



Dokumentation

XR Selection Package

Eddie Bo Spencer
inf105826@stud.fh-wedel.de

Jonathan Tietz
inf105821@stud.fh-wedel.de

2. März 2025

Inhaltsverzeichnis

1	Einleitung	1
1.1	Ziel der Library	1
1.2	Voraussetzungen	1
1.3	Abhängigkeiten	2
2	Installation	2
3	Überblick	3
4	Grundlegende Nutzung	4
4.1	Erste Schritte	4
4.2	Selectors	5
4.2.1	SimpleSelector	5
4.2.2	ScopedSelector	6
4.2.3	GazeSelector	7
4.3	Selectables	8
4.3.1	TintSelectable	8
4.3.2	TeleportSelectable	9
4.3.3	EventSelectable	10
4.4	Methods	11
4.4.1	Methode 1	11
4.4.2	Methode 2	13
4.4.3	Methode 3	15
5	Architektur	16
5.1	Selectable UML-Diagramm	16
5.2	Selector UML-Diagramm	17
5.3	Interaktion der Komponenten	19
6	Samples	20
6.1	Installation	20
6.2	Bekannte Probleme	20

1 Einleitung

In modernen XR-Anwendungen ist die Interaktion mit virtuellen Objekten ein essenzieller Bestandteil der Nutzererfahrung. Besonders die Selektion von Objekten mittels Gaze-Tracking und Handtracking ermöglicht eine intuitive und effiziente Steuerung in immersiven Umgebungen. Dieses Package bietet eine flexible und modular aufgebaute Lösung, um verschiedene Selektionsmethoden leichtgewichtig in Unity-Anwendungen zu integrieren.

1.1 Ziel der Library

Das *XR Selection Package* bietet eine generische und erweiterbare Architektur zur Auswahl von Objekten in XR-Umgebungen. Durch die Nutzung von OpenXR wird eine plattformübergreifende Kompatibilität sichergestellt, sodass die Library mit einer Vielzahl von XR-Systemen eingesetzt werden kann. Gleichzeitig bleibt die Lösung unabhängig von spezifischen Unity-XR-Paketen, wodurch eine flexible Integration in verschiedene Projekte ermöglicht wird. Entwickler erhalten somit eine modulare Grundlage, um Interaktionsmechanismen an individuelle Anforderungen anzupassen und zu erweitern.

1.2 Voraussetzungen

Es müssen folgende Voraussetzungen für die Nutzung des *XR Selection Package* gegeben sein:

- Unity-Version (Unity 6 oder neuer)

1.3 Abhängigkeiten

Es existieren folgenden Abhängigkeiten:

- Input System (com.unity.inputsystem)
- OpenXR (com.unity.xr.openxr)
- XR Hands (com.unity.xr.hands)

2 Installation

Das *XR Selection Package* kann über das Unity Package Manager (UPM) mittels einer Git-URL installiert werden. Die folgenden Schritte beschreiben die Installation des Pakets in ein Unity-Projekt:

1. Öffne Unity und navigiere im Menü zu **Window** → **Package Manager**.
2. Klicke auf das **+**-Symbol oben links im Package Manager-Fenster und wähle **Add package from git URL....**
3. Gib die folgende Git-URL ein und bestätige die Eingabe:

```
1 https://github.com/FlamieLP/MasterProjekt.git
```

4. Unity lädt das Paket und fügt es dem Projekt hinzu.
5. Nach erfolgreicher Installation erscheint das Paket in der Liste der installierten Pakete im **Package Manager**.

Weitere Informationen zur Installation von Paketen über eine Git-URL finden sich in der offiziellen Unity-Dokumentation: Unity Package Manager – Installation via Git URL.

3 Überblick

Das *XR Selection Package* stellt drei zentrale Komponenten bereit, die die Selektion von Objekten in einer XR-Umgebung ermöglichen:

- **Selector** – Eine Komponente innerhalb der Unity-Szene, die die Selektion von *Selectables* initiiert. Sie wird in der Regel vom Nutzer gesteuert, beispielsweise durch Gaze- oder Hand-Tracking.
- **Selectable** – Eine Komponente, die Objekte innerhalb der Unity-Szene beschreibt, mit denen *Selectors* interagieren können.
- **Method** – Eine Kombination aus *Selectors*, die eine spezifische Selektionsmethode sowie das Verhalten der Selektion definiert. Sie bestimmt maßgeblich die Interaktionsweise für den Endnutzer.

