EPiC Series in Computing
Volume 58, 2019, Pages 271–281
Proceedings of 34th International Conference on Computers and Their Applications

An Alternative Natural Action Interface for Virtual Reality

Jay Woo[1], Andrew Frost[1], Tyler Goffinet[1],
Vinh Le[1], Connor Scully-Allison[1], Chase Carthen[1], and Sergiu Dascalu

Department of Computer Science and Engineering
University of Nevada, Reno
Reno NV, 89557
jjwoo95,vdacle,chasec2331@gmail.com
cscully-allison@nevada.unr.edu
dascalus@cse.unr.edu

Abstract

The development of affordable virtual reality (VR) hardware represents a keystone of progress in modern software development and human-computer interaction. Despite the ready availability of robust hardware tools, there is presently a lack of video games or software in VR that demonstrates the gamut of unique and novel interfaces a virtual environment can provide. In this paper, we present a virtual reality video game which introduces unique user interface elements that can only be achieved in a 3D virtual environment. The video game, titled Wolf Hunt, provides users with a menu system that innovates on traditional interfaces with a virtual representation of a common item people interact with daily: a mobile phone. Wolf Hunt throws users into a procedurally generated world where they take the role of an individual escaping a wolf assailant. Deviating from traditional locomotion options in VR interfaces, such as teleportation, Wolf Hunt measures the displacement of hand-held VR controllers with the VR headset to simulate the natural action of running. Wolf Hunt provides an alternate interfacing solution for VR systems without having to conform to common 2D interface design schemes.

1 Introduction

Virtual reality is on the bleeding edge of technology and is actively transforming fields like medicine, environmental science, and entertainment. New advancements in VR hardware, like the commercial releases of the HTC Vive and the Oculus Rift, have driven an explosion of new and interesting development in virtual reality software [11, 12]. However, modern VR development has a number of challenges that inhibit effective interface design.

Virtual reality software, despite providing an immersive 3D environment to interact in, often draws inspiration for user interfaces from designs made popular in application development. This usually takes the form of 2D menus existing in 3D space, as well as 3D controllers serving as a digital mouse or pointer. Additionally, the actions made by the locomotive action of teleporting and displacing locations instantaneously often runs into the issue of causing motion sickness, or "VR Sickness." This phenomenon is often attributed to users encountering a situation or perceiving something which contradicts other sensory information [7].

The game introduced in this paper, Wolf Hunt, has the users take up the role of either a wayward traveler escaping a wolf assailant through a dense and unrecognizable forest or the role of the wolf assailant hunting after the traveler. Wolf Hunt employs a design philosophy which attempts to make navigation of menus and interaction with the virtual world as naturalistic as possible. This approach provided a more immersive user interface for VR by centering the interaction with the system around 3D movements made with the headset and both controllers in each hand. Additionally, a key goal was to address motion sickness concerns by providing a natural-feeling alternative to common teleportation as locomotion.

To engender a sense of replayability and increased immersion, the world and terrain in Wolf Hunt is procedurally generated to give the impression that the player is truly lost. To enable these features, Wolf Hunt was developed with the Unity game engine and the Virtual Reality Toolkit (VRTK), as well as the HTC Vive, it being the VR system of choice [3, 10]. In using these technologies we were able to exploit the synergy between Unity and the HTC Vive to quickly develop a robust and polished game.

The remainder of this paper is structured as follows: Section 2 introduces background information about the topics of locomotion, realistic interactions, and procedural generation, Section 3 goes into the specifications of the system, Section 4 discusses the overall design of the Wolf Hunt game, Section 5 contains a discussion about the core features of the Wolf Hunt game, and Section 6 wraps up the paper with the conclusion and future work.

