**Erstmal quasi ungefiltert**

**Wekit-Quelle:**

* Haptic Feedback
* Directed Focus/Gaze
* Movement Instruction (Stand here)
* Autopause/-play
* Gaze to select/Cursor („activate spot by staring“)
* Shared pointer (Multi-user)
* Voice commands

**Andere Quellen**:

* OIV und was damit zusammenhängt
* Information filtering, je nach Bedürfnissen
* POI
* Gesture-Based Interaction
* Fiducial (auch nochmal bei Dunnleavy gucken)
* Audio Augmentation (Ternier)

**Name**: Haptic Feedback

**Forces/Problem**: User may miss visual or audio feedback; user does not receive appropriate feedback when “touching” augmented objects

**Feature/Solution**: Use haptic feedback either accurately or generally

**Examples**: Vibrotactile feedback is a standard feature of various game controllers. The Myo Armband can give such feedback while keeping the user’s hands free. The NormalTouch and TextureTouch devices are able to give somewhat accurate haptic representations of virtual objects.

**Effects/Consequences**: User may be more responsive to *Haptic Feedback*, especially if their attention is not on the augmentation. The AR device has to implement haptic technology, or other devices have to be added – if the user has to hold the feedback device in one or more hands, their freedom of movement will be decreased. (Inappropriate) *Haptic Feedback* may break immersion.

**Sensors**: If accurate feedback corresponding to body movements is to be given, hand or full-body tracking must be used (e.g. with Leap, Kinect, …)

**Name**: Directed Focus

**Forces/Problem**: How do you direct a user’s attention to something when they have full control of their view?

**Feature/Solution**: Use an icon to indicate something the user should direct their attention towards; affix the icon to the screen, pointing towards the object of interest if it is not currently visible.

**Examples**:

**Effects/Consequences**: *Directed Focus* icons can make for a more structured AR experience and avoid user confusion. However, they may obstruct other, possibly important elements and cause screen clutter. If multiple focus points exist at a time, *Information Filtering* may prove necessary to avoid this. To communicate to the user where the system currently assumes their focus or gaze is directed, use a *Gaze Cursor*.

**Sensors**: The system always needs to be aware of the position of the focus point relative to the user or the user’s gaze. How this works is dependent on the AR device, but generally, IMUs will be most useful for this.

**Name**: Directed Movement

**Forces/Problem**: Some applications may require the user to move to certain locations. How do you communicate these?

**Feature/Solution**: Display an icon at the target location; affix the icon to the screen, pointing towards the object of interest if it is not currently visible.

**Examples**: On a macro-scale, a *Point of Interest* can serve as a form of *Directed Movement* if permanently displayed on a map and/or in direction.

**Effects/Consequences**: *Directed Movement* icons can make for a more structured AR experience and avoid user confusion. However, they may obstruct other, possibly important elements and cause screen clutter. If multiple focus points exist at a time, *Information Filtering* may prove necessary to avoid this.

**Sensors**: The system always needs to be aware of the position of the focus point relative to the user’s position and possibly gaze. How this works is dependent on the AR device; vision-based systems can use IMUs, while location-based systems can also utilize e.g. GPS sensors.

**Name**: Auto-play

**Forces/Problem**: How can events in Augmented Reality be triggered without requiring or despite varying kinds of user input?

**Feature/Solution**: Some events may work best if started, paused or stopped automatically. Automatic here refers to a lack of directed efforts to create input, for example based on location or gaze. It is possible to automatize only parts of any event. Results of *Directed Focus* or *Directed Movement* are not considered *Auto-play* as the user is clearly instructed to perform actions.

**Examples**: Applications using *Points of Interest* can use varying levels of *Auto-play*: They either immediately make the content accessible (as with the audio recordings in Ternier, De Vries et al.’s field trip application), they automatically retrieve some content or they simply make the user aware of the existence of such content.

**Effects/Consequences**: Content presented automatically can allow the user to avoid tunnel vision and make using the application more pleasant as less effort is required. Automatic warnings can prevent injury or incorrect use. If too much information is communicated too suddenly, *Auto-play* could also confuse or disorient the user or cause screen clutter. If there is no apparent way to close unwanted content, the user may become frustrated.

**Sensors**: What sensors are necessary for this is entirely dependent on the application. Automatic content may for example be based on position, gaze/orientation, audio input, or fiducials.

