A Bridge Between Bodies: Puppetry-Based Interfaces for Virtual Reality

Pierce McBride
Georgia Institute of Technology
85 Fifth St. NW
Atlanta, GA 30308 USA
pmcbride7@gatech.edu

Michael Nitsche, Ian Bogost, Jay Bolter
Georgia Institute of Technology
85 Fifth St. NW
Atlanta, GA 30308 USA
michael.nitsche@gatech.edu, ibogost@gatech.edu, jay.bolter@lmc.gatech.edu

ABSTRACT

This project applies the principles of puppetry to virtual reality game design and interface design. It consists of a virtual reality prototype that showcases three possible third person interface designs based off three different forms of puppetry. Each has been testing in a pilot IRB study and iterated on as a result of the provided feedback. In this paper we outline the background research and question that led into the project, as well as the project design process and outcome.

INTRODUCTION

Although Virtual Reality (VR) as a technology has been experimented with in various academic labs since the early 1980s, the recent resurgence in VR has so far been the most public-facing push for the medium. As of the time of writing, two platforms have grown out of early prototypes and achieved what is arguably the widest distribution of VR hardware yet seen: the Oculus Rift (Oculus) and the HTC Vive (Vive).

Notably, both of these platforms now utilize a similar stack of hardware interfaces, which include two motion-tracked handheld controllers and a head-mounted display also tracked within physical 3D space. These hardware interfaces lead to what Marco Gillies describes as “Movement Interaction” (Gillies 2016), or interaction design based up physical motion of the body. This type of interaction and hardware design has lead the majority of VR video games to rely on the first person perspective, or a camera perspective for the player that directly aligns with that of the avatar’s perspective. However, this leads us to a question. What could a third person VR video game play like?

PROBLEM

The question at hand, what could a third person VR video game play like, was born from an observed lack of video games that align with what we understand as a third person game, or a video game where the camera’s perspective, and that of the player’s, view the avatar from a distance. In a sense, VR’s reliance on first person perspectives is indicative of the hardware it rests upon. Gillie’s overview of movement interactions, which are head tracking, walking, object manipulation and body language, all lean into the notion that smart consideration of the limits of current technology can “reproduce our cognitive and emotional engagement with the world and our movements” (Gillies 2016). While this does not necessitate a first person perspective, it does suggest that the perspective is *implicit* in VR.

The same, however, cannot be said of video games broadly. Although input methods vary between platforms, such as controllers for video game console, mouse and keyboard for PC or embedded hardware input for handheld systems, these systems all share the use of the a screen as their display. In this other format, the perspectives used are far more varied. Many of the most widely known video games, such as *Super Mario Bros* (Nintendo 1985), *The Legend of Zelda: Ocarina of Time* (Nintendo 1998) and *Grand Theft Auto V* (Rockstar North 2013) all almost exclusively use a third person perspective. Furthermore, each of these video games is played from one avatar’s perspective at a time. There are other widely known video games where it would be difficult to argue that the player embodies a single avatar, but are nonetheless widely popular. *The Sims* (Maxis 2000) and *Civilization* (MicroProse 1991) fit this description.

However, if one turns their attention to VR platforms, they would find a limited set of third person video games. *Lucky's Tale* (Playful 2016) uses a third person perspective in a game design similar to *Super Mario Bros*. *Moss* (Polyarc 2018) and *Chronos* (Gunfire Games 2016) do so as well but bear more similarities to *The Legend of Zelda*. This is not an exhaustive list, but it is indicative of the broader state of VR game design because all three games utilize a gamepad as the primary hardware interface, and only *Moss* uses any movement interactions beyond that of head tracking.

The limited set of VR video games that use a third person perspective suggests we need to look elsewhere for design references. Our project addresses this dearth of examples by providing our own. We designed multiple third person VR interfaces within the context of our own video game prototype, and we hope that our interfaces act as a design reference for future academic and commercial projects.

However, because there are no examples, we had to look beyond video game design for our own reference. During this process, one reference became acute; Puppetry. We see the practice and design of puppetry as a way to approach our driving question, what could a third person VR video game play like?

RELATED RESEARCH

Although the choice is not an arbitrary one, we will first step through the design research we undertook before considering puppetry, which includes both previous experimental research and VR media artifacts. This research process clarified what we saw as the important characteristics of VR and, in time, how puppetry aligned with those characteristics.

We see both methodological connections between the design process of puppets and game characters and a reference for what we can describe as physical interfaces between the puppet and puppeteer. While the goals of these two disciplines are different, we see these connections as compelling, and in our research phase we further examined these fields through previous research and existing VR media.

A Character in Your Hand

Pierce McBride and Michael Nitsche, two of the researchers on this project, have previously worked on a different project with similar design goals called *A Character in Your Hand* (Nitsche and McBride 2018). In that project, they sought to design and develop a web-based prototype and a VR prototype for digital archival remediation of real puppets. This project was completed with funded from the National Endowment for the Humanities (NEH) and we collaborated with the Center for Puppetry Arts (CPA) in Atlanta, GA, so we could make use of they're puppet archive. Botht the web and VR portion of this project used recreated 3D models of 10 puppets from the archive and create simulacra of each that could be used. In the VR portion through the simulacra could be performed as well as used.

Fig 1. Example model transfer from physical to digital

A Character in Your Hand didn't seek to necessarily transpose puppetry design into VR or video games. What it sought to do was remediate existing puppet artifacts into digital technologies, and focused on interaction and interface design that supported the puppets we remediated. The process of doing so was one of compromise though. The motion-tracked controllers of the Oculus and Vive do provide high fidelity tracking of the overall hand position, they don't offer as much fine-grain manipulation of one's physical fingers. This limitation was what we primarily sought to design around, in the context of the way puppeteers would expect to control puppets.

In this project, puppets can only be held or manipulated from their handles or apperati that a puppeteer would use in real life, and can be picked up and manipulated by moving one's hand to a given puppet, pushing down on the triggers of the Oculus or Vive controllers (henceforth described as motion controllers, one for each hand) and dropped by pushing and holding down the same trigger on whichever hand is holding a puppet handle. Each hand can hold multiple puppets or objects by pressing down the trigger multiple times over multiple objects.

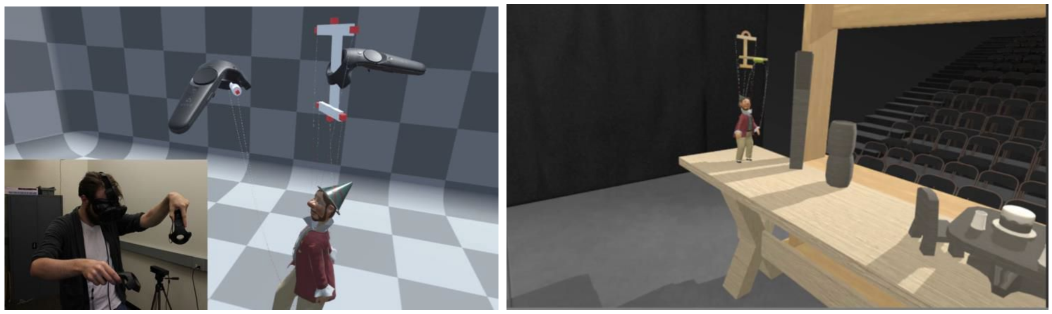


Fig 2-3. Example puppet implementation of marionette, showing manipulation (Left) and environment (Right)

This approach afforded a puppeteer the ability to properly use puppets designed with more than two handles, which is a common design practice. For example, some marionettes use more than one cross to manipulate the puppet. What parts of the body each cross controls differs per puppet, but one reason this is sometimes done is to split control of the legs and arms onto two separate puppeteer hands. This allows a puppeteer to walk a puppet legs by tilting a handle left and right without also moving the arms synchronously (see Fig 2).

By supporting this interaction we made it possible to use a wider array of puppets as intended by their designers, but the interaction was not without shortcomings. We did not, for example, support a way for the puppeteer to alter their grip on a handle without dropping it first. This could be done physically with a combination of loosening one's grip and using fingers to push or pull the handle within the hand's grasp. Similarly, when using a marionette puppeteers sometimes pull or pluck specific strings separately from rotating the entire handle. We did not support either of these interactions in ours tests, but they were commonly brought up by professional research participants.

The goal of *A Character in Your Hand* was to remediate real puppets into VR and web-based interfaces. Even though we didn't seek to completely recreate the source puppets, we did maintain their general design and function. *A Character in Your Hand* stands in contrast to the goals of this project, where we seek to design VR interfaces based off puppetry practice, not specific designs. In addition, this project is not funded from the NEH, but because of the continuity of researchers from *A Character in Your Hand* we can design around the learned limitations of existing hardware input and carry forward these lessons into this current project.

What is Movement Interaction in Virtual Reality for?

Our focus on VR interaction design in *A Character in Your Hand* led us to consider what existing theoretical frameworks exist in the field. Our work suggests that motion tracking is important but it's existing limitations need to be designed around. Marco Gillies in *What Is Movement Interaction in Virtual Reality For?* (Gillies 2016) makes the same case, that movement interactions in VR are at a minimum very important, if not essential, in VR. Gillies also provides a taxonomy of interaction design we can use to evaluate our own work and other VR media artifacts. Gillies outlines four categories of movement interactions in this paper: head tracking, walking, object manipulation and body language. With each in turn, Gillies connects these categories to that of Slater's Theory of Place Illusion which argues that VR is immersive because it "reproduces the same sensorimotor contingencies as the real world" (Gillies 2016).

First, head tracking, which is an interaction that result from the motion of the user's head, either leaning, rotating or pivoting. This interaction is nearly universal in VR systems. According to Gillies, head tracking aligns with Slater's theory of place (Slater 2009) in that turning one's head in VR turns the virtual camera to the same degree, for example.

Second, walking interactions, which are the result of the player moving their head through whatever sized environment is supported by the head tracking technology. Gillies cites additional research that reports on user studies where users felt a greater degree of immersion when physically walking through a environment than when they used a joystick or walked in place, suggesting that the act of walking itself is important for the experience (Usoh et al. 1999).

Third, Gillies cites object manipulation as an important movement interaction, which he defines as mapping hand movements and rotations directly onto virtual objects. Again, this one-to-one mapping aligns physical motion with virtual motion. Gillies even cites research that studied user preferences for various object manipulation mechanics and found that this kind of interaction design, one-to-one motion matching, is prefered by all participants (Bowman and Hodges 1997).

Forth, and last, Gillies discusses body language as a final method of movement interaction in VR that, in a sense, combines the three prior categories. Motions like nodding or moving to maintain a comfortable speaking distance are physical parts of rhetoric that, if one sought to design interactions between virtual and real people, or multiple real people in VR, one would want to support.

While we consider the fourth category, body language, interesting, it does not directly relate to our project because we target third person control. The other three however we use as a categorical definition of different motions through our documentation..

Homuncular Flexibility

While we found Gillie's framework useful, we also wanted to consider if or how these principles influenced embodiment, or way a user's sense of self extends to virtual bodies . Gillies himself touches on this subject in his paper as well, citing the theory of embodied cognition, or the theory that "much of our cognition occurs in our perceptuo-motor system" (Gillies 2016). Cognitive science research suggests that the way physically perform an action shapes how we think (O'Regan and Noë 2001). Given the way VR interaction design maps motion, this research suggests physical action impacts how we design virtual/physical interactions in VR as well.

Andrea Stevenson Won in *Homuncular Flexibility in Virtual Reality* (Won et al. 2014) expands upon this notion by studying the brain's ability to map and remap bodily motion onto new or existing limbs in VR. Won tests this notion through two experiments, one where participants pop virtual balloons with either a normally tracked avatar or an avatar who's arm and leg mappings are either reversed or extended, and another where participants physically select boxes of an appropriate color with or without the aid of a third arm protruding from the chest.

In the first experiment, participants with unusual bodies altered their motion to align with the avatar's capabilities within 10 minutes of the experiment starting. In the second experiment, participants with the third arm actually performed the given task *faster* than those without the third arm (Won et al. 2014).

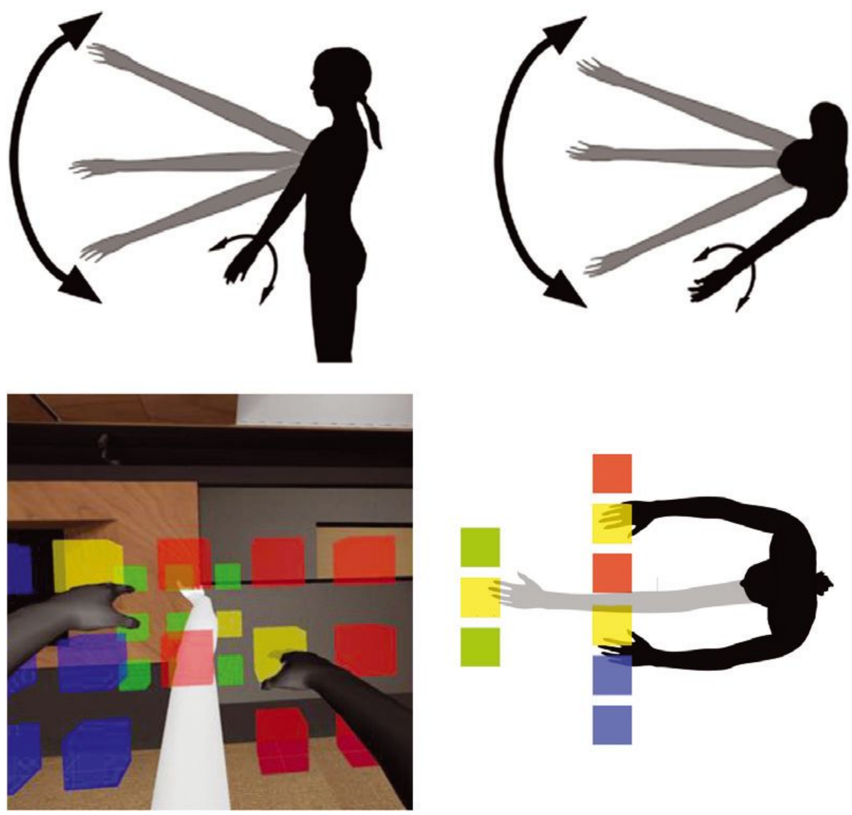


Fig 4. Representations of "third limb" control system and testing condition (Won 2015)

These results further suggest that the brain is quite capable of rapidly adapting to novel bodies. These experiments did not, however, seek to ask whether the participants felt that they embodied the novel elements of their avatar body. *Human Tails: Ownership and Control of Extended Humanoid Avatars* (Steptoe, Steed, and Slater 2013) does however investigate just that. One portion of the *Human Tails* experiment involved setting fire to a virtual tail attached to the tracked avatar body, and regardless of the simultaneity of the tail participants reported feeling anxious, which led to their attempts to put out the fire on the tail.

