

# Tilt Five™ Design Tips & Guidelines

Version: May 19, 2022

Welcome to Tilt Five! This document provides tips and guidelines for building amazing augmented reality experiences for the table using Tilt Five.

While this document is not a set of rules or a Technical Requirements Checklist (TRC) itself, these guidelines are based on observed best practices and will help ensure performance and quality that is consistent across all games and apps on the platform.

Note that Tilt Five has created a TRC for use by all developers looking to formally support Tilt Five in their games. The TRC is consistent with many of the points and ideas provided here, and can be downloaded on [docs.tiltfive.com](https://docs.tiltfive.com).



## Basic Guidelines

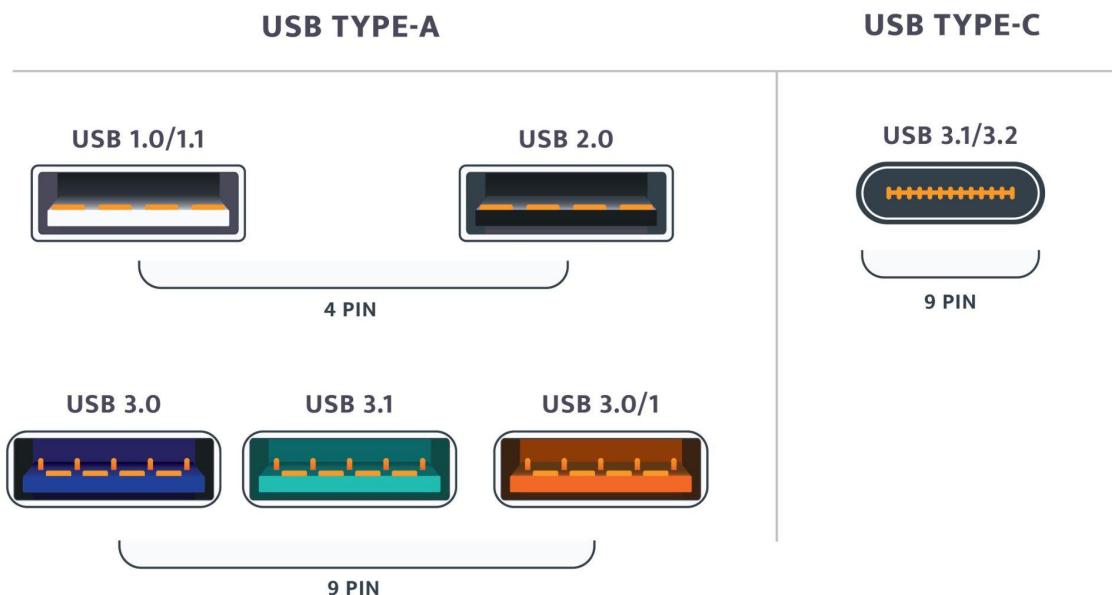
### Overall Experience

The Tilt Five glasses tether to a compute device such as a **PC (Windows 10)** or an **Android OS device (Android 10 or later)**. Typically, the player will launch an experience from the device that runs the game and once the experience is switched on for Tilt Five, the player turns their attention to the Tilt Five gameboard using the glasses and wand.

**GUIDELINE:** a Tilt Five app should ideally function in its entirety while viewed through the Tilt Five glasses and should not require use of additional or external display devices for gameplay, UI or menus.

## Hardware Interface Guidelines

The Tilt Five glasses can be plugged into a USB Type-C port or a USB Type-A port via an adapter. Note that Tilt Five's system requires ports that support USB 3.0 to be able to transmit video. USB 3.0 ports can often be identified by a distinct color or the official certified [USB SuperSpeed logo](#).



Note that a player may plug the glasses into their PC before / during / after the start up of your app.

**GUIDELINE:** at any time when the user plugs in the Tilt Five glasses, or when appropriate, have your app prompt the user if they wish to switch interactions to the Tilt Five glasses & wand.

From time to time, the user may lose connection to the wand and/or glasses – e.g. if they accidentally disconnect the cable, if the wand goes to sleep or power runs low.

**GUIDELINE:** the app should make the user aware if & when there is a connection loss to the Tilt Five glasses and/or wand.

