

Das offizielle Game Design Dokument zu

Project Wallker

- If there is a wall, there is a way -

Inhaltsangabe:

1. Die grobe Spielmechanik	S.2
2. Rahmenbedingungen des Projektes	S.3
3. Inspiration	S.4
4. Steuerungsmöglichkeiten (Controls)	S.5
5. Spielinhalte und Eigenschaften	S.7
• Statische und dynamische Objekte	
• Interaktionsmöglichkeiten des Spielers	
• Spielweise	
• Bewegungsmöglichkeiten (Movement)	
6. Allgemeines FAQ	S.11

Derzeitiger Stand: 12.05.2019

1. Die grobe Spielmechanik

Project Wallker ist ein 3D Puzzle-Platformer basierend auf der Idee durch das Laufen gegen bestimmte Wände Energie aufzubauen und diese wieder per Knopfdruck freizusetzen. Dabei muss der Spieler sich durch ein Level voller Hindernisse, wie zum Beispiel Abgründe, bewegen um ein Ziel zu erreichen. Das Aufbauen der Energie, von uns Vektorkraft genannt, läuft über folgende Schritte ab. Erstens ist nicht jede Wand dazu nutzbar, Vektorkraft aufzubauen. Nur spezielle Wände können benutzt werden, derzeit dadurch gekennzeichnet, dass sie komplett Weiß sind. Später wollen wir noch möglicherweise einen Partikeleffekt zum Aufladen hinzufügen. Gegen die Wand zu laufen reicht dabei aus um das Aufladen zu aktivieren. Während der Kollision mit der Wand wird eine Leiste aufgeladen, welche die Stärke der Kraft darstellt. Die Richtung der Vektorkraft wird von dem Winkel, in welcher man gegen die Wand läuft bestimmt. Bedeutet, wenn man Richtung Norden gegen eine Wand läuft, wird die Vektorkraft den Spieler nach Norden schleudern, läuft man Richtung Nordosten, so wird der Spieler beim Freisetzen der Kraft auch nach Nordosten geschleudert.

Derzeit wird die Richtung der Vektorkraft durch die letzte Kollision bestimmt, wenn man also die ganze Vektorkraft Richtung Norden aufbaut, aber dann zum Schluss nochmal Richtung Osten gegen eine der speziellen Wände läuft, wird die Vektorkraft den Spieler nach Osten mit voller Kraft schleudern. Die derzeitige Richtung der Vektorkraft wird dabei in Form eines Pfeiles angezeigt. Des Weiteren soll die Vektorkraft nicht nur in alle Himmelsrichtungen, sondern auch nach oben und unten nutzbar sein, was es uns ermöglicht unser Level vertikaler aufzubauen. Der Spieler soll dazu in der Lage sein, gleichzeitig zwei separate Vektorkräfte zu besitzen, welche er getrennt voneinander aufladen und nutzen kann. Dies bedeutet, dass der Spieler zum Beispiel eine Vektorkraft speichern kann, welche die Richtung Norden besitzt und die Stärke 10 hat, und eine zweite, welche Richtung Westen geht und die Stärke 15 hat. Der Spieler soll in der Lage sein, diese beiden Kräfte entweder nacheinander zu aktivieren, also sich zum Beispiel erst nach Norden zu schleudern und dann im Flug die zweite Vektorkraft aktivieren um nach Westen geschleudert zu werden. Eine zusätzliche Funktion soll auch noch sein, dass der Spieler die beiden Vektorkräfte zusammenlegen kann, also das er im Falle des Beispiels beide Vektoren per Knopfdruck kombiniert und dann einen Vektor in Richtung Nordwesten mit der Stärke 12,5 hat. Noch ist nicht beschlossen, ob der Spieler in der Lage sein soll, immer weiter Vektoren zu kombinieren, oder ob er nach dem Kombinieren gezwungen sein soll, den entstandenen Vektor zu benutzen.

