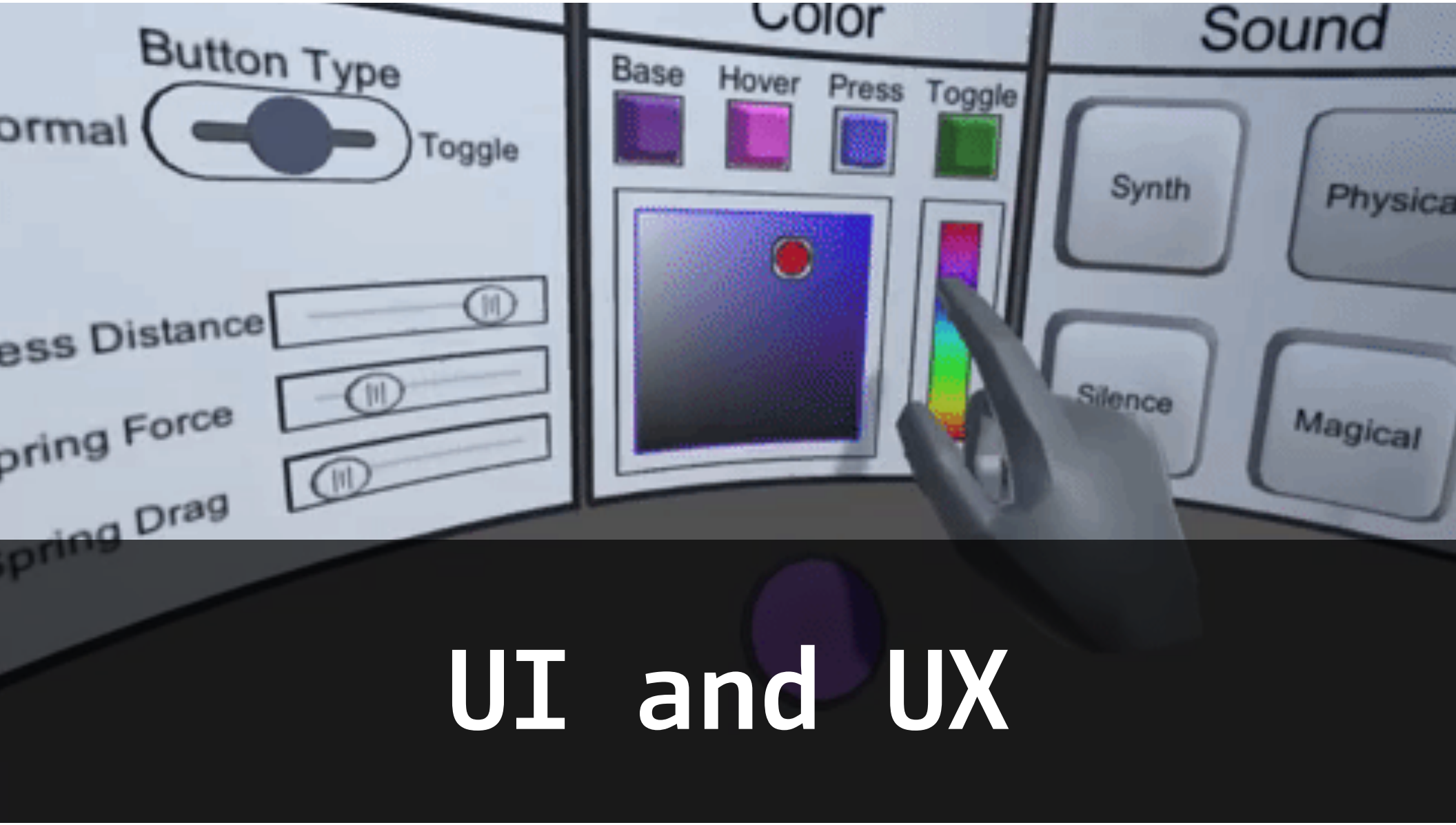
TECH 1711 - Mixed Reality Studio


MUSEUM OF
FAILURE


VIRTUAL BOY
VIRTUAL BOY


VFX1
H·E·A·D·G·E·A·R


POWER GLOVE


Button Type
ormal
Toggle
ess Distance
pring Force
pring Drag
Color
Base
Hover
Press
Toggle
Sound
Synth
Physica
Silence
Magical

