Comparison of 3D Navigation Interfaces

Joshua McClymont, Dmitri Shuralyov, Wolfgang Stuerzlinger
Computer Science and Engineering
York University
Toronto, Canada
joshua@cse.yorku.ca

Abstract— **This paper evaluates two different interfaces for navigation in a 3D environment. The first is a 'Click-to-Move' style interface that involves using a mouse. The second is a gaming style interface that uses both mouse and keyboard (i.e. the 'WASD' keys) for view direction and movement respectively. In the user study, participants were asked to navigate a supermarket environment and collect a specific subset of items. Results revealed significant differences only for some of the sub-measures. Yet, some revealing observations were made regarding user behavior and interaction with the user interface.**

3d navigation, 3d virtual environments, gamer interface, supermarket simulation, user studies.

I. INTRODUCTION

Navigation is an essential tool in any type of 3D Virtual Environment (VE). Regardless of the size of the environment, most tasks in VEs require a change in viewpoint. Thus and although navigation is one of the most common user interface facilities, it is still one of the hardest to understand for novices. This paper explores two types of navigation interfaces. The first interface maps all navigation related actions to a single input device, namely the mouse. The second interface is inspired by common conventions in the gaming community. It is similar to the user interface for 'first-person-shooter' games in which user movement is mapped to the keyboard and view direction to the mouse.

The goal of this study is to investigate how these two different alternative control mappings affect the ability of users to navigate in a 3D environment. For this we evaluate the two interfaces using a task that involves collecting a specific subset of items in a supermarket environment. The layout of the items to be retrieved was fixed in order to enforce navigation to certain areas of the supermarket. We then record several time related metrics to determine any significant difference between them. Since this supermarket task involves both navigation and manipulation, both interfaces must provide facilities for both while also maintaining a simple-to-use interface for the user.

II. RELATED WORK

Both interfaces presented in this paper draw upon existing implementations for navigation in 3D environments. The Click-to-Move interface was already implemented as part of a 3D desktop puzzle assembly system [1]. This type of navigation interface was designed for large-scale environments in which movements are mapped to a combination of keys and mouse buttons. The movement methods was inspired by previous work [3], which presented a multi-scale navigation method that can transition between navigation at different scales. This method uses the depth buffer to identify the object under the cursor as well as to compute an average movement vector. When the user clicks the mouse button the method travels to the object under the cursor. Combined with collision resolution this enables the user to click on any area they would like to navigate to and the system computes an appropriate path. Our user study modifies this method by removing some of the movement features that permitted the user to adjust viewpoint height. This was done to keep both navigation interfaces comparable in terms of capabilities.

The second user interface draws upon one of the many different navigation interfaces that have been purposed by the gaming community. There, each genre of game usually involves a different approach to navigation. For 'first-person shooters', navigation typically uses a combination of mouse and keyboard. The keyboard keys 'WASD' are used for movement forward, a side step ("strafe") left, backward, and a side step right, respectively. The user's view direction is controlled by the mouse so that moving the mouse rotates the view direction of the users head. Normally, in 'first-person shooters' a crosshair is fixed in the center of the screen and serves the same purpose as a cursor (an indicator where actions can be executed). However, our interface modifies this aspect to allow a more seamless transition between navigation and manipulation.

III. SYSTEM DESIGN

In this section we give more detail on the implementation of both navigation interfaces as well as the virtual environment. As mentioned earlier, the environment used was a virtual supermarket (Figure 1), which has been used for 3D user interface competitions. The interior contains the basic architecture of a supermarket such as shelving on the walls and in the center area to create shopping isles. This store is stocked with items that are typical of what one might find at a supermarket: a bread section, freezer foods, and general grocery. At one end of the environment three unique items are displayed on a table to indicate the items that are to be collected and then placed to complete the required task (left side of Figure 1).

