The SNaP Framework: A VR Tool for Assessing Spatial Navigation

Michelle ANNETT[a,1] and Walter F. BISCHOF[a]
[a]*Department of Computing Science, University of Alberta, Canada*

Abstract. Recent work in psychology has leveraged the power of Virtual Reality (VR) to study the deterioration of navigation abilities in the elderly. Much of this research has focused on determining the behavioral measurements and paradigms appropriate for such diagnoses. We present a system, the Spatial Navigation Paradigm (SNaP) framework, which can be used to implement a battery of spatial navigation paradigms. This framework integrates a popular VR environment development platform with an extensible representation medium to allow for the precise control of paradigms, the switching between input and output devices, and the recording of accurate behavioral measurements. A preliminary study of the framework indicates that novice and expert VR users are able to quickly and easily specify and deploy experiments and that expert VR users can easily modify and extend existing paradigm implementations.

Keywords. Virtual Reality, Spatial Navigation, VR System, VR Peripherals

Introduction

Virtual Reality (VR) has been used to provide controlled environments for assessing, among other things, the deterioration of navigation abilities in elderly individuals and patients with Alzheimer's disease and dementia. VR as an assessment tool has not found widespread adoption, mostly because the development and deployment of VR environments exceeds the technical abilities of the average experimenter or technician.

Problems similar to this must be overcome if VR is to be widely adopted: We must make it easy for experimenters to deploy a VR environment, to choose between different input and output devices, to obtain a range of behavioral measurements, and to interface with other systems (EEG, fMRI, etc.). Similarly, we want to make it easier for experts to develop new environments and paradigms. As most existing systems [1-3] do not support these requirements, the Spatial Navigation Paradigm (SNaP) framework was developed to achieve these goals.

In this paper, we discuss the design and evaluation of our proposed solution, the SNaP framework. First, we review some of the approaches used to create VR-based spatial navigation experiments. Second, we briefly detail the architectural design of the SNaP framework, highlighting its features and configuration media. Third, we describe a usability study that was performed with both novice and expert VR users to assess the ease of use and flexibility of the SNaP framework.

[1] Corresponding Author: Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada; E-mail: mkannett@cs.ualberta.ca.

1. Existing Systems

In spatial navigation research, two methods have been used to implement VR-based experiments: (1) creating custom in-house systems composed of freely available or tailor-made components, and (2) using pre-packaged open source or commercial systems that integrate specific VR devices with a virtual environment generator.

Many research teams have created custom VR systems. Mraz et al. developed an fMRI-compatible VR system that relied on OpenGL, the C and Visual Basic programming languages, and the WorldUp environment generator software [4]. Maguire et al.'s and Pine et al.'s virtual city tasks used the Duke Nukem 3D game engine to create and render environments and to record behavior-based measurements [5, 6]. All of these teams used a custom VR setup that required at least one team member to have an extensive programming background. This requirement makes it very difficult for novices to use these, or similar, systems. These systems are often largely inflexible; it can be very time consuming to change an environmental landscape, monitor new behavior metrics, change the types of allowable user actions, or introduce new deployment contexts.

Although less popular, there have also been a number of pre-packaged systems created. The Presentation system [1] by Neurobehavioral Systems has been adapted for use in spatial navigation studies. Presentation uses a drag-and-drop GUI to create and control 3D stimuli, navigable spaces, and experiments. It interfaces with a variety of input devices, supports eye-tracking hardware, allows for the integration of fMRI and MEG devices, and contains a scripting language to handle complex paradigm logic. Although Presentation has been used for navigation studies, the created environments are neither immersive nor realistic enough to produce generalizable results.

Psychology Software Tools Inc. has developed VR Worlds 2 [2], a software system focused on creating realistic virtual environments for several research domains (e.g., drug rehabilitation, phobia therapy, and anxiety disorders). VR Worlds 2 combines a drag-and-drop interface with 3D object libraries, custom event handling, data logging, peripheral device interfacing, motion tracking, and fMRI support to create realistic virtual environments. Unfortunately, VR Worlds 2 does not give the user much control over the created environments: the user is not allowed to create an arbitrary environment and is limited to using pre-programmed environment options.