Wir haben auch noch eine weitere Idee, welche wir aber nur einbauen werden, sollten wir noch die Zeit dazu haben. Diese Idee ist, dass man die Vektorkraft auch auf bestimmte andere Objekte übertragen lassen kann. Zur Erklärung: Nehmen wir an, wir haben im Level einen Ball, und dieser muss in eine spezielle Mulde transportiert werden, um den Weg zum Ziel zu öffnen. Jedoch ist zwischen dem Ball und der Mulde ein Abgrund. Um den Ball also auf die andere Seite zu befördern, lädt man in eine spezielle Richtung die Vektorkraft auf, und nutzt diese auf den Ball, welcher dann anstatt dem Spieler geschleudert wird. Andere Anwendungsmöglichkeiten wären zum Beispiel, dass ein schwerer Block den Weg versperrt, und man muss durch das gesamte Level gehen um einen Vektor aufzuladen, welcher am Ende die Stärke hat, den Block zu bewegen. Ein sich drehendes Rad wäre auch eine Möglichkeit, dieses müsste man dann mit einer bestimmten Kraft beeinflussen, damit es sich in eine bestimmte Position dreht.

2. Rahmenbedingungen des Projektes

Projekt Wallker soll natürlich für den PC entwickelt werden. Nach einigem Testen der Steuerung, dem Bewegungsverhalten des kontrollierten Avatars und anderen Faktoren haben wir uns jedoch dafür entschieden, dass die Steuerung für Controller-Nutzung optimiert werden soll. Zum einen gab die Steuerung mit dem Controller in Verbindung mit dem Bewegungsverhalten in unseren Tests ein angenehmeres und gefühltes kontrollierbares Spielgefühl. Ein weiterer Faktor, welcher uns zur Controller-Steuerung brachte, ist die Komplexität unserer Steuerung. Man wird zum einen Bewegen und Springen können; hat aber auch zwei auflade - und aktivierbare Vektoren, welche man auch noch kombinieren und aktivieren kann. In einem Level, in dem man diese Aktionen schnell hintereinander oder sogar gleichzeitig durchführen muss, sind komplexe Inputs erforderlich, welche durch das Layouts des Controllers einfacher zu betätigen sind.

Das Genre von Project Wallker ist ein 3D Puzzle-Platformer, in welchem wir sowohl langsame Puzzle Elemente wie auch schnelle Bewegungselemente verbinden wollen. Wir wollen sowohl Level bauen, welche sich langsam spielen und auf das Lösen komplexerer Rätsel konzentrieren, als auch schnell spielbare Level, wo es darum gehen wird, die Vektorkraft Mechanik so zu benutzen, um so schnell und flüssig wie möglich ans Ziel des Levels zu kommen. Schlussendlich könnte man auch noch eine Verbindung dieser beiden Level Arten anstreben, in welchen man unter Zeitdruck die Rätsel lösen muss.

Als graphisches Design wollen wir eher auf ein einfacheres und eintönigeres Design gehen. Unser Vorbild hierbei sind die Testkammern aus den Portal spielen. Da sich unser Spiel auf einen Puzzle Aspekt konzentriert sehen wir dieses Design als ausreichend und passend. Es hilft außerdem dabei, die speziellen Wände, an denen man sich aufladen kann, besser herausstechen zu lassen, zum Beispiel wieder genau wie in Portal, wo man die Portale nur an die weißen Wände setzen kann.

3. Inspiration

Die ursprüngliche Idee zu unserem Gameplay-Herzstück stammt von einem Glitch aus dem Spiel Super Mario 64. Durch diesen Glitch kann man Geschwindigkeit durch eine Wandkollision, unter bestimmten Voraussetzungen, aufbauen. Dies wird unter anderem in Speedrunning verwendet. Unser Gedanke war es, dass es interessant wäre, ein Spiel auf einen Glitch aufzubauen. Besonders in Situationen, bei denen der Spieler die Kontrolle über Charaktere oder Objekte hat, welche sich mit großer Geschwindigkeit bewegen oder auf die starke Kräfte wirken, können unerwartete aber sehr unterhaltsame Szenarien entstehen. Wir hoffen durch die Anwendung des Prinzips der Vektorkraft in einem Puzzle-Platformer, den Spieler dazu zu bringen, sich in diese etwas andere Weise der Fortbewegung und Rätsellösung einzudenken, während immer noch die Möglichkeit besteht, dass durch die Wirkung der Vektorkraft unerwartete oder unvorhergesehene Situationen entstehen.



