



Dokumentation zum Latenz Projekt

Virtual Reality Testumgebung: VR-Tasks mit Latenz Jittering
bei Navigations- und Interaktionsmethoden

Praxisprojekt Sommersemester 2024
im Studiengang Medieninformatik und Allgemeine Informatik
an der Fakultät 10 für Informatik und Ingenieurwissenschaften
der Technischen Hochschule Köln

vorgelegt von: Patricia Henskes Okan Leenen
Matrikelnummer: 11146126 11138565

betreut von: Prof. Dr. Florian Niebling

Gummersbach, 01.09.2024

Inhaltsverzeichnis

1 Einleitung	1
1.1 Projektbeschreibung	1
1.2 Software und Equipment	1
2 Überblick	2
2.1 Navigations- und Interaktionsmöglichkeiten	2
2.1.1 Teleport mit Niagara System	3
2.1.2 Locomotion	5
2.1.3 Grab Object Left/Right	7
2.1.4 Raycast mit anpassbarem Abstand	8
2.2 Jitter Latenz	10
2.3 Levels	11
2.3.1 Tutorial Levels	11
2.3.2 Reguläre Levels	12
2.4 VR-Task	14
3 Installationsanleitung	16
Abbildungsverzeichnis	18

1 Einleitung

Die folgende Dokumentation legt den den Aufbau und die implementierten Funktionen der von Okan Leenen und Patricia Henskes erarbeiteten Virtual Reality Testumgebung für das Praxisprojekt im Sommersemester 2024 der Studiengänge Allgemeine Informatik und Medieninformatik dar.

1.1 Projektbeschreibung

Das Latenz Projekt ist eine Virtual Reality Testumgebung, die als Grundlage für Nutzertests dienen soll, um dabei die Auswirkungen von Jitterlatenz bei Interaktion und Navigation zu untersuchen. Das Projekt umfasst zwei verschiedene Navigations- und Interaktionsmethoden, die mit einer einstellbaren Jitter-Latency versehen sind. Die Jitter-Latency kann über ein Admin Widget reguliert werden.

Das Projekt umfasst einen VR-Task, bei denen der Nutzer unterschiedliche Levels absolvieren muss. In jedem Level ist die Aufgabe, einen Würfel in mindestens eine Box zu werfen, um eine Tür zu öffnen und durch einen Teleporter zum nächsten Level zu gelangen. Die Levels variieren dabei in der Kombination der Navigations- und Interaktionsmethoden, wodurch die Auswirkungen der Jitter-Latency auf unterschiedliche Aktionen untersucht werden können.

Zu Beginn gibt es zwei Tutorial-Levels, das dem Nutzer eine kurze Einführung in die Steuerung bietet. In diesem Tutorial ist die Jitter-Latency deaktiviert, um den Nutzern die Möglichkeit zu geben, sich mit der Steuerung vertraut zu machen. Nach dem Tutorial starten die regulären Levels, in denen die Jitter-Latency dann aktiv ist.

1.2 Software und Equipment

Das Projekt ist in Unreal Engine 5 Version 5.3.2 entwickelt. Zum Testen der Umgebung werden die Brille Meta Quest 3 und die dazugehörigen Meta Quest Touch Plus Controller verwendet. Zusätzlich wird zum Testen die Oculus App genutzt, um die VR Brille über WLAN mit dem Computer zu verbinden. Das Projekt befindet sich im diesem *GitHub Repository*.

2 Überblick

Als Grundlage für das Projekt dient die Unreal Engine 5 Virtual Reality Projektvorlage. In der Vorlage sind Funktionen wie das Hand-Tracking, die Fortbewegung mittels Teleport und das Greifen von Objekten bereits mit Blueprints implementiert. Die bereits vorhandenen Funktionen wurden angepasst, erweitert und neue Funktionen wurden implementiert.

Die angepassten und neu implementierten Funktionen befinden sich unter All → Content → VRTemplate. In den anderen Ordnern wurden keine Änderungen vorgenommen.

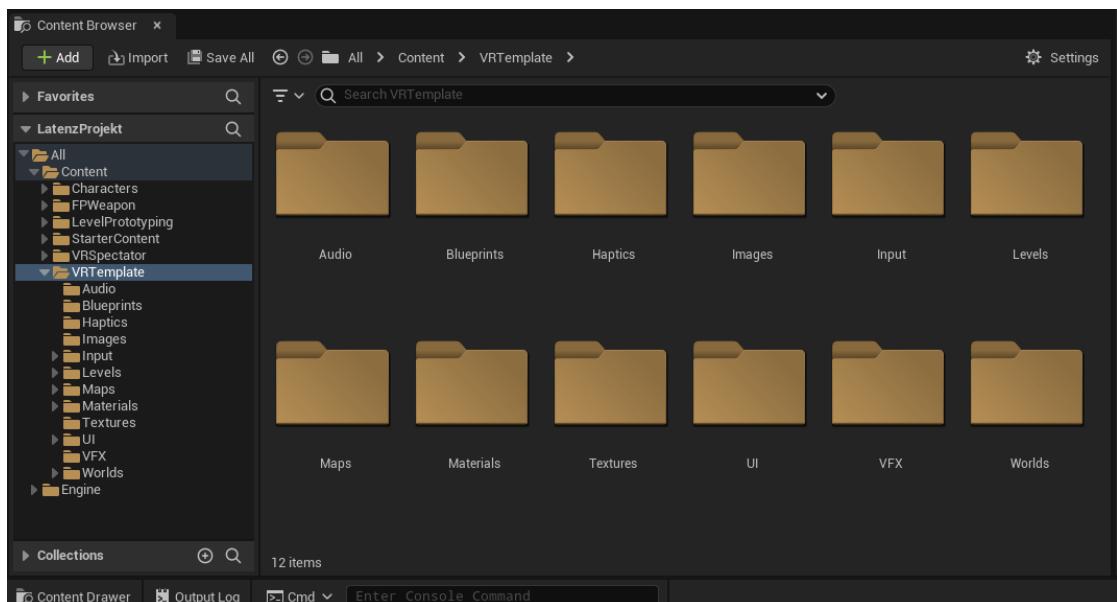


Abbildung 2.1: Ordner VRTemplate

2.1 Navigations- und Interaktionsmöglichkeiten

Im Event Graph des VR Pawns sind die Funktionen für die verschiedenen Navigations- und Interaktionsmöglichkeiten zu finden. Jede Steuerungsmöglichkeit startet mit einem Input Action. Durch den Input Action werden die Tastenbelegung und der Datentyp, mit dem gearbeitet werden soll, festgelegt.

Es sind zwei Navigationsmöglichkeiten zur Fortbewegung implementiert:

- Teleport mit Niagara System
- Locomotion

Zur Interaktion mit Objekten sind zwei Möglichkeiten implementiert:

- Grab Object Left/Right
- Raycast mit anpassbaren Abstand

2.1.1 Teleport mit Niagara System

Die Teleportationsfunktionalität ist bereits in der Virtual-Reality-Projektvorlage mit dem Niagara-System implementiert. Der Teleport Trace wird dabei als Bogen dargestellt, wie in der Abbildung 2.2 zu sehen ist.

