

## Testat 3: Supermarkt VR Nutzerstudie

### Login

Nutzen Sie zur Anmeldung das lokale Login

. \student

mit Passwort

student1

Für diesen Account sind die Oculus-Headsets vorkonfiguriert.

## Einführung in das Open Source Unity SDK: Innoactive Creator

Das Unity Plugin „*Innoactive Creator*“ ist ein simples Tool, um Prozessabläufe und Trainings für VR-Headsets zu erstellen. Es ermöglicht Ihnen, per HMD mit einer virtuellen Welt zu interagieren und das ganz ohne Coding! (Zumindest in der ersten Aufgabe ist das noch nicht nötig.)

Abläufe werden grafisch konfiguriert.

Sie entscheiden selbst, was Sie damit erstellen wollen:

- ein virtuelles Training,
- ein VR-Spiel,
- eine VR-Nutzerstudie.

Da Sie UX-Design studieren, werden wir in den kommenden Übungen Anwendungen erstellen, wie sie bei Nutzerstudien zum Einsatz kommen könnten. Unsere Aufgaben sollen Sie dazu befähigen, ein selbst erstelltes UI-Konzept mithilfe von VR erlebbar zu machen.

### Lerninhalte

- 3D-Objekte greifbar machen.
- Das Drehbuch einer Studie in eine VR Simulation überführen.
- Wireframes aus Illustrator in Unity importieren.
- Buttons der Wireframes für eine VR-Nutzerstudie mit Interaktion versehen.

### Voraussetzungen:

- VR-Brille (z.B. Oculus Rift oder Quest, HTC Vive)
- Unity 2019.4.x
- Unity Projekt „Supermarkt VR Nutzerstudie“ aus Moodle (enthält Innoactive Creator v 2.7)

## Dokumentation - Innoactive Creator

Die Offizielle Dokumentation des Entwicklers finden Sie hier:

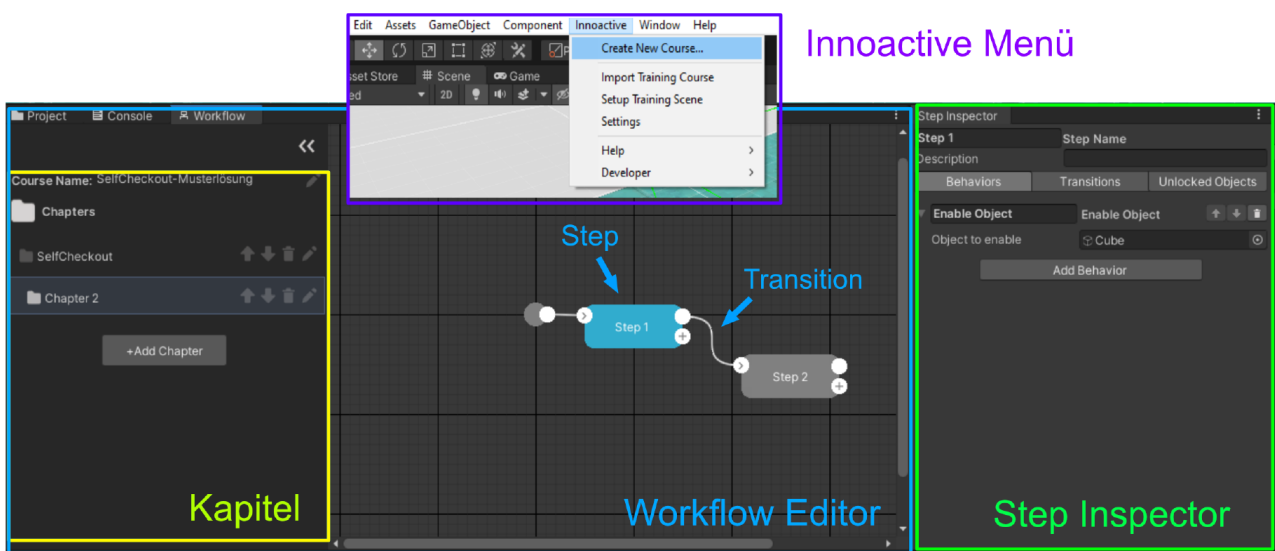
<https://developers.innoactive.de/documentation/creator/v2.8.0/>

## Download - Innoactive Creator

In unserem Kurs ist das Plugin bereits enthalten. Falls Sie selbst am Ball bleiben wollen, können Sie das Open Source Plugin auf der Github Seite von Innoactive beziehen.

<https://github.com/Innoactive/Creator>

## Wichtige Begriffe



Innoactive Menü	Das Menü finden Sie oben in der Menüleiste. Hier erstellen Sie einen neuen Kurs.
Workflow Editor	Hier reihen Sie grafische Bausteine aneinander, um einen Kurs zu erzeugen.
Kapitel	Damit Ihre Szene übersichtlich bleibt, können Sie Ihren Kurs in Kapitel strukturieren.
Step	Ein Baustein des Workflow Editors. In Jedem Step entscheiden Sie, was geschehen soll.
Step Inspector	Wenn Sie auf einen Step klicken, erscheint der Step Inspector. In diesem Fenster legen Sie die Funktionalität und Eigenschaften eines Steps fest.
Transition	Nach jedem Step gibt es eine Transition zum nächsten Step Ihrer Wahl. Hier können Sie auch Schleifen oder if / else Entscheidungen einfügen.
Condition	Jeder Step hat eine Condition. Diese muss erfüllt werden, damit die Transition zum nächsten Step durchlaufen wird. Wenn ein Step keine Condition besitzt, wird dieser sofort „erfolgreich“ abgeschlossen
Behaviour	Mit Behaviours können Sie Einstellungen an der Szene vornehmen. Behaviours haben keinen Einfluss auf den Verlauf Ihres Kurses. Sie werden verwendet, um beispielsweise ein Objekt zu aktivieren / zu deaktivieren