Andere Spiele die uns auch inspiriert haben sind zum Beispiel „Portal“, welches ein ikonisches Rätselspiel ist, und auch mit einer besonderen Art der Fortbewegung beziehungsweise des Rätsellösens arbeitet. Dies ist einer der Gründe, warum wir uns auch graphisch an dem Spiel orientieren wollen.

Ein weiteres Spiel welches uns inspiriert hat ist „Human Fall Flat“, welches als Stärke seine Physiksimulation aufweist, durch die verrückte und unterhaltsame Situationen während des Spielens geschaffen werden. In unserem Spiel wollen wir zu einem gewissen Grad auch diesen chaotischen Spaß einbauen.



4. Steuerungsmöglichkeiten (Controls)

Das Mapping und die Controls im Allgemeinen sollen sich so natürlich wie möglich anfühlen. Es werden zwei mögliche Control Schemes getestet.



Kontrollschema A (Control Scheme A)

Input	Aktion	Beschreibung
A-button	Jumping	Standardmäßig für Jump-and-Runs oder andere Third-Person-Games, deshalb werden wir dies ebenfalls so verwenden.
B-button	Clearing	Mit B soll der aktuell aktive Vektor und dessen Kraft auf Null gesetzt werden.
X-button	Transfer	Der aktuell aktive Vektor und dessen aufgeladenen Kraft wird an das anvisierte Objekt übergeben.
Y-button	Combine	Beide Vektoren und deren Kraft werden verrechnet um eine neue Kraft und eine neue Richtung festzulegen. Ausgelöst wird dieser neue Vektor dann entweder mit „Use Dash 1“ oder „Use Dash 2“
LB	Select Dash 1	Vektor 1 wird ausgewählt und kann ab diesem Moment aufgeladen werden und mit Clearing zurückgesetzt werden.
LT	Use Dash 1	Die Richtung von Vektor 1 wird mit der Energie die Aufgeladen wurde zusammen gerechnet und der Charakter somit in diese Richtung katapultiert.
RB	Select Dash 2	Vektor 2 wird ausgewählt und kann ab diesem Moment aufgeladen werden und mit Clearing zurückgesetzt werden.
RT	Use Dash 2	Die Richtung von Vektor 2 wird mit der Energie die Aufgeladen wurde zusammen gerechnet und der Charakter somit in diese Richtung katapultiert.
D-Pad	TBA	TBA
Menu	Pause-Screen	Der Pause-Bildschirm soll angezeigt werden.

Kontrollschema B (Control Scheme B)

Input	Aktion	Beschreibung
A-button	Jumping	Standardmäßig für Jump-and-Runs oder andere Third-Person-Games, deshalb werden wir dies ebenfalls so verwenden.
B-button	Clearing	Mit B soll der aktuell aktive Vektor und dessen Kraft auf Null gesetzt werden.
X-button	Transfer	Der aktuell aktive Vektor und dessen aufgeladenen Kraft wird an das anvisierte Objekt übergeben.
Y-button	(Optional)	TBA
LB	Select Dash 1	Vektor 1 wird ausgewählt und kann ab diesem Moment aufgeladen werden und mit Clearing zurückgesetzt werden.
LT	Use Dash 1	Die Richtung von Vektor 1 wird mit der Energie die Aufgeladen wurde zusammen gerechnet und der Charakter somit in diese Richtung katapultiert.
RB	Select Dash 2	Vektor 2 wird ausgewählt und kann ab diesem Moment aufgeladen werden und mit Clearing zurückgesetzt werden.
RT	Use Dash 2	Die Richtung von Vektor 2 wird mit der Energie die Aufgeladen wurde zusammen gerechnet und der Charakter somit in diese Richtung katapultiert.
LT + RT	Use Combined Dash	Beide Vektoren und deren Kraft werden verrechnet um eine neue Kraft und eine neue Richtung festzulegen. Dieser wird direkt ausgelöst.
D-Pad	TBA	TBA
Menu	Pause-Screen	Der Pause-Bildschirm soll angezeigt werden.