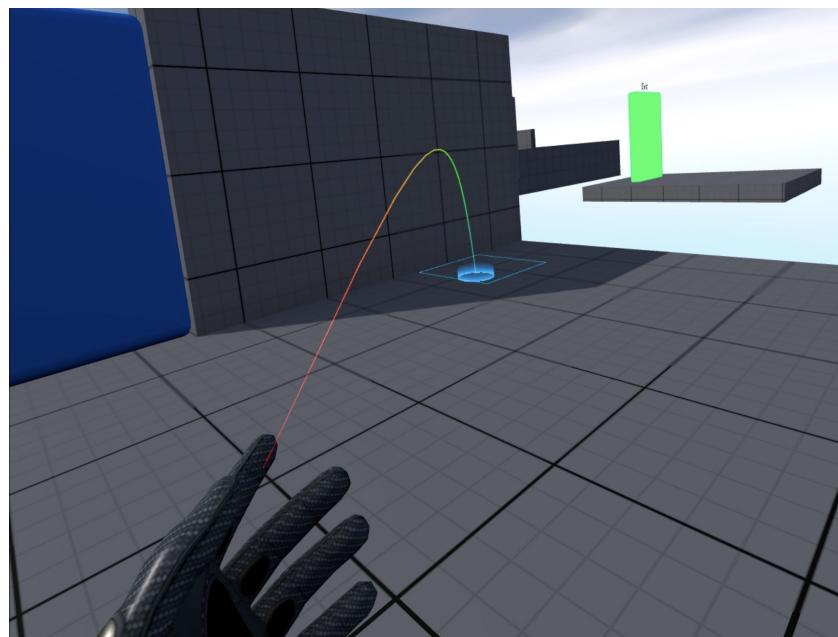


Abbildung 2.2: Teleport Trace Niagara System

Wenn der Nutzer die entsprechende Input Action auslöst (Status: Started), wird der Teleport Trace sichtbar. Während der Nutzer die Taste gedrückt hält (Status: Triggered), wird die World Location und der Forward Vector vom linken Controller abgerufen. Diese Werte werden verwendet, um die projizierte Teleportationsposition in Echtzeit zu setzen und zu aktualisieren. Sobald der Nutzer die Taste loslässt (Status: Completed), wird die letzte projizierte Teleportationsposition überprüft und der Nutzer zu dieser Position teleportiert. Zu der bestehenden Funktion wurde eine Boolean-Variable namens

CanUseTeleport als Condition hinzugefügt. Mit dieser Variable kann kontrolliert werden, ob die Teleportfunktion ausgeführt werden soll. Wenn CanUseTeleport auf true gesetzt ist, wird die Funktion nach dem Branch fortgesetzt. Ist CanUseTeleport auf false gesetzt, endet die Funktion sofort nach dem Branch.

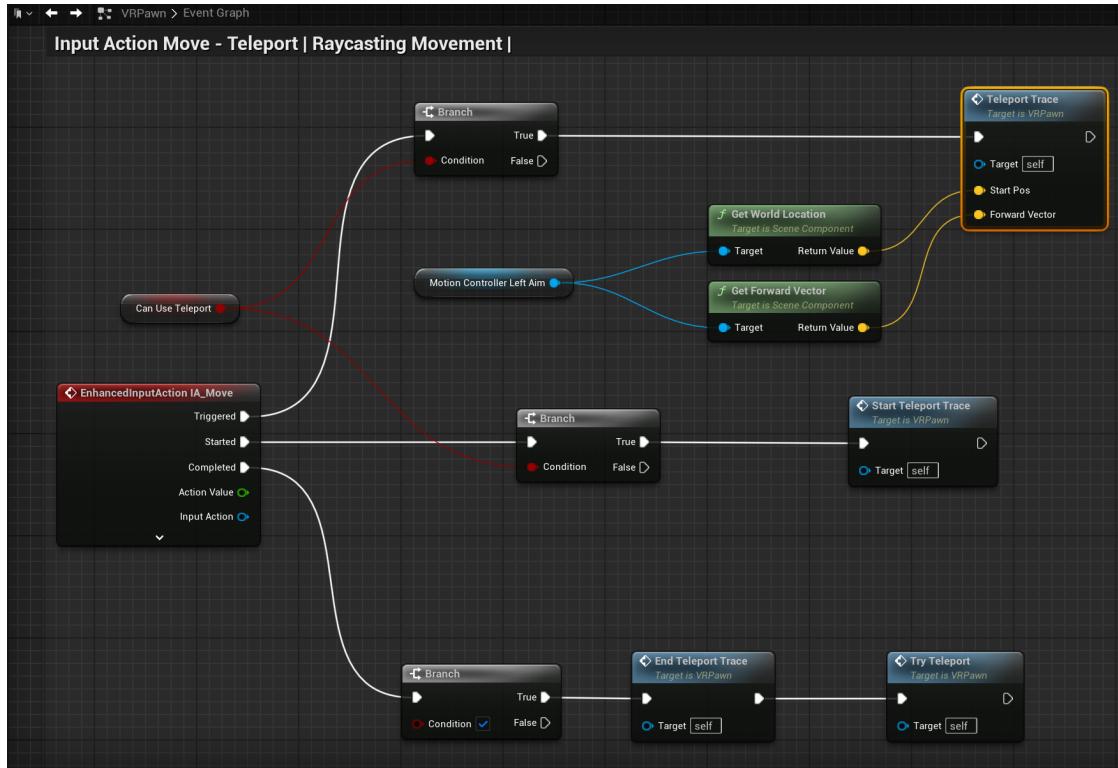


Abbildung 2.3: Teleport Blueprint

Bei der Funktion wurde die Reichweite des Teleports wurde auf 800 erhöht. Um die Reichweite der Teleportation anzupassen, kann der Float in der Variable Local Teleport Launch Speed entsprechend erhöht oder verringert werden. Die Variable befindet sich innerhalb des Teleport Trace Blueprint Nodes, der im Event Graph des VR Pawns implementiert ist. Siehe Abbildung 2.3 für die Position des Teleport Trace Nodes. Für die Position innerhalb des Teleport Trace Nodes siehe Abbildung 2.4.