## Übungsaufgabe

- A) Öffnen Sie das bereitgestellte Unity Projekt im Unity Hub. Öffnen Sie anschließend die Szene *Assets* → „00 VR Nutzerstudie“ → *Scenes* → *Übungsszene*. Hier ist eine kleine Einführungsaufgabe zum Ausprobieren, damit Sie sich mit der Bedienung der Oculus vertraut machen können.
- B) An Ihren Oculus Controllern benötigen Sie nur drei Tasten zur Interaktion mit der virtuellen Welt.

Den seitlichen *Grip Button* (Mittelfinger) verwenden Sie zum Greifen von Objekten, den an der Front befindlichen *Trigger Button* (Zeigefinger) verwenden Sie zum Teleportieren. Mit den oben befindlichen *Thumb Sticks* können Sie die Ansicht rotieren.

**Greifen:** Halten Sie den virtuellen Controller in ein Objekt und drücken Sie den *Grip Button*.

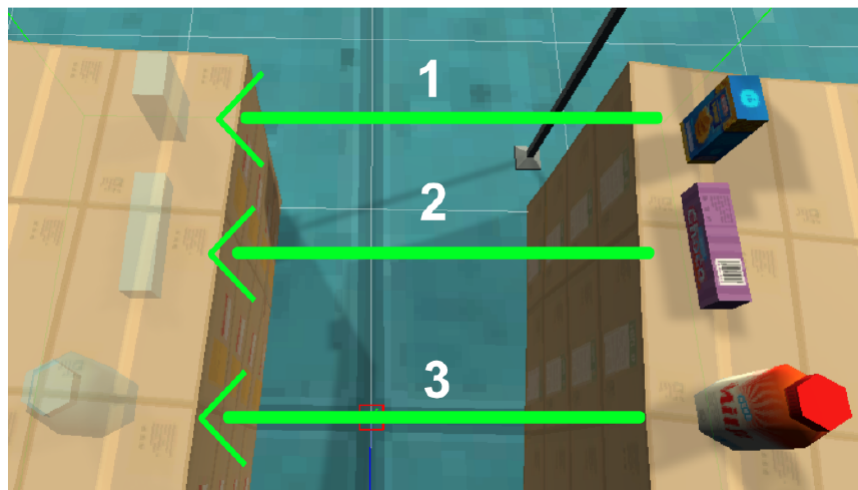
**Teleportieren:** Halten Sie den *Trigger Button* gedrückt. Ein Strahl wird in die Umgebung geschossen, falls das Teleportieren möglich ist, ist der Strahl weiß, ansonsten rot. Beim Loslassen der Taste werden Sie an die anvisierte Stelle gebeamt.

**Drehen:** Bewegen Sie den *Thumb Stick* kurz nach links oder rechts, um in die jeweilige Richtung zu rotieren. Die Rotation erfolgt bei jeder Aktivierung um ein festes Winkelinkrement (*Ratcheting*).



- C) In der Szene müssen Sie ein paar Artikel greifen und auf der gegenüberliegenden Seite ablegen. Der zu greifende Artikel wird immer gelb hervorgehoben. Die Ablagezone ist immer leicht bläulich hervorgehoben.

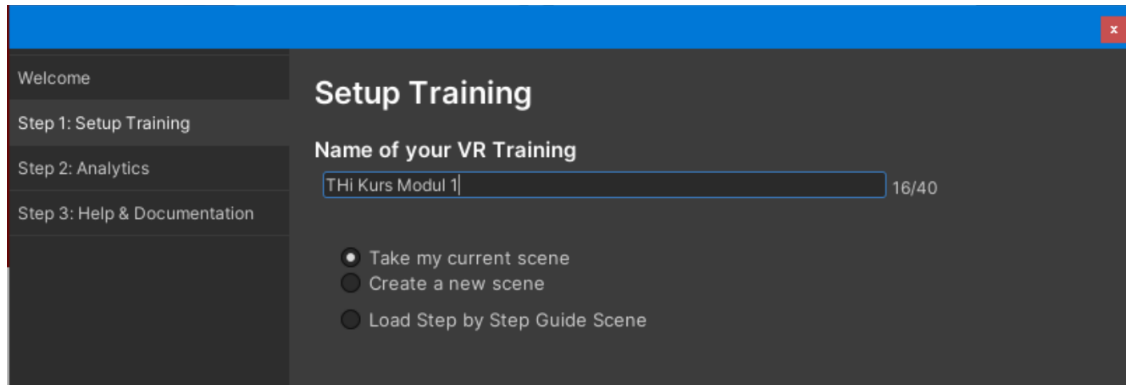
Nach dem Platzieren der Artikel, müssen Sie sich zum Weinregal teleportieren. Sobald Sie dort ankommen, prasselt zur Belohnung ein Konfettiregen auf Sie nieder.