**Name**: Gaze Cursor

**Forces/Problem**: In an environment with multiple interactive objects, how do you select which ones to apply actions to?

**Feature/Solution**: Base actions on the user’s gaze or orientation and communicate this process graphically as with a computer’s cursor. The *Gaze Cursor* can be used either for only selecting objects or for completing actions, for example based on duration of gaze.

**Examples**: Various HoloLens applications utilize *Gaze Cursors*. In a broader sense, the entirety of the scene taken in by the camera in a marker-based AR system can be considered to be under the *Gaze Cursor*.

**Effects/Consequences**: Using a *Gaze Cursor* should make interactions more clear to the user if the cursor is accurate and consistent. Inaccurate, inconsistent, or lagging cursors may however make interaction frustrating and unreliable. A graphical representation of the user’s gaze will take up screen space – one should ensure that the object of the user’s interest is not obscured by the cursor. There might also be a danger of tunnel vision.

**Sensors**: The system must be aware of the user’s gaze, which can be achieved by tracking their head movements. Eye tracking can provide a higher degree of accuracy if available.

**Name**: Shared Pointer

**Forces/Problem**: How can multiple people in an Augmented Reality environment communicate efficiently?

**Feature/Solution**: Use (gaze) cursors which are also visible to the other players or allow users to leave markers at set points.

**Examples**: Nilsson, Johansson & Jönsson used *Shared Pointers* in the form of settable icons in all versions of their cross-organizational collaboration application. The first version also had a form of cursor but this was dropped in favor of participants simply pointing with their hands as the application was centered on a small map and participants were unlikely to direct their attention elsewhere.

**Effects/Consequences**: *Shared Pointers* can allow users to communicate more clearly. This approach requires either a *Gaze Cursor* or some other kind of directed user input. For consequences of users setting markers, see *Directed Focus* and *Directed Movement* (though depending on your application it may not be necessary for the markers to be pointed towards when off-screen).

**Sensors**: If a *Gaze Cursor* is used, the sensors related to it are needed. If only virtual objects can be selected, special sensors may not be necessary. If real objects can also be targeted, the relative positions and orientations of the users must be taken into account, derived from e.g. IMU data.

**Name**: Voice Commands

**Forces/Problem**: How do you allow user input while keeping their hands free?

**Feature/Solution**: Have the user perform actions by speaking appropriate phrases.

**Examples**: *Voice Commands* are a standard feature of HoloLens applications.

**Effects/Consequences**: If implemented well, *Voice Commands* can greatly enhance user experience. However, if phrase recognition does not work well or the phrases are too complicated, the user may become frustrated. If the application is to be used in shared spaces, loudly talking and recording audio without consent may face social acceptance issues; the system could also pick up other people’s voices and act accordingly, against the user’s wishes.

**Sensors**: One or more microphones.

**Name**: Obscured Information Visualization

**Forces/Problem**:

**Feature/Solution**:

**Examples**:

**Effects/Consequences**:

**Sensors**:

**Name**: Information Filtering

**Forces/Problem**:

**Feature/Solution**:

**Examples**:

**Effects/Consequences**:

**Sensors**:

**Name**: Point of Interest

**Forces/Problem**: How can information be provided to users in a location-based system? How can information be bound to locations that are not reasonably accessible to developers?

**Feature/Solution**: Bind information to location data, automatically making it available to the user upon getting within a predefined range and allowing you to direct users to nearby Points of Interest.

**Examples**: Several Augmented Reality browsers have implemented Point of Interest approaches. Ternier, De Vries, et al. (2012) utilized it to guide students on a field trip.

**Effects/Consequences**: Points of interest either can not overlap or the system needs a method to handle such overlap. Constantly gathering information about user location and comparing it to points of interest may consume a lot of energy, which could negatively affect the user. If the user is directed towards points of interest, care must be taken to avoid screen clutter and information overload. A degree of precision is lost if only location data is used.

**Sensors**: Location technology such as GPS sensors; IMUs in vision-based systems with a local coordinate system.

**Name**: Gesture-based Interaction

**Forces/Problem**:

**Feature/Solution**:

**Examples**:

**Effects/Consequences**:

**Sensors**:

**Name**: Fiducial

**Forces/Problem**:

**Feature/Solution**:

**Examples**:

**Effects/Consequences**:

**Sensors**:

**Name**: Audio Augmentation

**Forces/Problem**:

**Feature/Solution**:

**Examples**:

**Effects/Consequences**:

**Sensors**: