# Accounting for Object Weight in Interaction Design for Virtual Reality

Jesper Rask Lykke, August Birk Olsen, Philip Berman, J. Andreas Bærentzen, Jeppe Revall Frisvad
Department of Applied Mathematics and Computer Science, Technical University of Denmark
Richard Petersens Plads, DTU - Building 324
2800 Kongens Lyngby, Denmark
jerf@dtu.dk

## ABSTRACT

Interaction design for virtual reality (VR) rarely takes the weight of an object – let alone its moment of inertia – into account. This clearly reduces user immersion and could lead to a break-in-presence. In this work, we propose methods for providing a higher fidelity in interactions with virtual objects. Specifically, we present different methods for picking up, handling, swinging, and throwing objects based on their weight, size, and affordances. We conduct user studies in order to gauge the differences in performance as well as sense of presence of the proposed techniques compared to conventional interaction techniques. While these methods all rely on the use of unmodified VR controllers, we also investigate the difference between using controllers to simulate a baseball bat and swinging a real baseball bat. Interestingly, we find that realism of the motions during interaction is not necessarily an important concern for all users. Our modified interaction techniques, however, have the ability to push user performance towards the slower motions that we observe when a real bat is used instead of a VR controller on its own.

## 1 INTRODUCTION

Regardless of the immersivity of a virtual reality experience, the body of the user of course never leaves the actual reality. This becomes a concern when we interact virtually with objects that appear to have very different physical properties from the controllers that we actually hold in our hands. In particular, virtual objects may have an expected mass and moment of inertia that is very different from the very light controllers. Nevertheless, we often employ a very direct *mapping* [Jer16] from the position and orientation of the controller to corresponding properties for virtual objects such as long sticks or heavy stones.

Studies have shown that a natural control mapping leads to more immersive interactions, which makes for a more enjoyable experience [SCP11]. The most physically realistic control mapping is known as *realistic tangible mapping* which makes use of a physical analogue to whichever object the user is interacting with [STS*11]. Shooting at digital targets with a gun-shaped controller, like in old arcade games, is one such example. While this control type may be the most realistic, it is not a viable method in applications where the user will interact with several different objects of varying shape, size, and functionality.

In lieu of a physical controller for every interactive object, a modified controller mapping might be used to simulate the physical properties and indicate the affordances of the objects. This is our point of focus.

To evaluate the influence of our techniques on the level of spatial presence, we use performance tests, and in some experiments also the Temple Presence Inventory (TPI) [LDW09]. The TPI method was developed and validated using psychological measurement procedures and has the purpose of quantifying the user's dimensions of presence. We thus use it for measuring spatial presence, but we also extend the questionnaire to provide a measure of how enjoyable the interactions felt to the test participants. Our performance tests measure how objects are handled. To assess the realism of a user's interaction with an object, we measure user performance with the controller attached to a baseball bat. We observe the qualitative change in user performance with and without a real bat, and we use this as a guideline for assessing the impact of our interaction techniques with respect to interaction realism.

To experiment with the handling and weight of objects, we set up a test scene containing a lightweight sword and a heavy hammer. To add to the atmosphere and interactivity, an axe, torches, and a shield were also included. Using the virtual weight of objects, we test unphysical downscaling of the controller velocity upon



This is the authors' version of the work. The definitive version is available at http://wscg.zcu.cz/DL/wscg_DL.htm
Lykke, J. R., Olsen, A. B., Berman, P., Bærentzen, J. A., Frisvad, J. R. Accounting for object weight in interaction design for virtual reality. *Journal of WSCG 27*(2), pp. 131-140. 2019.
https://doi.org/10.24132/JWSCG.2019.27.2.6

transfer of this to the in-flight velocity of a thrown object. To assess the effect of this and of the size and appearance of objects, we also perform a test where users throw different balls into a barrel. Our performance tests are hidden within small games to sustain user interest and immersion during each test. In a baseball bat test, we use a crude baseball stadium with a pitching machine as the test environment.

## 2 RELATED WORK

Simeone et al. [SVG15] investigated the change in people's spatial presence when presented with virtual objects of different levels of mismatch to that of the physical world. They found that having virtual objects that correspond almost one-to-one with physical objects (e.g. a flashlight as a lightsaber) would greatly increase the levels of spatial presence. However, they also found that a mismatch in weight or fragility would be particularly undesirable (unless we want to simulate super-human strength). However, setting up a one-to-one mapping between virtual and physical objects is in many cases problematic.

A technique like haptic retargeting [AHB*16] can be employed to let one object represent multiple virtual objects, but the weight of the object will be unchanged. As the moment of inertia of an object is directly related to its perceived weight and length, a device offering dynamic weight-shifting capabilities can significantly enhance perceived changes in shape and weight of virtual objects [ZK17]. Haptic devices applying gravity-like forces to the hand grabbing a virtual object can perhaps increase the level of realism even further [MFK*07, CCM*17]. These devices are however significantly beyond what standard VR controllers can do. To explore our options when such supplementary devices are not available, we consider software solutions. Our assumption is that the interaction design can influence the user experience in a way so that the objects (to some degree) seem to have a different weight and shape even if the controllers are unchanged. Software-based object interaction will obviously not feel as realistic as substitutional reality or haptic solutions, but it may take the user some of the way in cases where other solutions are not available.