2 Background

Oculus Rift, HTC Vive, and Playstation VR developers have been creating and introducing their own games which use various forms of movement within a VR world. Due to the ever decreasing costs of VR technology, more content developers are introducing and testing various, novel VR movement techniques. Since modern VR systems are still new and popular, significant research is being done on how to best move inside a VR world while being physically limited by the amount of space in the real world. This area of study is known as locomotion. There are many different forms of locomotion being tested, each with varying results of immersion and motion sickness. In the paper "HCI Lessons From PlayStation VR" [5], multiple locomotion techniques from different Playstation VR games are enumerated and juxtaposed with the rating score of the game. An example of a popular and common locomotion technique is teleportation, which is shown in Figure 1. With this technique a user depresses a button, points to their intended destination, and appears there after releasing the button. Another method, illustrating the plurality of novel approaches to locomotion and covered in the above research, involves moving forward after the user closes their eyes for a set period of time.

For many games, a form of variability is introduced to increase overall replayability of the game. This could come in many forms such as hidden secrets revealed within a game, extra levels which are unlocked after completing a game's main objective, an added level of difficulty to increase the challenge a game offers, and more. A technique which has become popular lately can dramatically increases variability within a game: procedural generation. Procedural generation makes it so that user's experiences between play sessions will never be exactly the same. Many aspects of a game can be handled this way, different map layouts, different types of enemies, or even different types of objectives that the user must accomplish. Accordingly, procedural generation in a VR environment has been a prominent topic in modern VR research for many years [9, 6]. In fact, the paper "A Survey on the Procedural Generation of Virtual Worlds" [4] discusses the feasibility of different procedural generation algorithms on increasing the realistic look of objects. There exists many different procedural generation algorithms, each

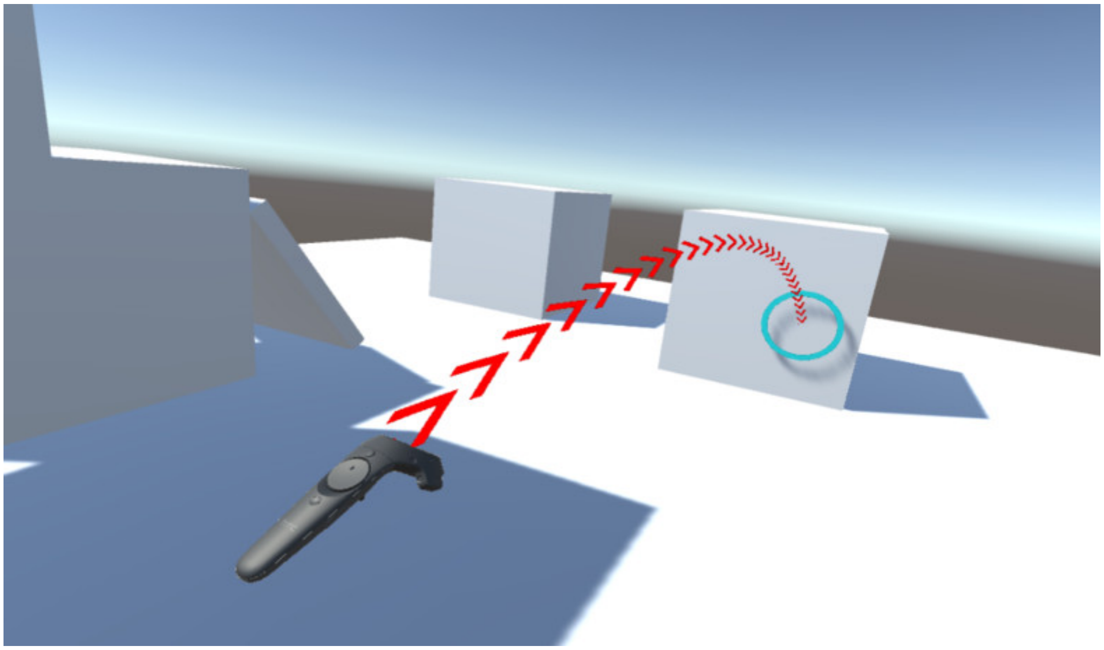

Figure 1: An example of a form of locomotion in VR. The player points to the location they want to teleport towards.

with its own advantages and disadvantages.

