

(index.html)

# Coordinate systems

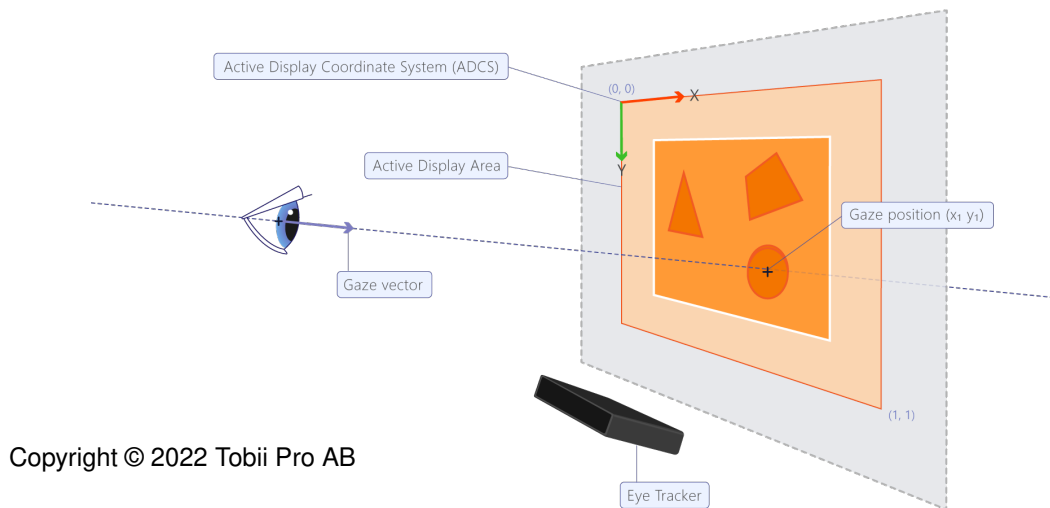
In the TobiiPro SDK you will come across a few different coordinate systems. In order to be able to correctly interpret coordinates delivered from the SDK it is important that you understand the difference between these coordinate systems.

In the description of the coordinate systems, the word *user* denotes the person whose eyes are being tracked by the eye tracker.

## Active Display Coordinate System (ADCS)

All gaze data are mapped into a 2D coordinate system aligned with the Active Display Area. When using an eye tracker with a monitor, the Active Display Area is the display area excluding the monitor frame. For eye trackers used without a monitor, the Active Display Area is the area used to show the calibration points when doing the calibration of the eye tracker to the user's eyes.

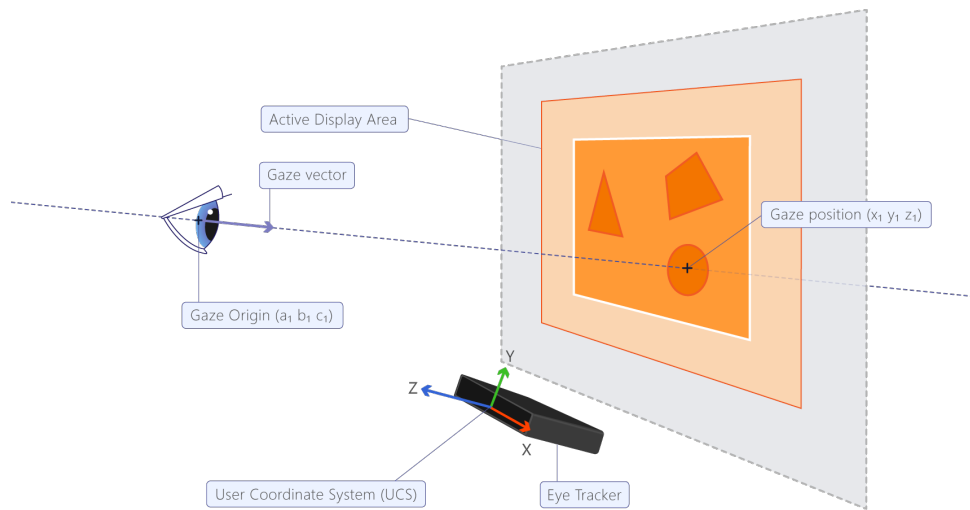
The origin of the Active Display Coordinate System is the upper left corner of the Active Display Area. The point (0, 0) denotes the upper left corner and (1, 1) the lower right corner of it.



## User Coordinate System (UCS)

Most data, available from screen based Eye Trackers from Tobii, that describe 3D space coordinates are given in the so-called User Coordinate System or UCS for short. The UCS is a millimeter-based system with its origin at the center of the frontal surface of the eye tracker.

The coordinate axes are oriented as follows: the x-axis points horizontally towards the user's right, the y-axis points vertically towards the user's up and the z-axis points towards the user, perpendicular to the front surface of the eye tracker.