Die Basisfunktionalitäten des *XR Selection Packages* bilden ein leichtgewichtiges Grundgerüst für die Integration von XR-kompatibler Selektion in Unity-Projekten. Für spezifische Anwendungsfälle ist eine Erweiterung dieser Basisfunktionalitäten notwendig. Kapitel 5 zeigt die Struktur nach der neue Komponenten in das *XR Selection Package* integriert werden können und dient als Referenz für potentielle Erweiterungen.

4 Grundlegende Nutzung

Dieser Abschnitt erläutert die Nutzung des *XR Selection Packages* und den Aufbau der einzelnen Komponenten.

4.1 Erste Schritte

Die grundlegende Nutzung des *XR Selection Package* erfolgt in mehreren Schritten:

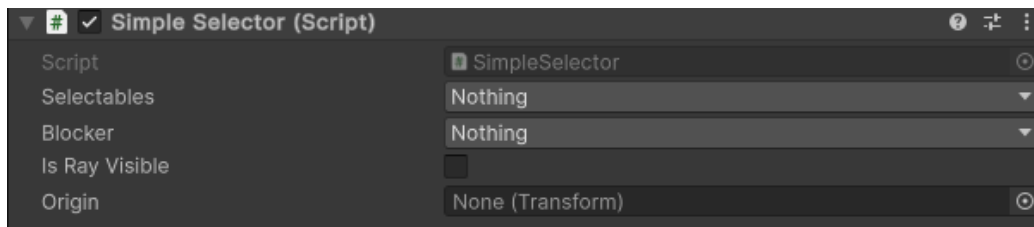
1. **Selector-Skript hinzufügen** Füge einem Objekt ein beliebiges *Selector*-Skript hinzu, beispielsweise *SimpleSelector*.
2. **Methoden-Skript hinzufügen** Füge einem Objekt ein *Method*-Skript hinzu, beispielsweise *Method3*.
3. **Verknüpfung von Method- und Selector-Skript** Übergib dem *Method*-Skript eine Referenz auf das *Selector*-Skript. Einige Methoden benötigen mehrere *Selectors*, um korrekt zu funktionieren.
4. **Eingabequellen für das Method-Skript festlegen** Übergib dem *Method*-Skript die gewünschten *Inputs*, über die die Selektion erfolgen soll. Einige Methoden erfordern mehrere *Inputs* für eine korrekte Funktion.
5. **Selectable-Skript hinzufügen** Füge einem Objekt, das selektierbar sein soll, ein *Selectable*-Skript hinzu. Es empfiehlt sich, einen eigenen Layer für selektierbare Objekte zu verwenden.
6. **Layer-Einstellungen überprüfen** Stelle sicher, dass alle *Selector*-Skripte, die das Objekt selektieren sollen, den festgelegten Layer als *selectables* ausgewählt haben.

4.2 Selectors

Im folgenden werden die verschiedenen Selectors vorgestellt und beschrieben.

4.2.1 SimpleSelector

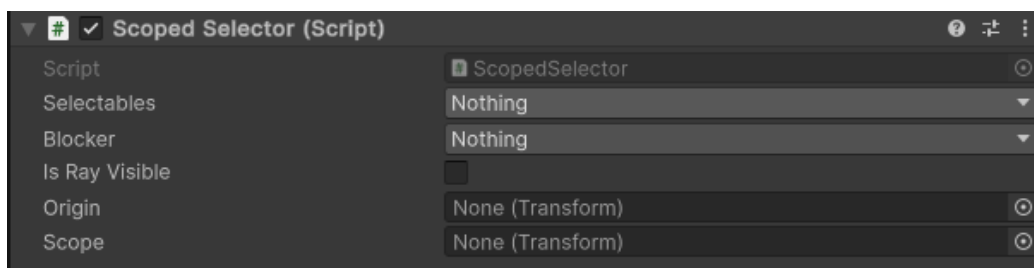
Der SimpleSelector definiert einen Selector, dem ein Transform zugewiesen wird, das als Basis für das Selektieren verwendet wird. Das Transform bestimmt sowohl Startpunkt als auch die Richtung in die selektiert wird.



