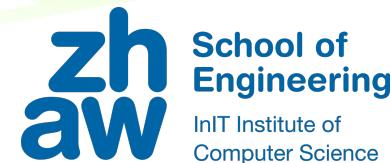


Community Platform to Promote **AR Patterns**

Philipp Ackermann, ZHAW

Alessandro Holler, ZHdK

October 29, 2024 @ AWE EU 24, Vienna



Zürcher Hochschule der Künste
Zurich University of the Arts

About Us



Philipp Ackermann

Zurich University of Applied Sciences ZHAW

- Software Engineer
- Lecturer (Visual Computing)

philipp.ackermann@zhaw.ch



Alessandro Holler

Zurich University of the Arts ZHdK

- Designer (Visual Communication)
- Lecturer (Scientific Visualization)

alessandro.holler@zhdk.ch

Scope of AR Patterns

Challenges in Augmented Reality Projects

«Examples of AR applications are impressive but somehow mysterious.»

«It is hard to express myself when talking about user experiences in AR, because I have no conceptual understanding of what matters.»

«Although experienced in developing AR apps, it is difficult explaining to clients and team members what is relevant for implementing AR apps.»

«It is challenging for me to understand implementation details and estimate efforts needed for creating AR apps.»

Goals

- Establish conceptual understanding and a common language about often-used concepts applied in Augmented Reality
- Create reusable building blocks for ideation, conception, and development in AR
- Improve communication on AR between clients, managers, designers, and developers
- Provide a toolset for designing and documenting AR projects
- Support community with an open source platform

Why Not XR Patterns?

- When XR is mentioned, in most cases VR is meant
but: AR is different from VR
- AR experiences take place in uncontrolled real world
 - Scene understanding for tracking the user's spatial context is key
- Authors of AR have no control over occurrence and timing during the creation process
 - AR applications are heavily event-driven, reactive systems
 - Authoring in AR means designing intents of augmentations
- **Therefore: We do focus on AR only!**

Solution Building Blocks

Event-Condition-Action Diagram

Streamlined abstraction of reactive AR applications

AR Patterns Catalog

Catalog of reusable solutions to common problems in designing AR experiences

Illustration Toolkit

Visual language and graphical template to illustrate AR scenes

AR Scenarios

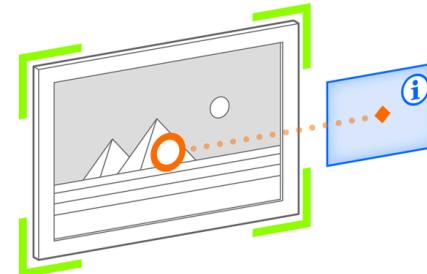
Use-cases with source code demonstrating benefit of AR patterns, illustrations, and ECA diagrams

AR Patterns @ AWE EU 24, Vienna



Augmentation

- Geolocated Remark
- Segment Overlay
- Area Enrichment
- Captured Twin
- Anchored Supplement
- Superimposition
- Tag-along
- Hand/Palm Pop-up
- Ahead Staging
- Pass-through Portal
- Staged Progression
- Attention Director

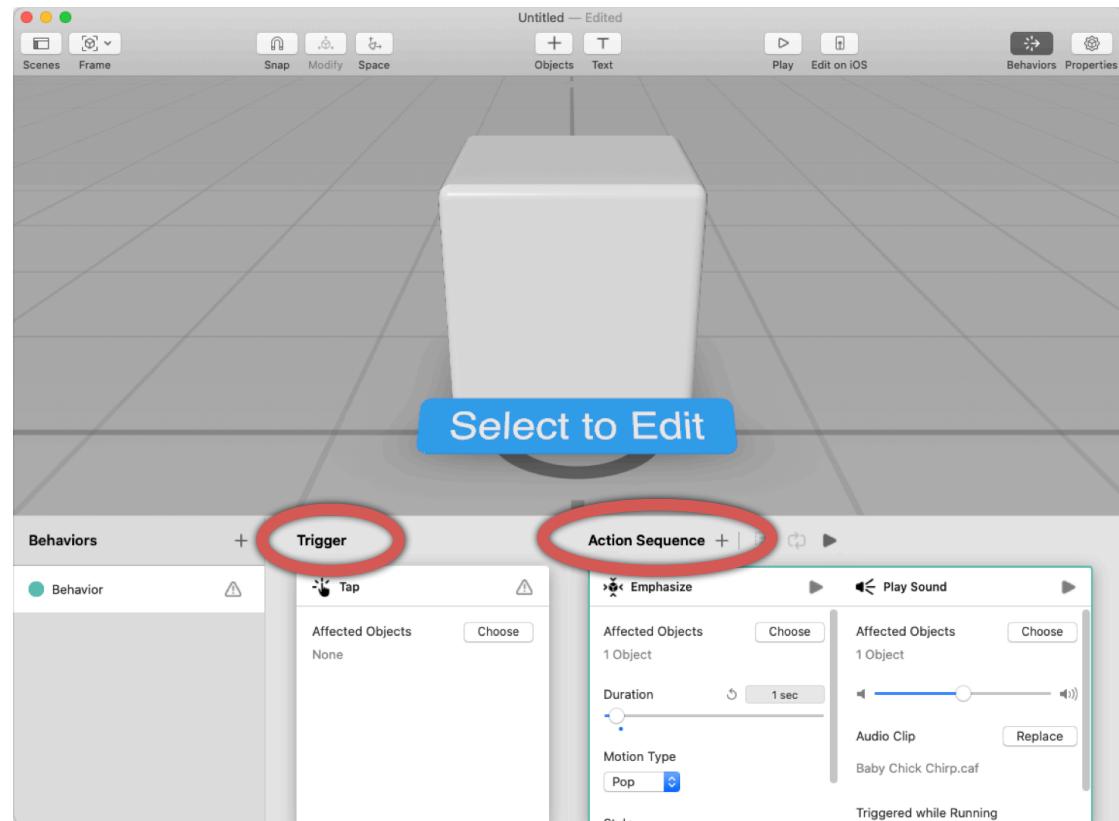


Reactive AR Systems

- **Detection Events**
 - Image-based: Features, segments, text, humans, ...
 - Spatial in 3D: Location, planes, images, objects, body, face, ...
- **User Events**
 - Pose/motion, hand gestures, touch interaction, speech recognition
- **Temporal Events**
- **Data-driven Events**
- ...

