Events kann man natürlich einfach als Konstante, die sich aus der Häufigkeit ergibt, ausdrücken, sodass Events immer in den gleichen Abständen auftreten.

Da unser Spiel aber auf Chaos und Unberechenbarkeit aufbaut, bietet es sich an, die Events in **zufälligen Intervallen** zu planen. Erstmal würde ich das ganze wirklich zufällig machen, später ist es auch möglich mit verschiedenen Verteilungsmustern zu experimentieren. Die einzigen Parameter, die ich dabei noch einstellen würde, sind ein Minimaler und ein Maximaler Abstand.

Wichtig für die Event-Timer Klasse ist natürlich noch, dass sie weit genug in die Zukunft plant um sicherzustellen, dass die eingestellten Events pro 30 Sekunden auch eingehalten werden können. Zu Problemen könnte es dabei aber kommen, wenn der Timer im 30 Sekunden plant und erst danach die nächsten 30. Die Anzahl an Events passt dann natürlich, aber für die Einhaltung des Maximalen Abstands zwischen den 30 Sek. Intervallen kann nicht garantiert werden. (Siehe Abbildung.) Deswegen sollte der Timer zuerst die ersten 30 Sek. planen und dann nach jedem Event wieder ein neues Event anhängen, natürlich unter Beachtung aller Parameter. Natürlich werden die Events in der Realität für eine gewisse Zeit aktiv sein. Die Abstände sollten deswegen von Ende des aktuellen bis zum Anfang des nächsten Events gemessen werden.



Zum Schluss sollte der Event-Timer noch eine Möglichkeit haben von außen gesondert Events, die durch andere Geschehnisse ausgelöst werden einzufügen.

POWER-UPS

//NEW

Neben Events sind Power-Ups die offensichtlichste Methode um Chaos und Abwechslung in das Spiel zu bringen. Genau wie bei den Events gibt es wieder zwei Haupt-Aspekte die man festlegen muss: **Wie und welche** Power-Ups erhält man und **wie oft** erhält man Power-Ups.

POWER-UPS AUFSAMMELN

Die wohl häufigste Art, in Videospielen die Power-Ups an Spieler zu verteilen ist die ?-Box. Die ?-Box ist eine sehr abstrakte Repräsentation für die zufällige Verteilung von Power-Ups ohne Bezug zu Realität. Power-Ups **zufällig zu verteilen** wäre auch für unser Spiel praktisch, da wir so dem schlechteren Spieler helfen können („Self-Balancing“) ohne das es zu offensichtlich ist.

Ich würde aber eine andere Metapher als die ?-Box verwenden. Da unsere Kletterer die Form eines Roboters annehmen sollen bietet es sich an als Form für die Boxen etwas Mechanisches/Technisches zu wählen. Eine weitere Überlegung ist, **verschiedene** Arten von **Boxen** einzuführen, beispielsweise ein Kanister für „Speed-Ups“ oder Munition für „Waffen“. Das hätte auch den schönen Effekt, dass der Spieler wenn er diese Boxen auf einer bestimmten Route sieht sich überlegen kann ob er sich lieber für eine Waffe oder ein Speed-Up entscheiden will. Das gibt den Entscheidungen, die die Spieler treffen eine einfach zu verstehende Bedeutung ohne dass eine Variante offensichtlich besser ist als die andere.

Bleibt noch zu klären wie die Power-Ups in das Level eingebaut werden und wie sie aufgesammelt werden können. Da wir ja ein eher futuristisches Setting gewählt haben, ist eine Möglichkeit die Power-Ups an eine **Drohne zu hängen** und diese dann automatisch dem Spieler zu geben, wenn er den entsprechenden Griff erreicht hat. Die Drohne kann dann einfach wegfliegen.



Eine letzte Sache gibt es aber noch: Wie **aktiviert** man die Power-Ups eigentlich? Dafür brauchen wir eine Bewegung, die nicht mit anderen Eingaben verwechselt werden kann, sich am besten mächtig anfühlt und schnell ausgeführt werden kann. Mein Vorschlag für diesen „**Power-Move**“ ist, von der Ausgangsposition mit beiden Händen auf Brusthöhe vor dem Körper, beide gleichzeitig und schnell zur Seite auszustrecken. Das fühlt sich irgendwie mächtig und sehr plötzlich an, da bekommt das „**Power-Up**“ eine ganz neue Bedeutung.

VERSCHIEDENE POWER-UPS

Jetzt ist geklärt, wie Power-Ups prinzipiell funktionieren aber noch überhaupt nicht, **welche** es überhaupt gibt.

Wenn man sich Mario-Kart als Vorbild nimmt merkt man, dass Power-Ups in einem Rennspiel immer entweder einem **Gegner behindern** oder dem Spieler selber einen **Vorteil verschaffen**. Power-Ups die einem Spieler einen Vorteil verschaffen lassen sich weiter unterscheiden in Effekte, die einfach nur einen Parameter verändern (schnellere

Geschwindigkeit in Mario-Kart) oder Effekte, die Spielregeln verändern und abwandeln (über Grass fahren ohne langsamer zu werden). Am besten ist es natürlich wenn sie beides gleichzeitig erfüllen (Roter Pilz in M-K). Wie im letzten Abschnitt beschrieben wollen wir alle Power-Ups in eine Kategorie packen. Ob es nur bei „Kanister“ und „Munition“ bleibt kann ich noch nicht sagen, es gilt auf jeden Fall wieder das gleiche wie bei Events: mehr Auswahl ist besser, allerdings sollte es für die Spieler nicht zu unübersichtlich werden, sie müssen die Power-Ups ja bewusst einsetzen.