Eigenschaften	Beschreibung
Selectables	LayerMask die definiert, welche Layer für die Selektion berücksichtigt werden.
Blocker	LayerMask die definiert, welche Layer, Objekte beinhalten, die eine Selektion blockieren können.
Is Ray Visible	Ermöglicht das Anzeigen eines Strahls zur Visualisierung der Selectors. Diese Visualisierung wird ausschließlich im Editor angezeigt.
Origin	Definiert Ausgangspunkt und Richtung der Selektion. Wird kein Transform zugewiesen, wird das Transform des GameObjects verwendet, an dem sich das Skript befindet.

4.2.2 ScopedSelector

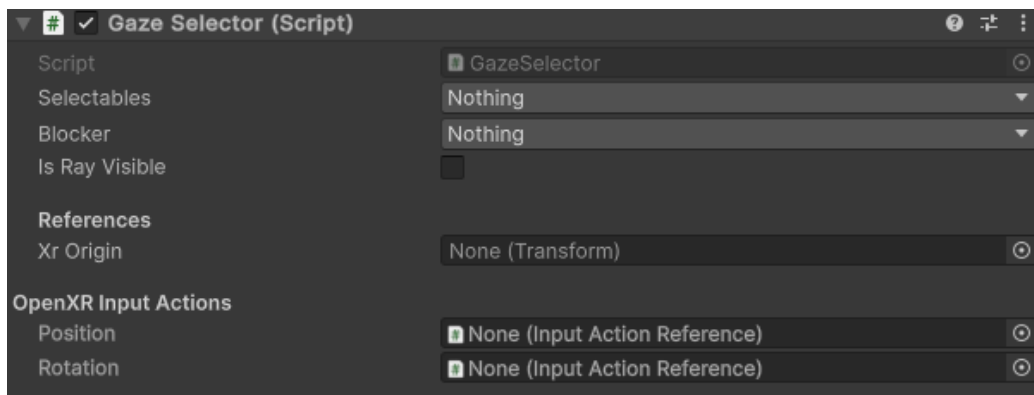
Der `ScopedSelector` verwendet einen Start- und einen Zielpunkt, um die Selektion durchzuführen. Diese beiden Punkte bestimmen den Ausgangspunkt sowie die Richtung der Selektion.



Eigenschaften	Beschreibung
Selectables	LayerMask die definiert, welche Layer für die Selektion berücksichtigt werden.
Blocker	LayerMask die definiert, welche Layer, Objekte beinhalten, die eine Selektion blockieren können.
Is Ray Visible	Ermöglicht das Anzeigen eines Strahls zur Visualisierung der Selectors. Diese Visualisierung wird ausschließlich im Editor angezeigt.
Origin	Definiert Ausgangspunkt der Selektion.
Target	Definiert Richtung der Selektion, ausgehend vom Origin.

4.2.3 GazeSelector

Der GazeSelector nutzt die Position der Augen sowie die Blickrichtung als Referenzen zur Selektion. Für eine korrekte Funktionsweise müssen zuvor InputActions | definiert werden.



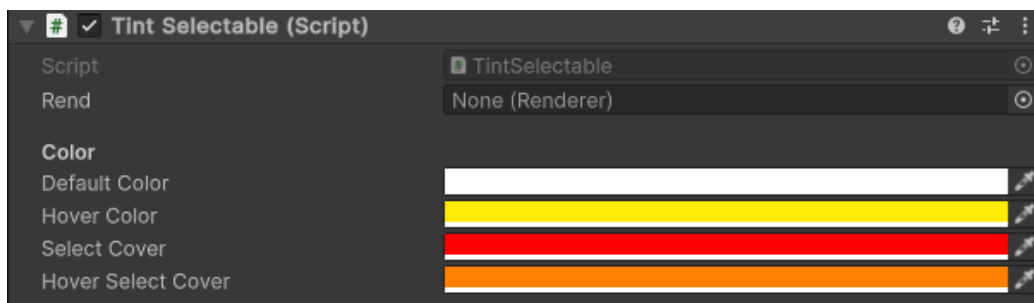
Eigenschaften	Beschreibung
Selectables	LayerMask die definiert, welche Layer für die Selektion berücksichtigt werden.
Blocker	LayerMask die definiert, welche Layer, Objekte beinhalten, die eine Selektion blockieren können.
Is Ray Visible	Ermöglicht das Anzeigen eines Strahls zur Visualisierung der Selectors. Diese Visualisierung wird ausschließlich im Editor angezeigt.
Xr Origin	Definiert Referenzpunkt für Position der Augen.
Position	Referenz auf die InputAction, die die Position der Augen bereitstellt.
Rotation	Referenz auf die InputAction, die die Blickrichtung bereitstellt.