Zusätzlich gibt es die Überlegung, ob man die Aufladung nur auslösen kann, während man die Select Dash Aktionen gedrückt hält. Wir werden nach dem Feedback aus dem Playtest dann das bestmögliche Control Scheme bauen.

5. Spielinhalte und Eigenschaften

Statische Objekte:

Zu statischen Elementen zählen wir Objekte, die durch die Interaktion mit der Spielfigur sich weder verändern noch verbrauchen. Durch die Implementierung unseres Projektes auf dreidimensionaler Ebene können wir als statische Objekte den Levelboden als auch die Levelwände aufzählen. Aufgrund der Tatsache, dass das Projekt eine möglichst einfach gehaltene Projektkomplexität anstrebt, wird sich an der Anzahl der statischen Objekte nicht ändern. Nichtsdestotrotz ist es notwendig näher auf die Wände einzugehen, denn die Eigenschaft der Wände ist von bedeutsamer Relevanz für die Implementierung unserer Spielmechanik.

Wände:

Wir haben zwei verschiedene Arten von statischen Wänden: Die reguläre Wand und die Vektor-Wand.

Wir sprechen zuerst über die **reguläre Wand** denn diese ist leicht erklärt. An dieser Wand kann man die Vektorstärke nicht aufladen. Solch Wände werden typischerweise an den Leveleingrenzungen entlang platziert. Durch die Puzzle-Eigenschaft der Spielweise müssen wir die Liste der Spielmöglichkeiten insoweit einschränken sodass die Rätsel, welche durch das Leveldesign entstehen, überschaubar bleiben. Ein quadratisch angelegtes Level würde sonst von sich aus 4x Vektorrichtungen ermöglichen, welche entweder die Rätselkomplexität beeinträchtigen oder die Wanderung von Start zum Ziel durch den Spieler trivialisiert.

Neben der regulären Wand sprechen wir nun über die **Vektorwand**. Die Vektorwand ist farblich hervorgehoben sodass der Spieler diese Wand einfacher identifizieren und von der regulären Wand unterscheiden kann. Diese Vektorwand kann die Vektorstärke aufladen wenn der Spieler sich dagegen drückt. Die Vektor-Wand an sich besitzt aus diesem Grund drei verschiedene Faktoren, welche eine erhebliche Relevanz für die Spieleimplementierung haben.

- **Die Platzierung der Wand** in Bezug auf die Himmelsrichtung:
Die Tatsache, dass der gespeicherte Vektor eine Kraft und eine Richtung beinhaltet, gibt zu erkennen, dass das Leveldesign diesen Sachverhalt bei der Erstellung von Levelrätseln berücksichtigen muss. Somit ist es klar, dass die Platzierung der Wand wichtig ist, denn wenn der Spieler nicht in der Lage ist die Wand als Vektoraufladungshilfe ausfindig zu machen, dann kann dieser das Rätsel nicht erfolgreich abschließen. Zudem muss die Anzahl der Vektorwände an die Komplexität des Rätsels angepasst werden. Platziert man nun zu viele Vektorwände mit unterschiedlichen Richtungsausrichtungen, so geben wir dem Spieler eine unnötig hohe Anzahl an Kombinationsmöglichkeiten. Das ist besonders kritisch, da unser Spiel es dem Spieler erlaubt zwei verschiedene Vektoren nach Belieben aufzuladen und zu kombinieren.
- **Die Stärke der Vektorübertragung:**
Die Vektorwände haben keine einheitliche Vektorübertragungsrate. Jede Wand hat einen eigenen Übertragungsfaktor, welche nicht einheitlich sein müssen. Für das