All of these studies investigate the players relationship with their virtual body, but don't seek to compare across input hardware. Our target for this project are the motion controllers used with VR hardware, rather than other hardware used such as controllers or keyboards. *I'm in the Game: Embodied Puppet Interface Improves Avatar Control* (Mazalek et al. 2011) specifically compares a custom-built puppetry-based interface to conventional Xbox and keyboard input. The comparison is made across two experiments, the first of which involves moving a virtual body to tap on floating teapots as quickly as possible, the second of which is done before and after the first and tasks the player with mentally rotating a 2D projection of a matrix. They find that the group that used the puppet interface showed the most improvement in the mental rotation task and had the highest performance in the teapot task.

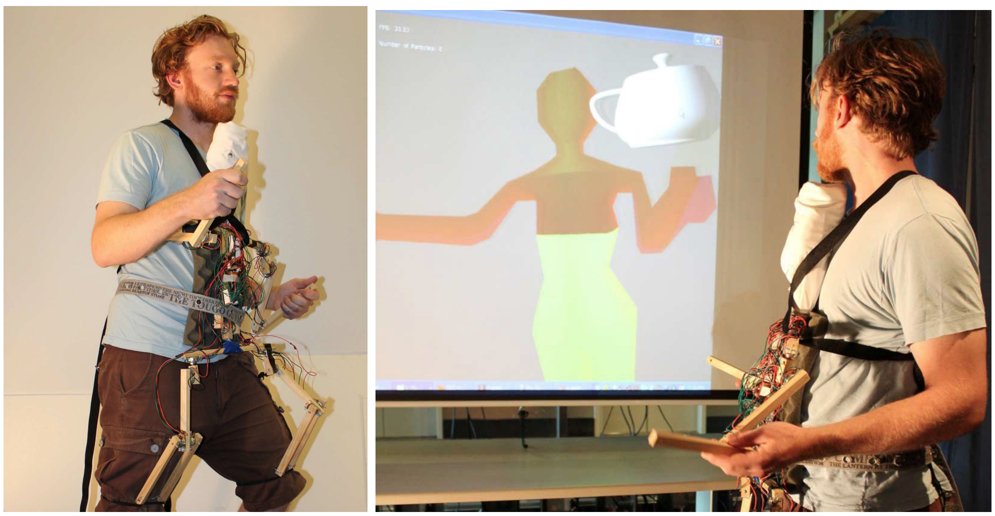


Fig 5-6. Two photographs of the full-body puppet interface in use (Mazalek et al. 2011)

Mazlak's findings, according to the author, suggest that "the value of embodied interfaces for the development of more personalized and less restricted interaction with virtual worlds (such as video games)" (Mazalek et al. 2011) . While VR interfaces are not strictly full-body, as the Mazlak interface is, VR hardware's ability to track hand, head and body position suggests that these results would apply to a similarly designed experiment in VR.

Together, these findings suggest that, even if designer's intent is to create a VR video game that provides players with a sense of embodiment, they need not limit themselves to human forms or bodies. This is due to the hardware interface itself, which faithfully utilizes body motion as input. While we find these results encouraging, our central question considers third person VR video games. If a player may feel embodiment with a fictitious limb, might a player feel the same about a character they can see?

Why Can I See My Avatar?

Daniel Black, author of *Why Can I See My Avatar? Embodied Visual Engagement in the Third-Person Video Game* (Black 2017), considers this very question the context of game design. However, Black's argument stems from an alternate interpretation of the question at hand. Rather than arguing that the third person perspective is uniquely suited to embodiment, he instead suggests that first or third person perspectives are equally capable of providing a sense of agency through player intent. Black argues "even in a first-person game, the viewpoints of the player and the game character are never truly unified, meaning that the difference between a first person and third person representation is only one of degree" (Black 2017). In other words, because technology is always an mediator between the experience of play and the physical act of playing a video game, some level of abstraction must exist in either case.

Black's first point, that first person perspectives are not synchronous relationships between physical and game bodies, is made primarily by considering the input devices used and the abstractions needed for even the most synchronous input mappings. Video games played via keyboards or controllers easily suggest this discontinuity. Mappings may differ from video game to video game, but because the hardware will continue to be buttons and analog input some level of abstraction must occur. Even motion tracked controllers for VR devices rely on abstractions to function, consolidating entire groupings of fingers to individual buttons or abstracting large spatial motions like movement due to tracking limitations. While this does not discredit these input methods, it highlights how in even the best of circumstances we must accept that any perspective's mappings rely on design models of abstraction

Second, Black considers synchronicity of intention, or the way a video game interface supports a mapping of player intention to avatar body. High synchronicity between intent and action is what Black compares to the sensation of the controller "melting away" (Black 2017) from conscious thought. This can also be compared to a film audience identifying with characters on screen. Despite the editing and the, sometimes rapid, changes in perspective, viewers still wince from near misses in car chases, just as players of video games may physically duck from the onslaught of danger above their avatar.

Fig 7. Tutorial message illustrating a holster interaction in Robo Recall (Epic Games 2017)

This argument has interesting implications when applied to VR design. We as designers already design around limitations in VR hardware through interaction design. We, for example, use teleportation mechanics or use object "holsters" (see figure 7) to interact with inventories and large environments in VR, such as in *Robo Recall* (Epic Games 2017). If these abstractions must exist, even in first person VR video games, then an abstraction of avatar motion from physical body motion is a logical extension of the same argument.

RELATED MEDIA

When we reviewed existing VR media we found relatively few third person VR video games, none of which heavily utilize movement interactions. We did review both third person and first person VR video games which we felt either well utilized movement interactions, had strong senses of embodiment or simply provided a third person VR perspective in any form.

Chronos

One such third person VR video game we examined was *Chronos* (Gunfire Games, 2016), an action role playing game that takes place in a post-apocalyptic version of our world circa mid 1960s and in a second world that closely matches high fantasy tropes.

Chronos is played with a traditional gamepad and the player views their avatar from a series of designated positions in the environment. Each position gives the player a filmic, fixed place to view the environment and the character from. These positions change automatically as the player directs the hero to move around, ensuring that their avatar can always be seen. This camera system most resembles that of *Resident Evil* (Capcom, 1996) which also made use of fixed camera positions but did so primarily for hardware performance reasons.

Fig 8-9. Two screenshots from Chronos, showcasing different camera positions in one room

Chronos uses a nearly identical camera system to avoid the pitfalls of motion sickness. Because one of the leading reasons for motion sickness is the speed of movement and acceleration in virtual environments (Oculus 2018), the design decision to affix the camera to specified places in each environment neatly avoids the need for the camera itself to move without the player's input. Transitions are instantaneous, and occur whenever the designers have decided that a given position no longer provides an adequate view of the player's avatar. This decision also has the added benefit of focusing the player's attention in a particular direction, which is notably difficult for VR game design (Oculus 2018). However, we can't argue that the camera system for *Chronos* benefits the rest of the design instead of simply paving over problem spaces. This is clearest when examining the game's combat.

Chronos's combat mechanics most closely resemble the combat mechanics of *Dark Souls* (FromSoftware, 2011), albeit simpler. Both video games feature difficult combat with melee attacks, avatar dodging, blocking and parrying as well as meaningful systems based around avatar death. Notably, both titles also include a lock-on mechanic where the player can designate a particular monster to "focus" on. In *Dark Souls*, this makes the camera and the player avatar automatically orientate themselves to face the position of that given monster. In *Chronos*, that same mechanic only orients the avatar.

Fig 10. Example of Chronos's lock-on mechanic

These lock-on mechanics commonly exist in three-dimensional video games to assist with the complexity of simultaneous avatar movement and camera orientation. While the need for a mechanic like that is clear in non-VR games, in VR presumably the act of camera orientation is done by the player's physical head. So, in the case of *Chronos*, We suggest that this lock-on mechanic exists in order to compensate for the camera's fixed position coupled with the ambiguity of analog stick rotation on a three-dimensional object. In non-VR video games, the camera usually remains behind the avatar's back as often as possible. This design decision has multiple benefits, but one of which is less ambiguity in three-dimensional movement. This ambiguity stems from disparity between the orientation of the camera to the avatar. In the cases where both are facing the same forward vector, there's no issue. Right on the control maps to rotating the avatar right. However, right on an analog stick may mean "rotate the avatar to face the camera's right, which is behind the avatar" or "rotate the avatar to face the avatar's right, which is camera-forward". Usually video games favor the camera, which is a good solution but the problem exists as much in VR as it does in non-VR. In fact, because the camera's position is fixed this ambiguity is **more** common in *Chronos* than it would be in a game like *Dark Souls*, because the camera can't track the avatar's back by definition.

However, it's important to remember that these issues exist because *Chronos* uses VR and a traditional controller, not movement interactions. Ambiguity between joystick abstractions and the avatar body are an understood problem in game design but if *Chronos* had not been a VR video game the issues would have been easier to deal with. Given this conclusion, we suggest that, although we don't consider *Chronos* a useful design reference, we do consider *Chronos* an inadvertent argument for movement interactions in VR because their absence problematizes other aspects of VR.

Lucky's Tale

Lucky's Tale (Playful 2016) showcases how movement interactions in VR can impact a video game's design without utilizing motion controllers. *Lucky's Tale* does this with just head tracking, and uses a traditional gamepad for avatar input.

Fig 11. The player avatar in Lucky's Tale, the titularly named Lucky

In *Lucky's Tale*, the player controls a cartoony anthropomorphic animal avatar named Lucky who explores largely linear levels. The video game draws close parallels with the platformer genre, *Super Mario Bros* (Nintendo 1985) being a prototypical example. The titular character Lucky can jump, bounce and spin his tail, each of which is used to attack or avoid monsters that get in his way as he explores each level. *Lucky's Tale* even goes as far as maintaining a core interaction from *Super Mario Bros,* whereby jumping onto a monster eliminates that monster from the game.

Lucky's Tale also draws from some of the camera conventions of platformers, most notably the general left-to-right motion of the game. However, unlike other platformers, the player still has control over the tilt or lean of the camera, which leads to a compromise where the camera pans left-to-right alongside Lucky but the player has control of local orientation. This is akin to riding a slow moving train alongside Lucky. The player can lean and look about from their seat but most of the motion happens outside of their control.

While the overall structure and design of *Lucky's Tale* leans heavily on platformer genre conventions, it also includes a few VR specific flourishes. The environments and characters are undeniably small when observed from the headset. This gives the video game the feeling of looking at a diorama, with small motion and intriguing detail hard to make out without observing it closely (see figure 12). Furthermore, the game features collectible objects not unlike coins in Mario, but adds to this is a modification where they are sometimes semi-translucent. They would be invisible if not for their idle motion, which produces small visual artifacts are possible to spot, and therefore the objects can be collected.

Fig 12. Example of the semi-translucent coins. Look above Lucky's head.

We claim that this is *Lucky's Tale*'s most compelling feature. By hiding in the periphery of the levels these objects are hard to spot. They are often either above or below the straight on field of view, or are hidden behind the geometry of the level. *Lucky's Tale* creates tangible rewards for close observation. This trait well aligns with platformer design but also suggests ways in which the viewpoint of the camera, and therefore the player, allows for game designs that take advantage of a third person perspective in VR and head tracking broadly.

The Climb

Lucky's Tale poses an interesting presentation of third person VR and head tracking but makes no case for any other categories of movement interactions. Because of the challenge of finding appropriate examples of both third person VR and movement interactions at once, instead we examine of a few first person VR video games which well-utilize just movement interactions. One such game, *The Climb* (Crytek 2016), serves as a compelling example because it adherence to the motions of a physically demanding sport.

The Climb is a level-based video game about free-climbing mountains in exotic environments. The video game tasks players with physically moving their hands from handhold to handhold to slowly climb up and around a given peak. *The Climb*'s levels take on shapes similar to mazes in this way. Handholds are laid out in predetermined locations to form paths that either loop back on themselves, lead upwards, or are dead ends.

This leads to a intermittent pace, where players identify a path, travel along it and rest periodically as they determine the next stretch of handholds they will climb across. Aside from navigating across these handholds, players must also manage stamina for each hand, which deteriorates over time if the player holds a particularly small handhold or uses only one hand. Although it's possible for the player to fall or lose their stamina, it's relatively easy to avoid. Instead, the challenge comes from the player literally reaching their arms around their room and pull their virtual body along.

Fig 13. Example of a handhold in The Climb

The moments where the level design dictates that the player must reach, or adjust their physical body to progress, are where the game most successfully matches it's theme. It can be a genuinely physical task to stretch one's arms across a virtual gap, and the game also tracks the head's position so if the player's head interacts with a wall it must be moved to continue. This adherence to the physicality of the game, and the real sport it draws inspiration from, is it's strongest design success. Without the physical affordances of it's virtual-tangible environment, the game would lack much of it's engaging play.

However, *The Climb* necessarily avoids certain actions that a free-climber would perform with their legs due to hardware limitations. While walking as a movement interaction works perfectly well, it's based off the position of the head. Therefore, actions like angling the feet to fit on a small outcropping or sidling along a rockface don't have an analogue in *The Climb*. While this is not strictly an argument against the design of *The Climb*, it does highlight how, even from a first person perspective with no teleporting, holstering or other abstraction, concessions must be made to the verisimilitude of the design.

Nonetheless, *The Climb* succeeds in imparting the idea of how physically demanding the sport is, and would likely not be as compelling if played without movement interactions. In *The Climb*'s case, it is playable with a controller but not without a VR headset. There are, however, a few video games that were redesigned for VR after an initial release, which make for an even stronger argument for the value of movement interactions because of the arguable improvements to each of the next two video game's original intentions.

Surgeon Simulator ER

In 2013, a full three years before the release of the Vive or the Oculus Rift, a video game called *Surgeon Simulator 2013* (Bossa Studios 2013) became a commercial hit largely for it's complex, difficult controls that often resulted in comedic failure. The game tasked players with performing some sort of surgical operation, from heart transplants to brain transplants, with physically simulated tools.

Much of the complexity of the original game stemmed from it's control scheme. Because each tool was physically simulated, and to move any tool the player needed to independently articulate each finger on a virtual hand with 5 different keyboard inputs at once, the video game was difficult. Accidently dropping or stabbing the patient was likely. That design was intentional though. *Surgeon Simulator 2013*'s theme, and even it's title, were meant to make failure an engaging part of the experience. Given the video game's reliance on the physical simulation of tools and surgery, it resulted in an interesting case study on how VR recontextualizes that interaction design.