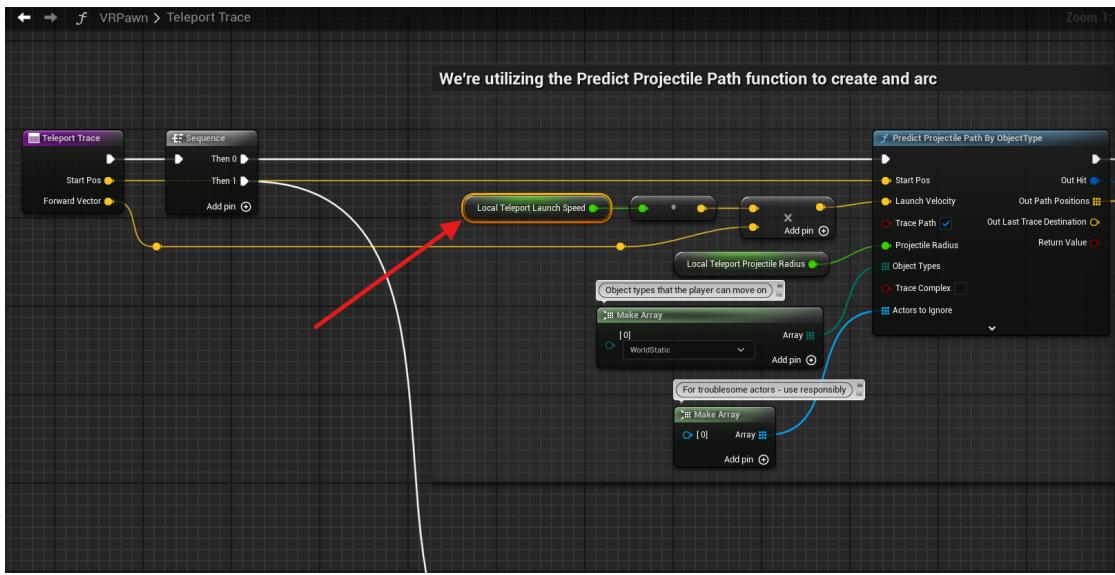


Abbildung 2.4: Local Teleport Launch Speed

Der Input Action ist standardmäßig auf den Thumbstick des rechten Controllers gesetzt. Der Input Action wurde auf die Taste X des linken Controllers umgelegt, da der rechte Thumbstick für eine andere Steuerung genutzt wird.



Abbildung 2.5: Teleport Input Action

2.1.2 Locomotion

Die Locomotion Funktion als Navigationsmethode ist selbst implementiert. Die Funktion verwendet den linken Thumbstick als Input Action. Dabei wird im Input Action ein 2D Vektor als Value benutzt.

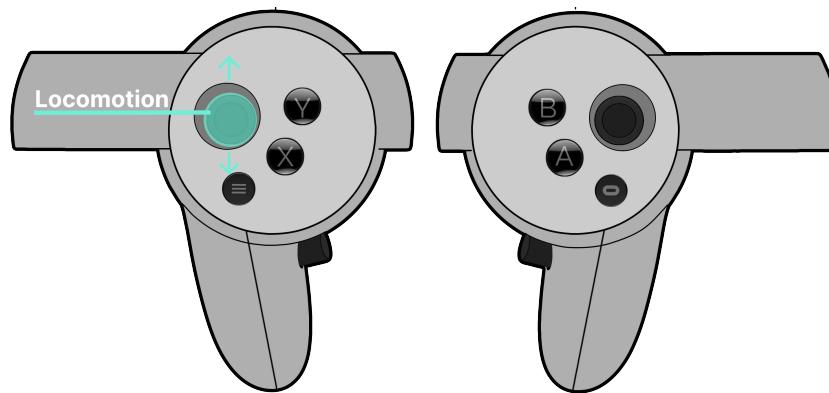


Abbildung 2.6: Locomotion Input Action

Während der Nutzer den Input Action gedrückt hält (Status: Triggered), bewegt sich der Spieler. Durch die Verbindung des Action Values Y vom Enhanced Input Action mit dem Scale Value des Add Movement Input wird das Vorwärts- und Rückwärtsbewegen des Thumbsticks erkannt und entsprechend an der World Direction skaliert. Damit der Spieler sich in Blickrichtung bewegen kann, wird von der Camera der X und Y Value vom Forward Vector geholt. Der X und Y Value wird jeweils mit 0,5 multipliziert und dem Movement Input als World Dirction X und Y übergeben. Da der Action Value X vom EnhancedInputAction mit nichts verbunden ist, löst das Bewegen des Thumbsticks nach rechts oder links nichts aus.

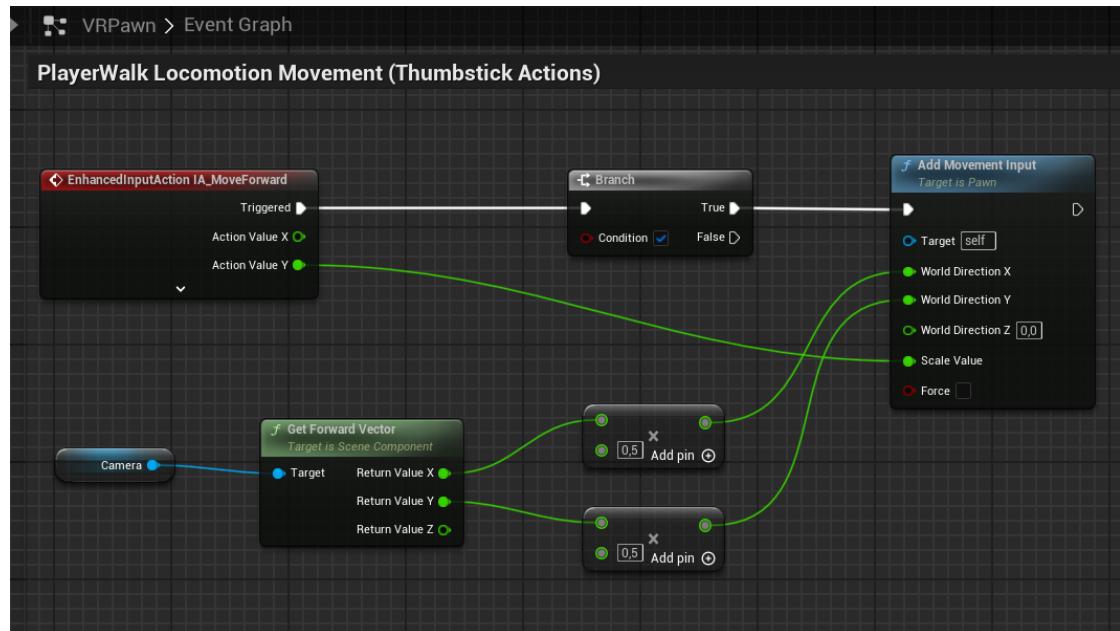


Abbildung 2.7: Locomotion Blueprint

2.1.3 Grab Object Left/Right

Die Funktion zum Greifen eines Objekts ist bereits in der Virtual-Reality-Projektvorlage implementiert. Den Blueprint gibt es sowohl für den linken als auch für den rechten Controller jeweils separat. Mit der Grab Funktion kann der Nutzer greifbare Objekte direkt greifen. Solange der Input Action gedrückt ist, wird das Objekt gehalten. Sobald der Input Action losgelassen wird, wird das Objekt losgelassen. Als Input Action wird die Grip-Taste am Controller verwendet.