Erstellen der Levels müssen brauchbare Rätsel eingeführt werden, die davon Gebrauch machen. Möglicherweise wird die Höhe der Übertragungsrate durch den Farbton der Wand visuell dargestellt. Eine niedrige Übertragungsrate wird durch eine dunkle Wandfarbe dargestellt und eine hohe Übertragungsrate durch eine helle Wandfarbe. Diese beschriebenen Überlegungen sind natürlich noch ohne Gewähr. Ebenso wird noch daran gearbeitet, ob eine spezielle Art von Vektorwand zusätzlich eingeführt wird, die eine statische Vektorkraft vorgibt. Das kontinuierliche Aufladen wird ersetzt durch die automatische Vermittlung einer unveränderlichen Vektorkraft. Dadurch werden neue interaktive Rätselmöglichkeiten präsentiert aber ist noch in Bearbeitung.

– **Die Erschöpfung der Wand:**

Im Projekt wird oft über die Verbrauchbarkeit der Wand diskutiert. Sinn dieser Überlegung liegt im vorgeheuchelten Realismus, dass die Wand durch kontinuierliches Aufladen auch irgendwann einmal nachgeben muss. Das ist besonders wahr, wenn der Spieler bereits eine immense Vektorstärke besitzt und diese noch weiter aufladen möchte. Das Konzept der Wanderschöpfung ist gänzlich zu Gunsten der Spielbalance eingeführt worden denn solch ein Konzept hilft es dem Spieler vor sich selbst zu schützen. Denn wenn Wände erschöpfen können, dann schränkt es den Spielraum des Spielers ein und somit auch seine Möglichkeiten. Die Natur unseres Spiels erlaubt es dem Spieler mit immensen Vektoren zu arbeiten, was nicht unbedingt harmlos ist. Zu groß ist die Sorge, dass der Spieler durch unüberlegtes Handeln sich durch Levelwände und Böden glitcht und somit sein Spielvergnügen negativ beeinflusst. Gleichsam zu dieser Überlegung sollte auch die Möglichkeit gegeben werden, dass sich verbrauchte Vektorwände sich wieder von selbst herrichten denn sonst gelangt der unbewusste Spieler schnell in eine ungemütliche Situation in der es kein Zurück gibt.

Bewegende Plattformen als Zusatz:

Das Leveldesign ist zum Teil eine artistische Auseinandersetzung, welches eine Auswirkung auf die Spielweise und auch auf die Ästhetik hat. Bewegende Plattformen erhöhen die allgemeine Schwierigkeit des Levels, jedoch würde dieses Objekt auch mehr Bewegungsfreude ermöglichen. Die Durchquerung von einer Reihe von sich autonom bewegend Plattformen würde einen neuen Aspekt der Spielbarkeit ansprechen: Das Gefühl von Timing.

Dynamische Objekte:

Dynamische Objekte lassen sich durch den Spieler beeinflussen. Das wohl offensichtlichste Augenmerk eines dynamischen Objektes ist die Tatsache, dass der Spieler dieses verschieben kann. Doch in unserem Spiel existieren auch dynamische Objekte, die sich zwar nicht bewegen lassen können jedoch aber zerstören. Zu Gunsten der Komplexität wird mit einer überschaubaren Anzahl an dynamischen Objekten gearbeitet obwohl diese Aussage sich im Laufe des Projektes auch ändern kann.

Kisten und Blöcke:

Kisten und Blöcke können geschoben werden. Es wird eine klare Unterscheidung gegeben, denn es wird noch diskutiert ob die Vektorkraft durch das Schieben von Blöcken aufgeladen werden können. Die natürliche Verwendung von solchen Objekten ist es, dass der Spieler auf diesen Objekten stehen kann. Das hilft um höhere Ebenen zu erreichen. Das Stapeln von Objekten ist ein Konzept, welches noch ausdiskutiert werden muss. Wichtigerweise besitzen sowohl Kisten und Blöcke ein gewisses Gewicht. Dieses Gewicht ist ausschlaggebend ob der Spieler dieses Objekt frei im Ausgangszustand bewegen kann. Falls das nicht der Fall sein sollte, dann ist noch nicht alles verloren.