One of the biggest advantages of VR technology is that it enables physical movement to be a form of input. Using these input methods, a user is capable of more natural interactions when playing a game, especially when compared to more traditional methods, such as using a joystick or a controller. Having more realistic interactions provides an increased feeling of immersion and gives a deeper feeling of presence in the virtual world. Significant research is being done to find novel intuitive ways in which users can interact with a VR systems [1, 2]. These forms of physical movement also include inputs from other parts of the body, instead of just hand-held controllers. In one such case, a paper exploring natural eye-gaze-based interaction for immersive virtual reality [8] studied the effects of natural eye movement tracking technology in a 3D virtual environment.

3 Software Specification

To better provide a blueprint for reproducability of the work outlined in this paper, this section presents an outline of the specifications that were elicited from project stakeholders and potential users. The functional and non-functional requirements derived from those interviews are outlined in the next two sections. Functional requirements define the specific inputs, outputs, and behaviors of components within a system. Alternately, non-functional requirements define specific criteria that serve as constraints on the system.

3.1 Functional Requirements

Functional requirements for Wolf Hunt dictated how a user would interact with the game and are broken up into priority levels, with high priority requirements being critical for system operation. The first high-priority functional requirement of interest to this paper specified that a user should be capable of navigating their virtual character by using motion controllers. This requirement ensured for an extra level of immersion within the game compared to more traditional character movement methods such as using a joystick or controller. The next functional requirement dictated that menu options should be displayed on a object in the game world. This provides a user with an intuitive form of game mode navigation. A third functional requirement specified that a user must be able to choose between two different game modes. A user could either play as a traveler or play as a wolf, with each mode having a different objective.

Following from these, a fourth high priority requirement was that users shall have the option of choosing between a daytime and nighttime map. This adds an element of variability to each play session. The fifth high-priority requirement was, when playing as the traveler, a user would be capable of throwing a form of bait. This enabled users to indirectly defend themselves. Alternatively, when playing as a wolf, a user should have a scent tracking ability in order to help locate the traveler. Another high priority functional requirement is victory and loss conditions during a play session. This gives the user a goal to accomplish and a sense of urgency to complete the goal. The last high priority functional requirement is that the map is composed of a procedurally generated forested terrain. This helps to reinforce a user's immersion within the game by providing a new environment each time and to break up the monotony of multiple sessions.

One key lower priority functional requirement implemented, required the generation of a map during run-time. Another low priority functional requirement necessitated a dynamic weather system. This allows for a more realistic depiction of the real world and helps to keep a user's interest during a play session. The last functional requirement was to have a variety of terrain type which the user could choose from. Different terrain types enhances the variability of the game.

3.2 Non-functional Requirements

The first critical non-functional requirement indicated the need to handle input via a VR headset and motion controllers. This was necessary since one of the goals of the game was that it used a VR system. Another requirement was that the game must be capable of running on Microsoft Windows 10. The third requirement was that Unity 3D be used to implement the game due to Unity 3D's robustness and compatibility with the HTC Vive [11]. As a fifth requirement, the C# programming language is to be used to implement the game, as it is the main codebase for development in Unity. Next, the sixth requirement was the use of the VR library known as VRTK to integrate software and hardware components for Wolf Hunt. The seventh non-functional requirement was that the game would run on an Intel Core i5-7600 equivalent or better CPU, a GTX 1060 equivalent or better GPU, and at least 4GB of RAM. This was to detail what hardware was needed in order to be capable of running the game at a moderate speed. The last non-functional requirement was that the procedural map generation be multi-threaded, in order to dramatically decrease loading times during a play session.

4 Software Design

Wolf Hunt is comprised of several software components that bring a unique touch to user interaction within the game. One of the most significant ideas was to eliminate user interaction with a 2D menu and this was achieved rerouting all menu options through a mobile device generated in the VR world. The phone model used inside the game which when selected, displays various options to the player. These options are: selecting the mode in which the player wants to play as, general settings, game options, and an option to exit the game. This interface takes full advantage of 3D movement as a form of input by providing the users with a game menu system that maps to common real-life actions, such as navigating a mobile device. Compared with other menu interaction methods, such as using a 2D menu, Wolf Hunt's menu interaction method brings an additional level of immersion to the VR experience.

