Virtual Reality Gesture Interface based on Human-animal Interaction

Ziyin Zhang
zzhang391@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia, USA

Abstract

There currently is not a standard unified interface in Virtual Reality (VR), and even though interfaces start to allow gesture-based interaction many user interfaces are directly borrowed from 2D surface interactions. Too few experiences fully leverage the embodiment in VR. To address those issues and to explore a new design space, this project proposes an alternative gesture control scheme based on human-animal interaction. A interactive prototype is presented where the player use gestures to command an animal companion to finish tasks.

CCS Concepts: • **Human-centered computing → Virtual reality**; **Gestural input**.

Keywords: virtual reality, human-animal interaction, gesture recognition, user interface, gesture interface

ACM Reference Format:
Ziyin Zhang. 2018. Virtual Reality Gesture Interface based on Human-animal Interaction. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/1122445.1122456

1 Introduction

Virtual Reality (VR) is an emerging media platform that has became more popular in the past few years. Numerous new hardware platforms and thousands of VR games, VR applications, and 360° videos are being released each year. Compared to traditional media, the attractiveness of VR stems from its vastly experience of immersion, achieved by providing the user combinations of rich stimuli such as stereo images for 3D perception and spatial sound. High-end VR systems also provide rich interactions with the virtual environment through the use of headsets and controllers capable of 6-degree tracking. The headset and controllers tracking motion data was mapped to the user's head and hands in the virtual space. A common interaction scheme is to use the controller as a laser pointer to point at objects or menu items, then press a button on the controller to interact with that object. Many games also try to mimic object manipulations where the user moves a hand to or on the object and then presses a button on the controller to grab or manipulate it. Though this scheme of interaction is effective in many scenarios, it does not address all the interaction needs.

This current interaction scheme relies on the synergy between VR hardware and interaction design. However, there currently is not a standard unified interface in VR. Different platforms have different controller designs and different applications have different button mappings for the same tasks, causing a fragmented experience. To make things worse, some interaction designs are directly borrowed from 2D planar interactions, not leveraging the embodiment of VR. Most current interaction designs also heavily rely on pressing buttons which, if not designed well, is not analogous to the embodied experience so typical for virtual reality. Current interaction schemes rarely address the need to communicate with other agents in the VR.

To address those issues in the current control scheme and to explore a new design space, this project proposes an alternative gesture based control scheme where a sequence of gestures is used to perform core interactions. People use lots of gestures in their non-virtual life and the project is inspired by these usages. For example, people use gestures to communicate from afar or interact with pets. Hard of hearing individuals use gestures to have conversation. However, this is not widely explored in VR interaction design. Traditionally, gesture is not an ideal medium for computer human interaction because of its lack of discoverability, lack of affordance and lack of visual feedback. However, it is not a problem in VR since it can provide real-time visual aid and feedback.

To showcase the potential for the gesture control scheme, this project presents a gesture control framework and designed an interactive game where the player uses gestures adapted from human-animal interaction to command an eagle to hunt prey.

2 Problem

In real life, people interact with the world mostly through direct manipulation with their physical environment. To mirror this paradigm, traditional VR games allow users to directly manipulate the virtual environment with the aid of VR controllers.

A common interaction scheme is to track user's hands that are holding the controllers, map to movement of the user's avatar in virtual space, then use controller button input to execute commands. Most VR games use this interaction scheme to achieve direct manipulation of objects in the virtual environment. Though this control scheme is effective for directly mapping motion to the user's avatar, it does not address all the interaction needs in VR.

Currently, we do not have a well-defined, unified VR user interface. Most experiences use the aforementioned laser pointers to control 3D UI such as menus. However, UI design are usually ported from 2D, without much effort to adapt to the VR medium and different types of controllers. In addition to 3D UI, many experiences have custom designed interfaces to meet their specific interaction needs. Therefore, button mapping for basic controls, teleportation for instance, can be different among experiences, game installments or controller mappings.

VR provides the sense of embodiment by mapping the player's movement to the avatar in virtual space. However, most experiences do not utilize this feature, and the player's embodied virtual body is not being used. Furthermore, the player manipulates the VR environment by pressing buttons on the controllers, which could potentially break immersion.

VR controllers are specially-designed game controllers with sensors that can be tracked in 3D spaces. While some controllers are capable of tracking motions and hand gestures, very few VR games utilize this function. Instead, most VR games use the buttons on the controllers, which have button mappings similar to traditional game controllers. Since VR interaction is particularly powerful when it uses embodiment, holding on to traditional game controllers could potentially break the immersion. Thus, it is necessary to design a user interface for VR games that truly embraces the nature of virtual reality and create better experience for players.

