Evolving Mixed Reality

Designing on the Real World

Ran Gal
Microsoft Research

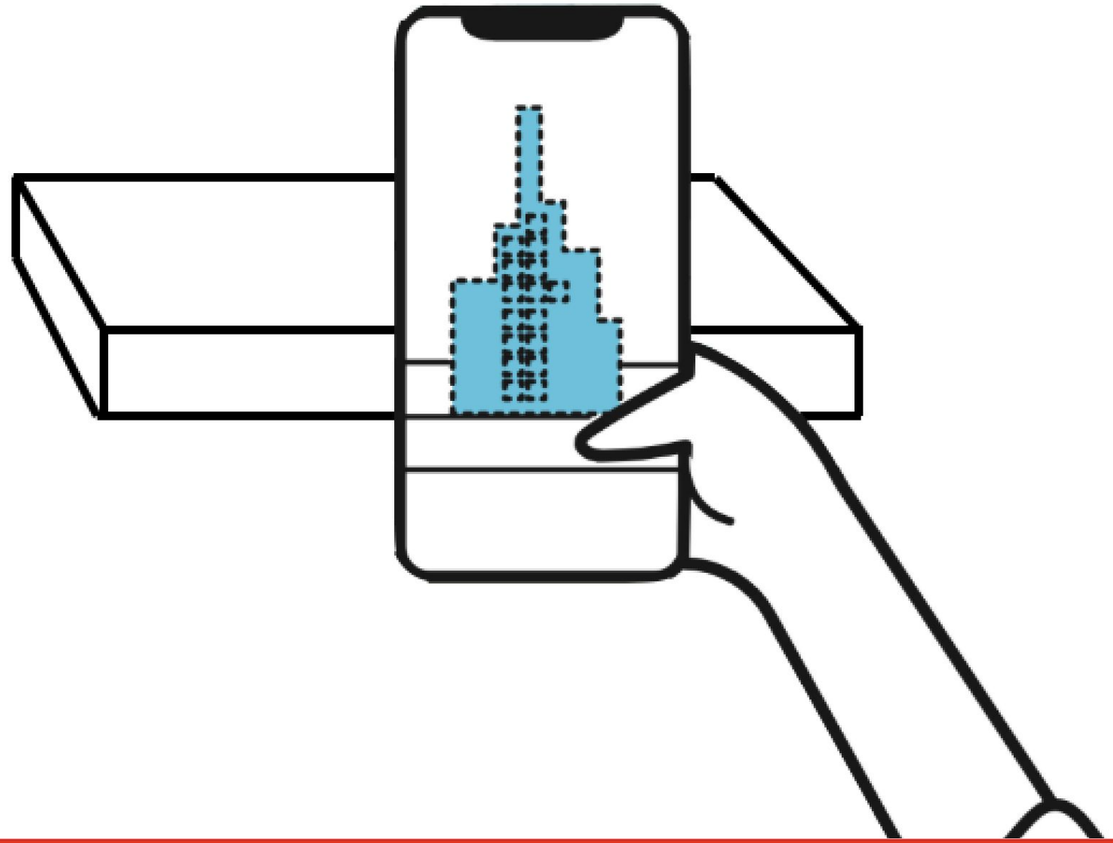
Samantha Gorman
Tender Claws

Brian Schwab
Magic Leap

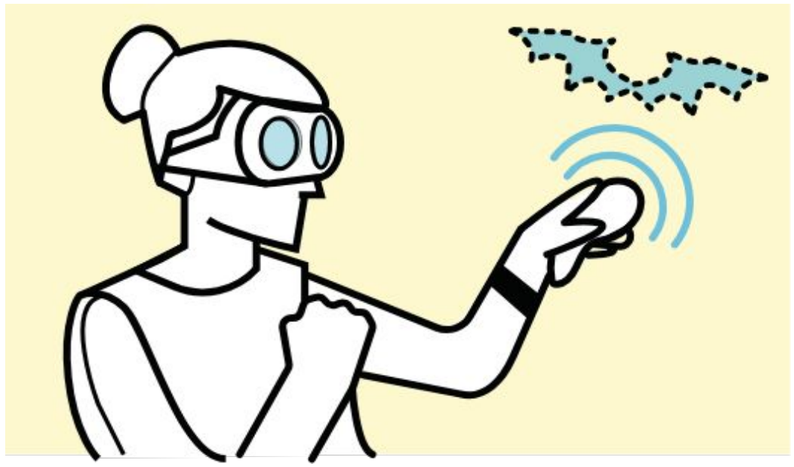
Jono Forbes
Unity

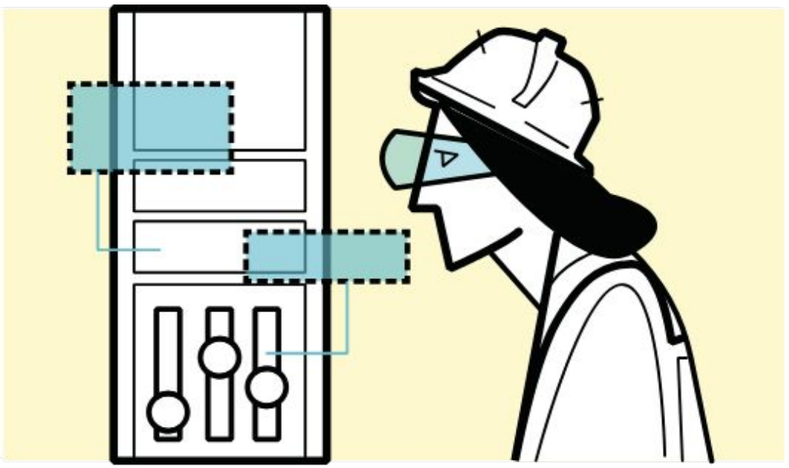
VIRTUAL REALITY DEVELOPERS CONFERENCE
MARCH 18–19, 2019 | #GDC19

