Virtual Reality Systems

EPFL Immersive Interaction Group

Outline

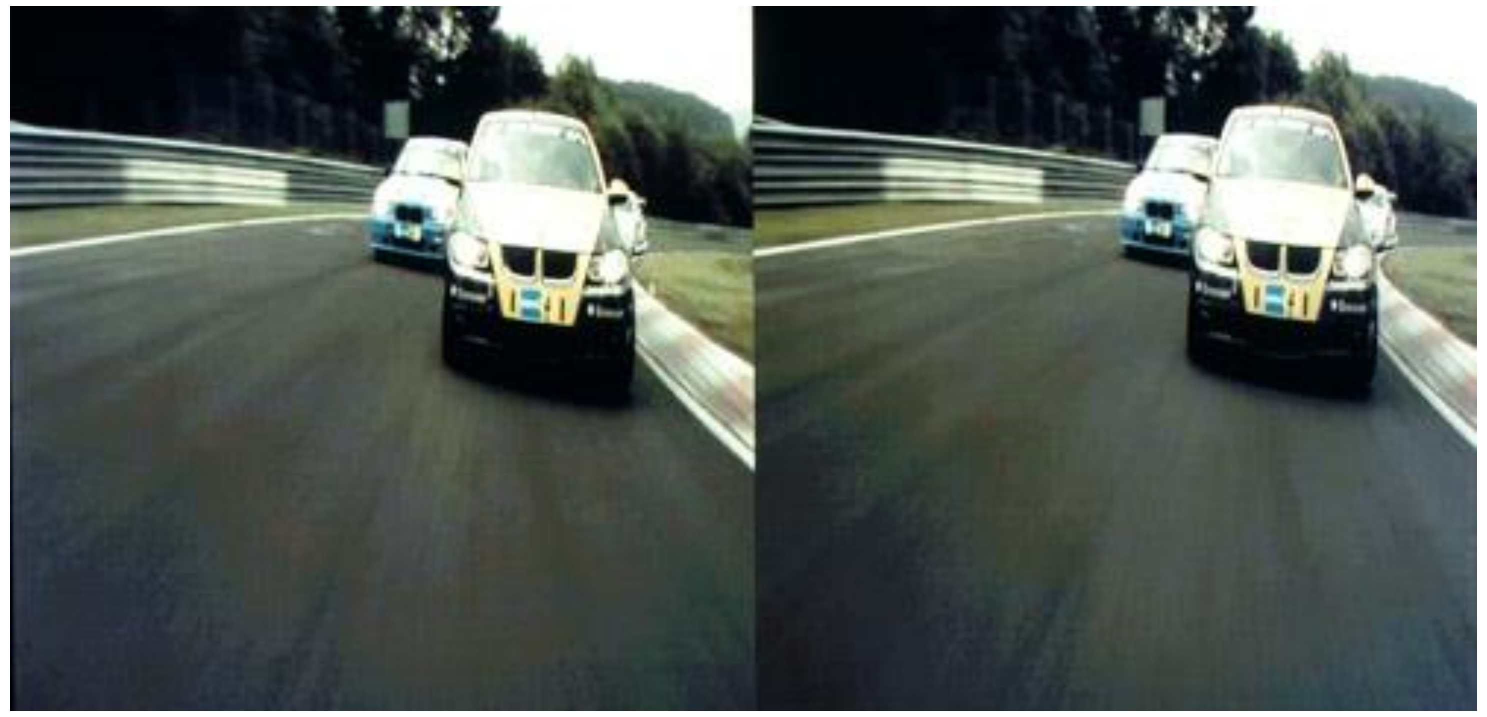
- Overview of Head-Mounted Display (HMD)
- Tracking System
- Input Devices
- Software Environment
- Project description
- Example of Tutorial
- Project deadlines

Head Mounted Display

Optical solution: Head-Mounted Display (HMD)

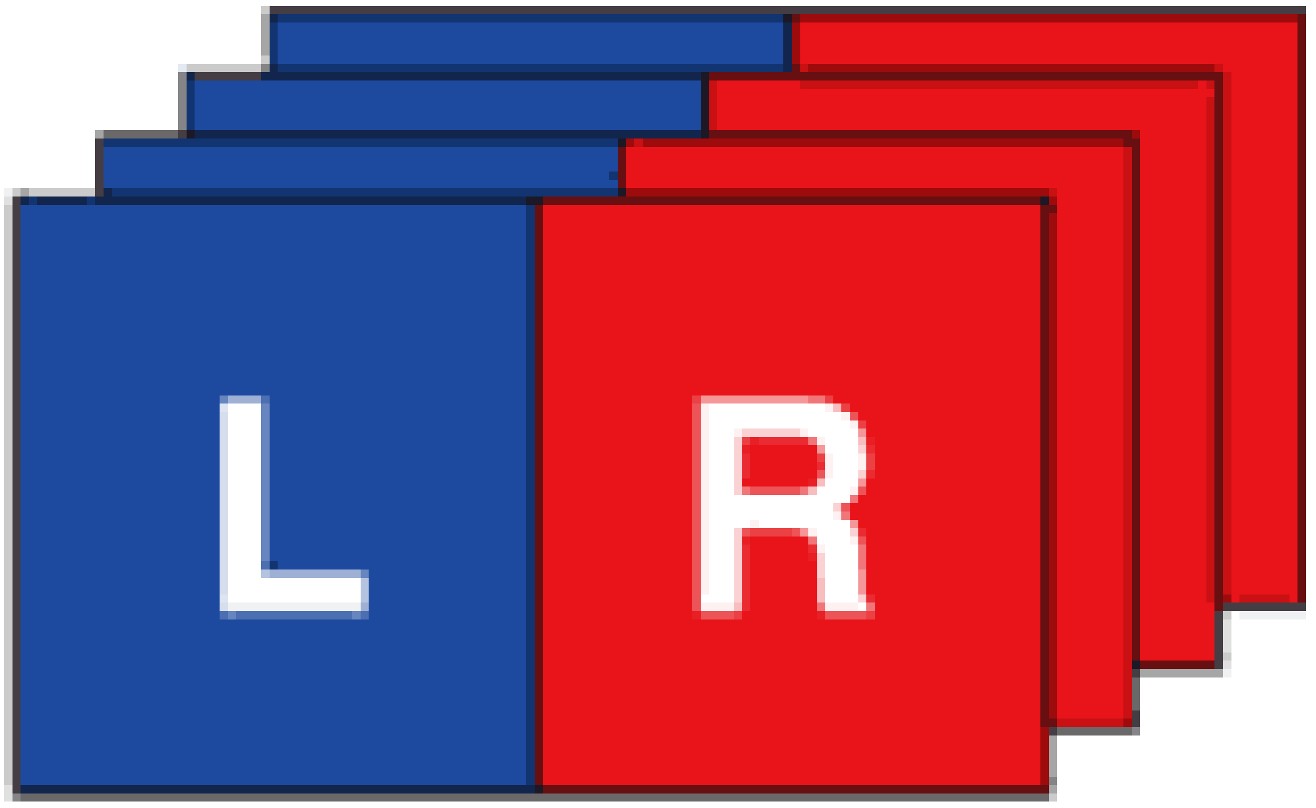
➔Providing 3D visual stimuli through head-worn systems

- Displaying the correct image directly to **each eye**
- Sutherland thesis (60s), commercially available in the 80s (Lanier)



Most common 3D format : **Side by side**

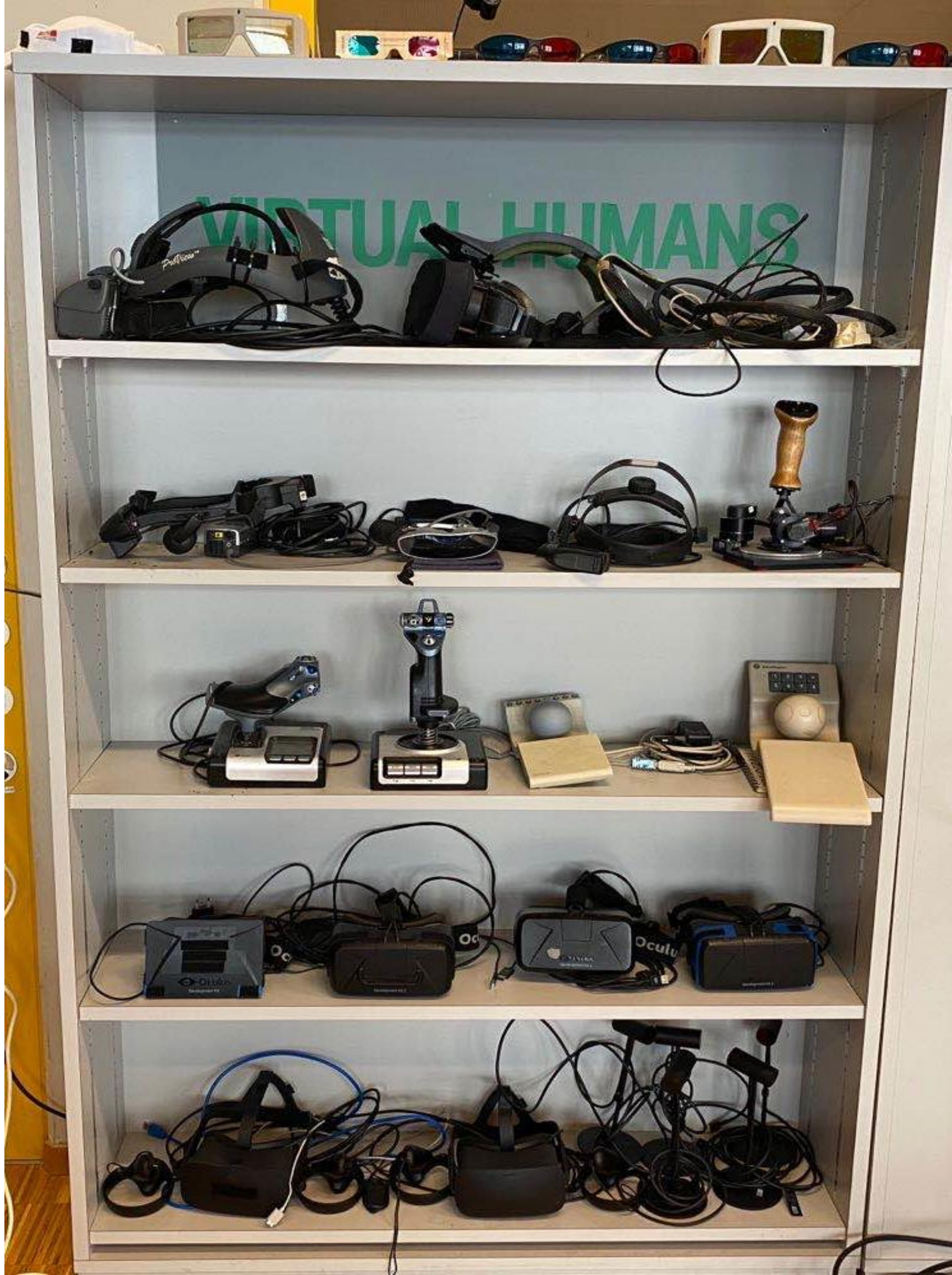
- Simultaneous display of left and right eye images at each frame
- Provide full frame rate at the cost of image resolution



Head Mounted Display

Some old models are visible in IIG

The rest of the slides focus on recent models



Head Mounted Display

- Oculus Series (Rift, Rift S, Quest & Go)
- HTC Series (Vive, Vive pro eye, Cosmos)
- Apple vision
- Samsung Gear VR
- Pimax (5k Plus, 8K X and Plus)
- Playstation VR
- Google Cardboard
- Nintendo Labo VR
- Valve Index

Head Mounted Display

Oculus Quest

- All-In-One VR Gaming
- Oculus Insight Tracking
- OLED Display
- Selling point: No PC, No wire, No limits

	Oculus Quest	Oculus Quest 2	Oculus Quest 3
Starting price	$299	$399	$550
Pixels per eye	1440 x 1600	1832 x 1920	2064 x 2208
Screen refresh rate	72 Hz	120 Hz	120 Hz
Field of view	100°	100°	110°

Head Mounted Display

HTC Vive Pro Eye

Valve Index

- PC powered
- motion tracking with base stations

Selling point: Embedded Tobbi Eye tracker

→ Gaze-based + Blink-based interactions

Head Mounted Display

Device comparison

Features	Oculus Quest / Quest2 / Quest 3	HTC Vive Pro (Eye)	Valve Index	PiMax 5k Plus
Minimal requirements	A smartphone for the setup only	GTX 1070 Quadro P5000	GTX 970 AMD RX480	GTX 1070
Display technology	OLED	OLED	LCD	CLPL
Remote connection	Limited	DisplayPort 1.2+ USB 3.0	DisplayPort 1.2+ USB 3.0	USB 2.0/3.0 + DP1.4
HMD sensors	IMU, Gyroscope, Cameras	IMU, Gyroscope, (eye tracking -> IPD)	IMU, Gyroscope	IMU, Gyroscope
Controllers inputs	Buttons | Hand tracking	buttons| eye tracking	Capacitive touch / Force sensors	
Field of View	~ 100 / 110 / 110 degrees	~ 110 degrees	~ 130 degrees	~ 200 degrees
Resolution (per eye)	1440 x 1600 px / 1832x1920 px / 2064 x 2208 px	1440 x 1600 px	1440 x 1600 px	2560 x 1440 px
Refresh Rate	72 Hz / 120 Hz / 120 Hz	90 Hz	90 / 120 / 144 Hz	120 Hz
Price (AVG)	< 600 CHF	CHF 1700	CHF 1100	CHF 810