3 Background

3.1 Interaction Design Framework

Shneiderman defined three principles of direct manipulation for designing a controllable user interface. His notion of direct manipulation was to differentiate from the traditional command-line interface at the time. It still provides valuable insights on how to design for a predictable and controllable user interface [13]. A second main reference is Hornecker et al.'s framework for designing interaction for tangible interfaces. Their discussion on haptic direct manipulation is especially helpful [6]. The project was inspired by this research to combine the direct feedback model from Shneiderman with a direct manipulation model, like suggested by Hornecker et al. This led to a focus on gesture based interaction.

3.2 Gesture Interface

Most related work in gesture-based interaction design is conducted on 2D surfaces like computers and tablets. Devices used to track gestures ranges from touchscreens, depth sensors, and cameras with computer vision, etc. However, a few works attempt to apply gesture controls to VR.

Lee et al. provided a set of hand gestures to control avatars in their Virtual Office Environment System (VOES). Most of the gestures match the established mental models, for examples, the "Wave Hand" gesture is waving hand and the "Stop" gesture is a fist. Due to the limitation of technology at the time, however, some of the gestures had to be altered from their base form, e.g. the three fingers in the "Agree" gesture are straight and close to each other, which feels unnatural compared to the basic OK gesture. Nevertheless, they provided a good pipeline for how to process gesture data and how to break down the different data and use different classifiers [8].

3.3 User-defined Gesture Sets

Morris et al. conducted a study on 20 participants to define their own gestures for 27 commands on a large tabletop touchscreen. The participants can use either 1 or 2 hands to interact with the simplified graphics. A total of 1080 gestures were recorded and analyzed. The results shows that users' mental models were influenced by desktop idioms such as Windows OS. For instance, some users double tap on the icon of an object to "open" the object, which is the same as the default "open" command on Windows OS. The research methods presented in the paper are valuable for studies on user-defined gestures [10].

Obaid et al. analysed gestures defined by 17 participants. Their taxonomy for full body gestures used to control a humanoid robot provides new insights on gesture design. This includes gestures to control the non-player "companion," which can be: user-centric, robot-centric, and independent of either perspective [12].

Nacenta et al. compared user-defined gesture sets and pre-designed gesture sets in terms of memorability, learning time, consistency, etc. They found that participants significantly preferred user-defined gesture sets, and they took less time to learn compared to pre-designed ones [11].

To summarize, user-defined gestures are a strong field for dynamic gesture-based interaction. They remain challenging, though, and have not let to unified results.

3.4 Related Works in VR Games

3.4.1 *The Unspoken.* *The Unspoken* is a VR action game in which players duel with other magicians by casting spells using gestures. During the duel, the players can summon various magical artifacts using arm gestures. Then, the players can push the artifact towards the direction of the opponent to initiate the attack, or towards a desired position to place the artifact as mines. The gesture interactions take advantage of embodiment in VR, which makes the game much more compelling.

The gestures are distinct from one another. It takes time and a few tries to perfect the gestures, hence the game have a rather long tutorial compare to most VR games. It includes combat gestures such as:

- **Cross-arms Guard**: Crossing your arms across your chest while holding the grip buttons to charge the magical shield. Then to complete the cast, swing your arms out and down and behold your shield.
- **Y-shape Volley**: Stretched out your arms in a Y shape and hold the grip buttons to charge an attack.
- **Push Attack**: Grip the controller grip buttons about shoulder width apart in front of your stomach and hold to charge. This will generate a magical artifact. Then, push the artifact towards the direction of the opponent.
- **Conjure Motion**: Grip the controller grip buttons about a shoulder width apart with your arms held down and your palms out to charge.

3.4.2 *Falcon Age.* *Falcon Age* provides great examples of communicating and interacting with an animal companion As you play and direct your virtual falcon, the interactions involves various gestures:

- **Whistle to call it over**: put the controller near the mouth(right below the headset) and press a button. It will trigger a whistle sound effect and the falcon will land on the player's hand shortly.
- **Combat**: release the bird, then use laser to point at the target (enemies or prey) to instruct the bird to fly over and attack the enemy

Other types of interactions includes:

- **Heal**: pet the falcon to heal it
- **Feed**: put a piece of food near the falcon's head
- **Play**: Play with the falcon to improve relationship, includes petting its head, doing a fist-bump, give it a toy to play
- **Dress**: put a hat or other accessories onto the falcon to change its abilities. Some accessories provide more protection, others are decorations.