4.3 Selectables

Im folgenden werden die verschiedenen Selectables vorgestellt und beschrieben.

4.3.1 TintSelectable

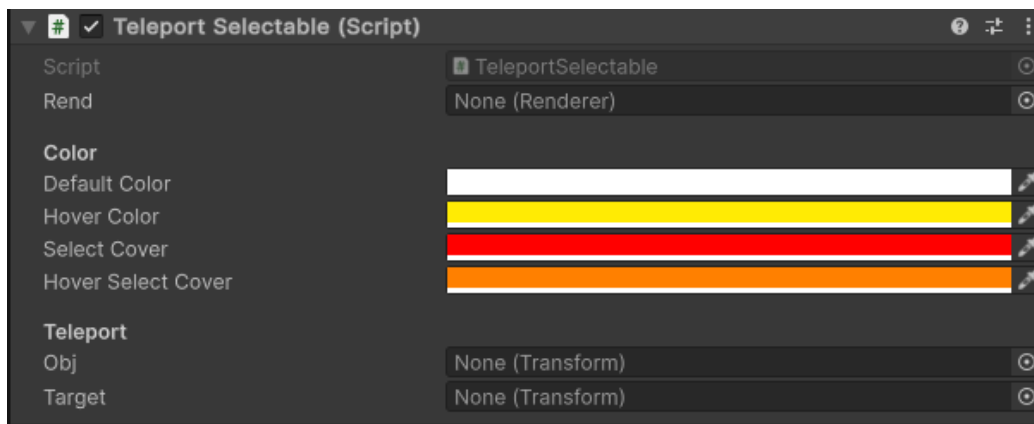
Das TintSelectable ermöglicht das einfärben von Objekt, abhängig davon ob sie gehovert oder selektiert werden.



Eigenschaften	Beschreibung
Rend	Renderer des Objektes dessen Farbe beeinflusst werden soll. Wird kein Renderer zugewiesen, wird der Renderer des GameObjects verwendet, an dem sich das Skript befindet.
Default Color	Farbe des Objektes wenn es weder gehovert noch selektiert wird.
Hover Color	Farbe des Objektes wenn es weder gehovert wird.
Select Color	Farbe des Objektes wenn es weder selektiert wird.
Hover Select Color	Farbe des Objektes wenn es gehovert und selektiert wird.

4.3.2 TeleportSelectable

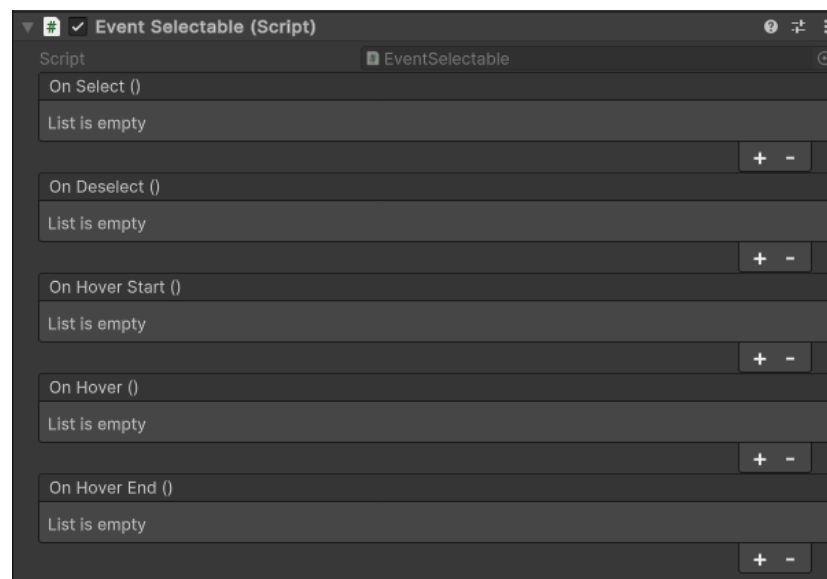
Das TeleportSelectable erlaubt bei Selektion, das Teleportieren von Objekten zu einer ausgewählten Position. Es stellt ebenfalls alle Funktion des TintSelectables bereit.