Event-Driven Behavior

- **Trigger-Action**
 - Common approach in low-code/no-code AR editors
 - e.g., Apple Reality Composer
 - e.g., Adobe Aero



Apple Reality Composer

Event-Condition-Action Abstraction

Event-Condition-Action (ECA)

- Enhancement of Trigger-Action approach
- ECA as Abstraction Layer
 - Generic concept covering AR behavior
 - Technology-agnostic (HW, Toolkits, Prog. Language)
- ECAs can be seen as active rules evaluated at state transitions

Event-Condition-Action Diagram

ECA Diagram

- A kind of pseudo code
- Rule-reaction block
 - Event-Condition-Action rule
 - Changed state as reaction
- Sequential ECA
- Simple diagram creation
 - Rendered as Markdown

Event	Condition	Action
-------	-----------	--------

| changed state as reaction

in:20 sec	if:items.@count == 0	do:say
-----------	----------------------	--------

| "you may add an item" 🎙

on:command	→	do:detect:image
------------	---	-----------------

| Install image detector 0.1x0.1 ← on:response ... marker.png 🐍

on:detect	→	do:add to AR anchor
-----------	---	---------------------

| 'scene.3D' +

AR Patterns

- Typical, ready-to-use solutions to common problems in designing Augmented Reality experiences
- Catalog of AR Patterns
 - Behavior Patterns
 - Augmentation Patterns
- Check out AR Patterns paper
 - Presented at EuroXR 2023
doi.org/10.1007/978-3-031-48495-7_6

Behavior	Augmentation
→ Instant Reaction	 Geolocated Remark
⌚ Timed Sequence	 Segment Overlay
➡️ Conditional Reaction	 Area Enrichment
➡➡ Indirect Reaction	 Captured Twin
➡➡➡ Chain of Reactions	 Anchored Supplement
➡➡ Request Async Response	 Superimposition
➡️ 0 Complementary Reactions	 Tag-along
➡️ ⓘ Publish-Subscribe Notification	 Hand/Palm Pop-up
⌚ Continous Evaluation	 Ahead Staging
➡️ ⏪ Detector Reactivation	 Pass-through Portal
	 Staged Progression
	 Attention Director

Designing with AR Patterns

How to Describe an AR Pattern?

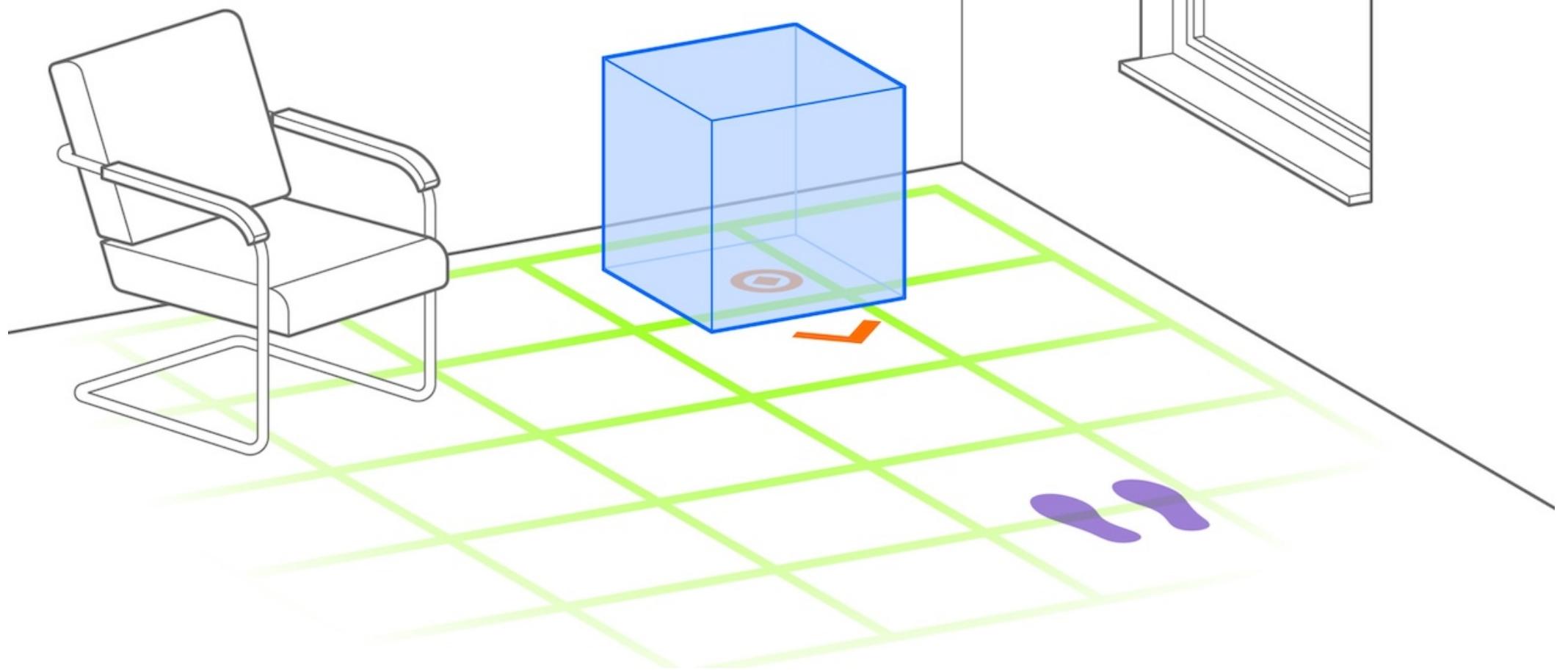
"Ahead staging is a technique for presenting 3D content in a way that it is aligned to the spectator's position and view direction.