Some modifications were then made to the existing implementation to remove any potential biases in terms of navigation patterns. In the original task the user started in the bottom left corner of Figure 1. This starting position, along

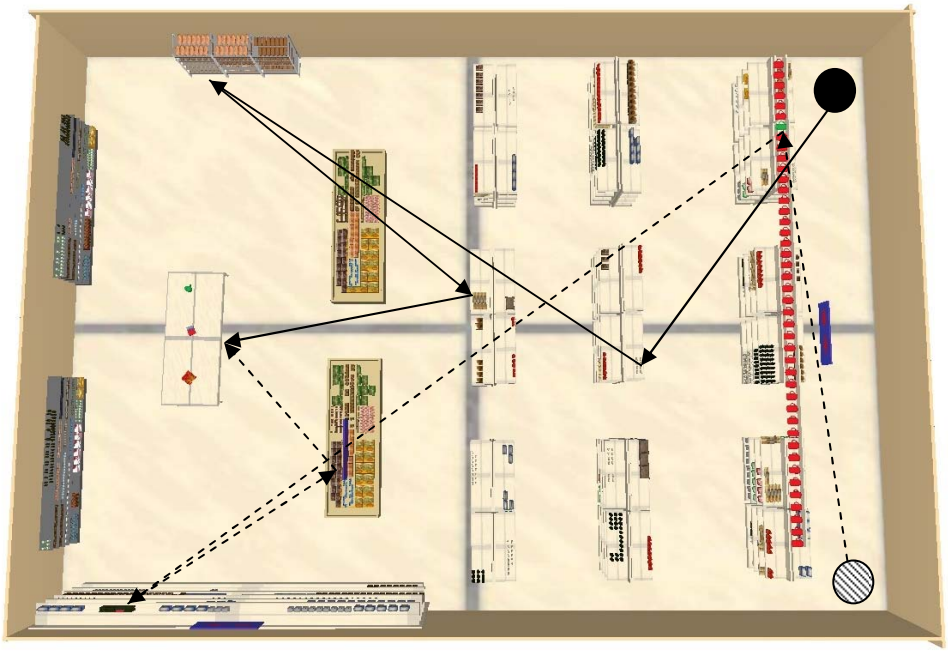

Figure 1. Bird's eye view of the supermarket environment. Both initial positions are displayed and the directions to their three respective items.

with the ability to select objects at *any* distance, enabled the user to collect items with very little navigation. Thus, we added a restriction to the selection of objects so that the user had to get 'within reach' of any item they wanted to select. We also placed the user initially at two different positions, where the target items were not directly visible (the two marked positions in the right of Figure 1). These changes encouraged more navigation through the environment.

We also added a shopping cart, a common feature in supermarkets. This allowed the user to first visit all required items in the supermarket and collect them in the cart and thus to bring all of them at once to the final position. The shopping cart mechanism supports both adding items and selecting them from the cart to add them back to the scene. To add an item the user first selects the item with the cursor, highlighting the object, and then presses the 'Q' key to add that item to the cart. To retrieve an item from the cart the user presses the 'E' key. The user is then shown a separate interface for selecting items from the cart. Once an item has been selected the user is brought back to the environment with the item fixed under the cursor. The user can then place and manipulate the item as they see fit. This shopping cart feature is available in both navigation interfaces.

Along with the previously mentioned features some actions and bindings are also shared. Common to both interfaces, object manipulation is done using the left mouse button. Clicking and holding will move the object around the environment using a sliding technique similar to that of SESAME [4]. The right mouse button is used for rotating the object, which is enabled but was not required for this user study.

A. Navigation Interfaces

Both navigation interfaces considered in this paper were restricted to two input devices; mouse and keyboard. One method maps all control to a single device while the other uses both. The aim of our work is to evaluate whether this division of responsibilities provides a significant difference in performance.

1) Click to Move

The 'Click-to-Move' navigation interface uses only the mouse for navigation. This method is a modification of an existing navigation technique for large 3D environments [3]. For their technique movement is based on distance of the geometry underneath the mouse cursor to the viewer. It uses the depth buffer to identify nearby objects and computes an average movement vector based on the distances to all nearby objects.

In our implementation, we used a mouse that has two side buttons (buttons three and four), which are situated at the left side of the mouse. These two buttons move the user towards or respectively away from the position under the cursor. This type of navigation coupled with collision resolution allows the user to quickly move around objects in the environment. Moreover, pressing and holding both side mouse buttons simultaneously is used to change the view direction. In this case, the user will remain stationary but be able to look around the environment. The underlying system implements more camera manipulation features via combinations of mouse buttons and keys [3]. All these other features were disabled for this evaluation, both to keep the user study simple and to restrict the freedom of the user when navigating through the environment (such as adjusting the viewpoint height to go over obstacles).