Tracking system

Camera based

- Marker based active tracking
- Marker based passive tracking
- Markerless tracking

Pros

- Absolute position without drift over time
- relatively accurate devices

Cons

- Occlusions

Camera free

- Mechanical capture (exoskeleton)
- IMU (accelerometers, gyroscope)
- Magnetic sensors
- Deformable gauges (mostly used in gloves)

Pros

- No occlusions

Cons

- Lower accuracy (all)
- distortion induced by metallic objects (magnetic)
- Drifts (IMU)

Tracking system

Lighthouse / Base station

- Active tracking marker based
- Rotating laser @6000rpm
- Range of 7m per base station
- FoV : 160° x 115°
- 4 Base stations can cover up to 10 x 10 m surface
- The device scan the environment to identify without error the ID of each device

HTC Vive Tracker

Tracking system

Vicon Shogun

- Passive marker based solution
- High refresh rate
- High accuracy
- Unable to identify markers without context
- Expensive system
- Targets a professional market

Performer equipped with passive suits for motion capture using Vicon Shogun

Tracking system

Oculus Quest Tracking

- **Inside-out Computer vision tracking**
- Use computer vision with wide angle camera based sensors to locate the headset in space
- Doesn’t requires external devices
- These cameras also provides a **markerless finger tracking**

Input Devices

Oculus Touch

Each controller contains

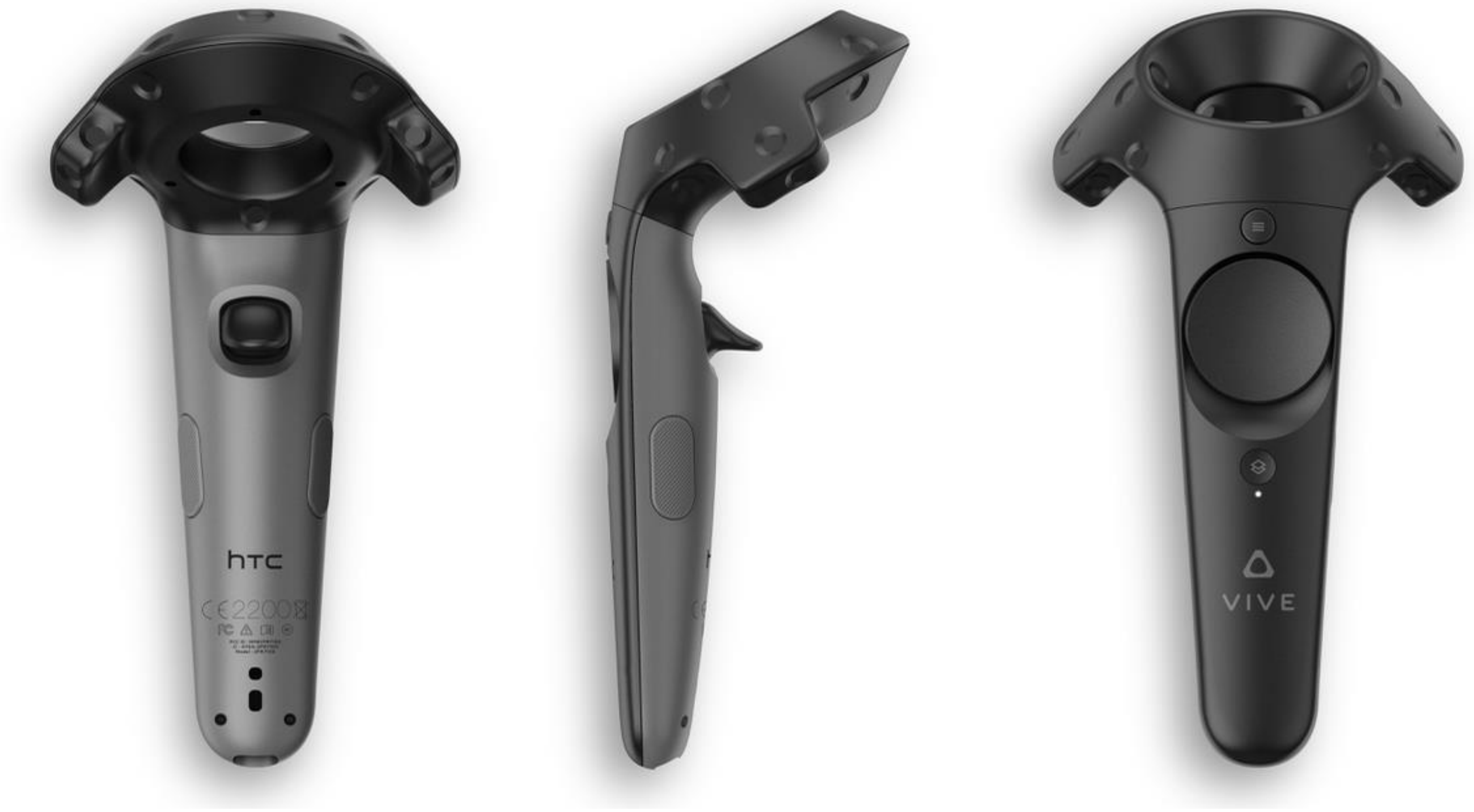
- One joystick
- Two press buttons
- *Two trigger buttons*
- One meta button
- Infrared tracking
- IMU and Gyroscope
- Vibrators

Input Devices

Vive Controller

Each controller contains

- A trigger
- Two meta buttons
- *A tactile button pad*
- Two lateral buttons
- IMU and Gyroscope
- Infrared tracking
- Vibrators



Input Devices

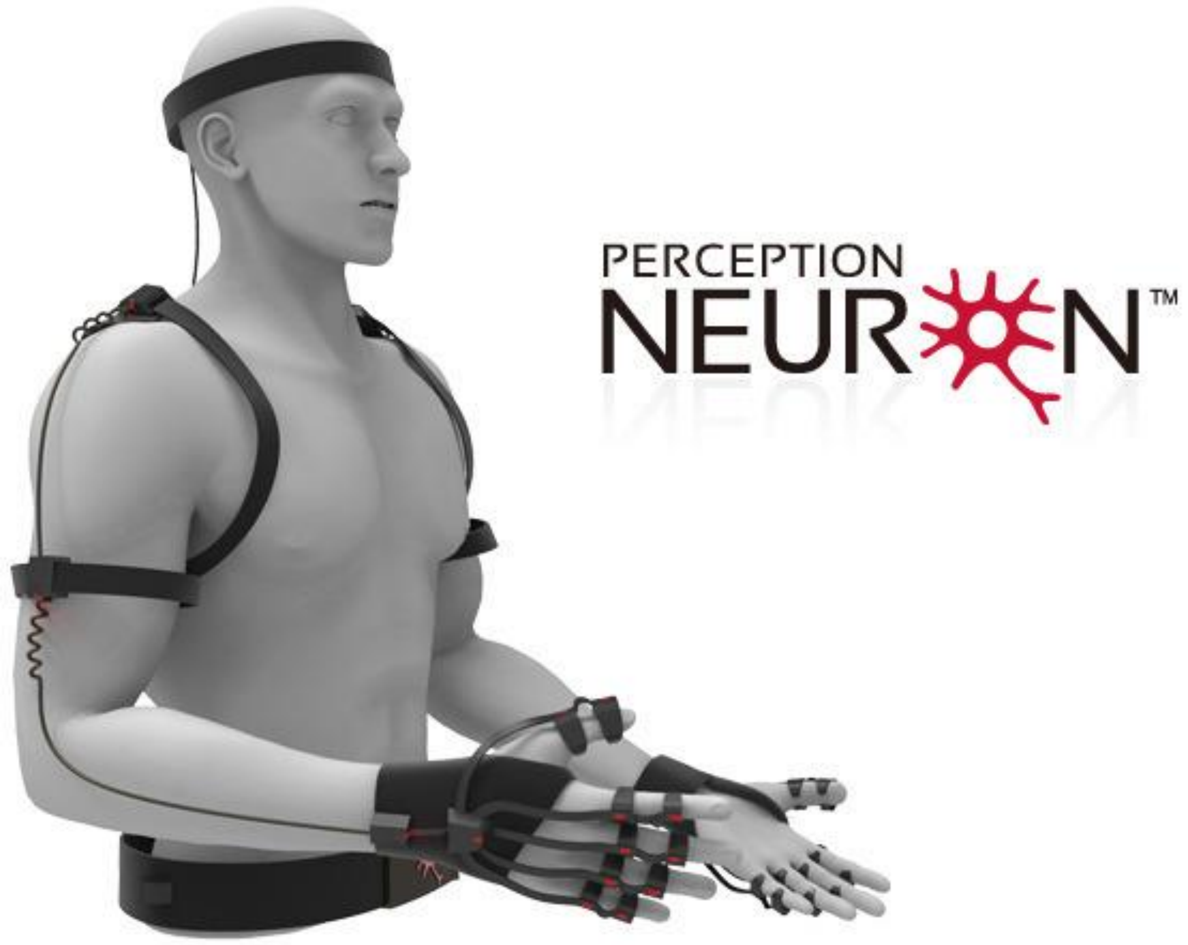
Knuckles

Each controller contains

- One joystick
- Two press buttons
- One trigger
- One meta button
- *Finger tracking through proximity sensors*