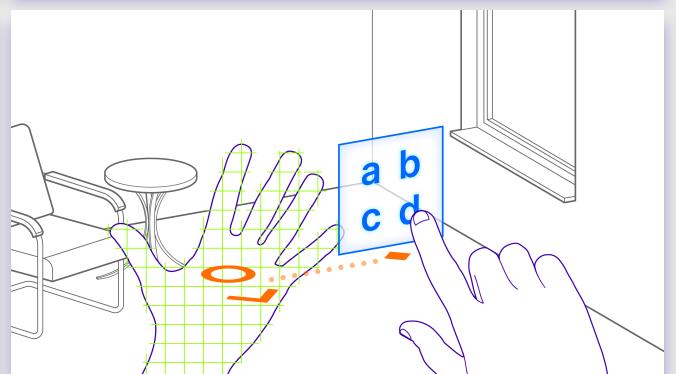
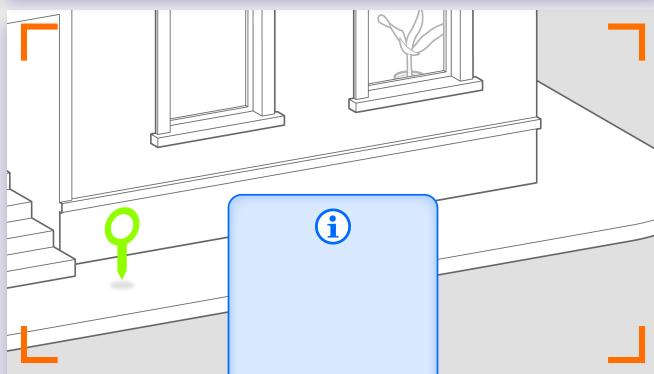
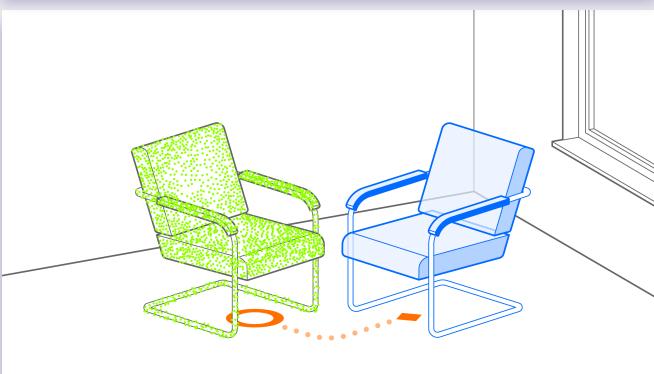
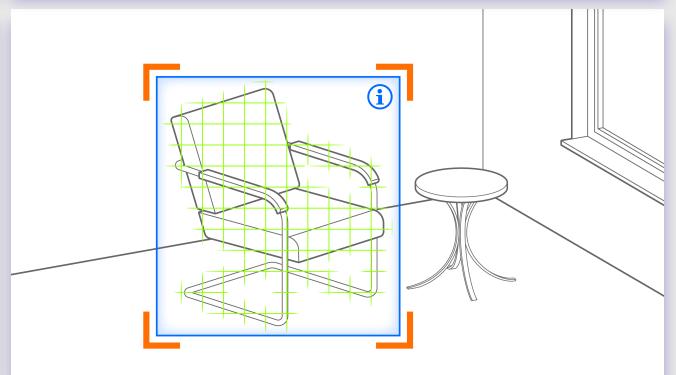
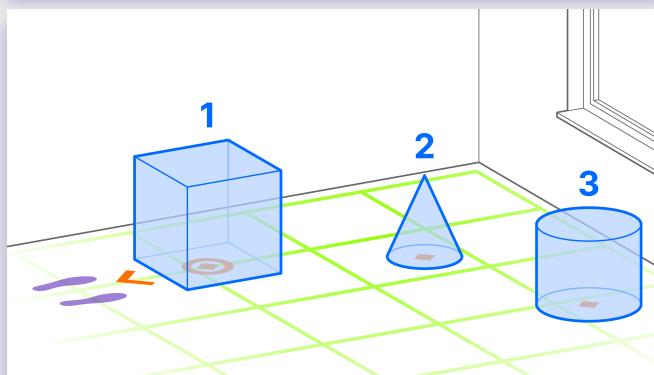
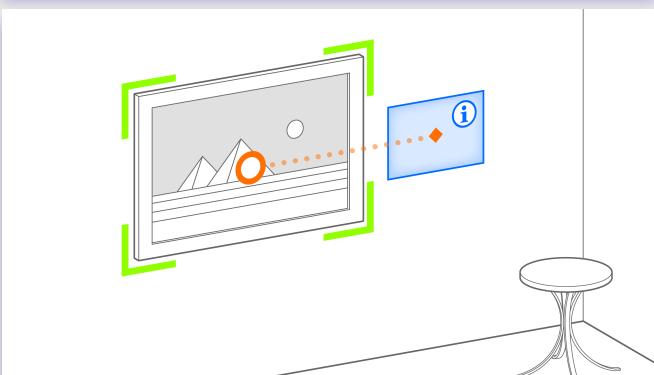
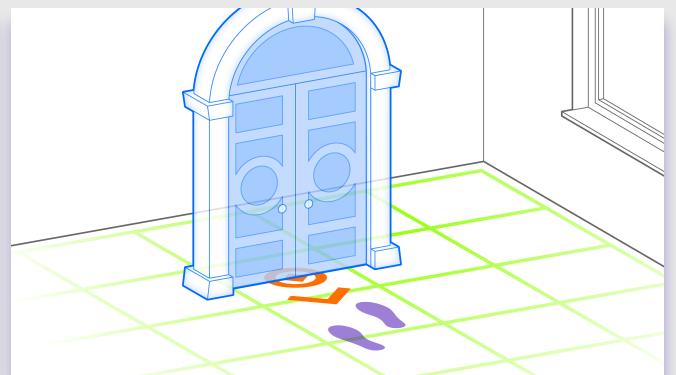
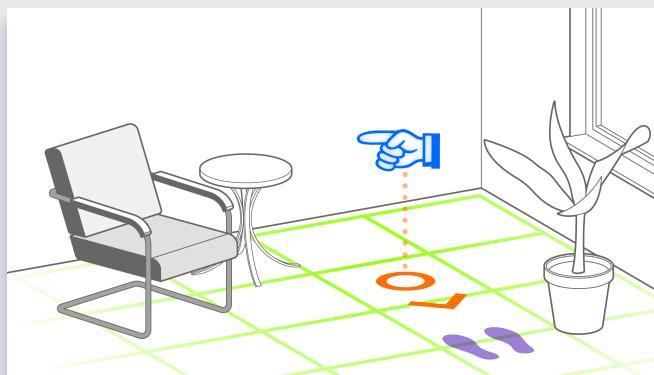
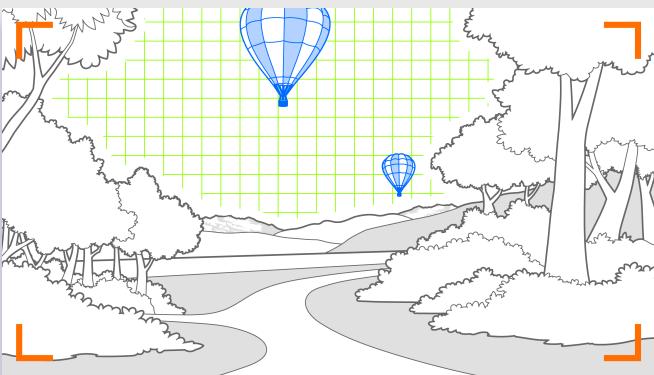
After the initial staging, users can interact with the virtual scene from their current position or move toward and around the staged content."