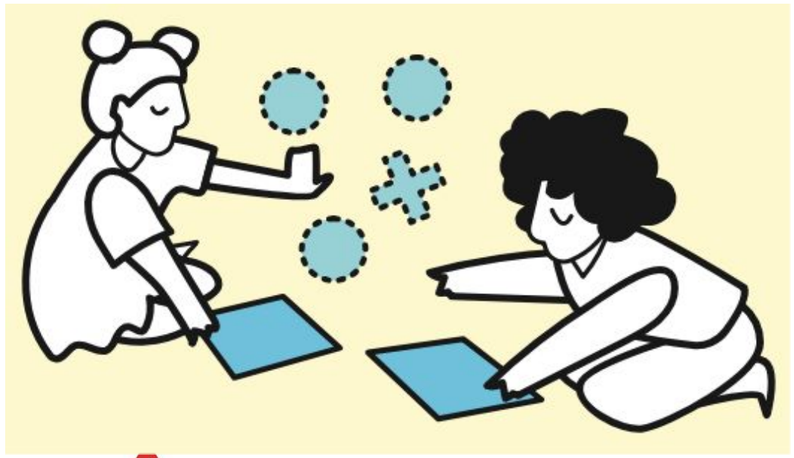
Where We Are Today



Where We’re Going

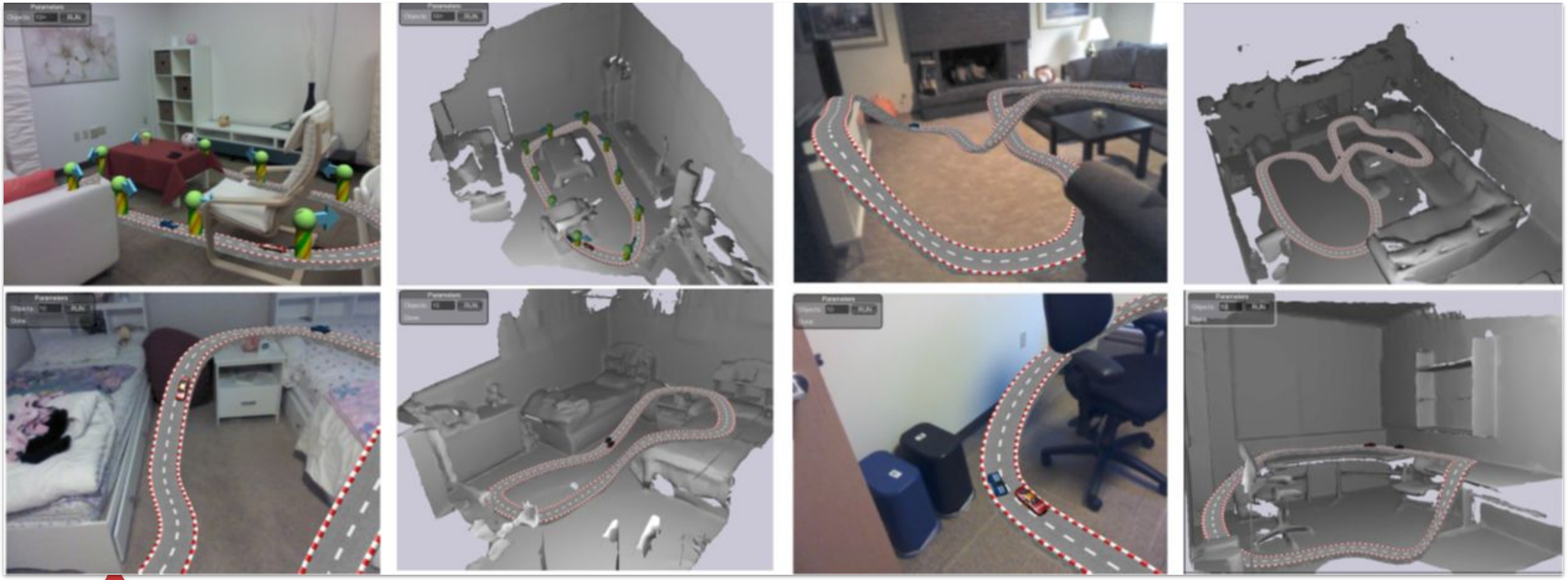




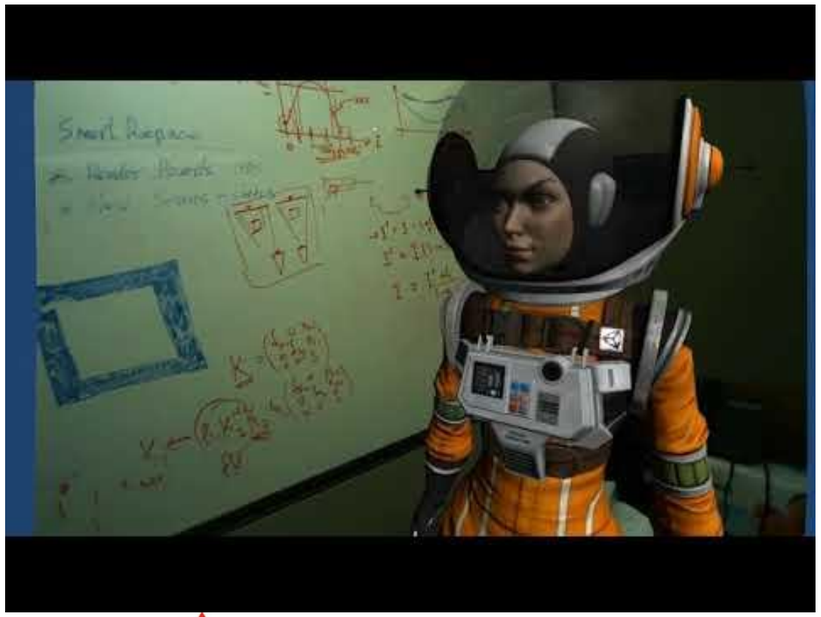


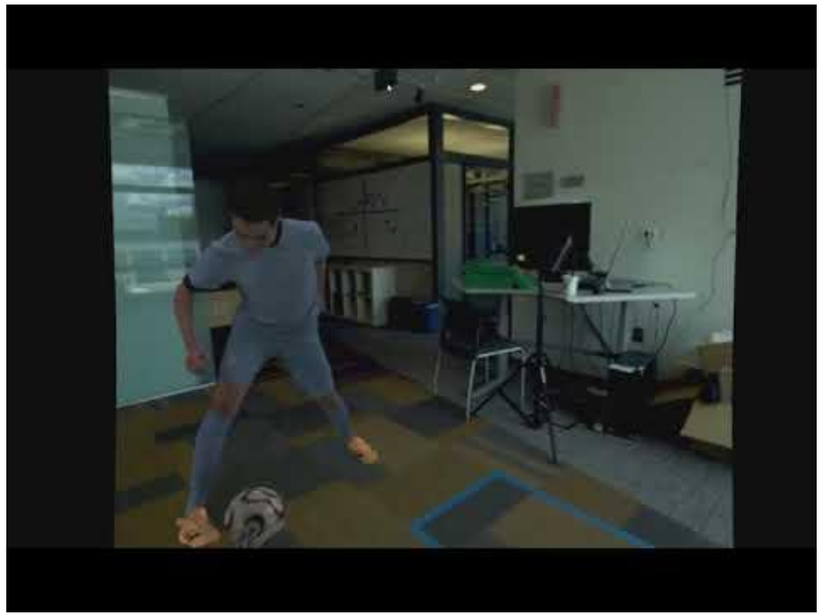

VRDC
Ran Gal
Microsoft Research
VIRTUAL REALITY DEVELOPERS CONFERENCE
MARCH 18–19, 2019 | #GDC19

FLARE



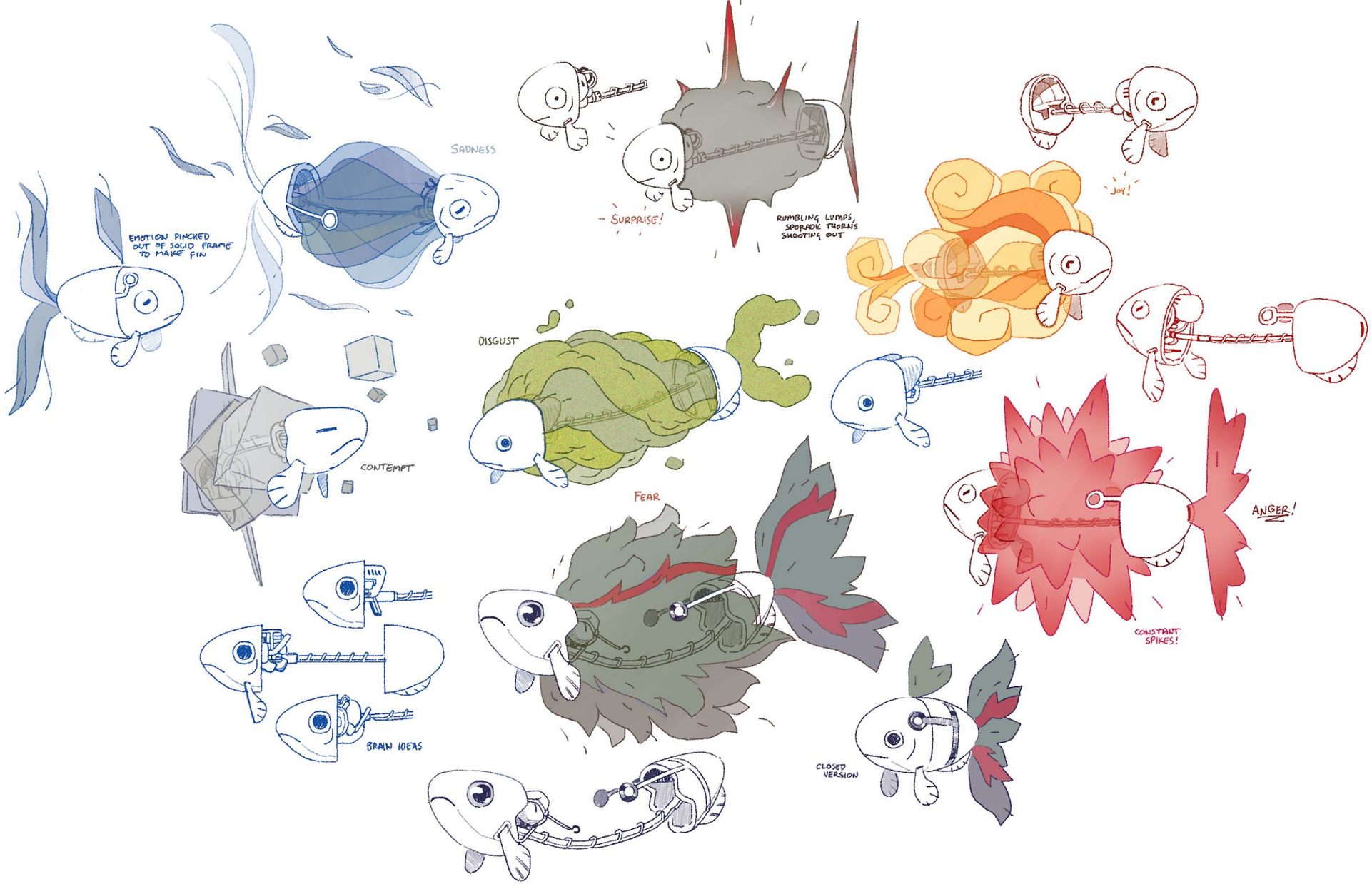
More Examples





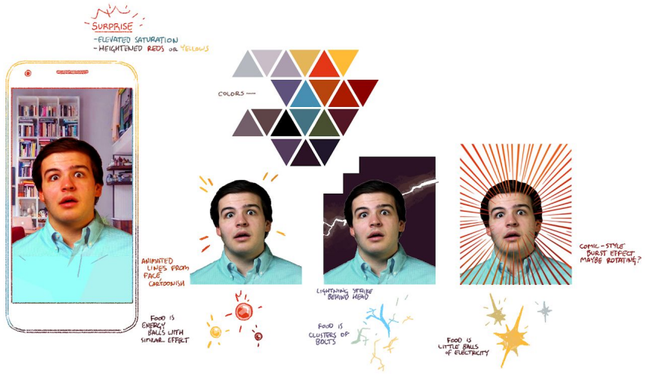

VRDC
Samantha Gorman
Tender Claws
VIRTUAL REALITY DEVELOPERS CONFERENCE
MARCH 18–19, 2019 | #GDC19

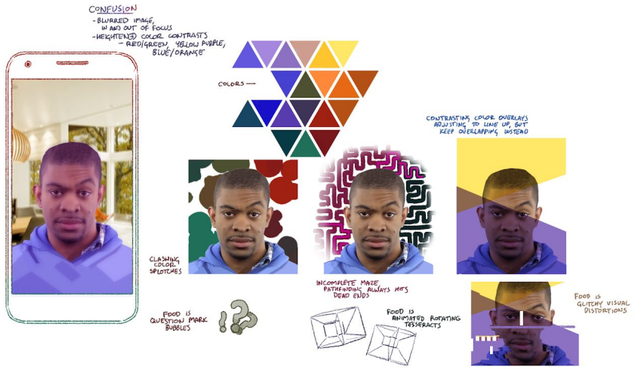

28,561
I feel so light and bouncy
Like I could float into outer space
And touch the stars.

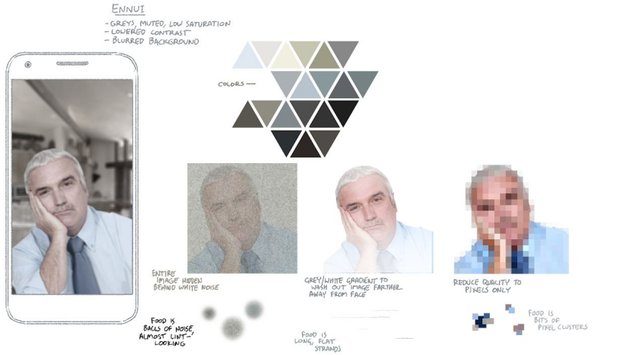

SADNESS
EMOTION PINCHED OUT OF SOLID FRAME TO MAKE FIN
SURPRISE!
RUMBLING LUMPS, SPORADIC THORNS SHOOTING OUT
JOY!
DISGUST
CONTEMPT
FEAR
ANGER!
CONSTANT SPIKES!
BRAIN IDEAS
CLOSED VERSION