Eigenschaften	Beschreibung
Rend	Renderer des Objektes dessen Farbe beeinflusst werden soll. Wird kein Renderer zugewiesen, wird der Renderer des GameObjects verwendet, an dem sich das Skript befindet.
Default Color	Farbe des Objektes wenn es weder gehovort noch selektiert wird.
Hover Color	Farbe des Objektes wenn es weder gehovort wird.
Select Color	Farbe des Objektes wenn es weder selektiert wird.
Hover Select Color	Farbe des Objektes wenn es gehovort und selektiert wird.
Object	Objekt, das teleportiert werden soll.
Target	Zielposition des Objektes.

4.3.3 EventSelectable

Das EventSelectable erlaubt es Event-gesteuert auf Selektionen zu reagieren, als alternativer Low-Code Ansatz.



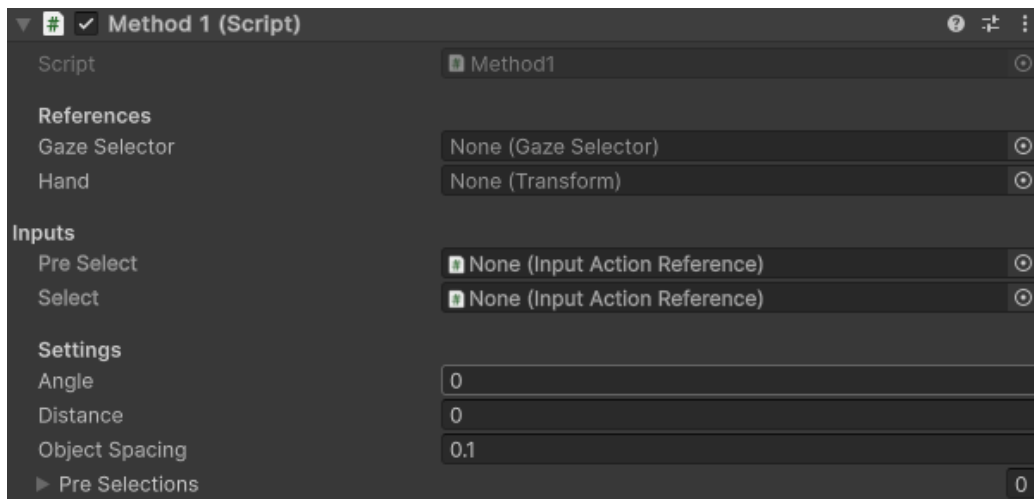
Eigenschaften	Beschreibung
On Select	Event das bei Selektion des Objektes ausgelöst wird.
On Deselect	Event das beim Deselektieren des Objektes ausgelöst wird.
On Hover Start	Event das ausgelöst, wenn ein Objekt gehovert wird.
On Hover	Event das in jedem Frame geworfen ist, solange ein Objekt gehovert wird.
On Hover End	Event das ausgelöst, wenn ein Objekt nicht mehr gehovert wird.

4.4 Methods

Im folgenden werden die verschiedenen Methods vorgestellt und beschrieben.

4.4.1 Methode 1

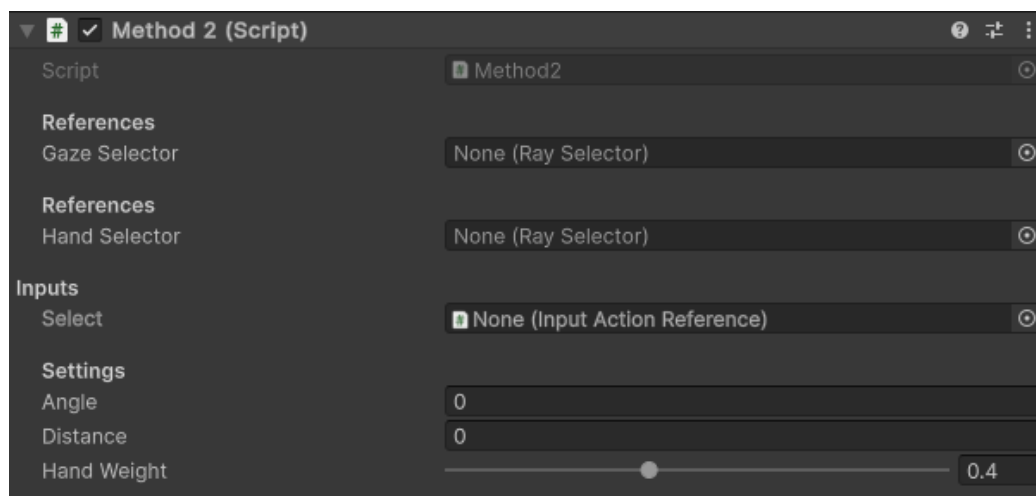
Diese Methode ermöglicht es dem Nutzer, eine Gruppe von Objekten zunächst mittels Gaze-Tracking vorzuselektieren. Anschließend kann durch Handbewegungen eine präzisere Selektion innerhalb dieser Gruppe durchgeführt werden.



