SAPIENZA
UNIVERSITÀ DI ROMA

Project R.3:

Motion Retargeting Methods Comparison

Medical Robotics Course 2024/2025

Group Members:

Alemanno Giulia 1953800
Coletta Emanuele 2001600
Lazzaroni Lisa 1983551
Negrenti Alessandra 2006031

Professors:

Vendittelli Marilena
De Santis Emanuele

Medical Robotics and Virtual Simulators

Main Goals:

- Improving procedural precision
- Reducing risks for patients
- Training for doctors
- Rehabilitation

Applications:

- Training for blood procedures
- Evaluation and training of suturing
- Rehabilitation of post-stroke patients
- Upper limb rehabilitation

Main Objective

Realistic simulation of <u>grasping</u> through motion retargeting using two methods:

Haptic Glove (WeArt)

WeArt TouchDIVER G1

Software:

- *Unity*
- *WeArt SDK*

Virtual Reality (Meta)

Meta Quest 3

Software:

- *Unity*
- *Meta Interaction SDK*

WeArt Workflow

1 — Computing direct and inverse kinematics

In order to compute an accurate path trajectory each finger is modeled as a 3R planar manipulator

2 — Path Configuration

Two strategies were taken into consideration to compute the most accurate trajectory true to the natural hand movement

3 — Grasping System

Finally a new grasping system was introduced utilizing the direct and inverse kinematics model of the hand

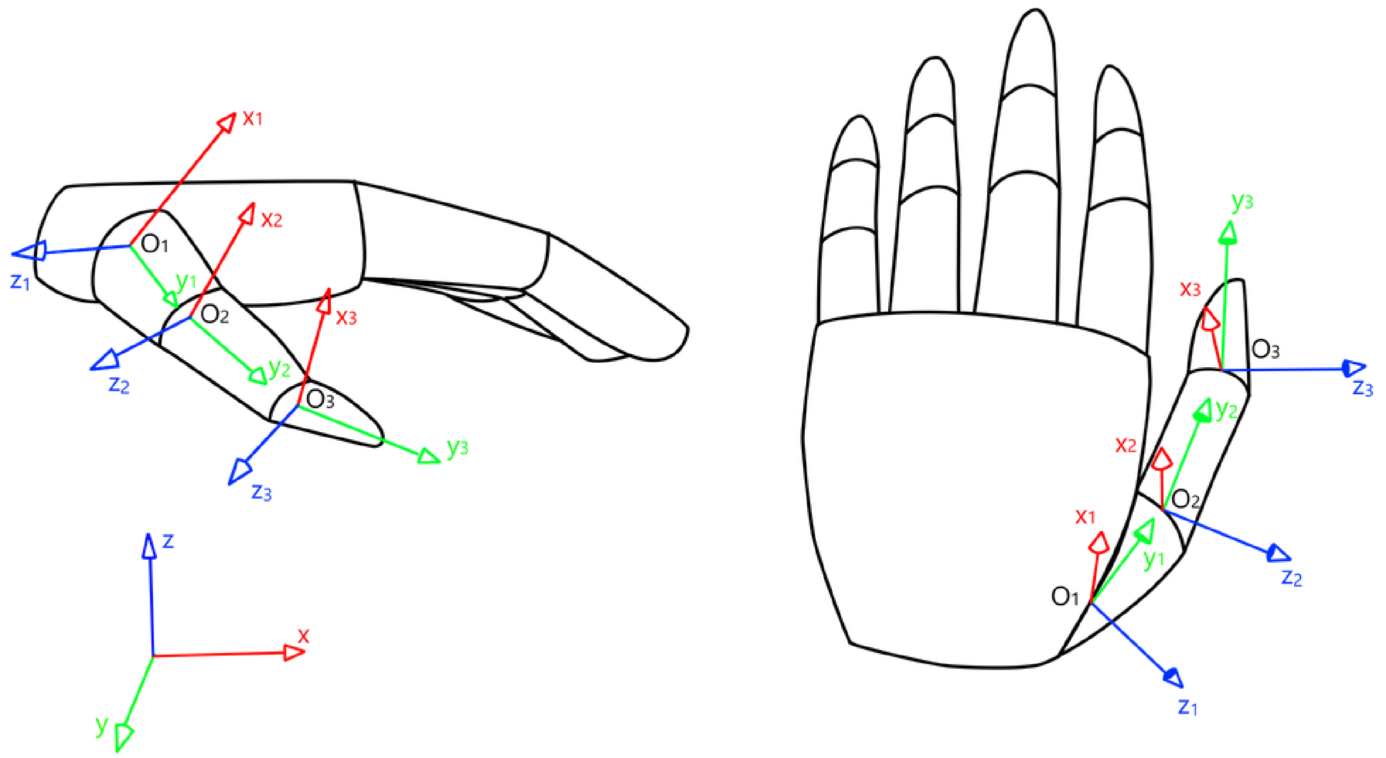
4 — Integration with VR

Direct and Inverse Kinematics

Each finger is modeled as a 3R planar manipulator, moving in the XY-plane.