Beispiele für „**Munition**“ Power-Ups:

1. Laser-Kanone: Bei Aktivierung aktiviert sich die Laser-Kanone im Arm des Roboters. Der Spieler kann mit seinem Arm zielen und versuchen seinen Gegner zu treffen, damit dieser runterfällt.
2. Rakete: Eine Rakete die abgefeuert werden kann und den Gegner selbstständig trifft (Red-Shell in MK).
3. Ein Stein (oder ähnliches) den der Spieler fallen lassen kann um seinen Gegner damit zu treffen.

Beispiele für „**Kanister**“ Power-Ups:

1. Greifarme: Der Roboter aktiviert seine Greifarme und kann immer schon einen Griff weiter greifen. So wird der Spieler effektiv doppelt so schnell.
2. Jetpack: Das Jetpack gibt dem Spieler die Möglichkeit einfach eine große Distanz aufzuholen. (ähnlich Bullet-Bill in MK)
3. Ein Power-Up, das für den Spieler eine Extra Route anlegt, die steil nach oben geht und sehr einfach zu klettern ist. Also immer ein Griff Links ein Griff Rechts, sodass der Spieler schnell Klettern kann und in einen Flow-Zustand kommt.

Hier gilt auch wieder: Das sind keine Definitive Vorschläge und alles was euch sonst noch einfällt ist gut und wichtig.

POWER-UP DISTRIBUTION

So jetzt bleibt nur noch zu klären, welches Power-Up der Spieler genau kriegt, wenn er eine „Box“ einsammelt. Für den Spieler soll es so wirken, als ob die Power-Ups **zufällig** verteilt werden. Für uns ist die „Box“ aber natürlich eine goldenen Gelegenheit, „Self-Balancing“ anzuwenden. Soll heißen, wir geben dem führenden Spieler **weniger starke** und seinem Gegner deutlich **stärkere** Items. Umso größer die Distanz zwischen den Spielern, umso stärker sollten die Power-Ups für den abgeschlagenen Spieler werden. Dafür würde ich die vertikale Distanz zwischen den Spielern messen und ihnen dann in eine der unten aufgeführten **Kategorien** zuweisen. Jede Kategorie hat einen Pool an Power-Ups mit **Spawn-Wahrscheinlichkeiten** aus dem ein zufälliges ausgewählt wird, wenn der Spieler eine Box einsammelt. Dabei kann ein Item auch in mehreren Pools vorkommen mit geänderter Wahrscheinlichkeit, aber immer nur in dem Pool der einer benachbarten Kategorie. Soll heißen, es ist in Ordnung, wenn die Rakete im Pool für den weit Abgeschlagenen Spieler und dem für den leichten Rückstand vorkommt. Der führende Spieler darf sie aber nicht kriegen.

Führend: Abst. > 10	Der führende Spieler sollte keine Power-Ups kriegen, die ihm erlauben den Abstand zu vergrößern. Sie sollten sich gegen den Gegner richten, dabei aber nicht sehr effektiv sein. (Banane in MK)
Gleichauf: Abst. <= 10	Bei Gleichstand sind die Power-Ups ausgeglichen und sind stark darauf ausgelegt, dass die Spieler sich bekämpfen können, da sie schon nah zusammen sind. (Green-Shell)

Leichter Rückstand: 10 < Abst. <50	Bei geringem Abstand erlauben die Power-Ups dem Spieler deutlich effektiver, gegen seinen Gegner vorzugehen oder geben ihm einen leichten Geschwindigkeitsvorteil.(Red-Shell)
Weit Abgeschlagen: Abst.<50	Liegt der Spieler weit zurück, geben Power-Ups ihm die Möglichkeit einfach und effektiv die Distanz zu verringern und so dem Spiel wieder Spannung zu geben. (Bullet Bill)

ANDERE LEVEL-ELEMENTE

//TO-DO

Möglichkeit für in das level integrierte Features, wie Teleporter oder Jump-Boosts.

VERSCHIEDENE KLETTERER

//TO-DO

Möglichkeit verschiedene Charaktere auswählen zu können mit unterschiedlichen Fähigkeiten (siehe Streak) und Aussehen.

KLETTER-ANIMATION

//TO-DO

Überlegungen für die Umsetzung der Kletterer Animation.

STAGES

//TO-DO

Ideen für verschiedene Stages und welche Ressourcen dafür angepasst werden müssten.

LOBBYS

//TO-DO

Layout und Funktionen für eine Lobby, die erlaubt Einstellungen vorzunehmen Stages zu ändern und Charaktere auszuwählen. (ähnlich Super-Smash)

STREAK

//TO-DO

Streak-o-Meter, dass Spielern, die schnell und ohne Unterbrechung klettern belohnt und ihnen ein positives Feedback gibt. Belohnung kann die Charakter spezifische Fähigkeit sein.

SETTING

//TO-DO

Nicht essentiell. Erklärung für Kletter-Wettkampf, Charaktere und Stages. Soll weder realistisch noch ausführlich sein. Vielleicht auch noch bessere Namen für Kletterer, Stages etc.