Eigenschaften	Beschreibung
Gaze Selector	Referenz auf den GazeSelector, der für die Vorselektion verantwortlich ist.
Hand	Transform der Hand, die zur präzisieren Selektion innerhalb der vorselektierten Objekte genutzt wird.
Pre Select	Referenz auf die InputAction zur Aktivierung der Gaze-basierten Vorselektion.
Select	Referenz auf die InputAction zur Bestätigung der finalen Selektion mit der Hand.
Angle	Winkelbereich, in dem die Vorselektion vorgenommen wird.
Distance	Maximale Distanz der selektierbaren Objekte ausgehend von der Position der Augen.
Object Spacing	Bewegungsdistanz der Hand, die nötig ist, um innerhalb der vorselektierten Objekte eine präzise Selektion vorzunehmen.

4.4.2 Methode 2

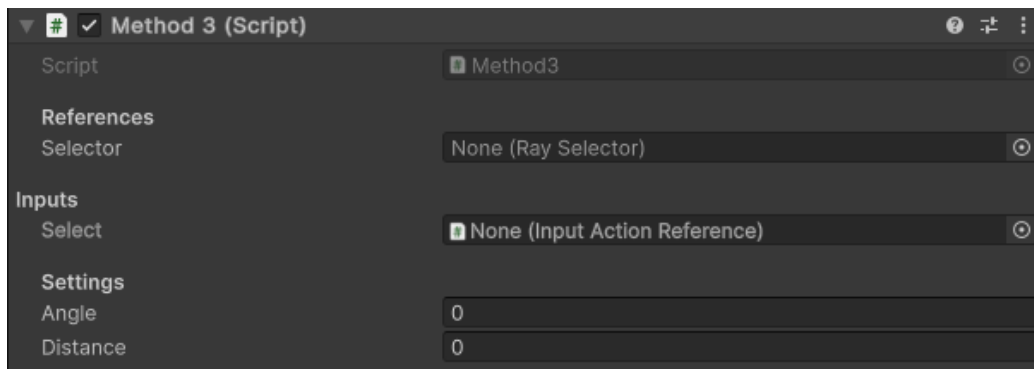
Diese Methode erlaubt es dem Nutzer, Objekte durch eine Kombination aus Gaze- und Hand-Tracking zu selektieren. Ein einstellbarer Parameter ermöglicht die Anpassung der Gewichtung beider Selektoren bei der Auswahl von Objekten.



Eigenschaften	Beschreibung
Gaze Selector	Referenz auf den Selector basierend auf Gaze-Tracking.
Hand Selector	Referenz auf den Selector basierend auf Hand-Tracking.
Select	Referenz auf die InputAction zur Bestätigung der Selektion.
Angle	Winkelbereich, in dem die Selektion vorgenommen wird.
Distance	Maximale Distanz der selektierbaren Objekte ausgehend von der Position des jeweiligen Selectors.
Hand Weight	Bestimmt die Gewichtung der Selektion im Wertebereich $[0, 1]$. Ein Wert von 0 berücksichtigt ausschließlich den Gaze Selector und 1 ausschließlich den Hand Selector

4.4.3 Methode 3

Diese Methode erlaubt es dem Nutzer, Objekte durch einen einzelnen Selector zu selektieren.