- D) Drücken Sie in Unity auf *Play* und versuchen Sie es.

### Aufgabe 1 (Einen neuen Kurs anlegen)

- A) Navigieren Sie im Unity Projekt zur Szene *Assets* → „00 VR Nutzerstudie“ → *Scenes* → *Self-Checkout* und öffnen diese.
- B) Über das *Innoactive Menü* lässt sich ein neuer Kurs anlegen. (*Innoactive* → *Create new Course*) Folgen Sie Schritt für Schritt den Anweisungen des Pop-Up Fensters und vergeben Sie einen Namen für den Kurs.



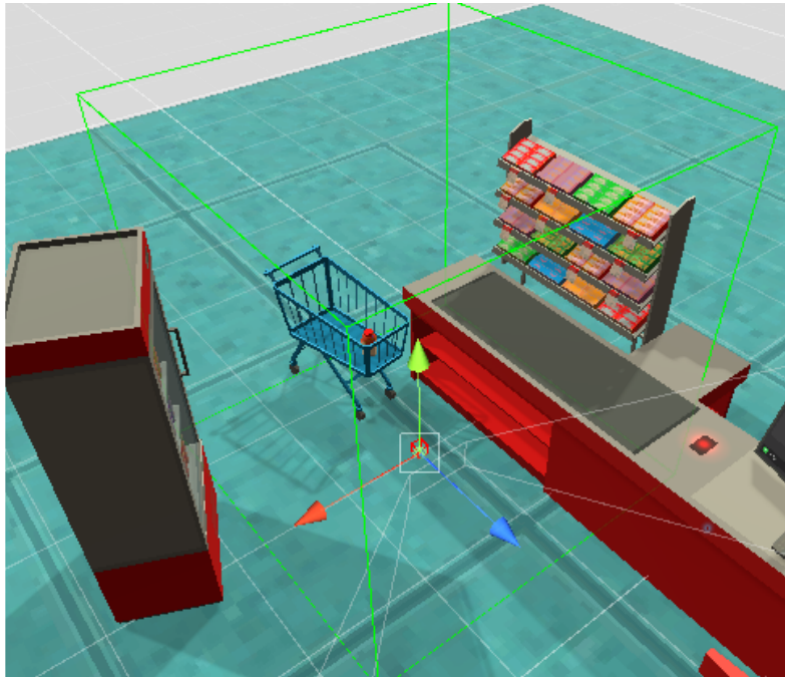
C) Das Fenster *Workflow Editor* wurde erstellt. Positionieren Sie es passend in Unity.

D) In der Hierarchie wurden drei Objekte erstellt:

<b>[XR_Setup]</b>	Enthält das virtuelle HMD sowie die Controller der Szene.
<b>[Course Controller]</b>	Hier können Sie zwischen Standalone und Kabelgebunden HMD's wechseln.
<b>[TRAINING_CONFIGURATION]</b>	Hiermit wechseln sie zwischen allen vorhanden Kursen des Unity Projekts.  <i>Tipp:</i> Im Inspektor finden Sie bei diesem Objekt den Button „Open Course in Workflow window“, mit dem Sie den <i>Workflow Editor</i> jederzeit zurückholen können, sollten Sie ihn einmal versehentlich schließen.

Mit dem [XR\_Setup] wird festgelegt, an welcher Stelle einer virtuellen Welt man sich befinden wird. Verschieben Sie das [XR\_Setup] so, dass Sie sich beim Starten der Applikation mittig vor dem Einkaufswagen befinden. Rotieren Sie das Objekt auch noch, so dass Sie beim Start auf den Einkaufswagen blicken.

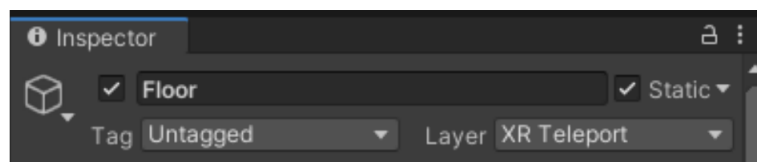
Überprüfen Sie die Position und Orientierung im *PlayMode*.



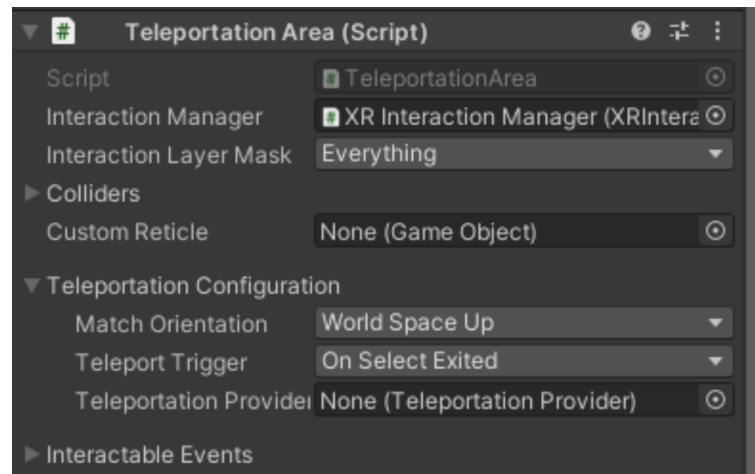
## Aufgabe 2 (Teleportieren)

Wenn eine virtuelle Welt größer ist als der verfügbare Tracking Bereich, muss man per Teleportieren durch die Simulation springen. In Unity XR muss dazu festgelegt werden, auf welche Objekte man sich teleportieren darf. Die Objekte werden mit dem *Layer XR Teleport* gekennzeichnet.