Diagram

```
on:command → do:add ahead 0 0 -1.2  
| 'red.box' +
```







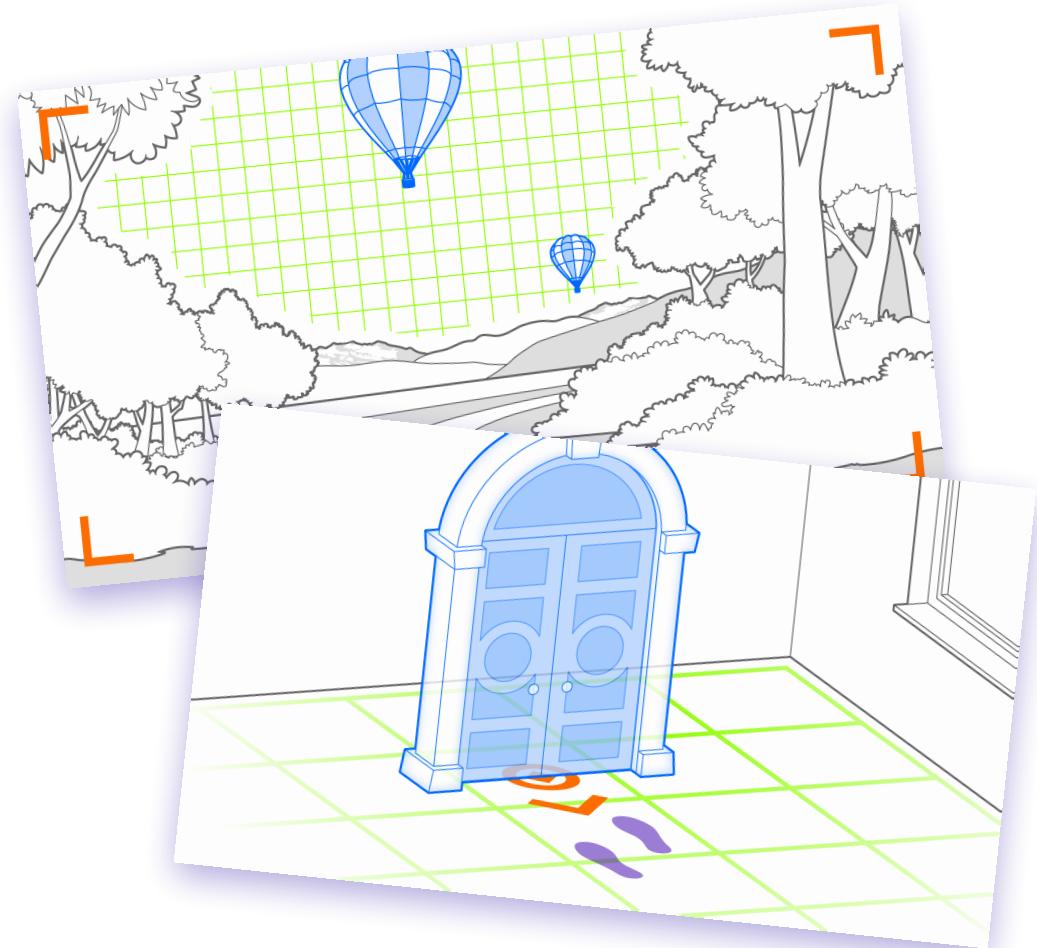
Challenges in Visual Design

- Reaching programmers and designers alike
- Creating a visual design language:
 - Precise
 - Distinctive
 - Accessible
 - Open
- Promoting discussion about AR Patterns, scenarios, and implementations

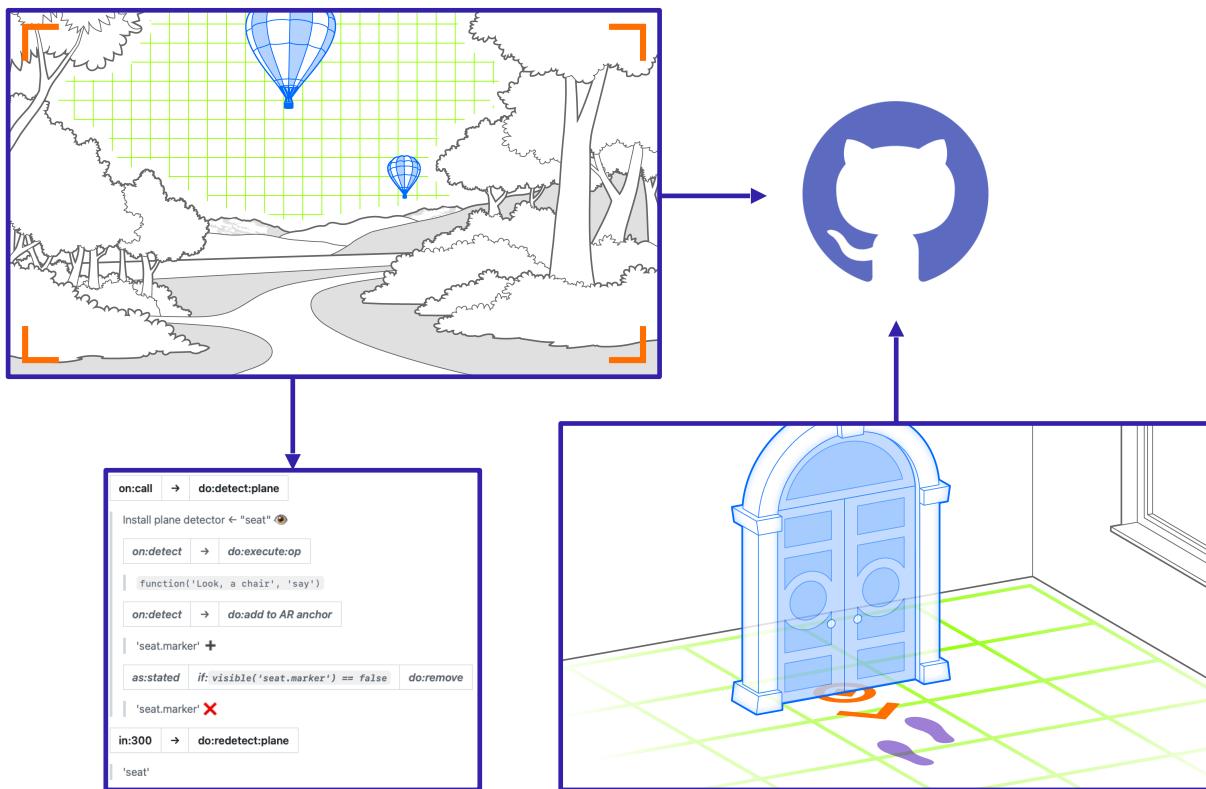
Reaching Designers

"My client wants me to design an AR experience.

I wish there was a **catalog** where I can browse all the different possibilities I didn't think about!"



Reaching Programmers

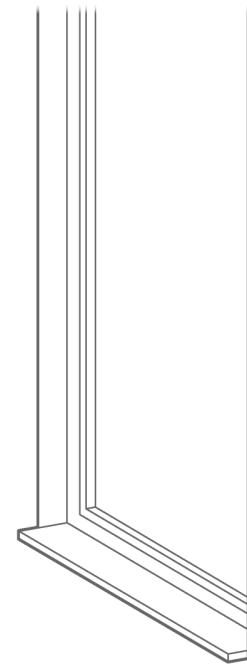


"My project partner wants me to program convincing AR scenarios.