Das Spiel erlaubt es dem Spieler durch die Verwendung der gespeicherten Vektorkraft sich gegen das Objekt zu rammen. Das gibt dem Spieler eine Möglichkeit seine Vektorkraft an andere Objekte abzugeben: Ein Konzept, worüber noch diskutiert werden muss denn das revolutioniert unsere Spielmöglichkeit auf eine sehr komplexe Art und Weise. Im Grunde geben wir dem Spieler durch das Einfügen einer Kiste oder eines Blocks zu erkennen, dass dem Spieler die Freiheit gewährt wird dieses Objekt durch Vektorstärke zu beeinflussen. Rätsel müssen diesen Sachverhalt anerkennen und womöglich angepasst werden.

Zerbrechliches Glas und zerfallene Wände:

Diese Objekte lassen sich zwar nicht verschieben, jedoch kann durch Einwirkung von einer bestimmten Menge an Vektorkraft diese zerstört werden. Diese Tatsache assistiert dem Leveldesign denn somit können künstliche Barrieren geschaffen werden, welche der Spieler überwinden muss. Wichtig hierbei ist, dass solch Objekte zwar keinen Gewichtungsfaktor besitzen, sondern einen Brechfaktor. Dieser Brechfaktor bestimmt wieviel Vektorkraft notwendig ist, um dieses Objekt zu überwinden. Es wird noch diskutiert, ob die Information des Brechfaktors dem Spieler in irgendeiner Weise präsentiert werden sollte.

Interaktionsmöglichkeiten des Spielcharakters:

Es wird versucht dem Spieler eine überschaubare Liste an Interaktionsmöglichkeiten zu geben, damit dieser sich ausgiebig und intuitiv mit der Spielwelt und dessen Objekten beschäftigen kann.

- An farblich markierten Vektorwänden kann sich der Spieler **aufladen**.
- Alle rumliegenden Objekte lassen sich vom Spieler in irgendeiner Weise **bewegen**
- Rissige Wände und zerbrechliches Glas lassen sich **zerstören**

Spielweise:

Für die Spielweise werden zwei verschiedene Paradigmen verfolgt.

Die Erste versucht durch ein künstlich erzeugtes Gefühl von Hektik, den Spieler schnellstmöglich zum Ziel zu bringen. Hohe Geschwindigkeiten und eine breite, in eine Richtung gezerzte, Oberwelt unterstützen diese Art der Spielweise. Diese Spielweise ist vergleichbar mit Videospielen der *Sonic*-Reihe oder vergleichbare Spiele mit einem Fokus auf

Plattform-Parkour und das Überwinden von mit Hindernissen versehenen Strecke. Anstelle von Rätseln reden wir hier über Abkürzungen und Streckenoptimierer. Auch ohne den Einsatz von Vektorkräften kann die Strecke zu einem bestimmten Grad überwunden werden. Der geschickte Einsatz verbessert jedoch die Streckenüberwindung und das erzeugt eine vom Spieler ausgehende Auffassung, dass er die Bewegung gemeistert hat.

Die Zweite ist vom Tempo her sehr gelassen, denn sie bietet dem Spieler genug Freiraum um unbeschränkt und ungezwungen die Rätsel zu absolvieren um ans Ziel zu gelangen. Diese Spielweise ist angelehnt an 3d Puzzle Plattformer, wie es Videospiele á la *Portal* und *Human Fall Flat* machen.

Das Projekt wird beide Spielweisen mit jeweils zwei Levels vertreten, somit können wir für beide Spielweisen Feedback erhalten.

Bewegungsmöglichkeiten (Movement):

Das komplette Movement soll sich schnell und möglichst flüssig spielen, es soll sich natürlich und intuitiv anfühlen die Sprünge, Dashes und ähnliches miteinander zu kombinieren um schnell und spaßig durch die Levels zu kommen. Das Movement soll Physik-basiert und dennoch flüssig und spaßig sein. Das ist wahrscheinlich das Hauptproblem und gleichzeitig eines der Hauptziele dieses Spiels.

