Rotatable Base Plate

Maximilian David
s1madavi 1349951
maximilian.david@uni–bayreuth.de

December 7, 2022

Another feature participants wished for during the pilot studies was the ability to rotate the base plate within the horizontal plane since this is a possible action when building with real-life LEGO.
It was decided to make the plate turn by letting the player grab and pull it in the rotation direction. This seemed to be the most natural way of creating the interaction in VR.
To achieve this new interaction possibility the "Interactable" and "Circular Drive" scripts were added to the base plate's game object. The "Interactable" script enables the user to interact with the base plate using their hands. The "Circular Drive" is a built-in script of SteamVR and makes it possible to rotate an interactable object around a given axis by grabbing and dragging it.

Now it was possible to rotate the plate. However, since the plate's origin is not located within the plate's center but at the plate's bottom right corner, rotating the plate around its origin does not yield the expected results. Contrary to this behavior, the user presumes that the plate rotates around its center, not one of its corners. Therefore, the plate needs to be rotated around its center, which excludes the option of turning the base plate using the circular drive's built-in "Rotate Game Object" feature. This feature only allows rotation around the game object's origin and does not provide options to change the rotation's center.

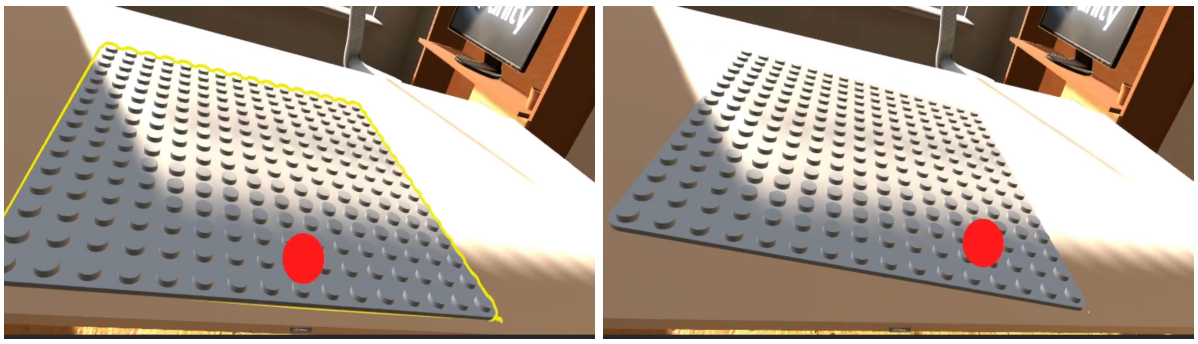

Figure 1: Plate rotated around its origin

To achieve the desired rotational behavior, the rotation angle from the circular drive was used within the plate's update function to rotate the plate around its center using the "RotateAround" function that Unity provides for every transform.
This method provided the desired rotational center and axis but also introduced glitches that caused the plate to rotate abruptly or kept it turning infinitely.

After thoroughly investigating the cause of this bug it was determined that the "RotateAround" function simultaneously rotated and translated the plate. This, in turn, conflicted with the local angular values used by the circular drive. Therefore, by using the "RotateAround" function and the circular drive on the same game object, two mismatching object angles were utilized to rotate the plate. To solve this problem a new game object was created and set as a parent to the base plate. This new parent object received the "Interactable" and "Circular Drive" scripts, while the drive's output angles were still read inside the plate's update function and used to rotate the plate. By shifting the circular drive to a parent object and only rotating this parent's child, the object containing the circular drive now remains sta-

tionary, which removes the interference between turning the plate and the drive's angle calculations.