2) WASD

The second interface uses both mouse and keyboard for navigation. This divides navigation control into two parts, where each device is responsible for a certain component of

navigation. This design follows the most frequently used design in the 'first-person shooter' genre closely. There, the keyboard is designated to translation of the camera. The 'WASD' keys invoke movement forward, a side step left, backward, and a side step right respectively. This movement is relative to the current view, independent of the mouse cursor position. To change the view direction the user presses and holds either of the side mouse buttons. When pressing and holding either side button the cursor disappears as a secondary indication that the view direction is tethered to the mouse movement. Likewise, when the buttons are not pressed the cursor is displayed so the user can then use it to select (and manipulate) objects. This user interface implementation enables a seamless transition between navigation and manipulation without the need for designating a button or keys solely for mode switching.

An important attribute to note about this interface is that some movements can only be achieved by using the keyboard and mouse buttons in conjunction. To turn while moving forward or backward it is necessary to hold down the 'W' or 'S' key simultaneously with a mouse side button, and then moving the mouse left or right to "steer". Orbiting an object can also be achieved by holding 'A' or 'D' simultaneously with a side mouse button and moving the mouse to change the view direction simultaneously while the user side steps. This makes this user interface more powerful than 'Click-to-Move'. If done properly, one can keep a point of the virtual environment fixed in the center of the view while moving around it.

IV. METHOD

In the following section we outline the details of the user study as well as the data acquisition. The user study used a computer with the following configuration: a 3.2 GHz QuadCore CPU, GeForce GTX470 graphics card, and 4 GB of main memory. The input devices were a standard keyboard and a Logitech G9 laser precision gaming mouse. A 17" 1440x900 LCD monitor was used for output. The supermarket simulation was rendered using OpenGL at a fixed frame rate of 30 frames per second.

A total of 12 participants were recruited for the study. Of this 8 were male and 4 female with an average age of 25. Most were experienced computer users averaging 6 hours per day. Most participants had no gaming experience. Only a few had a maximum of 4 hours per week and we did not consider these "gamers". A single participant had over 5 hours of gaming usage a week, but was not playing PC games, i.e. was used to console games. We targeted non-gamers as one of the navigation interfaces mimics an interface frequently used in PC gaming.

The task was to collect three predetermined items from the supermarket and place them on a table situated on one side of the environment. Two different sets of items were used to enforce navigation to all parts of the environment and to remove the potential for shortcuts. Each user was asked to do this task five times for each set of items for a total of 10 times per navigation technique. The data recorded started with the first mouse/key action and ended when the user pressed the spacebar to indicate that they were finished.

Each user was given a one-minute introduction to the system, which involved an explanation of the controls and the location of items. This allowed them to get acquainted with the navigation facilities and also to locate the items. We did this, as we were not interested in evaluating the ability to search for unknown items. Instead we focused only on navigation performance. Each participant then was given two minutes of practice with each navigation technique. Thus, the complete introduction lasted a total of five minutes.

A. Data Definition

We collected timings to assess the performance of each navigation technique. The following defines the meaning of each of these measurements. Note that all measurements are in terms of seconds.

1) Total Time

Total time was recorded from when the simulation was initiated until the spacebar was pressed by the user to indicate that they had completed the task.

2) Camera Translation

This was the time spent translating the camera. For the 'Click-to-Move' interface this recorded how long a single side mouse button was pressed. For 'WASD' this was the time any of the 'WASD' keys were pressed. Both were incremented only while the key/button was being pressed and this stopped when the respective key/button was released again.

3) Camera Rotation

This was the time spent rotating the view direction of the camera vertically or horizontally. For 'Click-to-Move' this recorded the time when both side mouse buttons were pressed simultaneously, whereas for 'WASD' it recorded when only a single side mouse button was pressed. Like for camera translation, this was only incremented while the appropriate key/button was pressed.

4) Camera Total

The time spent either translating or rotating the camera as per the definitions above.