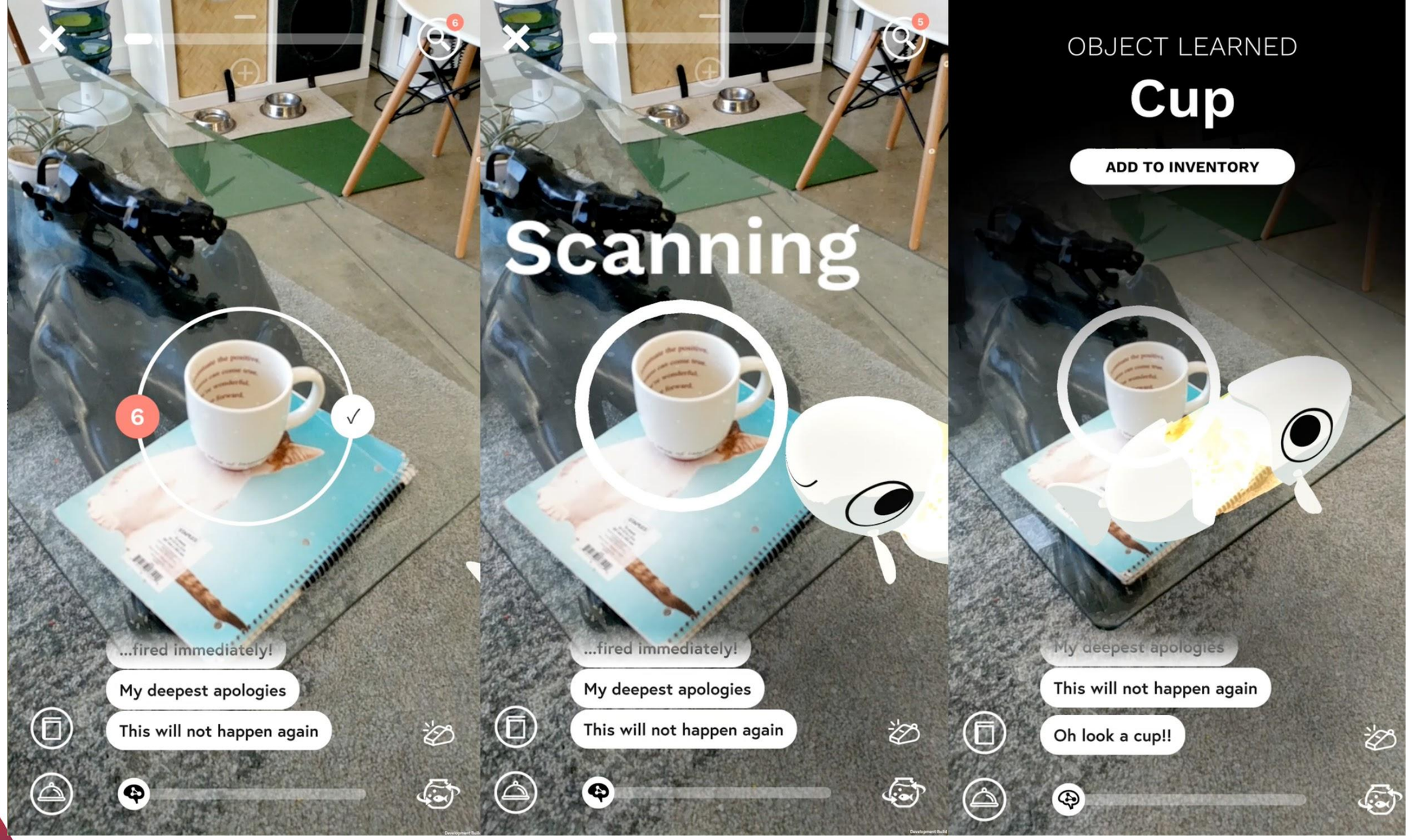

Tendar

World Context Independent








6
...fired immediately!
My deepest apologies
This will not happen again
Scanning
5
...fired immediately!
My deepest apologies
This will not happen again
OBJECT LEARNED
Cup
ADD TO INVENTORY
My deepest apologies
This will not happen again
Oh look a cup!!

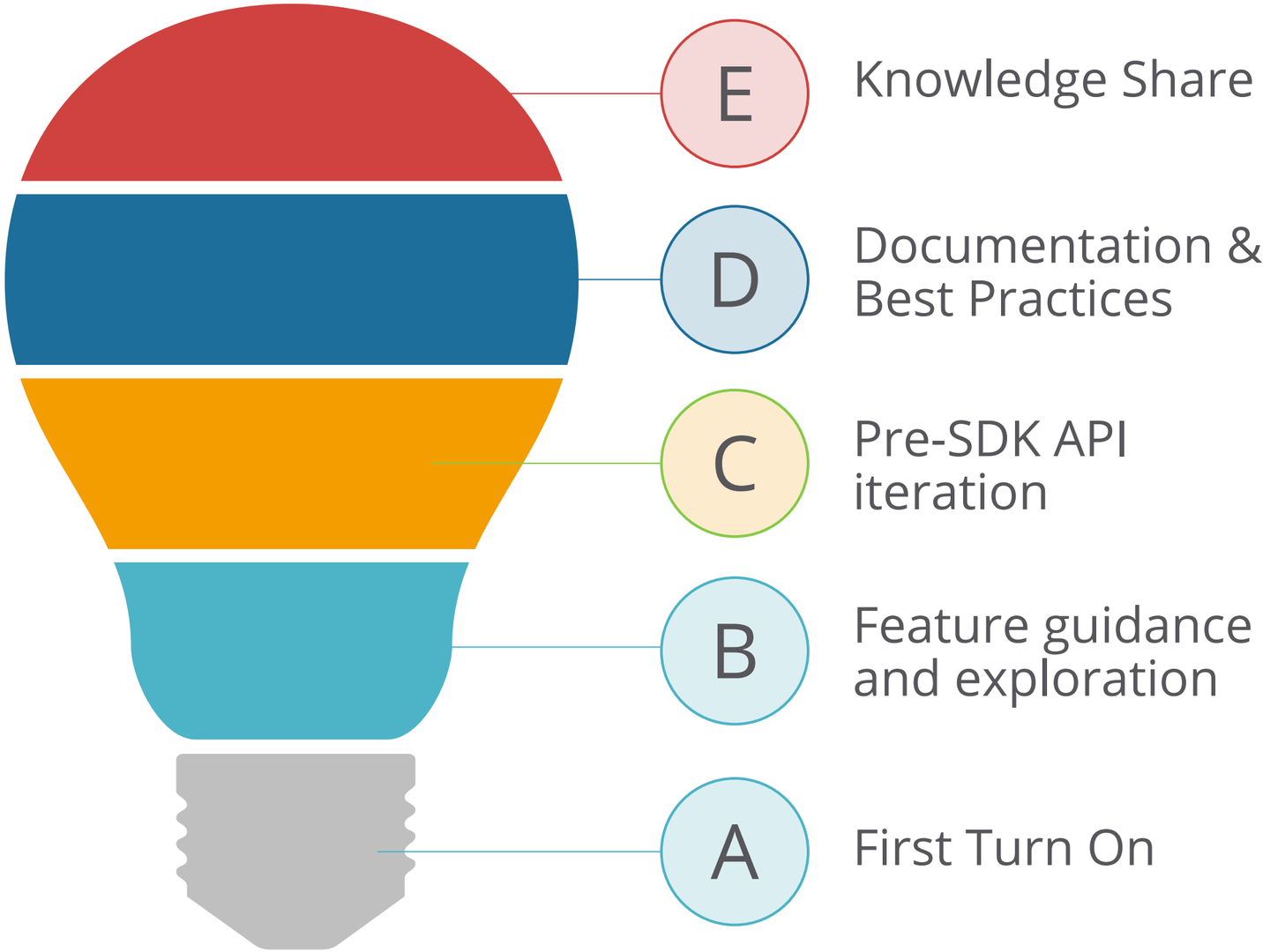
Meaningful Social Play


VRDC
Brian Schwab
Magic Leap - Interaction Lab
VIRTUAL REALITY DEVELOPERS CONFERENCE
MARCH 18–19, 2019 | #GDC19

What is the Lab?

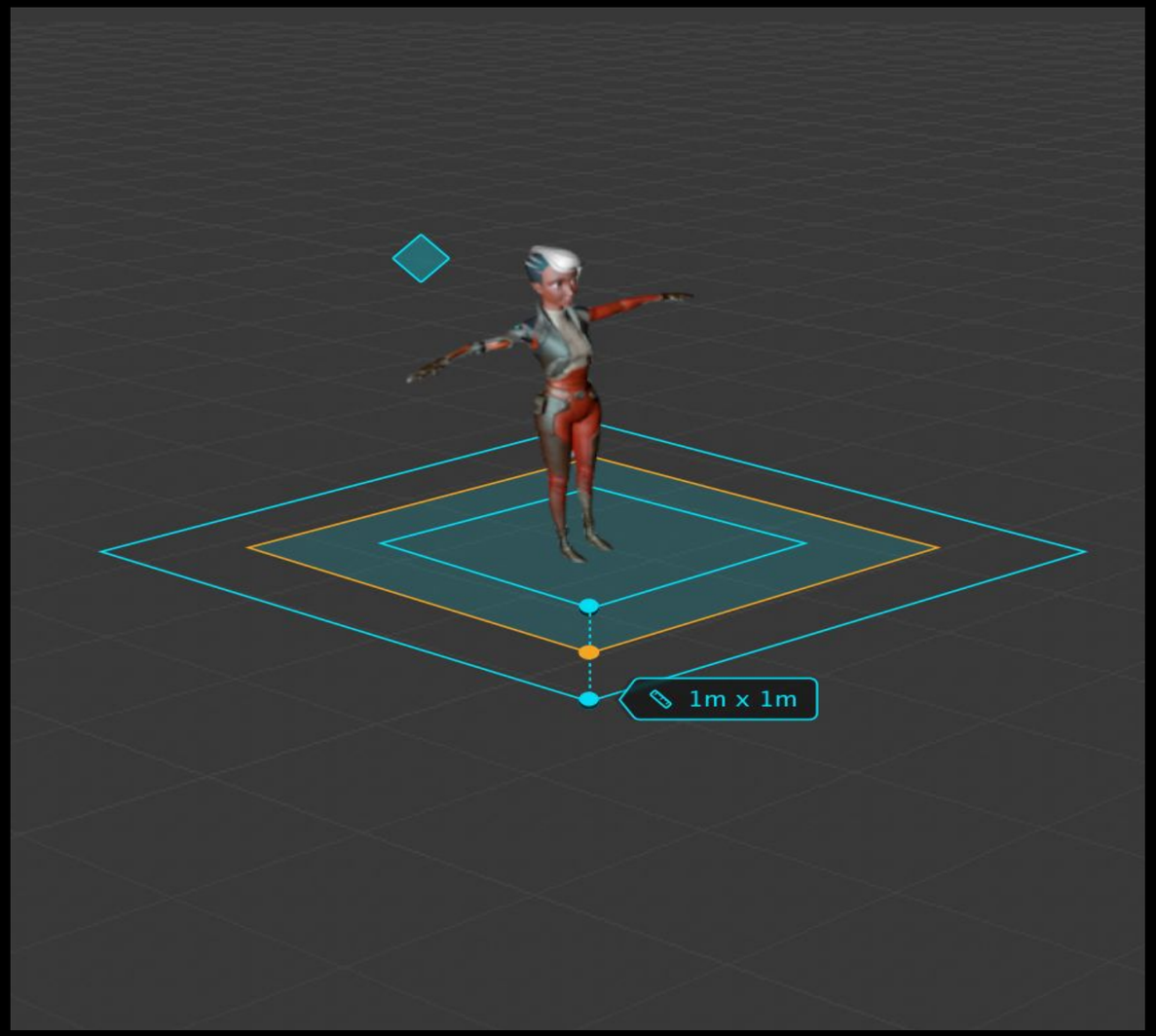
Small team,
mostly engineers,
rapid prototypers.