Although this game presented multiple ways to interact with the falcon, only a few are incorporated into the actual gameplay. Existing titles clearly attempt to benefit from a gesture-based approach but fall short to fully implement it.

4 Approach

In order to create an effective VR interface with embodied gesture-based interactions, I decided to design a gesture interface based on human-animal interactions. Human-animal relationships are wide-spread and - as we do not share a single language - they often include gestures as part of our communication. Because these gesture references are already in place, it is more natural for the players to do these gestures and easier to memorize them. In this sense, the gestures are more effective and ready to be adapted to VR.

4.1 Gesture Interface

VR is uniquely suitable for exploring gesture controls because it already has tracking for hand gestures and the virtual environment allows endless possibilities of visual cues and feedback for gesture interface.

Gesture controls, on the other hand, are rarely used in current commercial devices, despite efforts from big companies to push them into commercial products. Such efforts include the Soli gesture control in Google Pixel 4 smartphone and gesture control on Microsoft HoloLens. Though popular culture depicts gesture controls in 3D space as cool and effective like the movie Minority Report [9], it never took off and became popular due to the lack of affordance, discoverability, and high learning curve. Nevertheless, gesture control still has its place in real life, especially in communicating with other agents e.g. American sign language, communication with pets, Special Ops, etc. Inspired by these examples, this paper explores gesture controls in VR.

4.2 Human-animal Interaction

Humans have always been interacting with animals, from animal husbandry to keeping them as animal companions.

Through reviewing papers on dog training [4] and analyzing video documentary such as *The Eagle Huntress* [1], lots of overlaps in tool usage and training were identified. Both dog training and eagle hunting use a combination of voice and body gesture. The process of raising and bonding with the animal is essential to effective communication with the animal. The design phase explore the development of basic interactions based on human-animal gestures.

5 Design

The design of the gesture interface went through several iterations before settling down on human-animal interactions the final prototype version.

5.1 Gestures inspired by Eagle Hunters

The first design of the gesture interface was inspired by eagle hunters. Through analyzing documentaries on Mongolian eagle hunters, the body languages used by eagle hunters to communicate with the eagles were identified.

The sketches show the gestures design in Figure1 and 2. The main gestures to interact with the bird companion are described below:

- **Direct**: direct the bird companion to interact with a target. Similar to the way eagle hunters kept their eagles on their arms, the bird companion starts off on the player's arm. Once the player raises an arm, the bird will fly upwards, awaiting further directions. The player then uses the point gesture to **direct** the bird to fly over and interact, or in this case, attack the target.
- **Return**: instruct the bird companion to return after (or during) the interaction. Once the bird has finished interacting with the target, the player can instruct it to **return** by swinging the arm in a backwards motion. Once the bird starts to fly back, the player can raise the arm for the bird to stand on.

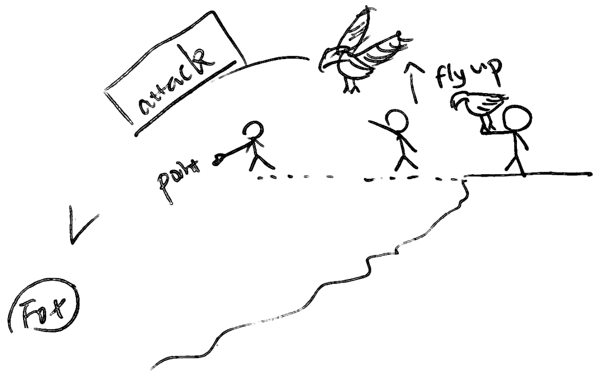

Figure 1. Gesture for **Direct**

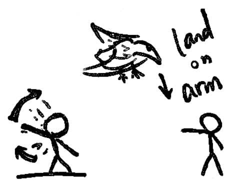

Figure 2. Gesture for **Return**

Another human-animal condition is that between humans and dogs. In fact, the final design was inspired by how I interact with my dog at home (see Figure 3). Here, the user-defined gestures emerge in a kind of shared condition between the dog and the owner. In my case: the rotation of my hand leads to the dog to roll over on the floor.