Fig 14 - 15. Comparative Screenshots of the same surgery in the non-VR (Left) and VR (Right) versions of Surgeon Simulator

Like in *Surgeon Simulator 2013*, *Surgeon Simulator ER* (Bossa Studios 2016) tasks the player with safely performing an operation on a patient. The player is given a variety of mostly realistic surgical tools, such as scalpels, bone saws and bone hammers. All of these tools carry over from the original version. Just like in the original version, the video game's closest mechanic analogue is the board game *Operation* (Spinello 1964), because each tool can severely harm the patient if applied to parts of the body not intended to be operated on. In fact, the biggest interaction design difference is in how objects are picked up and held. In the VR version the interaction is done with a single trigger on either of the motion controllers.

However, simply because the interaction is one input and not five doesn't mean the interaction lacks any of the complexity of the original. Player's still must navigate their hands carefully around vital organs to cut or otherwise remove parts of the body in the way of the operation. In *Surgeon Simulator ER* plays even more off it's mechanical similarities to *Operation.* In both, the natural motion of the hands and arms must be carefully calculated to avoid touching off-limits areas of body being operated on. This leads both games to lean into the dexterity of the player's real hands.

Interestingly, this leads *Surgeon Simulator ER* away from the humor the original version heavily leaned on. Without the comically complex mapping of the original game, the humor found in the gulf between the intentions of the player and the accidents that result evaporates. The inherent accuracy of the motion controllers makes it much harder for the player to laugh at the absurdity of task because the task is no longer comically complex. Now, failure is the result of the player mishandling a schapel, not the player fumbling with one of the five keys that control just a single hand.

This suggests that the mere inclusion of movement interactions allows for new amounts of mechanical preciseness in interaction design. Normally, the the amount of precision required to manipulate individual objects would expand the cognitive burden of the interface, because this manipulation results from motions humans are already comfortable with the interface does not become burdensome.

SUPERHOT VR

This is also true in the case of *SUPERHOT VR* (SUPERHOT TEAM 2016b), which makes substantive use of movement interactions. Consequently, it's design showcases not just the mechanical precision possible in VR but the emergent properties of video games designed with movement interactions in mind.

SUPERHOT VR is based off an earlier version of the game, not in VR, simply titled *SUPERHOT* (SUPERHOT TEAM 2016a). In both versions of the game, players are tasked with defeating all enemies in a given level without being shot once. However, the pace at which time advances is proportional to the player's movement. If the player aims slowly, time passes slowly. If the player sprints across a room, time progresses quickly as well.

By designing the video game around motion, in any form, the designers recontextualize common motion, such as aiming and walking, as deliberate choices the player must make in each fight. Simply moving on a whim will almost always advance time so fast that avoiding danger is nearly impossible. A single hit restarts any given encounter, but each encounter plays out out the same way during each iteration. The player can always be assured that a particular enemy will approach from the same door with the same weapon on each try. That regularity makes the challenge of the game revolve around the planning of a multi-part strategy to be executed with mechanically precise movement.

Fig 16. Example of a typical SUPERHOT level

This time-progression mechanic incentivizes interesting strategies in both the VR and non-VR versions of the game. For example. by default there is no way to advance time while taking no physical action. However, pausing and waiting can be a strategic choice if the player would prefer to play defensively. In both versions of the game small, controllable movements like shaking the player's hands or nodding their head become ways of "waiting" for a better strategic moment to act.

The game also has emergent qualities when brought into VR that are not present in the non-VR version. In the non-VR version, there's a throwing interaction where players can toss objects or guns at enemies with the click of a button. This mechanic expanded in *SUPERHOT VR* and requires the player to mime the actual motion of throwing, greatly expanding the complexity and possibility space of throwing. A skilled player can manipulate the force and arc of a throw to hit an enemy that otherwise may not have been possible to hit in the non-VR version.

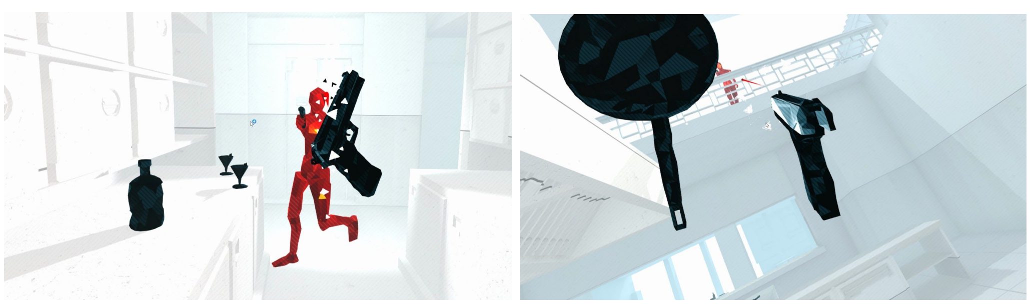


Fig 17 -18. Two examples of blocking, one with a pistol (left), another with a frying pan (Right)

Movement interactions also come into play when considering blocking bullets, an action that only exists in VR. Objects like a frying pan can be used to block bullets by moving the object with one's hands to an interceding position. The ability to do this only makes sense though when the objects' real shape and position must be considered by the player when performing this action. The frying pan works well because it's wide, and blocking with other objects like bottles is possible but more difficult due to their smaller and more condensed volume.

Like *Surgeon Simulator ER*, *SUPERHOT VR* showcases how interface and interaction design can change when designed for movement interactions in VR. However, *SUPERHOT VR* goes a step further and includes interactions that are unique to the VR version and would not be possible in the original. This is a boon for interaction design, because these motions do not necessarily need to be taught, freeing up the game design to explore what could be done with bodily motion.

DEVELOPING OUR APPROACH

This background research lays out two useful considerations for our project.

First, we understood the movement interactions are important characteristics of VR and successful VR video games make substantive use of them. This is due to the sensorimotor contingencies of head tracking in VR, which affords walking as well and highly suggests the use of hand-tracking (Gillies 2016). While there are examples of VR video games that do not utilize hand tracking, from our analysis we felt these were the weakest of the media artifacts we reviewed (Playful 2016; Gunfire Games 2016)..

Second, we believe that there’s not substantive difference between a first or third person video game. Neither perspective can be completely in-sync with player intent and avatar action (Black 2017), and the brain is quite adept at mapping unusual body motion (Won et al. 2014). The gap between player intent and avatar action is the result of the computer necessarily acting as an intermediary between the two. This gap in the fidelity of the play experience will always be an aspect of digital design (Black 2017). What’s more interesting though is how this isn’t a barrier, even when the intermediary is controlling vestigial body parts, copies of existing limbs (Won et al. 2014). While we cannot say what to attribute to this outcome, we can say that interface design with the player body in mind can help players identify their own motion in video games (Mazalek et al. 2011).

Given this reasoning, we return then to puppetry to consider how puppetry aligns with these principles.

Puppetry

The craft, or making, of puppets has arguably existed alongside the growth of the earliest human settlements (Blumenthal 2005) and continues to exist as a form of theater today. Notably, because the practice of puppetry involves the performative manipulation of something external to the body of the puppeteer. Puppetry contends with the question of abstractions of motion. In Eileen Blumenthal’s book *Puppetry: A World History* (Blumenthal 2005), she defines this factor as “the degree of intimacy or distance from the puppet” and describes it as the most important one in determining “the basic nature of any puppetry technique” (Blumenthal 2005).

Blumenthal also provides us with a short taxonomy of puppetry forms, which she places into six main types: Hand or “glove” puppets, marionettes, rod puppets, body puppets, bunraku-style dolls and shadow puppets (Blumenthal 2005). While she readily admits this is a limited classification, and proceeds to problematize each of those classifications in turn, they still stand as a useful framework for thinking about how the human body interacts with the puppet body. Specifically each of these forms interacts with motion differently, and each offers different physical interfaces that afford different kinds of motions.

As laid out by Blumenthal, puppeteers have devised multitudes of ways to manipulate puppets from within these broad classifications. Marionettes, for example, vastly differ from instance to instance. In some instances a marionette may only have one or two strings. In other instances, a marionette may have that used forty-four strings (Blumenthal 2005). Puppeteers have animated three puppets at once while lying on the floor and tapped on a wooden plank to bounce a puppet along to the rhythm of music. In the design of any given puppet's control "interface", puppeteers seek to meet the specific needs of a character.

Fig 19. A marionette violinist made by puppeteer Joseph Cashore (Cashore n.d.)

Puppetry in Video Games

This practice of designing bespoke physical interfaces and puppet body affordances aligns with the way game designers tend to approach designing games. Specifically, game designers take great care in designing interactions that match with the kind of game a designer seeks to make. During the development of Super Mario 64 (Nintendo 1996), for example, the game director Shigeru Miyamoto (Itoi, Miyamoto, and Eguchi 2010) wanted to articulate a specific swimming motion for the game's avatar, and did so by miming it "completely sprawled out on the desk doing these swimming motions", as described in a 2010 interview (Itoi, Miyamoto, and Eguchi 2010).

In other cases, game designers have made not so subtle nods to puppetry already. Some existing video games make use of physically simulated characters as their avatars. This, coupled with unusual interface decisions, make the challenge of the video game based around the player's ability to learn the interface and how it maps to the avatar Two such examples of this design practice are *QWOP* (Foddy 2008) and *Octodad: Dadliest Catch* (Young Horses 2014).

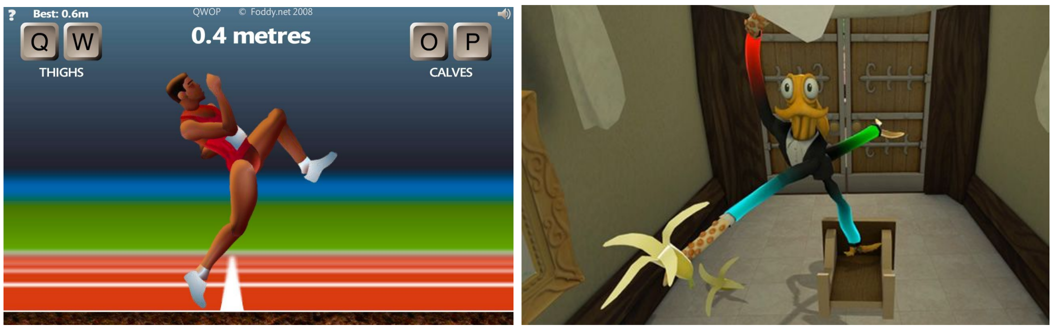


Fig 20 - 21. QWOP (Left) and Octodad: Dadliest Catch (Right)

In *QWOP*, the player acts as an olympic runner, but the avatar's motion is mapped to four keyboard inputs: Q, W, O and P. Each apply force to a specific leg joint, which makes just the act of walking in the game exceedingly difficult. The same can be said of *Octodad: Dadliest Catch*, which has the act as an octopus pretending to be a human. The player moves each limb of the octopus independently, so normal human interactions like opening doors and walking are complex behaviors (as outlined above).

In both cases, the video games sought to design themselves around the complexities of navigating a body. In both real puppetry and these video games, the skillful use of a body through it's interface is itself an expressive act.

Motion and Puppetry

The inclusion of these avatar body motion in existing video games suggests that the use of motion is a useful design practice. We can also say that the design of game avatars share similarities with puppetry design. However, the intersection of the two, the use of motion in puppetry, is our key factor.

We argue that puppetry as a practice extends from a core notion of manipulating an external body, the puppet, using motion from the puppeteers body. This, in some cases, is understood as a distance between puppeteer and puppet. Blumenthal quotes those that suggests that "the degree of intimacy or distance from the puppet" (Blumenthal 2005) is the most important factor. In this case, distance is way of understanding how direct the manipulation of the external body is. In some cases, this is literal distance. Marionettes, for example, cause the puppet body to be physically distant from the puppeteer. Others, like hand puppets, are nearly as close as possible to the puppeteer. But in either case, distance is a way of understanding how the puppet interface mediates motion.

The various forms of puppetry are what we are primarily interested in. They offer different affordances and distances from the bodies they control.. Bunraku puppets, which are multi-person puppets manipulated with rods and strings protruding from the back, offer a close degree of distance from puppeteer to puppet. The cross of a marionette offers a great amount of distance. Again, the strings rods, direct controls and their implementation serve as *intermediary* between the puppeteer and the puppet.

The interfaces we reference, from bunarku to marionette, are what we focus on for this project. They offer a potential way for a VR player to control an avatar they embody. The physical requirements of making a real puppet are now gone, but the physical affordances of the human body exist in VR, even if the avatar is not a real body.

We suggest that the creation of a playable proof of concept prototype video game, which demonstrates an alignment between VR movement interactions and puppetry practices. This is the ideal approach to answering our question, which we now further clarify as "How can puppetry practices inform the design of third-person VR interfaces?"

DESIGN CRITERIA

Given the outlined primacy of movement interactions in VR, the lack of adequate third-person VR game design references, and the fact that interface practices in puppetry act as intermediaries between a puppeteer intent and a puppet's action, we suggest that to answer our question we develop a prototype video game. This proposed video game would suggest, or the start of further inquiry, and not a fully-feature product. What's important to our argument though is that the video game is built upon an VR interface designed after puppetry and movement interaction principles. The primacy of the video game would be on the interface itself, and what we design must then act as an intermediary between a player avatar and a player's actions, which in turn are mapped through the motion of the player's physical body.

This suggests to us a useful distinction between **body** and **pose**. In our definition, body is the *actual* position of the player avatar. In contrast, the player directly influences the pose of the puppet, which we define as the *intended* position of the player avatar. The mapping between the two is then procedural, in that we will rely on some of the same physical laws that govern puppet bodies, as simulated by physics engines, but include digital, procedural elements that support another aspect of the design.

This distinction, outlined in detail below, will provide the flexibility to design goals for the player that toy with the design of interfaces and mappings mapping to avatar bodies.

Mapping of Puppetry to Pose

The mapping of pose as an *intended* position of the avatar body directly defined by the player, is where the practice of puppetry is brought into our project. The manipulation of bodies is the practice of puppetry in the physical world, and we see space for those practices to serve as the baseline design for the digital manipulation of bodies.

We see this as fruitful because the affordances of VR, as defined by the movement interactions of Gilles, with the way puppeteers directly map their hand motion to puppet interfaces and the way they alter the perspective that they view puppets from based on the puppet design.

Puppets often use physical objects in some way attached to the body of the puppet as handles meant to be grasped, moved and rotated by a puppeteer. These handles can be crosses the strings of a marionette are attached to, rods attached to the hands or feet of a puppet or other graspable elements. The key is that a physical object attached to the puppet suggests a useful place to grasp and manipulate it, and that is a feature readily emulatable by our project.

The perspective then, which aligns with Gillies notion of head tracking as a motion interaction, is defined by puppeteer and the interface. The interface often dictates how a puppet is most effectively used, but puppeteers can still contort their own body to view their interaction with the puppet from a different angle. Because we seek to base out design off existing puppetry forms, we will inherit this dichotomy.