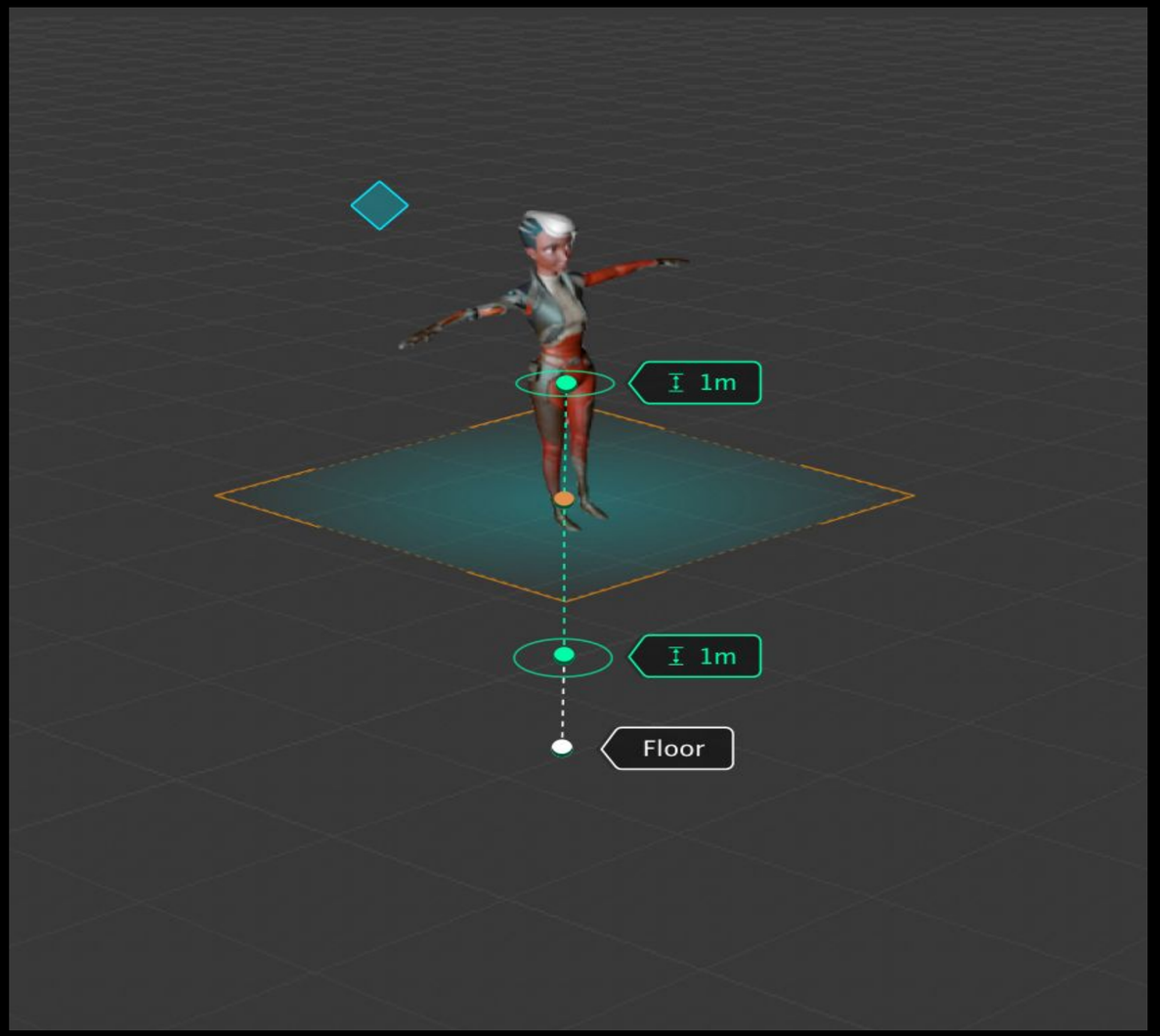
Sit between
hardware/perception
and the SDK group.




VRDC
Jono Forbes
Unity Labs - MARS
VIRTUAL REALITY DEVELOPERS CONFERENCE
MARCH 18–19, 2019 | #GDC19


Unity 2018.2.20f1 (64bit) - [PREVIEW PACKAGES IN USE] - Spawners-Zeldify.unity - mars-project - 2019-2-27 zeldify - Android <DX11>
File Edit Assets GameObject Component AR Tools MARS Window Help
Pivot Global
Scene Simulation View Sim Device
Shaded 2D
Simulation Synthetic : 03_living_room
View Options
Collab Account Layers Layout
Project Hierarchy MARS Panel
World Scale
Simulation View
Type Synthetic
03 living room
Persp
Environment
Augmented Objects
Camera Parent
Spawn Vines (Stretch to Extents)
Floor
Table -> Dropship
Spawn Grass on Horizontal Planes
Spawn Cliff on Vertical Planes
Surface -> Chomper
Surface -> Ellen
Providers
Placement Options
Data Visualizers
Query Viewer


1m x 1m


1m
1m
Floor

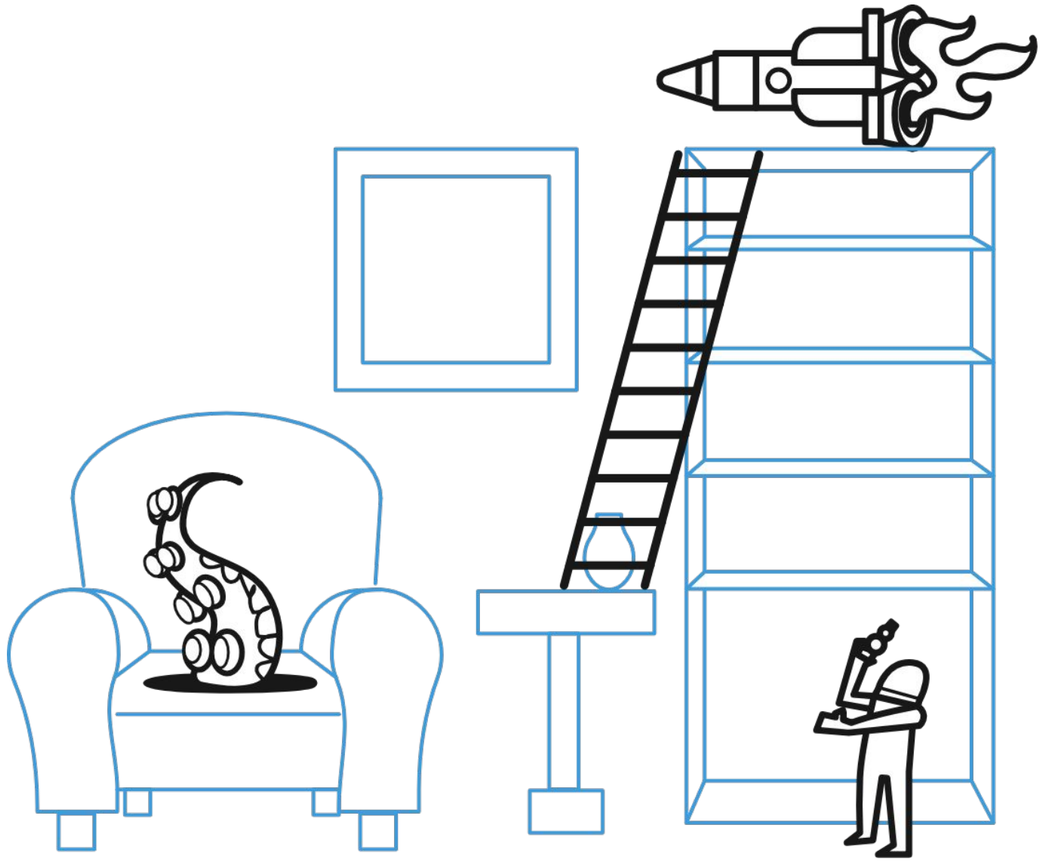

9:41
1.5 x 3.2
Min
Max
W 1.5
H 3.2
Cancel
Done

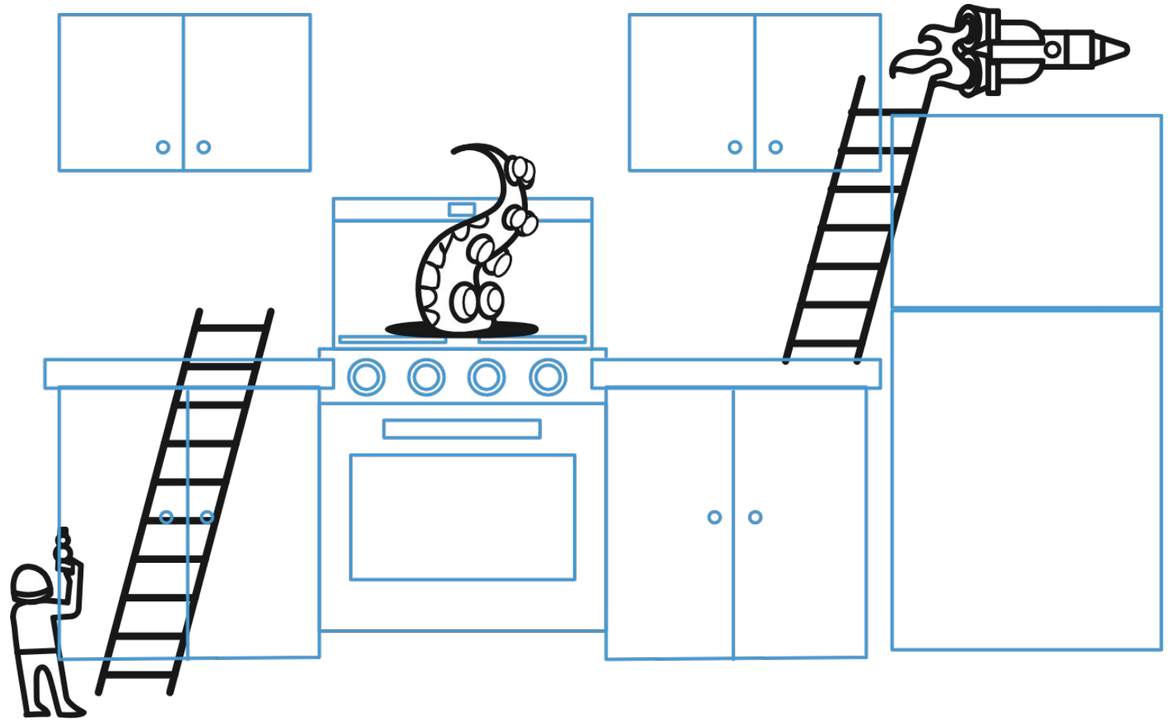

VRDC
Challenges of AR
VIRTUAL REALITY DEVELOPERS CONFERENCE
MARCH 18–19, 2019 | #GDC19

The Unknown Real World


VRDC
Rules, Constraints,
Conditions, Fallbacks
VIRTUAL REALITY DEVELOPERS CONFERENCE
MARCH 18–19, 2019 | #GDC19