Thus each finger's direct kinematics is computed through:

$$\begin{cases} x = l_1 \cos q_1 + l_2 \cos(q_1 + q_2) + l_3 \cos(q_1 + q_2 + q_3) \\ y = l_1 \sin q_1 + l_2 \sin(q_1 + q_2) + l_3 \sin(q_1 + q_2 + q_3) \\ \alpha = q_1 + q_2 + q_3 \end{cases}$$



The controller features a term proportional to the error in cartesian space and a Jacobian null-space term to keep the joints within the [0, π/2] limit.

$$\dot{q} = J^{-1}\left[K_p(p_d - f(q))\right] + (I - J^{\#}J)\dot{q}_o$$

Path Configuration

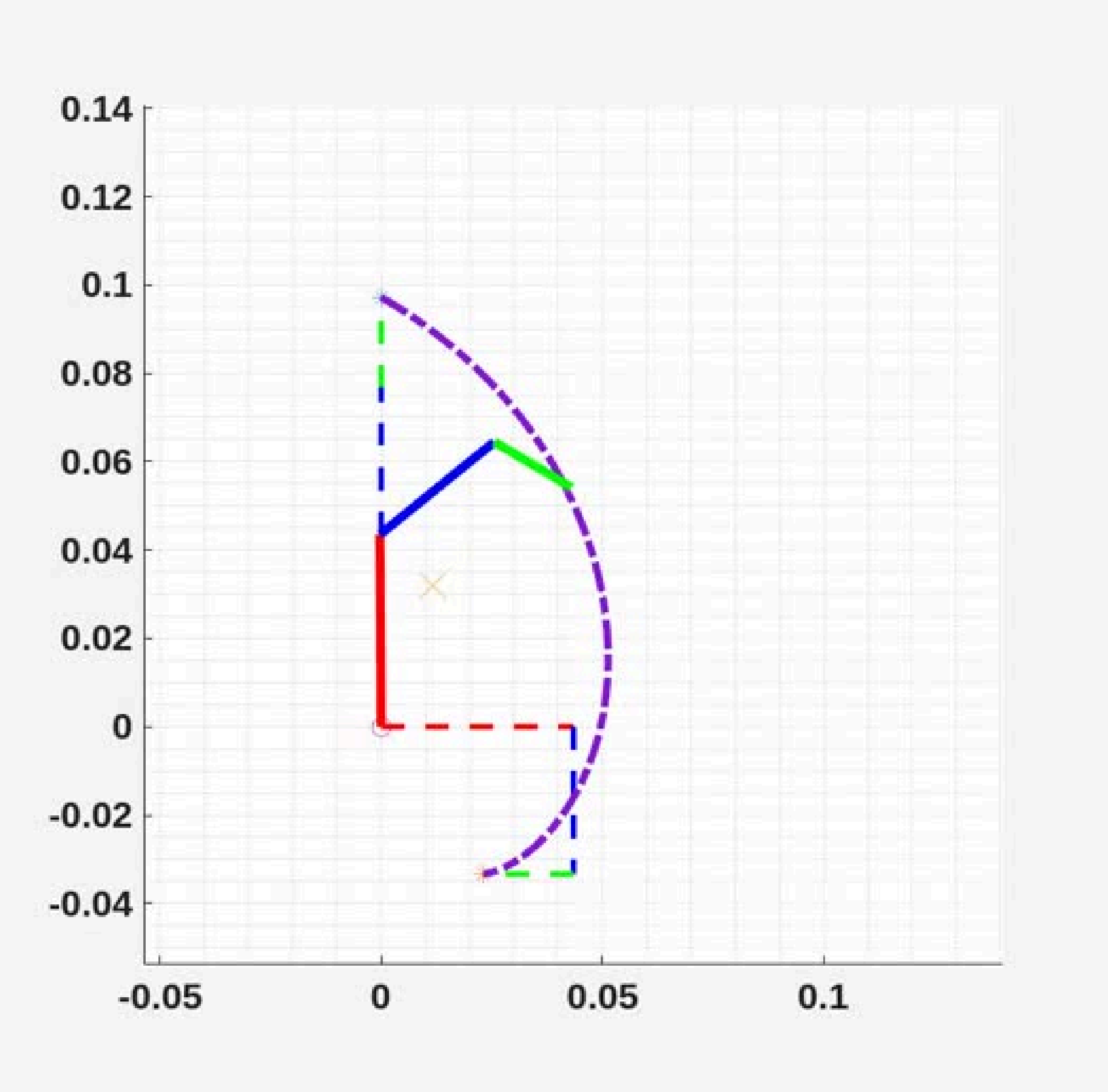
Two main approaches were taken into consideration:

Arc of a Circle

Since each finger responds differently to different radii, the radius had to be calibrated manually for each finger.