Mapping of Procedurality to Body

If the mapping of puppetry to pose only informs the *intended* position by the player, then the mapping of the body to pose becomes the way in which the computer procedurally acts upon the body *based* on the intended position.

We see this as an extension of a model that exists in puppetry and the primary design challenge of the project. In puppetry practice, bunraku puppets are often manipulated by multiple puppeteers, one usually senior to the others. This imbalance of skill means that assistant puppeteers usually control the feet while senior puppeteers operate the hands, something with more performance power. We see the computer operating as the assistant in this paradigm. In that way, we implement the procedural quality of the game in the puppet approach itself.

This method has been used to blend between physically simulate states and animated states in games before, but it's usually intended to be unnoticed, and in our case we need the player to understand that the distinction exists, and they only directly manipulate one of the two parts.

PROJECT

Here we report on the design process, implementation, final prototype and evaluation. We report on each of these aspects of the project separately, though chronologically these aspects were worked on simultaneously and influenced one another. We highlight the design process and implementation first.

We then focus on the final prototype, and describe the pilot IRB study we undertook to evaluate our work. We show our gathered data there, as well as our changes based on feedback and our suggestions for future work.

Design Process

We started our design process with open questions about what kind of video game prototype we would make. This project's focus on video game interface design in VR necessitated a video game prototype, as opposed to just an interface prototype, so that the interface could be contextualized by use. That use however could be anything, as long as the proposed game prototype would allow use to design multiple puppets using different forms of puppetry.

Initial Concepts

One of the first ideas discussed was a inversion of another game's mechanics and structure, *Shadow of the Colossus* (Team ICO 2008). In *Shadow of the Colossus* the player acts as a individual person who fights and kills 16 massive monsters by climbing atop them. The game primarily relies on the scale difference between the player avatar and the monsters for it's engagement, and that difference in scale makes climbing atop the monsters thrilling on its own.

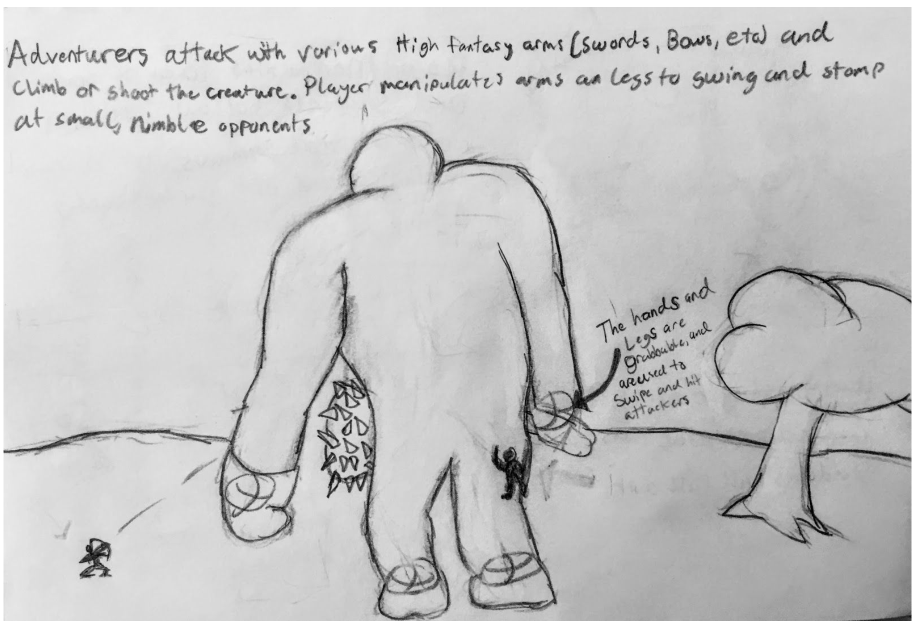


Fig 22. Sketch of boss monster initial concept

In this suggested idea, we would invert the roles and have the player act as an impossibly large monster with small people attacking it. They would climb atop the monster, and the player would need to manipulate the monster's body to fend them off. We liked the notion of playing with scale, as the problem of tracking small characters climbing atop a large create seemed like it would be an interesting challenge in VR for the player. This design suggested that the would be navigating the body of the monster visually, rather than navigating the body itself. This would also retain the core challenge of *Shadow of the Colossus*, which is the same navigation challenge as the human and not the monster.

However, this idea was predicated on a difficult AI and physics problem. AI characters would have to know how to pathfind to a dynamic body and climb atop it. This is no simple task, and from developer interviews it's clear that the monster collision system was custom-made (Nishikawa 2005). Given those issues, we felt that the idea would be too complex to implement.

Another fruitful idea discussed was a game where the player plays a yoga instructor. The player must lead a class of variously shaped, and variably athletic, people through an outdoor yoga class. Each person in the class mimics the motion of the player body, but not every person is athletic enough to do every pose. If they overexert themselves, they'll fall over. The player's job is to give each person a workout without causing a majority of the class to fall over.

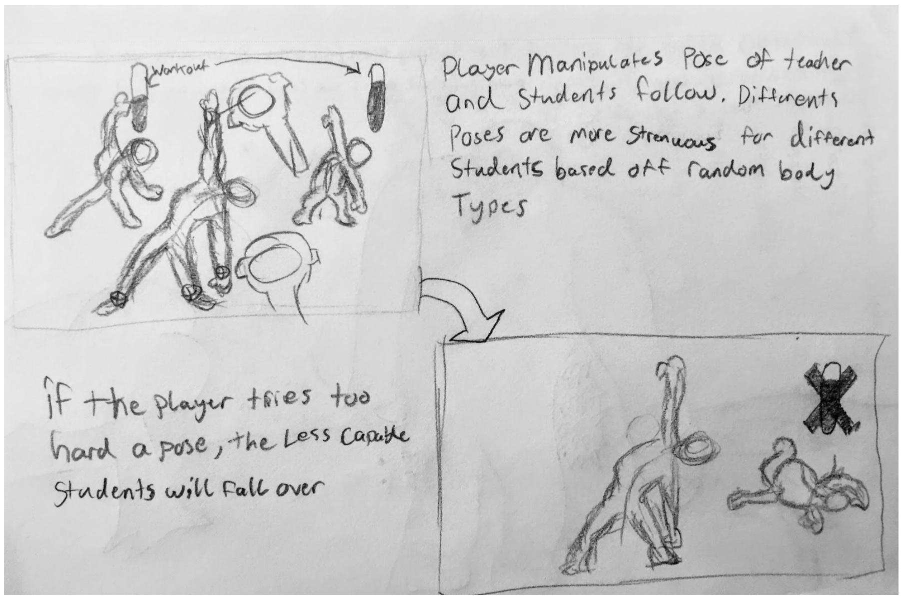


Fig 23. Sketch of yoga instructor initial concept

While we liked the way this idea economically utilized a single implemented character and transposed that interaction onto multiple other bodies. The main technical challenge in this idea would be the random variance of tuning of each body, which seemed about as complicated as making one custom-tuned body. We also appreciate that this idea does not require moving through an environment, and has clear goals for the player.

However, this design would only obviously support a single puppetry interface. While this isn't an issue for the idea, for our argument we sought to present multiple puppetry practices as discrete interfaces.

Our last, and most compelling design, was initially presented as a zoo. In the first version of this idea, the player would be the proprietor of a roadside zoo without any animals. Instead, you have a series of fake animals that you move about *as if they were alive* and trick visitors into thinking they're real.

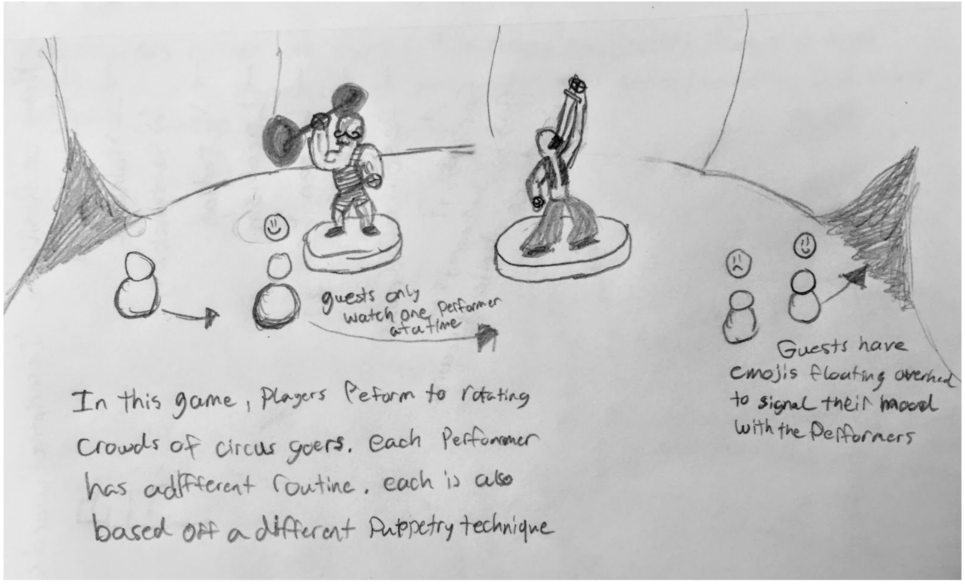


Fig 24. Sketch of circus performers initial concept

We liked this idea, as it played into the performative history of puppetry and had a natural way of including multiple puppet designs by including multiple animals. However, the fact that the design centered on animals was it's biggest drawback. Quadrupeds are more difficult to construct in 3D and rig properly, and by their nature they ability for the player to project themselves onto the body is more difficult. Bipedal humanoids look and behave like our own bodies, and therefore have less of a barrier to embodiment.

However, the idea shifted by changing the context to a circus instead. A circus provided a similar justification for the creation of multiple puppets, by using multiple acts, but avoided the complications surrounding quadrupeds. Furthermore, a circus could operate under precisely the same goal, entertain visitors, but instead of act as an animal the puppets would need to perform a give act. Furthermore, utilizing acts instead of animals would allow for the environmental context of each puppet to suggest it's proper use. Programmatically judging whether an animal is moving as if it were alive is tricky, but judging if a strongman is holding a dumbbell is far less complicated.

First Prototype

Before stepping into a prototyping and brainstorming phase we then needed to develop prototypes showcasing the basic technical requirements of the project in a simple puppet. The first puppet we would need to make would have to answer our design criteria dichotomy; the proposed distinction between body and pose, which would provide by player control via puppetry means and game control via procedural character manipulation.

Based on our design criteria, we outlined a system where the player would interact with puppet bodies through an intermediary interface. This interface would emulate forms of puppetry, and act as the player's way of dictating the pose of the puppet. The interface would also align with the best practice for object manipulation, that "once the object is selected, directly mapping hand movements and rotations onto object manipulations"(Gillies 2016). Given our prior research into other VR video games and our attempt to emulate physical interactions, this outlined for us three possible states to account for in the puppet interfaces: idle, touched and grabbed.

When idle, the interface would need to communicate that it could be grabbed by the player. When touched, the interface would need to communicate that it could be grabbed but is not currently. When grabbed, the interface would need to communicate that it's motion is directly mapped to the player's hands and can be dropped. In addition, at all times the interface needs to communicate that some part of the puppet body is "attached" to the interface, just as a real puppet handle is attached to a real puppet body part.

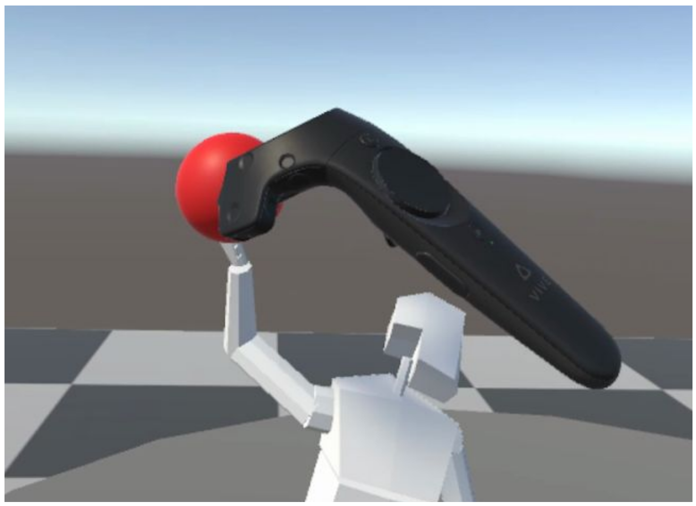
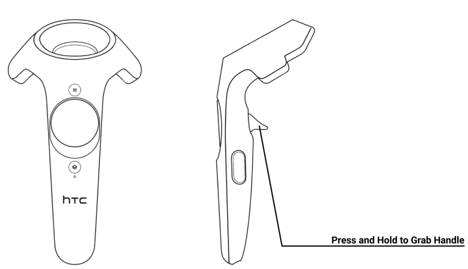


Fig 25. Early puppet prototype

Our first puppet iteration used a very simple version of this interface, which we started calling a handle for simplicity. The handle was a red floating ball that, if touched by the motion controller, would turn yellow. If grabbed, the ball would turn red again and track the player's hand.

The player, to interact with this puppet, only needed to move their hand to intersect with the ball, then press the back trigger on either the Oculus or Vive controller (see figure 43-44).



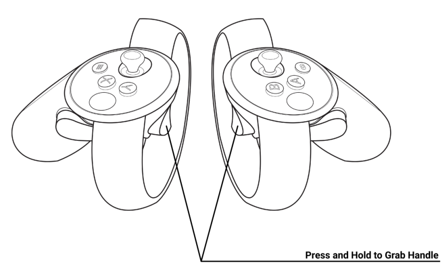


Fig 26 - 27. First designed input scheme

The player, to interact with this puppet, only needed to move their hand to intersect with the ball, then press the back trigger on either the Oculus or Vive controller (see figure 26-27). Players would then hold the handle, and therefore the puppet, which would then be directly mapped to player hand position and orientation. We describe this as a "hold" interaction, as opposed to a "toggle" interaction where the input toggles between an on and off state, as opposed to the state being on only when the input is held down.

Holding occurs when the player presses down on the trigger on the back of either of their motion tracked controllers. Players can only initiate holding a handle if their hand is physically close to a handle, and a held handle is released if the player releases the trigger. We describe this as a "hold" interaction, as opposed to a "toggle" interaction where the input toggles between an on and off state, as opposed to the state being on only when the input is held down.

We chose to design our input scheme this way, and maintain it throughout the design process, for two reasons. First, the use of the back trigger as the grab button in VR design was already a convention in VR video game design (Epic Games 2017) (SUPERHOT TEAM 2016b) (Crytek 2016) (Bossa Studios 2016) and it mimicked the real movement of closing one's hand to grab something. The decision to treat the action as a hold rather than a toggle though stemmed from our use case. We imagined that players would be grabbing and dropping handles often and adjusting their grip, both of which suggested that a fast way to handle object manipulation was more important than additional hand strain.