As explored by Dominjon et al. [DLB*05], it is certainly possible to influence a user's perception of the mass of a grabbed object. This can be done by modifying the visual motion of the object controlled by the user. Motion amplification leads to a feeling of a lower mass and, conversely, motion damping leads to a feeling of a higher mass. This means that an unmodified VR controller can potentially provide an impression of interacting with objects of different virtual weight. Animating a self-avatar to reflect the weight of virtual objects that the user interacts with is one way to influence the perceived object weight [JAO*14]. Our methods directly modify the user-object interactions without use of a self-avatar.

Figure 1: Our main test scene.

The work most closely related to ours is that of Rietzler et al. [RGGR18]. They present a VR interaction technique that modifies the motion of the virtual object based on its weight. A grabbed object then no longer directly follows the user's manipulation of the controller. This is similar to the "chasing" method that we propose in the following. On the other hand, our test cases, comparison to substitutional reality, and our other methods are quite different from their work.

## 3 TEST SCENE

An overview of our main test scene is shown in Figure 1. Every object, apart from the table and the viking, can be picked up. The torch flames and the viking are animated. As the scene has a fairly realistic appearance, the test participants should expect a realistic interaction design. The implemented affordances and functionality will be explained in the following.

## 4 INTERACTIONS

We first describe our use of handle points. These are points on an object that people would normally grab when using the object as intended (such as the hilt of a sword).

### 4.1 One-Handed Interactions

When picking up an object with no handle points, the point of collision between the end of the controller wand and the object becomes a fixed joint. While picked up, the transformations of the controller are applied to the object, so that the same relative distance between controller and object is kept as illustrated in Figure 2. For heavier objects with no handle points, we introduce an alternative type of handling called scooping. This is further explained in Section 6.2.

Picking up objects with a handle point is a bit more involved. If the controller is within a certain distance from the object's handle point when the trigger is pressed, the object is translated so the handle point is at the same position as the controller, and its rotation is

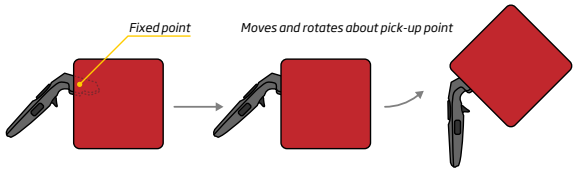


Figure 2: Object with no handle points.

set to that of the controller. When the object is in place, a fixed joint is created between object and controller.

If the controller is too far away from the handle point when picking up the object, a configurable joint is created between the controller and selected object. This type of joint gives the effect of lightly holding the object as if between the thumb and index finger, since it can be moved, but will always rotate its center of mass toward the ground.

When the trigger button is released, a release method checks if the selected object is held by the other controller. If this is the case, the releasing controller simply loses control of the object. If not, the joint between controller and object is removed and the velocity of the object is set to the velocity of the controller at the moment of release divided by the virtual mass of the object. This is unphysical, but it means that a user will not be able to throw heavy objects as far as lighter ones.

## 4.2 Two-Handed Interactions

If a user picks up a two-handed object outside the reach of a handle point, we create a configurable joint as for objects with one handle point. Alternatively, if a handle point is within reach, the controller grabs it. The object is then moved so that the handle point is at the position of the controller and a configurable joint is created. When the second controller grabs the object, the configurable joint is removed and the object is wielded with two hands.

A wielded, two-handed object is translated so that its first handle point is at the position of the controller that grabbed it first (Figure 3, left). The object is then rotated using a quaternion to have its forward vector pointing toward the second controller (Figure 3, middle). Since the local axis of our objects point upward by default, we apply a 90° rotation about the object's local $x$-axis. Finally, in order to be able to turn the object around its handle, the rotation of the last controller that grabbed the object is concatenated with the rotation of the object (Figure 3, right). The rotations are visualized in Figure 3.

Letting go of a two-handed object works in roughly the same way as with a one-handed object. When the trigger button is released on one of the controllers, the velocity of the controller is transferred to the object and the user loses control of it.

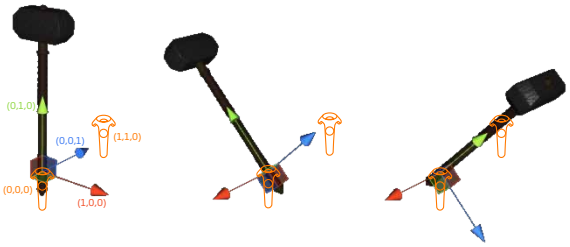


Figure 3: The rotations performed to make the two-handed object position itself between the controllers.

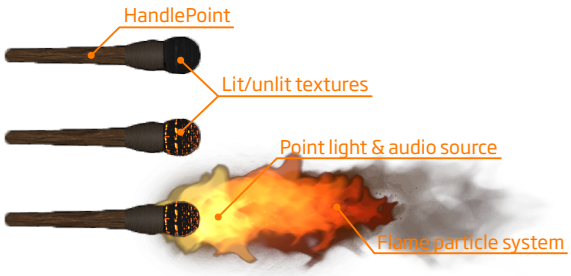


Figure 4: Torch components.

# 5 AFFORDANCES

## 5.1 Torch Light

Torches were implemented as both an interactive object and as a way of lighting the scene. The scene has a total of four torches: one on the floor and three hanging from holders on the walls. The torch on the floor is on fire from the beginning, giving the indication that all torches may be lit. Flames were added using noise-based particle systems. The torch components are illustrated in Figure 4.