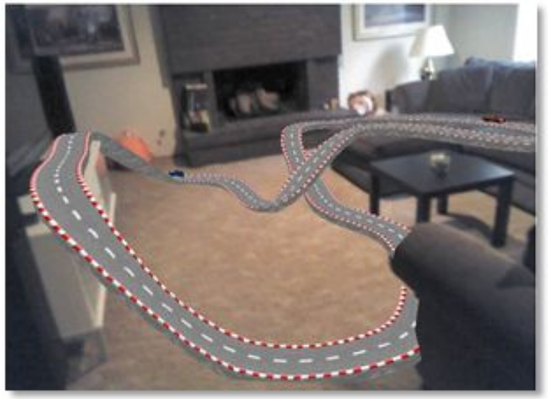
Responsive Design





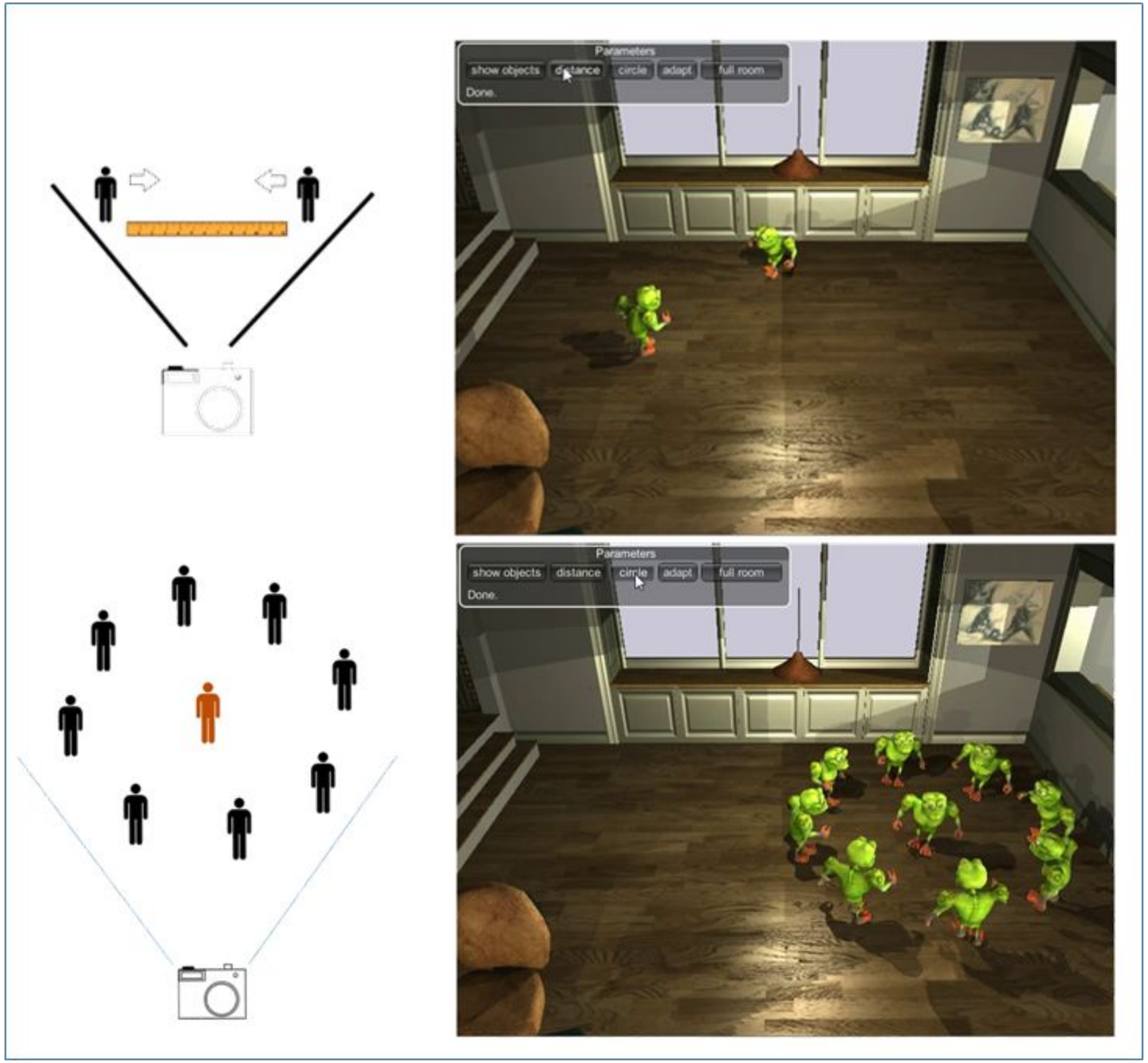
Reality is Complex

- Challenges
  - Layout consistency
    - With Other Virtual Content
    - With the Physical Environment
  - Handling Several Application in the same space
  - Handling Several Users in the same space



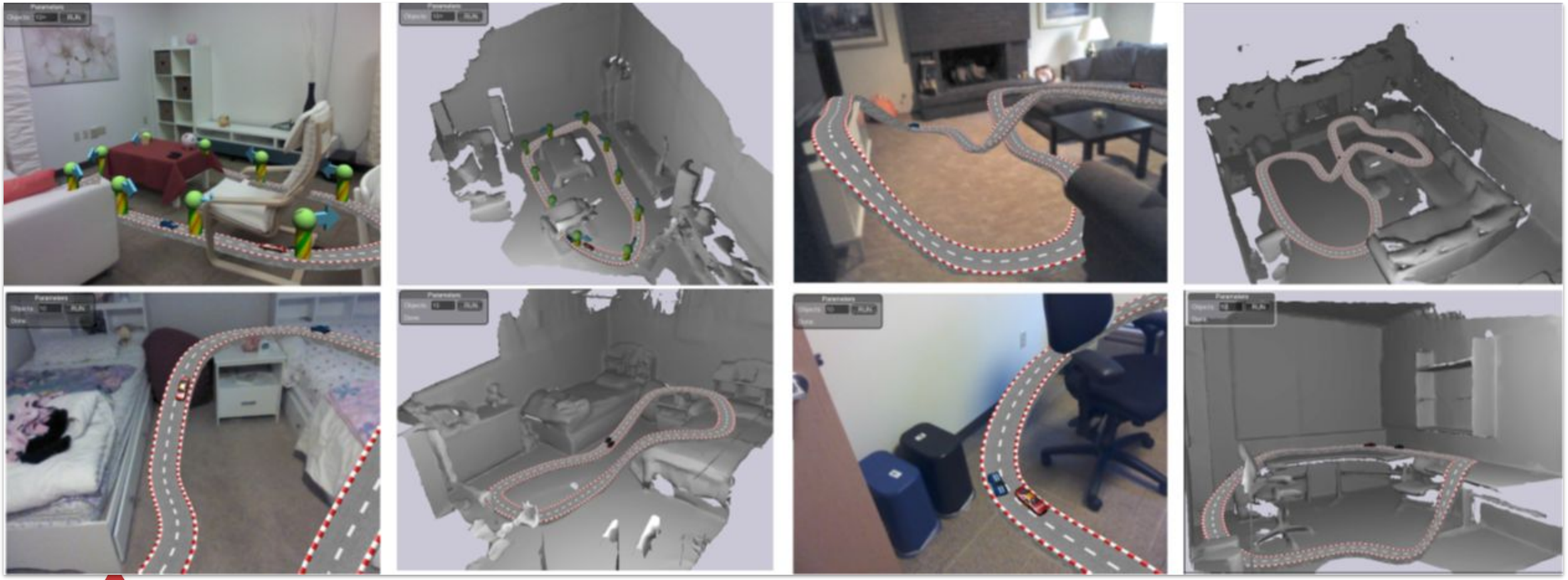
FLARE

- **Use Declarative Rules**
  - Constraining Elements Properties using Cost Functions
- **Solve as a constraint satisfaction problem**
  - Solution’s quality depends on allocated cycles



Race Track

Race Track



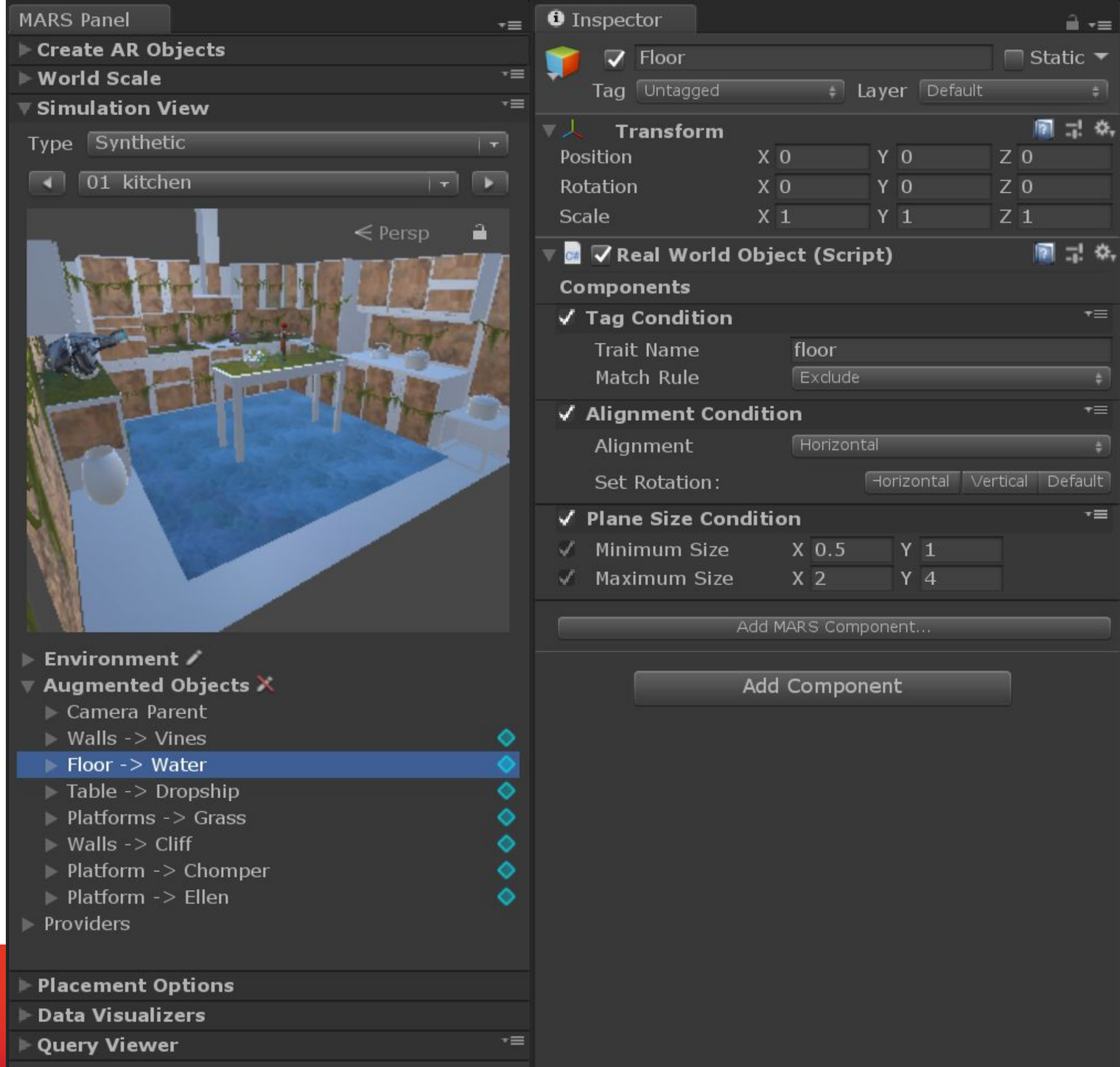
Graceful Adaptation

[redacted]