Another important software component within Wolf Hunt is the procedural generation of the map in which the player navigates. The choice to implement the procedural generation was made so that the user's familiarity with the map would lessen and to invoke a sense of way-wardness that follows the theme of Wolf Hunt. An algorithm was conceived and implemented to provide this procedural generation. First a grid is formed to construct a map using the procedural generation process. The grid is then filled in using preset tiles consisting of different biomes. The biomes are placed randomly onto the grid and tiles placed next to each other are stitched together. The procedural generation process is implemented in such a way that the map mimics a realistic environment. For example, a pond will not be placed on top of a tree.

Figure 2: A wolf controlled by AI is chasing down a pack of deer. The deer are also running away from the wolf.

Figure 3: The utility belt which the player has when playing as the traveler. From left to right the belt contains bait, a flashlight, and a mobile phone.

The edge of the map is composed of mountainous terrain so that the player cannot see the edge of the map while establishing a boundary and the middle of the map is composed of a variety of tiles to create a unique landscape.

In conjunction with the procedural map generation are animals which inhabit different areas of the map. Artificial intelligence (AI) determines the behavior of the animals inhabiting the map. All animal behavior in Wolf Hunt was designed to mimic the skittish behavior of animals present in a forest. When the user gets near one of the animals, the animal will attempt to run away from the user. Another key feature in the AI portion is that occasionally, the wolf, also controlled by AI, will divert attention momentarily and chase animals in an attempt to eat it, which is shown in Figure 2. This is done through a series of back end calculations that use a random number generator to effectively randomize animal behavior. The random number generator generates numbers using an embedded world seed which loops through a series of preset values.

In addition, the AI is implemented in such a way that animals will never traverse over areas of the map which are not navigable. For example, if a deer is running away from the user the deer will never run up a mountain. This was done so as to make the game have more realistic properties since a mountain is normally too steep to climb. When the AI is controlling a wolf, the AI will find a navigable path which can be traversed to reach the traveler. Obstacles in which the AI avoids traversing through are mountains, lakes, trees, and the log cabin.

The choice of having animals inhabit the map was made so that it provides a sense of realism to the user that they are moving through a forested terrain. The AI also determines the behavior of the user's adversary, which is the wolf or the traveler depending on the character the user chooses. The adversary is set to track and chase the user. This allows the user to feel urgency in completing their objective as quickly as possible.

Figure 4: A user can activate the track ability when playing as the wolf. The track ability appears as a red mist which leads the wolf to the traveler.

Another significant software component present inside Wolf Hunt is one specific to a user playing as the traveler. While a player is playing the game, a belt will appear on the player's hip carrying various useful tools that the player can use. The tools include a pouch which holds bait, a flashlight which the player can turn on and off, and a phone which acts as an interactive way for the player to navigate through menus. The bait from the pouch has the ability to momentarily distract any incoming wolf, the phone serves as the main menu interface, while the flashlight is aid the player in situations that hamper visibility such as dense sections of the forest. All items on the utility belt can be seen in Figure 3. This method allowed users to access their items intuitively instead of having them activate a menu, browse for an item, and then generating the item in their hands.

The next software component described is specific to when the user is playing as the wolf assailant. The user can generate tracking functionality by pressing down both trigger buttons on each controller and gaze up to create a temporary scent trail which will lead them to the traveler. This ability has unlimited uses but can only be invoked once every few seconds in order to provide a sense of challenge for the user. This scent trail generation feature takes the form of a red mist-like cloud that constantly elongates to the direction of the traveler AI. An instance of the the wolf's scent tracking ability can be seen in Figure 4.