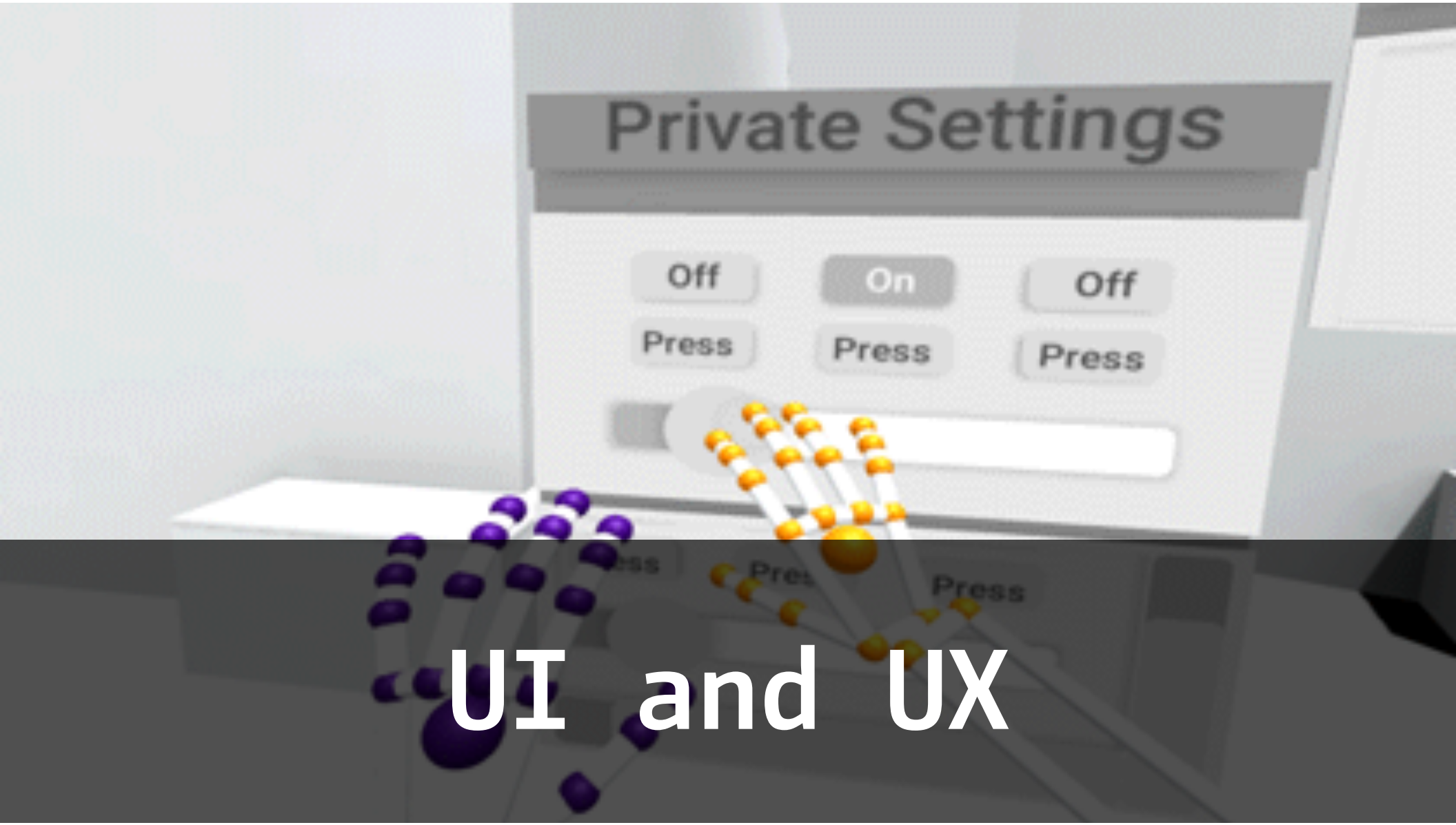
UI and UX

Manipulation

Exploration

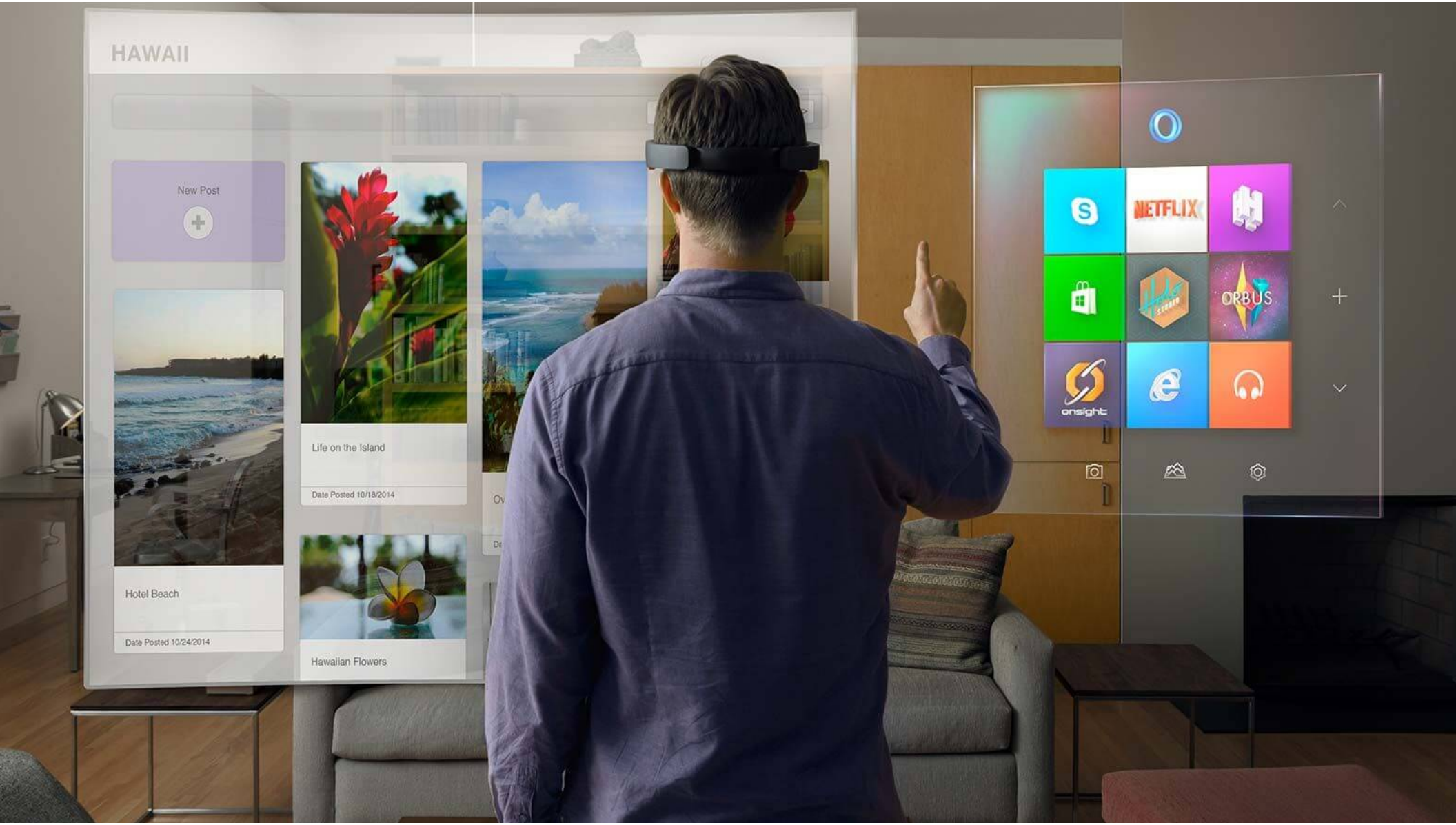

Expectations

Skeuomorphism



UI and UX


hello.


HAWAII
New Post
Hotel Beach
Date Posted 10/24/2014
Life on the Island
Date Posted 10/18/2014
Hawaiian Flowers
NETFLIX
ORBUS
onsight

Diegetic vs Non-Diegetic

Diegetic vs Non-Diegetic

Diegetic vs Non-Diegetic

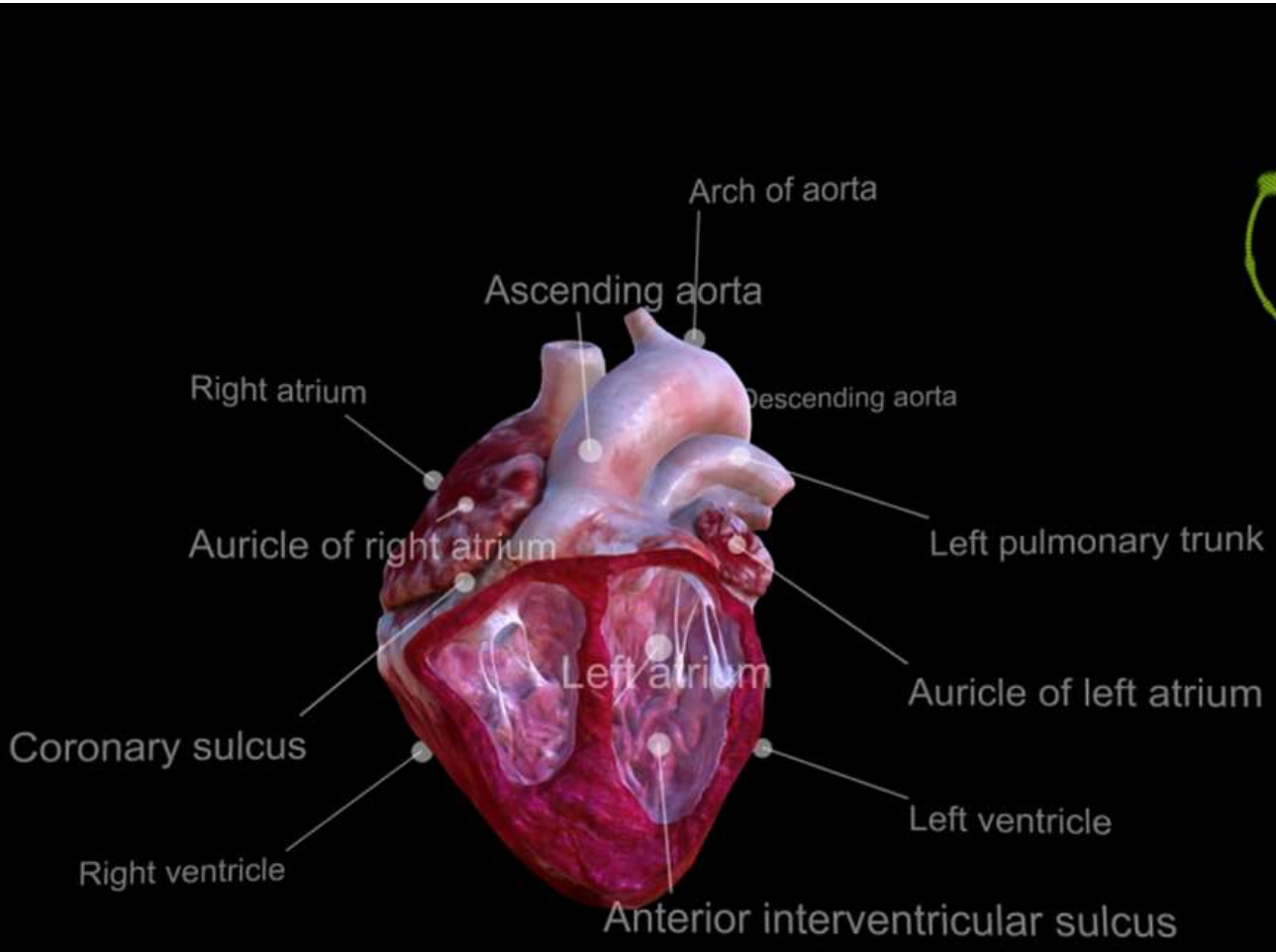

SCAN MODE 13894
SIZE ASSESSMENT
ASSESSMENT COMPLETE
FIT PROBABILITY 0.99
RESET TO ACQUISITION
MODE SPEECH LEVEL 78
PRIORITY OVERRIDE
DEFENSE SYSTEMS SET
MATCH