- A) In unserer Supermarkt Szene soll man sich auf dem gesamten Boden teleportieren können. Klicken Sie auf das Objekt *Floor* und wählen Sie das *Layer XR Teleport* aus.



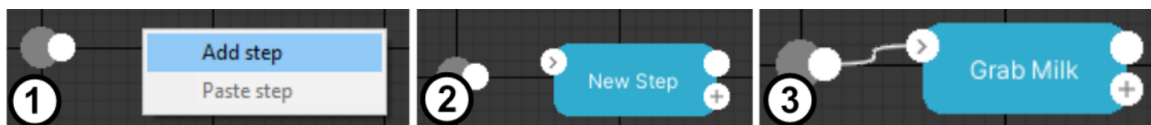
- B) Fügen Sie dem Objekt *Floor* zusätzlich die Komponente *Teleportation Area* hinzu.



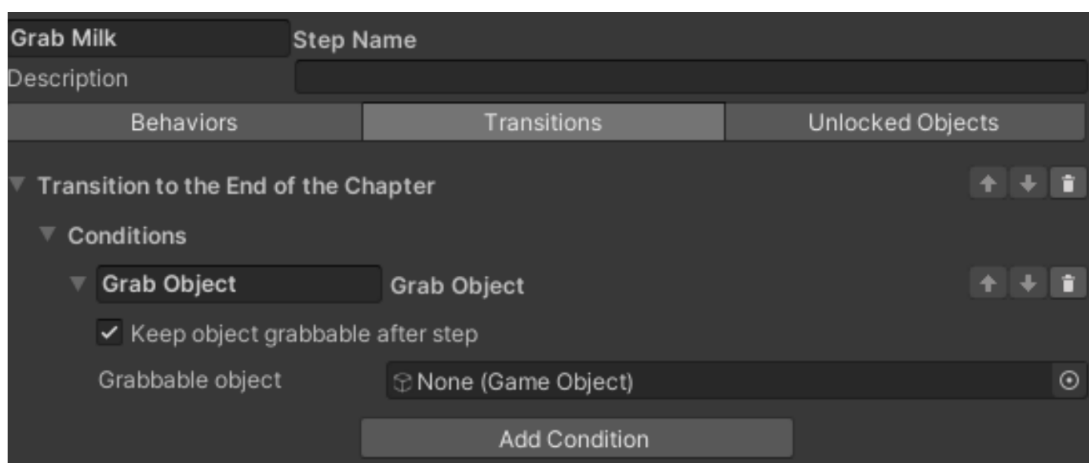
### Aufgabe 3 (Objekte greifen)

Wir machen das erste Objekt mit der Condition *Grab Object* greifbar. Dazu muss das Objekt nur einen Collider besitzen.

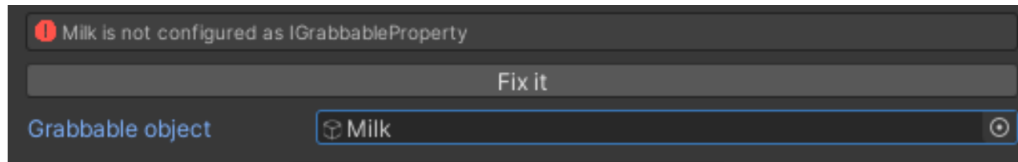
- A) Per Rechtsklick im *Workflow Editor* kann ein neuer Schritt angelegt werden (1). Wird der erstellte Schritt angeklickt (2), erscheint ein zusätzliches Fenster, der *Step Inspector*. Geben Sie dem Schritt zuerst einen passenden Namen z.B. „Grab Milk“ und verbinden Sie den Schritt mit dem Startknoten (3).



- B) Navigieren Sie anschließend im *Step Inspector* zum Reiter *Transitions* und erstellen Sie mit dem Button *Add Condition* eine *Grab Object*-Condition.



- C) Im Einkaufswagen ist nur eine Packung Milch. Die wollen wir greifen und auf der Kasse ablegen. Wählen Sie die Milch in der Hierarchie aus und ziehen Sie diese per Drag & Drop in das Feld *Grabbable Object*.



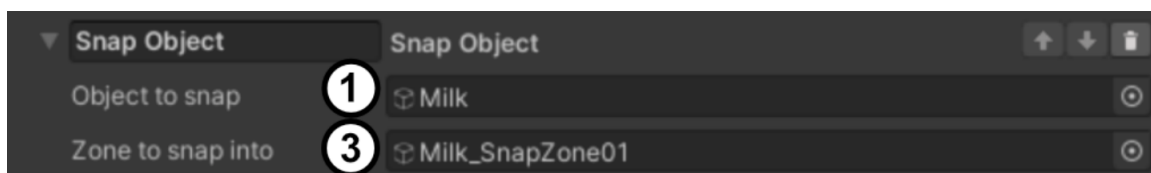
Die Milchtüte ist noch ein 3D-Objekt ohne Logik. Um es mit ihren virtuellen Händen greifen zu können, werden einige Skripte benötigt. Diese werden durch Drücken von *Fix it* hinzugefügt.

D) Testen Sie im Playmode, ob die Milchtüte gegriffen werden kann.