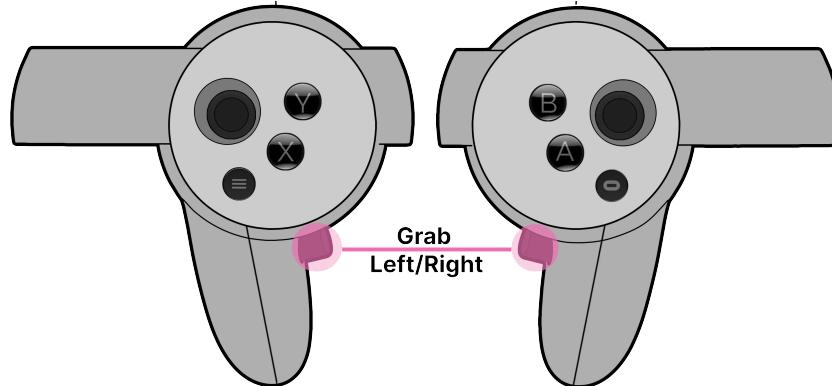


Abbildung 2.8: Grab Input Action

Zu der bestehenden Funktion wurde eine Boolean-Variable namens CanUseGrab als Condition hinzugefügt. Mit dieser Variable kann kontrolliert werden, ob die Grabfunktion ausgeführt werden soll. Wenn CanUseTeleport auf true gesetzt ist, wird die Funktion nach dem Branch fortgesetzt. Ist CanUseTeleport auf false gesetzt, endet die Funktion.

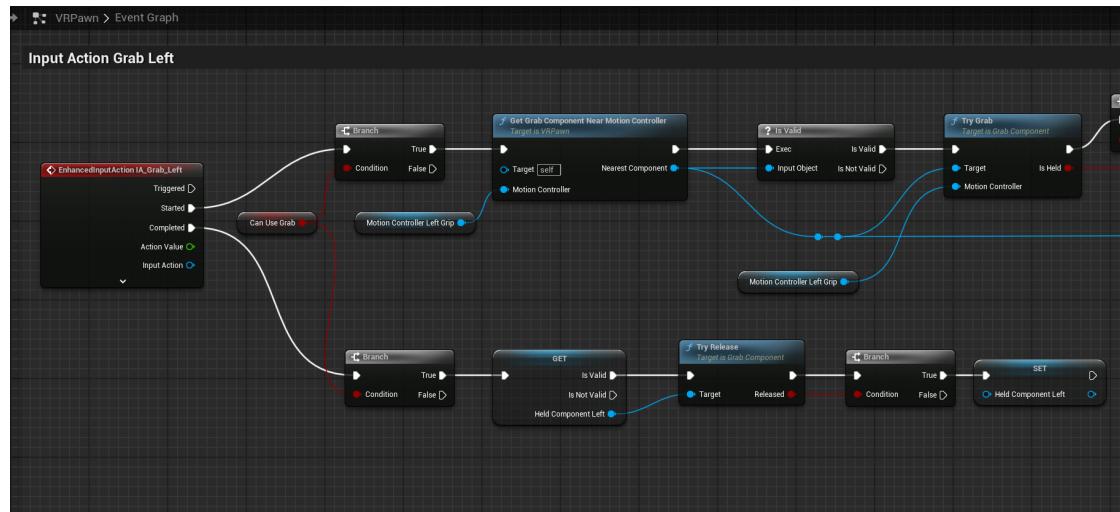


Abbildung 2.9: Grab Blueprint

2.1.4 Raycast mit anpassbarem Abstand

Die Raycast Funktion als Interaktionsmethode zum Bewegen von Objekten ist selbst implementiert. Hierfür gibt es zwei Input Actions:

- Die Shoot-Taste auf dem rechten Controller wird verwendet, um ein Objekt aus der Entfernung aufzuheben und zu halten.
- Der Thumstick auf dem rechten Controller kann die Variable Grab Distance erhöhen oder verringern, wodurch der Abstand vom gehaltenen Objekts angepasst werden kann.

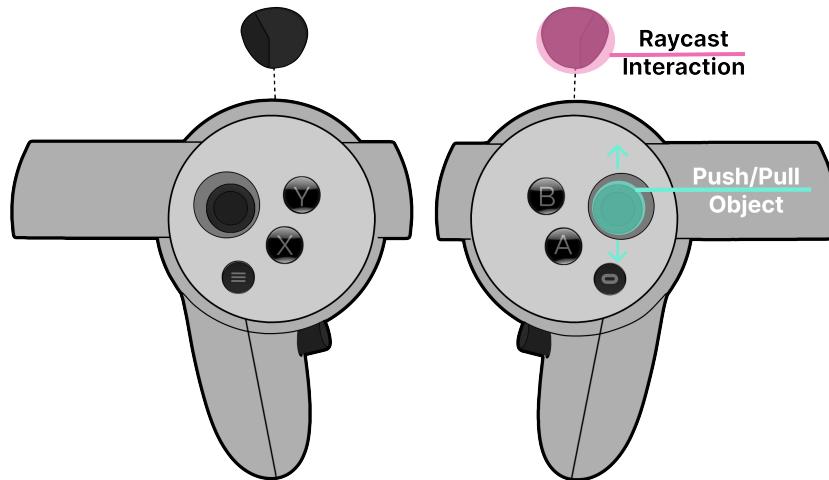


Abbildung 2.10: Raycast Input Action

Beim Input Action der Shoot-Taste mit dem Status Started wird die Funktion Line Trace by Channel aufgerufen. Als Startvektor wird die World Location des Motion Controller Right Aim verwendet. Der Endpunkt wird durch Multiplikation des Forward Vectors mit 500 und anschließender Addition des Startvektors bestimmt. Die Zahl 500 definiert dabei die Länge des Line Trace, siehe Abbildung 2.11.

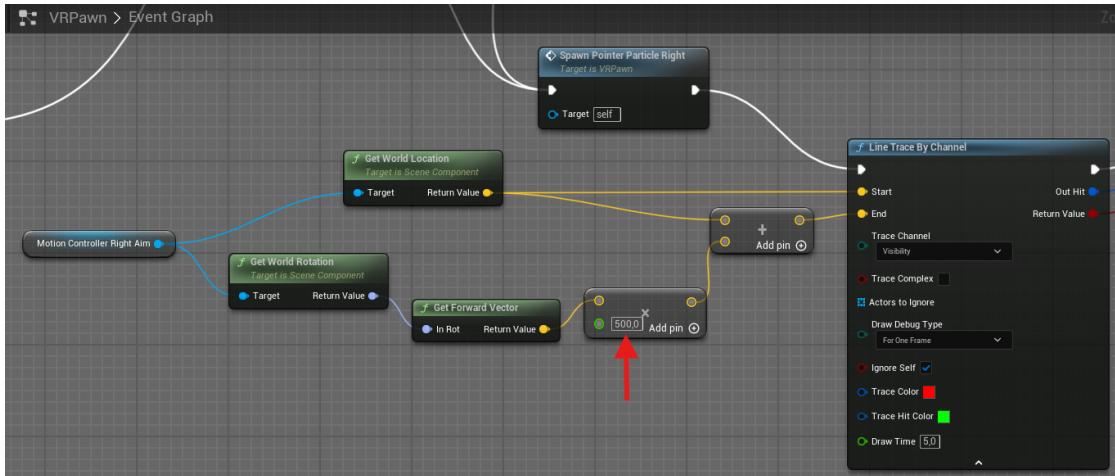


Abbildung 2.11: Status Started Line Trace By Channel

Wenn ein greifbares Objekt durch den Line Trace getroffen wird, wird die World Location des getroffenen Objekts erfasst, und die Variable Is Object Grabbed wird auf true gesetzt. Der Status Triggered und Ongoing überprüft, ob Is Object Grabbed auf true gesetzt ist. Wenn dies der Fall ist, wird die Position des getroffenen Objekts entlang des Line Trace gesetzt. Die Position auf dem Line Trace wird durch die Variable Grab Distance bestimmt. Beim Input Action Status Completed wird die Grab Distance immer auf 235 zurückgesetzt, siehe Abbildung 2.12.