Cardiod Curve

Given the orientation one and only one cardiod can be defined through the two selected points, so there is no need of manually calibrating a radius for each finger.



→ This resulted in a more natural movement

Path Configuration

Two main approaches were taken into consideration:

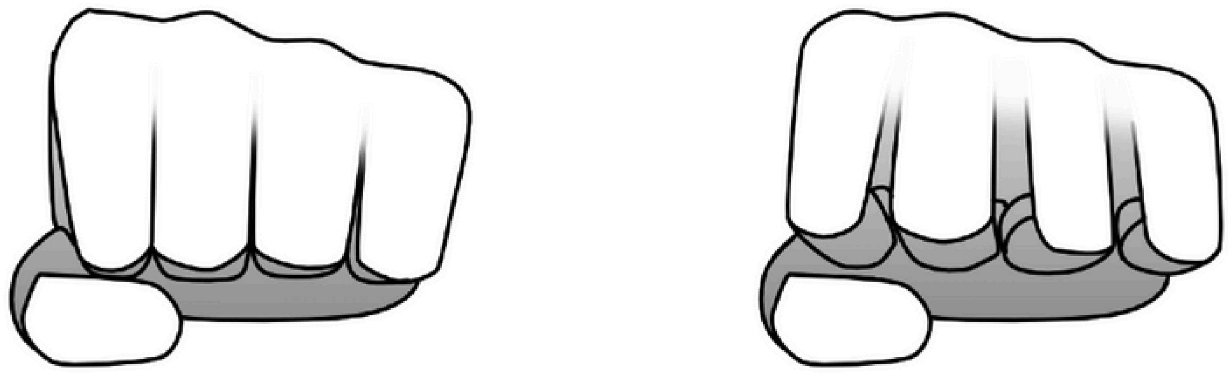
Arc of a Circle

Since each finger responds differently to different radii, the radius had to be calibrated manually for each finger.

Cardiod Curve

Given the orientation one and only one cardiod can be defined through the two selected points, so there is no need of manually calibrating a radius for each finger.