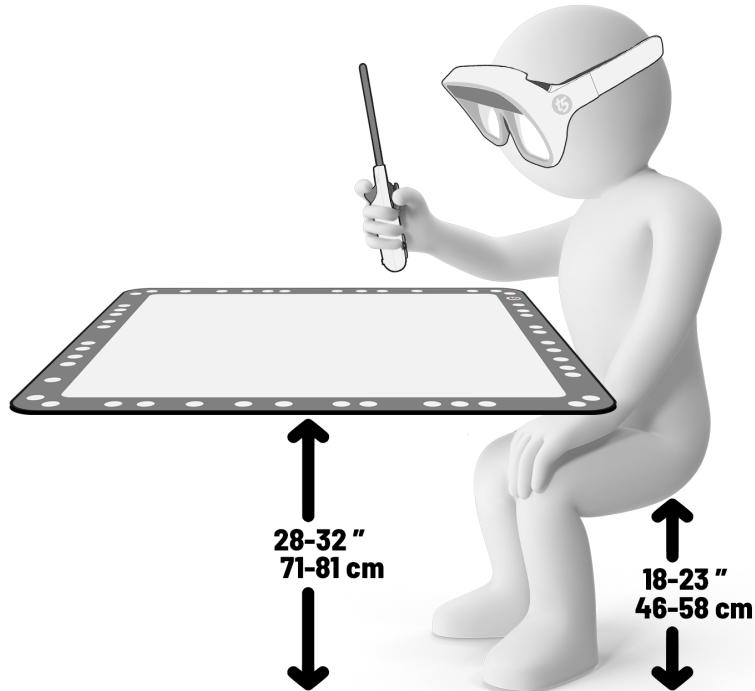
## Seated Experiences

Because the Tilt Five projection technology works from nearly any angle-of-view, experiences and UI presentation may look and perform differently depending whether the Tilt Five user is seated or standing and whether the gameboard is resting on a low or high table.

Ideally, games for Tilt Five should be seated experiences. This is mainly for comfort. Video game and board game sessions can last hours. And while our glasses are designed to be lightweight and comfortable, players may tire much quicker if they are standing.

**GUIDELINE:** we strongly recommend that developers design, build and test their experiences as seated experiences and assume that that will be the primary mode of play, with the gameboard resting flat on an average-height dining table.

- Dining room table heights are typically between 28 and 32 inches (71cm to 81cm). These tables are most compatible with 18 to 23-inch (46cm to 58cm) chairs, stools, and benches.

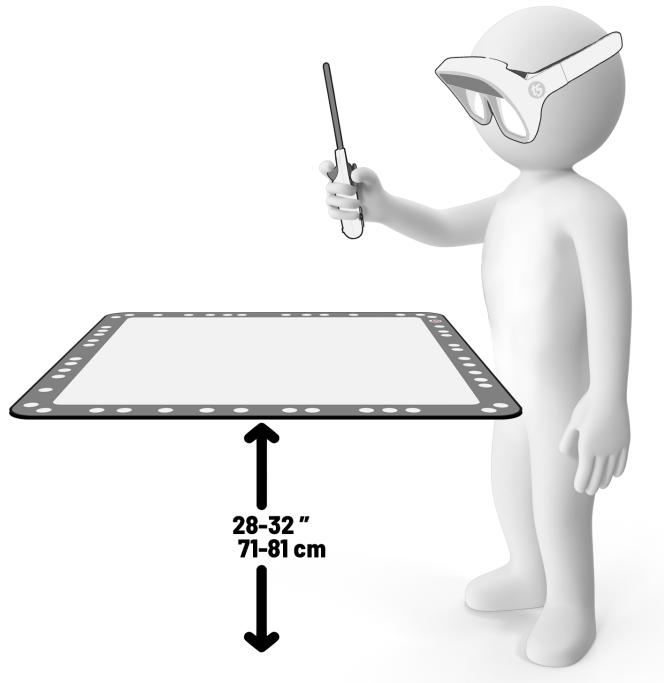


**GUIDELINE:** even if your experience looks fine with the gameboard at table height, you should verify that your UI, text, and necessary game objects are clearly visible and legible at other heights as well (e.g. on the floor, on a coffee table, while standing).

## Standing or Seated-Only Experiences

The Tilt Five system will work in both seated and standing positions.