Handle Interaction Design

When interacting with the relatively simple handles in the first prototype, it became clear that we would need to introduce a design language across all handles. Our intention behind the design of physical interfaces that influence the puppet body's position was for the player to *only* be able to interact with the puppets through their handles. This necessitates that we design a way for the player to visually identify what can and cannot be interacted with.

In addition, the large red balls we used for our initial prototype unearthed another problem. Because they float in air when untouched, it's possible for them to obscure, or become obscured, inside the 3D model of the puppet. We needed a way to make sure the handles were visible to player at all times, but also a way to have them not obscure the puppets themselves.

Finally, our color switching system when touching the handle worked well as an indicator of when the player could or could not grab the handle, so we wanted to expand that design language to encompass a clear visualization of the state of the handle at all times.

Fig 28 - 29. Two early versions of the handle design

We experimented with a few different ideas for handle design, one of which we ultimately didn't use but ended up leading to our final design. In this experiment, we used a 2D sprite, which would highlight if touched, as a universal handle identifier. This proved unsuccessful because it did not indicated the current orientation of the handle, only the position. However, in developing this iteration, we ended up needing to keep the sprite always facing the player's head (least they notice that it's a flat image). This led us to thinking about other ways rendering process could help.

The issue of orientation was mitigated by reverting to a 3D handle. However, this time we tried writing a shader that rendered the 3D handle as one solid color, with an opacity value, and always in front of any other 3D object. Normally an programming abstraction known as a depth buffer determines which objects are closer or further away from the virtual camera. Our shader overrode that, and rendered the handle as long as it was in the field of view of the player.

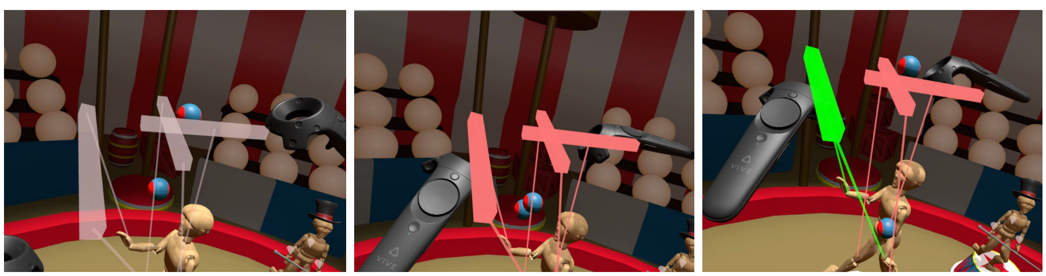


Fig 30 - 31 - 32. Idle Handle (left), Touched Handle (middle), Audience/Grabbed Handle (right)

This proved successful, and resulted in the the nearly final version of our handle design language. The last aspect, color to indicate state, went through a few variations. We opted to use a solid reddish-hue to make the change distinct to the player from the partially opaque grey of the idle state. We also used a bright green to indicate that the audience (whom the player is performing for) likes the motion the player is doing.

Puppet Concepting

While we were iterating on handle design, we also began to concept additional puppets. We started this process by brainstorming a number of different designs based off different forms of puppetry, which included the following.

- A Stilt Walker, which would be based of a marionette and focus on using a handle to emulate a walking motion from above.
- A Mime, which would be based off Bunraku and use an invisible collision box to mimic the mime “trapped in a box” routine
- A Fire Eater, which would be based off rod puppetry and expel a particle effect of fire from the mouth when the hands, holding torches are raised to touch the mouth
- A Unicyclist, which would be based off rod puppetry and have the player move the puppets feet to push pedals and the unicycle body.
- And a Contortionist, which would be based off Bunraku, and use extremely loose joint constraints on the puppet body to simulate the flexibility of the human body.

Through our brainstorming, we concluded two things. First, we did not have the expertise to model this many different puppet models. We had a simple boxy character we were using for early development (see figure 25) but we felt it did not give the impression of a deliberately chosen aesthetic.We decided to instead use an educational puppet modeled for *A Character in Your Hand*, which was modeled by a Georgia Tech student named Kalani Strange. Second, we concluded that there were two forms of puppetry we would avoid for this project due to hardware and structural factors.

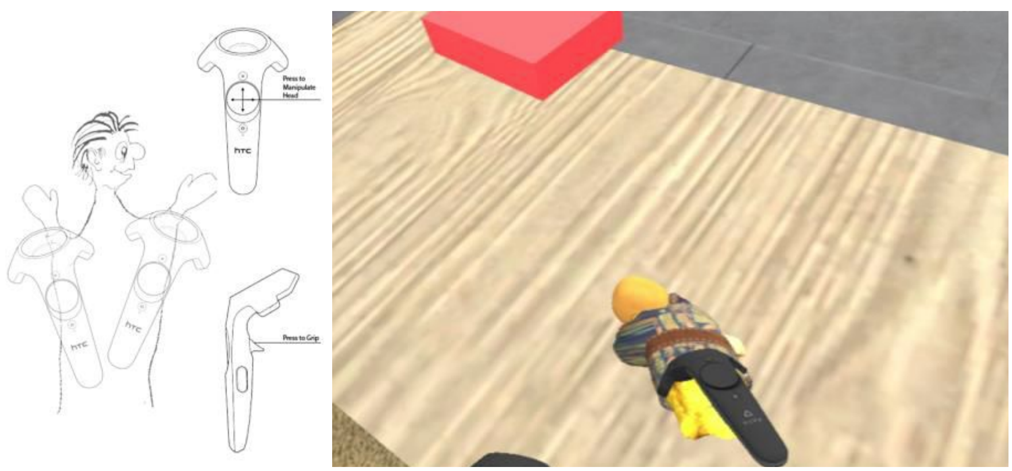


Fig 33.Hand puppet design from A Character in your Hand (Nitsche and McBride 2018)

First, we struggled to imagine how we could utilize hand puppetry due to hardware limitations. Hand puppetry prevalence in the medium is well know (Kermit the Frog, for example, is a hand and rod puppet). However, in our case we lacked the use of fingers. The motion controllers are adept at tracking the position of the hands, but only the Oculus controller has rudimentary finger tracking. In addition, Pierce and Michael had attempted to design a hand puppet for *A Character in your Hand* and were largely unsuccessful (Nitsche and McBride 2018) .

The other form we failed to make use of was shadow puppetry. Technically this form was easy implement in VR, but shadow puppets are often flat and only designed to be manipulated in 2D. For a shadow projection, this is ideal, but wasn't as suitable for the 3D environments required for the other forms of puppetry.

As we progressed, we tried new variations of these designs and found additional stumbling blocks. At one point, we attempted to design fire dancers, which would use a spinning rod with fire effects at each end. This ran into complications however when we realized that the puppet had difficulty keeping the arc of the rod out of the way of the body. The body continually obstructed the rotation of the rod, and we had difficulty giving the player enough control over the orientation to alleviate the problem.

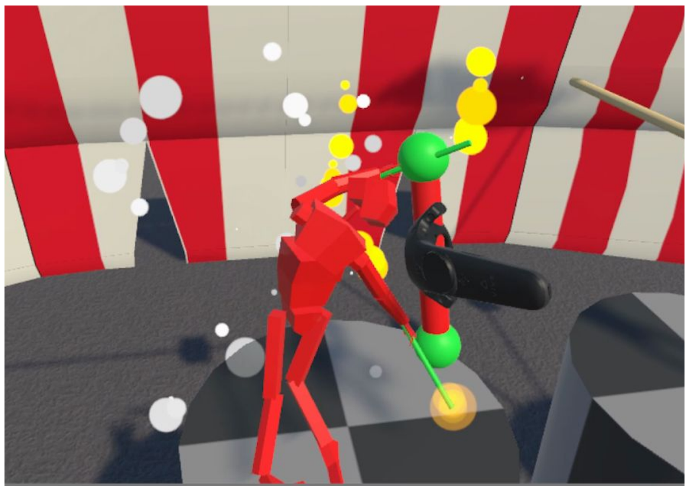


Fig 34. Attempted design of a fire dancer

We also attempted a unicyclist, but ran into issues recreating the functionality of a wheeled vehicle. A wheeled vehicle, like a unicyclist, would not be able to travel orthogonally to it's forward vector, meaning wheels can't strafe. We wanted to simulate the act of actually riding the unicycle though, so we needed instead to design a unicycle that could reorient itself over time by rotating and traveling forward, like a car. This ran counter to the affordances of the handles though, which suggested freedom of movement not present in a unicycle context.

We eventually landed upon three types of puppetry to use matched with 3 different performer designs: a bunraku strongman (figure 35), a balance beam rod puppet (figure 36) and a marionette juggler (figure 37).

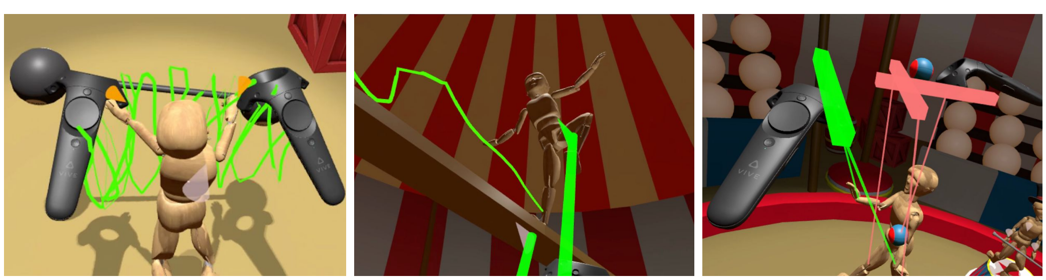


Fig 35 - 36 - 37. Strongman (left), Balance Beam (middle), Juggler (right)

Here we outline the design considerations that went into each of the three final puppets.

Strongman

The first puppet we prototyped, and the first we worked on further, was the strongman. The strongman circus performer, or a performer who showcases feats of strength. These feats of strength are often performative over athletes as well. Weightlifters, for example, perform standardized deadlifts or weight lifts, while a strongman may bend steel, life a heavy weight with one arm or pull a train car. In the cartoonish representation of many circus performers, the strongman lifts a comical weight over their head in a leotard.

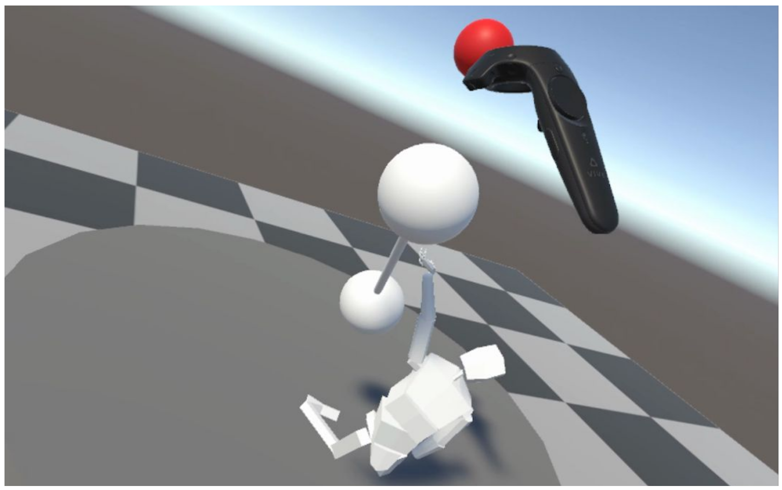


Fig 38 Early Strongman prototype

The strongman is a remediation of the bunraku style of puppetry, where multiple puppeteers control different parts of the puppet body but none have complete control of the entire performance. Individual puppeteers often only control the arms or legs, and work together to bring life to the performance. They often also wear black, to draw attention away from themselves and towards the puppet.

Fig 39 - 40. Bunraku Performance (Left) and Bunraku Design (Right) (Hil 2005)

In our remediation of this design, the player only controls the arm and the body with handles. This was the rough implementation of our first iteration of the strongman and remained the basic design throughout the design process. However, the strongman also had a unique feature among our puppets. It had to interact with another object, the barbell.

Early versions (see figure 38) had the barbell permanently attached to the puppet's hand. This worked, but we wanted to prototype an alternate system where the player can pick the barbell themselves *through* interacting with the strongman. We wanted to do this to show that our interface is flexible enough to support secondary actions on top of controlling the avatar's position. In addition, this kind of action in puppetry is a complicated endeavor. A puppeteer might use glue or velcro, or simply have the object affixed to the puppets hand manually, but would likely not be able to fully act it out.

Digitally, it's a far simpler interaction. If the player is holding one of the two arm handles and the arm is nearby an object, pressing down on the touchpad on the Vive controller or the second trigger on the Oculus controller will cause that hand to grab the object. We use this interaction to support lifting the dumbbell, but we also support lifting crates and barrels in the environment to show the interaction's flexibility.

We see this puppet design as a potential answer to a basic premise of many game designs. The ability for the character to grab or manipulate other objects is a common feature. This is sometimes done automatically, in that the character picks up objects without player input. Other games however make use of pushing or pulling boxes, crates, or swinging weapons and tools. We designed this grabbing system to be flexible enough to potentially support any of these hypothetical interactions.

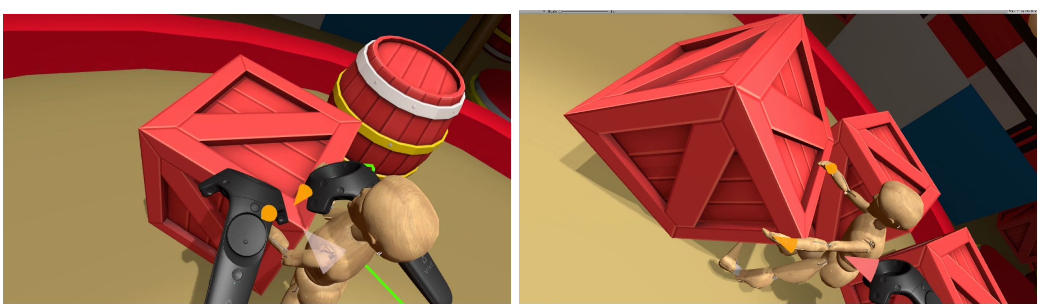


Fig 41 - 42. Strongman grabbing a crate (left) and pulling it (right)

Balance Beam Walker

The second puppet is a balance beam walker, or a performer who walks along a narrow beam high above the circus audience. In a real circus context, this act is often either a part of a trapeze performance or a tightrope, not beam. In our context, the player walks the puppet along a straight, narrow beam from one end of the tent to the other. The player only has control of the feet while the body and arms are procedurally controlled.