Input Devices

Miscellaneous Inputs



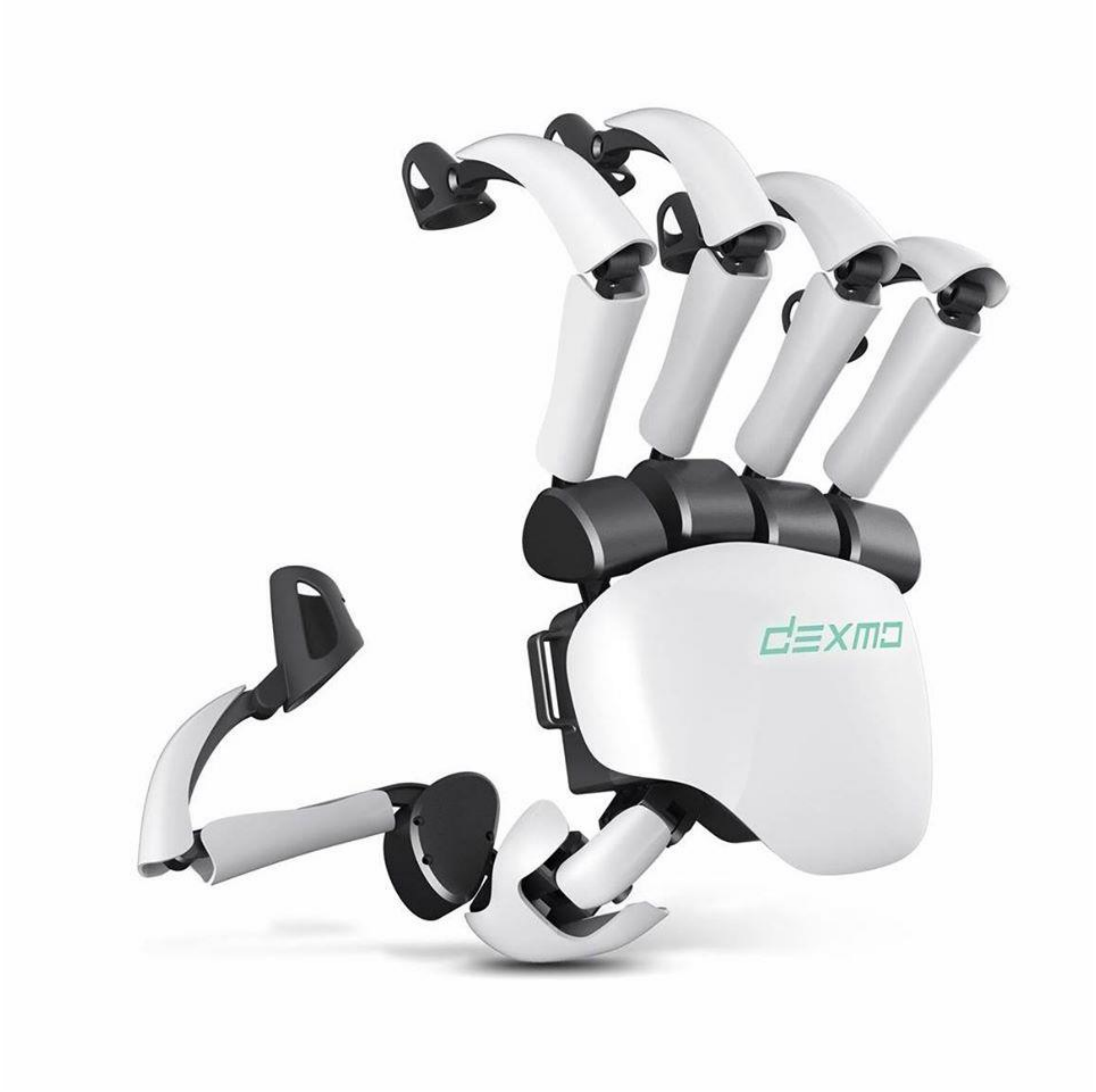
Manus VR

PlayStation Controllers

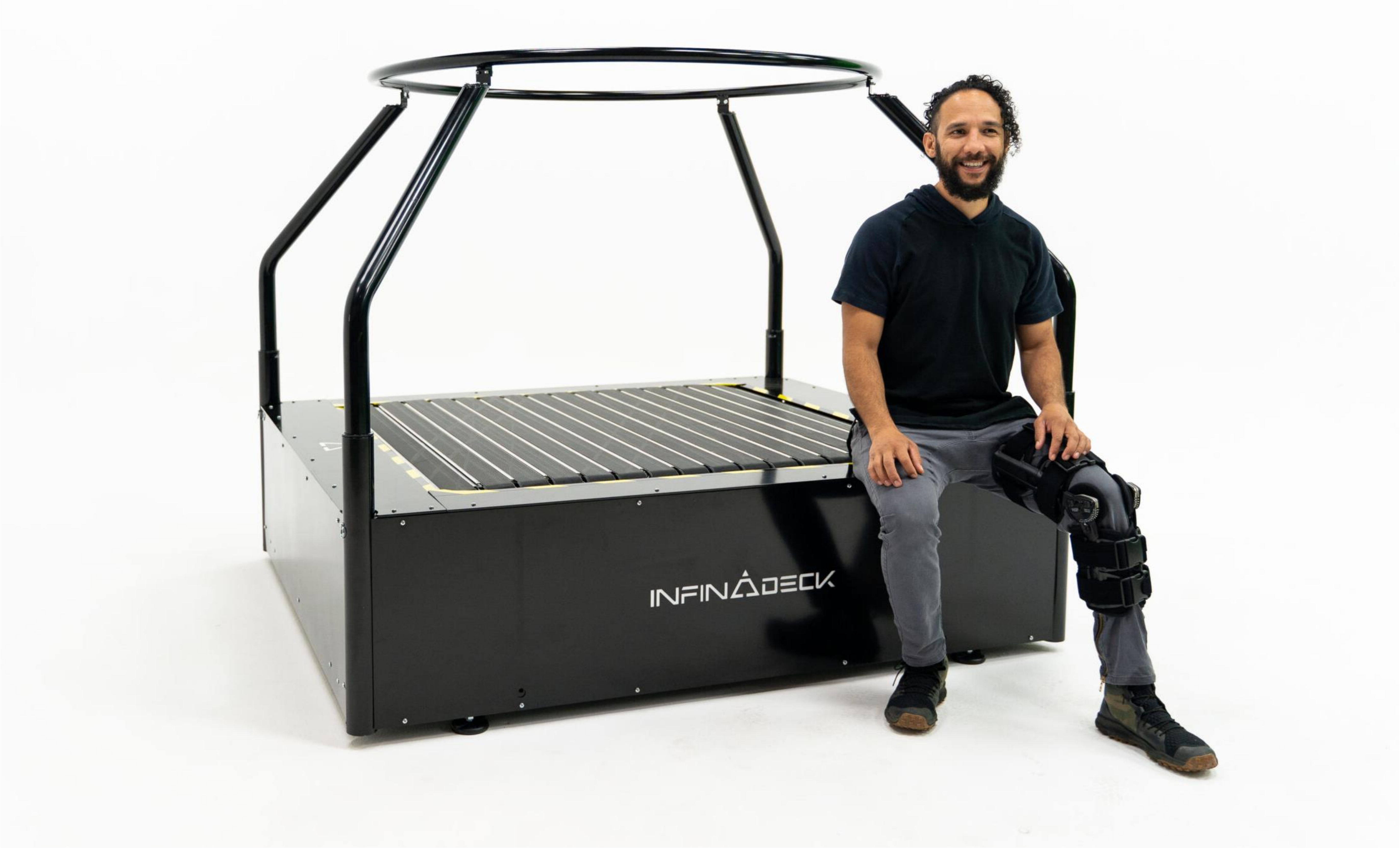
Windows Mixed Reality

Input Devices

Miscellaneous Devices (+/- non finished prototypes)



Dexmo Glove



Infinadeck’s Bi-directional treadmill

Software Environment : Game Engine / Editor

- Handles the core of the Game such as
  - Frames
  - Rendering
  - Sound
  - Collisions
  - Physics
  - Etc.

- Provides a framework for developers

Software Environment : Game Engine / Editor

Common Games Engines

Proprietary	Open Source
UNREAL ENGINE unity	GODOT Game engine
lumberyard CRYENGINE®	PANDA3D

Software Environment : Game Engine / Editor

Unity 3D

- Widely used (**many forums** available with tips)
- Licensed software (**free** for education / personal use)
- **Multi-target support** (Linux, **Android**, Windows, Mac, PS4, Switch, ...)
- Scripting in **C#** (or JS)
- Perfect integration with **Visual Studio**
- Many resources through the **asset store**
- Technology we use within the IIG

Project description: concept

Group project : 3 or 4 students

➔ Build a **fully playable (and stable) game**

+ **Associated Tutorial System** detailing the mechanics and interactive methods on how to play their game, **must be self-sufficient**

+ **Associated 3D environment**

+ **written detailed report**

+ **video showing a playthrough of the game**

Freeform game as long as it respects the requirements of the project

Project description: requirements

Build a **fully playable game** with at least **4 types of player interactions (5 types for groups of 4 students) and 2 locomotion methods** (i.e. player displacement; one must be teleportation)

Possible player interactions

- **Throwing**: taking into consideration the amount of force used
- **Filling and Grabbing a Container**: place certain objects within another. The player must then be able to pick up the container with the objects inside and hold on to it
- **Swing and Hit**: should take into consideration the amount of force used. Example: hitting a ball with a golf club
- **Alternate Grab:** grab and hold objects. Students must use an alternate method for this interaction than the ones implemented during the TP class.
- **Joint Interaction**: object with a joint component which the character needs to interact with. Example: a door or a chest being opened and closed by grabbing and pulling (without animation)
- **Two Hand Interaction**: Interaction that requires both hands interacting on the same object simultaneously. Examples: Pull apart an object by pulling from two sides
- **Shooting System**: Players should physically aim, shoot, and manually reload firearms. This system can be combined with haptic feedback, ammo system (weapon reloading) etc.

Project description: requirements

Build a **fully playable game** with at least **4 types of player interactions (5 types for groups of 4 students) and 2 locomotion methods** (i.e. player displacement; one must be teleportation)

Possible player interactions

- **Gesture-Based Spell Casting**: perform special actions by making specific gestures with their hands. The system would recognize gestures, allowing for a wide range of possible abilities, each with unique effects on the environment or enemies
- **Precision Crafting**: manipulate small components or materials to create or repair items. Would require precise hand movements, could utilize haptic feedback to simulate the tactile sensation of handling different textures and resistances

  Example: assembling a clock or fixing a device
- **Puzzle Solving**: Players should engage in physically interactive puzzles, like aligning components or manipulating lights for problem-solving. Puzzle solutions can unlock new areas, reveal story elements, or change the environment

Locomotion

- **Teleportation**
- **Alternate Locomotion**: another locomotion method than the ones implemented during the TP class

Project description: tools

Usable

- **Head and Oculus Controller Tracking**
- **3D Models, Animations, Audio and Texture/Shader Assets**
- **Interactions Implemented during the VR Course**: You can use all interactions developed during the TP classes as additional interactions to the ones asked for implementation. Keep in mind that all interactions asked must be substantially different than the ones previously implemented during class

Non-Usable

Interaction Frameworks: You cannot use popular frameworks such as VRTK or even copy examples out of the Oculus/SteamVR applications. **All code relating to both Locomotion, Interactions and the Game itself must be fully implemented by the students**

→ all students must comment their code detailing their methods correctly

→ In case of doubt : ask us

Project description: evaluation criteria

Evaluating Interactions

❑ **Basic Functionality**: Does it achieve the basic intended action? how well is it achieved? Students were careful to make the interaction stable

❑ **Quality / Usability** : Is the interaction as intuitive as possible within the context of the game? Is the way the interaction is triggered coherent with the other interactions implemented?

❑ **Feedback** : Is the interaction visually pleasing, is the audio feedback appropriate ? Students used passive/rumbling haptics as a mechanism to improve the interaction and its usability

Evaluating Locomotion

Same 3 criteria

+ Motion Sickness Inducing criteria: Is the motion highly susceptible to motion sickness ? Students were careful to consider the theoretical concepts relevant to motion sickness induction and attempted to develop something around it

Project description: evaluation criteria

Evaluating the game

□ **Creativity and Quality of the Game:**

- How original is the game? e.g. is it just a copy of an existing game?
- How well the interactions are integrated into the game?
- Is the game fun to play?
- Is the soundscape appropriate ?

□ **User Friendly:**

- How friendly is the game to new players.
- Is play intuitive?
- Are the controls easy to use (i.e. not frustrating)?

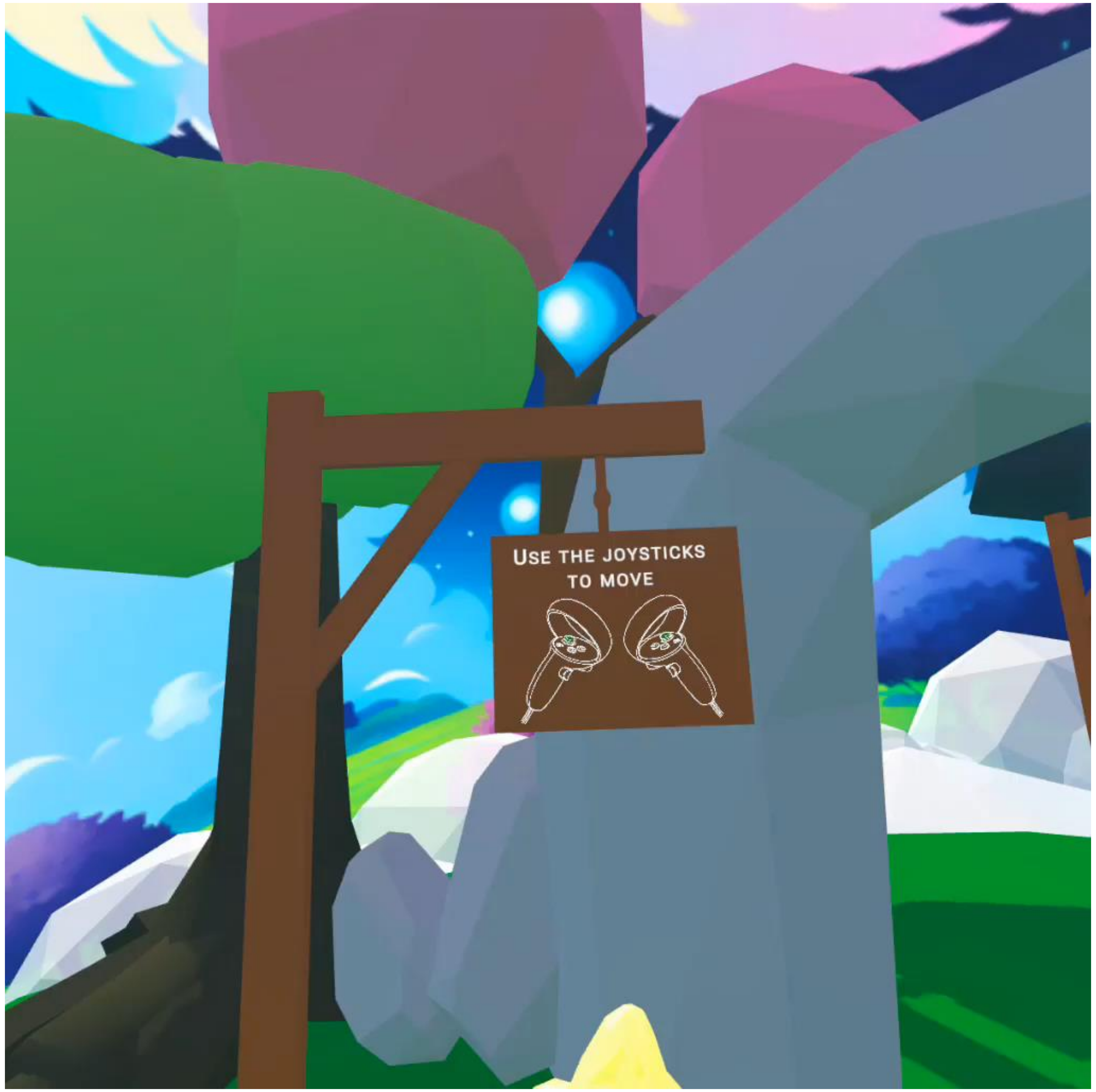
□ **Tutorial Quality:**

- Can the game teach us how to play?
- How well does the game explain its concepts to the player and how self-sufficient it is without any external help?