However, if your experience is designed to function in only one of these two scenarios (i.e. is a seated-only experience or a standing-only experience), you should instruct your player to "sit down" or "stand up" for your experience through an in-game prompt right away.



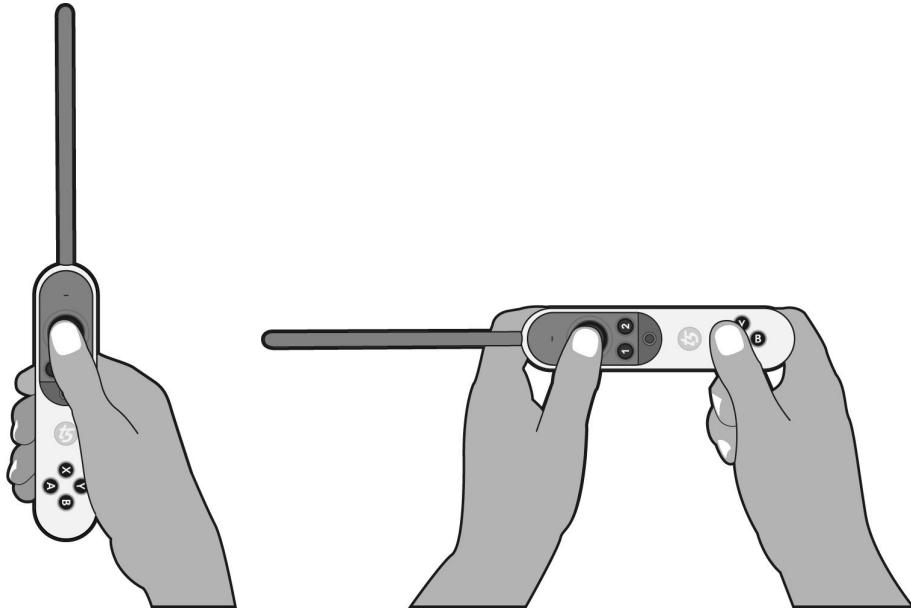
## Basic Wand Guidelines

**GUIDELINE:** it is strongly recommended that each experience for the Tilt Five system make use of the 6-DOF wand controller.

The Tilt Five wand can be used to manipulate items in the world using the standard buttons, joystick and trigger controls common to most video game controllers.

The wand can be held in two different ways:

1. A single-handed WAND GRIP - good for accessing the joystick, the wand trigger, as well as 6 degrees of freedom movement of the wand (think: magic wand!)
2. A two-handed SIDE GRIP - good for accessing the A/B/X/Y buttons and two-handed inputs, similar to traditional and classic controllers, like Nintendo



**GUIDELINE: communicate to the player early in your experience how they need to hold the Tilt Five Wand controller.**

Tilt Five has provided vector illustrations of the wand (including wand-grip and side-grip illustrations) for use in your UI. They can be downloaded from [docs.tiltfive.com](https://docs.tiltfive.com)

## World Scale

This applies to Unity (TBD if it applies to Unreal as well): Changing the world scale can affect various other aspects of a game in non-intuitive ways:

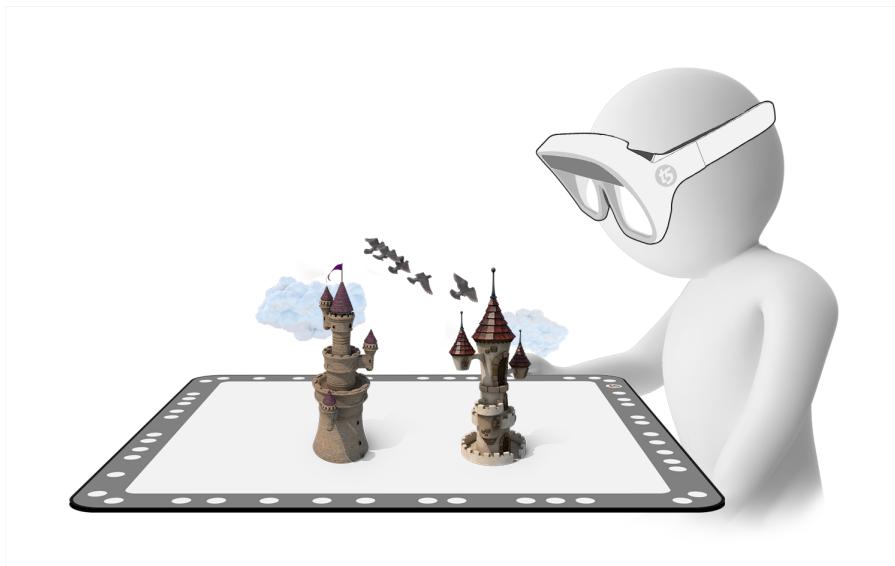
- Gravity needs to be scaled up if it is desired for it to behave like the physical world.
- Audio listener falloff may need to be changed to account for the increased distance to the camera.