Ink
Brushes
Light


Smarter Objects by Valentin Heun, Shunichi Kasahara, Pattie
MIT Media Lab


SCAN MODE 13894
SIZE ASSESSMENT
ASSESSMENT COMPLETE
FIT PROBABILITY 0.99
RESET TO ACQUISITION
MODE SPEECH LEVEL 78
PRIORITY OVERRIDE
DEFENSE SYSTEMS SET
MATCH


Condition 1
Normal Heart Rate
Condition 2
Myocardial Infarction
Condition 3
Arterial Hypertension
Condition 4
Atrial Fibrillation
120:80
BP
80
BPM
Normal Heart Rate
A normal heart rate is defined by 60-100 heart beats per minute in resting adults. Bradycardia is per definition below 60 and Tachycardia above 100 beats per minute.
Gestures
Scale
Move
Rotate
Voice Commands
Reset Scene
Condition 1
Condition 2
Condition 3
Condition 4
ANIMA RES
Studio for 3D medical animation

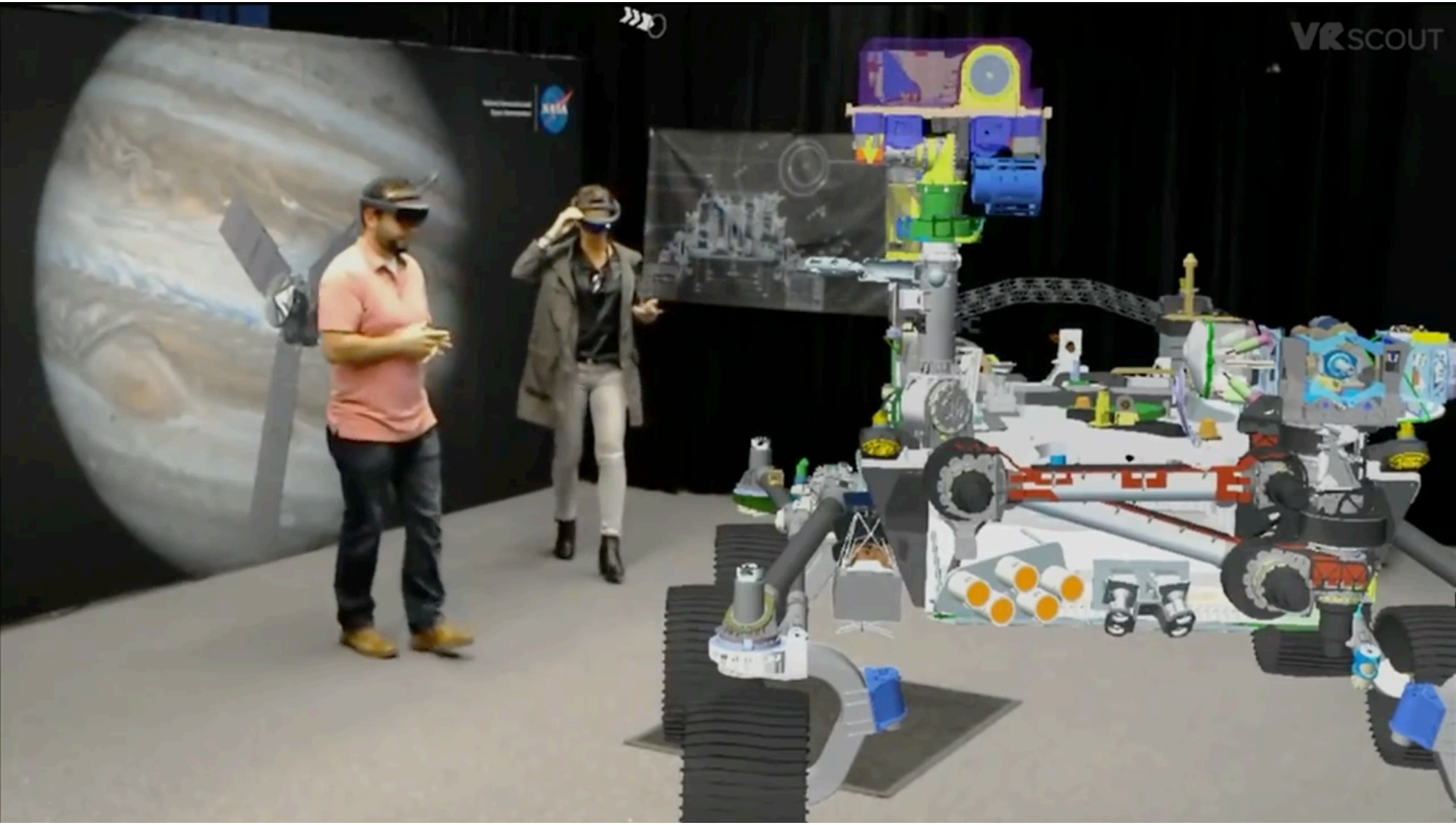


INSIGHT HEART 2.0

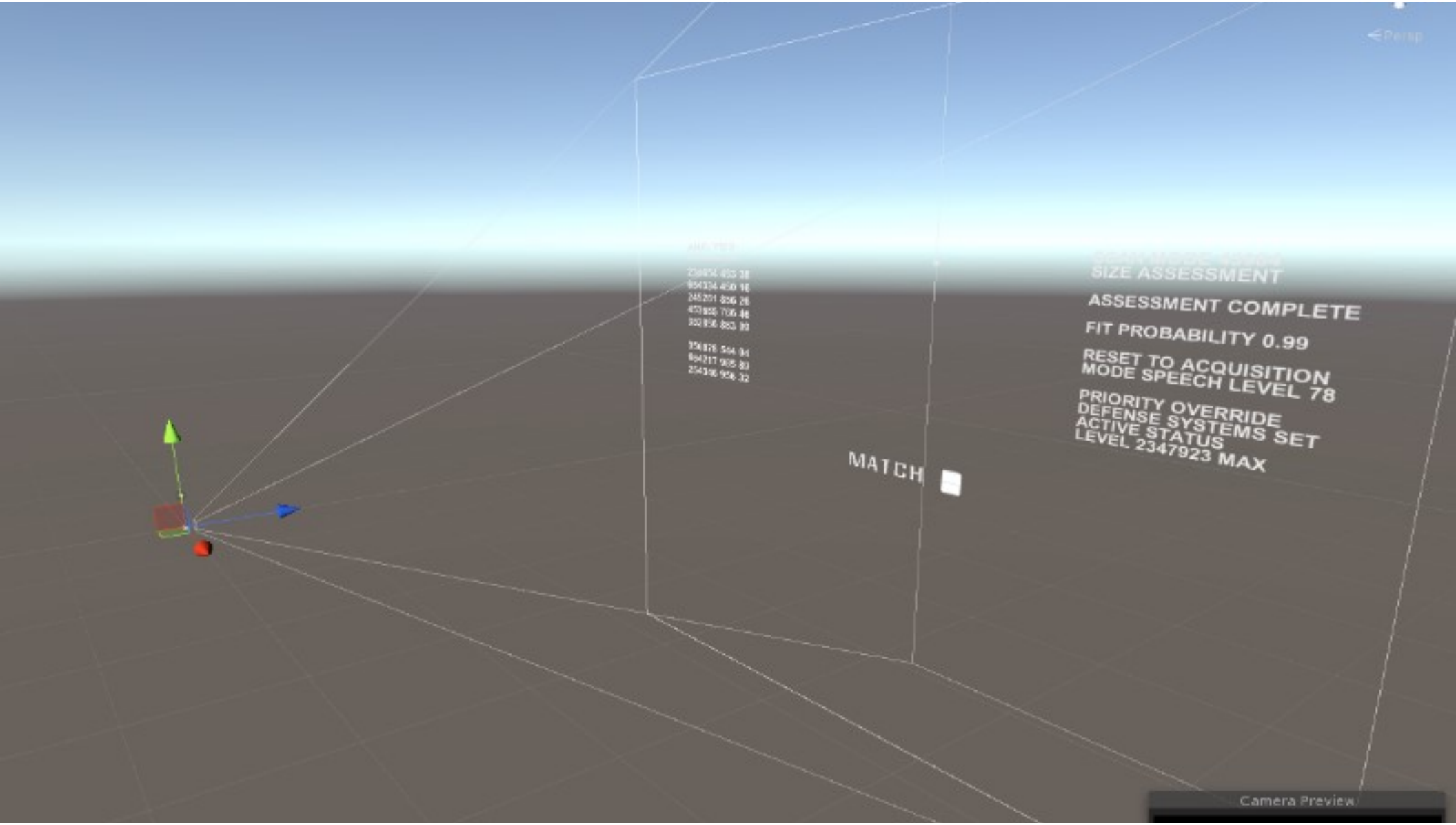
taking medical education to the next level

ANIMA RES

Studio for 3D medical animation


VR SCOUT


SCAN MODE 13894
SIZE ASSESSMENT
ASSESSMENT COMPLETE
FIT PROBABILITY 0.99
RESET TO ACQUISITION
MODE SPEECH LEVEL 78
PRIORITY OVERRIDE
DEFENSE SYSTEMS SET
MATCH


SIZE ASSESSMENT
ASSESSMENT COMPLETE
FIT PROBABILITY 0.99
RESET TO ACQUISITION
MODE SPEECH LEVEL 78
PRIORITY OVERRIDE
DEFENSE SYSTEMS SET
ACTIVE STATUS
LEVEL 2347923 MAX
MATCH
Camera Preview


ANALYSIS:
****
234654 453 38
654334 450 16
245261 856 26
453665 766 46
382856 863 09
356878 544 04
664217 985 89
254346 956 32
SCAN MODE 43984
SIZE ASSESSMENT
ASSESSMENT COMPLETE