5.2 Sign language Inspired Gesture Design

Through learning basic American sign languages, I was especially drawn to the concept of Spatial Grammar. As sign

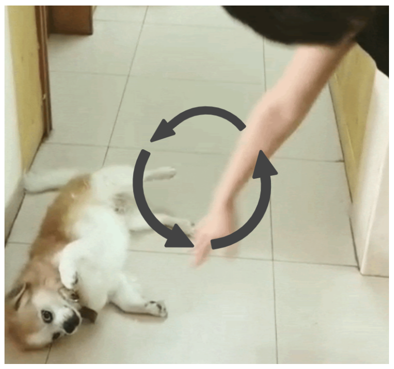

Figure 3. Interaction with my dog

languages are not used in a linear sense, the subject and object of a verb cannot be derived from word order, hence the usage of spatial grammar. Usually, before signers describe an event, they establish the spatial relationship of the entities by "putting" each entity on an imaginary plane in front of them.

Inspired by this concept, I designed two interaction scheme that make use of spatial grammar to define entities. Once the entities are defined, the players can perform gestures that describe the interaction between the entities, and informs the direction of the action if necessary.

Two different interaction schemes based on the function of dominant vs. non-dominant hand:

- **Different Functionality**: While the dominant hand is pointing at a target, the non-dominant hand does the gesture to perform actions on the target. After the gesture is performed, the dominant hand then points at another target to indicate the object of the action.
- **Same Functionality**: The left and right hand each point at the target objects to interact (the player can be one of the objects). After selecting the objects, both hands perform the gesture to show the intended action between the two objects.

5.3 Gestures based on Human-animal Interaction

The design targets for the gesture interface are defined in Table 1.

Table 1. Gesture Design Criteria

Interaction	Gesture	Pet Gesture
SELECT	Point	Call name to get its attention
ACTIVATE	Circle	Draw circles to make it roll over
ACTION	Throw	Throw sticks for it to fetch
CANCEL	Stop	Call name while clapping

These criteria were informed by basic interaction design requirements as outlined by Shneiderman [13] and mapped established human-animal gestures onto them.

6 Implementation

The gesture recognition system is implemented using an HTC Vive headset, two Valve Index VR controllers, SteamVR Unity Plugin v2.5 and Unity's built-in physics engine. Unity made the implementation of the VR visualization itself relatively simple but the project had to solve a number of challenges to implement particularly the gesture recognition.

6.1 First Stage: Arm Gesture Detection

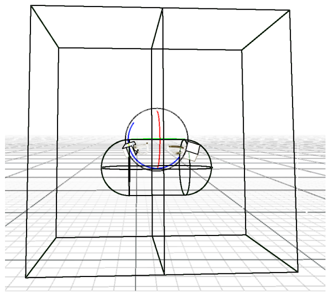

Figure 4. Gestures is detected using relative position of the hands and body

The first prototype is a simple scene where the relative position of hands with other parts of the body is used to detect arm gestures (Figure 4).

The relative position of hands (represented by the two small cubes in the middle) compared to the position of the head (represented by the sphere) and chest area (represented by the capsule) is used to detect simple gestures. The two outer boxes mark the areas for the left and right arm span, used to detect which side of the body. This stage served as an exploration of the technical design space.

6.2 Second Stage: Gesture Recognition based on TraceMatch

Basic gestures like SELECT and ACTIVATE were implemented in this stage. The detection of hand gestures involved using the "finger curl" data from the Valve Index Controller, which can be retrieved via SteamVR Unity Plugin. The curl value for each finger ranges from 0 to 1. However, due to the differences in controller buttons, the threshold for a curled finger is different for each finger. Therefore, a curl threshold and a straight threshold have to be defined for each finger.

Shown in figure 5, SELECT is detected when the player's hand is doing a pointing gesture. pointing with the index finger. It is defined as: the index finger is within the straight threshold, and the other four fingers are within the curl threshold. While the user is doing the SELECT gesture, the system uses Raycast from the Unity physics engine to detect where the user is pointing. If the Raycast hits an object that can be interacted by the user, it will start glowing blue to indicate that it is interactable. This responds again to Shneiderman's principle of direct feedback [13].