**GUIDELINE: if it's necessary to change the engine's default world scale, keep in mind the ramifications of this and additional changes will need to be made as a result.**

## Gameplay Height

With Tilt Five, objects in-game can be presented to appear virtually on top, above, and even below the gameboard's physical surface.

**GUIDELINE: for the best impact, it is recommended that core gameplay be presented at the surface of the gameboard itself, with objects and scenery extending above, several inches from the gameboard surface.**

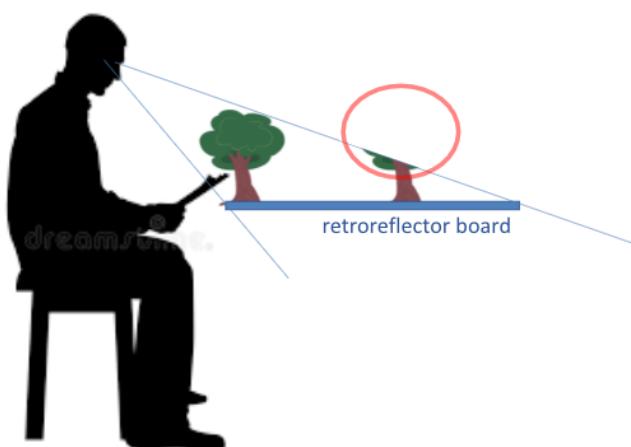


Depending on the angle of view, rendered 3D objects can appear to hover far above the surface of the gameboard, approaching within a short distance of the glasses. However, it is not recommended that you allow objects to render too close to the player's eyes, as they will have difficulty focusing on them.

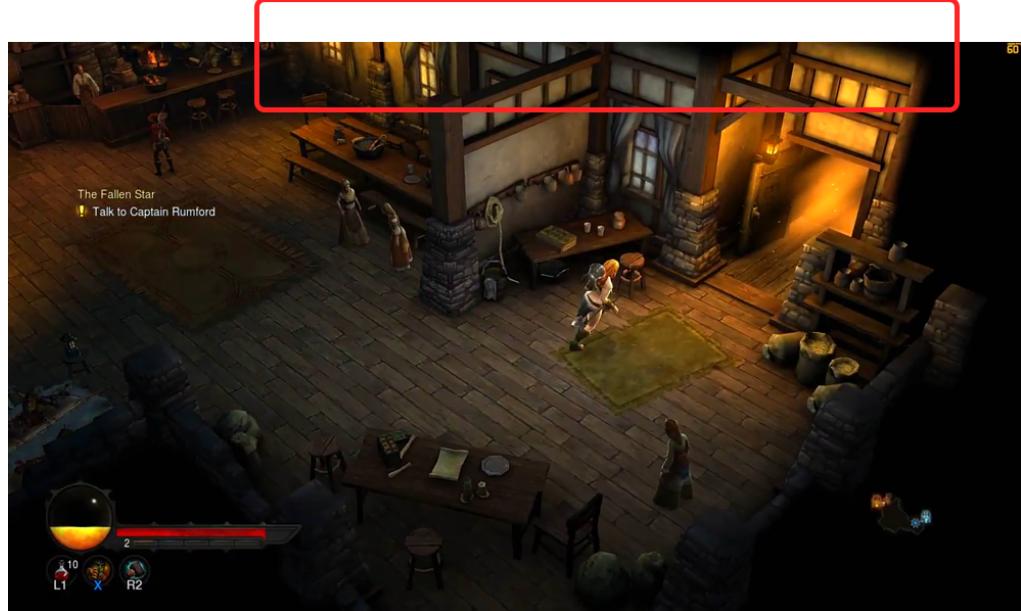
**GUIDELINE:** prevent rendering objects too close to the player's face by adjusting the *near clipping plane*, or using a shader that fades/dissolves objects at a certain distance.