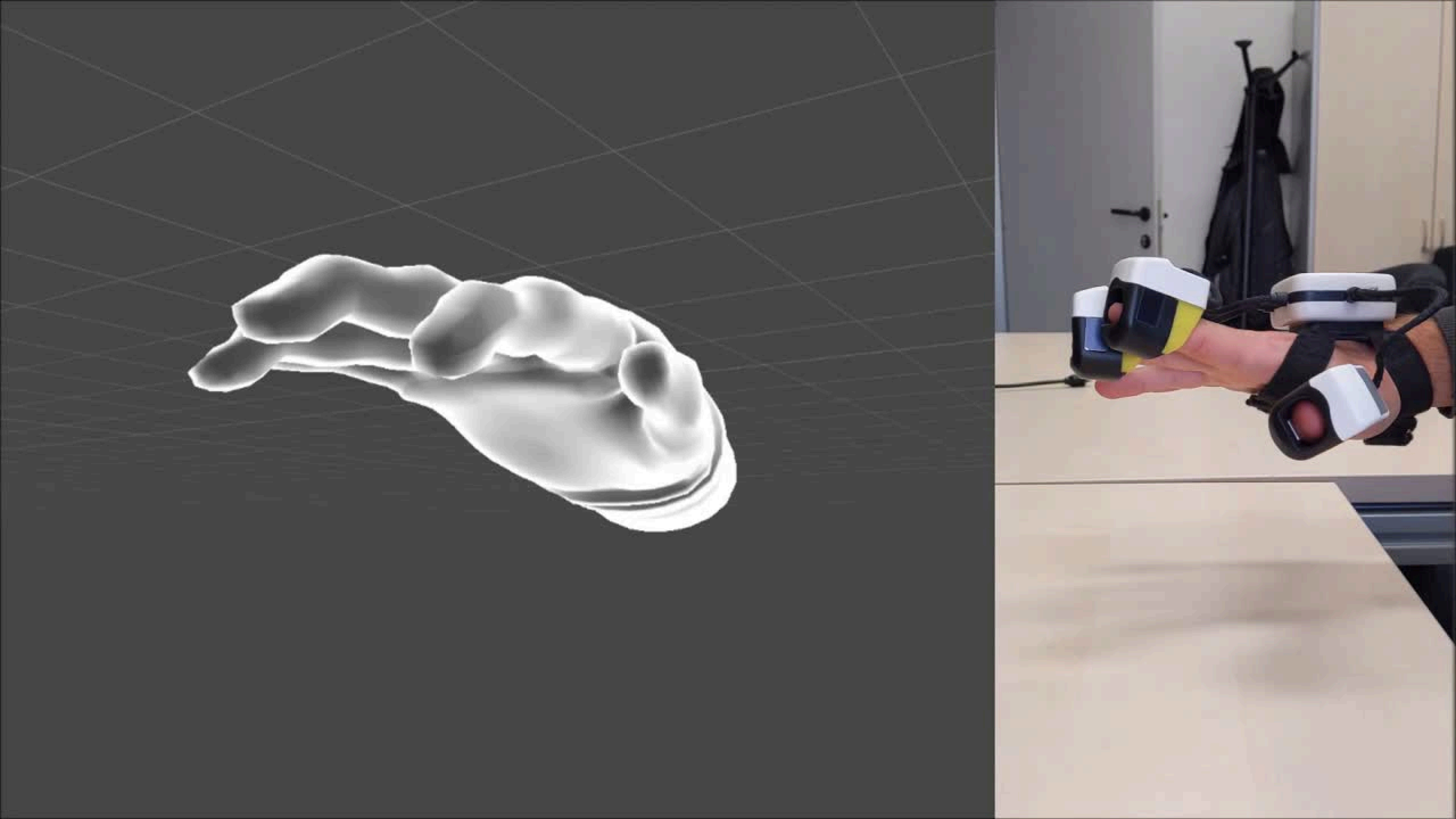
Splaying



Fist position, with and without splaying correction.

↓

This resulted in a more natural movement



Grasping System

Each finger has an **haptic collider** and a **proximity collider** used to capture collision events.

Once a collision is detected the system locks the finger to prevent it from closing beyond the point of contact and the grasping event can begin.

Grasping Conditions:

- the thumb is in contact
- at least one other finger is in contact

The global state variable is updated from Released to Grabbed, and a grasping event is emitted.

VR Integration

The tracking provided by the headset was used solely to determine the position and orientation of the hand in three-dimensional space.

During grasping, finger closure is regulated by TouchDiver, while global positioning and hand rotations are governed by headset tracking.

The TouchDiver data continued to govern the dynamics of finger closure.

Meta SDK Workflow

1. **OVR Hands Architecture** — OVR Hands is the main component that controls all the other components responsible for hand rendering.

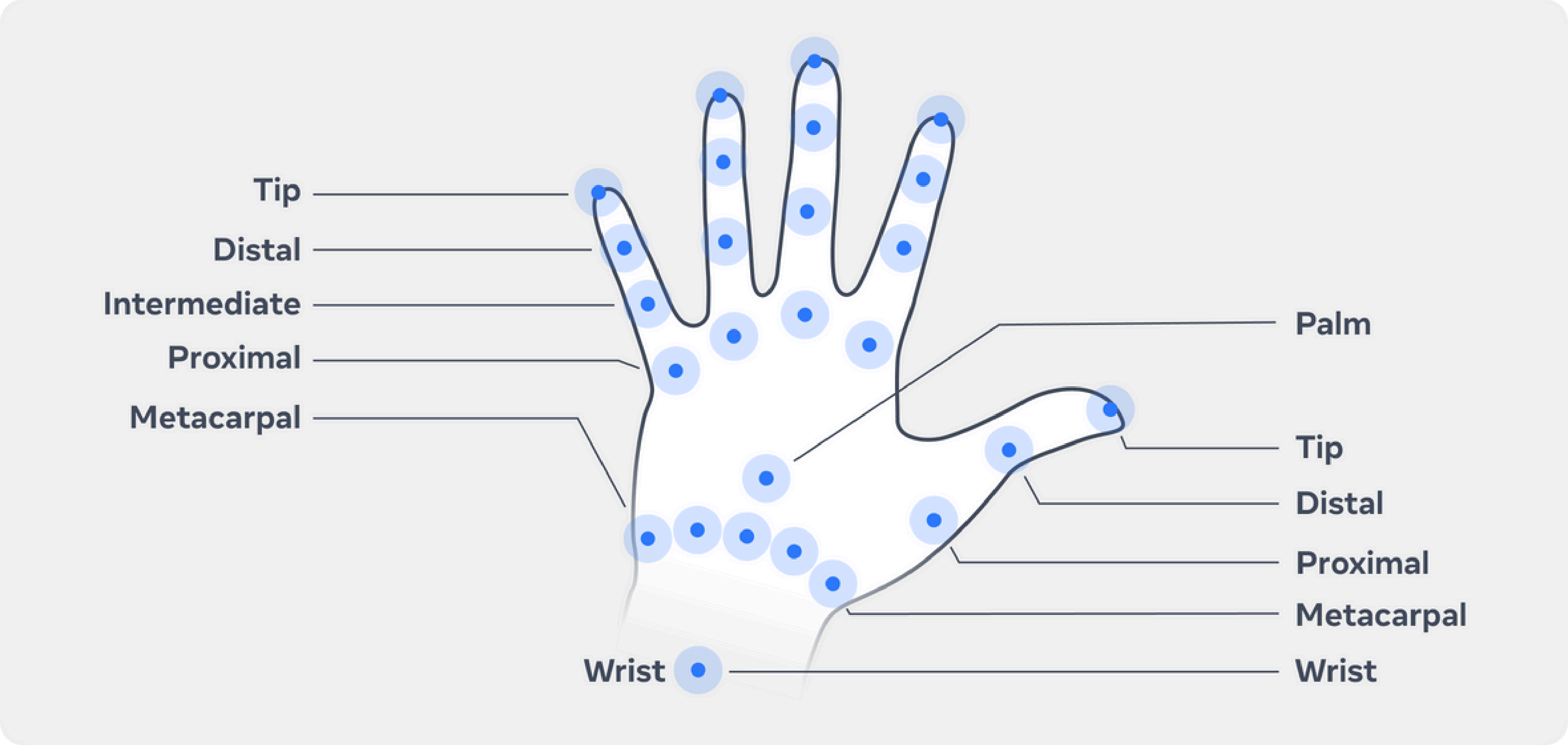
2. **Grasping System** — The main objective was to implement a feature that was coherent with the physics interactions between rigid bodies.