5) Navigation

This recorded the time spent navigating the environment as opposed to manipulating objects. Whenever a key or button related to navigation was pressed (either translation or rotation of the camera) this timer was started. When the user performed any object manipulation action this timer was paused. For 'Click-to-Move' the timer was started when any combination of side mouse buttons was pressed. For 'WASD' navigation this time was started for any press of either a 'WASD' key or the side mouse buttons. This differs from previous measurements in that the timer was started when a button was pressed (and did not have to be held down for the timer to continue incrementing).

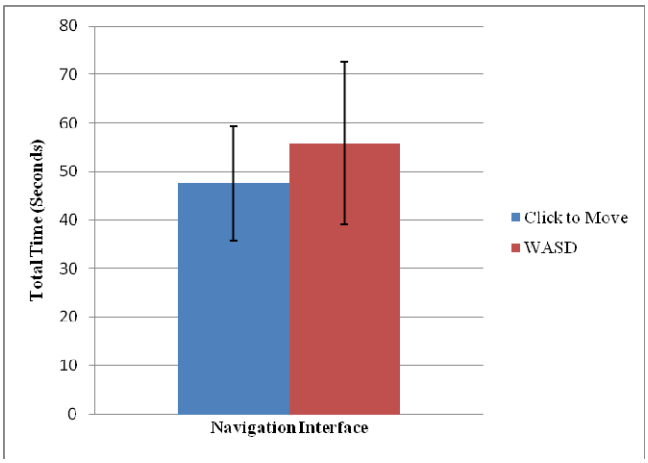

Figure 2. Average total time of task with standard deviation lines.

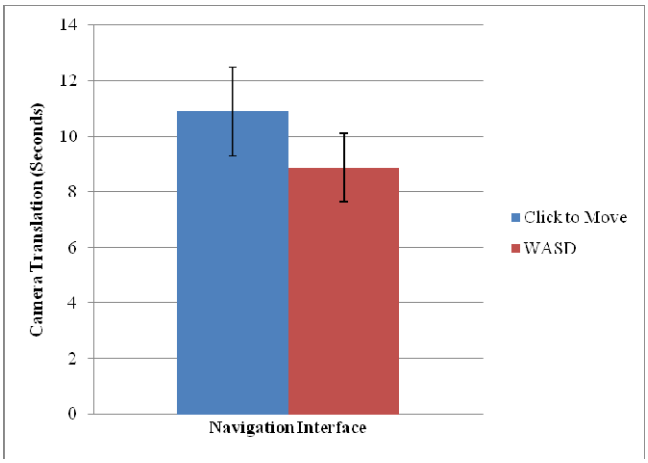

Figure 3. Average camera translation time of task with standard deviation lines.

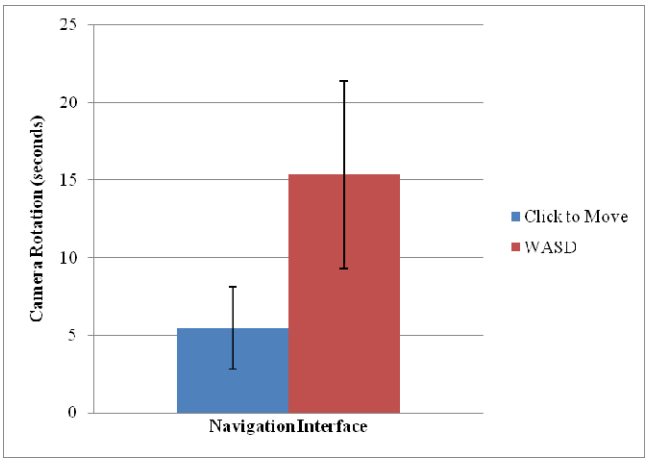

Figure 4. Average camera rotation time of task with standard deviation lines.

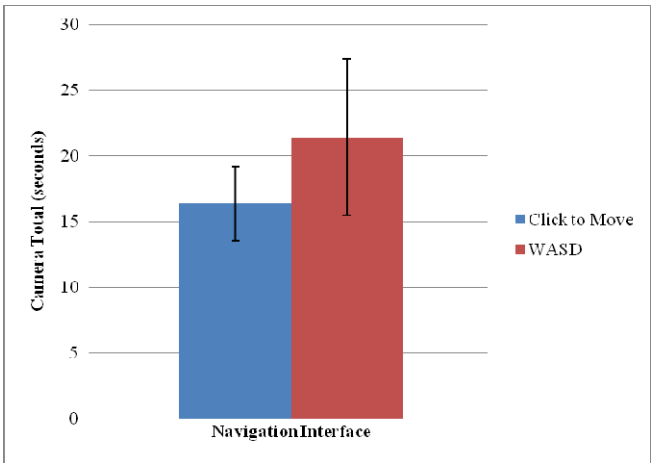

Figure 5. Average camera translation or rotation time with standard deviation lines.

