

# A more immersive HoloLens experience

21 July 2017 | **Augmented & Virtual Reality, User Experience** | [Michael Sattler](#)

**Reading time:** 6 minutes

Recently we've been sorely feeling the need for better interaction mechanisms on the HoloLens, as my colleague Jörg already wrote in his blog post on [HoloLens UI](#). Even if you get used to the AirTap and the English accent that HoloLens expects for voice commands (and that's two big if's), both the AirTap and voice commands always feel tedious over time. Not to mention industrial service scenarios, for example, where you need to trigger some interaction in a noisy environment with both your hands occupied with something else. Therefore, we wanted to come up with better approaches to user interaction that actually made use of three-dimensional space instead of just slapping the 2D Point & Click pattern onto 3D holograms.

## Concept collection

At this year's AR/VR Camp at Zühlke, we put our heads together and gathered all the possible mechanisms in 3D space that we could come up with. Here are some examples for possible user input methods that came to our mind:

- Just omit the AirTap and trigger an action by **simply looking at an object** for X seconds.
- Interaction through proximity: React to the user's head **being close to a certain point in space**. This could be used to trigger a distinct action or present an entirely different toolset to the user.
- Initialize an interaction already when the user starts **moving towards some location**.
- **Nod/shake your head** to confirm/decline an action.
- Use **custom gestures**, for example the Minority Report trademark "swipe" gesture or a "thumbs up" gesture.

For the user to learn these mechanisms, we'd also need some visual or audible cues. Of course, there's always the two obvious options of spoken advice and just displaying text, but why not try to convey an object's affordance in a more subtle way? New gestures would certainly still need to be taught "the hard" way, but telling the user to simply keep his gaze focused on an object or moving his head closer could probably be done in a more elegant manner:

- Start some kind of progress indicator when the user's gaze enters an object or area.
- Show some glow effect or object that piques the user's curiosity and makes him move closer.
- Playback spatial sound to direct the user's attention.
- Show "breadcrumbs", wave patterns or just navigation arrows on the floor that tell the user to move somewhere.

## Time for some coding!

In order to validate our ideas, a colleague of mine and myself went on and did some coding. Of course, having just three days of time limited our options – and having the amusement park “Europapark” just outside the hotel doors also didn’t exactly contribute to the time invested into implementation. So we just picked the first three of the ideas for interaction mechanisms mentioned before.

The **Proximity Interaction** pattern was pretty straightforward to do. For the proximity trigger, we just needed to check the user’s (i.e. the main camera’s) position on each frame and fire an event once it was close enough to an object. We combined this with the “Approach Interaction” pattern by calculating a “proximity percentage” when the user moved closer to a point of interest and letting it decrease when the user stayed still or moved away. Based on this approach percentage, we used an animation at the point of interest that grew larger and more intense when the user moved closer, up until he reached the trigger point. Combined with some object at the point of interest to initially make the user move closer, this pattern worked quite well in the end.

The second pattern we implemented was the **Gaze Only** interaction pattern. The basic idea is fairly easy to get working: Check whether the user is gazing at the target object on each frame, gradually increase a percentage over X seconds while he is and trigger the action once we hit 100%. For the purpose of affordance, we’d use a circular progress indicator that started once the user gazed at the object. A cancellation event would be triggered once the user’s focus left the object.

But what about small targets? What about ongoing events such as spoken explanation on the focused object? Would be rather annoying if the explanation started over each time the user looked away briefly, right? To smoothen out the experience, we came up with the idea of just counting down the focus percentage while the user looked away and only stopping the interaction if the object stayed out of focus. The result was a lot more satisfactory to use.

The last and by far most complex interaction we tried to implement was the **Nod Confirmation**. We noticed early on that shaking one’s head while wearing the HoloLens was pretty uncomfortable, with the nose piece slapping against the nasal bone, so we stuck with just the nod. However, even there we faced several challenges from the get-go: How to avoid interpreting sneezing as a nod? How to tell a nod directed to the app apart from a nod directed at someone else? What about people nodding several times, or slower, or more intensely? Eventually we went with a learning component: Let the user show the app once how he nods, so the app can recognize this exact nod going forward.

It took us quite some time to get this working. At first, we looked for head acceleration, but that turned out to be far too complex. Instead, we just tracked the head's angle over time to record and compare the pattern. This worked reasonably well in the end, although we had to fiddle a lot with epsilon values on the movement angles and time frames in order to avoid too many negatives and still get no false positives.

## **Conclusion**

At the end of our AR/VR Camp, we found that it was **well worth experimenting** with new interaction patterns. Aside from the ideas we had collected for future projects, we also extracted **re-usable assets** from our experiments afterwards, so we can easily reproduce these patterns in the future.

Of course, there's still a lot of work to do. Especially the more complex interaction patterns such as Nod Confirmation (not to mention custom gestures) **need a lot of fine-tuning** in order to be really intuitive, feel good and work reliably. But after all, that's where the fun with building for HoloLens is, right? We're trailblazers on the path to a better and more immersive holographic user experience! So let's continue walking off the beaten path and keep exploring!