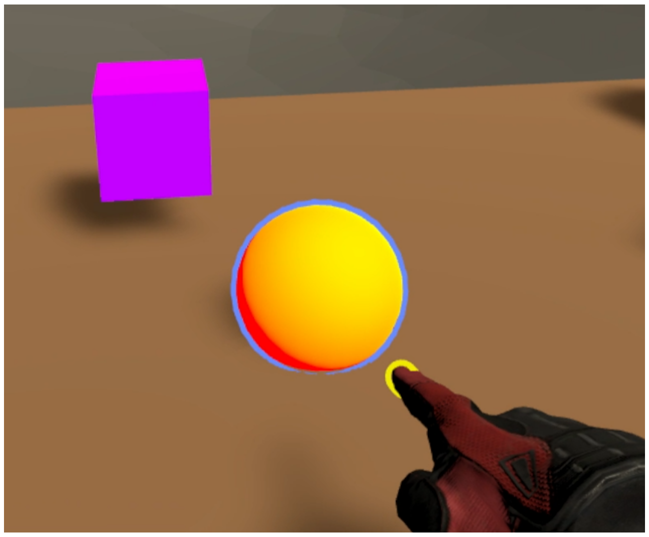

Figure 5. SELECT Gesture: a blue outline will appear on objects being selected

For the ACTIVATE gesture, the system needs to detect when the player's hand is drawing circles. The algorithm is inspired by the motion matching process described in TraceMatch [3]. To detect if the user's hand is in a circular motion, three points were randomly chosen from the hand's trajectory to calculate the center of the circle and the rest of the points were tested to see if they fit that circle.

In a sample rate of 33.3 Hz (every 30 ms), the position of the hand is saved in a circular buffer of size 30. The center of a circle is calculated using 3 data points, including the latest data point, with intervals of 10 samples. When 25 out of the 30 data points are within the radius threshold, the hand is considered moving in a circular motion. The earliest data point is marked as the start of the circle, which is used to calculate the angle of the curve. Once the angle between the start of the circle and the latest data point has accumulated over 360°, this means the hand has finished drawing a full circle, and thus the ACTIVATE gesture is fired.

Rapid incremental operations were implemented for the ACTIVATE gesture, as it is mentioned in Sheniderman's three principles of direct manipulation [13]. ACTIVATE is fired multiple times if multiple rotations of circle are detected, e.g. each time when the cumulative angle goes above 360°, 720°, 1080°, etc.

6.3 Final Prototype

As the design emerged and the technical gesture recognition fell into place, the VR world was developed and an animal companion was added into the project. This stage consisted of a gradual build up of a 3D world that would showcase the gesture-based interaction in a VR condition. It also required gradual adjustment of the gestures to the actual VR condition as it became functional.

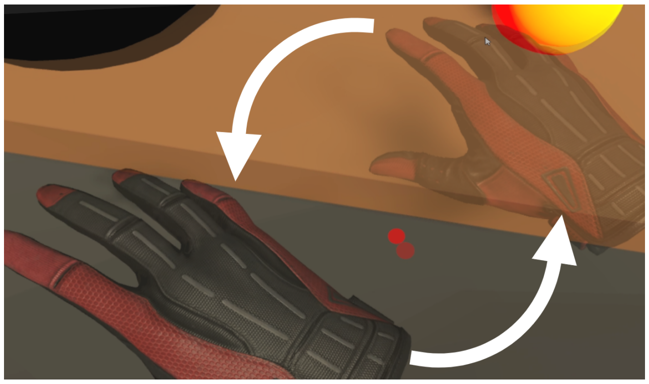

Figure 6. ACTIVATE Gesture: the hand is doing circular motion as shown by the white arrows; the center of the detected circle is shown by the red dot

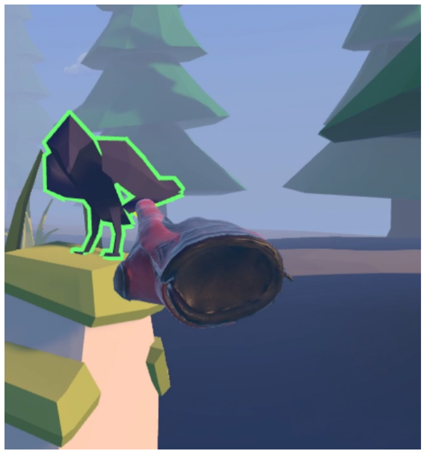

Figure 7. SELECT Gesture

6.3.1 SELECT gesture.

The SELECT gesture remained the same as the 2nd iteration. However, the color of the outline is changed to bright green for better visibility as shown in Figure 7.

6.3.2 ACTIVATE gesture.

To better indicate that the gesture ACTIVATE is triggered and the companion is activated, the animation of the companion changes to a different state. In this case, the bird companion changed from a standing pose (Figure 8) to a flying pose (Figure 9), showing that it is ready to do the next action.