3. **Avoidance of Hand Object Interpenetration** — This was a necessary implementation in order to render a much more accurate and believable grasping action.

Ovr Hands Architecture

During runtime, upon detection of the hand, all the child components of OVRHand are built in order to return an accurate rendering: →

- **OVRSkeleton**
- **OVRMesh**
- **OVRSkeletonRenderer**



The SDK provides a computer vision model to estimate the pose (position and rotation) of different bones and joints of the hand.

Each highlighted section on the picture represents each of the capsule colliders present in the hand.

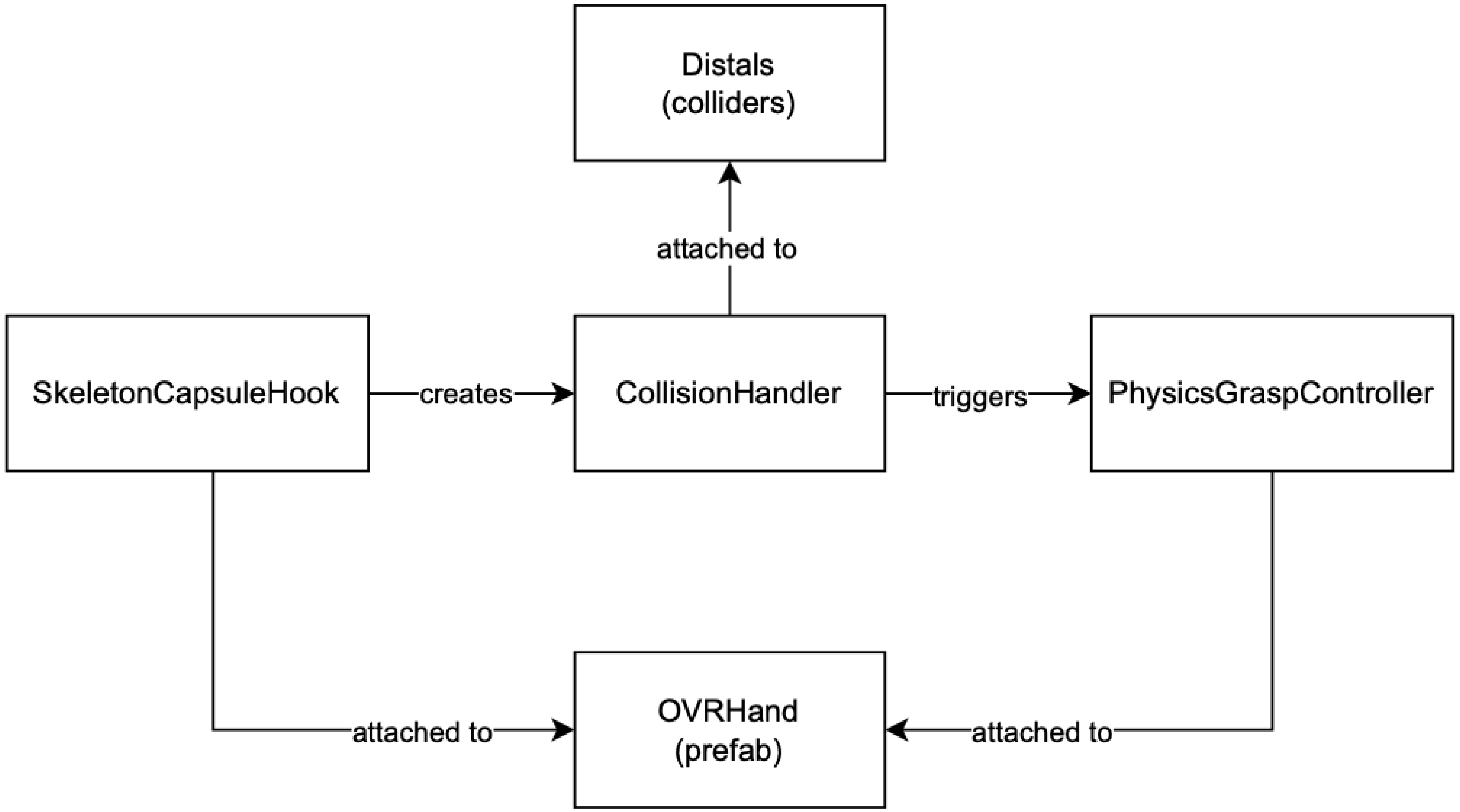
Grasping System

Grasping Conditions:

- the thumb or palm is in contact
- at least one other finger is in contact
- palm is facing the rigidbody within a threshold distance

↓