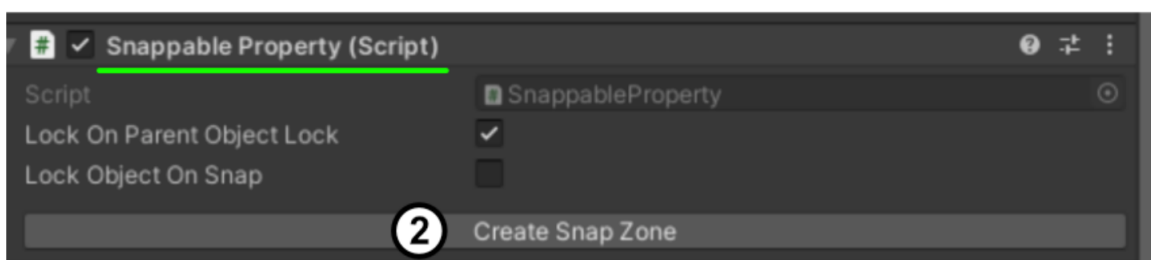
#### **Aufgabe 4** (Objekte ablegen (snappen))

Mit der Condition *Snap Object* kann ein bereits gegriffenes Objekt an einer definierten Zone abgelegt werden. Diese Zone wird Snap Zone genannt.

- A) Erstellen Sie einen neuen Schritt. (*Rechtsklick* + *Add Step*). Fügen Sie in diesem Schritt die Condition *Snap Object* hinzu.
- B) Gesnappt werden soll die Milchtüte. Wir fügen diese ins Feld *Object to Snap* ein ① und erstellen zusätzlich benötigte Skripte mit dem Button *Fix It*.

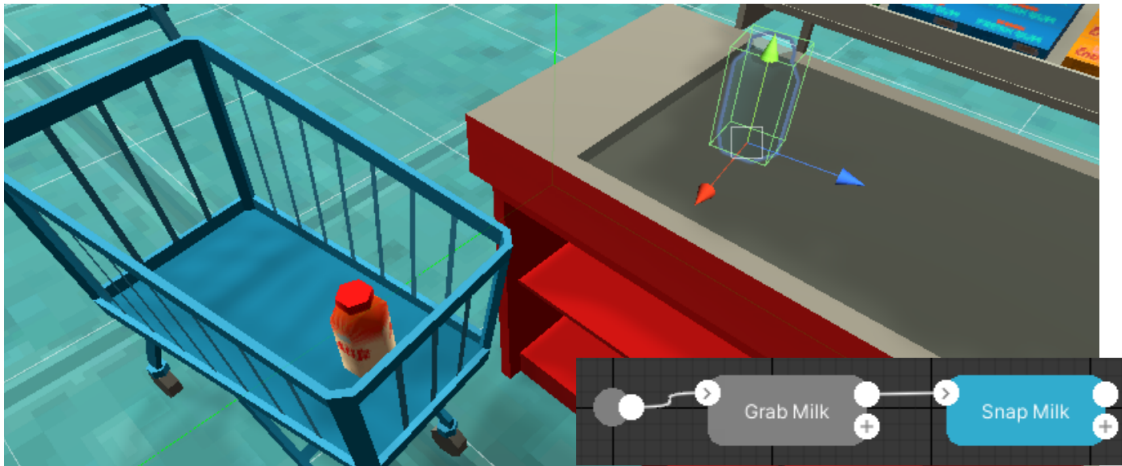


- C) Jetzt wird nur noch eine Zone zum Ablegen benötigt. Jedes Snappable Object kann seine eigene, passende Zone erstellen. Wählen Sie die Milch aus und suchen Sie im Inspector die Komponente *Snappable Property*.



- D) Mit dem Button *Create Snap Zone* wird eine passende Snap Zone erstellt ②. (Die SnapZone heißt „Milk\_SnapZone“ und leuchtet bläulich)
- E) Ziehen Sie die erstellte Snap Zone per Drag & Drop in in das zweite Feld der Condition ③.
- F) Platzieren Sie die Snap Zone am Anfang des Kassenbands.
- G) Testen Sie ihre Szene erneut.

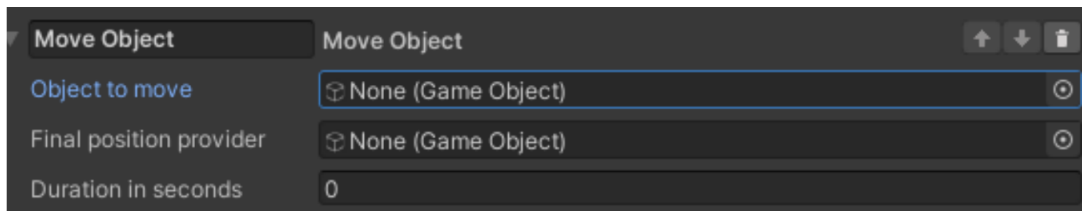




### Aufgabe 5 (Objekte animieren)

Mit dem Behaviour *Move Object* können Objekte bewegt und rotiert werden. Hierbei wird ein frei wählbares Objekt als *Object to move* gesetzt. Dessen Position und Rotation wird auf die Werte eines Referenz-Objekts, dem *Final position provider*, angepasst. Die Zeitspanne für diese Animation kann frei über *Duration in Seconds* gesetzt werden.