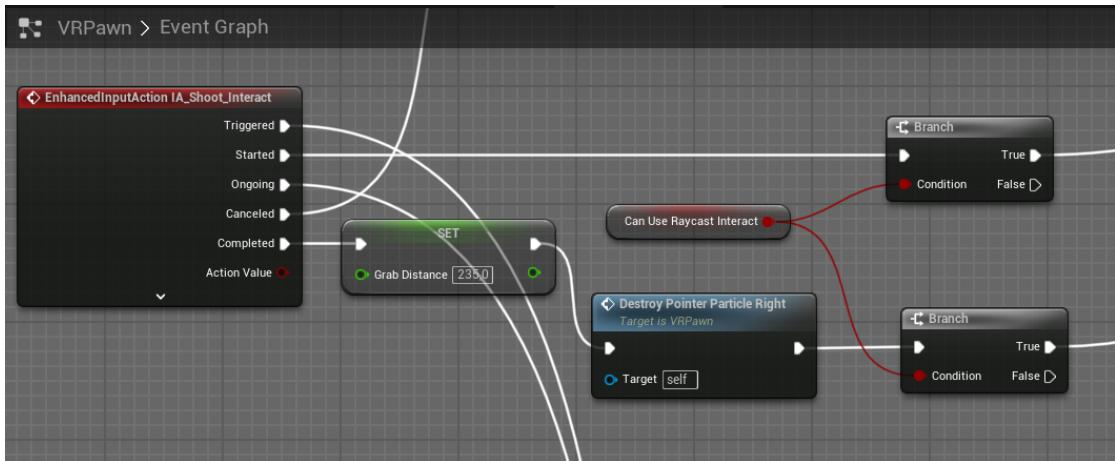


Abbildung 2.12: Status Completed Grab Distance 235,00

Die Boolean-Variable CanUseRaycastInteract kontrolliert, ob die Funktion aktiv oder inaktiv ist. Wenn CanUseRaycastInteract auf true gesetzt ist, wird die Funktion nach dem Branch fortgesetzt. Ist CanUseRaycastInteract auf false gesetzt, wird die Funktion nicht weiter ausgeführt.

2.2 Jitter Latenz

Bei den zuvor genannten Navigations- und Interaktionsmöglichkeiten wird an mehreren Stellen eine Jitter-Latenz hinzugefügt. Zur Kontrolle der Jitter-Latenz gibt es zwei Variablen:

- Max Delay: Bestimmt die Länge der Latenz in Sekunden.
- Percent Delay Occur: Gibt die Wahrscheinlichkeit an, dass die Latenz eintritt.

In den Navigations- und Interaktionsfunktionen wird an mehreren Stellen ein Branch mit einer Condition hinzugefügt. Als Condition wird ein zufälliger Float-Wert zwischen 0 und 1 generiert und überprüft, ob dieser größer ist als die Variable Percent Delay Occur. Wenn die Condition true ist, findet kein Delay statt. Ist die Condition false, tritt eine Latenz mit der in der Variable Max Delay festgelegten Dauer ein.

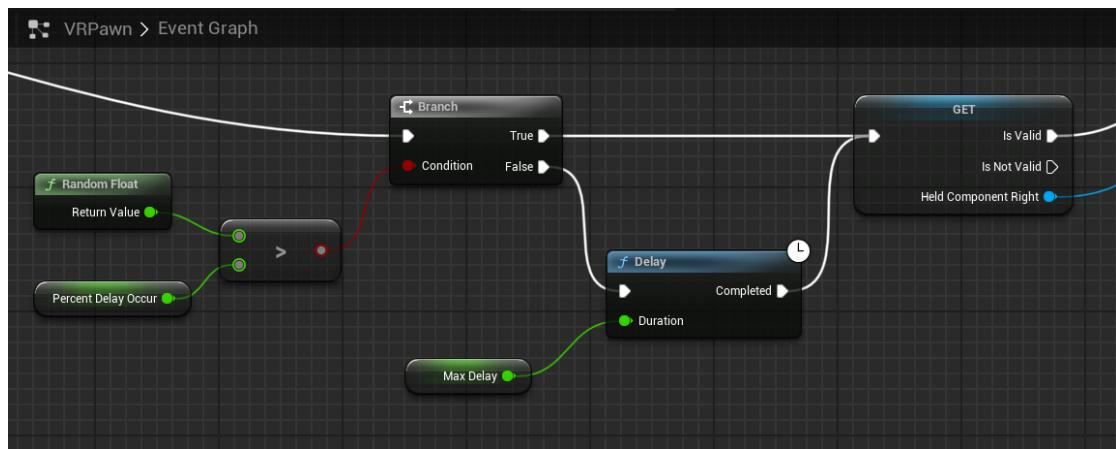


Abbildung 2.13: Branch für Jitter Latenz

Im Admin-Widget können die Werte für Max Delay und Percent Delay Occur jederzeit angepasst werden und der aktuelle eingestellte Wert wird an der Stelle, wo in Abbildung 2.14 'test' steht, angezeigt. Zum Öffnen und Schließen des Admin-Widgets müssen die Tasten X und A gleichzeitig gedrückt werden.

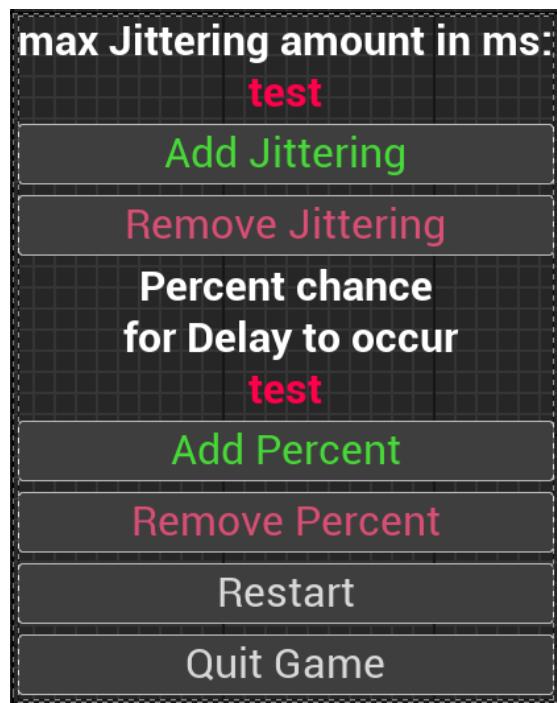


Abbildung 2.14: Admin Widget

2.3 Levels

Es existiert eine Map namens LobbyMap, in der sämtliche Levels enthalten sind. Die Levels sind entsprechend der aktivierten Navigations- und Interaktionsmöglichkeiten, wie in der Abbildung 2.15 benannt. Insgesamt gibt es acht Levels. In jedem Level ist genau eine Interaktionsmethode sowie entweder eine Navigationsmethode oder keine Navigationsmethode aktiviert. Dies wird durch die Boolean-Variablen CanUseTeleport, CanUseRaycastInteract und CanUseGrab ermöglicht. Diese Variablen werden je nach Level, zu der der Nutzer teleportiert wird, entsprechend aktiviert oder deaktiviert.