The user is able to pick up the torch on the floor and take it with them as they walk around the room, using it to illuminate anything they wish to inspect. They can also use it to ignite the remaining torches that are hanging on the walls. Those torches are held in place by fixed joints, but if the user attempts to pick one of them up, the joint will be overridden by the one created by the user interaction.

## 5.2 Shield Orientation

The shield is meant to be carried on the side of the arm, which is why it has both a handle and an armstrap as shown in Figure 5. When it is picked up by the handle, the shield is rotated so the arm strap moves roughly to where the users arm would appear inside the scene.

The orientation of the shield is different depending on which hand is used to pick it up. A function checks which controller that picked up the object and rotates the shield accordingly. This is an example of an object that must be oriented differently for each hand.

## 5.3 Sword and Hammer

To compare one- and two-handed weapon interactions, a light sword and a heavy hammer were added to the

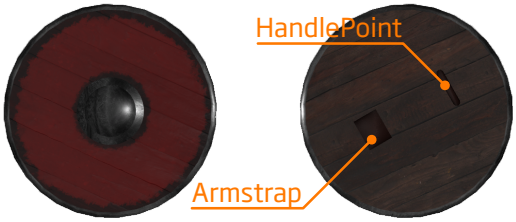


Figure 5: Front and back of the shield.

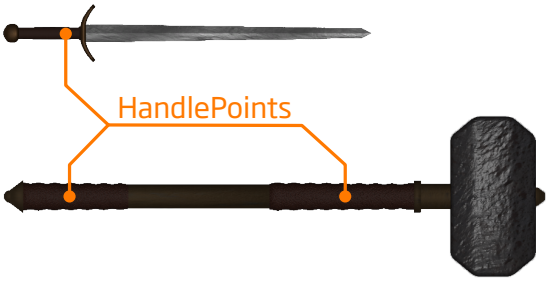


Figure 6: Handle points on sword and hammer.

scene. These two objects vary greatly both in size and material properties. The sword was modeled with a narrow hilt and thin steel blade, while the hammer was made quite a bit longer with a thick handle and large cast iron head. The sword has a single handle point, as it is only meant to be carried with one hand, while the hammer has two and can thus only be held correctly using both controllers. The hammer's handle contains two large leather grips, further indicating the need to carry it with both hands as shown in Figure 6.

### 5.4 Weapon Impact

With the purpose of having something to test the weapons on, we created an opponent who reacts to weapon impacts. The viking opponent contains several animated colliders. To give a visual indication of the force behind the objects being swung, several different reaction animations were created for the viking opponent: an idle stance, struck from the left, struck from the right, knocked back, and knocked down. Transitions back to the idle state are used when an animation has completed and none of the other possible transitions are active.

The viking enters the idle state at the beginning of the scene and awaits collision with other game objects. Upon collision, the force and direction of impact are used to calculate the next state. If the force is low enough nothing will happen, but if it exceeds the minimum amount, the viking will pull back his shoulder and turn his upper body. If he is struck with a greater amount of force a knock-back animation will trigger, where the viking takes a step back while bending his upper body backward. Finally, if he is struck very hard with a blunt object like the hammer, the viking opponent will fall backwards onto the ground.

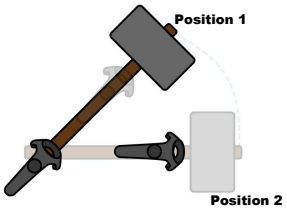


Figure 7: Hammer chasing the controller during rotation. Rotation is delayed when user motion is too quick, so that the hammer needs a brief moment of slower user motion to catch up with the controller closest to the hammer head.

## 6 WEIGHT

Based on experiments regarding object handling and weight, our main objective is to investigate whether software-based interaction methods can make a user interact with virtual objects more realistically. Given the test scene, its grabbing mechanics and its affordances, we developed three methods to better indicate the weight of an object. We refer to two of these methods as *chasing* and *scooping*. As a third method, we added haptic feedback from the controller during specific interactions.

### 6.1 Chasing

Chasing was chosen as the name for our method of making the hammer, or potentially other heavy objects, "chase" the controller during the interaction as a way of indicating its heaviness. The hypothesis is that the hammer might seem heavier, as it acts more slowly, and spurious movement is limited.

We use linear interpolation of quaternions (deliberately not spherical linear interpolation) to make the hammer chase the controller in the desired manner. The hammer only chases the controller near the hammer head, as this is where the rotation takes place. The other controller determines translation, as the centre of mass is not close to this point, and the user should still feel in control of the hammer. We find that this gives the desired indication of heaviness. Figure 7 illustrates an example of the chasing functionality where the rightmost controller has moved from position 1 to 2, but the hammer is (for the moment) still waiting at position 1. A couple of frames later, the hammer will have moved to position 2. The interpolation is greatly exaggerated, as the hammer will still be much closer to the controller at most times. The leftmost controller in Figure 7 is unchanged, indicating that the translation is still one-to-one.

From frame to frame, we use the difference between the new and old velocity of the hammer as an acceleration measurement. If the acceleration factor is much too high, the object will be released (dropped), making the user lose control of the hammer completely. If the acceleration factor is slightly too high, the interpolation

Figure 8: Handling of balls after the implementation of Scooping.