6.3.3 ACTION gesture.

The ACTION is a throwing gesture. The first version of the gesture is implemented using TraceMatch algorithm. Once the hand is detected as moving in a circular motion, the angle of the curve is calculated. Unlike ACTIVATE which requires a full circle, ACTION is detected when the angle of the curve has accumulated over 60°.

However, through extensive testing, this version of the ACTION gesture detection turned out to not work very well. The throwing motion did not match a circle most of the time,

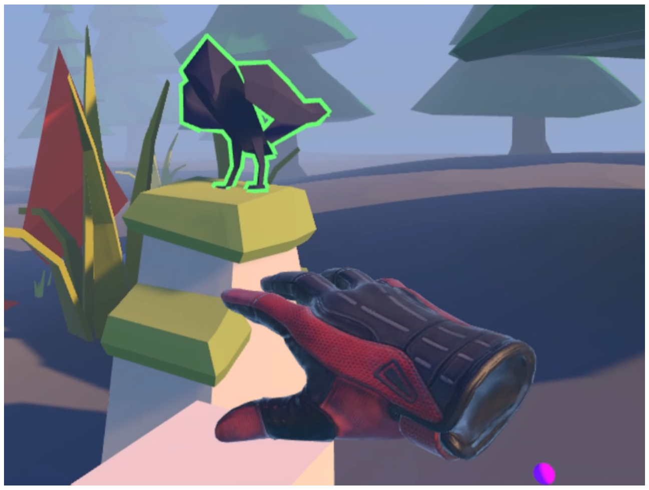

Figure 8. Start of **ACTIVATE** Gesture: the player starts the activation process by drawing circles (center of the circle is shown as the purple dot)

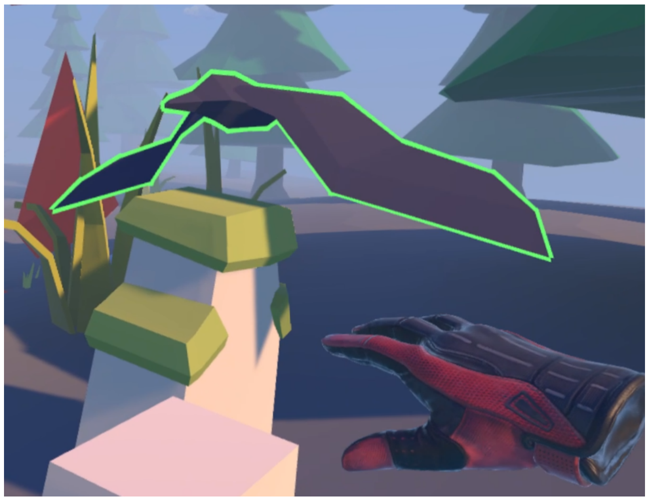

Figure 9. ACTIVATE Gesture: the companion starts flying to indicate that it is activated

as it might be a short curve in the shape of an ellipse or in a straight line. Therefore, in the final version the detection algorithm is changed as follows: the start of the throwing gesture is detected when the palm of the hand is closed (to mimic the action of throwing items held in the hand), at which point the hand position will be recorded; the end of the gesture is detected when the palm is opened again (indicating that the hand has released the items thrown), at which point the hand position will be again recorded. The direction of the ACTION gesture is then calculated using the vector bounded by the two positions.

6.3.4 CANCEL gesture.

The CANCEL is a "stop" gesture: both palms open with the tip of the hands pointing upwards.

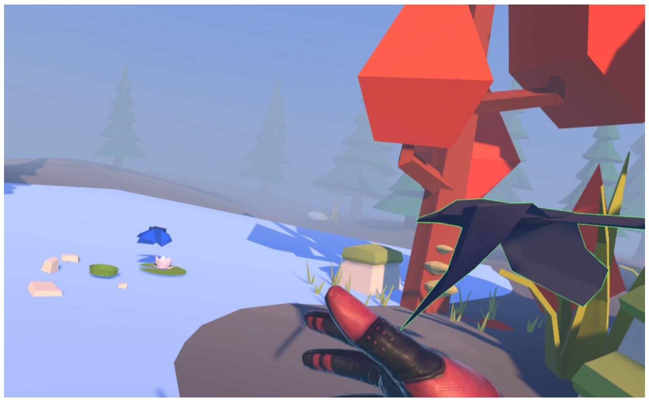

Figure 10. ACTION Gesture: the player do a throwing motion towards the blue target

This is implemented by calculating the rotation of both hands to see if they are both within the threshold of the upward angle.