□ **Visual Fidelity/Quality:**

- How does the game look, and does it have a thematic consistency.
- Did students take some degree of care about the visuals of the game?

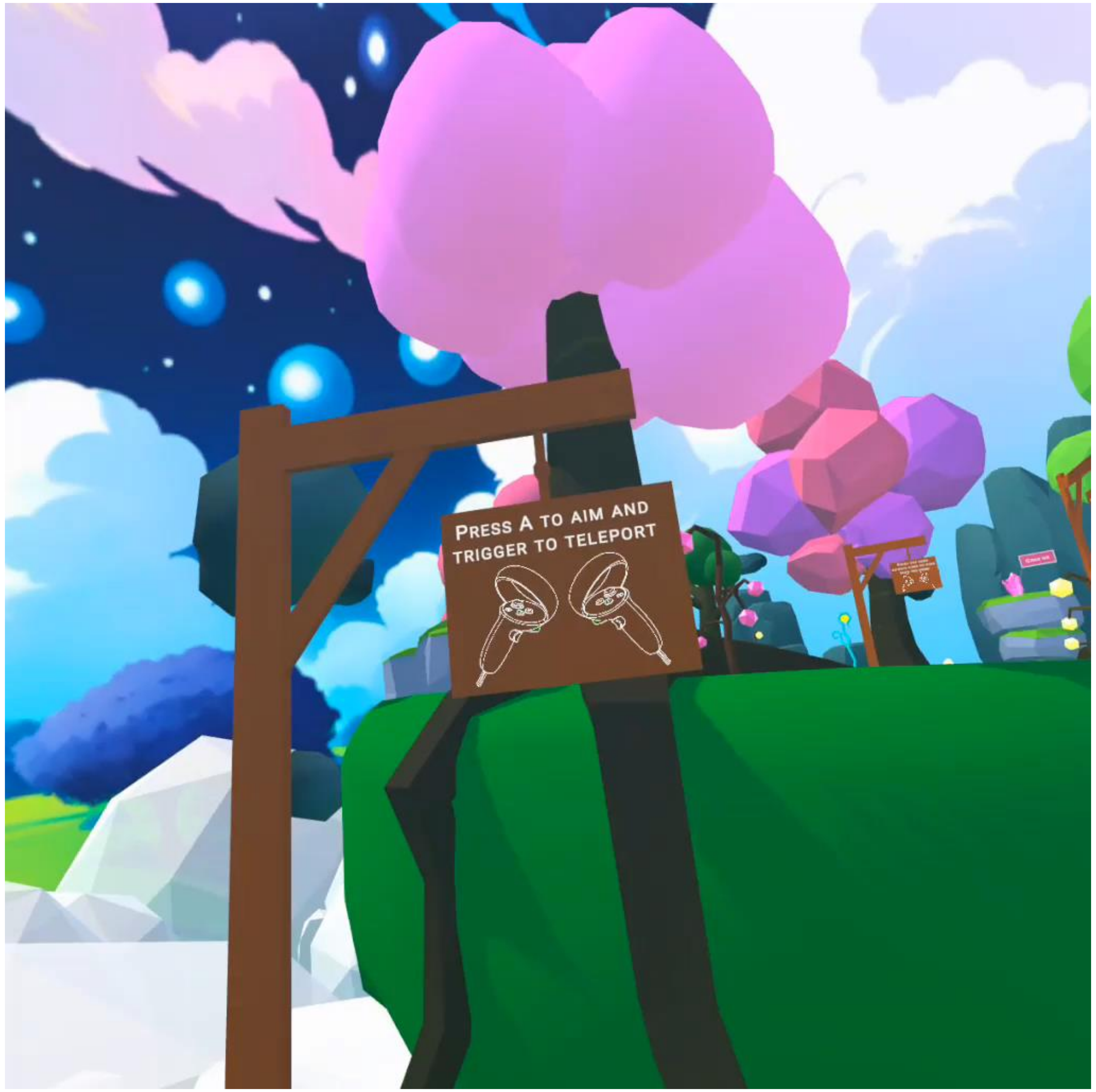
Tutorial example



Locomotion :

- **Linear** (oculus default): snap rotation to limit cybersickness

Tutorial example

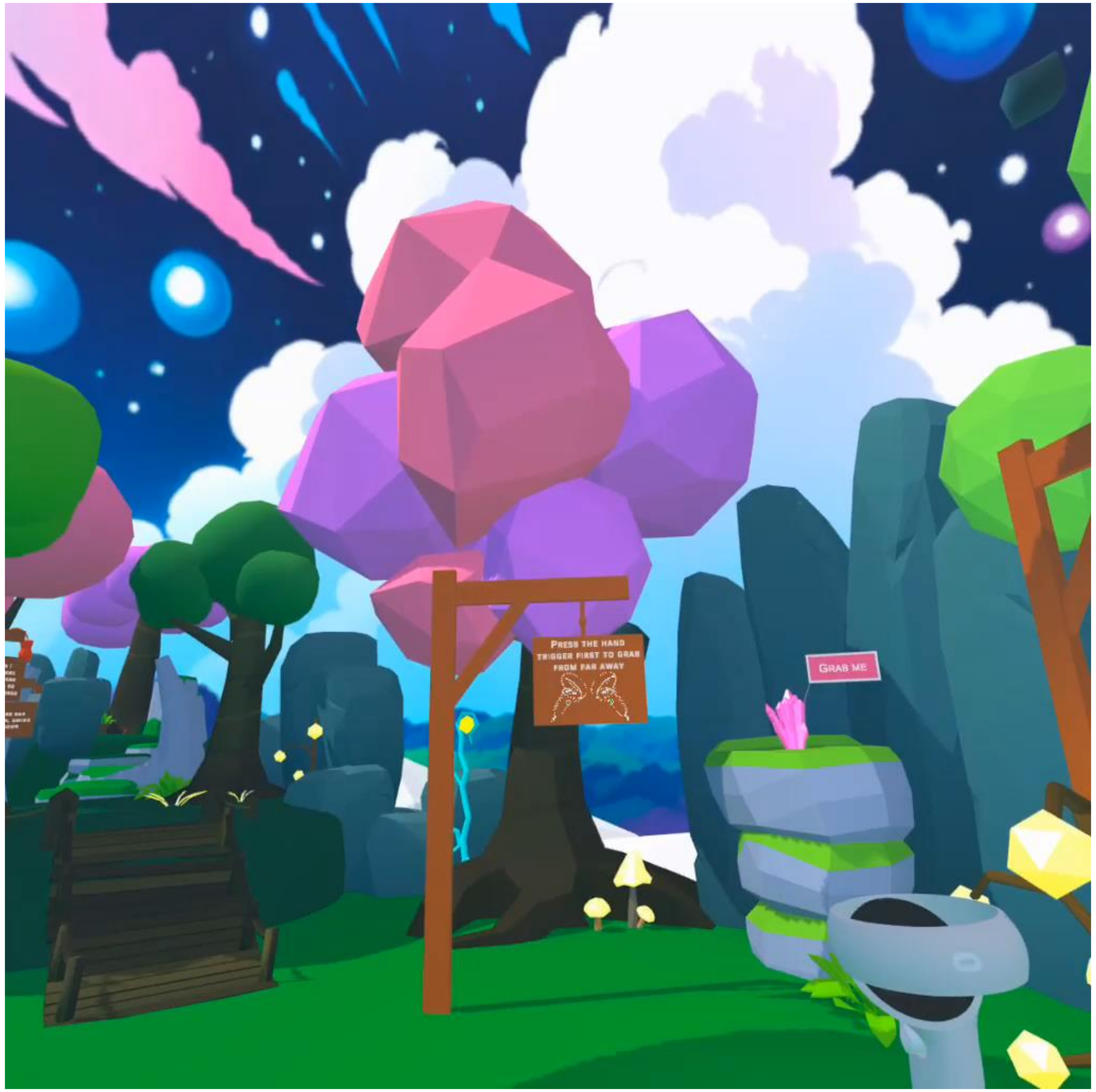


Locomotion :

- **Linear** (oculus default): snap rotation to limit cybersickness
- **Teleportation:** ideal to limit cybersickness
  - ❑ **Visual feedback:**
    - Red or Yellow marker + parabola ray

      → color depends on the state of the interaction
    - Black screen Fade on trigger to smooth the transition
  - ❑ **Audio Feedback:** sound on trigger
  - ❑ **Haptic feedback:** on trigger, on the controller triggerring the teleportation

Tutorial example



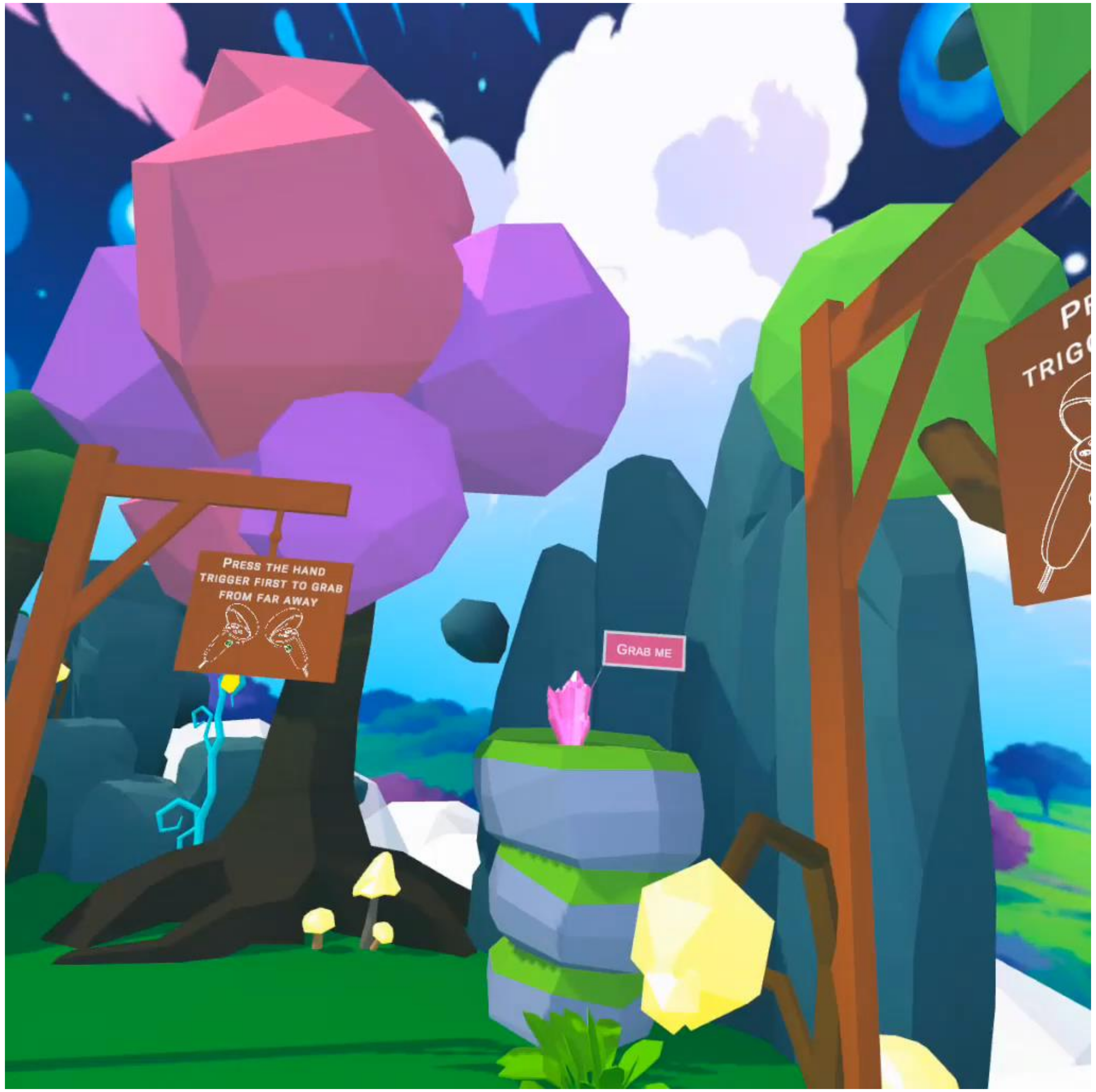
Grab: Triggered by pressing both controllers triggers

→ **Press both trigger near a grabbable object**

- ❑ **Visual feedback:** Outline when a hand is close enough to grab the object
- ❑ **Audio Feedback:** sound on trigger, depending on the object that is being grabbed
  → « crystal-like » sound here
- ❑ **Haptic feedback:** on trigger, on the grabbing hand