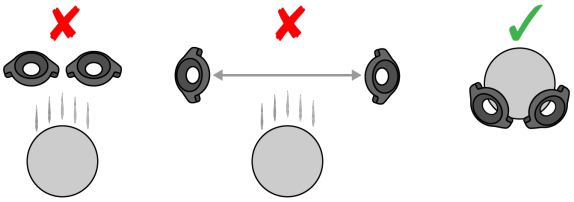

Figure 9: From left to right: Hands too close to the top, hands too far apart, Scooping is done correctly.

is set to be very slow, which forces the user to move the controller back to the correct position on the hammer before doing another swing, or wait for the interpolation to finish. However, the increase of the interpolation factor will accelerate with 5% for each frame, to ensure that the user needs not wait too long. If the hammer is swung with a realistic acceleration factor, it will move normally.

## 6.2 Scooping

Scooping is our method for handling heavy objects that we want to lift using two hands. When scooping, the user makes a gesture that takes the form of a bowl or a shovel. This can be used to "scoop" certain objects. Figure 8 illustrates the scooping gesture.

We implemented scooping for a set of test balls. The idea behind scooping is that many people would not be able to hold a heavy steel ball with a clawed fist, therefore they should not be able to do so in the virtual environment. However, many people would still be able to lift the same steel ball using two hands. If the hands are both directly under the ball applying force in the parallel opposite direction of gravity, the user would have the strongest grip. Sliding both hands towards the top of the ball would make the applied force smaller. Hence, the grip would become weaker when closer to the top, and eventually we would drop the ball. If the user was to move his hands too far apart, the ball would also be dropped. This is illustrated in Figure 9.

As opposed to our other types of interaction, scooping is done by using a grip button on the side of the HTC Vive controller instead of pressing the trigger. This felt more natural. When the controllers collide with a ball while gripping, we check if the object has any handle points. If not, scooping commences. The ball then centers itself between the controllers. We can then compute the positions and rotations of the controllers.

Regardless of the controller rotations, if the distance of the controllers to the ball becomes larger than the ball radius plus a threshold, the object is released and the ball is then dropped. This threshold has been chosen so that it corresponds approximately to a 5 cm real life distance to the virtual ball from each controller.

The virtual weight of the test balls are in the interval $[0.7, 1.9]$. We therefore set up the following formula to estimate the weight $w$ that the user can lift when making the scooping gesture:

$$w = 2 - \frac{\cos(\theta_{\text{left}})\cos(\phi_{\text{left}}) + \cos(\theta_{\text{right}})\cos(\phi_{\text{right}})}{2}.$$

Here $\theta$ and $\phi$ are angles of controller rotation with respect to the two horizontal world space axes. The subscript indicates the left or the right controller. If the user does not rotate the controllers at all (zero angles, Figure 9, left), which corresponds to having the palms turned downwards, the calculated weight that can be lifted is $w = 1$, making it the weakest grip. Conversely, turning the hands 180° to have the palms pointing upwards ($\theta_{\text{left}} = \theta_{\text{right}} = \pi$, Figures 8 and 9, right) results in $w = 3$, making it the strongest grip. If the calculated weight is less than the virtual weight of the object, the object is dropped. For example, if neither of the controllers are rotated ($w = 1$), a light ball of fabric with the weigth of 1 may still be scooped, but a wooden ball with a weight of 1.2 would be dropped. By doing the implementation with this approach, a heavy steel ball will be dropped much faster than a wooden ball, as it should. The selected virtual weights of different test balls are in Table 1.

## 6.3 Haptics

Haptic feedback can for example be used as an indication of wind resistance and strain. We therefore use haptics when interacting with most objects, as well as for the chasing and the scooping functionalities. A force parameter controls the haptic feedback in a controller, so we set a force based on the object velocity whenever the user is swinging a weapon. This creates a good indication of wind resistance as the weapon slices or crashes through the air. When the hammer is chasing the controller, we apply the maximum haptic force, giving the indication of swinging the hammer in a wrong manner and feeling the strain of doing so.

We use the haptic feedback slightly differently when scooping. Here the haptics are used as a warning of when the ball is close to being dropped. This also corresponds to the strain one might feel, when struggling with carrying a heavy object, because it is held in a wrong way. The haptic force will be in the range from zero to maximum depending on the difference between the lifted weight $w$ and the virtual weight of the object. Haptic feedback (vibration) starts at an angle of approximately 35° before dropping the ball.

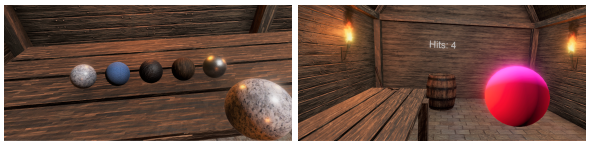


Figure 10: Test balls for scooping test.

| Material | Granite | Fabric | Cast Iron | Wood | Steel |
|---|---|---|---|---|---|
| Mass | 1.4 | 1.0 | 1.6 | 1.2 | 1.6 |

Table 1: Materials and associated virtual masses of the test balls. A purple ball is of undefined material and has a mass in $[0.7, 1.9]$.

## 7 TESTS AND RESULTS

Two test sessions were conducted: one without chasing and scooping and one with chasing and scooping. The first test had nine participants (three females and six males), the second test had eleven participants (three females and eight males). Seven test subjects participated in both tests and thus had some experience with the virtual environment when testing with chasing and scooping. None of the test subjects had any previous experience with virtual reality, but they all had an amount of experience with computer gaming. The test subjects were 19 to 26 years old and got as little help from us as possible during the tests. We got permission from everyone to film them, as well as their permission to use their test results in written work.