[redacted]

MARS

- **Real World Objects**
- **Conditions**
- **Rules**
- **Fallbacks**



The “generic” AR rules pipeline

- Sensing - depth, RGB, IR, motion, acceleration, audio, lighting, etc.
- Analysis
  - Room structure, Hole filling, Detection and Classification.
  - More semantic information leads to more intelligent layouts.
- Layout the experience in the current environment
- Apply the Experience’s logic
- Continue sensing and responding to environmental events
  - Allowing for Graceful Adaptation


VRDC
Machine Learning
VIRTUAL REALITY DEVELOPERS CONFERENCE
MARCH 18–19, 2019 | #GDC19

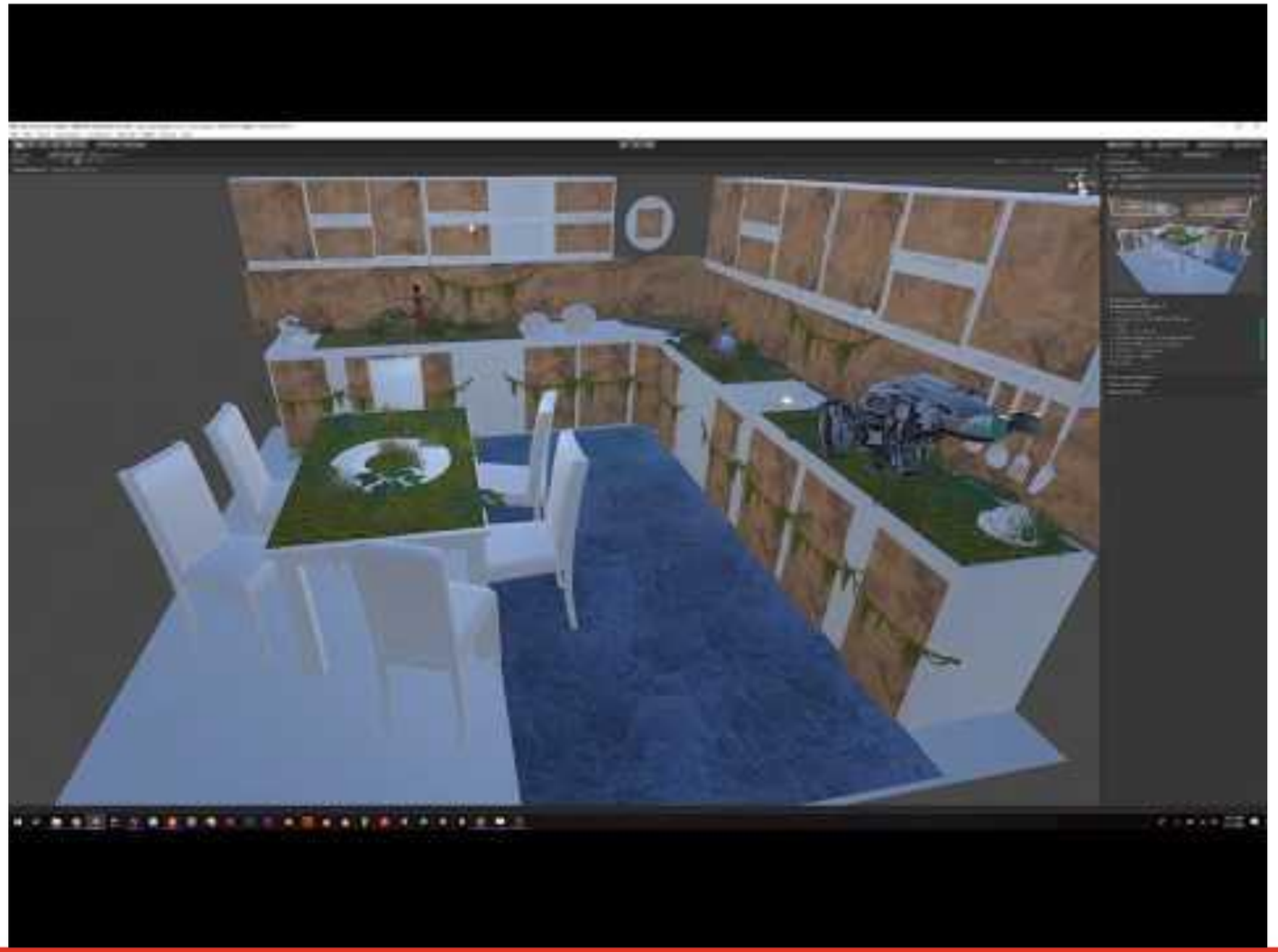

VRDC
User Input & Intent
VIRTUAL REALITY DEVELOPERS CONFERENCE
MARCH 18–19, 2019 | #GDC19

Going Procedural

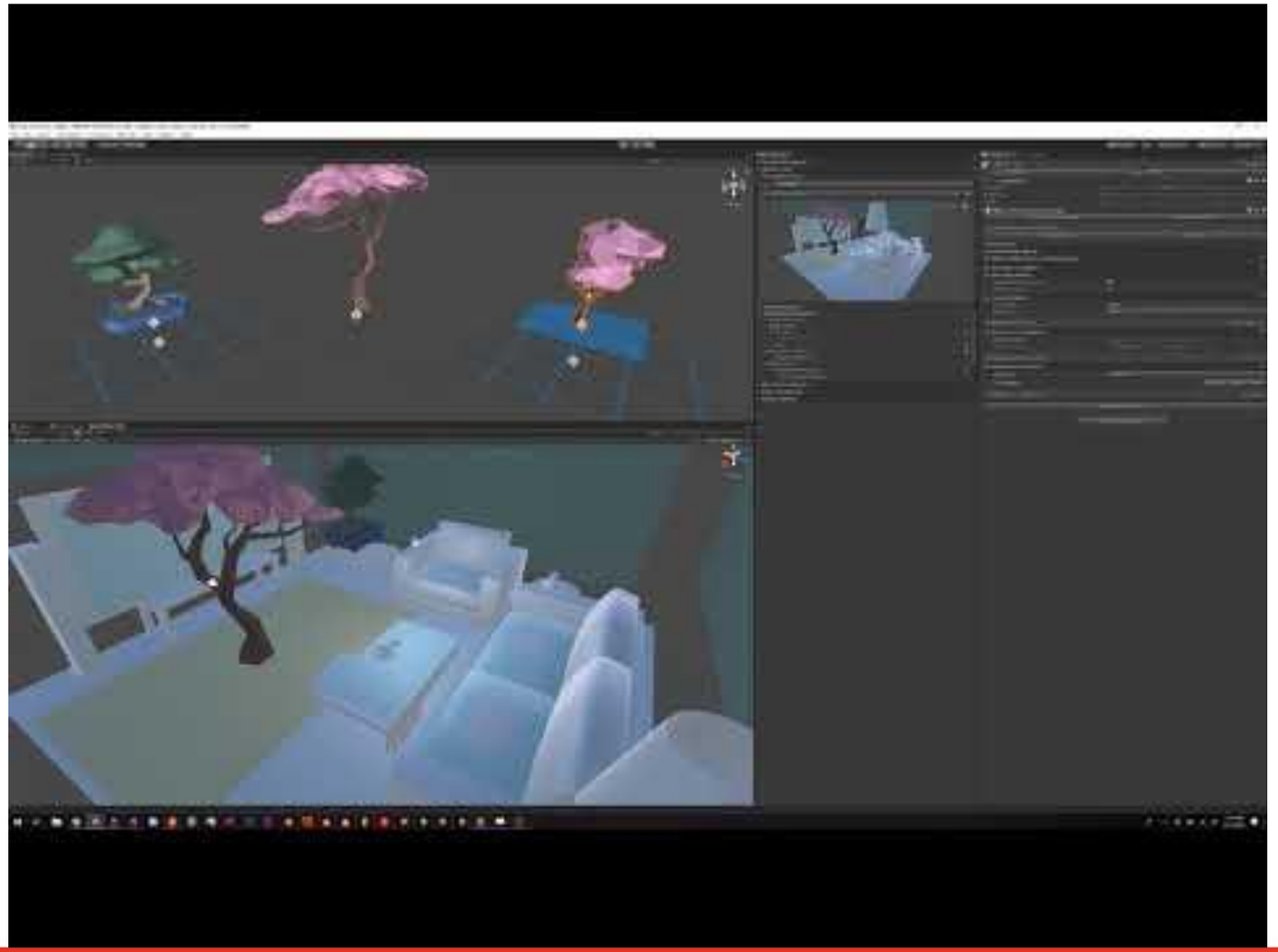
VIRTUAL REALITY DEVELOPERS CONFERENCE
MARCH 18–19, 2019 | #GDC19


VRDC
Tools & Workflows
VIRTUAL REALITY DEVELOPERS CONFERENCE
MARCH 18–19, 2019 | #GDC19