One of the newest suites that has become available is the open source NeuroVR platform by Riva et al [3]. NeuroVR is a Blender-based platform that contains a drag-and-drop, icon-based editor interface for creating and modifying rich virtual environments. NeuroVR comes with a library of pre-created 3D models, it can be deployed to a HMD or monitor, and it supports the inclusion of head-trackers, joypads, keyboards, and mice. Although targeted towards phobia and addiction research, it appears that NeuroVR could be an ideal platform for spatial navigation research. It currently lacks, however, many important features, including stereo rendering capabilities, physiological monitoring support, the ability to create blank canvas environments, and a scripting medium.

In short, there is no ideal approach that can be taken to create spatial navigation experiments. Custom systems are inflexible and require team members who are strong in programming. Pre-packaged systems are useful for novices but are purposely generic, do not support experimental protocols, and are limited by the types of peripherals they support. Extensions and additions to these systems are also difficult, thus complicating the setup of experiments. Our framework is aimed at eliminating some of these problems.

2. The SNaP Framework

Motivated by the problems and deficiencies inherent in both custom and pre-packaged systems, we designed a hybrid system. Our system combines the strengths of custom systems (e.g., environmental control, strict stimulus control, custom metric monitoring) with those of pre-packaged systems (e.g., drag-and-drop interface, visual programming language). The SNaP framework allows for on-the-fly usage of multiple hardware media and experimental paradigms, overcomes the limitations inherent in pre-packaged systems, and decreases the time and effort required to implement and deploy an experiment.

The SNaP Framework was built using the Virtools development platform [7]. It uses XML schemas, a VRPN server, and two Python modules to specify, configure, and deploy VR-based spatial navigation paradigms. The deployment of a spatial navigation paradigm proceeds as follows: First, the experimenter creates an XML-based *parameter file*. This parameter file contains information about the different experimental phases, blocks, and trials of an experiment, as well as the input and output peripherals to be used.

After the user has written a parameter file, a paradigm specific batch script is used to deploy the experiment. This batch script passes the parameter file to the first Python module, the *VR Configuration Creator*. This module controls and supervises the execution of the experiment. The VR Configuration Creator converts each trial specified in the parameter file into a *configuration file*. A configuration file is an XML formatted document that specifies the interface configurations (i.e., the input, output, and alternative devices and requested virtual environment), environmental setup, paradigm-specific information (e.g., goals, trial type, and presence or absence of feedback), and behavioral measurements that are required for a single experimental trial.

This configuration file is then used as input to the second Python module, the *VR Launcher*. This module starts the Virtools VR Player (to play the desired virtual world) along with the VRPN software (to capture peripheral device data) and opens the virtual world, or *Virtools composition file*, that was specified in the configuration file. Each Virtools composition file in the SNaP framework includes all of the 3D models and logic necessary to control the parsing of configuration files, virtual environment modifications, behavioral measurement recording, and trial goal monitoring. Once a participant has completed a trial, the VR Configuration Creator writes the next trial-specific configuration file and indicates to the VR Launcher that it can render and execute the next virtual environment. This process repeats until all trials have been completed.

To simplify the creation of new environments and paradigms, an expert user can start from, and expand, a template environment. This template contains all of the logic and modules required to implement new paradigms. Using this template, we were able to implement five popular spatial navigation paradigms; it only took approximately five hours to design, implement, and test each paradigm.

All paradigms share a single experiment specification file format, environmental objects, behavioral interaction techniques, navigational methods, support for input and output devices, and methods for gathering behavioral measurements. Multiple levels of results are recorded; participant path information and camera frustum bitmap files are automatically recorded for every paradigm. Paradigm specific behavioral results can also be easily added or adjusted to meet a user's needs. As most measurements are the same across paradigms, similar algorithms and timing schemes are used, making it easy to compare results obtained with different paradigms and participants.

With this architecture, it is easy to switch between paradigms, input devices, and output contexts, and to include a wide range of measurements. To switch between input devices, for example, a user only needs to change the 'input device' keyword in the parameter file. The SNaP framework currently supports joysticks, keyboards, mice, space mice, trackers, wands, EEG and fMRI devices, the Nintendo Wiimote and the Wii Balance Board as input devices. The SNaP framework also supports the use of CAVEs, HMDs, single and multiple monitors as output devices.

3. Evaluation

It is difficult to compare new VR systems to existing ones because each system supports different paradigms, virtual environments are rendered differently, behavioural measurements are recorded using custom techniques, and the steps required to design and deploy a virtual environment vary from one system to another. The best one can do to judge a new system is to evaluate its usability and determine if the system supports the skill sets of its target audiences.