As you design your application, keep in mind the different viewing angles users might use and how that will impact your level design, Wand inputs/UI, and game mechanics.

Note that tall objects on the furthest side of the gameboard may appear to be cut off (e.g., the top of the object will disappear at the very top edge of the gameboard).



This is not unsettling unless the object is essential to the game – e.g. it is the main character. In general, this issue is common to all video games on standard screens where scenery and objects are cut off at the top of the screen.

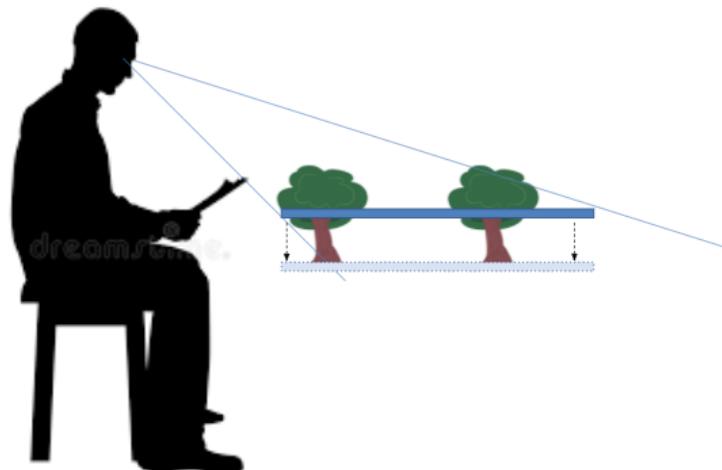


It is recommended that you keep essential objects, such as your main character, closer to the player if you are presenting your scenes directly at the gameboard surface. Keeping them around the middle of the gameboard is ideal.

**GUIDELINE:** keep essential objects and characters in the middle of the gameboard or closer to the player.

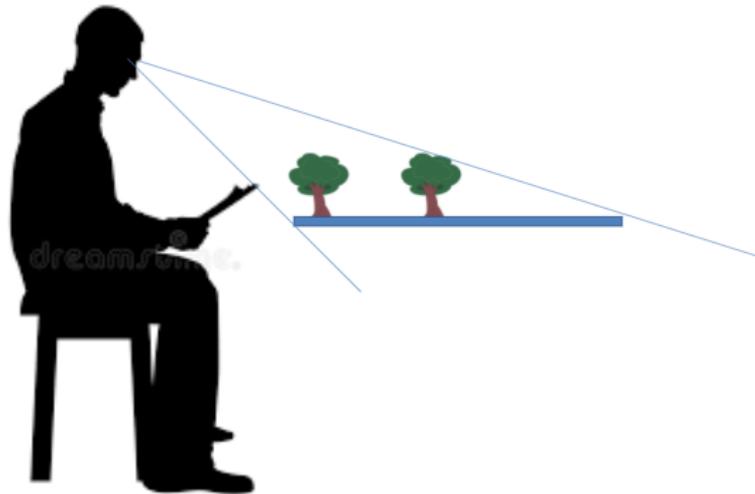
In your virtual scene you can also sink or lower the ‘floor’ into the gameboard, creating a *well effect*. Depending on how far you take this approach, sinking objects into the table may require the user to lean forward to look into the scene. The trade-off is that you can provide more simulated height by sinking objects & scenery into the table, but it may make things difficult to see when the player is viewing your scenes at low angles.