### 7.1 Different Virtual Masses

In many VR systems, a wooden table or a massive stone block can be thrown equally far. This can be fun, but an object not accounting for weight in its affordances is not very realistic. We set off one part of our test session for investigating whether realistic textures and different object sizes give an impression of the weight of an object. Figure 10 shows an overview of the test.

In the test, an operator would generate a ball by pressing a button. The system randomly selects one of six different ball types, and the test subject attempts to throw the generated ball into a barrel. In a typical test, the subject threw around 50 balls. The materials and masses of the balls in Figure 10 are listed in Table 1. The virtual mass is measured in arbitrary units. We refer to the ball with no texture as the "purple ball". The purple ball was assigned a random mass between 0.7 and 1.9, and its radius was scaled accordingly.

The test subject is presented with a ball, picks it up (with or without scooping functionality) and throws it toward the barrel from a marked position by the table. Every time the ball collides with the bottom of the barrel, a "Hits"-sign increases by one. The number of hits are not important, but the test subject got a little game out of the experience (trying to have as many hits as possible), which adds to the enjoyment. This should lead to more focused throws and thereby better test data.

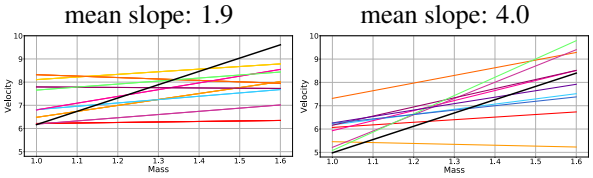


Figure 11: Plots of controller velocity by mass regression lines in ball throwing test without scooping (left) and with scooping (right). Each user has a separate color. Black user knows the masses.

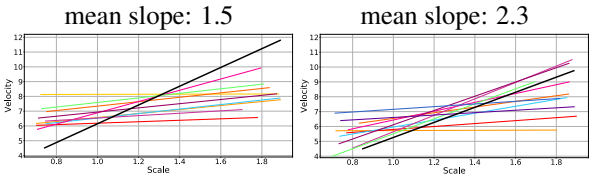


Figure 12: Plots of controller velocity by scale regression lines in ball throwing test without scooping (left) and with scooping (right). Each user has a separate color. Black user knows the masses.

The data logged in this test was the ball type, the controller velocity when the ball is released, and the mass of the ball. After filtering out probable outliers, we apply a linear regression to the data points. Outliers are typically the balls that were dropped by accident.

Going from lighter balls to heavier balls, the test should ideally show a positive regression slope. This seems counter-intuitive because it is unphysical, but larger controller velocity upon release becomes the same ball velocity in the virtual reality (and the same ball velocity is desired if the user wants to hit the barrel and score a point). The extra force applied by the user when moving the controller faster compensates for the fact that the controller weighs the same for all objects.

#### *7.1.1 Velocity Graphs*

Plots of controller velocity by mass and by scale are in Figures 11 and 12, respectively, where the "ideal" regressions shown in black are the results of testing ourselves, already knowing the mass of each ball.

In Figure 11 (left), we see that use of textures and division of the ball velocity upon release by the mass when released (as explained previously) gives an indication of the weight of the object. Subjects have a general tendency to throw the heavier balls with a higher controller velocity, albeit not a very strong one. A few of our test participants on average threw all of the balls with almost the same amount of force, regardless of their apparent weight. The regression lines of those test subjects still have some of the lowest slopes when our scooping method is employed (Figure 11, right). However, we generally see a stronger tendency to throw the balls with a force corresponding to their masses when scooping is employed. Lighter balls were thrown with a lower controller velocity than before, while heavier

Figure 13: Example of test-fight using the hammer.

balls were still thrown at a high velocity. Some of the test subjects even exceeded the "ideal" results achieved by ourselves, leading to the conclusion that the scooping method accompanied by material textures is a very effective method of communicating weight to the user.

The balls of varying size with no material texture were less effective. The difference in throwing force from the smallest to the largest ball was generally small. While a few test subjects came close to the ideal regression in Figure 12, most lines have a much smaller slope. Once again the scooping method made most of our test subjects throw the balls in greater accordance with their virtual masses.

Inspecting the average controller velocity for each material, we observe in tests without scooping that granite balls were thrown harder than steel balls, and that iron balls were thrown harder than steel and granite balls. This has had an impact on the regressions as the steel and iron balls had the same mass implemented, which is higher than that of the granite ball. Furthermore, the fabric balls were thrown at a rather high controller velocity, even though they were the lightest. Test subjects may have anticipated a wind resistance, but this was not implemented in the application.

When using our scooping method, the mean slopes increased significantly. The combination of scooping and material textures moved the average throw much closer to the ideal (slope 6.3). Altogether, both with and without scooping, the material textures seem to have communicated the weight of the balls a lot better than the size differences. The purple colour used for the scaled balls may have been misinterpreted by some as a light material (e.g. a rubber balloon).