FIT PROBABILITY 0.99
RESET TO ACQUISITION
MODE SPEECH LEVEL 78
PRIORITY OVERRIDE
DEFENSE SYSTEMS SET
ACTIVE STATUS
LEVEL 2347923 MAX
MATCH


38
16
26
46
09
04
89
32
SCAN MODE 43984
SIZE ASSESSMENT
ASSESSMENT COMPLETE
FIT PROBABILITY 0.99
RESET TO ACQUISITION
MODE SPEECH LEVEL 78
PRIORITY OVERRIDE
DEFENSE SYSTEMS SET
ACTIVE STATUS
LEVEL 2347923 MAX
MATCH

VR Sickness

WEARABLE TECH

The dangers of virtual reality

Commentary: Tripping over wires, accidental TV breakage and nausea. VR could get you hurt. Better to be prepared.

BY **SCOTT STEIN** / MARCH 29, 2016 2:59 PM PDT

How the battle to stop VR sickness will change game development forever

By Louise Blain October 14, 2016 News

COMMENTS

NEWS REVIEWS GAMING JOBS TALENT CLASSES

CATEGORY: EDITORIALS FOLLO

7 Things You Can Do to Overcome VR Motion Sickness

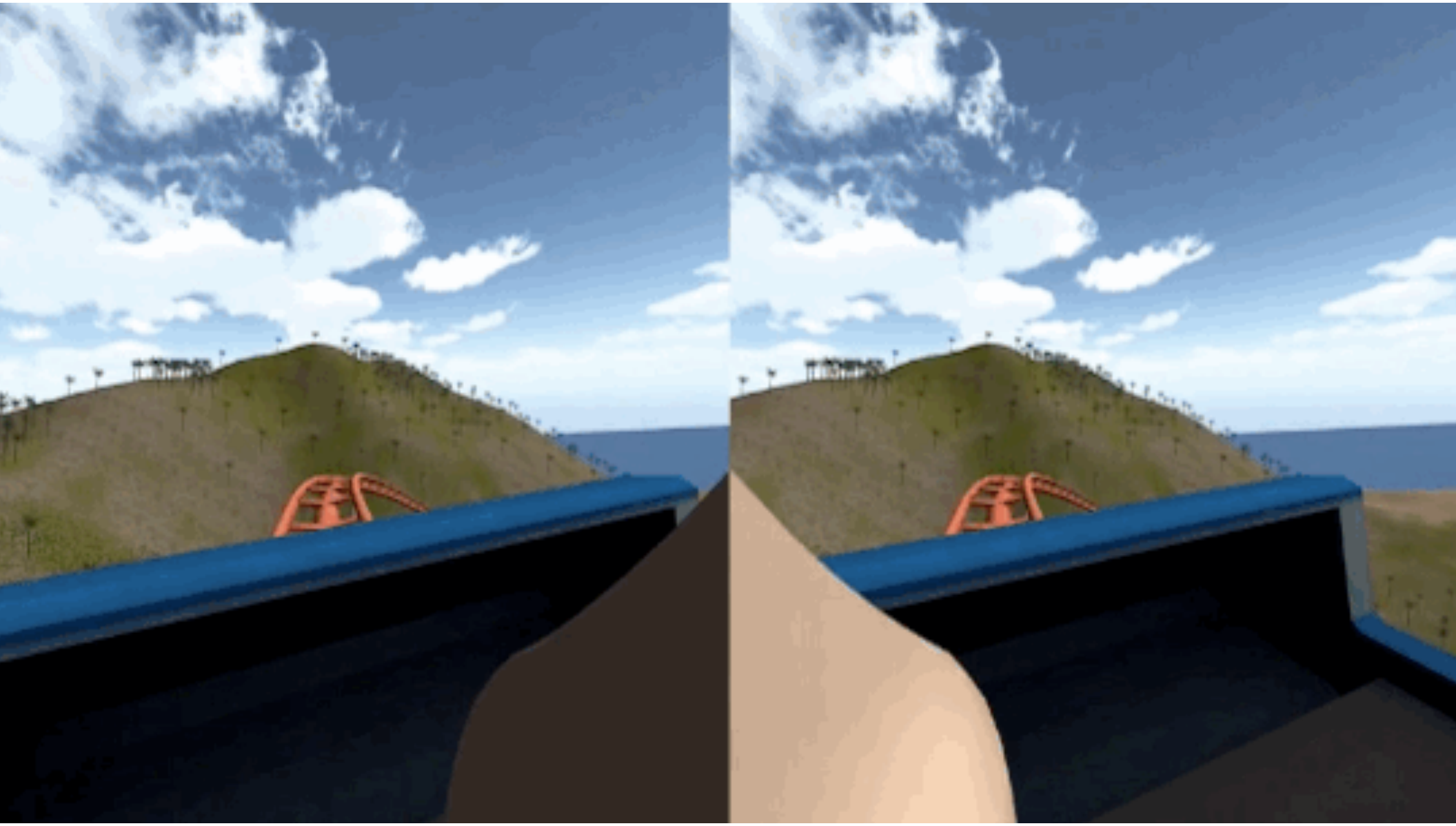

STEAM
STORE COMMUNITY IVAYLO ABOUT SUPPORT
HTC Vive
All Discussions Artwork Videos News Guides Reviews
HTC Vive > General Discussions > Topic Details
Litva Mar 16, 2016 @ 9:53am
Do you feel less motion sick in Rift then Vive ?
I`m planing to play mainly games like War Thunder or other 1st person view games, mainly Simulator games. This motion sickness they do mention more and more about feels like maby its to mutch and best you can do just play this simple games. Teleporting system in game sounds like not the best experiance but keeps you ok in VR. So how will be rift played with joystick if you get motion sick ?