- A) Die Milch steht aktuell am Anfang des Kassensbands, wir wollen diese an das Ende der dunkelgrauen Fläche fahren lassen. Erstellen Sie dazu einen dritten Schritt und fügen Sie das Behaviour *Move Object* hinzu.



- B) Versuchen Sie die benötigten Werte passend zu befüllen. Sorgen Sie dafür, dass die Milch in geeigneter Geschwindigkeit zum Ende des Kassensbands fährt.

### Aufgabe 6 (Einen Scanner nachbauen)

Nicht für alle Interaktionen gibt es eine genau passende Condition. Vieles kann allerdings bereits mit den vorhandenen Basiselementen abgedeckt werden.

In dieser Aufgabe simulieren wir das Scannen unserer Milchpackung an der Self-Checkout Kasse. (Der „Scanner“ ist das kleine graue Rechteck mit roter Lichtquelle.)

- A) Bevor wir die Milch scannen, müssen wir diese erneut greifen. Erstellen Sie einen neuen Schritt (*Grab Milk*) und machen Sie die Milch erneut mit der bereits erlernten Condition greifbar.

Der Workflow Editor sollte bei Ihnen in etwa so aussehen:



- B) Erstellen Sie einen weiteren Schritt (*Scan Milk*). Wir simulieren das Scannen mit der Condition *Move Object into Collider*. Für den Erfolg dieser Condition muss ein Objekt für eine bestimmte Zeit in einen definierten Collider gehalten werden.

Damit können wir einen echten Scanner simulieren, bei dem man ja auch ein Produkt für kurze Zeit in einen vorgegebenen Bereich halte muss, um den Strichcode zu scannen.

- C) Erstellen Sie einen Boxcollider über der grauen Fläche, in der der Scanner rot leuchtet. Passen Sie die Größe des Colliders auf ein realistisches Maß an.
- D) Aktivieren Sie für den BoxCollider die Checkbox *IsTrigger*.
- E) Versuchen Sie, die Condition *Move Object into Collider* mit den passenden Objekten zu befüllen.

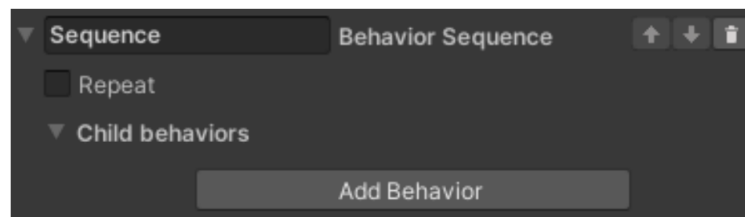
### Aufgabe 7 (Den Scanner animieren)

Noch ist die Szene etwas eintönig. Die Milchtüte wurde zwar gescannt, allerdings gab es kein Feedback des Scanners und der Bildschirm Ihres Wireframes hat das Scannen noch nicht registriert. Wir wollen zuerst den Scanner animieren, indem wir das Licht kurzzeitig von rot auf grün umschalten.

Dazu wird eine *Behaviour Sequence* verwendet.

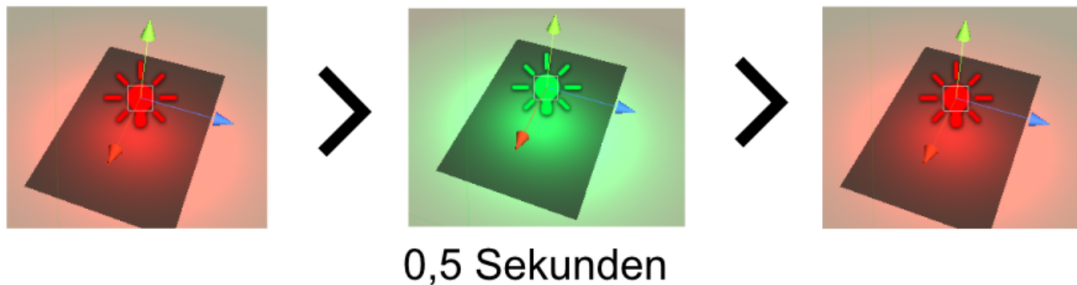
In einer solchen Sequenz werden alle Behaviours der Reihe nach aktiviert. Zusätzlich kann über die Checkbox „*isBlocking*“ bestimmt werden, dass der Schritt nicht abgeschlossen werden darf, bevor die Sequenz komplett durchlaufen ist.

- A) Erstellen Sie einen neuen Schritt (*Scan Animation*) und fügen Sie das Behaviour *Behaviour Sequence* hinzu.



- B) Fügen Sie ein grünes Licht zur Szene hinzu.
- C) Verwenden Sie die Behaviours: *Disable Object*, *Enable Object* und *Delay*, um eine Animation zu basteln
- Das rote Licht deaktivieren (disable object)
  - Das grüne Licht aktivieren (enable object)
  - 0,5 sekunden warten (delay)

- d) Das grüne Licht deaktivieren (disable object)
- e) Das rote Licht erneut aktivieren. (enable object)