Der Nutzer beginnt in der LobbyMap, wo ein Informationstext angezeigt wird. Um den Testverlauf zu starten, muss der Nutzer den Start-Button betätigen. Nach Betätigung des Start-Buttons wird der Nutzer in das Tutorial teleportiert.

2.3.1 Tutorial Levels

Es gibt zwei Tutorial-Levels, die der Nutzer zunächst abschließen muss, um zu den regulären Levels zu gelangen. Das erste Tutorial-Level ist TutorialGrabFirstThenMove, in dem die Funktionen Grab und Locomotion aktiviert sind. Anweisungen zur Steuerung und zum auszuführenden VR-Task werden an der Wand gezeigt. Nachdem der Nutzer die

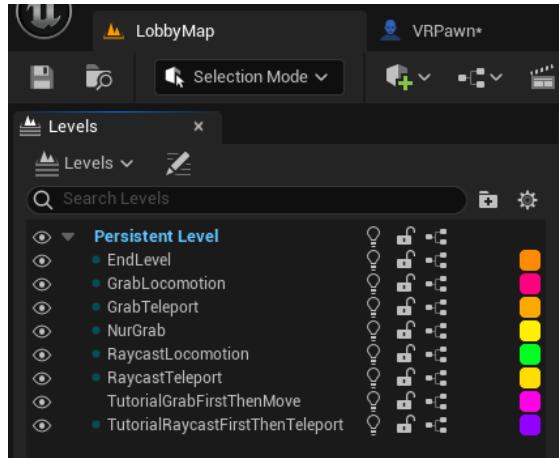


Abbildung 2.15: Übersicht der Levels

Aufgabe erfolgreich abgeschlossen hat, wird er in das nächste Tutorial-Level, Tutorial-RaycastFirstThenTeleport, teleportiert. In diesem Level sind die Raycast-Interaktion mit Objekten sowie Teleportation zur Fortbewegung aktiviert. Auch hier werden die Steuerungsanweisungen an der Wand dargestellt. Sobald der Nutzer sich durch den Teleport bewegt, wird er in die regulären Levels teleportiert.

2.3.2 Reguläre Levels

Nach den Tutorial-Levels wird die Reihenfolge der regulären Levels zufällig bestimmt, wobei der Nutzer jedes Level genau einmal betreten muss. Um ein Level abzuschließen, muss der Nutzer den Teleporter durchqueren, wodurch das Event ActorBeginOverlap ausgelöst wird und der Nutzer zum nächsten Level teleportiert wird. Der Teleporter selbst verfügt über eine Trigger Box, und jedes Level enthält eine Kapsel, die an einer spezifischen Position im jeweiligen Level platziert ist. Durch Berührung mit der Trigger Box wird die Teleportation zum nächsten Level ausgelöst, wobei die World Location des Nutzers auf die World Location der Kapsel im nächsten Level gesetzt wird.

Die Randomisierung der Levels ist im Event Graph des Teleporters implementiert. Dort werden zwei Arrays erstellt: Ein Array namens Max Array Int mit Integer-Werten von 0 bis 4 und ein Array namens All Levels Array mit den fünf regulären Levels. Es ist wichtig, dass beide Arrays die gleiche Anzahl von Elementen enthalten, da die Werte im Max Array Int den Indizes des All Levels Array darstellen.

Das Max Array Int wird zunächst randomisiert, und ein zufälliges Element wird ausgewählt. Anschließend wird überprüft, ob dieses Element dem Index eines der Elemente im All Levels Array entspricht. Wenn dies der Fall ist, wird das entsprechende Level geladen, und die Position des Nutzers wird auf die World Location der Kapsel in diesem Level

gesetzt, siehe Abbildung 2.16. Danach wird das ausgewählte Element aus dem Max Array Int entfernt.

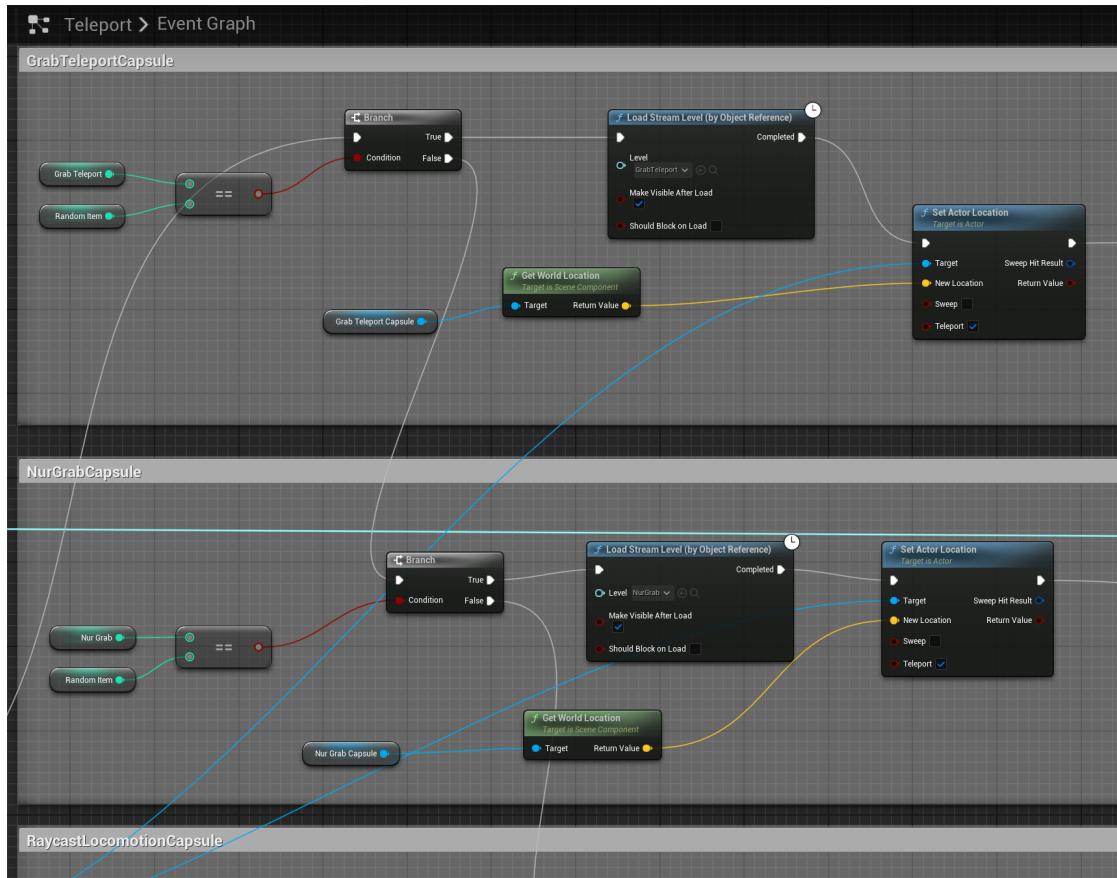


Abbildung 2.16: Überprüfung Random Item = Item Index

Dieser Vorgang wird so lange wiederholt, bis keine Elemente mehr im Max Array Int vorhanden sind. Wenn der Nutzer das letzte reguläre Level durch den Teleporter verlässt, wird seine World Location auf die Kapsel im EndLevel gesetzt, und der Teleport erfolgt. Das EndLevel besteht aus einem Raum mit einem Informationstext sowie einem Quit-Button.