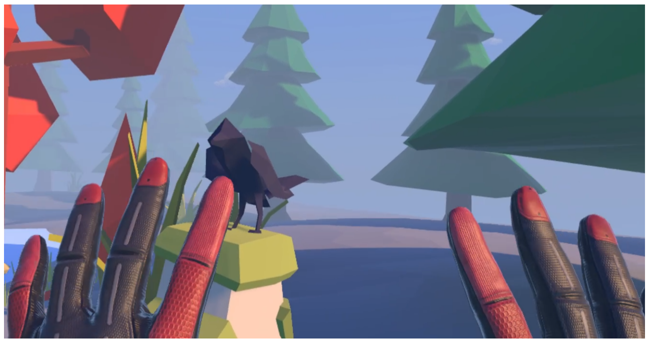

Figure 11. CANCEL Gesture: after the companion is activated, the player can use a "stop" gesture to cancel the activation

6.3.5 Gesture Recognition Pipeline. The gesture recognition algorithm was implemented using a finite state machine. Transitions between states are triggered by various hand gestures shown in Figure 12.

The gesture state machine contains 4 main states: Selection, Activation, and Action, and Cancellation, each handles the corresponding gesture `SELECT`, `ACTIVATE`, `ACTION` and `CANCEL`. 3 sub-states handle in-between situations: Idle, Finish Activation, and Buffer.

The gesture recognition system is computationally fast and it works in real-time. The system is also versatile, as it can be adapted to use any kinds of controllers capable of 6-degrees tracking and detecting simple hand gestures like pointing or opening palm. While the main goal of this project was to implement a particular proof-of-concept prototype for a sample human-animal VR condition, the openness of this architecture could be used for other interaction design conditions as well.

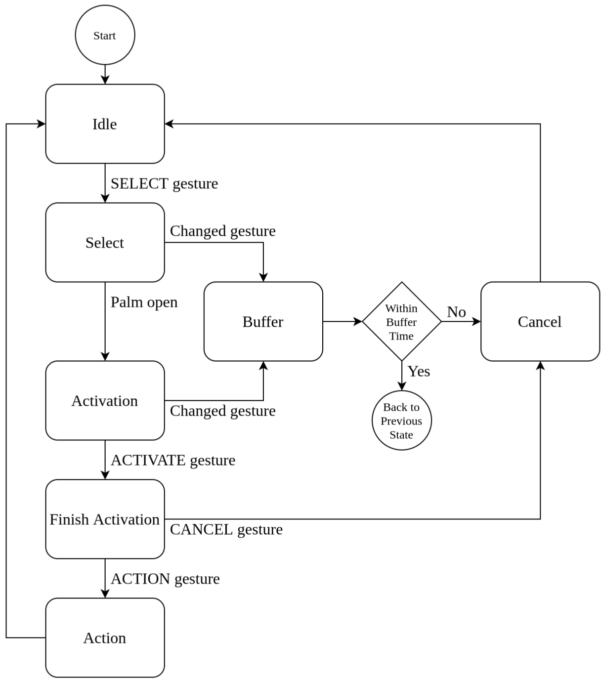

Figure 12. Gesture State Machine.

6.4 Technology

The prototype was developed in Unity Engine v2019.3.9f1 using SteamVR Unity Plugin v2.5. HTC Vive Pro was chosen as the VR headset because of the stability of tracking and room-scale play area. Valve Index Controllers were chosen as the VR input device because it is a commercial device capable of hand and finger tracking. In addition, the strap design on the controllers allows the player's hand to perform gestures which involve the open palm, such as `ACTIVATE` and `CANCEL`, in a more natural way.

7 Evaluation

No formal evaluation was possible, due to the Covid-19 pandemic. But the targeted design criteria would focus on compliance with Shneiderman's principles of designing comprehensible, predictable and controllable user interfaces [13]. The principles the following:

1. Continuous representation of the objects and actions of interest
2. Physical actions or presses of labeled buttons instead of complex syntax
3. Rapid incremental reversible operations whose effect on the object of interest is immediately visible

User studies would need to evaluate the gesture interface presented in the prototype along those principles. The following tasks could be designed for players to complete, during which think-aloud data and video recordings of the physical and the digital performance (via screen capture) could be collected for qualitative studies:

- Use gestures to command the companion to fetch an item (no time constraint)
- As part of the gameplay, a fox has captured the player's valuables. Before the fox run a way, the player needs to use gestures to direct the companion to attach the fox, then fetch the valuables back to the player (with time constraint of 1 minute)

For quantitative study, collect and analyze the result of the following questionnaires for each user: NASA TLX (Task Load Index) [5], CSI (Creativity Support Index) [2], and UES (User Engagement Scale) for VR game with gesture hand interface [7].