Finally, the last major software component present in Wolf Hunt is the style of locomotion used by the player in order to navigate the game world. Contrary to the most common VR locomotion style, teleportation, Wolf Hunt employs a system that involves the 3D movement of the controller in relationship with the physical location of the headset in the 3D play area. While in the play area, the physical location of the headset on the player is taken note, as well the placement of the VR controllers in that play area. In order to move the character, traveler or wolf, the user must swing the VR Remotes to and from the VR Headset in fashion that mimics

Figure 5: Shows the pause menu for Wolf Hunt. The user pauses Wolf Hunt by interacting with the mobile phone object.

the swinging of one's hands during the act of running. The system then calculates the distance changes between the remote and the headset in order to register a movement command, while the rate of swinging determines the distance and speed in which the traveler or wolf traverses the game world. The distance and speed rate differs for both traveler and wolf since the wolf will innately be faster than the traveler. So accordingly, each swing of the VR controller will be more meaningful for the wolf than it would be for the traveler. This form of locomotion was added into Wolf Hunt because it mimics the form of movement that humans traverse and so that it would lessen the effects of motion sickness while inside a 3D game world.

5 Discussion

There are various features that are the core components of Wolf Hunt. These core components are a new and intuitive form of locomotion which was introduced in Wolf Hunt, the procedurally generated terrain, and the new user interface within the VR world. Each of the core components greatly contributed and helped to benefit the entertainment value and the user interaction of Wolf Hunt.

Locomotion within Wolf Hunt involved the user swinging each arm up and down. By doing this form of movement a user would move their in game character towards the direction the motion controllers were facing. This form of physical movement closely resembles the physical movement a person would do when jogging or running in real life. Since the in game character moves towards the direction the motion controllers are facing towards the user can move in one direction while looking towards a different direction. This more closely mimics real life jogging or running since people can move forward while looking left, right, or backwards. An informal user study was conducted during the Innovation Day Workshop at the University of Nevada, Reno and participants stated little to no form of VR sickness when using Wolf Hunts locomotion technique. However, a possible consideration as to why participants experienced very little VR sickness was that they only played the game in short bursts.

Tracking of the motion controllers is done using infrared sensors which are located on each controller. Infrared light is projected from hardware components known as base stations. The base stations use infrared light to calculate the distance and location of each motion controller. Two base stations were used to increase the accuracy of detection for each controller. Each controller also uses an inertial measurement unit (IMU) to reduce latency. We did not consider full body locomotion tracking and focused all our efforts on using a VR headset and motion controllers to detect locomotion.

Within Wolf Hunt the procedurally generated map involved having mountains, lakes, trees, hills, grassy terrain, rocky terrain, and a cabin which was positioned randomly within the map. The mountains were always positioned in such a way that they would surround the outside edges of the map. This was done so that a user could not easily get outside the map since the mountains would be blocking their way and the user would not be capable of climbing up the mountains. This proved to be effective since no users managed to get outside the map. The procedural generation of the map served as a means to add variability and an element of randomness to each play session. Since the map formation changed with each play session a user would be capable of playing Wolf Hunt multiple times and discover something new each time.

The new interface, as shown in Figure 5, introduced within Wolf Hunt was the mobile phone object which the user could interact with. This interface method benefited Wolf Hunt because it is a novel and new interface which takes full advantage of using VR technology. Since VR technology allows for more natural realistic user interaction with a game it would be best to find new more intuitive methods in which users can interact with a game. The mobile phone demonstrates an intuitive method in which a user could navigate through menus within a game. Most users already have real life experience using mobile phones and can thus use that experience to help them easily learn how to use the mobile phone within Wolf Hunt. The mobile phone menu interface also helps to immerse a user in the VR world since the action of carrying a mobile device or smart phone is a norm in most parts of modern society. This new user interface could not be easily used in other applications but does set the basis for a framework which could be customized so that it could be easily adjusted to be used in any application. Some attributes which might need to change in the user interface for each application is the size of the phone, the size of the screen on the phone, and the size of the text on the screens.