A usability study was performed to assess the effectiveness and user satisfaction of the SNaP framework. In the study, eight participants (four male and four female) were asked to perform tasks using the SNaP framework. The participants were between 19 and 48 years of age. They included computer and VR novices (3 participants), computer experts and VR novices (3 participants), and computer and VR experts (2 participants). Novice users had minimal or no exposure to either computer programming or virtual reality; experts users were very familiar with the technology in question and used it on a regular basis.

All participants were required to use the configuration media to specify and then deploy two spatial navigation experiments. Participants had to create a 2 phase, 8-trial parameter file that would be used to deploy an existing spatial navigation paradigm. Once a parameter file was written, the participant was required to run the batch file associated with the implemented paradigm. If the task was completed successfully, the participant was able to run through the sample experiment that had just been written. This task was performed once more using a different existing paradigm.

VR experts were asked to perform two additional tasks. Both tasks required the VR experts to make significant modifications to the types of feedback available in an existing paradigm. In the first task, participants made a number of hidden objects appear temporarily; in the second task, participants changed an implementation to enable a dissonant sound to be played whenever there was a collision with an object.

Recorded measurements included the time to task completion, the number of errors made, and the amount of help requested. All participants were able to complete the first two tasks in less than twenty minutes (a mean time of 10 minutes and 8 minutes for the two tasks), they made fewer than two XML typographical errors, and they asked the experimenter, on average, one question. For the additional expert tasks, each task was completed in less than twenty minutes, with an average of four compilations made and two questions asked.

At the completion of the study, participants were asked to fill out a modified version of the IBM Post-Study System Questionnaire [8] to assess their usability beliefs. The results indicated that all users were satisfied with the system and felt that experiment specification and deployment and peripheral switching was easy. All of the computer and VR novices also felt that the usage of XML as a configuration media was an appropriate choice; they indicated that it was very easy to understand and that the XML format greatly helped them. The expert users (computer and VR experts) agreed that it was easy to extend the currently implemented paradigms and that it would be quite simple to implement a new paradigm using the provided template environment.

4. Conclusions

The SNaP framework assists novices and experts in designing, specifying, and deploying VR-based spatial navigation paradigms. The framework addresses the needs of experimenters and developers. Experimenters do not need an extensive programming background; they are given the capabilities to tailor different spatial navigation paradigms to their needs, and they can deploy experiments in a simple way with multiple input/output devices. Expert users are given control to edit existing paradigms and to create new paradigms using the provided template environment.

References

[1] Neurobehavioural Systems, http://www.neurobs.com, 2008.

[2] VR Worlds 2 - Software for Research and Education in Psychology, http://www.vrworlds2.com/index.cfm, 2008.

[3] G. Riva, A. Gaggioli, D. Villani, A. Preziosa, F. Morganti, R. Corsi, G. Faletti, and L. Vezzadini, A Free, Open-Source Virtual Reality Platform for the Rehabilitation of Cognitive and Psychological Disorders, *Virtual Rehabilitation* (2007), 159-163.

[4] R. Mraz, J. Hong, G. Quintin, W.R. Staines, W.E. McIlroy, K.K. Zakzanis, and S.J. Graham, A Platform for Combining Virtual Reality Experiments with Functional Magnetic Resonance Imaging, *CyberPsychology and Behaviour* **6** (4) (2003), 359-368.

[5] E.A Maguire, N. Burgess, J.G. Donnett, R.S.J. Frackowiak, C.D. Frith, and J. O'Keefe, Knowing where and getting there: A human navigation network, *Science* **280** (5365) (1998), 921-924.

[6] D.S. Pine, J. Grun, E.A. Maguire, N. Burgess, E. Zarahn, V. Koda, A. Fyer, P.R. Szeszko, and R.M. Bilder, Neurodevelopmental aspects of spatial navigation: a virtual reality fMRI study, *NeuroImage* **15** (2) (2002), 396-406.

[7] Virtools, http://www.virtools.com, 2008.

[8] J.R. Lewis, IBM Computer Satisfaction Questionnaires: Psychometric Evaluation and Instructions for User, *International Journal of Human Computer Interaction* **7** (1995), 57-91.