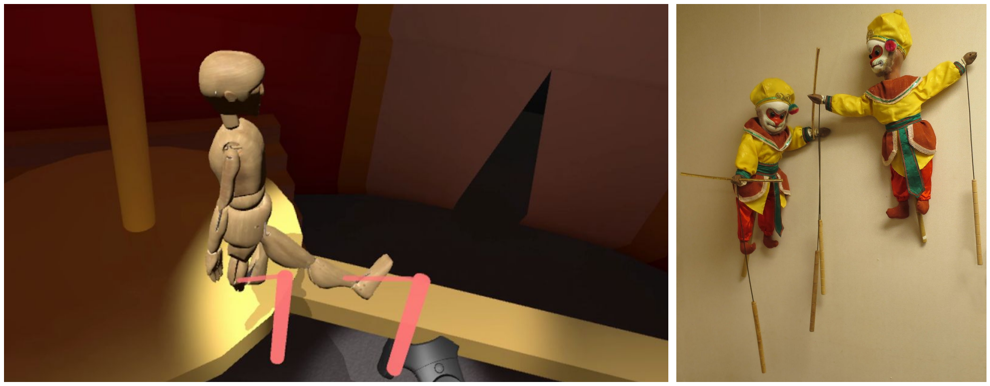


Fig 43 - 44. Early version of the balance beam walker (left), rod example (right) (Hiart 2014)

The balance beam walker is a remediation of the rod style of puppetry, where rods are attached to points of the puppet's body and are used as handles. This technique allows the puppeteer to act upon the puppet from a distance, usually from either above or below the stage. The audience in this case is watching for motion on the feet, so the rods are attached to the puppet's feet, similar to the way in which the rods in figure 44 are attached to the puppets arms.

We intended to use balance beam walker to showcase interacting from below upon a player avatar, which afforded us the use of a balance beam above the player's head, as well as the audience. This gave us a chance to test a different camera perspective as well, as this puppet in it's final design sits above the player at all times. In addition, the central task in this case, walking, is a core action in both puppetry and video games.

A vast number of video games involve the navigation of space, whether that space is a cluttered room or open field. This necessitates the use of walking or running in video games where the player controls a person or humanoid body. Like grabbing, this action is usually abstracted to an analog stick or button, but in our case we decided that the performance of the action mattered when considering both the affordances of the puppets and the design practices from puppetry. However, because this is a different interface for a different puppet in a different context, we use this design not as an argument that third person VR interfaces should support walking and grabbing at once, but that puppetry interfaces could support either. The choice of what to support would depend on the intentions of the designer or the context in the video game.

Juggler

The last of the acts is a juggler balanced upon a ball. This is not strictly a real act, but instead an amalgamation of multiple different acts. Juggling is a common circus or parlor trick, and sometimes in some cases it's done in a precarious situation, such as on a unicycle. It's not however often done on a large ball. That aspect is derived from cartoon representations of circuses.

The juggling is automatic, meaning that the puppet juggles the balls regardless of how the player interacts with the puppet. If, however, the player is manipulating the position of the arms, the audience begins to grow and cheer.


Fig 45 - 46. Early version of the juggler (left), marionette example (right) (Jacklee 2014)

The juggler is based off marionette puppet design, where the puppet body is the suspended from strings attached to a handle. Some designs use multiple handles or handles that can attach and detach from one another. Our design works similarly, where the juggler has two handles. One handles controls the position of the body and legs, while the other handle controls the arms. The handle for the arms is the one that, if held, the audience will cheer for.

This designs use of an automated juggling system was intended to showcase a way for certain actions to be abstracted from player control. Unlike our other two designs, which take an action and enact it in some literal fashion (grabbing or walking) we wanted to make it clear that even with these interfaces we could abstract away complex motion. This isn't to say that a form of juggling couldn't be designed for with our interfaces, but if the player needs to focus on multiple complex actions it's still possible to abstract one from the player's workload.

In our case, keeping the ball below the puppet's feet was the locus of player attention. The ball tracks the position of the body handle, and keeping the ball below the player's feet gives the appearance of balancing on the ball. While this isn't necessary, the handles won't fall and neither will the puppet, the performative act of balance resembles the act of finding a puppets gravity. Giving a sense of weight to motion is one of the hardests tasks for a puppeteer, and our juggler puppet resembles that challenge.

Player Body

While designing the puppets themselves, we also had to consider the question of scale and ergonomics. Because we what little VR references we could consider used a relatively small avatar (*Lucky's Tale*'s avatar and *Chronos*'s avatar are both quite small), we decided to use a small size as well and considered feedback as it came from testers. Our puppets are all about a foot tall.

In addition to scale, part of our design criteria involved the placement of the camera. However, because of our game structure, that of a circus, we could largely avoid the question of how to handle the player body and instead build around the features inherit in VR.

We did however adjust the environment to suit our design, but it better understood as moving the player body to adjust rather than moving the environment. During the creation of the very first puppet prototype, we realized that the scale of the puppets would be impactful on their design.

Our initial approach was to scale them close to what they would be in real life, approximately a foot or so tall. However, this led to discomfort because we had to lean down to use them. We accommodate this issue initially by using a pedestal and testing puppet prototypes on top of it (see figure 34 and 38). This worked, but we quickly realized that we were not bound to the same concerns as physical puppeteers. We tried moving the player the player down, so that the floor of the circus was roughly at waist-height. This removed the need to place puppets on pedestals and allowed us to design an normal looking circus environment.

Game Structure

The overall game design of the project, which we describe as it's structure, came next. Although we knew from the start that we wanted to make a video game prototype about performance, where each puppet is an act that performs a specific way to keep the crowd entertained. That design still had gaps though that we needed to fill. The two design gaps we could imagine were how would the player know the state of the crowd and how would the player know they were performing correctly for them.

Considering the first gap, our original intention was to use a moving cast of patreons walking in and out of the tent and looking at one puppet at a time. They would become happy and leave if the puppet they looked at performed for them. This would lead to a "spinning plates" design, which we felt would offer enough game structure to evaluate our interfaces as game interfaces. However, due to scope concerns we started to adjust that idea and instead increased or decreased the number of observers in the grand stands as the player performed with any puppet.

This need to perform with each puppet lended itself to a background score for each.We would keep track of the crowds enthusiasm as a score and give feedback based on that score. We didn't want to just show a number, so in our first attempt we opted for a meter above each puppet. The meter would track motion on a part of the puppet's body and would increase if that part was moving, or decrease if that part was not moving.

Fig 47 - 48. First version of a score deteriorating (left) and increasing (right)

This proved a poor solution. The meter was almost never visible at the same time as the puppet it related to was. This made it easy to ignore. In addition, the meter suggested that letting it go empty cause something to happen. This also wasn't the case. The meter did however help by forcing us to identify a set of colors, which we used on the meter and on a small ball attached to a part of each puppet we called the motion tracker. That ball would match the color of the meter so the player could focus on that rather than the meter itself.

This part of the design had promise, and we instead collapsed all of the system into the motion tracker, which would create a trail as it moved that would indicate that they had used the puppet recently and give a interesting looking visual effect as a reward for play we designed for.

We then built out a two ring circus with two different crowds to manage. We and intended to do this because the puppets felt cramped together in just one ring. This proved difficult though because it didn't fit into the walkable area afforded by the VR hardware. While not ideal, this is where instead decided to show only one puppet at a time and have the player move between puppets instead of interact with multiple at once. However, we lamented the loss of multiple puppet use at once, so we included a ringmaster puppet, who worked exactly like the strong sans the pick interaction, in each act. The ringmaster does not influence the crowd at all. It only serves to allow for multiple puppet use and acts as stage-setting for the environment.

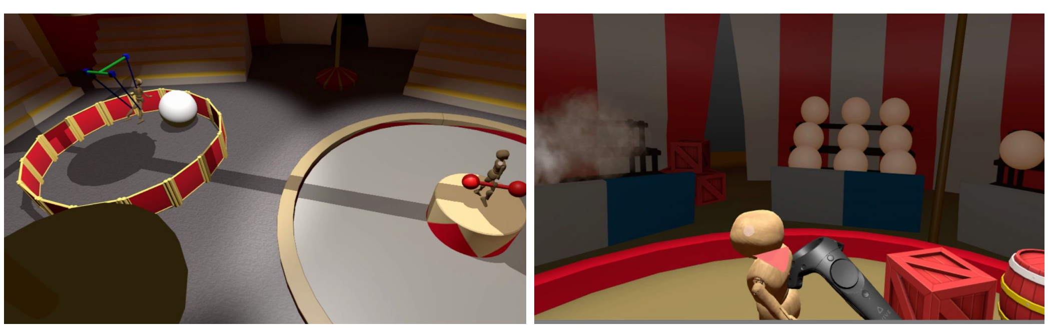


Fig 49 - 50. Two-ring circus implementation (Left), one-ring with crowd (Right)

Implementation

The final version of our project is a circus-themed video game prototype where players progress through three acts, each of which uses a different puppet as the start of the act. Each puppet is based off of a different archetypal circus performer and has a particular task, which if done by the player the audience will expand and the volume of their applause will increase, The volume and size of the crowd deteriorates if nothing pleasing to the audience is done by the player. The task specifics differ for each puppet but revolve around the manipulation of each puppet in a particular way, as informed by the environment and circus performer inspiration.

We used the game development engine Unity as the development environment for this project because Pierce had previous experience with the engine. In addition, *A Character in Your Hand* also used Unity, which allowed us to carry forward assets and learning. Lastly, Unity is compatible with both Oculus and Vive, and with the use of the VRTK framework we were able to develop for both platforms simultaneously. Our prototype runs on Windows only.

The actual work for this project arguable began when Pierce and Michael worked on *A Character in Your Hand*. Because of the continuity of researchers involved, we started the project withe the same solution and progressed from there.

Physics Constraints

In *A Character in Your Hand,* puppets were built using ragdolls, or rigged, physically simulated limp bodies. Using Physx, the built-in physic simulation engine in Unity, a rigged 3D mesh can approximate the motion of a limp body fairly easily by treating each bone as a connected but discrete physics object. This gives ragdolls a similar plasticity to the real life toy counterpart to the name.

The player then interacted with the puppets using joint constraints, which are a technique whereby two physics objects maintain a pre-specified distance and relative orientation to one another. This system works well in screen-based 3D games, but in VR the player hands are not subject to the same constraints as the virtual objects. If a player were to pick up an ragdoll by two seperate parts, such as the two arms, the player could easily move their physical hands well passed the constraints set by the virtual object. This leads to instability in the joints (see figure 51-52).

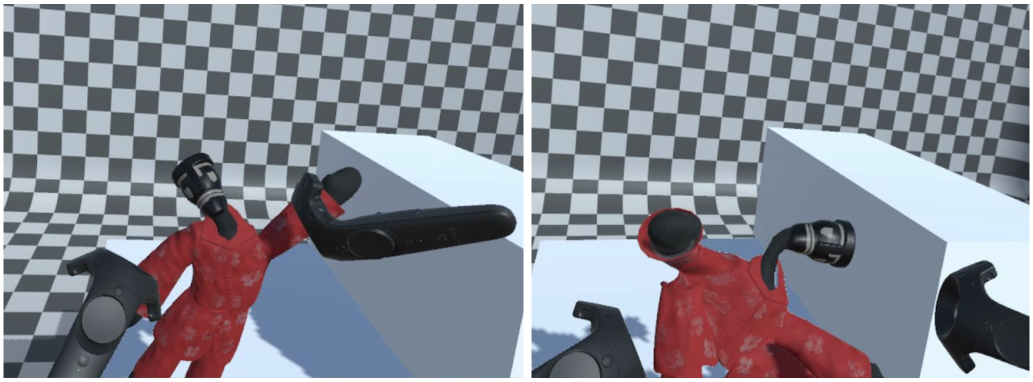


Fig 51 - 52. First technical prototype, where ragdoll can "break" and deform (Right)

part of our reasoning for a distinction between the body and pose of the puppet was to alleviate this problem. From a design and technical perspective, the player would never directly grab the puppet or ragdoll, and therefore would not be able to overextend the constraints of the puppet. This technical separation would both emulate puppetry design and solve for an otherwise difficult problem in VR interaction design, which more often than not make it impossible by not allowing a object to be held by two hands.

Multi-Skeleton Solution

Our actual technical solution though came from the combination of two sources: procedural animation and industrial. The technique known as procedural animation has been growing in use since *Octodad* (Young Horses 2014) and *Gang Beasts* (Boneloaf 2014) made ample use of ragdoll avatars animated through little to no traditionally pre-authored motion. Instead, both video games use a ragdoll which is physically simulated using the physics engine but held up and animated through targets which apply force to the puppet in a direction picked by the player. This technique shares similarities with our design solution of separating body and pose, but the technique is not well documented so we could not base our programming off of a framework or documented pattern.

Fig 53 - 54. Octodad ragdoll avatar falling (Left), Gang Beasts ragdoll avatars holding one another (Right)

What we could discern from playing the two video games in question and through developer interviews was that the systems appeared to work by setting targets for specific bones and applying force to those bones until them matched the position/orientation of their targets. One particular talk by a developer using a similar technique described applying force to the chest upwards to "float" the chest towards the target, then moving that target to simulate walking (A MAZE. 2017).

These techniques proved influential, but because our system allows for direct manipulation, we needed a way to grab an individual part of a puppet body. Simply floating up works with the chest but not with the rest of the ragdoll. Instead, we devised a technical abstraction based on our design criteria. In our system, there would be two skeletons. One of which would be targets, and not physically simulated. The other the actual ragdoll bones, which would constantly have force applied towards their target.

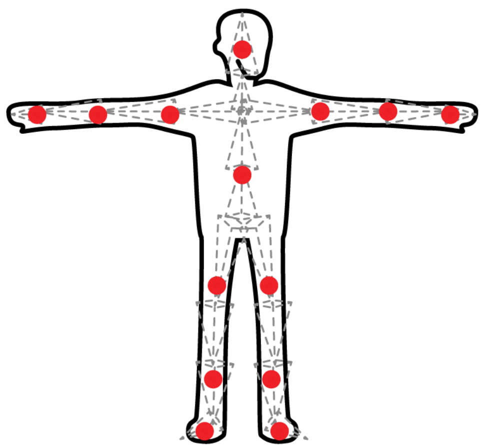


Fig 55. Two skeleton diagram. Red dots are targets, dotted ovals are bones.

This is outlined in figure 55. The red dots represent the targets, of which there are one per bone. The dotted outline represents the bones themselves. Neither the bones nor the targets need to be visible, they instead influence the body, which is a skinned mesh. The actual implementation of the target tracking system used distinct variables for each bone that affected how closely the bone matched the targets position and allowed the player to grab any target at any time. Each target would be constrained in the same was as the ragdoll skeleton, so constraints would keep both in sync with one another.. This made it very slow to tweak though, because each change needed to propagate across all targets.. In addition, tracking this many targets and once made the physics simulation very rigid. It was somewhat like an artist's mannequin, which holds it's position very closely.