8 Conclusion

The current prototype focused on interaction with a single animal companion. While the final prototype allows user to select different birds to activate one and direct its action, it would be interesting to test with more types of interactions, interacting with non-animal or different animal companions, using menu interfaces, etc. Could we rely on a comparable gesture-based interface to direct e.g. virtual fish or horses?

The underlying gesture recognition algorithm can be easily adapted to other types of sensor data. With small modifications, it should enable simpler mobile VR systems that do not have full spatial tracking to have a richer gesture based interactions.

For future work, it seems most valuable to test the prototype with more types of interactions. More specifically, this could include:

- interacting with multiple companions
- interacting with non-animal targets
- manipulating abstract targets (e.g. menu interfaces)

Due to the pandemic, it is hard to do user study. Would be nice to collect data from more subjects.

9 Summary

Most VR games do not take advantage of embodiment in VR or use gesture interfaces to provide embodied interactions. Traditionally, gesture control is not a viable way of control because of the lack of affordance, memorability, and discoverability. This paper explored an alternative gesture interface based on human-animal interaction. The VR platform has the advantage of displaying gesture prompts with real time visual feedback that makes it uniquely suitable for using such gesture control. This paper demonstrated a kind of gesture interface in this proof-of-concept prototype.

This paper aimed to provide insights on the user experience design of VR and AR gesture interfaces, as well as to inspire future VR experiences to use more embodied interactions like gesture controls

References

[1] Otto Bell, Sharon Chang, and Stacey Reiss. 2016. The Eagle Huntress. Sony Pictures Classics. Director-Bell, Otto.

[2] Erin Cherry and Celine Latulipe. 2014. Quantifying the creativity support of digital tools through the creativity support index. *ACM Transactions on Computer-Human Interaction (TOCHI)* 21, 4 (2014), 1–25.

[3] Christopher Clarke, Alessio Bellino, Augusto Esteves, Eduardo Velloso, and Hans Gellersen. 2016. TraceMatch: a computer vision technique for user input by tracing of animated controls. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 298–303.

[4] Jessica B Greenebaum. 2010. Training dogs and training humans: Symbolic interaction and dog training. *Anthrozoös* 23, 2 (2010), 129–141.

[5] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, 139–183.

[6] Eva Hornecker and Jacob Buur. 2006. Getting a grip on tangible interaction: a framework on physical space and social interaction. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. 437–446.

[7] Irma C Landa-Avila and Maria-Luisa Cruz. 2017. Engagement in a virtual reality game with gesture hand interface. An empirical evaluation of User Engagement Scale (UES). In *International Conference of Design, User Experience, and Usability*. Springer, 414–427.

[8] ChanSu Lee, SangWon Ghyme, ChanJong Park, and KwangYun Wohn. 1998. The control of avatar motion using hand gesture. In *Proceedings of the ACM symposium on Virtual reality software and technology*. 59–65.

[9] Gerald R Molen, Bonnie Curtis, Walter F Parkes, and Jan de Bont. 2002. Minority Report. Twentieth Century Fox. Director-Spielberg, Steven.

[10] Meredith J Morris, Jacob O Wobbrock, and Andrew David Wilson. 2010. User-defined gesture set for surface computing. US Patent App. 12/185,166.

[11] Miguel A Nacenta, Yemliha Kamber, Yizhou Qiang, and Per Ola Kristensson. 2013. Memorability of pre-designed and user-defined gesture sets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing systems*. 1099–1108.

[12] Mohammad Obaid, Markus Häring, Felix Kistler, René Bühling, and Elisabeth André. 2012. User-defined body gestures for navigational control of a humanoid robot. In *International Conference on Social Robotics*. Springer, 367–377.

[13] Ben Shneiderman. 1997. Direct manipulation for comprehensible, predictable and controllable user interfaces. In *Proceedings of the 2nd international conference on Intelligent user interfaces*. 33–39.