*I wish there was a **repository** with resources and code examples in order to implement efficiently..."*

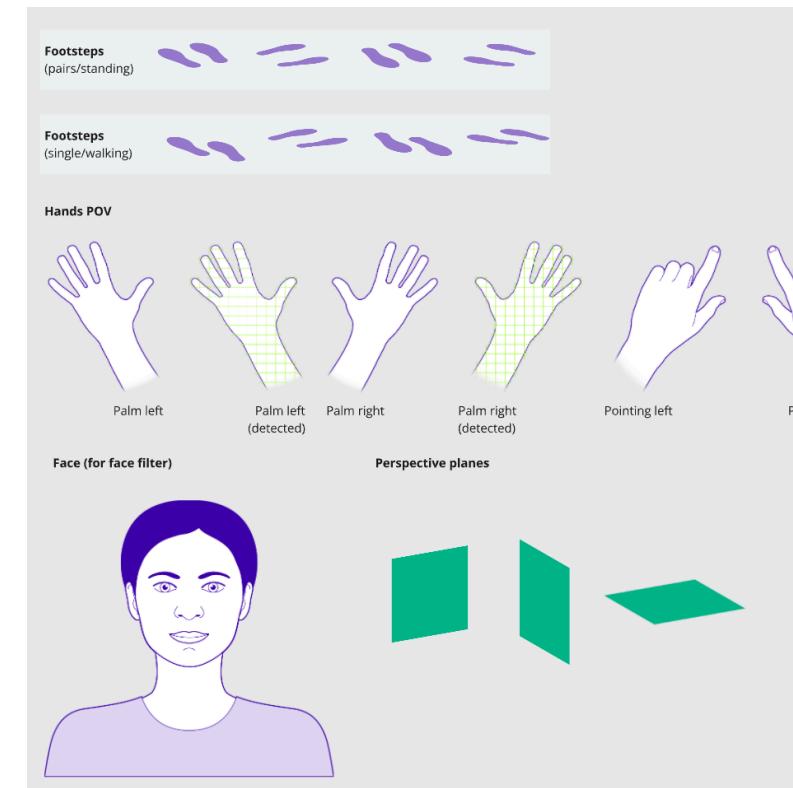
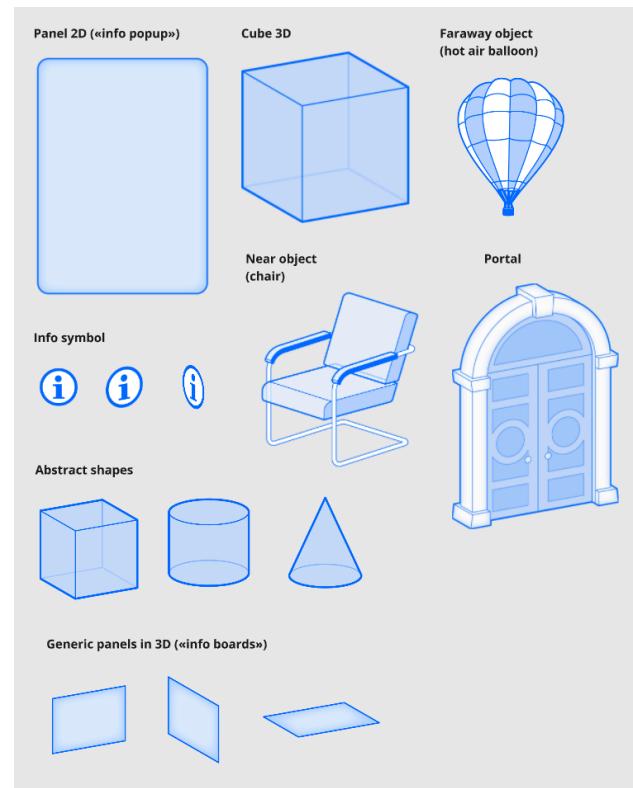
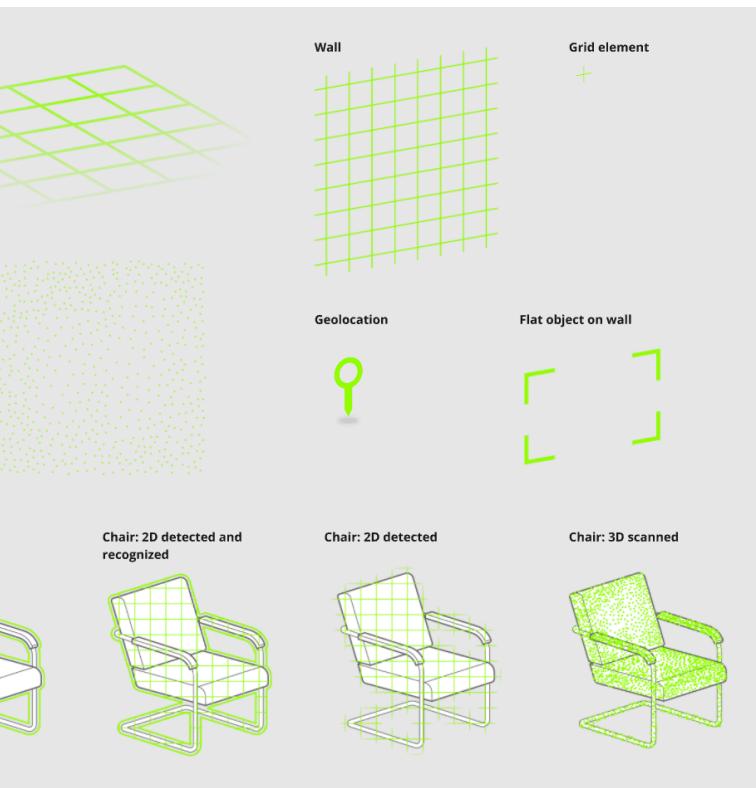
Creating a Visual Design System

Unified Perspective (parallel projection)