+ A grabbed object can be grabbed by the other hand

Tutorial example



Distance Grab: Triggered by pressing one trigger to aim and the other trigger to distance grab

→ Same buttons as classic grab but one by one

❑ **Visual feedback:**

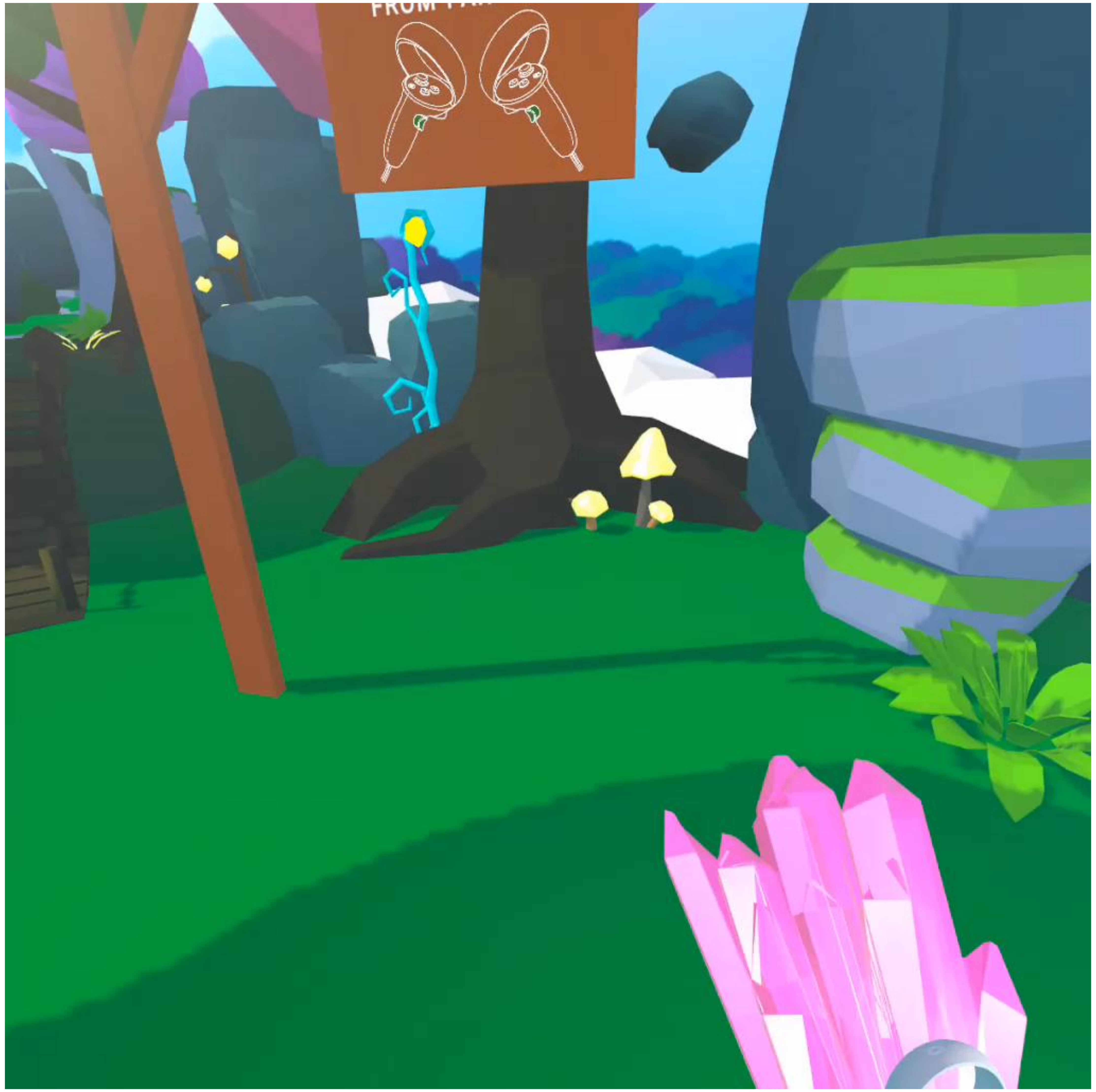
- Outline when a hand is close enough to grab the object
- Smooth movement toward the hand, slowing down when getting closer to the hand

❑ **Audio Feedback:** sound on trigger, depending on the object that is being grabbed → « crystal-like » sound here

❑ **Haptic feedback:** on trigger, on the controller triggerring the grab

+ A grabbed object can be grabbed by the other hand

Tutorial example



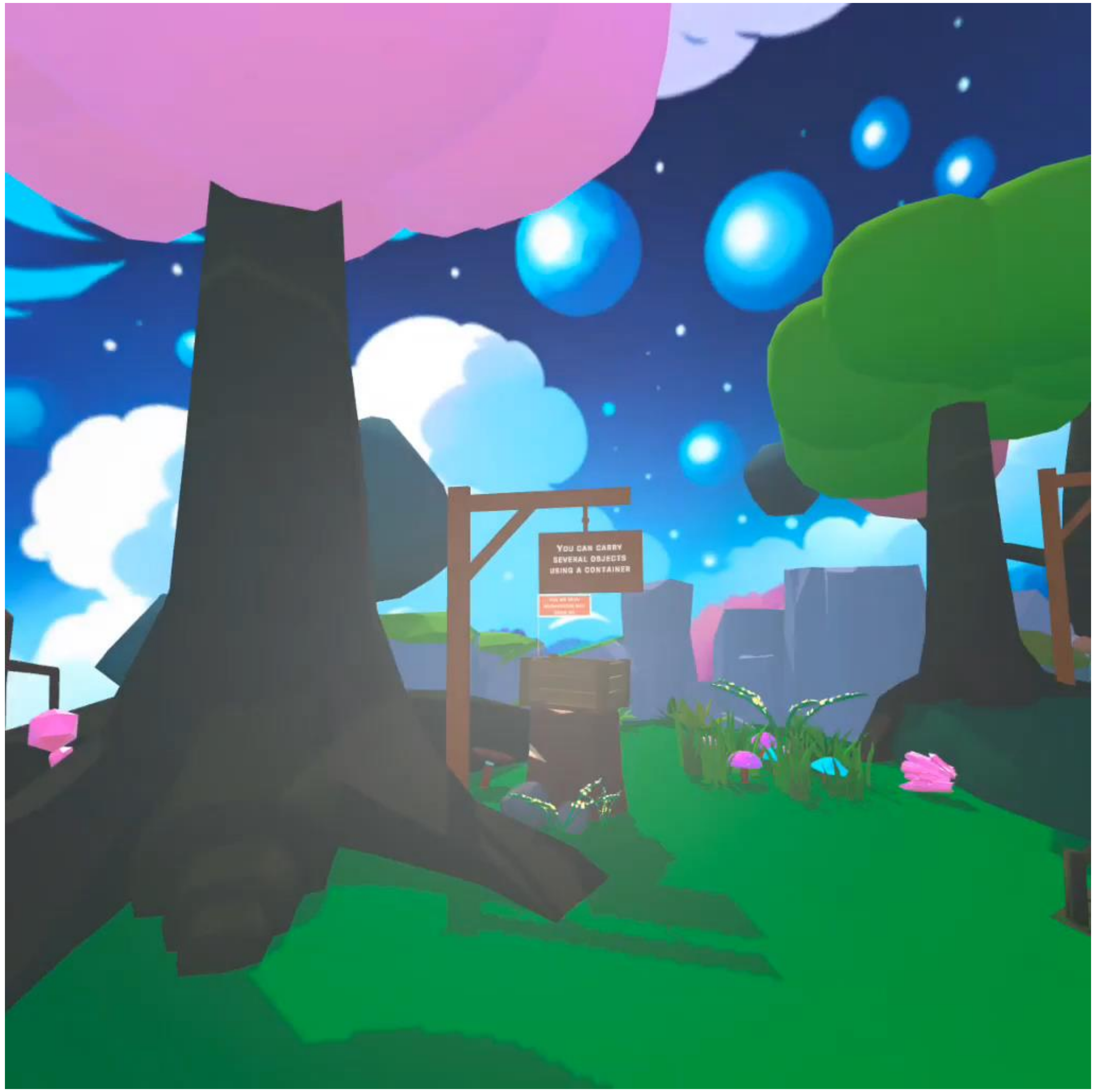
Throwing: depends on the velocity of the hand

→ Triggered by releasing the « grab » pose

❑ **Visual feedback:** direction and speed determined by the hand

❑ **Audio Feedback:** « woosh » sound if the speed exceed a determined threshold

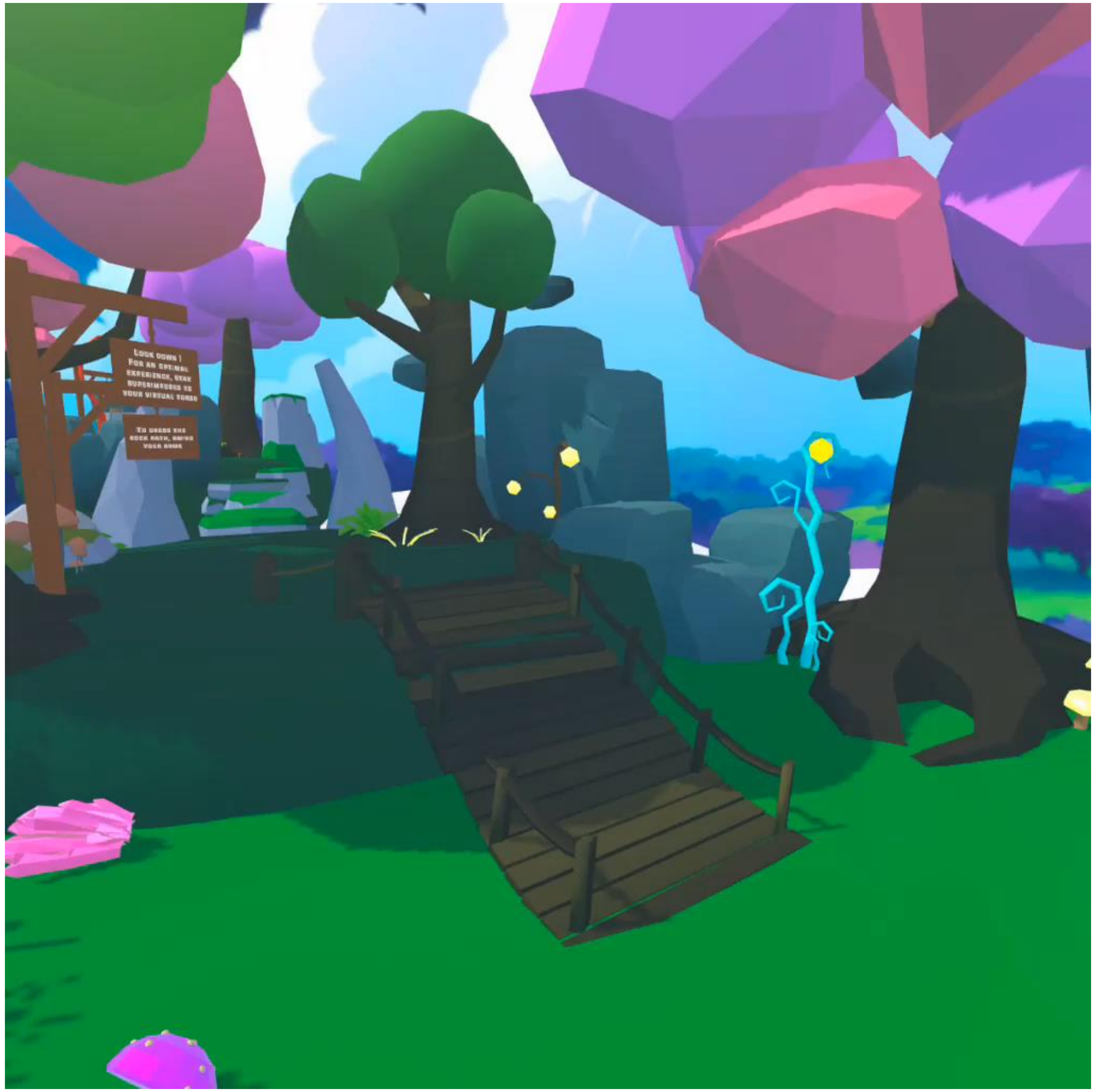
Tutorial example



Fill and grab a container:

- ❑ **Visual feedback:**
  - if objects are in the container when the container is grabbed, they are locked in the container
  - If the container is turned upside-down, the objects are released and fall
- ❑ **Audio Feedback:** grab sound, depending on the object that is being grabbed → « wood-like » sound for the container
- ❑ **Haptic feedback:** grabbing haptic

Tutorial example


Look down !
For an optimal experience, stay superimposed to your virtual torso
To cross the rock path, swing your arms