6 Conclusions and Future Work

Our project lays the foundations of an interesting and engaging virtual reality experience. Wolf Hunt is a survival horror game which makes use and takes full advantage of the newly emerging VR technology. Wolf Hunt introduces a new locomotion technique which could potentially reduce the amount of VR sickness a user experiences. Procedural generation was used to create the map within Wolf Hunt. With each play session the map changes which introduces an element of variability into Wolf Hunt. A novel and new user interaction was used within Wolf Hunt which is used in the form of menu navigation. This new user interface is an example of and demonstrates how VR can be used to increase realistic interaction between a user and a game.

It was mentioned previously in this paper that an informal user study was conducted to evaluate the effects of VR sickness when using Wolf Hunt's new locomotion method. In the future we would would like to conduct an official user study to gather statistical data on the overall effects of VR sickness in relation to using the new locomotion method. This official user study would have participants use the new locomotion method for longer periods of time compared to the informal user study presented in this paper.

Other work that can be done in the future could be adding in new maps, different game modes, and even a multiplayer system. New levels would add more overall replayability to Wolf Hunt since a user would not need to see the same environment and could instead switch between different environments. Different game modes would allow for a larger audience to be targeted since each game mode will bring a unique experience to the user. Possible game modes could be a timed mode and an exploration mode with no objective to complete. A multiplayer system would allow for two or more users to play competitively against each other which would bring about a sense of competition between users.

References

[1] Hrvoje Benko, Christian Holz, Mike Sinclair, and Eyal Ofek. Normaltouch and texturetouch: High-fidelity 3d haptic shape rendering on handheld virtual reality controllers. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 717–728. ACM, 2016.

[2] Evren Bozgeyikli, Andrew Raij, Srinivas Katkoori, and Rajiv Dubey. Point & teleport locomotion technique for virtual reality. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*, pages 205–216. ACM, 2016.

[3] HTC Corporation. HTC Vive — HTC Vive - discover virtual reality beyond imagination. (Available from: `www.vive.com/us/`). [Online; accessed March 13, 2019].

[4] Jonas Freiknecht and Wolfgang Effelsberg. A survey on the procedural generation of virtual worlds. *Multimodal Technologies and Interaction*, 1(4), 2017.

[5] M.P. Jacob Habgood, David Wilson, David Moore, and Sergio Alapont. Hci lessons from playstation vr. In *Extended Abstracts Publication of the Annual Symposium on Computer-Human Interaction in Play*, CHI PLAY '17 Extended Abstracts, pages 125–135, New York, NY, USA, 2017. ACM.

[6] Ken Hartsook, Alexander Zook, Sauvik Das, and Mark O Riedl. Toward supporting stories with procedurally generated game worlds. In *Computational Intelligence and Games (CIG), 2011 IEEE Conference on*, pages 297–304. IEEE, 2011.

[7] Hak Gu Kim, Wissam J Baddar, Heoun-taek Lim, Hyunwook Jeong, and Yong Man Ro. Measurement of exceptional motion in vr video contents for vr sickness assessment using deep convolutional autoencoder. In *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*, page 36. ACM, 2017.

[8] Thammathip Piumsomboon, Gun Lee, Robert W Lindeman, and Mark Billinghurst. Exploring natural eye-gaze-based interaction for immersive virtual reality. In *3D User Interfaces (3DUI), 2017 IEEE Symposium on*, pages 36–39. IEEE, 2017.

[9] Ruben M Smelik, Klaas Jan De Kraker, Tim Tutenel, Rafael Bidarra, and Saskia A Groenewegen. A survey of procedural methods for terrain modelling. In *Proceedings of the CASA Workshop on 3D Advanced Media In Gaming And Simulation (3AMIGAS)*, pages 25–34, 2009.

[10] The Stonefox. The virtual reality toolkit - vrtk. (Available from: `vrtoolkit.readme.io/`). [Online; accessed March 13, 2019].

[11] Unity3D. Unity - game engine. (Available from: http://unity3d.com/). [Online; accessed March 13, 2019].

[12] Oculus VR. Oculus VR — Oculus Rift - virtual reality headset for immersive 3d gaming. (Available from: www.oculus.com/). [Online; accessed March 13, 2019].