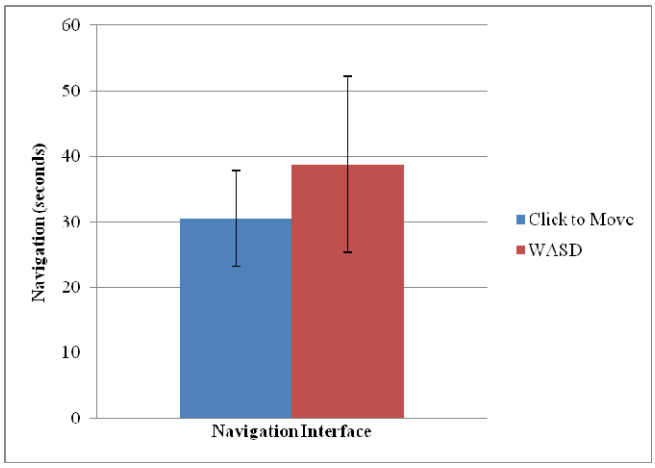

Figure 6. Average navigation time of task with standard deviation lines.

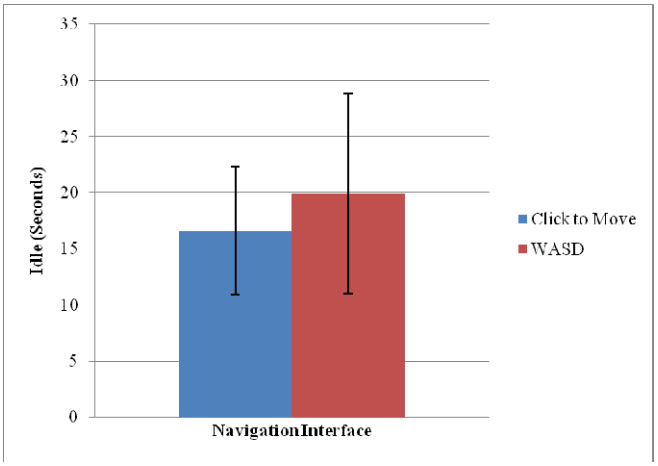

Figure 7. Average idle time of task with standard deviation lines.

6) Idle

This is the time the user spent doing no actions. This is calculated using navigation time minus camera total time. The idea behind this was to capture the time the user spent pausing while navigating.

V. RESULTS

The results of the study are presented in the appropriate sections for each of the measurements defined above. We elaborate more on these results in the discussion section.

1) Total Time

The average total time for 'Click-to-Move' interface was 47.54s with standard deviation (SD) 11.85s. 'WASD' average total time was 55.81s (SD 17.763s). See Figure 2. This difference of only 8.27s was not significant according to a one way ANOVA with $F(1,22) = 1.95$, $p > 0.1$.

2) Camera Translation and Rotation Time

An ANOVA revealed significant differences in terms of both camera translation and rotation. The average time for camera translation for 'Click-to-Move' was significantly different (Figure 3), with $F(1,22) = 11.6$, $p < 0.003$. The average time spent rotating the camera for 'Click-to-Move' was 5.49s (SD 2.66s) versus 'WASD' of 15.35s (SD 6.06s) which was also significantly different with $F(1,22) = 26.57$, $p < 0.003$. In this case 'WASD' was faster by nearly three times (Figure 4). These two measurements illustrate the behavior differences between the two interfaces. Although both offer practically the same functionality, the participants favored one type of interaction with the camera over the other depending on the interface.

Another interesting result is the average of 'camera total' times. For 'Click-to-Move' this was 16.38s (SD 2.85s) versus 'WASD' 21.43s (SD 5.93s) (Figure 5). The difference is significant, $F(1,22) = 7.03$, $p < 0.05$. Less total camera time means less interaction with the camera manipulation keys/buttons. We speculate that overall movement with the 'Click-to-Move' interface was less since users were able to directly click where they intended to go, whereas in some cases with 'WASD' participants would get stuck attempting to align themselves to a desired view.