## 7.2 Swinging with One or Two Hands

In another part of our test session, we compared the swing of the sword and the hammer. In the first test session, we let the user swing the hammer in the same manner as the sword, only using two hands instead of one. We wanted to test whether the hammer would be swung more realistically due to its heavier appearance. In the second test session, our chasing method was employed to further indicate heaviness, while the sword

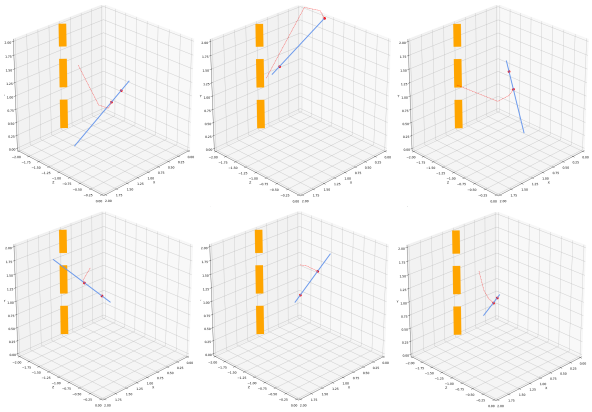

Figure 14: Hammer swings of two test subjects (one tester in each row).

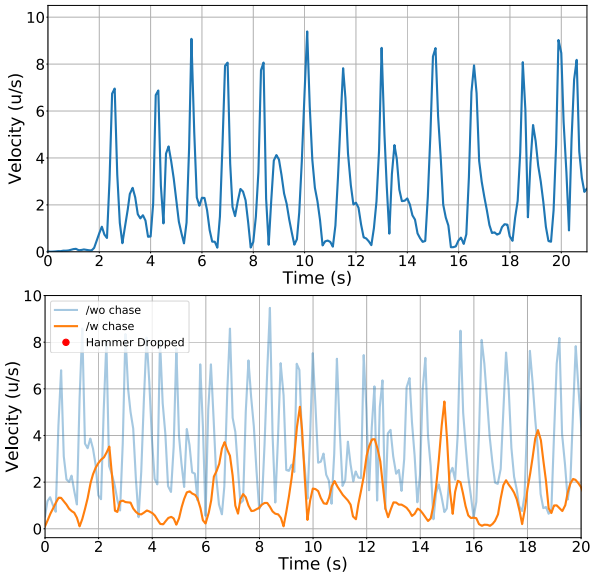


Figure 15: Examples of swing frequency plots for the sword (top row) and the hammer (bottom row).

would still follow the controllers one-to-one. Figure 13 shows an in-action example of a subject fighting the viking with the hammer.

Each test subject would fight the viking for about 40 seconds with both the sword and the hammer. We recorded each session and logged time, weapon velocity, and position and rotation of the controllers. These values were logged once every 10 milliseconds.

From the logged data we created two types of plots. The first type is an interactive 3D graph, that enables us to investigate the weapon swings from different angles and view the position of the two controllers throughout the swing, see Figure 14. The second type of plot shows the velocity of the weapon over time, see Figure 15. We use these plots to assess whether a test subject is using the hammer as one might in reality.

We investigate the hammer orientation during a swing to see whether the test subjects were immersed enough to hold the hammer in the same way as they would in

real life. We also investigate the sword and hammer swing frequency to see whether a test subject would swing the virtual weapon realistically or uncontrolled and vigorously.

#### *7.2.1 Hammer Orientation*

Logging the position of the controllers while the hammer is swung enables us to analyze both the path of each swing and where on the hammer handle the test subject would place the hands during a swing. The graphs produced from the data show the controller positions as red dots and the hammer as a blue line between them. The path of the swing during the past 300 ms is shown as a red dashed line and the opponent that the test subjects were told to hit is shown as an orange dashed line.

Figure 14 (top row) shows a couple of hammer swings from a test subject. From left to right, he swings the hammer downward into the opponent, lifts it up above his head, and swings it down again. As seen in the plot, this test subject would typically hold his hand near the head of the hammer when preparing for a swing and then move his hands closer together during the swing. These hand positions correspond well to the way one would swing a sledgehammer in real life, as you would want your hand closest to the center of mass when lifting it above your head, but slide it toward the bottom of the handle when swinging, allowing the sledgehammer to pivot about your hands.

In our implementation, since the hand at the bottom of the hammer controls its position and the other hand controls its orientation, the controllers do not need to be an equal distance apart throughout the swing. This allows for swings such as the one explained above, but also for ones that would not be possible in the real world.

Some of our testers would hold one controller statically in place and use the other to rotate the hammer about that point as shown in Figure 14 (bottom row). While this would seem unrealistic to do with a real sledgehammer, it was less noticeable inside the virtual environment. Perhaps enforcing an equal distance between controllers when holding the hammer or making the user hold the controllers at the same angle as the hammer would have yielded more realistic results among more of our test subjects, but adding too many restrictions could also greatly reduce their immersion.

#### *7.2.2 Sword and Hammer Swing Frequency*

In the test without chasing, the sword and the hammer were handled in the same way. The only difference was the heavier appearance of the hammer and the two handle points. We may contemplate based on Figure 15 whether the appearance and extra handle point of the hammer was enough for people to swing it differently from when swinging the sword.