Actual puppets are much looser in real life. Puppeteers in fact take advantage of this in order to help them perform. Kermit the Frog, for example, has a recognizable shake motion that is used to show enthusiasm, which is primarily just the puppeteer shaking the puppet back and forth and letting the arms move free. This kind of action wouldn't be possible in our current version.

Fig 56. Kermit the Frog shake motion(Fun Kids 2012)

To correct for this, we revised our two skeleton system slightly. We instead switched to a *partial* two skeleton system, which would be determined by each puppet's design. In this version, the bones that would have targets are the ones that have a handle attached, or have a need to be constrained for any given reason.

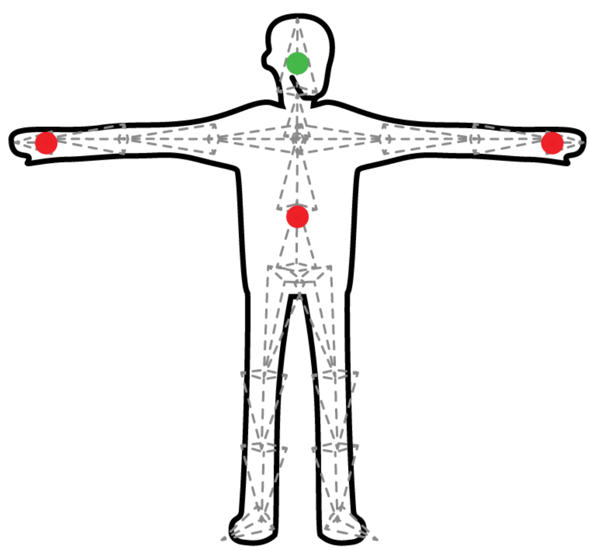


Fig 57. Revised Two-Skeleton system. Red dots are handles/targets, green are hidden targets

In figure 57, you can see that only the arms and chest have red dots, indicating targets. In this puppet, only the endpoints of the arms and the center of mass are controlled by the player. Just like a real puppet, the rest is all physically driven. The lone green dot represents a invisible target that can be used if a need arises. In this case, we often encountered heads which would bob endlessly. We solved this with targets, but that tweak wouldn't be necessary to show or communicate to the player.

PID Tracking

Despite our progress on the technical framework however, we still encountered issues where bones would not completely track the position of their targets. Despite lowering the number of targets we still had to endlessly tweak our target variables in order to properly simulate physics tracking. Up until this point, we were developing this part of the system without a plan or structure. We instead simply applying more or less force as needed.

This problem was solved accidently though, when we were investigating the best way to rotate each bone to match it's target. 3D rotations uses a 4D abstraction known as a quaternion to prevent a edge case of 3D rotation known as gimbal lock. This however makes the math complex, and in our searching for help we stumbled upon a pattern known as a PID controller

A PID (proportional-integral-derivative) controller is a feedback loop mechanism used to continuously modulate a measurable value *(y(t))* towards a desired value *(u(t))* by measuring the error value *(e(t))* or difference between the two and adding together the proportional, integral and derivative terms derived from the error. Aside from the desired value, current value and error, the pattern also takes a *P*, *I* and *D* coefficient that are each tunable based on desired outcomes. Increasing the *I* value, for example, increases the speed at which the pattern reaches the desired value, but increases the likelihood of the value overshooting. The *D* value, in contrast, mitigates oscillations, or the overshoots caused by the *I* value. The process repeats itself ad infinitum.

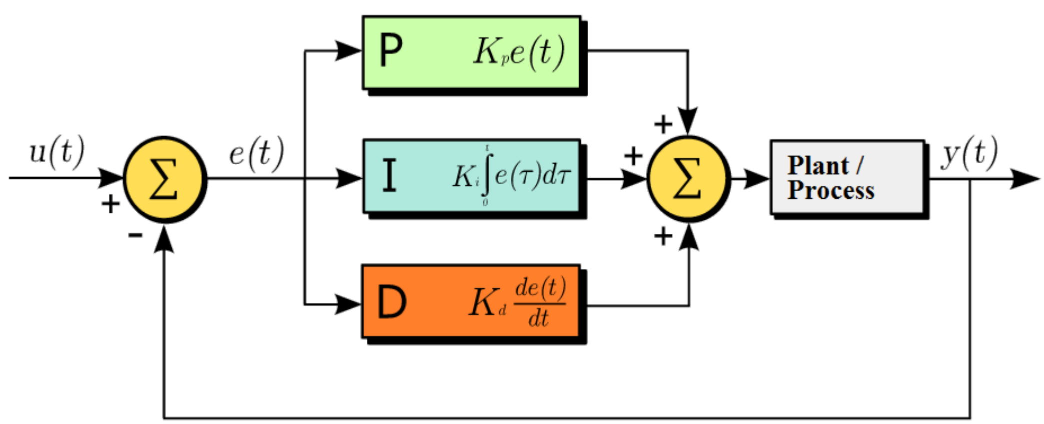


Fig 58. PID controller flow chart and formula (Chang and Warren 2011)

This pattern gave us an abstract way to provide our bone a target's positional X, Y, and Z value, as well as the targets rotational W, X, Y and Z value and return a iterative step towards those values based on the error between the target and the bone. This pattern proved extremely flexible for our needs.

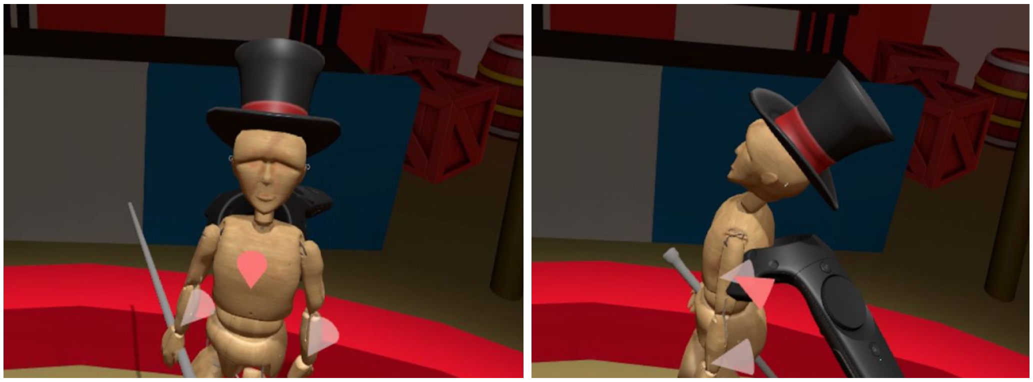


Fig 59 - 60. PID rotation tracking example. Note the orientation of the cone

This pattern still requires tuning, but because this pattern is an established community of people with suggestions on tuning strategies (Stack Exchange 2012). The benefit of this system being predictable tunable is that we can then apply tuning values to different puppets and expect similar results. In addition, this allows us, the designers, to determine how quickly any given body part should track the position and orientation of it's handle and with how much force, or overshoot. We see the use of a PID pattern as a potential abstract solution for developing similar video game prototypes in the future.

EVALUATION

Once we had completed the first version of our prototype, we undertook a pilot IRB study modeled after the *A Character in Your Hand* study. We specifically examined the prototype using the creativity support index (CSI), NASA Task Load Index (TLX) and asked participants open-ended questions on their feedback regarding the prototype. We tested 5 participants, 3 female and 2 male, each of which first completed a brief demographics and background questionnaire. Each participant then played with each puppet in the prototype and practiced think-aloud. Afterwards, each participant filled out the CSI survey, then the TLX and finished with a open-ended discussion. The entire study took approximately 30 minutes for each participant to complete at a time.

Our sample size is far too small to be quantitatively valuable on its own, so we won't focus much on theses results. We will report that our average CSI score across all participants was 82.74 out a 100 point score, 0 being not at all supportive of creativity and 100 being the most supportive of creativity possible.. For comparison, the authors of the CSI report an 87.73 score for Google Docs as a creative writing tool (Cherry and Latulipe 2014). This places our score fairly high (a B by the index author's standards). This score also aligns with our informal feedback, where multiple participants mentioned the creative aspect of our interfaces.

Our TLX data can be viewed below, where each measure is rated between 0-5 by all participants, 0 being lowest on the measure and 5 being the highest. Without more data for aggregation, this data must be considered with caution. However, we note that the overall frustration score is low, while the the performance score is high.

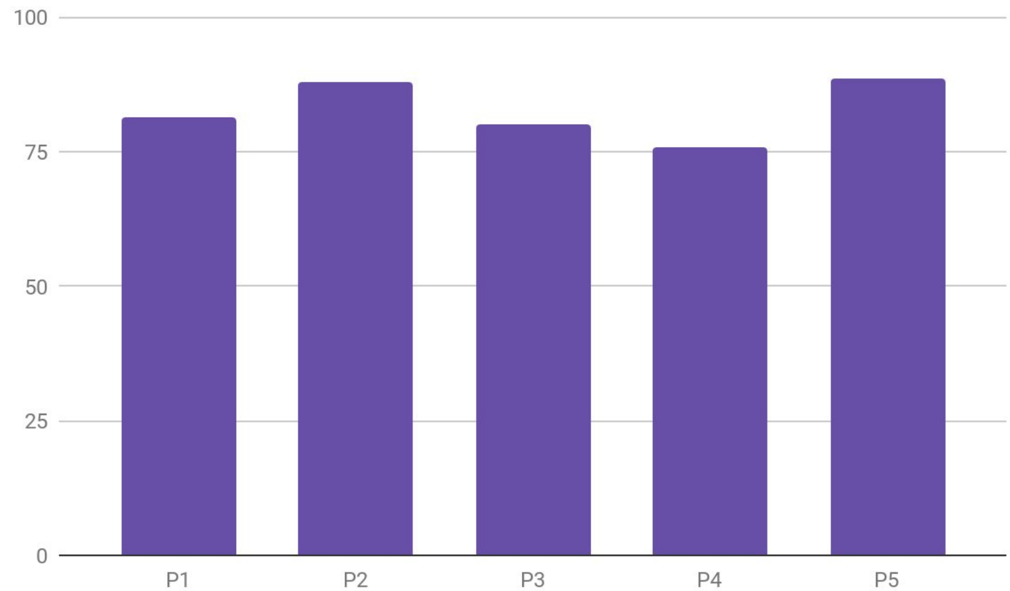


Fig 61. CSI Results

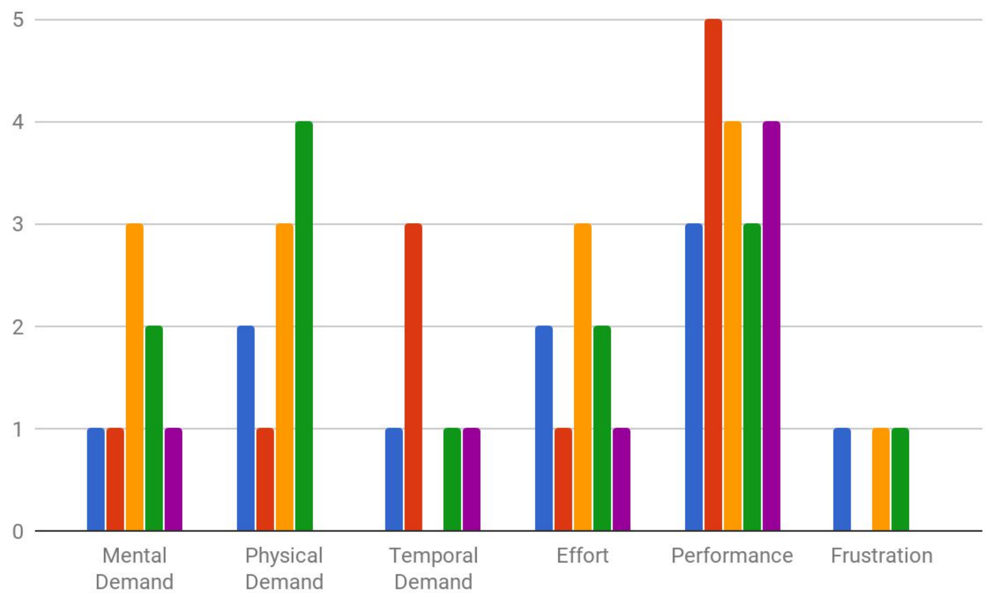


Fig 62. NASA TLX Results

Unlike our quantitative feedback, we found our informal feedback sessions quite illuminating. In these sessions, we asked participants a series of open ended questions, which were as follows.

1. What was the most enjoyable part of the experience? Why?

2. Least Enjoyable Part? Why?
3. What features did you miss?
4. How do virtual puppetry controls relate to physical ones?
5. Forward going, where do you see the most value for such an approach? Where could it go wrong?
6. Are there any additional comments you would like to give on the project or study?

From our five participants, three generalized lines of feedback emerged. First, multiple participants compared the experience of interacting with the puppets to that of toys. Many participants mentioned dolls, dollhouses, Barbie or GI Joe in their informal feedback. P3 (participant 3) in our study made a comparison to play with toys a kid, and specifically was interested in physical simulation of the puppets and how much it resembled known toys. P4 mentioned that they made an effort to move the controller hold the strongman's body in a manner similar to how they moved toys as if they walked. They tilted the controller, and therefore the body, slightly to the left and right as they moved it.

The perception of the puppets as toys, however, was an interesting finding. To us, it suggested that, even if the player does not perceive the self in an external body, there is still a wealth of play patterns to draw from for VR game designers. P3's actions are most interesting within this context. P3 describe the act of playing with the strongman "like playing with my barbies when I was a kid". During testing, they grabbed the ringmaster and strongman and wanted them to hold hands. They moved the two puppets close together and then walked them in unison, which supported their impression of toy play.

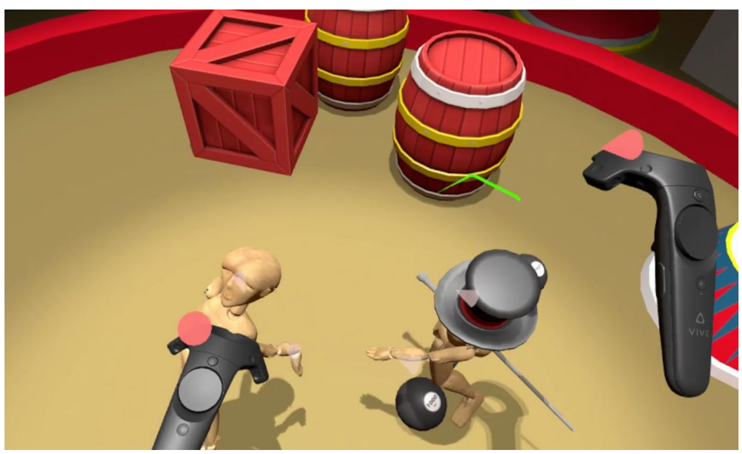


Fig 63. P3 moving both the ringmaster and strongman at once.