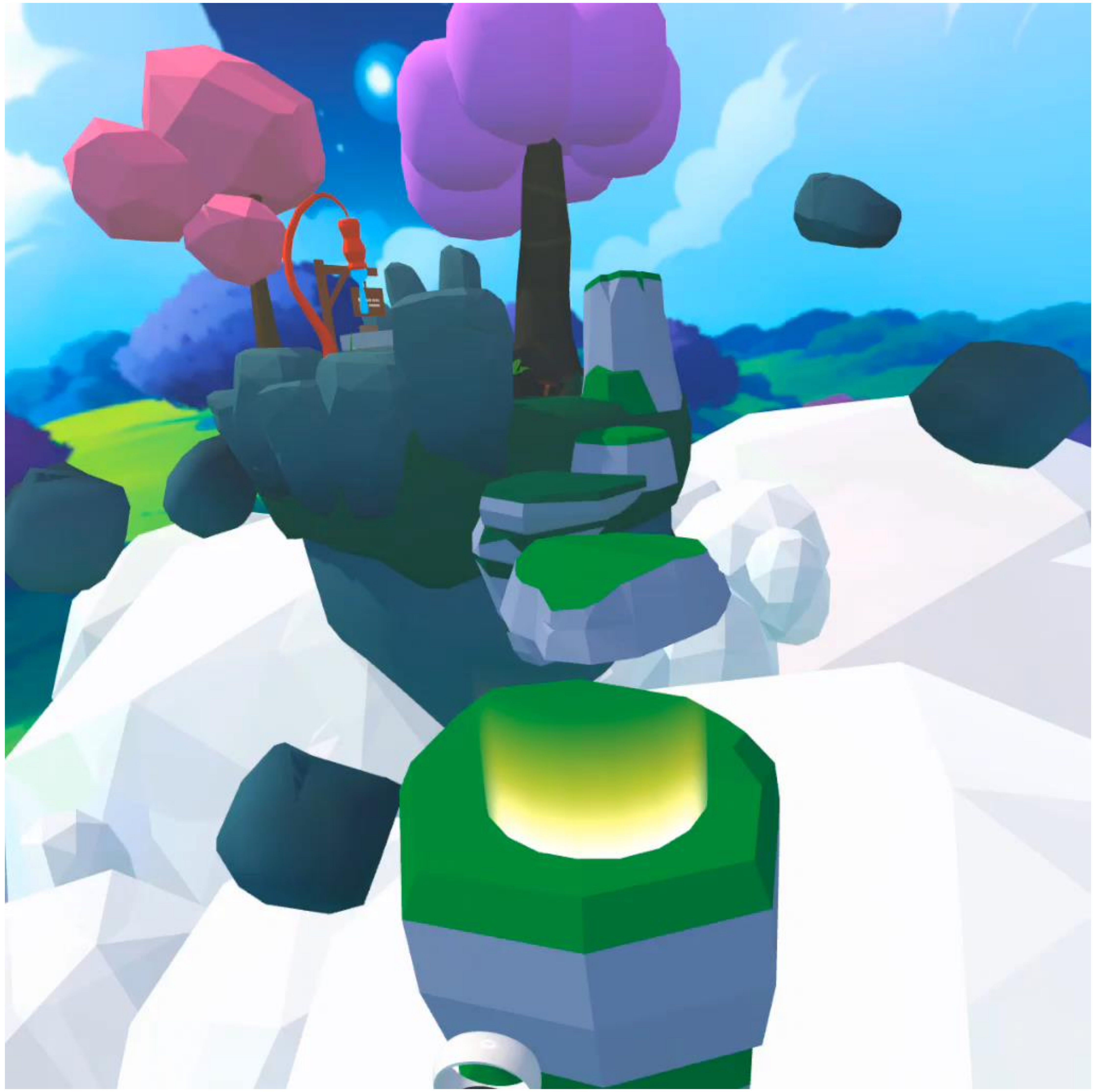
Alternative locomotion: jumping

→ Triggered by swinging the arms

Need the unity player to be aligned with the actual player

→ virtual cylinder as a virtual torso to help the player staying aligned with it

Tutorial example



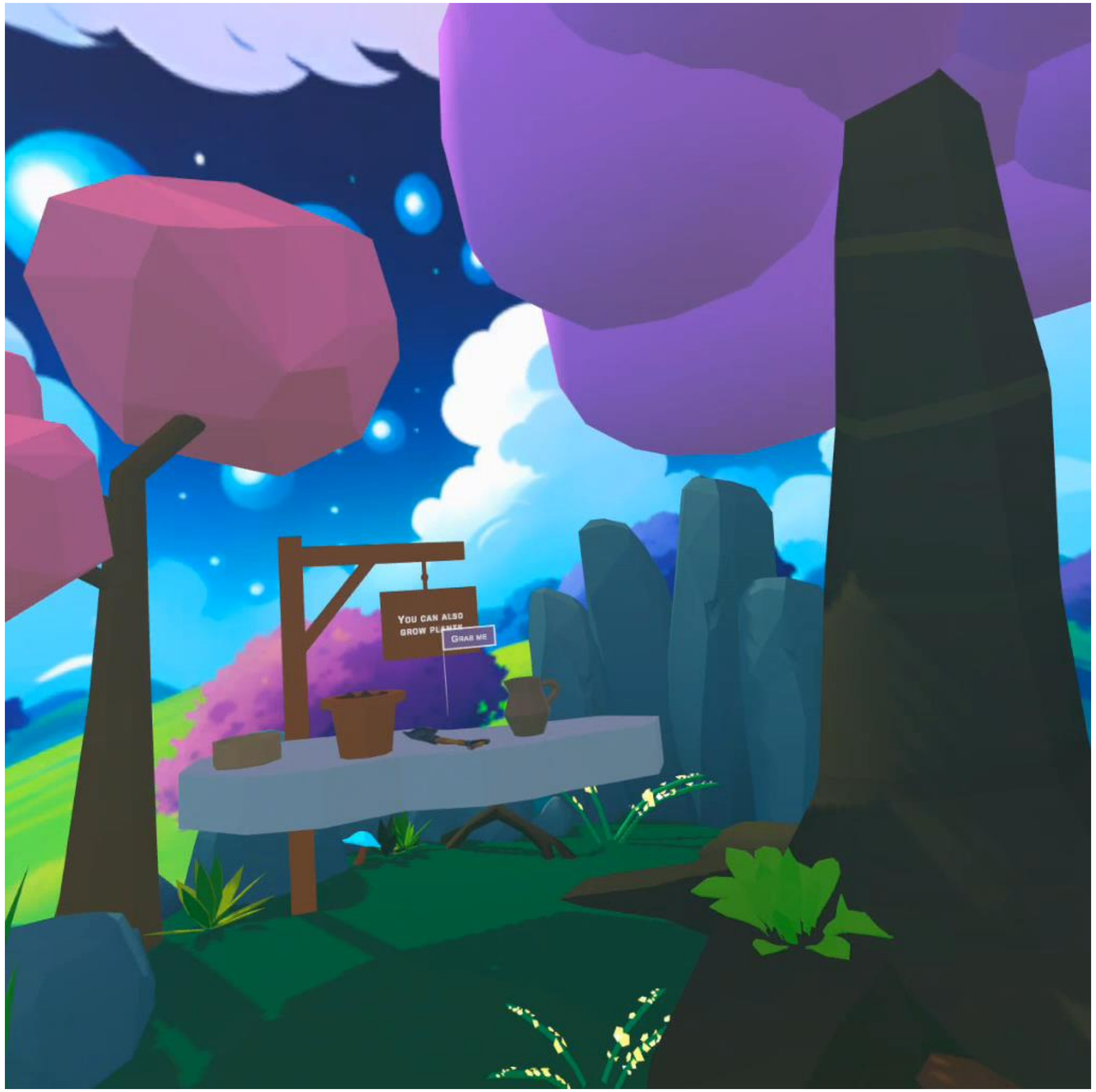
Alternative locomotion: jumping

→ Triggered by swinging the arms

- ❑ **Visual feedback:**
  - Marker on the targeted location (determined by HMD orientation) → similar to the Teleportation marker
  - Black screen fade on trigger to smooth the transition
- ❑ **Audio Feedback:** « woosh » sound on trigger
- ❑ **Haptic feedback:** on trigger, on both controllers

Tutorial example

Gardening activity: tooltips to guide the user step by step

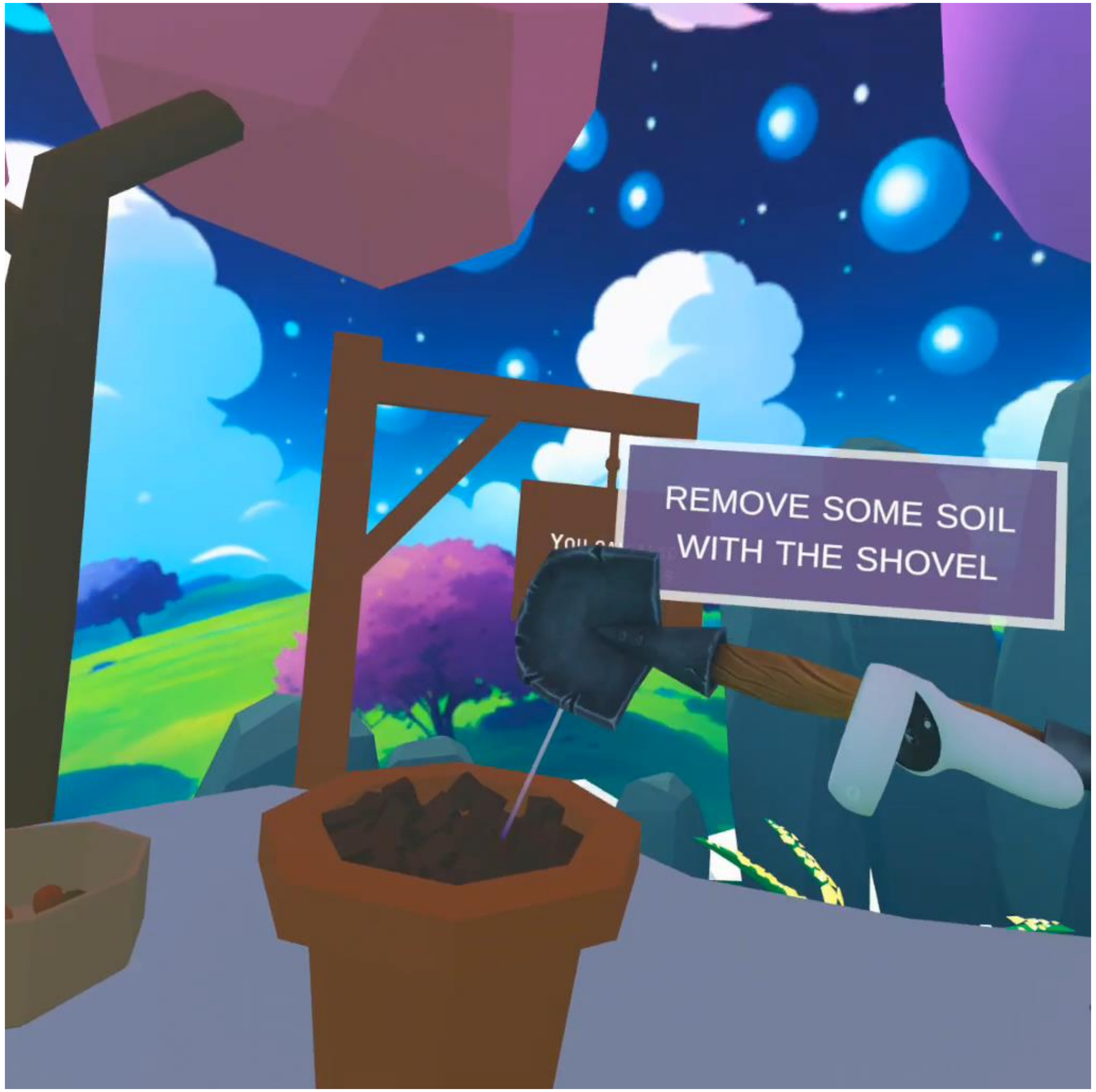


Hit and Swing: depends on the applied force

- ❑ **Visual feedback:** top soil disappearing when hit by the shovel with a sufficient amount of force
- ❑ **Audio Feedback:**
  - « ground-like » sound on hit
  - Volume depends on the force applied
- ❑ **Haptic feedback:** on the hand holding the shovel, when the force is sufficient