**GUIDELINE:** ‘sinking’ the virtual world into the table provides more room for simulated height.

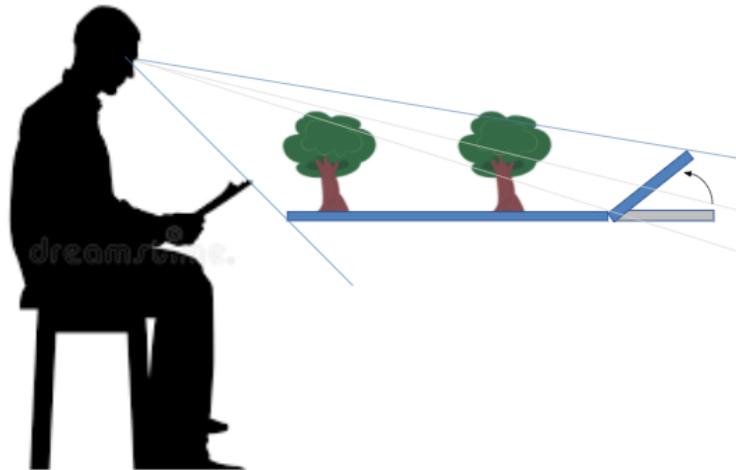


Another option is to reduce the scale of the scene so more objects appear on the gameboard at once.

**GUIDELINE:** set the scale of the gameboard widget (or scene) so that more of the environment can be viewed at once

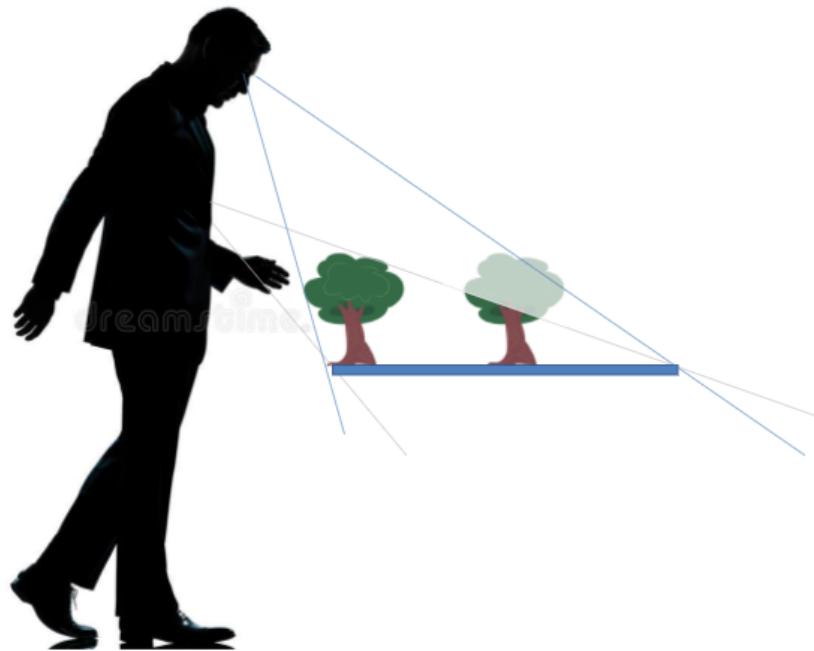


Note that the majority of Tilt Five owners will have the larger 'XE' gameboard which extends the retroreflector further by an additional 10-inches (25.5 cm) and an added 'kickstand' allows them to tilt the furthest end of the gameboard v.



**GUIDELINE:** it is recommended that you test your scenes for visibility using the XE gameboard laid flat as well as with the optional kickstand.

Lastly, it should be noted that leaning closer, leaning over, or standing by the gameboard will nearly always give viewers more spatial volume above the table.



## Graphics Guidelines

### Colors

The Tilt Five glasses use liquid crystal on silicon (LCoS) projectors using alternating red, green, and blue color fields. Using bright primary colors (i.e. dominantly red, green, or blue colors) can create a flickering effect, which should be avoided.

The projectors are quite bright and using very bright colors, especially white can create visual discomfort.

**GUIDELINE: Avoid creating scenes or objects dominated by bright whites or bright primary colors such as pure red, green, or blue.**

### Textures

As is common with many projection systems and displays, it is recommended that you do not use busy patterns with fine, repetitive details that exceed the resolution of the glasses. Doing so can create Moiré patterns – strange-looking wavy patterns that tend to break immersion.



**GUIDELINE:** Avoid using textures with fine, repetitive details that exceed the glasses sensor resolution, to avoid Moiré patterns.

## Contrast

Another common aspect of projection systems and displays is a varying loss of contrast, depending on the specific content involved and how much things may overlap with each other. Designing content that maintains a noticeable contrast between items, such as UI elements over top of a similarly colored background, can greatly enhance the player's ability to view and recognize the various content being shown.