The object pose is then attached to the hand, and the minimum distance between each distal and the object is saved.



We introduced a new condition in the **EndGrab()** function: if the hand is open the object is automatically released.

Avoidance of Hand-Object Interpenetration

The main objective was to avoid interpenetration within the object's surface. → This was essential in order to render the grasping action much more naturally.

This was achieved creating a carbon **copy** of the hand where all physics components are disabled, including OVRSkeleton which is responsible for the visual mesh.

Once a grasping action is detected the mesh of the hand is blocked at the last known state.

Testing Scenes

To better measure the abilities of both methods a test scene was built and implemented:

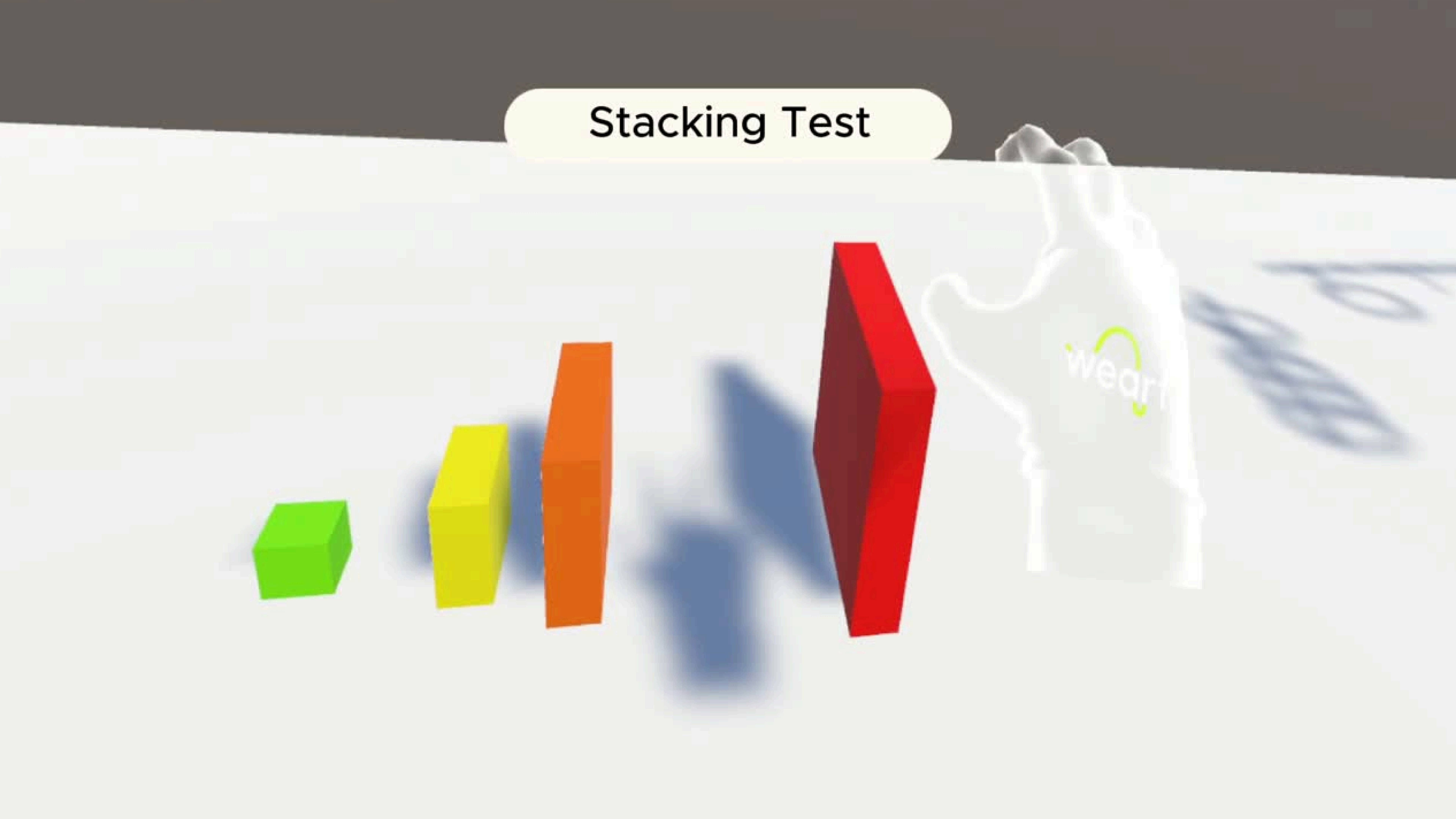
1. **Stacking Test**
   Multiple boxes are laid out to be balanced in a tower construction