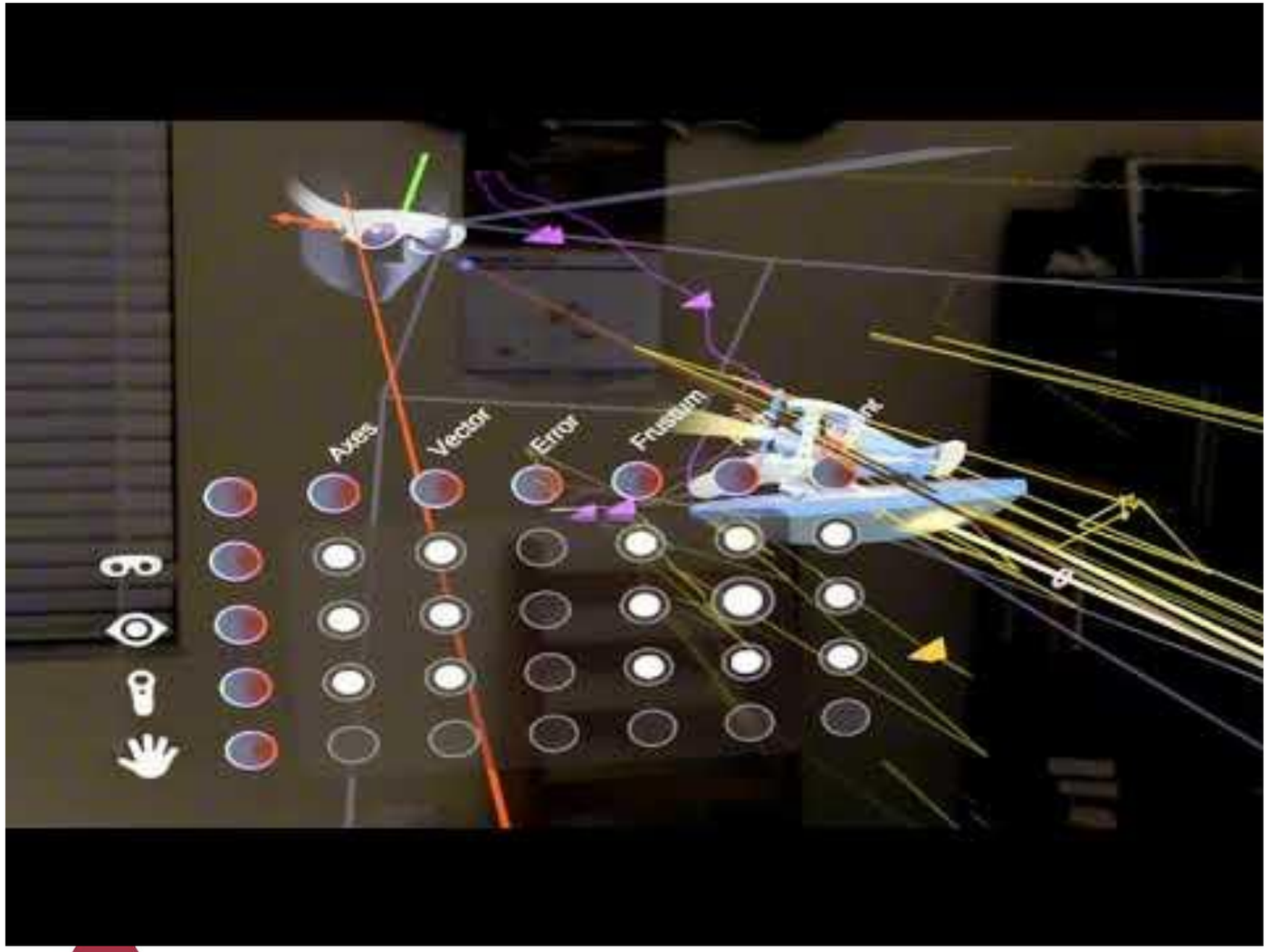
Simulation

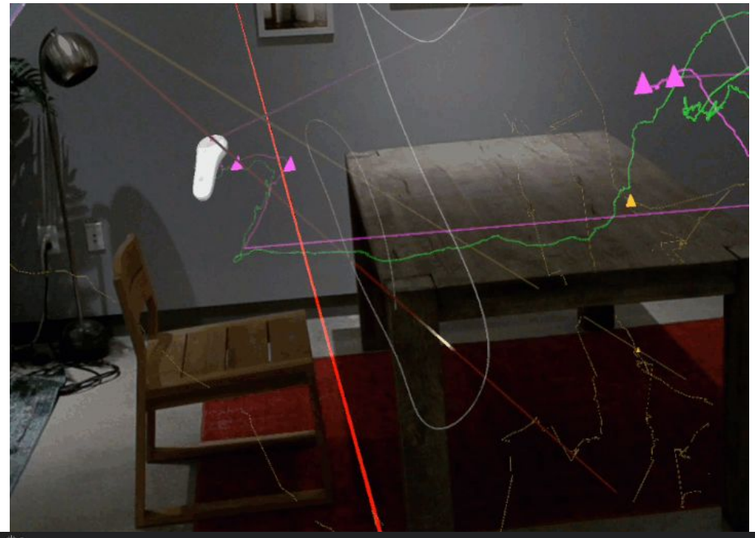


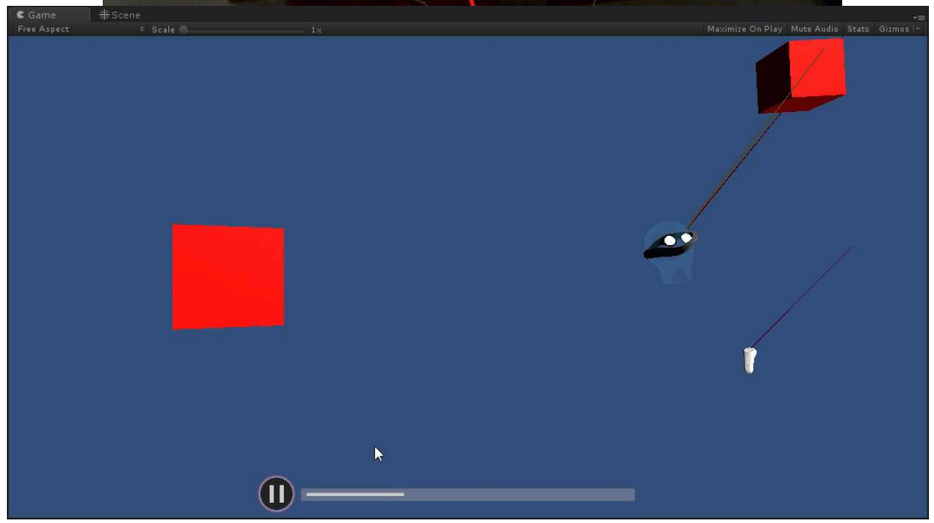
In Editor Debugging



In Experience Debugging








VRDC
XR Ecosystem
VIRTUAL REALITY DEVELOPERS CONFERENCE
MARCH 18–19, 2019 | #GDC19

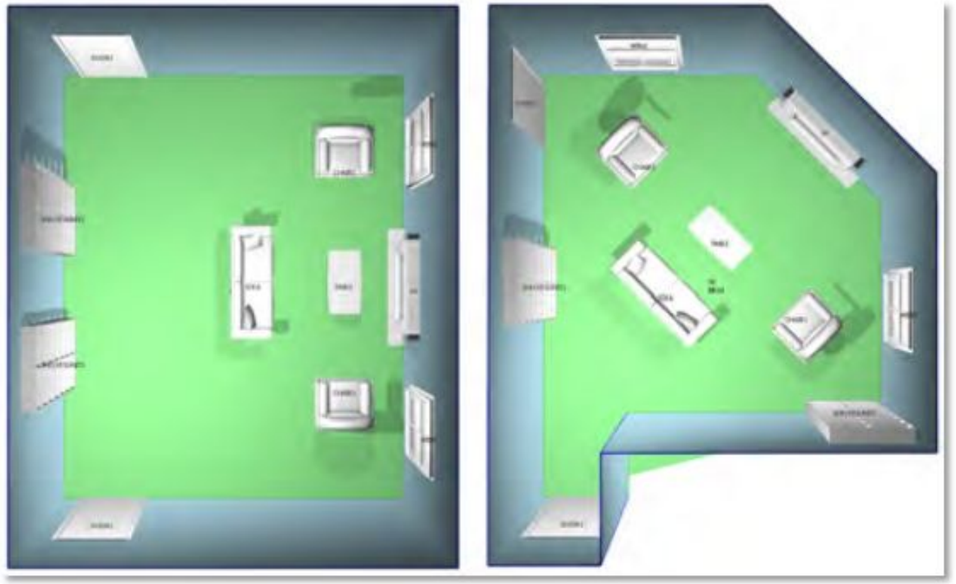
Many Moving Targets

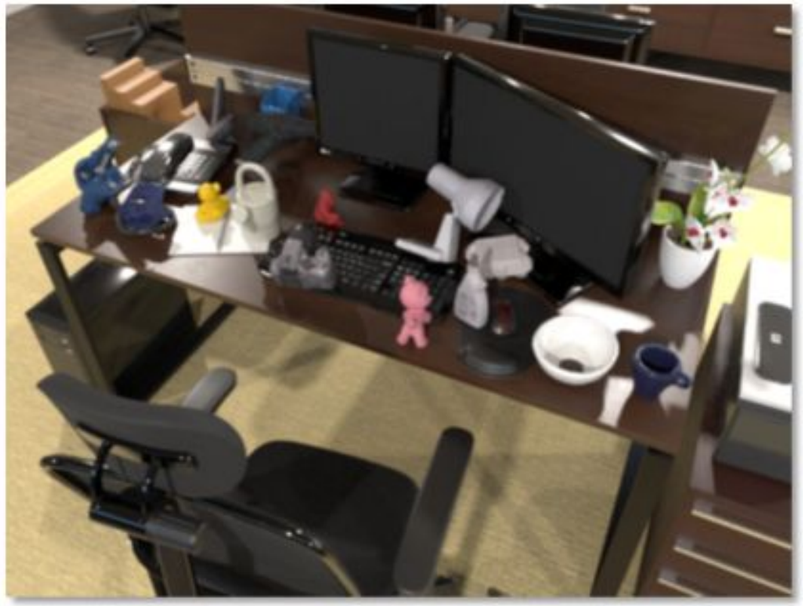
Name	Surfaces	Hit Tests	Meshi...	Markers	Relocaliz...	Camera Pose	Faces	Controller Tr...	3D Markers	Object Reco...	Light Estima...	Eye Tracking	Body Tracking	Hand Tracki...	Hand Pose	Object Seg...
ARKit	Yes	Yes		Yes	Yes	Yes	Yes		Yes	Partial	Yes					
ARCore	Yes	Yes		Partial	Yes	Yes	Yes				Partial					
Tango (defunct)			Yes		Yes	Yes					Yes					
Hololens		Yes	Yes		Yes	Yes								Yes	Yes	
Magic Leap	Yes	Yes	Yes	Yes	Yes	Yes		Yes				Yes		Yes	Yes	
6d.ai			Yes		Yes											
HTC Vive Pro	Yes	Yes	Yes		Yes	Yes		Yes								
Windows MR						Yes		Yes								
Mirage Solo						Yes		Yes								
HTC Vive						Yes		Yes								
Oculus Quest						Yes		Yes								
Oculus Rift						Yes		Yes								
Oculus Go								Yes								
Vuforia						Yes			Yes		Yes					
Placenote					Yes											
Selerio									Yes	Yes						Yes
ULsee							Yes									
Visage							Yes									
Google Mobile Vision							Partial			Yes						
Apple Vision							Partial			Yes						
Wrnch.ai													Yes	Yes		
Leap Motion														Yes	Yes	
OpenCV	Partial		Yes	Yes		Partial							Yes	Yes		
dlib			Yes													Yes
Resonai			Yes							Yes						

