TEST PROJECT GDD

OVERVIEW

The intention of this project is to emulate the behavior of grabbing and throwing an object at a target in VR with an elegant, non-intrusive aim-assist, but using only mouse/keyboard controls for ease and accessibility of testing. As such, this ‘game’ takes place in first-person perspective, and only the hand is visible on the player’ controlled character’s body.

During development you may find more effective/efficient methods of achieving the same result. If that is the case, it is encouraged to take the approach you feel most confident about in delivering the highest quality that most closely matches the intended gameplay outcome. Once completed, we’ll be focusing on the quality and predictability of the catch/throw behavior, specifically in regards to the catch/throw assist functionality, and the general feel overall.

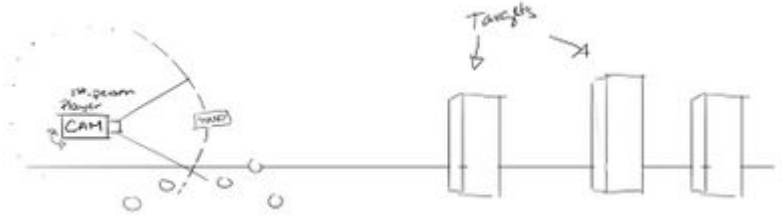
Of course mouse/keyboard input is very different than being able to track head and hands separately with gesture-based commands, and the limitations of this approach will be kept in mind during review.

GAME OBJECTS

PLAYER HAND: A basic shape (plane mesh) representing the hand. Your hand moves around with the mouse but remains arm’s length away from your body position (body is not visible—camera acts as player’s head position).

BALL: Basic sphere. Balls can be grabbed, thrown, and caught mid-air.

TARGET: Three human-sized rectangular meshes spaced far apart, and placed at different distances away from the player (close range, medium range, long range). These are the targets that we’ll use the throw the ball at.



CONTROLS

DRAG MOUSE: The player’s hand is positioned on a spherical track (the track is invisible to the player) with a radius that spans out to arm’s reach away in all directions from the camera position. Dragging the mouse moves the hand around the sphere in the direction of the dragging motion. The end result is a curvature applied to hand movement, much like extending your arm outward in front of you and moving it up or to the right of your body, rather than remaining in a fixed, always forward-facing position like a typical cursor/crosshair would be.

LEFT-CLICK: Hold to grip the ball. Release to launch ball.

WASD: UP/DOWN/LEFT/RIGHT adjusts facing direction (There is no walking and the hand moves separately from camera). Use these directional keys only to look around in 3D.

GRAB ASSIST

When your hand is within a short range of a ball object (an arm's length away) and the ball object is visible on-screen, pressing the GRIP BUTTON (left-click) will snap the hand onto the ball to grab it, and then snapping back to previous position with ball in hand. This occurs whether or not the ball is in motion, such as in the event of tossing a ball straight up into the air and catching it as it falls.

NOTE: 'snapping' will be done via quick movement from current position to ball's position, rather than an instant teleportation snap onto the ball object.

THROWING

Toss a ball by releasing LEFT-CLICK when a ball is held in-hand to lob it straight ahead (relative to hand position). Hold LEFT-CLICK to grip, release to launch. Gravity is applied to the ball being thrown so holding the ball straight out in front of you and launching it forward should send it approximately 20 feet before it hits the ground. Get more distance by aiming upward, much like shooting an arrow.

AIM ASSIST

To attain a mild form of assistive feedback we'll need to factor in a number of variables on the release of each and every throw. One important piece of information in determining the player's intended target is the current line of sight when throwing. Often times when a player is aiming at a target their line of sight (center point of the screen) is fixed on the intended target during the wind-up of the throw. This should be factored in when determining the pull the target has on the thrown object (more on that below), adjusting the pull radius a potential target has based on their proximity of the player's line of sight.

Any target that has a purpose to have an object thrown at it has a mild gravitational pull that affects objects thrown by the player. If an object is falling within range of this pull, the pull assists the object in moving inward toward the target it is close to. The pull itself should be minor enough that there is no noticeably abrupt alteration in the object's throw direction (i.e. the thrown object shouldn't suddenly change direction mid-air and snap to the nearby target, but rather curve subtly inward), and it should never be impacted by more than one target's pull radius at a time. For example, if an object is thrown directly between two potential targets, the two conflicting pull radii would theoretically slow down the thrown object mid-air as it is tugged between two opposing directions. To avoid this behavior we need to intelligently favor one target's pull over the other, taking the thrown object's proximity to each target's pull radius into account to determine which is favored.

IMPACT

After a ball is thrown it will either hit a target or miss. Either way, upon landing there will be a new ball that spawns on the ground in front of you and the previous ball will poof from existence. If you manage to hit the target, the target will quickly spin several full rotations before slowing down to a stop (spinning should not last more than 2 seconds), at which point it’s ready to take more hits.

DELIVERABLES

One build of the game, playable on MS Windows 10 OS