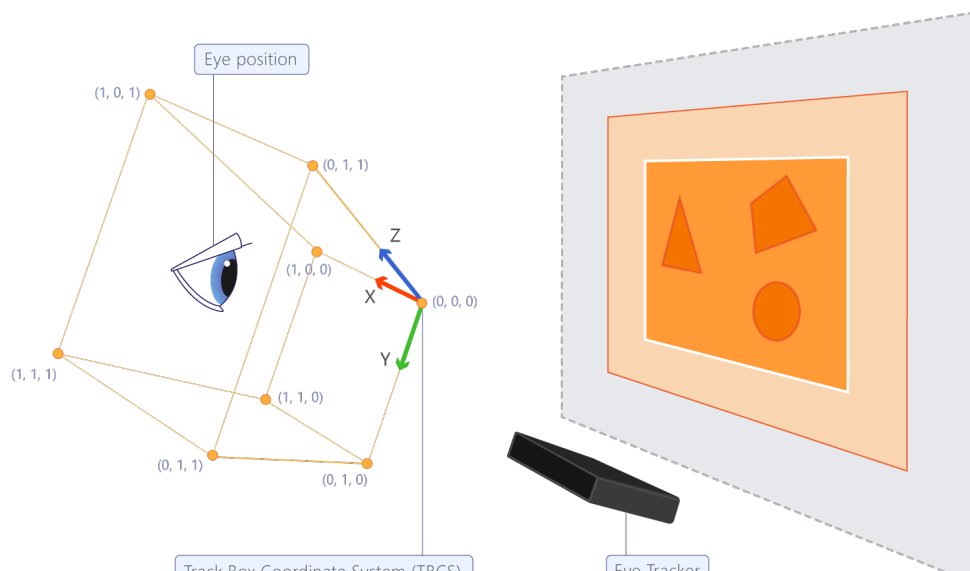


## Track Box Coordinate System (TBCS)

The track box is the volume in which the eye tracker is theoretically able to track the eyes. Thus, the user is able to move the head freely and still remain trackable as long as the eyes remain within the box. In the SDK there are methods to find out the size and position of the track box volume, but also where the eyes are within it. The coordinate system used to describe the position of the eyes within the track box is called the Track Box Coordinate System (TBCS).

The TBCS is a normalized coordinate system, which in this case means that the opposite corners of the trackbox has coordinates 0,0,0 (the upper, right corner closest to the eye tracker), and 1,1,1 (the lower left, furthest away, corner) respectively.

The coordinate axes are oriented as follows: the x-axis points horizontally towards the user's left, the y-axis points vertically towards the user's down and the z-axis points towards the user.



[\(index.html\)](#)[Track Box Coordinate System \(TBCS\)](#)[Eye Tracker](#)

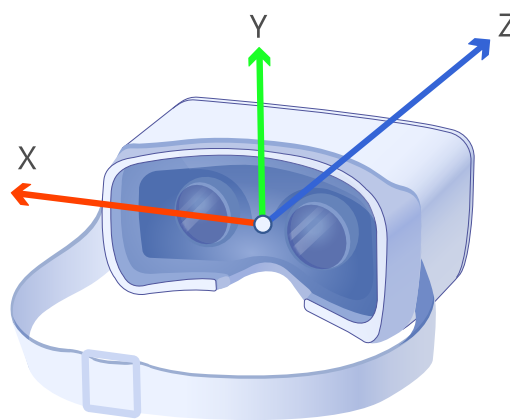
## HMD Coordinate System

All data, available from HMD eye trackers from Tobii, that describe 3D space coordinates are given in the so-called HMD Coordinate System. The HMD Coordinate System is a millimeter-based system with its origin at a point in between the lenses of the HMD device, at equal distance from each lens' center.

The coordinate axes are oriented as follows: looking from a wearer's point of view, the x-axis points horizontally towards the left side, the y-axis points vertically upwards, and the z-axis points (forward) away from the HMD, perpendicular to the lenses of the HMD tracker.

**Note:** When working with HMD eye trackers it is important to understand how the eye tracker's coordinate system relates to the headset's position in a virtual 3D world (such as Unity). *If you are using the Tobii Pro Unity prefabs, you do not have to worry about this as it is all taken care of by the prefabs themselves.* But, if you are using the "raw" .NET binding, or another programming language, the section below is recommended reading.

The eye tracker's coordinate system has its origin at a physical location within the headset, and all coordinates used by the eye tracker are in relation to that origin. Likewise, the location of the headset in a virtual world is based on a physical location (origin) within the headset. Unfortunately, these two locations do not always coincide. When there is a difference between the two locations, this offset needs to be taken into account when using gaze data in the virtual 3D world. For example, when using the SteamVR Plugin in Unity together with a HTC Vive, the origin of the HTC Vive headset in the virtual 3D world is *not* the same as the origin of the eye tracker's coordinate system. Read the Useful tips and hints for Unity ([./unity/unity-getting-started.html#TipsAndHints](#)) section for details.



## HMD Tracking Area

Since the HMD eye trackers allow for very little movement in the z-axis (of HMD coordinate system), the equivalent of the "track box" of the eye tracker (see section about TBCS above) is described in only two dimensions, x and y. One such two dimensional "track box", or "HMD Tracking Area", is defined for each eye.

The HMD Tracking Area is a normalized 2D coordinate system with its origin (0, 0) in the top right corner (from the wearer's point of view), and (1, 1) in the lower left corner.

The main use case for the HMD Tracking Area coordinates is to help position the headset correctly on the person being tracked. Ideally, each eye should be positioned in the middle of the tracking area, at coordinates (0.5, 0.5).

