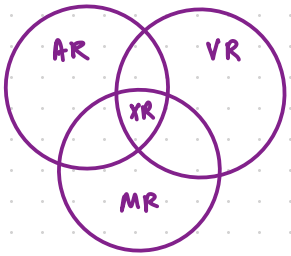
Variety of XR Devices

Common XR Devices



XR can be thought of as a collection of 3 categories
✓AR ✓VR ✓MR

VR
✓Meta Quest
✓HTC Vive headsets

AR
✓ magic leap
✓ Snap AR spectacles
↗ still mostly consumed by smartphones (their cameras)

MR
✓microsoft's hololens

Most devices were some form of head mounted display (HMD)
HMDs are the commonly perceived device when consuming XR experiences

VR HMD
↳ PCVR → connected to PC & maybe external sensors
↳ standalone VR or AIO (all in one) → no cables, has inside out tracking

Hardware of a HMD

Quest 2 → display screen
→ 2 magnifier glasses
→ corrective lenses? for those wearing glasses (e.g. glasses-free)

→ specialized controllers that emit infrared lights and have motion tracking sensors to detect those lights

→ motion tracking sensors in the headset themselves

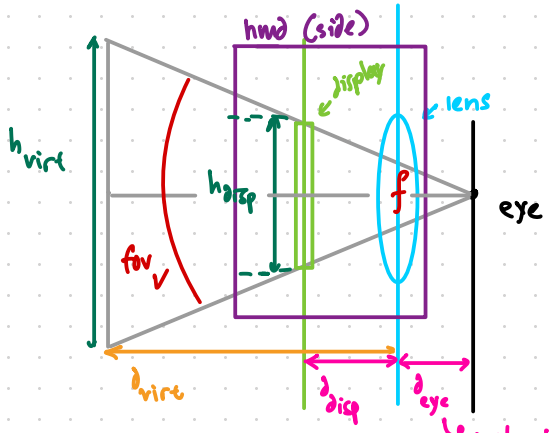
→ 'cameras all over'

→ battery, speakers, CPU, GPU, motherboard, other sensors on smartphones...

HMD and smartphone share many similar components

magnification components

image formation process google cardboard



f = focal length

↳ also known as eye relief

Gaussian Thin Len Formula

$$\frac{1}{d_{virt}} + \frac{1}{d_{disp}} = \frac{1}{f}$$

$$h_{virt} = M h_{disp}$$

$$d_{virt} = \left| \frac{1}{\frac{1}{f} - \frac{1}{d_{disp}}} \right|$$

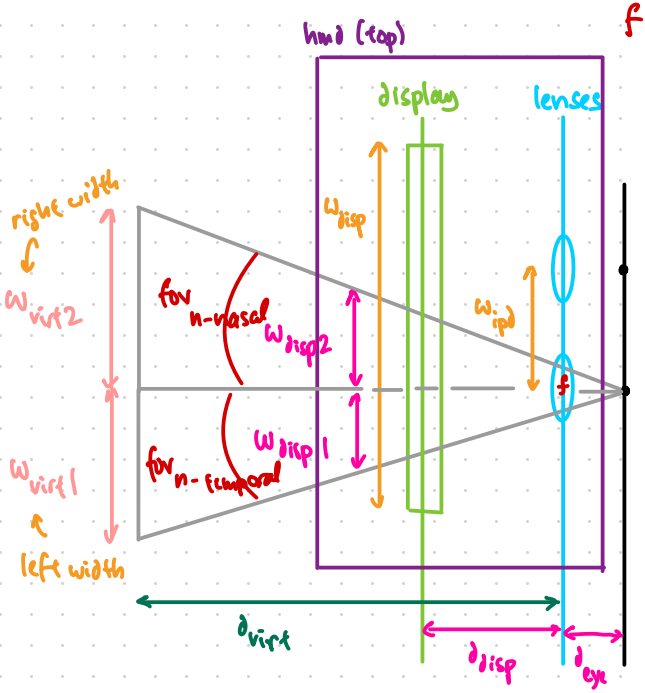
basically the view frutum and its view, etc

symmetric

magnification $M = \frac{f}{f - d_{disp}}$

so simple

$$fov_{vertical} = 2\tan^{-1}\left(\frac{M\left(\frac{n}{2}\right)}{d}\right)$$



f = focal length

ipd = interpupillary distance

↳ the better you can match this to your real eyes, the better your viewing experience will be

↳ for the left eye, the left width will be larger than the right

vice versa for right

↳ horizontally asymmetric view frustum

diff. images formed for left and right eyes

$$w_{virt2} = M\left(\frac{w_{ipd}}{2}\right)$$

$$w_{virt1} = M\left(\frac{w_{disp} - w_{ipd}}{2}\right)$$

$$z = d_{disp} + d_{eye}, \quad d = d_{virt} + d_{eye}$$

$$w_{disp2} = z\left(\frac{w_{virt2}}{d}\right)$$

$$w_{disp1} = -z\left(\frac{w_{virt1}}{d}\right)$$

with all these values it will change the view frustum

$$fov_n = fov_{n\text{-}nasal} + fov_{n\text{-}temporal}$$

$$= \tan^{-1}\left(\frac{M\left(\frac{w_{ipd}}{2}\right)}{d}\right) + \tan^{-1}\left(\frac{M\left(\frac{w - w_{ipd}}{2}\right)}{d}\right)$$

differentiating between nasal & temporal, helps determine binocular fov and monocular fov

combine visual span from both eyes ↳

↳ combined visual span but seen individually separately seen by both eyes

lens distortion algorithms

introduced to counter the natural distortion of the lenses

the further the point is from the centre, the more it needs to be shifted to counter the distortion introduced by the curved lenses

other than later image distortion, chromic aberrations — colour artifacts caused by different wavelengths of light refracting differently as they pass through the lens

other H/W

- wearable motion trackers
- eye tracking
- foveated rendering ⇝ improving image quality in the center of a user's field of vision by reducing the quality of their peripheral vision

software

✓ usually real-time simulations

✓ usually include
- ↳ rendering → drawing all graphic components on the scene
- ↳ physics → takes care of math needed to simulate real life physics
- ↳ input → processes input and how to do it based on what hardware is available
- ↳ audio → responsible for playing and handling audio
- ↳ AI → facilitate the development of behaviours in animated gameobjects

✓ stuff available in third party items
- ↳ pathfinding algorithms ⇝ A* and the like
- ↳ tools to make finite state machines or behavior trees or reinforcement algorithms

maybe other components....

ECS → to organize all your entities and components