2.4 VR-Task

In jedem Level, mit Ausnahme der LobbyMap und des EndLevel, muss der Nutzer mithilfe der aktivierten Navigations- und Interaktionsmethoden einen VR-Task erfüllen. Der Task besteht darin, mindestens einen Würfel in jede Box zu werfen, um eine Tür zu öffnen und den Teleporter zum nächsten Level zu erreichen.

Am Boden der Boxen sind versteckte Druckplatten (Pressure Switches) angebracht, die aktiviert werden müssen, um eine Tür zu öffnen. Jede Druckplatte ist einer bestimmten Tür im Level zugeordnet und in den Details der Tür ist angegeben, wie viele Druckplatten aktiviert sein müssen, damit sie sich öffnet, siehe Abbildungen 2.17 und 2.18. Sobald alle erforderlichen Platten aktiviert sind, macht die Tür den Weg frei.

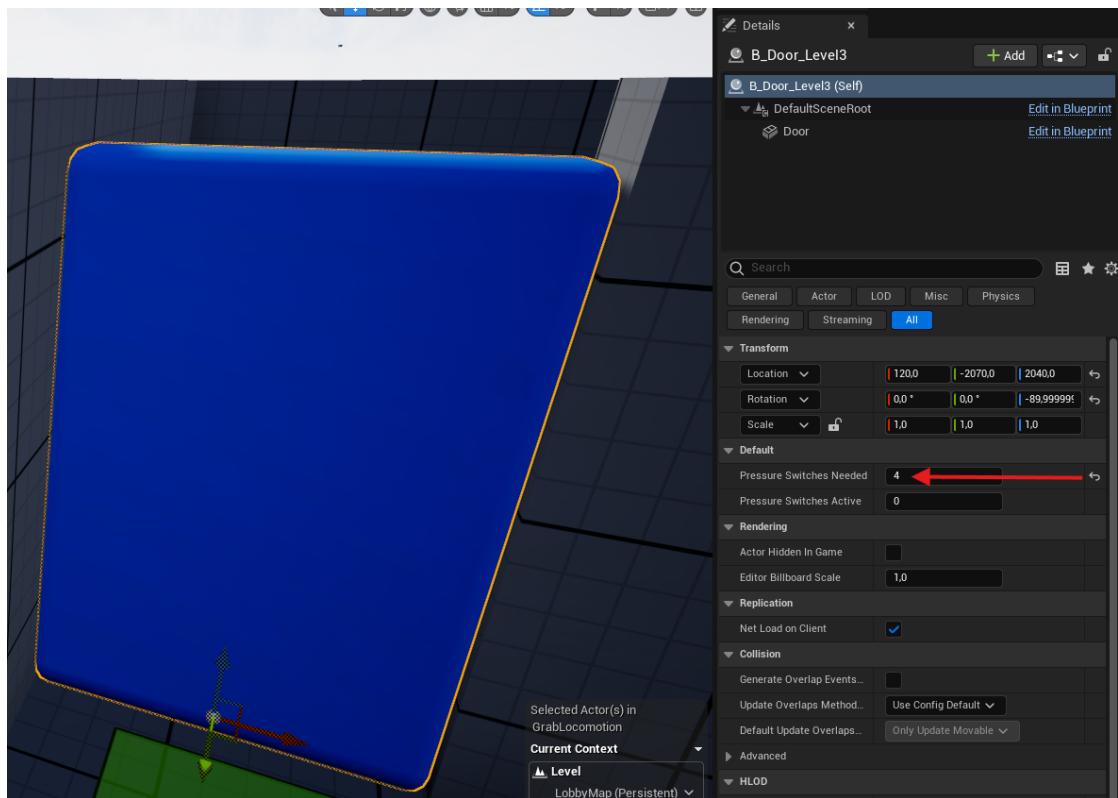


Abbildung 2.17: Details zur Tür

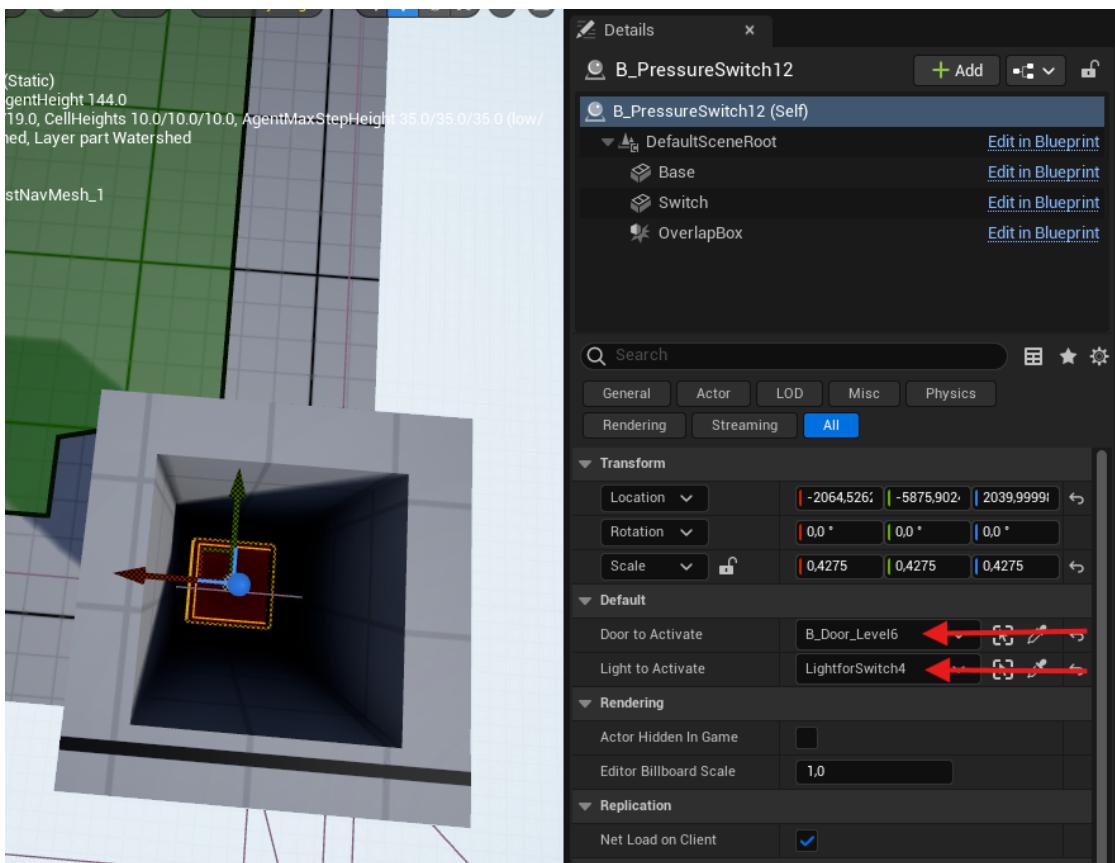


Abbildung 2.18: Details zum Pressure Switch

Jede Box verfügt über einen Indikator, der anzeigt, ob sich bereits ein Würfel auf der Druckplatte befindet. Der Indikator leuchtet rot, solange die Druckplatte nicht aktiviert ist. Sobald die Druckplatte in einer Box aktiviert wurde, wechselt der Indikator auf grün. Dazu wird in den Details vom Pressure Switch, wie in Abbildung 2.18 der Indikator für diesen Switch angegeben, der bei Aktivierung Grün werden soll.

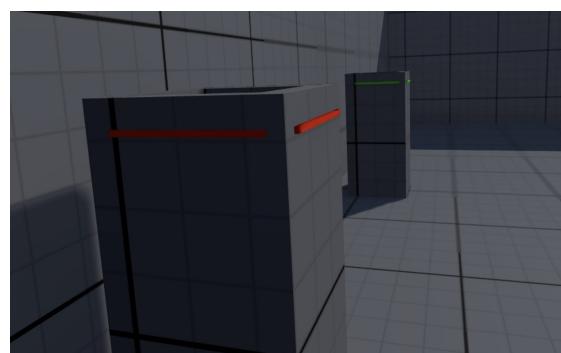


Abbildung 2.19: Pressure Switch Indikator

3 Installationsanleitung

Um das Projekt öffnen zu können, installiere Unreal Engine 5. Hierbei ist es wichtig die Version 5.3.2 zu verwenden, da es sonst zu Fehlern kommen kann.