HEALTH

Scientists Think They've Found A Way To Eliminate Virtual Reality Sickness

I'm nauseous... I'm nauseous...

By Ryan F. Mandelbaum *June 27, 2016*



33

DK1 made me very VR sick. DK2 does not self.oculus

submitted 3 years ago * by Dawiiz

I have a DK1 and have tried to push my self through many VR demos. I always ended up feeling sick, some times for days. Today I got my DK2 and I had no problems with motion sickness. Will try playing for some hours and give more feedback :)

Update: After some hours of gaming I finally got motion sick. But recovery was super quick. DK1 left me sick for a long while.

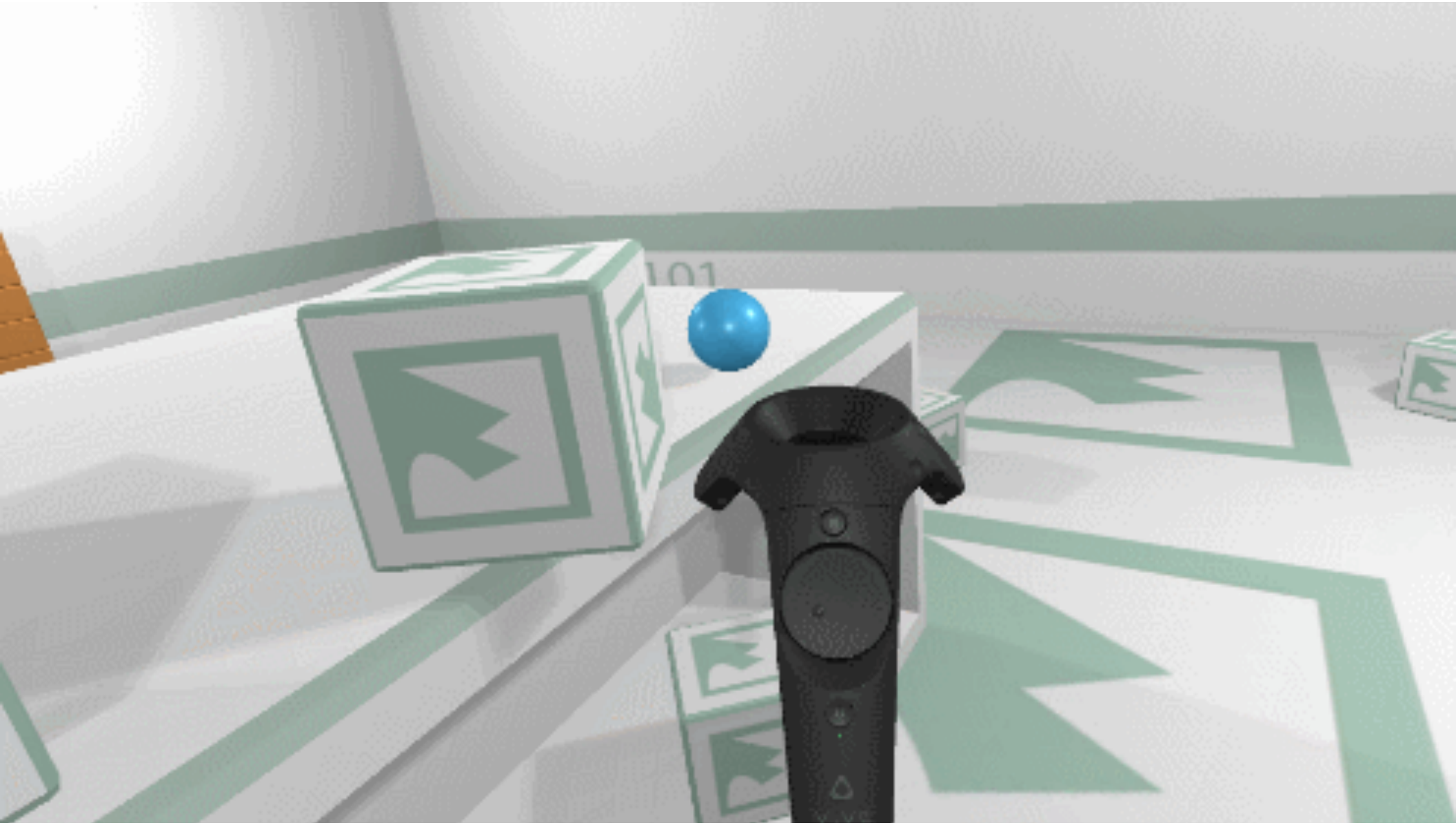
32 comments share save hide give gold report pocket