In the hammer swing frequency plot (Figure 15, bottom row), the orange curve is from the test with chasing, whereas the faded blue is without chasing. Red dots indicate if the hammer has been dropped from swinging it too violently, which has not happened in this plot. Additional plots are available in a supplementary document. Comparing the sword swing frequency plot with the one for the hammer without chasing, there is no noticeable difference, meaning that the sword and hammer were handled roughly in the same way. This corresponds to feedback in our TPI tests, where people answered that the sword and hammer weigh approximately the same. If the test subjects had actually been tricked by the size and texture of the hammer, the swing of the hammer would generally have had a lower velocity and a greater wavelength compared to the sword. Interestingly, this is exactly the case after the implementation of chasing. The orange curve has a much lower velocity in general, and a greater wavelength compared to the blue curve. This is due to the fact that our chasing method forces the user to stabilize and swing the hammer in a realistic manner.

Although the data used in Figure 15 is from one test subject only, most other subjects revealed the same trend in their versions of this plot (see supplement). However, a few test subjects performed roughly the same both with and without chasing. These subjects were possibly also the ones not fully immersed in the experience, making them move the hammer very slowly and carefully around in both test sessions.

To confirm that the orange curve in Figure 15 is indeed a more realistic two-handed interaction with a long, heavy object, we conducted a test in a different virtual environment. This is described in the following section.

### 7.3 Baseball Bat Test

In this test, we use substitutional reality to investigate the differences between using a standard VR controller and a real life object. We use a baseball hitting simulation to measure the speed of a virtual bat's barrel when trying to hit a baseball. The real bat is tracked via a controller attached to it (see Figure 16), but can be freely interacted with as a normal bat.

In the simulation, the speed of the virtual bat's barrel in global space is constantly recorded so that the speed of the swing attempts can be measured. As seen in Figure 17, real bat swings cannot reach as high a maximum speed as using a controller. The real bat also requires more time to accelerate and decelerate between rest and swings. This is likely because a lightweight controller in the center of one's grasp requires much less effort to manipulate due to its weight and center of mass. Perhaps this absence of strain is the reason why some users, after conducting both tests, would surprisingly prefer the unmodified controller to the real bat.


Figure 16: An Oculus Touch controller attached to the base of an aluminum baseball bat. We use this aggregate in a test conducted in a crude stadium scene with a pitching machine.

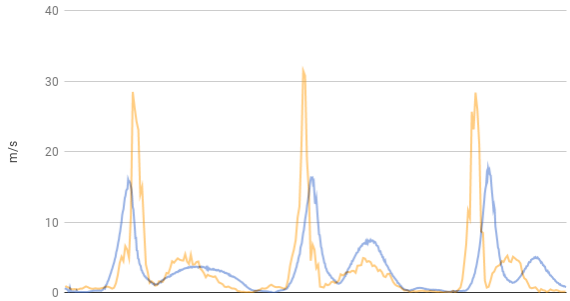


Figure 17: Virtual bat swing speeds using a controller (orange) and a real baseball bat (blue). The high spikes represent attempts to hit a virtual baseball.

We can now compare Figure 17 with the test results in Figure 15. Remarkably, our chasing technique seems to have qualitatively the same effect as if the user switches from using an unmodified controller to using a heavy, real object. The change to lower velocities and more stabilized motion between the hits is strikingly similar in the two figures. This is encouraging as we wanted to simulate realistic tangible mapping.

## 7.4 Temple Presence Inventory (TPI)

The TPI mentioned introductorily is a different kind of test. It is a questionnaire, where the test subject answers questions on a scale from 1 to 7. In our TPI, the questions are separated into five categories: ball interaction, weapon interaction, spatial presence, engagement, and perceptual realism. We provide a full overview of our TPI questions and results in a supplementary document.

The ball interaction category contains questions such as "how real did the wood ball feel regarding weight, size and overall texture?" And "how real did it feel to throw the balls?" The purpose of this approach is to assess what the test subject feels with respect to throwing the test balls. The weapon interaction category will ask if the hammer looks heavier than the sword, and whether or not it feels heavier to swing. The performance tests discussed previously were designed to assess the realism of user performance. In these ball and weapon interaction categories of the TPI, we ask the user what they felt about the interactions.

The last three categories: spatial presence, engagement, and perceptual realism together determine, each in their own way, how the user experienced the virtual environment. As mentioned previously, we believe the user will only expect the affordances of the objects to be realistic if the environment itself looks and feels realistic. The test subjects were therefore first free to walk around the virtual environment. They got to know the scene by lighting torches and trying the shield and an axe. We wanted to avoid low scores in these categories. Low scores here *and* poor performance tests would indicate a problem in the way the environment is portrayed rather than in the interaction design.

## 7.5 TPI questionnaire

The first section of the TPI was on the *ball interaction test*, asking how real the balls felt both individually and all together. In the test with scooping, subjects found that wood and iron balls felt more realistic. However, everything else got a worse result or nearly the same result as in the session without scooping. For steel and granite balls, which got lower scores with scooping, one explanation could be that the test subjects became frustrated with the controller rotations required to scoop those balls, while the wood ball weighed much less and was thus more forgiving. Hence, some balls seemingly got a worse score simply because they were more difficult and not as fun to throw. In any case, the test scores (with or without scooping) are quite high.

The next section of the TPI is the *weapon interaction test*. From this questionnaire, we observe that with our chasing method the hammer felt approximately 23% worse to swing. However, when asked about its heaviness compared to the sword, the score was about 36% larger than in the test session without chasing. Other answers indicate that the hammer does not feel quite as heavy as it looks, but it got closer in the test with chasing, where subjects found it much heavier than the sword. The conclusion resembles the one from the ball interaction test: the hammer was more realistic to swing and felt heavier, making it more difficult to use, and perhaps it was then not as much fun to swing it.