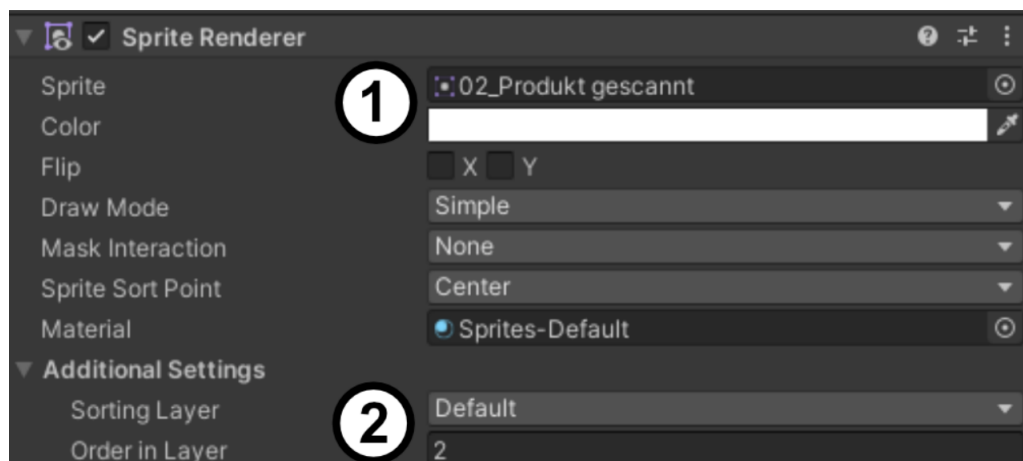


D) Testen Sie alles in der Szene. Der Scanner sollte beim Scannen seine Farbe kurzzeitig ändern.

### Aufgabe 8 (Wireframes in die Szene integrieren)

Beginnen wir, die Screendesigns in unsere Nutzerstudie zu integrieren. Im Ordner „00 VR Nutzerstudie“ → *Images* befinden sich bereits alle 4 benötigten Screens.

- A) Sehen Sie sich den Bildschirm (*Screen*) in der Szene genauer an. Dieser hat bereits den Wireframe „Willkommen“ als Kindobjekt. Die Komponente *Sprite Renderer* macht das Bild in der Szene sichtbar.
- B) Duplizieren sie das Objekt *01\_Willkommen*, ändern Sie den Namen auf *02\_Produkt gescannt* und tauschen Sie den Screen passend aus ①.

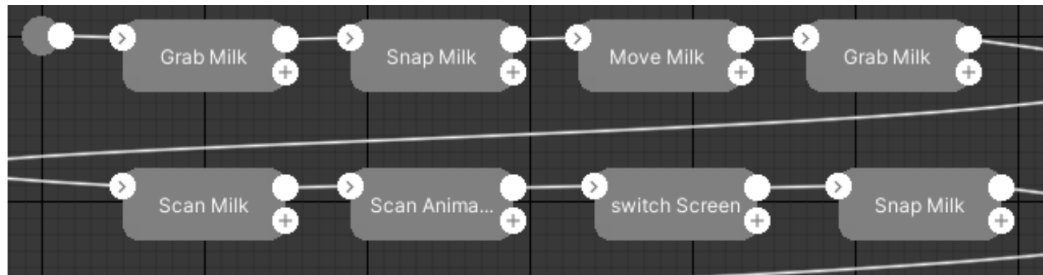


- C) Falls ihr zweiter Screen vom ersten Screen verdeckt wird, müssen Sie das Sorting Layer des Sprite Renderers richtig einstellen. Das Sorting Layer definiert, welches Bild priorisiert gerendert wird ②. Höhere Nummern = höhere Priorität.
- D) Erstellen Sie einen weiteren Schritt und aktivieren Sie darin per Behaviour den zweiten Screen (*enable Object*).

*Tipp:* Es macht Sinn, im allerersten Schritt des Kurses alles per Behaviours zu deaktivieren, was Sie erst ab einem definierten Punkt im Kurs aktivieren wollen. Alternativ können Sie das

Objekt auch einfach in der Hierarchie deaktivieren.

- E) Wir tragen die Milch schon sehr lange mit uns herum, legen Sie diese rechts vom Bildschirm in einem weiteren Schritt per *Snap Object* ab.



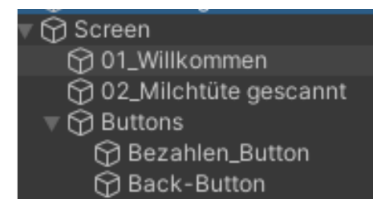
### Aufgabe 9 (Interaktion mit dem Bildschirm)

Wir wollen den Bildschirm als Touchscreen verwenden, um den Einkauf per Bezahlung abschließen zu können. Wieder bedienen wir uns an einer Standardfunktion.

Wir verwenden die Condition *Touch Object*.

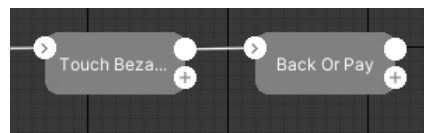
- A) Erstellen Sie einen neuen Schritt (*Touch Bezahlen*) und fügen Sie die *Touch Object*-Condition hinzu.

- B) Im Objekt Screen befindet sich bereits das Kind-Objekt *Buttons*. Wählen Sie den richtigen Button aus und fügen diesen zur Condition *Touch Object* dazu.



- C) Damit wir den Bezahlvorgang durchführen können, muss der dritte Screen in einem weiteren Schritt sichtbar werden (*enable Object*).