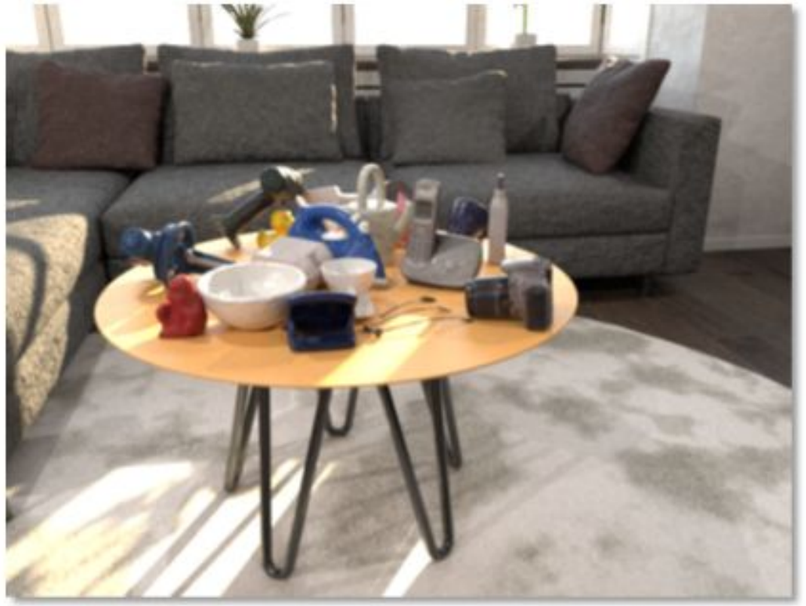
Other Applications

Other Applications

Other Applications







Other Applications

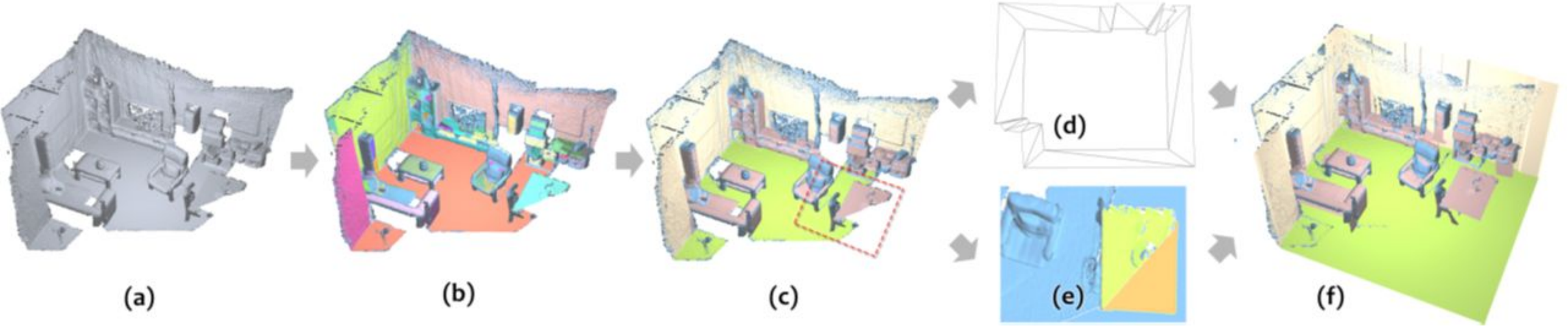
Much more to uncover

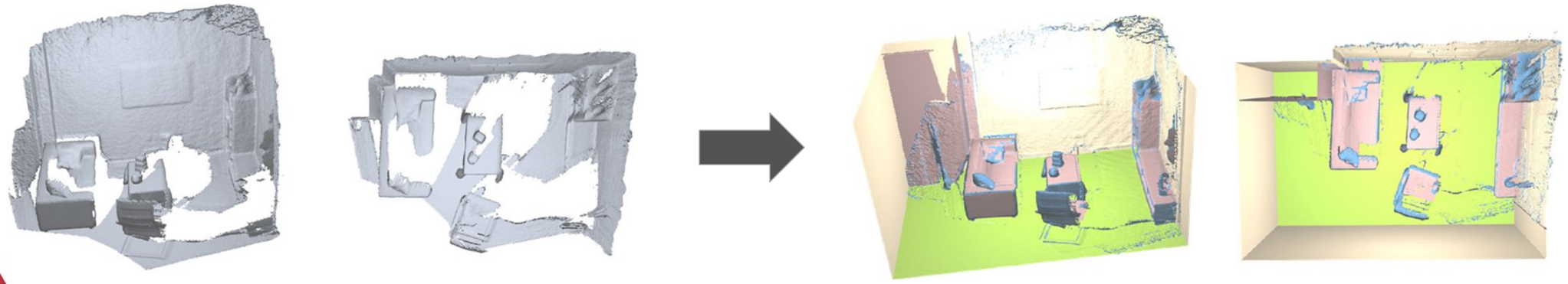
- Layers of apps, content, styles
- Permissions
- Privacy & Ethics
- ...


VRDC
Thank You!
Time for Questions?
VIRTUAL REALITY DEVELOPERS CONFERENCE
MARCH 18–19, 2019 | #GDC19

Additional Slides

Contour and Surface Completion



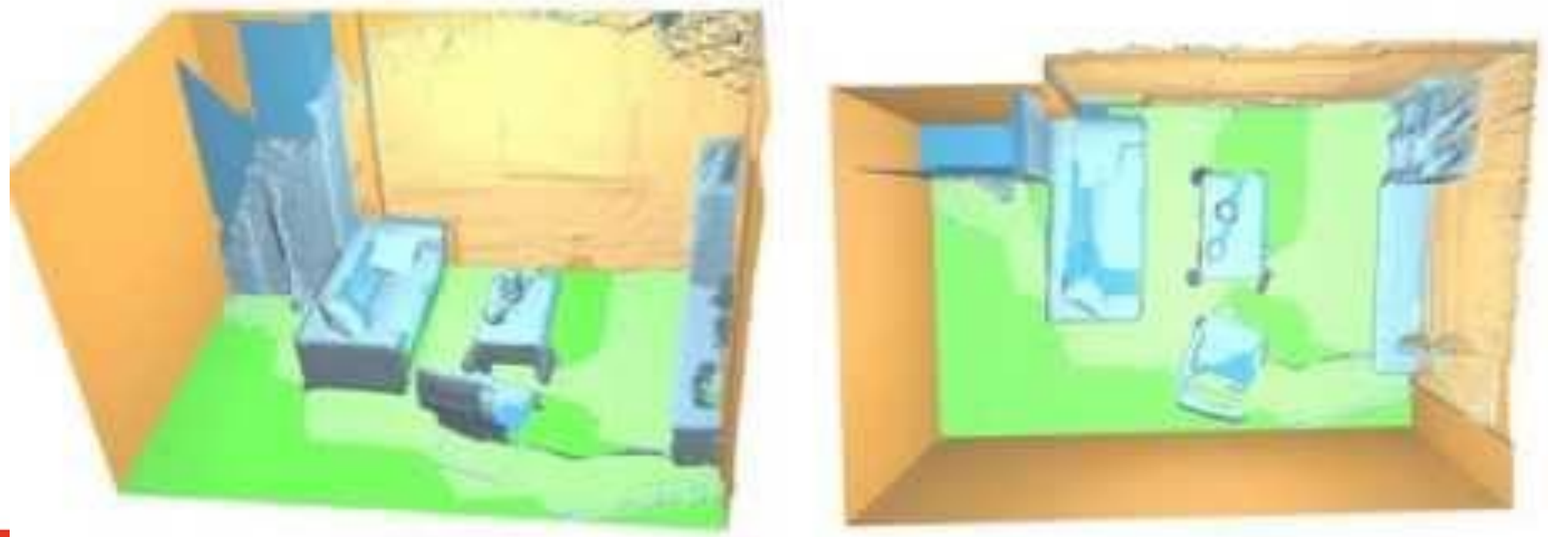


Contour and Surface Completion

[redacted]

Example 1: Living Room

Room after completion



[redacted]

Layout Design for AR Applications

- A designer specifies the layout of the application
  - Elements
    - 3D Object -
      - Type, Orientation, Scale.
    - Frustum
      - Orientation, Field of view, Clipping Planes.
  - Rules
    - Algebraic notation
    - A library of predefined routines
    - Comparison and Boolean operators.
      - cost(OR(a, b)) = min(cost(a), cost(b)),
      - cost(AND(a, b)) = cost(a) + cost(b).

Predefined Routines:

- Distance / Angles
- BB / Geometry Intersection
- Line of Sight
- Inside/Outside Frustum
- Material Properties (Color, Texture)
- Relative Positioning

$$cost(a < b) = \begin{cases} 0 & a < b \\ (a-b)^2 & a \geq b \end{cases}$$

$$cost(s) := \sum_i w_i \cdot r_i(\hat{s}_i)$$

```
// 'F' is the user's frustum.
Screen1:=Object3 ( [28 18 10], VERTICAL, OptPos );     // A screen on the wall

// 4 media objects.
O1:=Object3 ( [10 10 10], HORIZONTAL, OptPos );
O2:=Object3 ( [10 10 10], HORIZONTAL, OptPos );
O3:=Object3 ( [10 10 10], HORIZONTAL, OptPos );
O4:=Object3 ( [10 10 10], HORIZONTAL, OptPos );

Assert ( LocalZ ( FloorFrame.Frame, Screen1.Frame.Position ) < 50 );        // Prevent the hanged screen to
// be too high above the floor

// Bring Screen1 to the center of view (as much as possible)
Assert ( Dot ( Normalize(Screen1.Frame.Position-F.Frame.Position), F.Frame.Axis3) = 1 );

// Screen1 IS inside the frustum
Assert ( Inside ( F, Screen1.Frame.Position) );

// Screen1 is not occluded from the point of view of the user.
Assert ( Visible ( F.Frame.Position, Screen1.Frame.Position) = 0 );

// The 4 media objects are INSIDE the frustum.
Assert ( Inside ( F, O1.Frame.Position) );
Assert ( Inside ( F, O2.Frame.Position) );
Assert ( Inside ( F, O3.Frame.Position) );
Assert ( Inside ( F, O4.Frame.Position) );

// They are all visible to the user.
Assert ( Visible ( F.Frame.Position, O1.Frame.Position) = 0 );
Assert ( Visible ( F.Frame.Position, O2.Frame.Position) = 0 );
Assert ( Visible ( F.Frame.Position, O3.Frame.Position) = 0 );
Assert ( Visible ( F.Frame.Position, O4.Frame.Position) = 0 );

// Limits to the distance of media objects from the user.
Assert ( Distance (F.Frame.Position, O1.Frame.Position) > 50 &&
Distance (F.Frame.Position, O1.Frame.Position) < 450
);
// Make all the objects lie at the same distance from the user.
// (The user position is represented by the frustum 'F'.
Assert (   Distance ( F.Frame.Position, O1.Frame.Position) =
           Distance ( F.Frame.Position, O2.Frame.Position)
);
Assert (
           Distance (F.Frame.Position, O1.Frame.Position) =
           Distance (F.Frame.Position, O3.Frame.Position)
);
```