**GUIDELINE:** be mindful of similarly colored objects which may be overlapping in the player's view, and increase the contrast between them for improved viewability.

## Antialiasing

Due to the nature of the projection system, aliasing tends to be readily apparent and even amplified as compared to similar content shown on a monitor. Aliasing can be more noticeable in some content than others, especially if your experience contains fine details, such as certain text fonts, or single-pixel width lines (outlines around an UI element, for example). These and many other situations can be greatly improved by antialiasing.

**GUIDELINE:** use antialiasing to greatly reduce the “jaggies” caused by aliasing in order to improve the readability and overall look of your content.

\*TIP: Try testing your game on a separate monitor with a reduced resolution to see the effect of detail reduction.

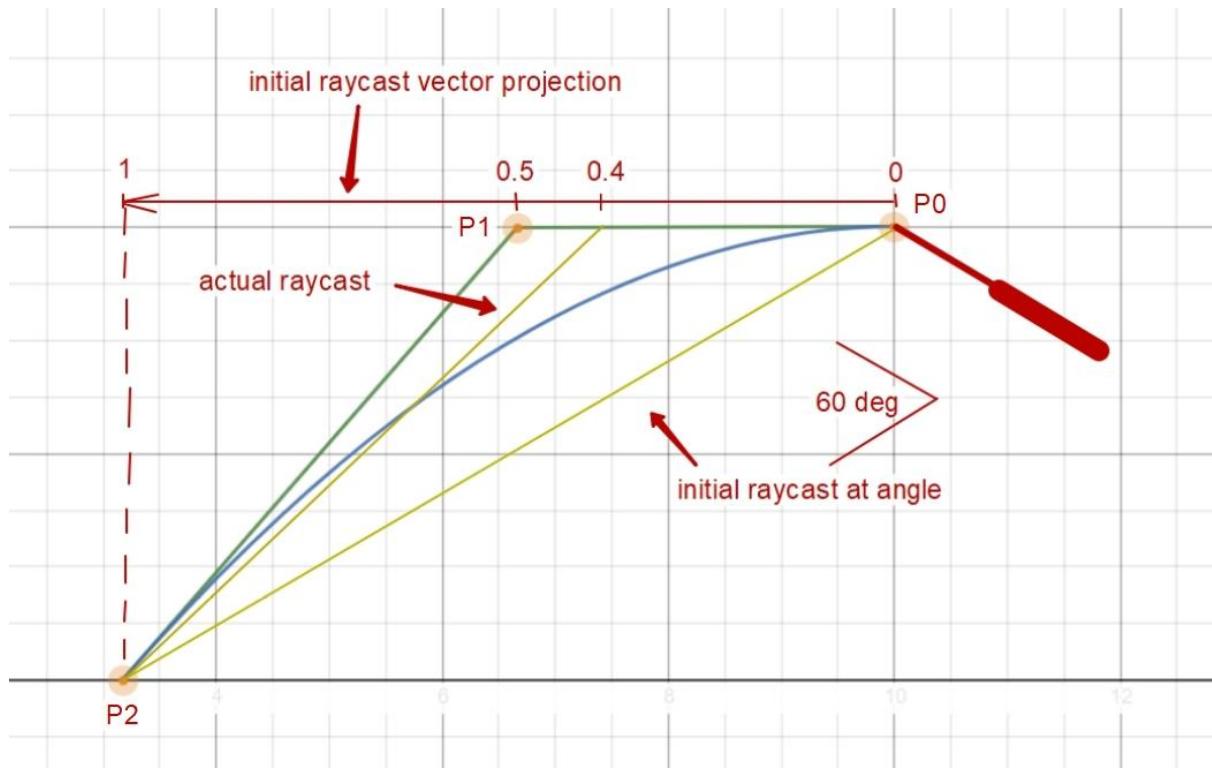
# Wand Controls & Input

Along with individual wand controls, additional inputs can be utilized by supporting combinations of controls (ex. holding trigger while moving thumbstick), or by implementing multiple states/modes which can be toggled on/off (ex. using thumbstick press to enable a radial menu with additional options).

You can also use the position of the wand to interact with objects in 3D space, turning the wand into a 3D stylus.

**GUIDELINE:** Avoid a straight line laser-pointer end-effector as that forces the player to either tip their wrist or pull their arm back behind their shoulder, both these gestures are physically uncomfortable.