Tutorial example

Gardening activity: tooltips to guide the user step by step



Shovel: Container variation

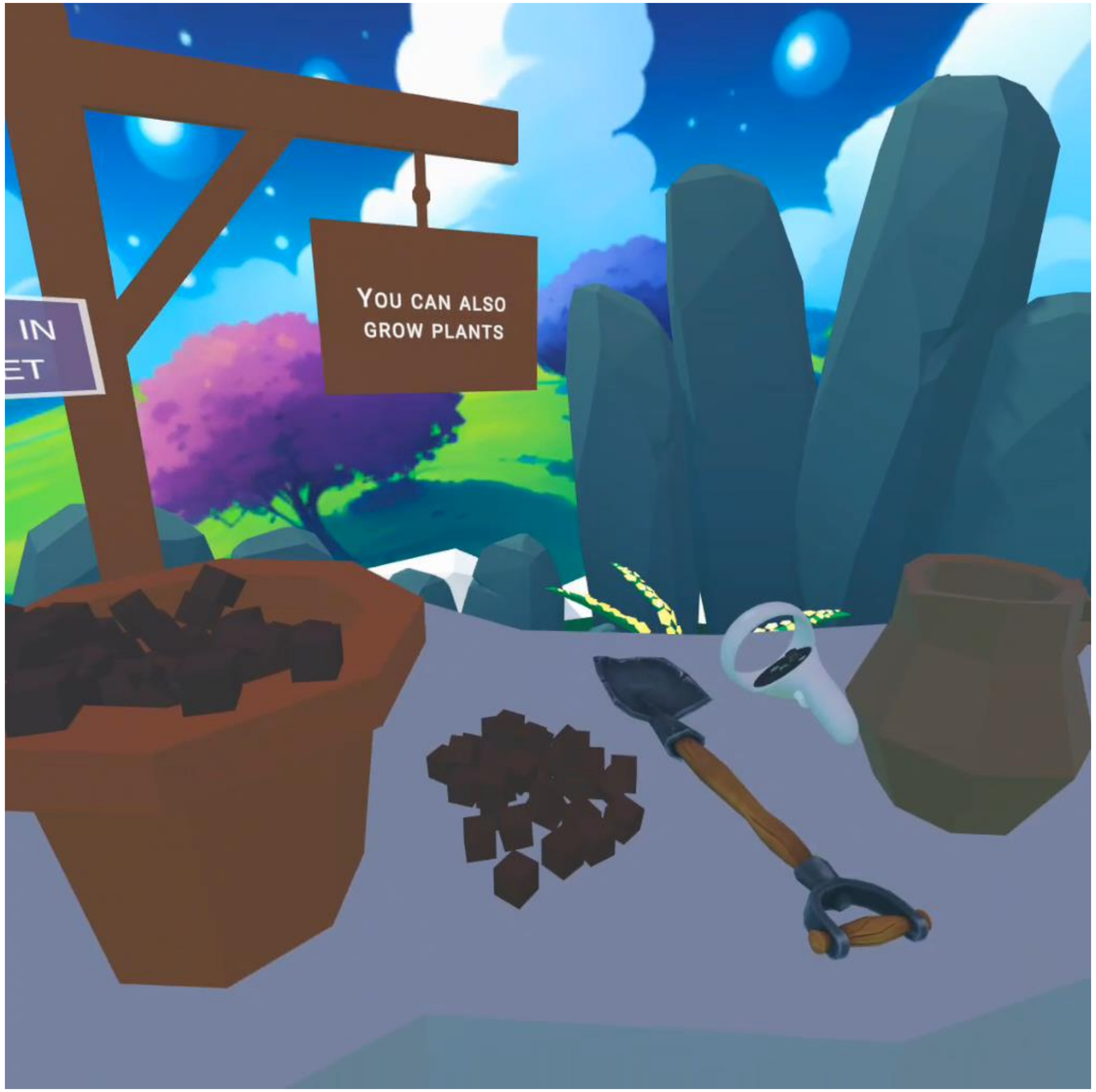
→ Allows to dig in the soil

→ Behavior depends on the orientation of the shovel

→ Allows to push soil cubes in the bucket

Tutorial example

Gardening activity: tooltips to guide the user step by step



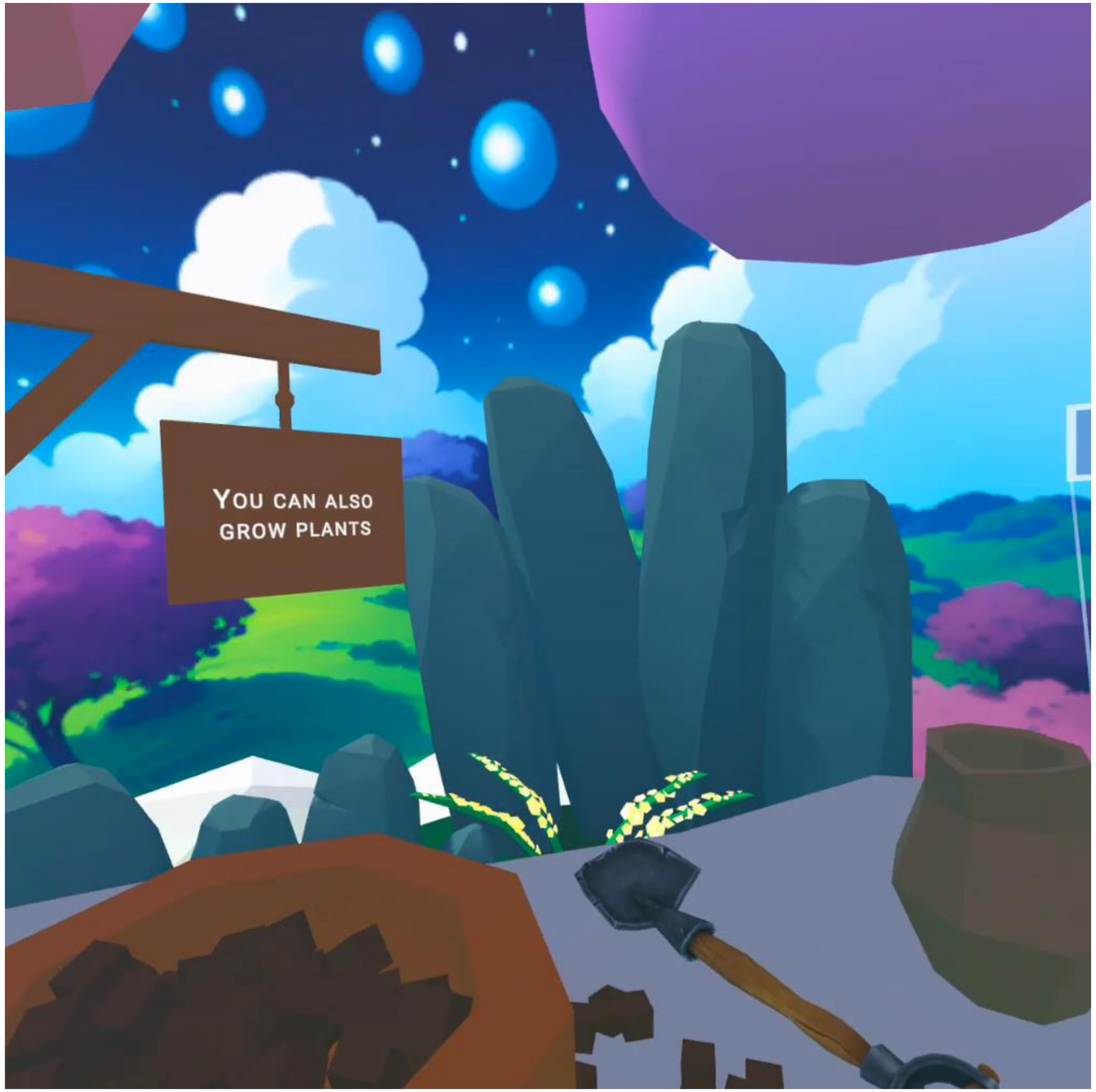
Planting a seed & covering it with soil:

- ❑ The bucket detects when a seed is placed
- ❑ The seed must be covered by enough of soil to be considered « planted »
- ❑ Each seed produce a different plant

  (plant generation system allowing to combine several types of bushes/leaves/twigs with different types of fruits/flowers/mutations)

Tutorial example

Gardening activity: tooltips to guide the user step by step



Pitcher: variation of a container

Starts empty → needs to be filled → **Water plant mecanism**

→ **Water plant detects when the pitcher is close**

❑ **Visual feedback:** pitcher filled with water cubes, quantity of water depends on the time spent under the water plant

❑ **Audio Feedback:**

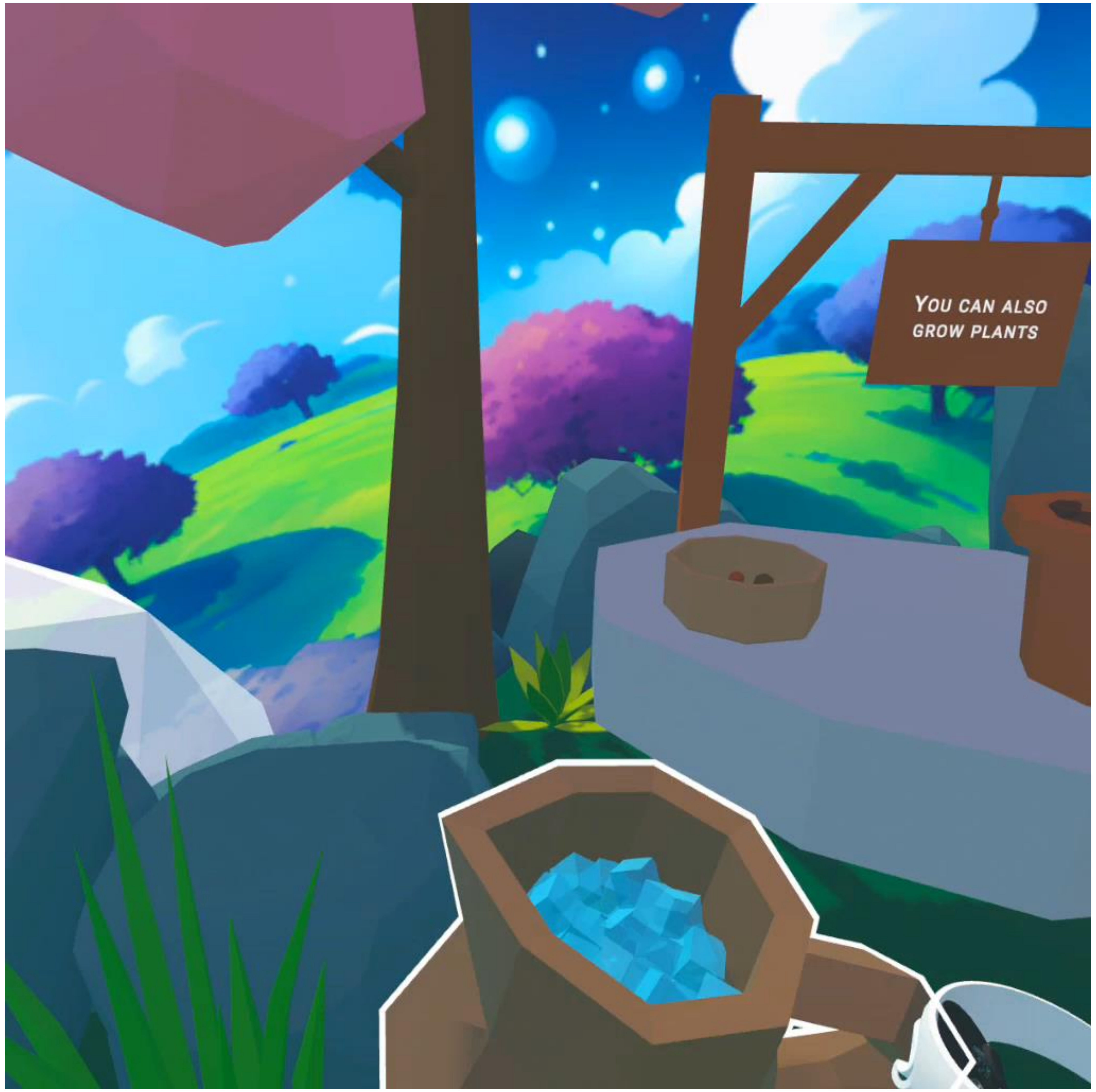
- water sound on refill
- No sound if the pitcher is already full

❑ **Haptic feedback:**

- Vibration on the hand holding the pitcher
- No vibration if the pitcher is full

Tutorial example

Gardening activity: tooltips to guide the user step by step



Pitcher: variation of a container

Water is poured by the pitcher depending on it’s orientation, grabbed state, and amount of water contained