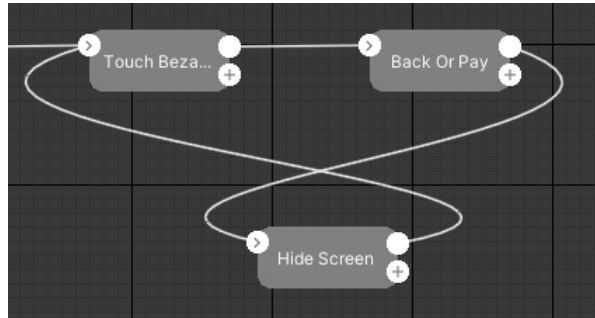
Erstellen Sie dafür den Screen und den zusätzlichen Step „*Back Or Pay*“, der sowohl für das Sichtbarsetzen als auch das weitere Handling des Screens zuständig ist.



- D) *Loops*: Im Screen „Bezahlen“ gibt es ebenfalls einen Zurück-Button.

Mithilfe eines *Loops* im *Workflow Editor* wollen wir es dem Anwender ermöglichen, im Kurs zurückspringen, sofern der Zurück-Button gedrückt wird.

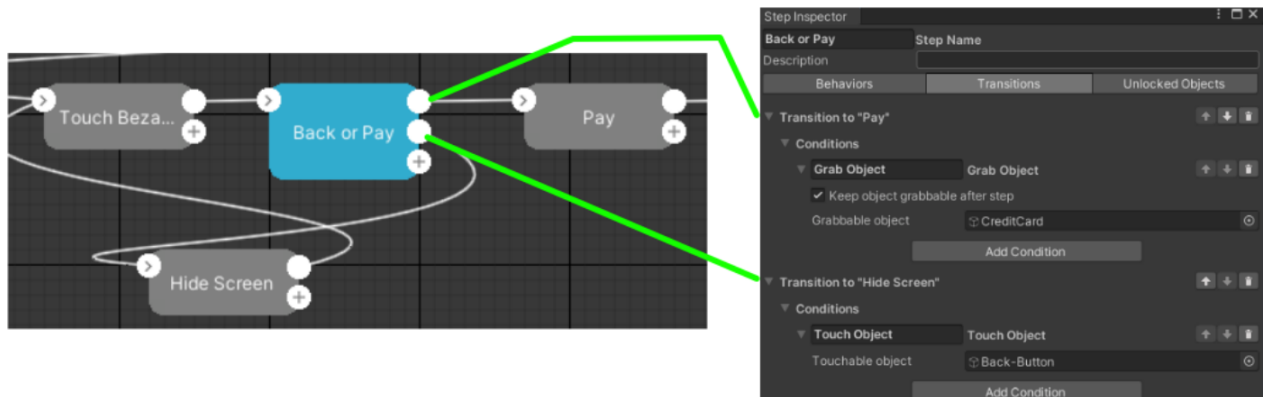
- E) Damit der Loop richtig funktioniert, müssen die Screens beim Zurückspringen erneut aus- und eingeblendet werden. Probieren Sie die richtige Verkettung aus.



### Aufgabe 10 (Bezahlen per Kreditkarte)

Wir wollen den Loop beenden und mithilfe einer selbst erstellten Kreditkarte den Bezahlvorgang abschließen.

- A) Damit der Loop beendet werden kann, muss eine weitere Transition im Step „Back Or Pay“ erstellt werden. (*Step inspector* → *Add Transition*)



- B) Der Loop wird beendet, sobald der Anwender eine Kreditkarte greift.
- C) Die Kreditkarte gibt es noch nicht. . . Erstellen Sie Ihre eigene, authentische Kreditkarte mit *Ihrem Name & Ihrer MatrikelNr.*

Wenden Sie Ihr bisheriges Wissen an, um ein Bild auf ein Objekt zu legen. Machen Sie die Kreditkarte anschließend greifbar, damit der Loop erfolgreich beendet werden kann.



- D) Den Bezahlvorgang modellieren wir genauso wie den Scan-Vorgang.

Bauen Sie einen Dummy Kartenscanner mithilfe des im Bild sichtbaren Kartenlesers (Condition: *Move Object into Collider*)

- E) Sobald der Bezahlvorgang abgeschlossen ist, soll der letzte Screen „*Danke für Ihren Einkauf*“ angezeigt werden.
- F) Wir haben unseren Einkauf bezahlt, es wäre natürlich angebracht, wenn sich die Schranke jetzt öffnen würde und wir den Supermarkt verlassen könnten.



### **Aufgabe 11** (Testat: Erweitern Sie ihre virtuelle Nutzerstudie um zusätzliche Funktionen)

Ein paar Inspirationen:

- A) Das Kartenterminal um ein kleines Display erweitern.
- B) Den Supermarkt vervollständigen: Regale, Wände, Artikel, Menschen, ... hinzufügen.
- C) Die Wireframes durch eigene Screendesigns austauschen.
- D) Zusätzliche Interaktionen im Workflow Editor ergänzen:
  - a) Die Milchtüte aus einem Regal greifen.
  - b) Einen zusätzlichen Artikel kaufen. (Komplex! Sie müssen dann ebenfalls die Screens anpassen.)
- E) Die Kreditkarte durch ein Smartphone ersetzen + Screendesign für das Smartphone.
- F) Den Einkauf in einer Einkaufstüte verpacken oder den Einkauf von einem Roboter verpacken lassen.
- G) Einen Kassenbon ausdrucken. Sie können den Supermarkt nur mit Kassenbon verlassen