It is recommended that you use the Bezier Curve Raycast method as an end effector.



To improve viewability of what a wand pointer may actually be pointed at, consider terminating the line with an easily distinguishable sphere, reticle, or some form of visual animation/effect.

**GUIDELINE:** Fully squeezing the wand trigger can cause small but detectable movement of a player's hand. While this movement may seem trivial, it may be enough to cause misaiming/misfiring and therefore frustration for your player, if precision aiming/firing is necessary for successful gameplay. Lowering the threshold your game uses to register a full trigger squeeze can help minimize or even eliminate this unintended behavior.

# UI Guidelines

There are a number of guidelines which can improve the viewability and readability of UI elements.

UI elements (artwork and text) that sit flush on the gameboard surface can become difficult to discern and read, due to the degree these elements are angled away from the player's viewing location. This issue is exacerbated the further away on the gameboard these elements are positioned from the player's viewing location, as they will be smaller and involve more effort by the player to focus their eyes on or even require leaning forward to get a closer look.

Another issue with UI elements, such as menus and panels, that sit flush on the gameboard is the risk of appearing very flat and 2D, even though they actually exist as 3D objects in the game's world space. Having these elements raised above (or even below, if it suits the particular game) the gameboard allows for their three-dimensionality to be in full effect, improving their viewing and offering a more impressive impact on the player overall.

Finally, a number of additional factors can affect UI text readability, such as font color, contrast, style, and size.

**GUIDELINE:** Place UI elements off the surface of the gameboard, and angle towards the assumed viewing location of the player (possibly starting with a 30-35 angle and adjusting as necessary from there), while trying to locate at positions closer to the player, and keeping in mind the various font specifications which can be used.

## Unity Reference App

This is a list of 20 scenes available in our Unity Reference App. Each scene is designed to demonstrate and provide instruction on a specific area of consideration when developing for Tilt Five. The Unity Reference App is included with our SDK installer, and can be found as "Tilt Five SDK Examples.zip" in <Your Drive>:\Program Files\TiltFive\SDK\UnityExamples. We strongly recommend running this Unity app and exploring its various scenes. Please note that additional scenes are currently being designed and will be added to this app in the future.

1. **Tracking an Object** : Have an object moving through a level with the ability to zoom in/out the camera that is following the object.
2. **Navigating a Scene** : Pitch, Yaw, Roll & Pan the camera looking down at a scene.
3. **Switching Scenes** : Cycle camera thru Players 1, Player 2, Spectator & Close-up view.
4. **Scaling Objects** : Scale the scene slowly up/down. Or jump up/down to predetermined camera scale positions.
5. **Scene Cursor** : Pointer follows the user's view. Use headset tracking to highlight & select objects in the scene.

6. **Glasses Detection** : Switches cameras movement based on glasses becoming available or unavailable.
7. **Input Visualization** : Visualize controller inputs by the user on a wand controlled graphic. Each button press is detected & shown on the controller graphic.
8. **Wand Tracking** : Wand movement is tracked over the board. Use the wand to interact with game objects.
9. **Board UI** : A UI that can be moved above or underneath a scene object.
10. **Glasses UI** : A UI that stays fixed facing towards the player's glasses.
11. **Rotating UI** : A flat UI that rotates in a circle towards the player's view.
12. **Hinge UI** : A flat UI that auto snaps to face forward the player based on board side.
13. **Second Screen UI** : Content is displayed on the source computer screen while AR content is shown in the Glasses.
14. **Pointer UI** : A UI with a pointer on the board that follows mouse, wand joystick or gamepad joystick movement.
15. **Glasses Detector UI** : PC program that auto detects glasses and activates them based on user input.
16. **Cylinder UI** : A cylinder-like UI that rotates depending on the user input.
17. **Wand View Pointer** : A pointer that displays from the wand end to select/deselect UI after a wand button press.
18. **Line Wand Pointer** : A line pointer that displays from the wand end to select/deselect UI after a wand button press.
19. **Touch Wand Pointer** : A pointer that displays from the wand end to select/deselect UI upon pointer touchind UI.
20. **Arc Wand Pointer** : An arc line pointer that displays from the wand end to select/deselect UI after a wand button press.