2. **Rings on Hooks**
   A hook stand is placed in front of the user with multiple ring objects to place

3. **Point to Center**
   The user must hold a thin pen-shaped object to aim at a target's center

4. **Moving Object**
   A free falling object has to be grasped mid-air

5. **Blind Test**
   A cube must be grabbed blindly, after its position has been shown to the user for 1s

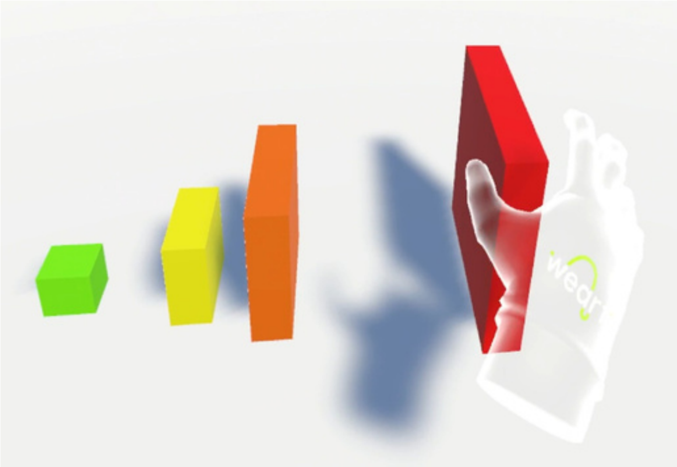
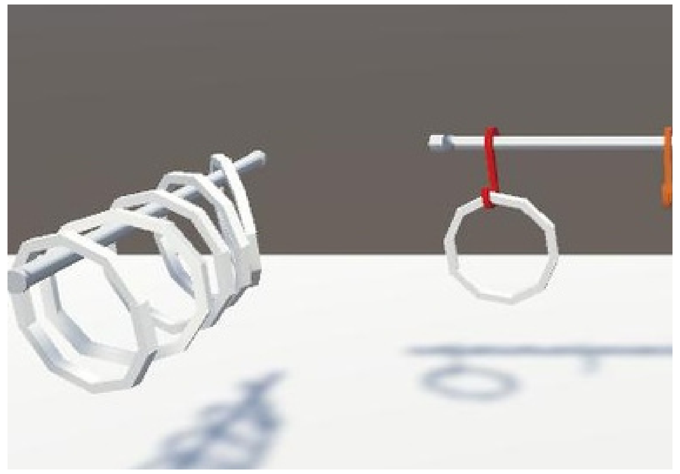
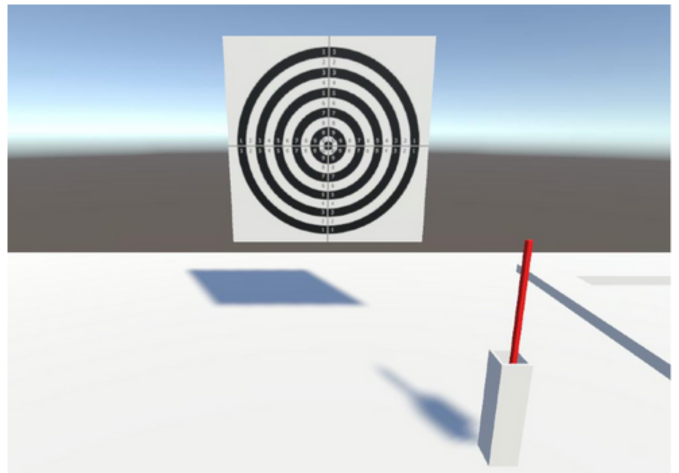
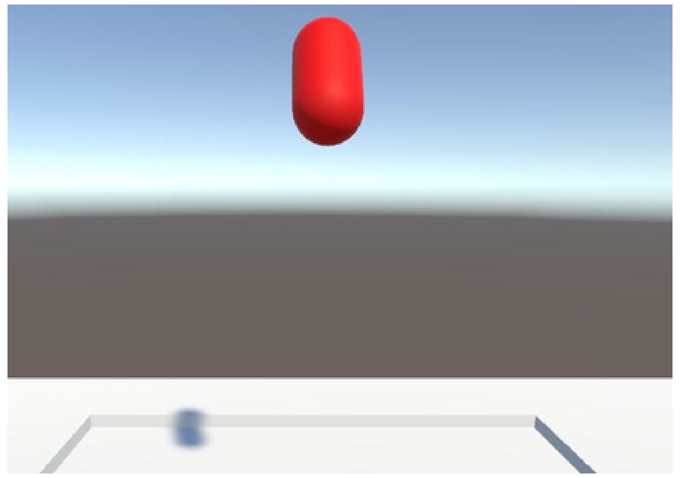