Creating a Visual Design System

Catalog of Instances

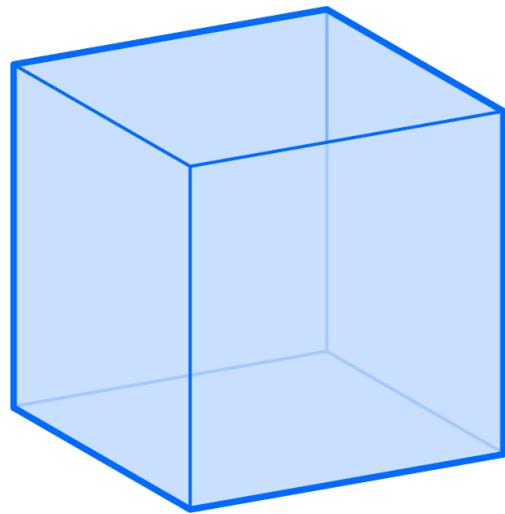


Creating a Visual Design System

Color Coding



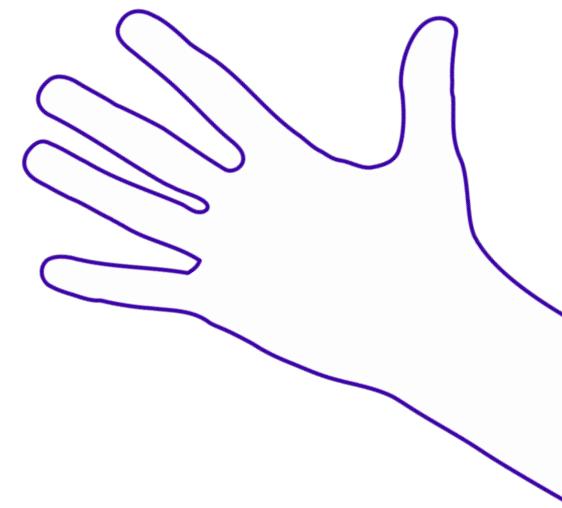
Detections



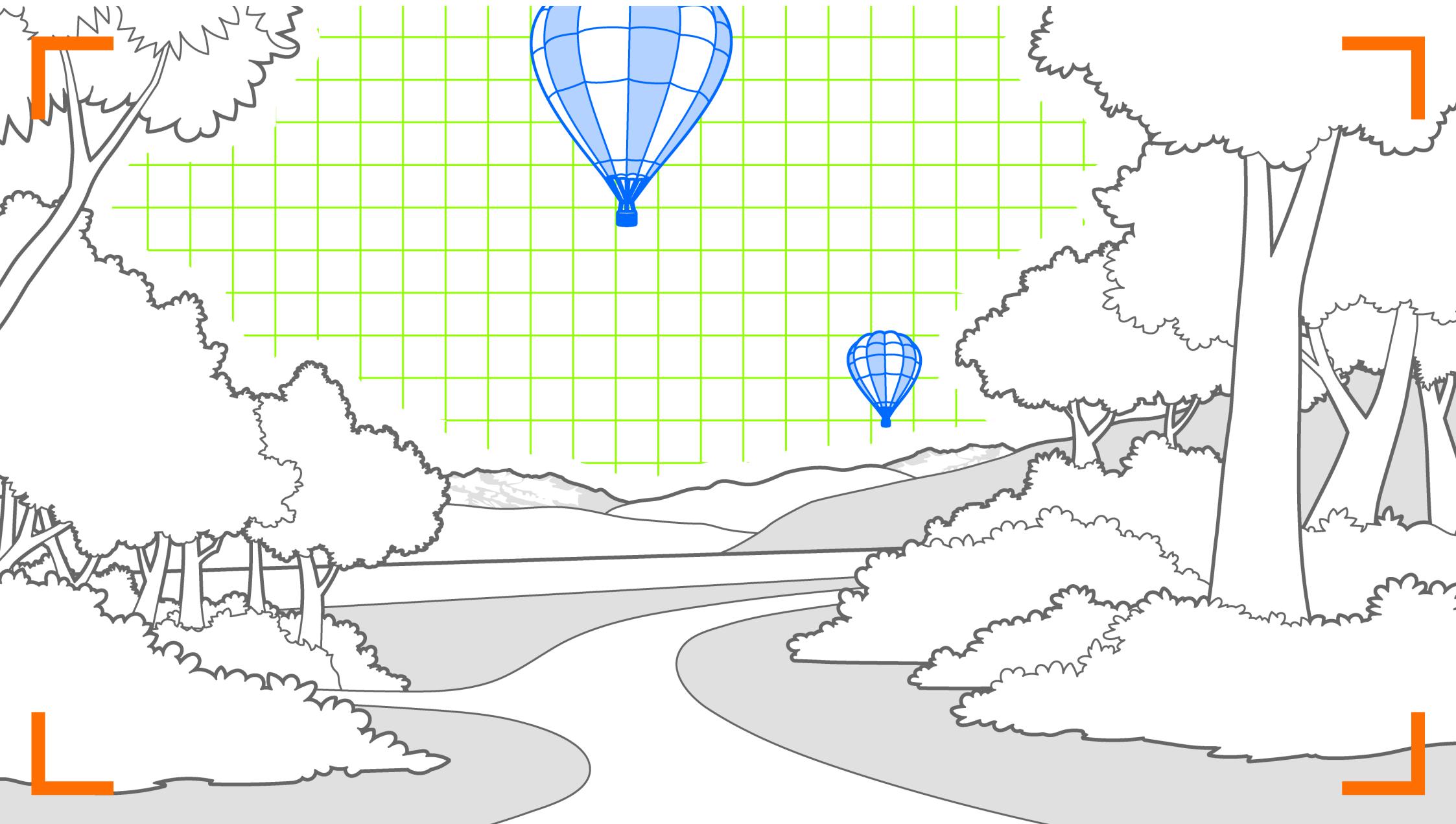
Augmentations

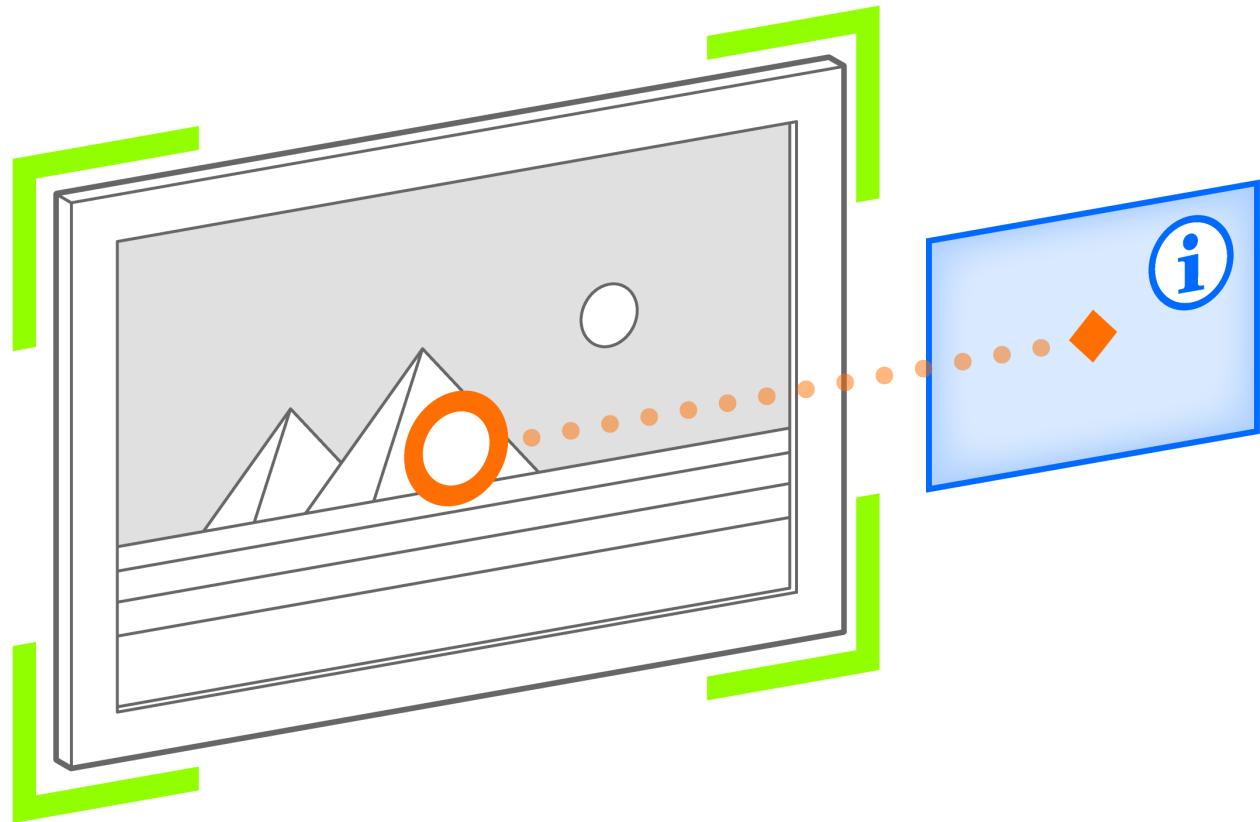


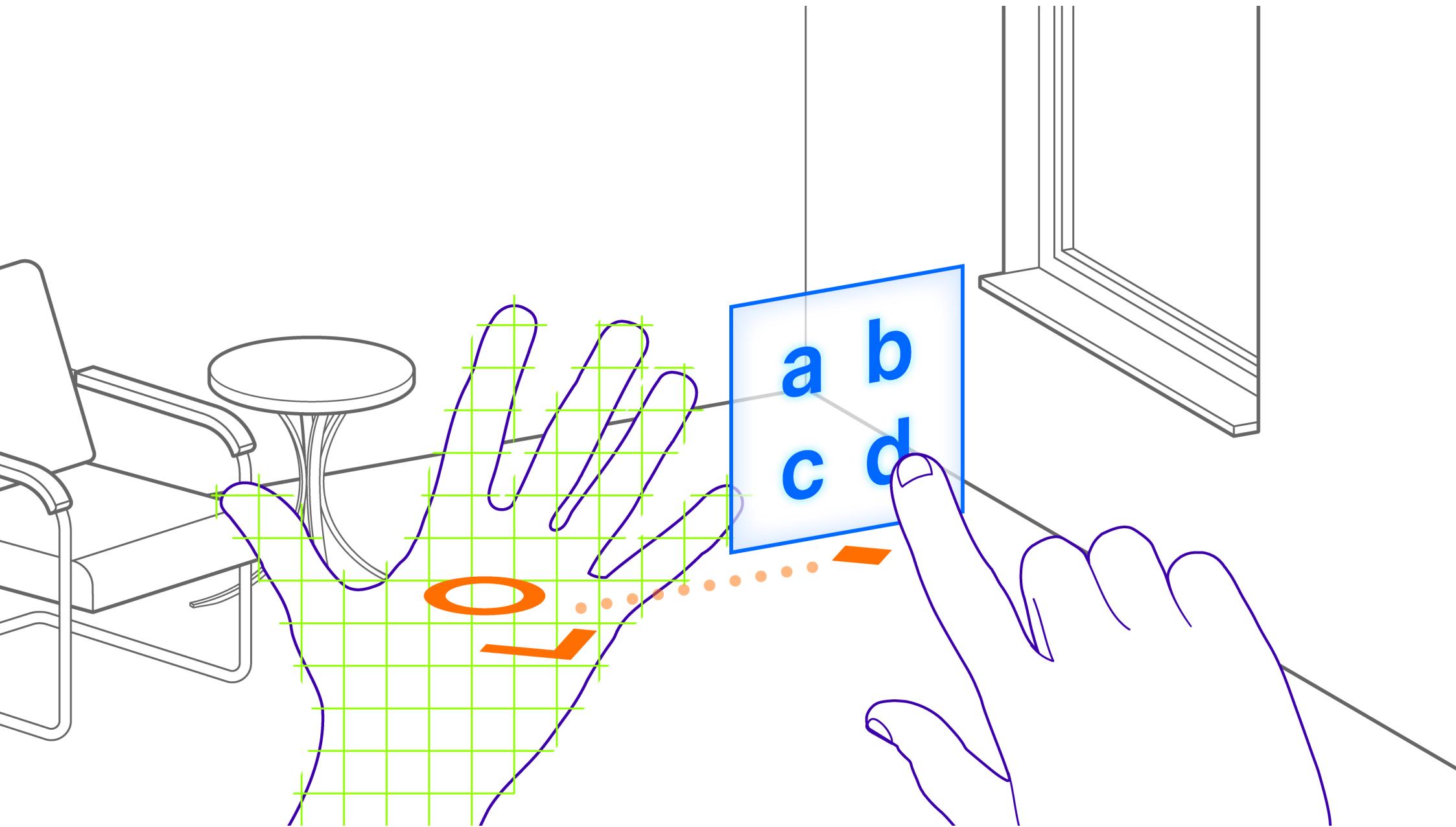
Anchors



User

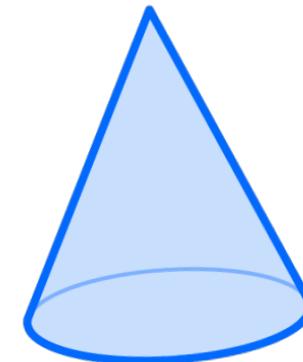




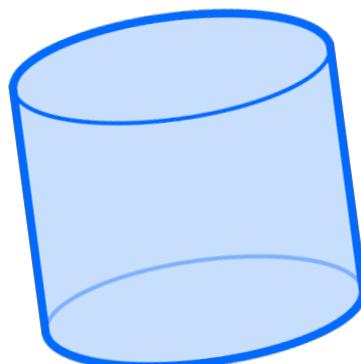


How to Use & Contribute

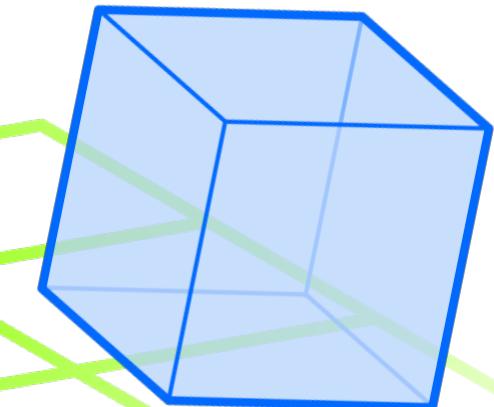
Supporting the AR Community



"I think I invented a novel way to implement AR!"



*I wish there was a **platform** where I can share my ideas and findings!"*



Design & Document AR Projects using

Illustration Toolkit

- Visual language and graphical template

Miro Board / SVG



AR Patterns Illustration Toolkit

AR Patterns Catalog

- Catalog of reusable solutions to design problems

arpatterns.dev



Event-Condition-Action Diagram

- Abstract pseudo code for reactive AR applications

Markdown styling



AR Scenarios

- Use-cases with source code in various technologies

github.com/ARpatterns



How to Use the Illustration Toolbox

1. Access the Miro Board or download SVGs on arpatterns.dev/illustrations
2. Choose a **backdrop**
3. Select your **props**
4. Describe the **detection**
5. Design the **augmentation**
6. Include the **user**
7. Specify the **anchor**
8. Enhance and **export**

milo AR Patterns Illustration Toolkit (demo) ⋮

03 ↗ Present Share

Examples from AR Patterns

1.1 Geolocated Re... 2.0 Segment Overlay 3.0 Area Enrichment 4.0 Captured Twin 5.0 Anchored Suppl...



7.0 Tag Along 8.1 Hand 9.1 Ahead Staging 10.0 Pass Through... 11.0 Staged Prog...

Scenario Parts

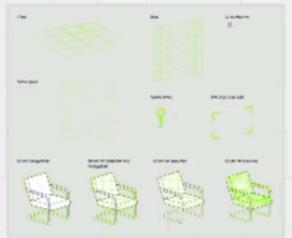
Start here!