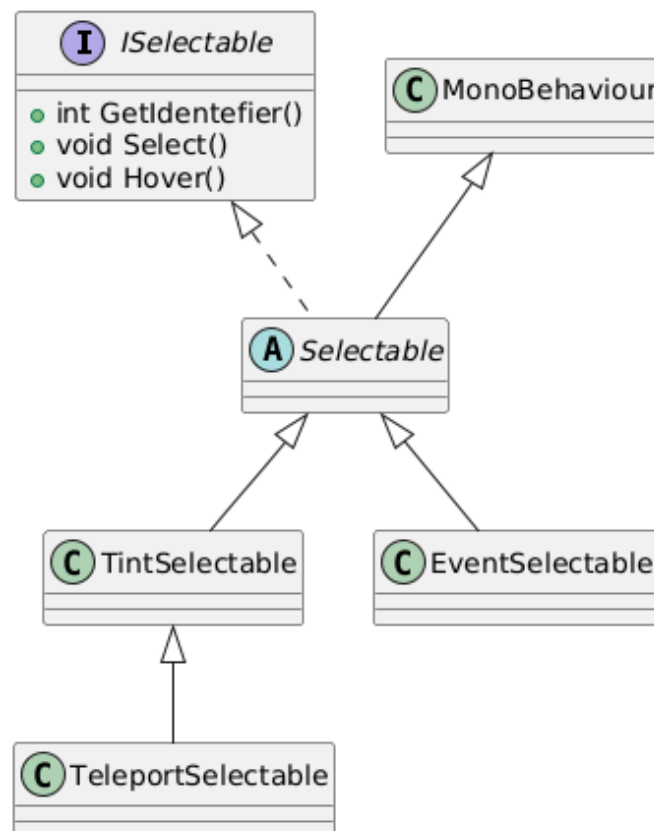


Eigenschaften	Beschreibung
Selector	Referenz auf den Selector, der zur Selektion verwendet wird.
Select	Referenz auf die InputAction zur Bestätigung der Selektion.
Angle	Winkelbereich, in dem die Selektion vorgenommen wird.
Distance	Maximale Distanz der selektierbaren Objekte ausgehend von der Position des Selectors.

5 Architektur

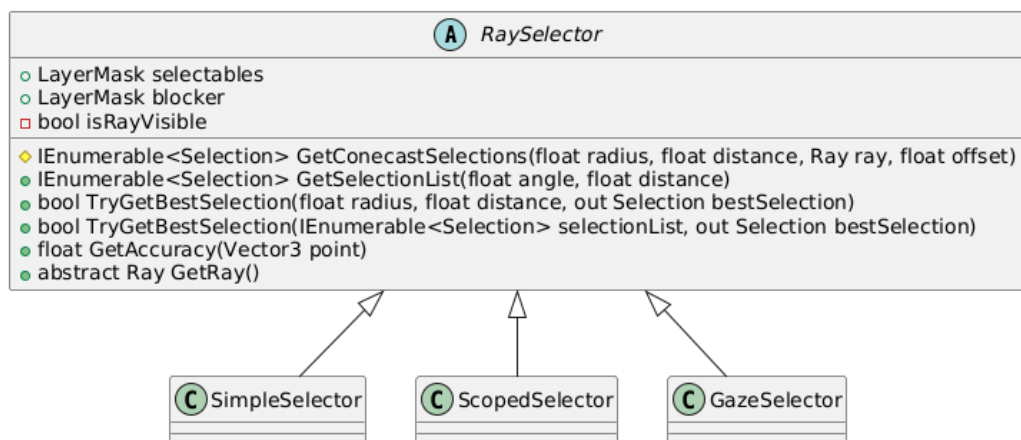
Dieser Abschnitt gewährt einen Überblick über die gewählte Architektur des *XR Selection Packages*. Hierbei wird vorgestellt wie bestehende Selector- und Selectable-Klassen ihre Basisklassen erben und welche wesentlichen Interfaces sie implementieren. Abschließend verdeutlicht ein Sequenzdiagramm die Interaktion der einzelnen Komponenten.