Zum Installieren des Projektes öffne das *GitHub Repository*. Navigiere zum Reiter Code und klicke auf das grüne Feld Code. Es sollten mehrere Optionen angezeigt werden, wähle hier Download ZIP aus, um das Projekt herunter zu laden.

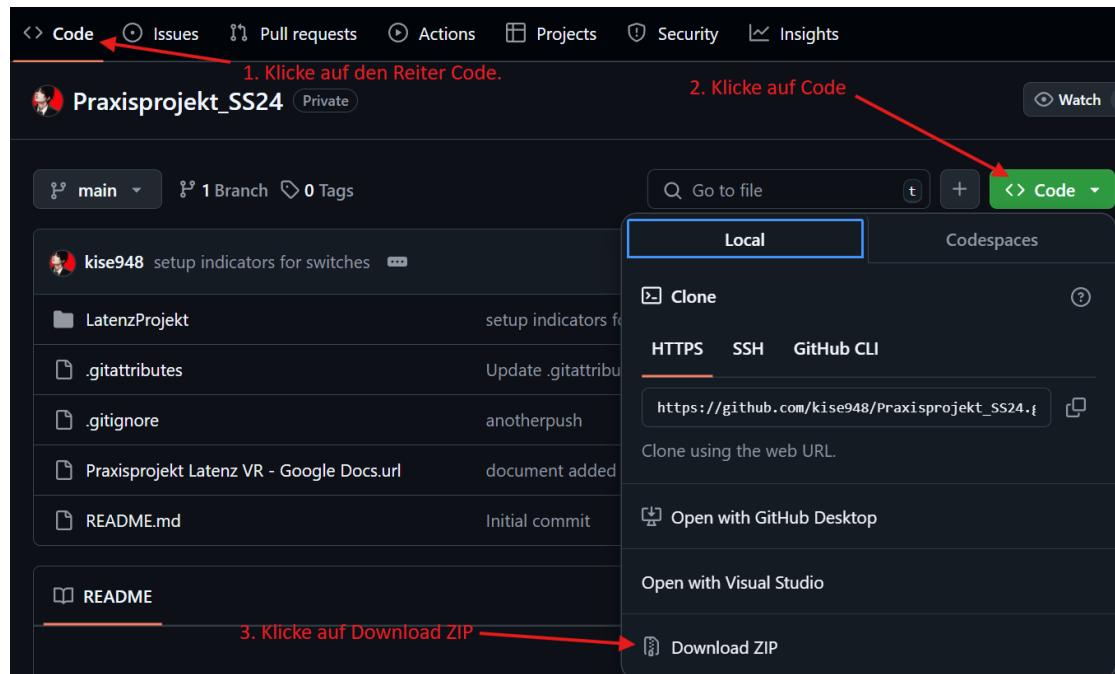


Abbildung 3.1: Installationsanleitung GitHub Repository

Nachdem die ZIP Datei herunter geladen ist, entpacke als nächstes die ZIP Datei. Öffne den entpackten Ordner und öffne darin den Ordner LatenzProjekt. Öffne die Datei LatenzProjekt.uproject mit Unreal Engine 5.3.2. Baue als nächstes das Projekt in Unreal Engine. Klicke hierfür in oben links in der Leiste auf Build und anschließend auf Build All Levels.

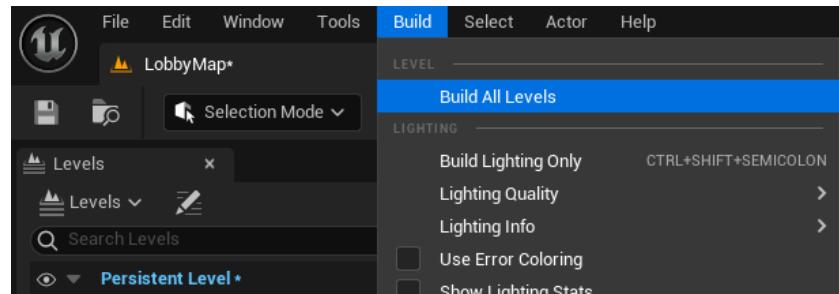


Abbildung 3.2: Installationsanleitung Build All Levels

Abbildungsverzeichnis

2.1	Ordner VRTemplate	2
2.2	Teleport Trace Niagara System	3
2.3	Teleport Blueprint	4
2.4	Local Teleport Launch Speed	5
2.5	Teleport Input Action	5
2.6	Locomotion Input Action	6
2.7	Locomotion Blueprint	6
2.8	Grab Input Action	7
2.9	Grab Blueprint	7
2.10	Raycast Input Action	8
2.11	Status Started Line Trace By Channel	9
2.12	Status Completed Grab Distance 235,00	9
2.13	Branch für Jitter Latenz	10
2.14	Admin Widget	11
2.15	Übersicht der Levels	12
2.16	Überprüfung Random Item = Item Index	13
2.17	Details zür Tür	14
2.18	Details zum Pressure Switch	15
2.19	Pressure Switch Indikator	15
3.1	Installationsanleitung GitHub Repository	16
3.2	Installationsanleitung Build All Levels	17