Second, because many participants saw the puppets as toys and not as themselves or their externalized body, we cannot report that our prototype elicited a sense of embodiment. P3 again mentioned that, because they were toys, they wanted to treat them as such and toss them around the room or rapidly shake them. P5 made a similar comment, talking about the desire to toss them about the room. This perception of the puppets as toys also led all participants to mischaracterize the procedural standing height of the strongman as a bug, or assume it to be a failure of understanding on their part.They reported expecting to be able to lift the puppet up as far as they desired, like a toy and not like a body with physical limitations.

This feedback, to us, suggests that despite the plasticity of the mapping of the brain showed in prior research it may be more difficult to separate oneself from one's body in VR than anticipated. This could be due to the presence off tracked hands and head motion, or due to the toy-like appearance of the puppets. P4 was another participant who reported wanting to throw the puppets. When asked why, they stated that (speaking about the puppets) "their size and appearance made me want to. They kinda looked like puppets. Though, if they looked like a dog or something I probably wouldn't want to throw them". This comment suggests that if the appearance of the puppets was related to this attitude.

Fig 64. P4 attempting to throw the strongman

One interesting addendum to add to this second finding though is a particular comment made by P2. In our prototype, the haptic, or vibration, feedback provided by the system is delivered while holding a handle. P2 commented that they found this feedback confusing, and wanted that feedback not when their *virtual* hand touched and grabbed a handle, but when the *physical* puppet body hit the environment. They however reported in the feedback session that they were an "actor in the scene, and not some sort of god looking down on everything". While interesting, the reported impression of the self is mixed when compared to this suggested feedback.

Third, all participants at some point lamented the lack of a "goal" or "task". When asked to clarify, participants cited uncertainty in what they were supposed to do with the puppets. The feedback was most pronounced with the strongman and juggler. The balance beam however was not cited as one where the task was uncertain, and even was cited as a favorite by P2, P4, P5. P4 as well specifically cited their uncertainty in goal during the strongman portion, and immediately remarked upon using the balance beam puppet that they liked this design better, as P4 described it, the balance beam puppet had 'the most purpose". P4 also mentioned the ergonomics of the balance beam puppet, citing that it was more comfortable to use in comparison to the strongman, which stood at roughly waist height.

This feedback did not surprise us, as we expected the game structure to the weakest part of the prototype. However, we were surprised by the reference of the balance beam walker as some participants favorite. Given how simple of a goal the puppet had (walk along a straight beam), it's interesting to consider how much that improved the player experience.

Changes Based On Feedback

With this feedback in hand, we sought to make what changes we could with a limited amount of time left in the project. We identified two relatively straightforward design changes we could make that we agreed would improve the project from our feedback sessions.

First, P4's comment on the ergonomics of the project was one we had already considered, but could further adjust. We already had placed the "floor" of the circus at approximately waist height for greater ease of use. Rather than need to bend down, users could interact with the puppets from a standing height. However, based on P4's comment we raised the floor further, this time to about chest height.

This had two ramifications for use. First, the one intended. The strongman puppet was far easier to grasp, now that it stood at chest level (see figures 64-65). However, this resulted in the balance beam walker's rods needing to be longer, as it was now out of reach (see figures 66-67).

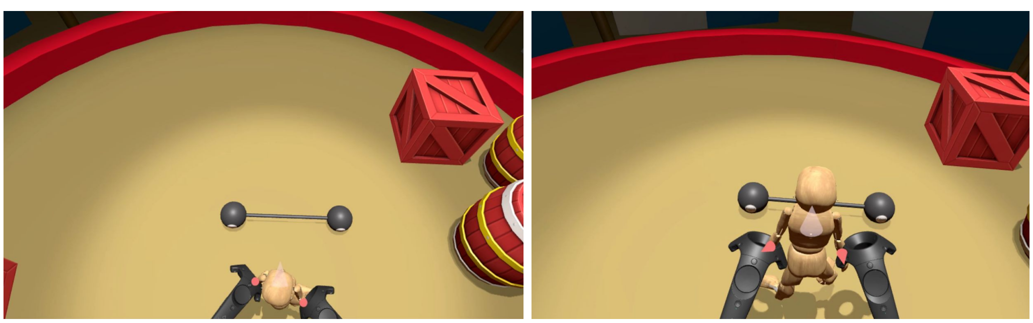


Fig 65 - 66. Strongman before change (Left), Strongman after change (Right)

Fig 66 - 67. Balance Beam Walker before change (Left), After change (Right)

Second, P2's comment on haptics from puppet interaction with the environment made sense. We had comments on how difficult it was to grab things as the strongman, so a change to haptics rules would give additional feedback. In addition, this would further suggest that the puppets are real objects and not the player. We made this change as well (see figure 67 for representation of when the haptics would fire.

Fig 68. Visual of when the haptics fire now i.e. when puppet body collides

FUTURE WORK

Given the lack of similar projects in the game research and design research fields, we see ample opportunity to expand upon this work based on our findings. First, we see our technical solution as a potentially useful way to approach mapping movement interactions to physically simulated objects in VR. Because the solution does not rely on constraints, its far less prone to instability and might offer technical methods to designers seeking to create more nuanced touch interactions in VR.

We also see an interesting way to consider the question of embodiment in VR, especially comparing between research findings of *A Character in Your Hand* and this project. *A Character in Your Hand*'s expert puppeteer participants reported a sense of embodiment with the same puppet models and similar puppet interaction design. This suggests that it is possible to instill that feeling, but the design may need to reflect the expectations of non-experts. One such approach may be to utilize the common practice of puppet building in puppetry and character creation in video games to give users a sense of ownership of their virtual puppets before having them complete certain tasks.

CONCLUSION

We see our work on this project only as a single brick, which may help build a foundational understanding of how to design for VR. It's clear from our review of existing media artifacts that utilizing prior design practices does not work when so much of VR design relies on movement interactions, or interactions based on real-world motions and expectations. This too applies to differing perspectives in VR, and the existing body of work does little to reconcile third person design practices with VR interaction design. This project does, however, bring these practices together and showcases how existing physical interfaces can be valuable design references when dealing with virtual representations that emulated physical form and motor function.

We hope that this project is one that future scholars can review and expand upon when considering alternative perspectives or interfaces in VR, and like early design practices in keyboard interfaces, touch interfaces and gesture interfaces we expect these practices to be supplanted as a greater shared understanding of what does and does not work in VR design is outlined.

BIBLIOGRAPHY

A MAZE. 2017. *[Panel] Fireside Chat with Free-Lives & Punches Bears: Animating Characters with Physics*. https://www.youtube.com/watch?v=DleSVzNMyiQ.

Black, Daniel. 2017. "Why Can I See My Avatar? Embodied Visual Engagement in the Third-Person Video Game." *Games and Culture* 12 (2): 179–99. https://doi.org/10.1177/1555412015589175.

Blumenthal, Eileen. 2005. *Puppetry: A World History*. New York, NY: Abrams.

Boneloaf. 2014. *Gang Beasts*. PC. English. Sheffield, UK: Boneloaf.

Bossa Studios. 2016. *Surgeon Simulator ER*. Oculus Rift. English. London, England: Bossa Studios.

Bowman, Doug A., and Larry F. Hodges. 1997. "An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments." In *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, 35–ff. I3D '97. New York, NY, USA: ACM. https://doi.org/10.1145/253284.253301.

Capcom. 1996. *Resident Evil*. Playstation. English. Osaka, Japan: Capcom.

Carroll, Erin A., and Celine Latulipe. 2009. "The Creativity Support Index." In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, 4009–4014. CHI EA '09. New York, NY, USA: ACM. https://doi.org/10.1145/1520340.1520609.

Cashore, Joseph. n.d. "The Cashore Marionettes." The Cashore Marionettes. Accessed April 18, 2018. http://www.cashoremarionettes.com/.

Chang, Caroline, and Scott Warren. 2011. "CMR Robot Arm." 2011. https://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/s2012/cc722_spw54/Final/CMR_Robot_Arm.htm.

Cherry, Erin, and Celine Latulipe. 2014. "Quantifying the Creativity Support of Digital Tools Through the Creativity Support Index." *ACM Trans. Comput.-Hum. Interact.* 21 (4): 21:1–21:25. https://doi.org/10.1145/2617588.

Crytek. 2016. *The Climb*. Oculus Rift. English. Frankfurt, Germany: Crytek.

Epic Games. 2017. *Robo Recall*. Oculus Rift. English. Cary, North Carolina: Epic Games.

Foddy, Bennett. 2008. *QWOP*. Flash. English. Oxford, England: Foddy.net.

Forward, Richard. 2007. *Bunraku Puppet in Gion*. Photo. https://www.flickr.com/photos/forward/730118975/.

FromSoftware. 2011. *Dark Souls*. Playstation 3. English. Tokyo, Japan: Bandi Namco Entertainment.

Fun Kids. 2012. *The Muppets - Kermit the Frog Interview on Fun Kids*. https://www.youtube.com/watch?v=I0H__I6A-lg.

Gillies, Marco. 2016. "What Is Movement Interaction in Virtual Reality For?" In *Proceedings of the 3rd International Symposium on Movement and Computing*, 31:1–31:4. MOCO '16. New York, NY, USA: ACM. https://doi.org/10.1145/2948910.2948951.

Gunfire Games. 2016. *Chronos*. Oculus Rift. English. Austin, Texas: Gunfire Games.

Hiart. 2014. *Chinese Guangzhou Rod Puppets for Performance about Sun Wukong (the Monkey King)*. Own work. https://commons.wikimedia.org/wiki/File:Chinese_Guangzhou_rod_puppets_for_performance_about_Sun_Wukong.JPG.

Hicks-Jenkins, Clive. 2012. "Project x: A Puppet Primer." *Clive Hicks-Jenkins' Artlog:* (blog). April 19, 2012. https://clivehicksjenkins.wordpress.com/2012/04/19/project-x-a-puppet-primer/.

Hil. 2005. *Bunraku-Style Puppets*. Photo. https://www.flickr.com/photos/spiritsdancing/17204813/.

Itoi, Shigesato, Shigeru Miyamoto, and Katsuya Eguchi. 2010. "Expressing the Inexpressible." Nintendo. 2010. https://www.nintendo.co.uk/Iwata-Asks/Super-Mario-Bros-25th-Anniversary/Vol-1-Itoi-asks-Miyamoto/1-Expressing-the-Inexpressible/1-Expressing-the-Inexpressible-217174.html.

Jacklee. 2014. *A Traditional Yoke Thé (Puppetry) Show at the Nanda Restaurant in Bagan, Myanmar.* Own work. https://commons.wikimedia.org/wiki/File:Traditional_puppet_show,_Bagan,_Myanmar_-_20141209-04.JPG.

Maxis. 2000. *The Sims*. Windows. English. Redwood Shores, CA: Maxis.

Mazalek, Ali, Sanjay Chandrasekharan, Michael Nitsche, Tim Welsh, Paul Clifton, Andrew Quitmeyer, Firaz Peer, Friedrich Kirschner, and Dilip Athreya. 2011. "I'm in the Game: Embodied Puppet Interface Improves Avatar Control." In *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction*, 129–136. TEI '11. New York, NY, USA: ACM. https://doi.org/10.1145/1935701.1935727.

MicroProse. 1991. *Civilization*. MS-DOS. English. Hunt Valley, MD: MicroProse.

Nintendo. 1985. *Super Mario Bros*. Nintendo Entertainment System. English. Kyoto, Japan: Nintendo.

———. 1996. *Super Mario 64*. Nintendo 64. English. Kyoto, Japan: Nintendo.

———. 1998. *The Legend of Zelda: Ocarina of Time*. Nintendo 64. English. Kyoto, Japan: Nintendo.

Nishikawa. 2005. "The Making Of 'Shadow Of The Colossus.'" *Watch Headline* (blog). 2005. http://www.froyok.fr/documents/making_of_sotc.pdf.

Nitsche, Michael, and Pierce McBride. 2018. "A Character in Your Hand: Puppetry to Inform Game Controls." In *DiGRA 2018*. Turin, Italy.

Oculus. 2018. "VR Best Practices." Oculus Developers. 2018. https://developer.oculus.com/design/latest/concepts/book-bp/.

O'Regan, J. K., and A. Noë. 2001. "A Sensorimotor Account of Vision and Visual Consciousness." *The Behavioral and Brain Sciences* 24 (5): 939–73; discussion 973-1031.

Playful. 2016. *Lucky's Tale*. Oculus Rift. English. Mckinney, Texas: Playful.

Polyarc. 2018. *Moss*. Playstation VR. Seattle, Washington: Polyarc.

Rockstar North. 2013. *Grand Theft Auto V*. Playstation 3. English. Edinburgh, UK: Rockstar Games.

Slater, Mel. 2009. "Place Illusion and Plausibility Can Lead to Realistic Behaviour in Immersive Virtual Environments." *Philosophical Transactions of the Royal Society B: Biological Sciences* 364 (1535): 3549–57. https://doi.org/10.1098/rstb.2009.0138.

Stack Exchange. 2012. "Control - What Are Good Strategies for Tuning PID Loops? - Robotics Stack Exchange." Stack Exchange. 2012. https://robotics.stackexchange.com/questions/167/what-are-good-strategies-for-tuning-pid-loops.

Steptoe, William, Anthony Steed, and Mel Slater. 2013. "Human Tails: Ownership and Control of Extended Humanoid Avatars." London, England: University College London. https://pdfs.semanticscholar.org/58ff/13b32073d44a8d83656fba1c91014732f1e6.pdf.

SUPERHOT TEAM. 2016a. *SUPERHOT*. PC. English. Lodz, Poland: SUPERHOT TEAM.

———. 2016b. *SUPERHOT VR*. Oculus Rift. English. Lodz, Poland: SUPERHOT TEAM.

Team ICO. 2008. *Shadow of the Colossus*. Playstation 2. English. Tokyo, Japan: Team ICO.

Usoh, Martin, Kevin Arthur, Mary C. Whitton, Rui Bastos, Anthony Steed, Mel Slater, and Frederick P. Brooks Jr. 1999. "Walking ≫ Walking-in-Place ≫ Flying, in Virtual Environments." In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, 359–364. SIGGRAPH '99. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co. https://doi.org/10.1145/311535.311589.

Won, Andrea, Jeremy Bailenson, Jimmy Lee, and Jaron Lanier. 2014. "Homuncular Flexibility in Virtual Reality." *Journal of Computer-Mediated Communication*, November, 19.

Young Horses. 2014. *Octodad: Dadliest Catch*. PC. Chicago, Illinois: Young Horses.