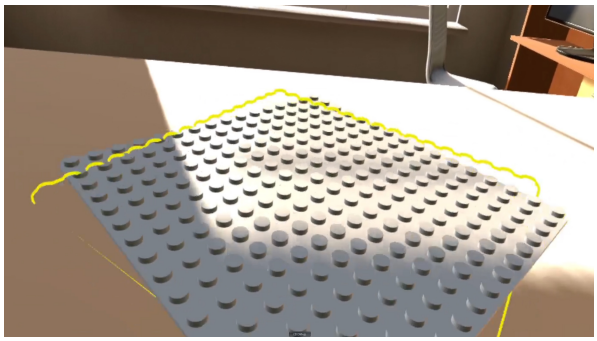

Figure 2: Correctly rotated plate with glitches

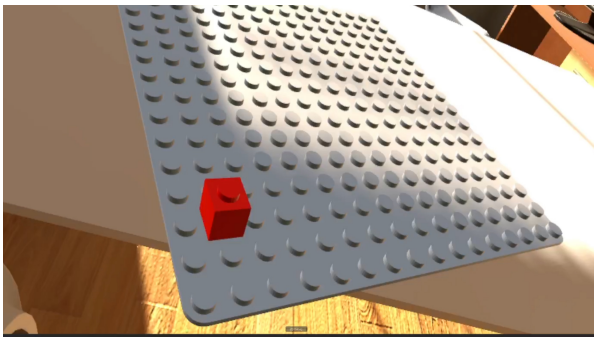

Figure 3: Rotated plate with attached brick

As a next step, all bricks connected to the plate also need to be rotated with the plate. This was achieved by simply iterating through a list of all bricks currently connected to the plate and rotating them around the plate's center by the specified angle using the "RotateAround" function.

To fully finish the implementation of the new feature, the placement estimation needs to be modified to work with a rotating plate. Whenever an intersection point is generated by casting a ray, that point is translated back to its position, where it would have been, if the plate had not been rotated. This is done by rotating the intersection point in the opposite direction around the plate's center. The resulting point can then be used to calculate the estimation as before. After estimating the placement the brick's transform is placed at the calculated position and then rotated around the base plate's center into its accurate position and angle for the current plate rotation.

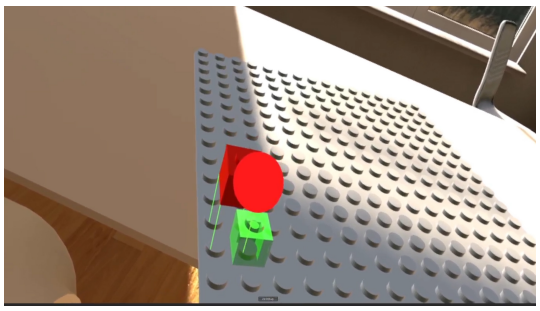

Figure 4: Estimation after rotation with 1x1 brick

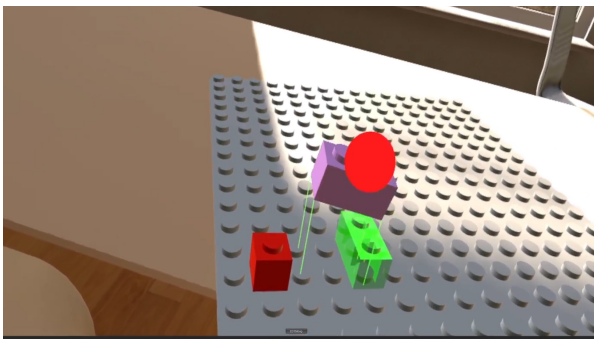

Figure 5: Erroneous estimation after rotation with 2x1 brick

This approach worked well for bricks occupying a single space. For longer and wider bricks, the rotation after placement and the preview's rotation did not match the held brick's current rotation. The underlying issue is the usage of the brick's world angles by the snapping and estimation system. These angles are independent of the plate's rotation and cause the described issue.
Additionally, the estimation's smoothness suffered greatly since turning the plate also caused the system to choose a non-ideal anchor to calculate the snapping point.
To eliminate these issues the brick's angle is now calculated relative to the plate. This guarantees not only the correct placement and angle for the brick but also the proper anchor choice, resulting in accurate and smooth placement for a rotating plate.

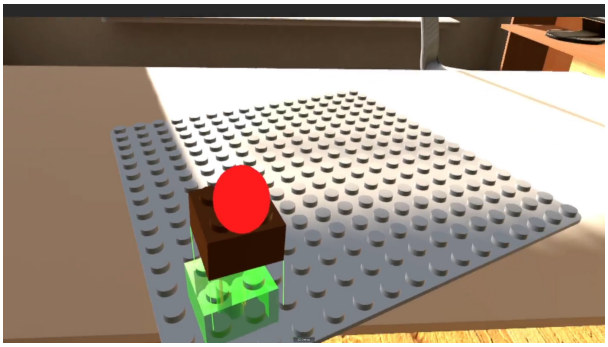

Figure 6: Correct estimation after rotation

The completion of this feature's implementation finalizes the development process for the VR LEGO clone experience which is trying to mimic real life as accurately as feasible.