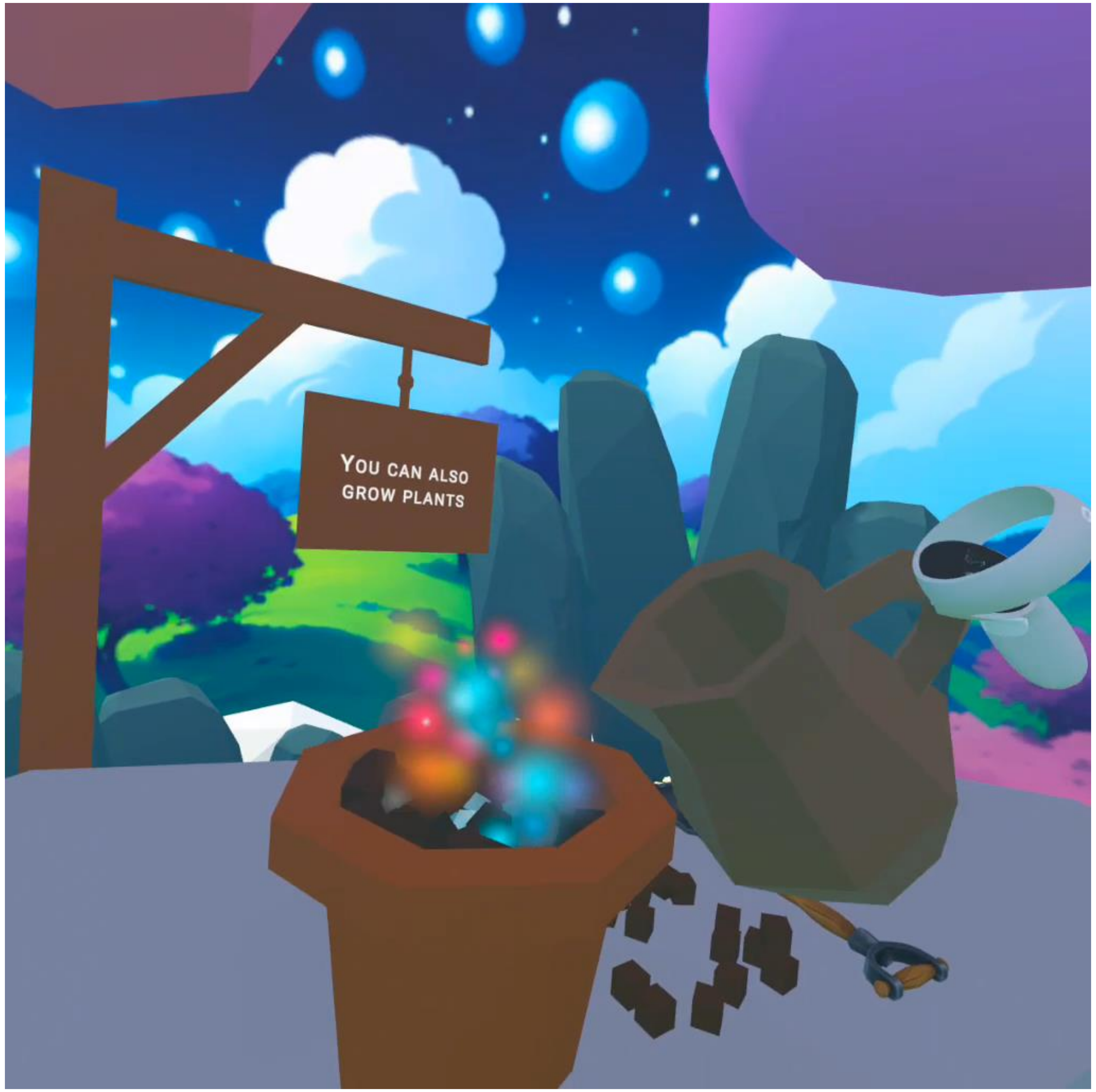
❑ **Visual feedback:** water cubes are poured

- Blue texture fading when colliding with soil cubes
- Soil cubes darkening when colliding with water
- quantity of water in the pitcher depends on the amount of water poured

❑ **Audio Feedback:** water sound

Tutorial example

Gardening activity: tooltips to guide the user step by step



Plant growth

- ❑ **Visual feedback:** particles when enough water has been poured in the bucket
- ❑ **Audio Feedback:** magic sound

- Resulting plant depends on the seed parameters
- Can be harvested
- Bucket can be used to plant another seed (soil renewed automatically to avoid empty bucket)

Clear end of tutorial (additional « play » button in the final build to start the main game)

➔ Feedback is the key for the project

Project Deadlines

- ❑ Project Pitch (18.03): 3min followed by 5min Q&A

  → Present project pitch to the lecturers and tutors before fully committing to an idea. This allows students to obtain feedback on their idea, its feasibility and if it goes inline with the project guidelines. **All groups should have their ideas ready and rehearsed so that it is easily communicated during this session.**

- ❑ Q&A Sessions during the “Project” slots in the planning
- ❑ Basic Game interactions peer assessment (15.04): ~10 min (to be confirmed)
- ❑ Project Playtesting with peer assessment (13.05): ~10 min (to be confirmed)
- ❑ **Delivery of the Game Executable + Report + Video: 22.05**
- ❑ Project Oral Presentations (twice): week 27-31.05 ; 15min per group + 5min per person

+ Paper study and theoretical exam to validate the course

→ Do not underestimate the amount of work needed

- Organisation for a group-project (oral exams: all group members must know how the whole project works, not only what they personally developped)
- New software with a specific way of handling inputs in a frame-by-frame way
- New language to learn (C#)
- Designing well-done interactions takes time
- Debugging can be very time-consuming
- Building a nice 3D environment takes time, even with premade assets - The soundscape and the light matter for the game atmosphere
- Building a nice scenario takes time
- Need for user feedbacks to iterate

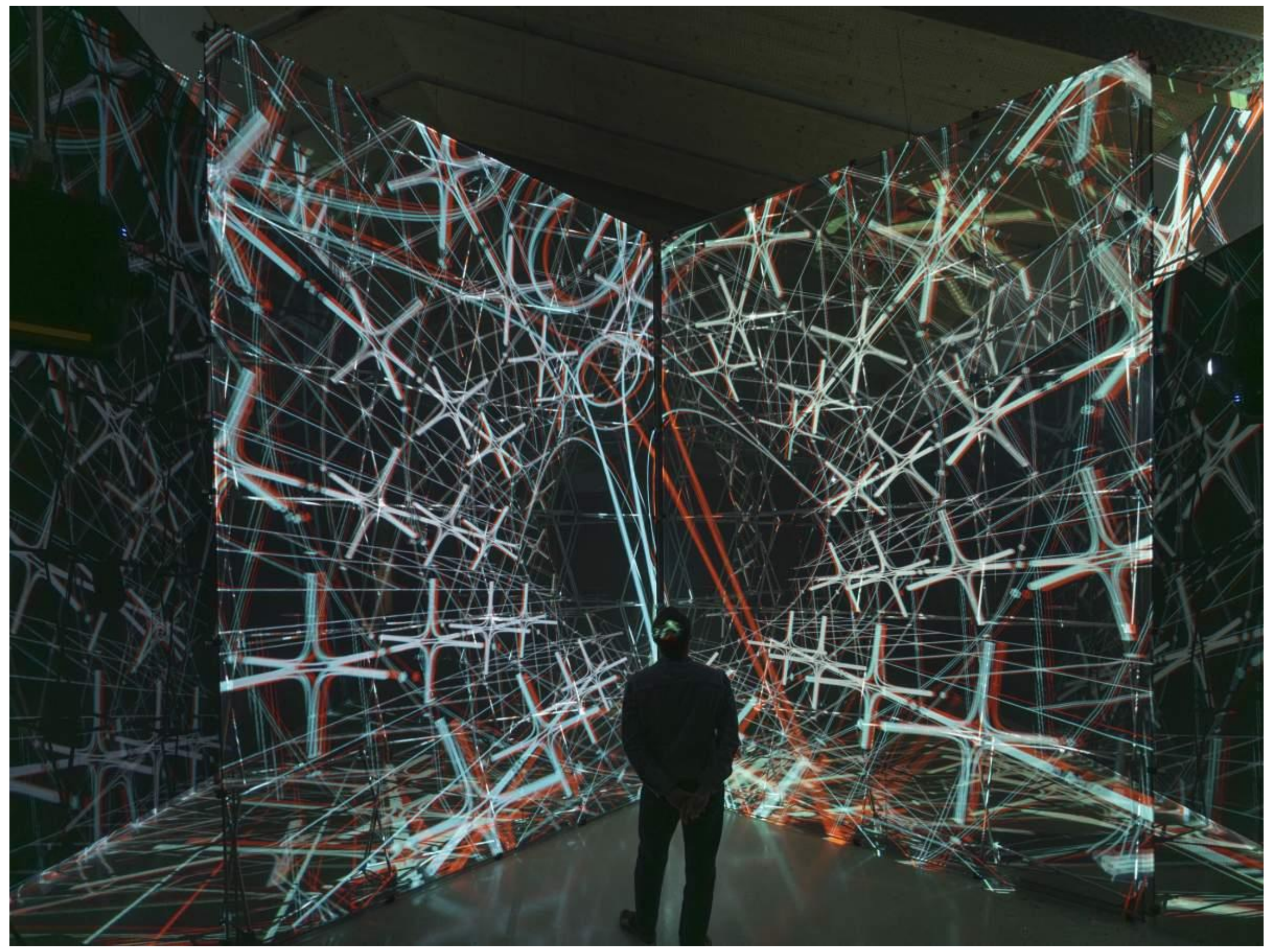
You can start installing the project environment with the Hands-On HO1-a

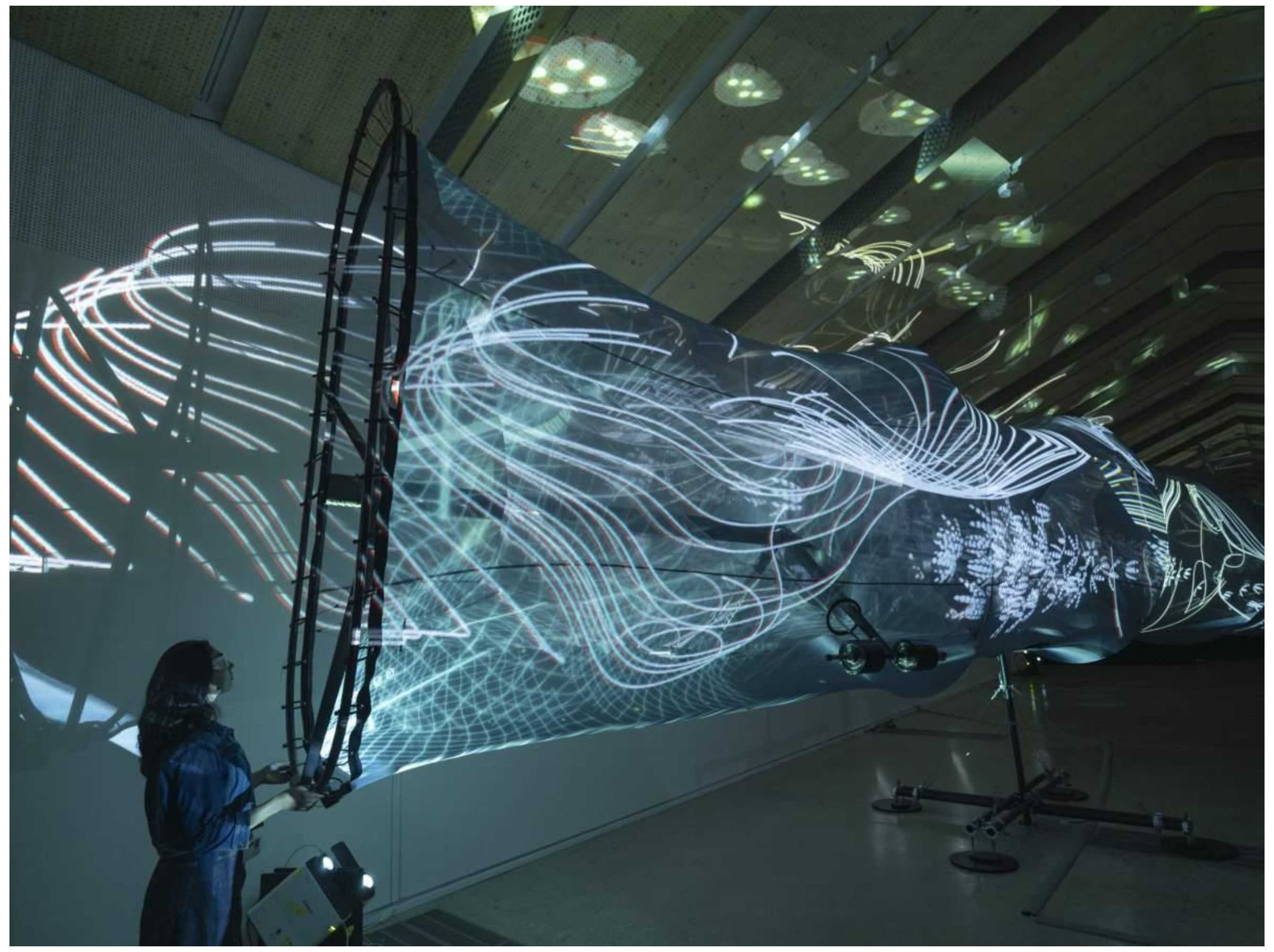
A nice exhibition to visit in link with virtual illusions

Cyber Physical : Architecture in Real Time

Exhibition with 3D glasses **Free Entry**

"At the intersection of art, technology and construction, Cyber Physical: Architecture in Real Time brings together four dynamic and immersive installations that reshape our perception of space by transforming virtual data into tangible and interactive kinetic sculptures. Through 3D projections, augmented reality and artificial intelligence, the exhibition allows publics to interact with ever-evolving architectural structures capable of listening, learning, and reshaping themselves in real time."





Location : EPFL pavilion

Opening hours:
From Tuesday to Sunday
11 am – 6 pm
Closed on Monday

End: 16.6.2024