5.1 Selectable UML-Diagramm



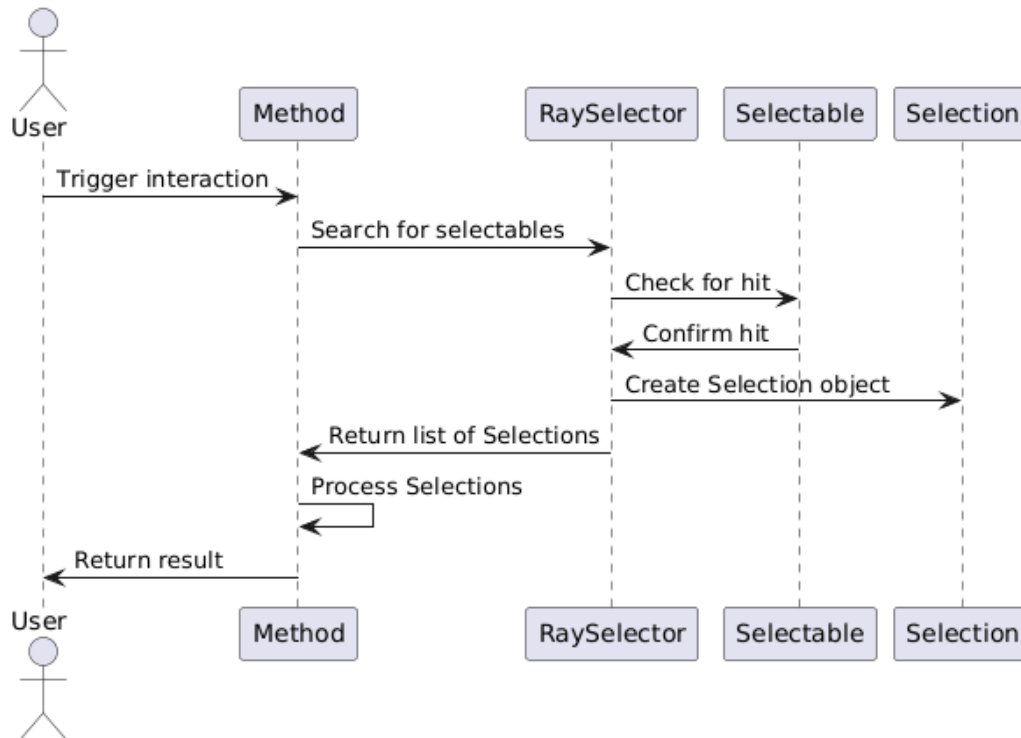
Methode	Beschreibung
GetIdentefier()	Gibt eine eindeutige Identifikationsnummer für das Objekt zurück.
Select()	Wird aufgerufen, wenn das Objekt selektiert wird.
Hover()	Wird aktiviert, solange der Nutzer mit einem Selektor über das Objekt fährt.

5.2 Selector UML-Diagramm



Methode	Beschreibung
GetConecastSelections	Führt einen Conecast durch und gibt eine Liste der selektierten Objekte zurück.
GetSelectionList	Ermittelt eine Liste von Selektionen basierend auf einem bestimmten Winkel und einer bestimmten Distanz.
TryGetBestSelection	Versucht, die beste Auswahl basierend auf den gegebenen Parametern zu bestimmen.
TryGetBestSelection	Bestimmt die beste Auswahl aus einer gegebenen Liste von Selektionen.
GetAccuracy	Berechnet die Genauigkeit der Selektion für einen bestimmten Punkt.
GetRay	Gibt den Startpunkt und die Richtung der Selektion zurück.

5.3 Interaktion der Komponenten



Das Sequenzdiagramm verdeutlicht die Interaktion der einzelnen Komponenten und visualisiert den Ablauf einer Selektion mittels des *XR Selection Packages*. Method bildet hierbei die Schnittstelle mit dem Endnutzer und ermöglicht das Selektionsverhalten zu steuern. Für eine flexible Integration des *XR Selection Packages* existiert keine feste Spezifikation für Method. Die technische Schnittstelle ist der Selector, der eine umfassende Funktionalität bereitstellt, um eine breite Masse an Anwendungsfällen abzudecken.

6 Samples

Das *XR Selection Package* stellt eine Beispielumgebung bereit, die den Einstieg erleichtert und einen Überblick über die verfügbaren Methoden bietet.

6.1 Installation

Das **XR Selection Package** enthält eine Reihe von Beispielumgebungen, die eine Testumgebung für die unterschiedlichen Selektionsmethoden bereitstellt. Diese Beispiele können direkt über den Package Manager installiert werden:

1. Öffne das **Package Manager** Fenster (Menü: Window > Package Manager).
2. Wähle das **XR Selection Package** aus der Liste der Pakete im Projekt aus. (Falls das Paket noch nicht zum Projekt hinzugefügt wurde, muss dies vorher erfolgen.)
3. Wechsle zum Tab **Samples**.
4. Klicke auf den **Import**-Button neben dem Beispiel, das du importieren möchtest.

Die Beispiel-Dateien werden standardmäßig im folgenden Ordner abgelegt: `Assets\Samples\XR Selection Package\[Version]\`. Diese Assets können anschließend an einen anderen Speicherort verschoben werden.

6.2 Bekannte Probleme

- In den **XR Scenes** Beispielszenen werden die vordefinierten `InputActions` initial nicht erkannt. Dadurch werden Eingaben nicht ausgelöst und müssen einmal manuell neu zugewiesen werden.