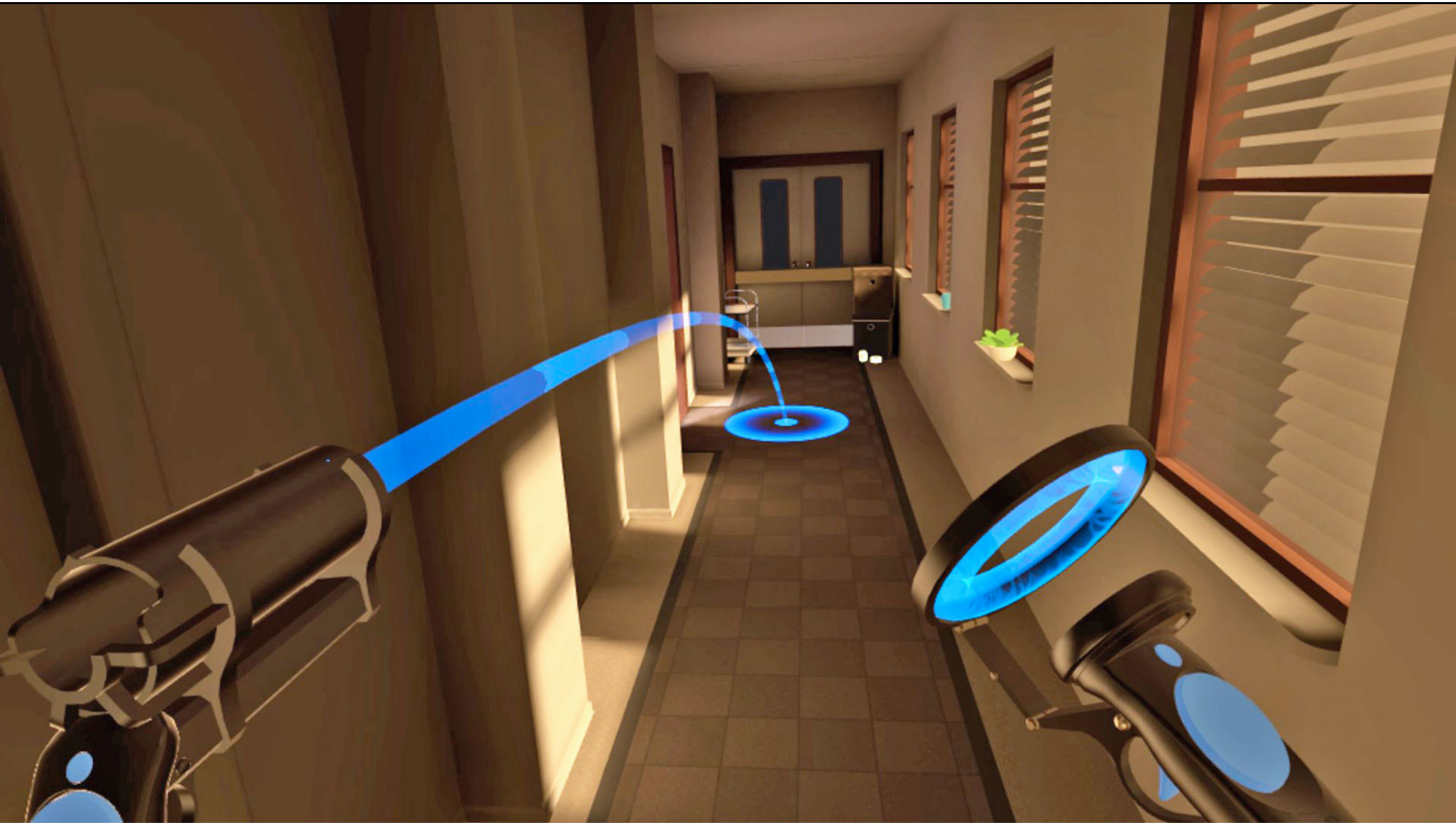

Oculus

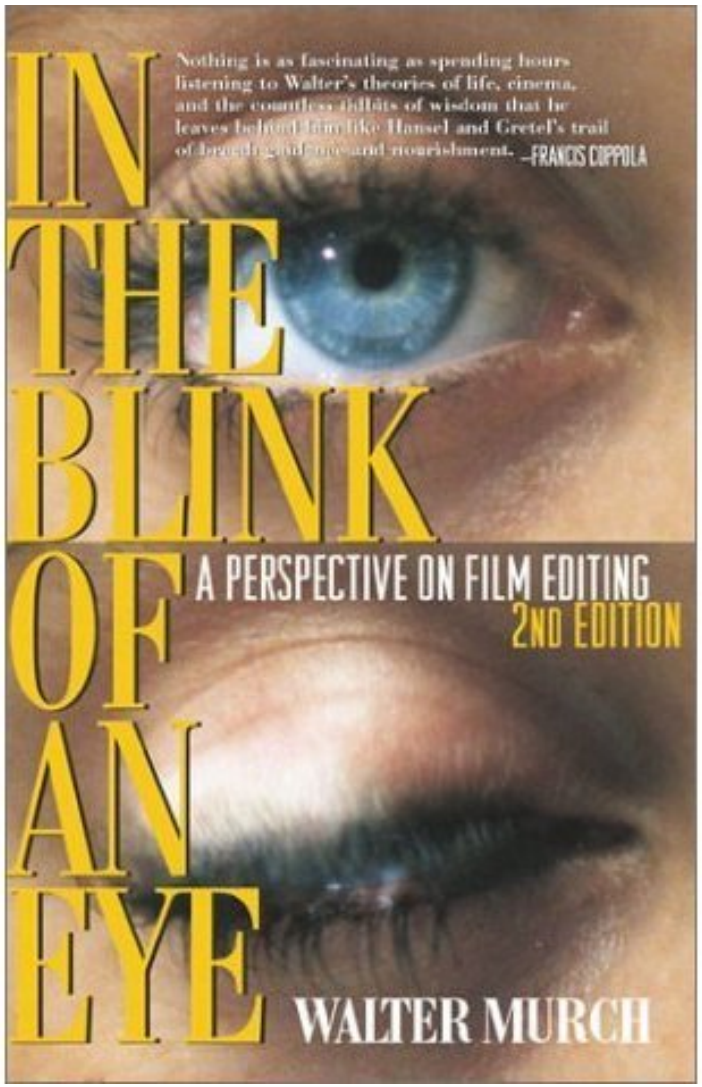

Oculus


positional tracking


TARGET SCAN
BOOST
089
FUEL
SHIELD
HULL
EMP
03/08
231
135 m
ROCKFISH
GAMES

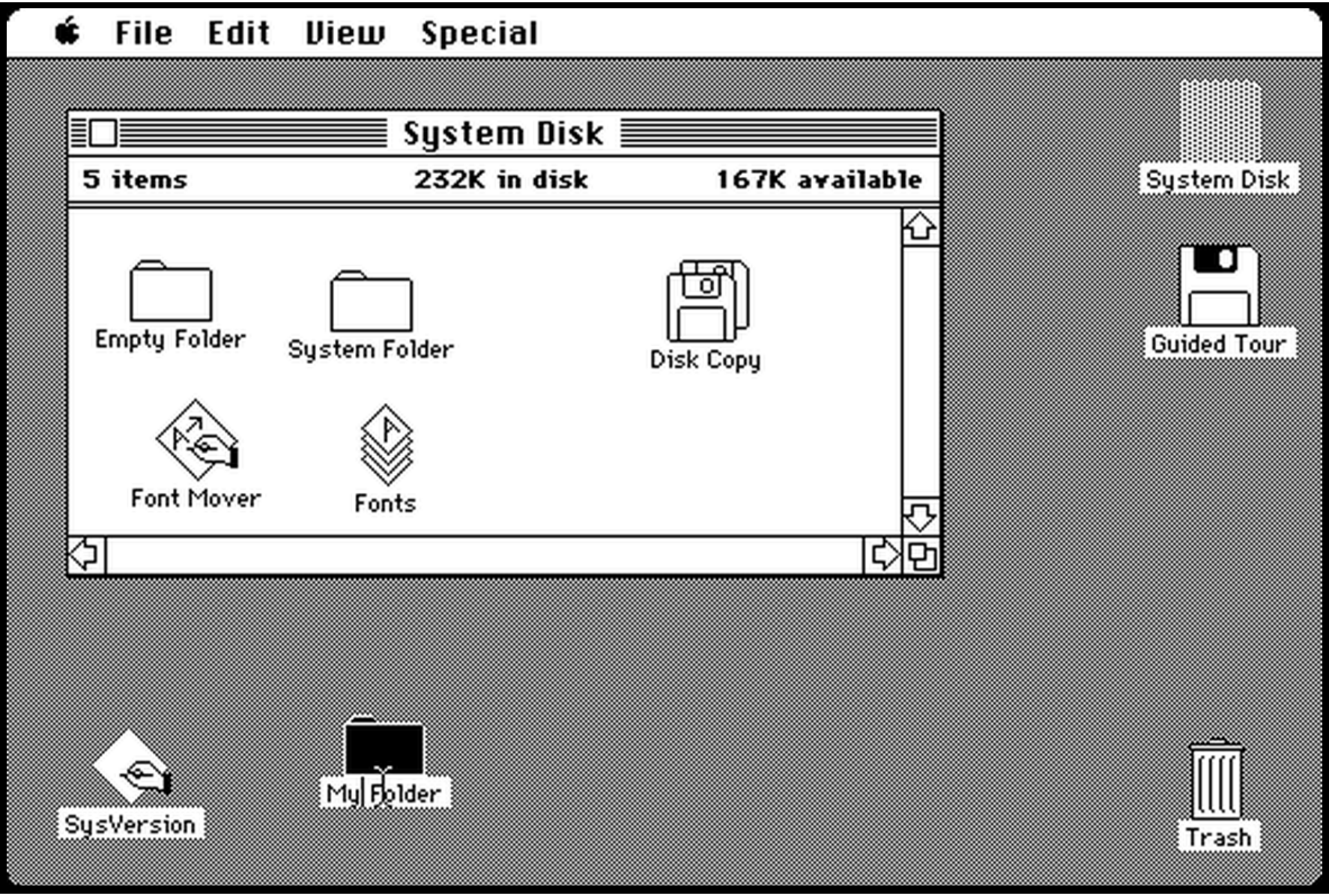





Nothing is as fascinating as spending hours listening to Walter's theories of life, cinema, and the countless tidbits of wisdom that he leaves behind him like Hansel and Gretel's trail of breadcrumbs: nice and nourishment. –FRANCIS COPPOLA
IN THE BLINK OF AN EYE
A PERSPECTIVE ON FILM EDITING
2ND EDITION
WALTER MURCH

“So it seems to me that our rate of ***blinking is somehow geared more to our emotional state and to the nature and frequency of our thoughts than to the atmospheric environment*** we happen to find ourselves in.”