Frame 16:9

Environments and real objects



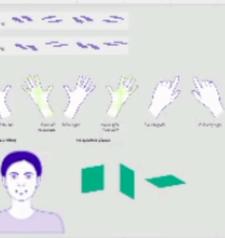
Detected instances



Augmented instances



User / Miscellaneous



AR Marker



46%

How to Communicate with AR Patterns

1. Checkout the **catalog**:

See arpatterns.dev or github.com/ARpatterns/catalog

2. Read the **paper**:

AR Patterns: Event-Driven Design Patterns in Creating Augmented Reality Experiences

3. Within your team **talk** using the terms of AR Patterns:

Discuss requirements and design options based on proven solutions

4. Write project and source code **documentation** using AR Patterns:

Describe implementation details referencing established solutions

5. If something is missing then **contribute**:

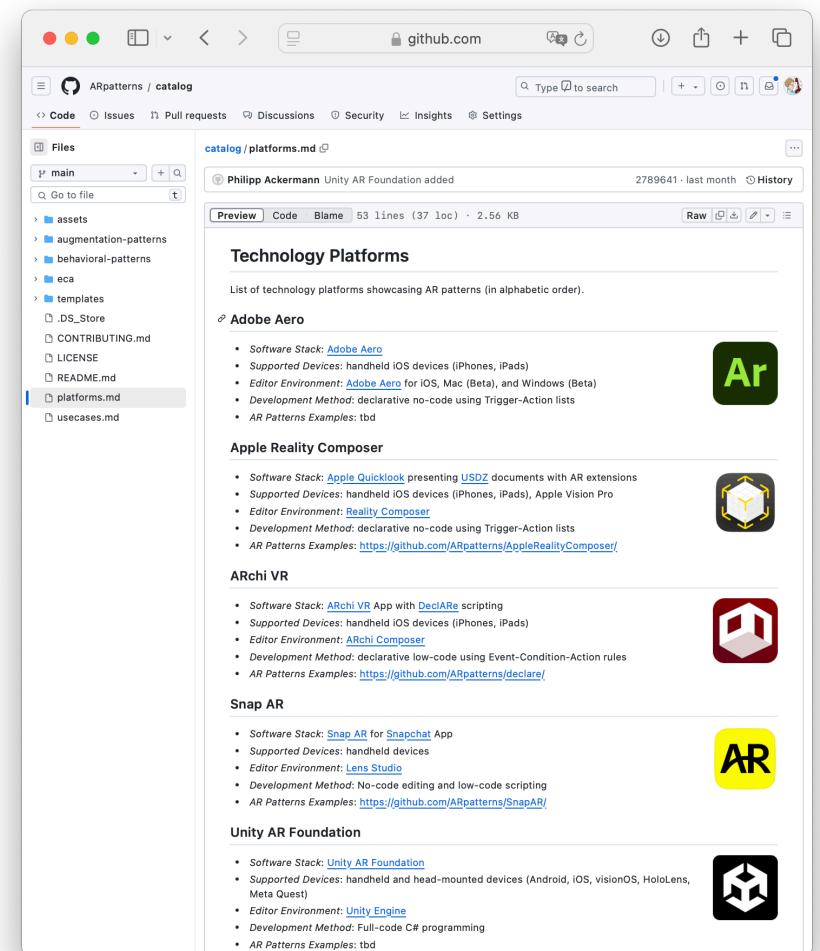
Develop and validate your idea for new AR patterns and share

How to Create an ECA Diagram

1. List the **3D assets** that will augment the real world.
These assets should then be handled by ECA rules.
2. Enumerate potential **events** that may occur during the AR session.
These events will trigger behavior and interactivity.
3. Arrange the enumerated items and events into **Event-Condition-Actions**.
4. Depict the **reaction** of the triggered action.
5. Write down the rule-reaction blocks with **Markdown**.
Apply the corresponding styling.
6. Integrate and render the **diagram** as embedded documentation.
E.g., in README files within GitHub

How to Get Inspired by AR Scenarios

- Browse through technology-specific **Use Cases** using and demonstrating AR Patterns
- Discover **source code** samples documented with scene illustrations, AR patterns, and ECA diagrams
- www.arpatterns.dev/scenarios.html
- This is work in progress...



Contribute Your Project

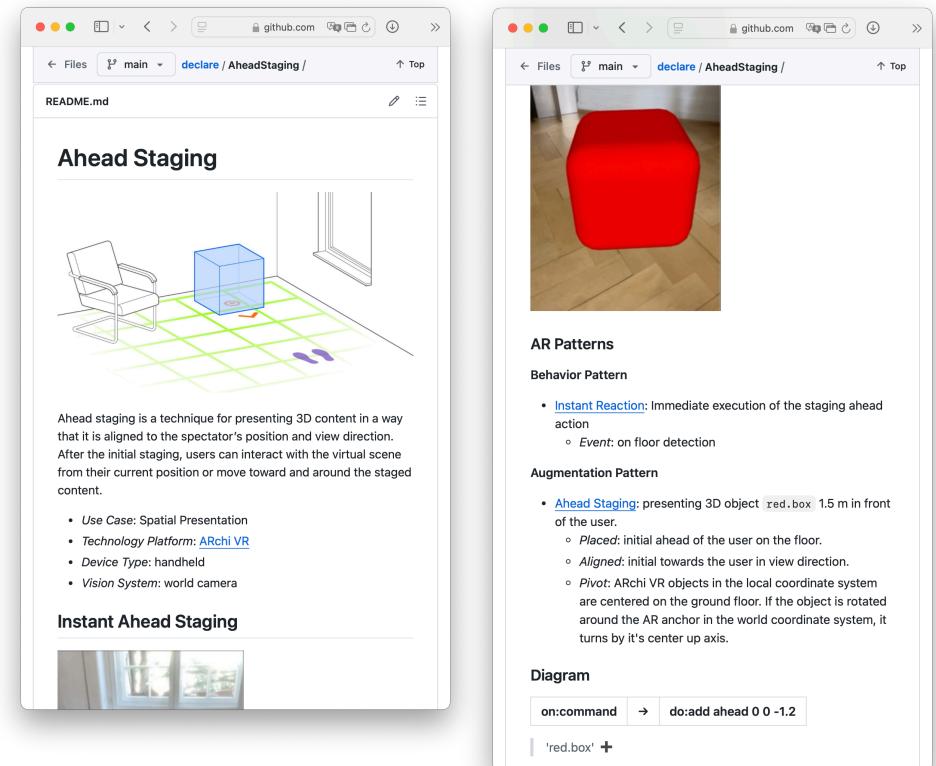
You have documented your project with

- Descriptions by using AR patterns
- Visual illustrations of the AR scene
- Explanation of code with ECA diagrams

Then you might contribute your project

- as an AR scenario / use case
- for a specific technology platform
- with source code

Link your project on
<https://github.com/ARpatterns>



See template at github.com/ARpatterns/catalog

Conclusions

Checkout and apply proposed solution building blocks in your AR projects

- Illustration Toolbox
- Catalog of AR Patterns
- ECA Diagram
- AR Scenarios

Looking forward to community involvement

- Please feel free to get in contact and to contribute

Links

- www.arpatterns.dev
- github.com/ARpatterns