The sections on *spatial presence*, *perceptual realism*, and *engagement* can be grouped together into an overall experimental measurement of immersion. The scores are very high with the average spatial presence of the

participants being 6. We believe initial interaction with torches and items like the shield were important in ensuring this high number. For perceptual realism, the average score was around 4.5, which is also great as these specific questions are slightly better suited for substitutional reality. The motion sickness score is also very low, which means that the users did not experience much lag, unsteady locomotion, or light flickering.

TPI answers indicate that test subjects went from a score of 5.3 to a score of 5.9 with respect to feeling mentally immersed in the experience when scooping and chasing were included in the tests. In addition, the participants were a bit more relaxed during these tests and less engaged regarding all their senses. However, the difference between the scores is not substantial. We may conjecture that the experience was perhaps slightly more relaxing as it was no longer possible to interact with everything as vigorously as before the use of chasing and scooping. The average score was about 5.5 in both test sessions, which is quite high.

From the TPI scores, we conclude that the experience of the virtual environment was altogether both realistic and immersive.

# 8 CONCLUSION

We sought to improve upon the realism of interactions in virtual reality by using software solutions to encourage natural handling of objects. A virtual environment for the study was created in the form of a room populated with various objects of differing size, shape, and functionality. Basic mechanics for grabbing and moving these objects were implemented followed by interaction possibilities corresponding to their fundamental affordances. More advanced interaction techniques depending on the mass and shape of objects were then implemented in the form of a chasing method and a scooping method.

Performance tests regarding throwing and swinging mechanics were conducted both with and without use of the chasing and scooping methods. We used a presence questionnaire to assess the user feeling with respect to interaction realism and immersion.

The test subjects found themselves highly immersed in the virtual environment, but interacting quite unrealistically with the test objects when our interaction techniques were not employed. Use of our methods led to a general improvement in the realism of object handling and the feeling of weight. Nevertheless, this led to only a small improvement in immersion. Test subjects also implied that the interactions were overall less enjoyable with our techniques. Their levels of enjoyment could perhaps have been maintained, had the implementations been a bit more forgiving.

In conclusion, it was possible to achieve a feeling of weight through the proposed methods of object handling, thereby simulating realistic tangible mapping.

# 9 REFERENCES


[AHB*16] Azmandian M., Hancock M., Benko H., Ofek E., Wilson A. D. Haptic retargeting: Dynamic repurposing of passive haptics for enhanced virtual reality experiences. In *Proceedings of Conference on Human Factors in Computing Systems (CHI)* (2016), pp. 1968–1979.

[CCM*17] Choi I., Culbertson H., Miller M. R., Olwal A., Follmer S. Grabity: A wearable haptic interface for simulating weight and grasping in virtual reality. In *Proceedings of ACM Symposium on User Interface Software and Technology (UIST)* (2017), pp. 119–130.

[DLB*05] Dominjon L., Lécuyer A., Burkhardt J.-M., Richard P., Richir S. Influence of control/display ratio on the perception of mass of manipulated objects in virtual environments. In *Proceedings of IEEE Virtual Reality (VR)* (2005), pp. 19–25.

[JAO*14] Jáuregui D. A. G., Argelaguet F., Olivier A.-H., Marchal M., Multon F., Lécuyer A. Toward "pseudo-haptic avatars": Modifying the visual animation of self-avatar can simulate the perception of weight lifting. *IEEE transactions on visualization and computer graphics 20*, 4 (2014), 654–661.

[Jer16] Jerald J. *The VR Book: Human-Centered Design for Virtual Reality*. ACM and Morgan & Claypool, New York, NY, USA, 2016.

[LDW09] Lombard M., Ditton T. B., Weinstein L. Measuring presence: the temple presence inventory. In *Proceedings of the 12th Annual International Workshop on Presence* (2009), pp. 1–15.

[MFK*07] Minamizawa K., Fukamachi S., Kajimoto H., Kawakami N., Tachi S. Gravity grabber: Wearable haptic display to present virtual mass sensation. In *Proceedings of ACM SIGGRAPH 2007 Emerging Technologies* (2007), p. Article 8.

[RGGR18] Rietzler M., Geiselhart F., Gugenheimer J., Rukzio E. Breaking the tracking: Enabling weight perception using perceivable tracking offsets. In *Proceedings of Conference on Human Factors in Computing Systems (CHI)* (2018), ACM, p. Paper 128.

[SCP11] Shafer D. M., Carbonara C. P., Popova L. Spatial presence and perceived reality as predictors of motion-based video game enjoyment. *Presence: Teleoperators and Virtual Environments 20*, 6 (2011), 591–619.

[STS*11] Skalski P., Tamborini R., Shelton A., Buncher M., Lindmark P. Mapping the road to fun: Natural video game controllers, presence, and game enjoyment. *New Media & Society 13*, 2 (2011), 224–242.

[SVG15] Simeone A. L., Velloso E., Gellersen H. Substitutional reality: Using the physical environment to design virtual reality experiences. In *Proceedings of Human Factors in Computing Systems (CHI)* (2015), ACM, pp. 3307–3316.

[ZK17] Zenner A., Krüger A. Shifty: A weight-shifting dynamic passive haptic proxy to enhance object perception in virtual reality. *IEEE Transactions on Visualization and Computer Graphics 23*, 4 (2017), 1285–1294.