Running

Es wird nur eine Laufgeschwindigkeit geben, diese sollte relativ hoch sein um ein Rennen oder Sprinten darzustellen, außerdem soll sich das Spiel schnell spielen. Inwiefern diese Geschwindigkeit in Puzzle-Passagen Sinn macht werden wir sehen und anschließend überdenken.

Jumping

Die Sprünge sollen Kontrolle ermöglichen und sich schnell vom Boden lösen, also soll man nach schnellem rennen und dashen möglichst ohne Verzögerung vom Boden abheben. Das sollte dazu führen ein möglichst intuitives Körpergefühl zu schaffen und ein problemfreies Vorankommen zu ermöglichen.

Die Kontrolle über den Charakter während dieser in der Luft ist kann sollte zwar nicht komplett sein, um zu erreichen das der Spieler auch plant wann er springt, gleichzeitig aber genügend Kontrolle geben um einen angenehmen Spielablauf zu ermöglichen. Ein weiterer Sprung nach einem Dash, oder ein allgemeiner Doublejump muss getestet werden, ist aber eine Möglichkeit, solange das Ergebnis ein spaßiges und schnelles Spielgefühl ergibt.

Dash

Der Dash ist das Herzstück unseres Movements und Hauptspielprinzip, dieser soll ein plötzlicher Impuls in die vorher festgelegte absolute Richtung sein und somit das Movement erweitern und zu Rätseln und Speedrun-artigen Abschnitten verwendet werden. Der Dash kann beim Laufen gegen spezielle Wände und andere Objekte aufgeladen werden.

Die Richtung der Kollision ist auch die Richtung des Dashes und wird somit absolut zum Level und dessen Orientierung verwendet.

Es können zwei voneinander unabhängige Dashes aufgeladen werden. Diese können auch nacheinander ausgelöst werden. Bei dieser Aktivierung wird getestet werden müssen ob der zweite Dash die Richtung des anderen überschreibt oder sie sich addieren.

Sie können ebenfalls miteinander kombiniert werden um einen neuen Dash in die kombinierte Richtung der beiden vorherigen zu erzeugen, die Kraft dieses Dashes wird der Durchschnitt der alten Kräfte.

6. Allgemeines FAQ:

F: Hallöchen Leutz, wollte nur fragen ob ihr hier auch Stage-Hazards einfügen werdet.

A: Stage Hazards wie Feuermauern, Wasser und rollende Indiana Jones Felsen?
Wir arbeiten daran sobald unser Grundkonzept solide ist.

F: Hallo Projekt Wallker! Kann man in diesem Spiel sterben?

A: Wenn wir Levels betrachten, die einen bodenlosen Abgrund haben, dann ja.
Dann kann man sich wieder zurück zum Levelanfang teleportieren und nochmal probieren. Wenn das Level zu lang wird, dann werden wir Checkpoints einfügen.
Es wird anfangs kein Game Over State geben, vielleicht wird das nachgereicht.

F: Was wird im Hauptmenü dargestellt? Wie kommt man in die Levels?

A: Im Hauptmenü gibt es 'Spiel starten', 'Level auswählen' und 'Spiel beenden'.
Es gibt keine allgemeine Oberwelt, man wählt die Levels direkt im Menü aus.
Bei 'Spiel starten' wird der Spieler nacheinander alle Levels durchspielen.

F: Wie sieht es mit Highscores oder Bestenlisten aus?

A: Alles in Bearbeitung.

F: Ich möchte gerne mit meinem kleinen Cousin spielen, ist das möglich?

A: Im jetzigen Stand, nein. Zur Zeit arbeiten wir nur an einer Einzelspieler-Anwendung.

F: Hat euer Spiel Musik?

A: Wir glauben nicht. Aber falls ja, dann wird Toni was billiges herbeizaubern.

F: Was wollt ihr noch implementieren, wenn ihr am Schluss noch Zeit habt?

A: Toni will noch an Charakteranimationen arbeiten um dem Spieler visuell darzustellen wenn die Figur gerade den Dash ausführt als seine Vektorkraft auslädt.