Stacking Test


Stacking Test

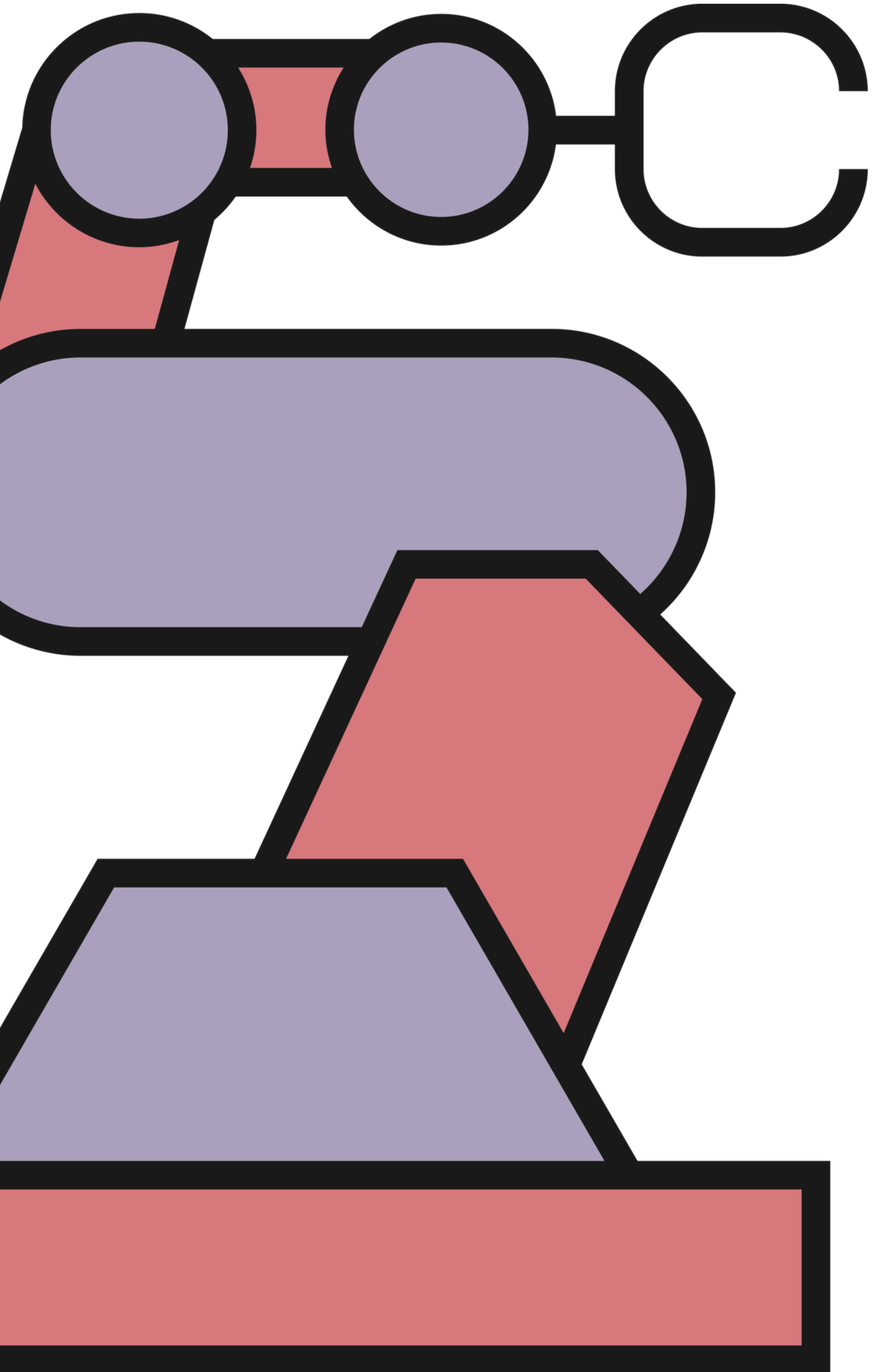
Results

Percentage of **success** for each test within the two methods:

	Stacking Test	Rings on Hooks	Point to Center	Moving Target	Blind Test
WeArt	19%	31,1%	69,7%	11,1%	44,4%
Meta	39,8%	45,2%	93,6%	70,3%	100,0%


Meta consistently outperformed WeArt in all tests, even though it's lacking touch-feedback.



Thank you for your attention!