3) Navigation Time

The average time spent navigating through the environment for 'Click-to-Move' was 30.56s (SD 7.37s) versus 'WASD' of 38.77s (SD 13.46s) (Figure 6). ANOVA revealed that this difference was not significant, $F(2,22) = 3.42$, $p > 0.05$. Since the difference in total time was not significant, this was not surprising as our study involved mostly a navigation task. The average idle time for the participants was different by only 3.3s (Figure 7), which was not significant, $F(2,22) = 1.13$, $p > 0.2$.

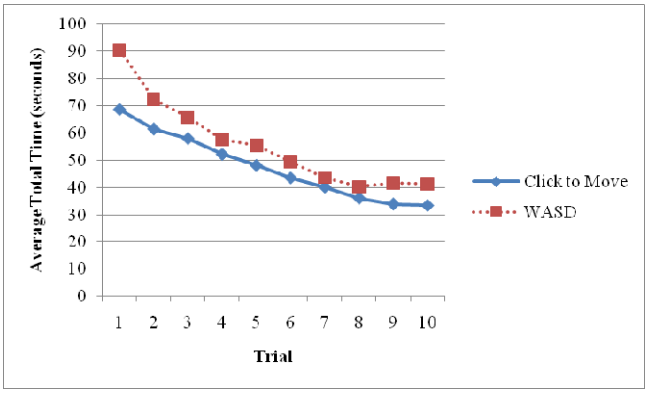

Figure 8. Plot of average total time versus trial. This shows the learning curve for each interface over 10 trials.

4) Learning Curve

Another interesting question was the rate at which participants were able to learn either of the navigation interfaces. For this we plot the average time that all participants took to complete each trial versus the respective trial (Figure 8). Fitting an exponential trend line yielded the formula $y = 73.447e^{-0.085x}$ for 'Click-to-Move', and $y = 86.822e^{-0.087x}$ for 'WASD'. Although initial values of 'WASD' on average took more time, both methods exhibited similar learning curves. Approximately at the 8th trial both seem to level out.

VI. DISCUSSION

Most of the differences discussed in the previous section are not significant between the two navigation interfaces. After the users completed the task they also completed a questionnaire that used a 7-point Likert scale. The users were asked questions such as if the controls responded to their actions as they thought they should (accuracy), and whether it was easy to use to perform the task (ease of use). Participants found the controls both similarly accurate (5.75 for 'Click-to-Move' and 5.25 for 'WASD') as well as easy to use (5.67 and 5.92). These results illustrate further that both interfaces exhibited little difference.

To further understand the results we elaborate on some of the observations made during the user study on how the participants behaved during the task and also their interaction with the controls. For both interfaces some participants preferred some behaviors over others. An example of this is the 'Click-to-Move' interface, where participants moved in a manner that resembled driving a vehicle. When faced with a corner instead of rotating the view angle away from the corner, they instead continually reversed and went forward in a similar fashion to a three-point-turn. For movements with the 'WASD' interface, in some cases participants would not press and hold the appropriate keys, but instead repeatedly tap them to make precise movements to avoid objects or to better align themselves. However, in the 'Click-to-Move' interface they used the collision resolution facility to help them navigate around objects by actively sliding against them.