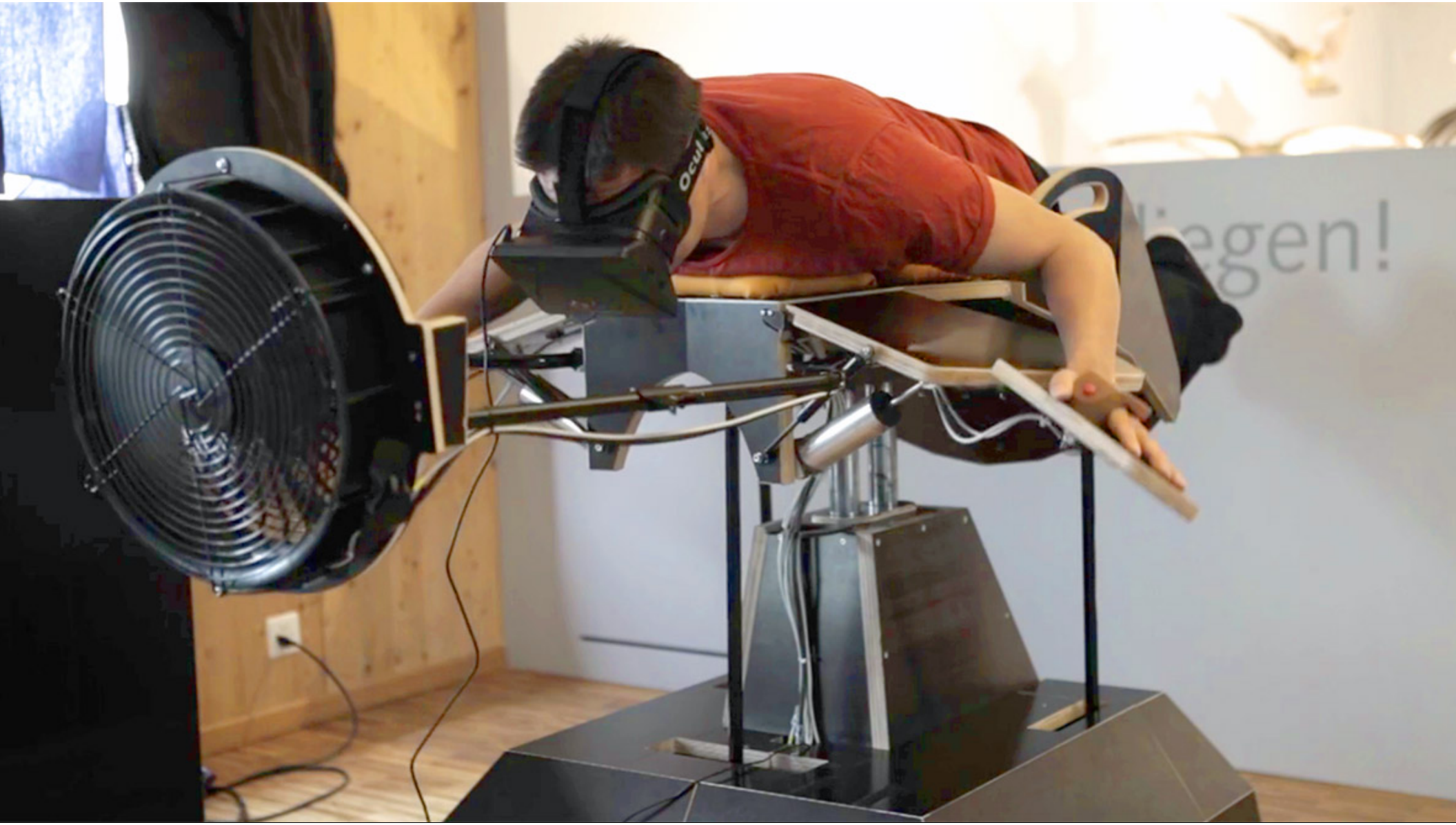
Modality

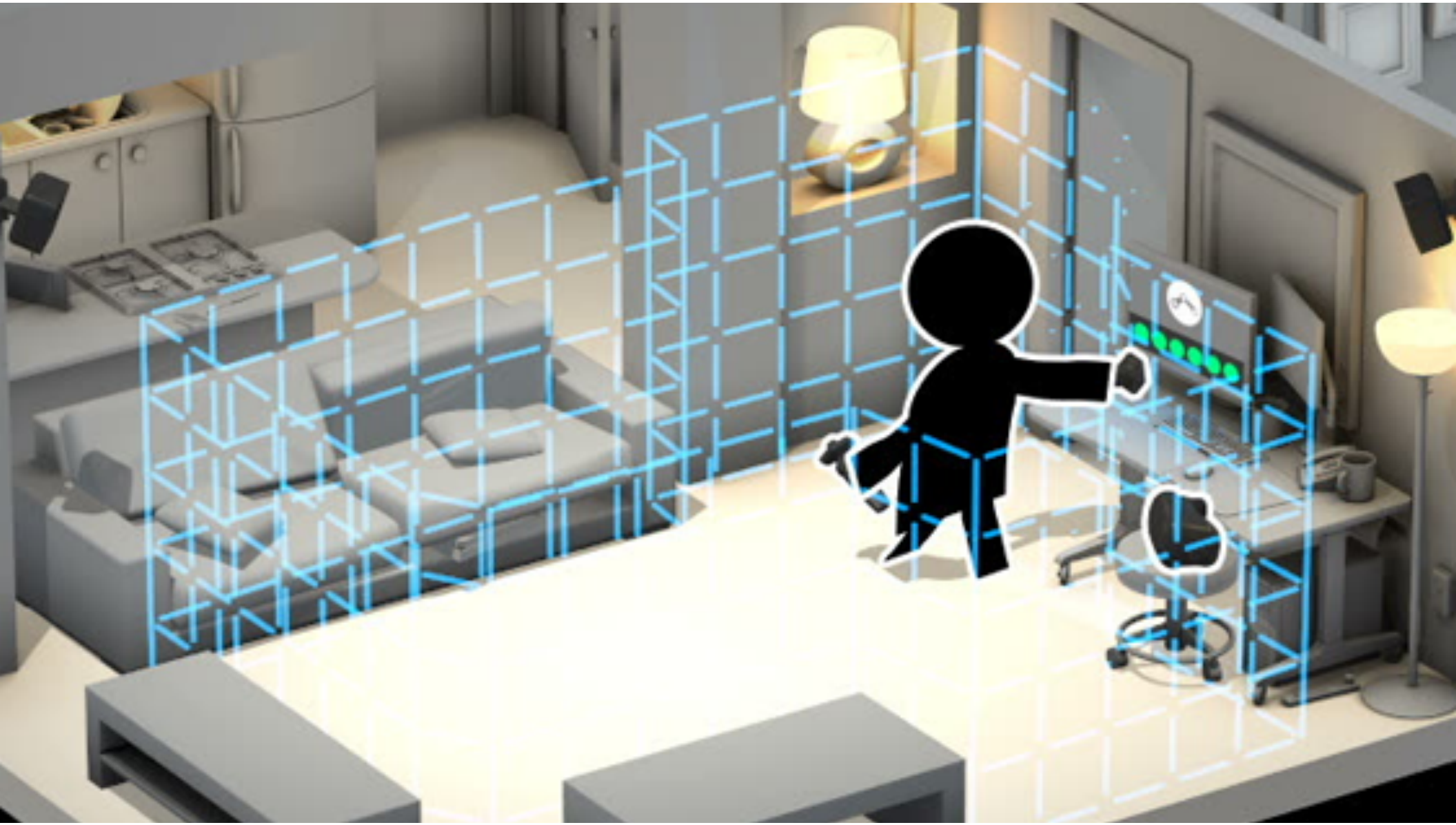

File Edit View Special
System Disk
5 items
232K in disk
167K available
Empty Folder
System Folder
Disk Copy
Font Mover
Fonts
System Disk
Guided Tour
SysVersion
My Folder
Trash

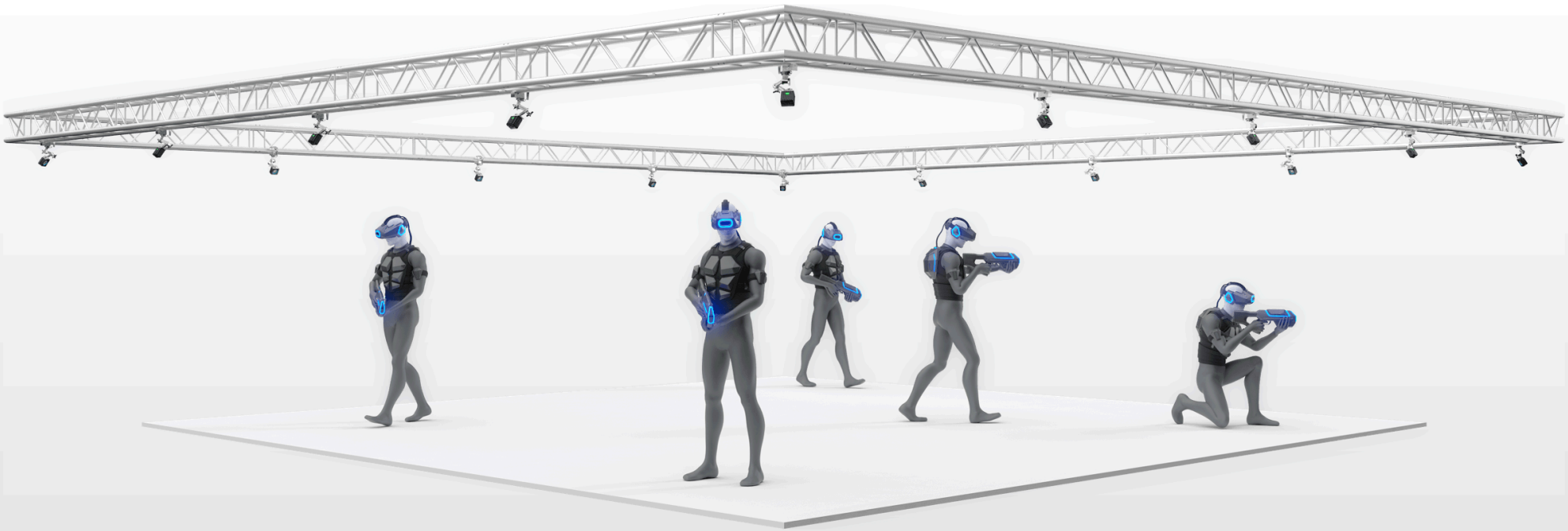

bigscreen


NETFLIX
MARCO POLO
BLOODLINE
ORANGE IS THE NEW BLACK
NETFLIX


reaaaallly cute








unity
C#

Vectors

In programming terms, you can think of Vectors as a way to store 2, 3, or 4 values in one easy-to-use package:

```
Vector2 someNumbers = new Vector2(1.0, 2.2);
Vector3 someOtherNumbers = new Vector3(5.3, 2.6, 12.0);
Vector4 evenMoreNumbers = new Vector4(7.4, 2.1, 12.0, 9.8);
```

Vectors

We can use vectors to:

- Store multiple numbers in one variable
- Describe the position of something in our world
  - For example: (2.1, 8.9, 7.4) represents the point in space 2.1 units along the X-axis, 8.9 units along the Y-axis, and 7.4 units along the Z-axis.

Vectors

We can use vectors to:

- Describe a direction
  - For example: (0.0, 1.0, 0.0) represents a point 1 unit directly above (along Y) the origin.
  - If we a drew an arrow from the origin to this point, it would point straight up.
  - It doesn't matter how long the Vector is:
    - (0.0, 1.0, 0.0) and (0.0, 5.2, 0.0) are different points, but they both describe the same *direction* (straight up).

Vectors

Unity has some built-in direction shorthands:

```
Vector3 example = Vector3.up;
```

is the same as:

```
Vector3 example = new Vector3(0.0, 1.0, 0.0);
```

Vectors

Other shorthands:

- `Vector3.up` (pointing along Y-axis)
- `Vector3.forward` (pointing along Z-axis)
- `Vector3.right` (pointing along X-axis)
- `Vector3.one` (Equal to (1.0, 1.0, 1.0)

RayCasting

RayCasting is when we shoot an invisible line into our scene to see if we hit something in that direction.

To understand RayCasting, you must understand **Vectors**.

RayCasting

`Physics.Raycast()` is a function built in to Unity. There are many, many different forms it can take. Here is the easiest:

```
Physics.Raycast(Vector3 originOfTheRay, Vector3 directionOfTheRay);
```

All this function actually does is return `true` or `false` to answer “did this Ray hit anything?”

RayCasting

To store information about *what* was hit, and more importantly *where* the hit is in space, we have to do two things:

1. Declare a variable of the type `RaycastHit` to store the information about the hit point.
2. Use a slightly different version of `Physics.Raycast()` to pass the hit info out of it:

```
RaycastHit hitInfoVariable
Physics.Raycast(Vector3 originOfTheRay, Vector3 directionOfTheRay, out hitInfoVariable)
```

RayCasting

So if wanted to Raycast from a GameObject (for example a Vive tracker or the user's headset POV):

We want to shoot a ray from:
gameObject.transform.position

in the direction of:
gameObject.transform.forward

(**gameObject.transform.forward** is the **local Z-axis** of the *object,* which may be different from the *world* Z-axis, which is Vector3.forward)

RayCasting

```
void Update() {
    RaycastHit hit;
    if ( Physics.Raycast(gameObject.transform.position, gameObject.transform.forward, out hit) ) {

        Debug.DrawLine(gameObject.transform.position, hit.point, Color.red);
        Debug.DrawRay(hit.point, hit.normal, Color.green);

    }
}
```

RayCasting

the **hit** variable that stores information about the result of the Raycast has a few useful properties:

hit.point (The coordinates of the collision as a Vector3)

hit.normal (A Vector3 direction that describes the direction coming *straight out* of the face of the hit object)

RayCasting

These visual Debug functions help you see what's going on. They will draw lines in your *Editor*, but never in the actual *Game* view:

```
Debug.DrawLine(Vector3 lineStartCoordinate, Vector3 lineEndCoordinate, Color color);

Debug.DrawRay(Vector3 lineStartCoordinate, Vector3 lineDirection, Color color);
```

ARx Designers: The Future Kings and Queens of Silicon Valley

https://goo.gl/dfWGeA

Design For Humanity - Parts 4,5

http://goo.gl/mWoxTm

TECH 1711 - Mixed Reality Studio