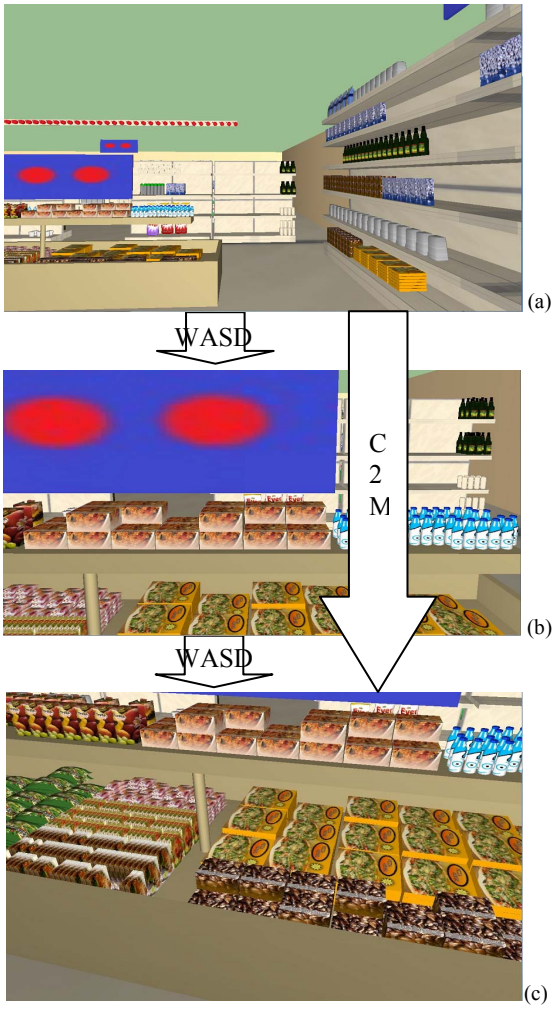

Figure 9. Movement from initial position (a) to final position (c). 'WASD' takes 2 steps (a)-(b)-(c) whereas 'Click-to-Move' takes only 1 step (a)-(c). Circles represent the target and dashed line is a path.

In many cases participants also did not use all the features available to them. The design of the 'WASD' interface requires users to use the keyboard simultaneously and in conjunction with the mouse for movement. However, most participants did not use both devices at the same time. Using both devices simultaneously permits 'parabolic' or 'arching' movements that are more optimal in some scenarios. Figure 9 depicts one scenario where participants typically took two separate steps to reach a suitable orientation to acquire the pizza item using the 'WASD' interface, whereas with 'Click-to-Move' they needed only one smooth flying motion. Finally, during repeated trials participants made no attempts to find alternate paths for completing the task. After the first one or two trials they would continue on the same path through the environment (even if not optimal).

Such preferred behaviors may be the result of the participants learning them within the first two times performing the task (either purposely or accidentally). Since they were comfortable doing the task in this way they would repeat their "solution" for the remainder of the trials. This can influence the outcome of the user study depending on whether the behavior was more or less efficient compared to other potentially more optimal behaviors. It also partially explains the pattern in the learning curve mentioned earlier. There is more discrepancy between interfaces for the first two trials since this is where the participants spend the most time to find out how to use the given user interface and thus establish these behaviors.

Another influencing factor concerns the experience of participants. As mentioned in section IV, participants were asked about video game experience and computer usage. Although participants were homogeneous in the sense that they were non-PC-gamers, we did not record how fast and accurately they can use keyboard or pointing devices. An increase in performance with both computer input devices may increase performance with one of the interfaces. For example, someone who is advanced at both keyboard and mouse and can use them simultaneously may perform better with 'WASD' interface since they would use both devices efficiently as intended by the design.

VII. CONCLUSION

This paper evaluated two different navigation interfaces for 3D environments. Our work investigated whether the division of controls inspired by the gaming community, or the use of a single input device permits faster navigation for non-gamers. The results collected from the user study showed few significant differences between the two. Along with this we noted some key observations that provided more evidence for the different behaviors the participants exhibited during the task. Finally we noted some of the outside influences that may be interesting to explore in future iterations.

REFERENCES

[1] D. Shuralyov, W. Stuerzlinger, A 3D Desktop Puzzle Assembly System, IEEE 3D UI Symposium 2011, 141-142, March 2011.

[2] D. Tan, G. Robertson, M. Czerwinski, Exploring 3D Navigation: Combining Speed-Coupled Flying with Orbiting, CHI 2001, 418-425.

[3] J. McCrae, I. Mordatch, M. Glueck, A. Khan, Multiscale 3D navigation, Interactive 3D Graphics and Games 2009, 7-14.

[4] J.-Y. Oh, W. Stuerzlinger, J. Danahy, SESAME: towards better 3D Conceptual Design Systems, ACM DIS 2006, 80-89.

[5] W. Stuerzlinger, C. Wingrave, The Value of Constraints for 3D user Interfaces, Virtual Realities: Dagstuhl Seminar 2008, Springer Verlag,2011,203