MY WARMEST THANKS TO:

DR. CHARLES OMAN,
who dared to welcome a French woman in his lab,
and has given me the opportunity to discover what the researcher’s life looks like.
Thanks for your patience, advice and encouragements along this project.

DR. ALAN NATAPOFF,
who shared with me the mysteries of the English language (and of statistics).
Thanks for cheering me up with your spirit and wisdom when I doubted.

JASON RICHARDS,
my nice predecessor, for designing and doing the programming of the original node experiment.
Special thanks also for helping part of the French community to enter the JSC in Houston.

PR. DAVA NEWMAN,
for letting me drive her boat among the lobster traps in Marblehead. Fair journey!

EVERY MVLER,
for sharing your enthusiasm and your multiple interests with me.

ERIC,
sometimes so far, sometimes so close, but always present in my heart.
I was so happy to share my American experience with you during one semester.

MY DAD, MY MOM, SOLENE AND QUENTIN,
always by my side despite the distance.

BRAD,
who has made those last months so fun.
Thank you so much for your devoted support in any circumstance.
I hope you will keep your kayaker’s soul.

JEROME,
and our common nasty friend, Feathers The Penguin.
Your friendship has been so reliable. Have a good trip with Titine.

I will bring a little piece of all of you back to France, within me.

The present research was supported by NASA Cooperative Agreement NCC9-58
with the National Space Biomedical Research Institute, USA.
THREE-DIMENSIONAL SPATIAL MEMORY TRAINING USING VIRTUAL REALITY: EFFECTS OF REFERENCE FRAME AND POINT-OF-VIEW

by
Bérengère Houdou

Submitted to the Department of Aeronautics and Astronautics
On May 24, 2002 in Partial Fulfillment of the Requirements for the Degree of Master of Science in Aeronautics and Astronautics

ABSTRACT

Humans are usually able to imagine how a terrestrial environment will appear from a new perspective, and usually avoid disorientation because of their ability to transform spatial mental models of their environment. However, astronauts must transform their spatial mental models in three dimensions, and our previous experiments have shown that this is initially not an easy task, though most subjects eventually learn to orient themselves, particularly those who are taught specific strategies. The present experiments are the first to investigate the effects of using a Self-rotation vs. an Array-rotation assumption when transforming mental models of an environment, and to see whether subjects can change rotation modes easily. Also, does performance depend on the point-of-view (Inside vs. Outside)? Subjects (n=29) had first to memorize the arrangement of 6 animal picture icons painted on the interior walls of a virtual cubic room. The room was kept in a fixed orientation during the first four trials in order to allow the subjects to memorize icon locations in a canonical orientation. In each subsequent trial, they were then asked to imagine that either they (Self-rotation mode) or the array (Array-rotation) had rotated. Based on the identification of icons displayed in front and below them, they had to deduce their new orientation and indicate the direction of a specific unseen “target” icon. The speed and accuracy of their responses were recorded. The experimental design was blocked by point-of-view (Inside vs. Outside). Each of the four groups was assigned a specific rotation mode in the first (“training”) phase of the experiment; they were then instructed to switch to the alternate mode in the second (“testing”) phase.

We looked for but found no strong evidence that the groups differed greatly in intrinsic skill. Most subjects managed to master their assigned point-of-view and first rotation mode by the end of the training phase, although learning rates differed between groups. During the training phase, Array-rotation groups performed significantly better than Self-rotation groups (Kruskal-Wallis, p=0.05). However, after training, the group that trained Inside but was tested in Self-rotation mode performed significantly better than the others (Kruskal-Wallis, p=0.005). Even though they had no practice trials during the testing phase, most groups were able to switch rotation modes gracefully, without significant performance decrement. Targets on the left and right were less accurately identified than those in other directions. Implications for spatial memory training of astronauts are discussed.

Supported by NASA Cooperative Agreement NCC9-58 with the National Space Biomedical Research Institute, USA.
Title: Director of the Man-Vehicle Laboratory.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>5</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>7</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>9</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>11</td>
</tr>
<tr>
<td>1. INTRODUCTION</td>
<td>15</td>
</tr>
<tr>
<td>1.1 Transformation of Spatial Models to Resolve Disorientation</td>
<td>15</td>
</tr>
<tr>
<td>1.2 Effect of Point-of-View</td>
<td>18</td>
</tr>
<tr>
<td>1.3 Effect of Rotation Mode</td>
<td>19</td>
</tr>
<tr>
<td>1.4 Overview of the Present Experiment</td>
<td>22</td>
</tr>
<tr>
<td>2. METHOD</td>
<td>24</td>
</tr>
<tr>
<td>2.1 Subjects</td>
<td>24</td>
</tr>
<tr>
<td>2.2 Virtual Reality Display System</td>
<td>24</td>
</tr>
<tr>
<td>2.3 Virtual Environment</td>
<td>25</td>
</tr>
<tr>
<td>2.4 Design</td>
<td>27</td>
</tr>
<tr>
<td>2.5 Procedure</td>
<td>29</td>
</tr>
<tr>
<td>2.5.1 Initial Cube Rotation Testing and Instructions to Subjects</td>
<td>29</td>
</tr>
<tr>
<td>2.5.2 Procedure for Training Phase</td>
<td>29</td>
</tr>
<tr>
<td>2.5.3 Procedure for Testing Phase</td>
<td>31</td>
</tr>
<tr>
<td>2.6 Data Analysis</td>
<td>32</td>
</tr>
</tbody>
</table>
3. RESULTS

3.1 Subjects Excluded from Statistical Analysis
3.2 Experimental Variables
3.3 Main Results
3.4 Evidence for Similarly Skilled Groups
3.5 Effect of Training for each Transformation Type
3.6 Effect of Target Direction Relative to Body Axis
3.7 Effect of Target Direction Relative to Gravity

4. DISCUSSION

4.1 Differences among Individuals
4.2 Learning within Training and Testing
4.3 Change in Rotation Mode and Transfer of Learning
4.4 Training Effect
4.5 Point-of-View and Rotation Mode Effects
4.6 Effects of Relative Target Direction

5. CONCLUSIONS

REFERENCES

APPENDIX A: Programming Code
APPENDIX B: Instructions
APPENDIX C: Questionnaires
APPENDIX D: Graphs of % Incorrect by Set for each Group
LIST OF FIGURES

FIGURE 1.1
THE FOUR TRANSFORMATION TYPES 17

FIGURE 2.1
THE SIX LANDMARK ICONS 25

FIGURE 2.2A
INSIDE VIEW OF THE ENVIRONMENT IN THE BASELINE ORIENTATION 26

FIGURE 2.2B
OUTSIDE VIEW OF THE ENVIRONMENT IN THE BASELINE ORIENTATION 26

FIGURE 2.3
EXPERIMENTAL DESIGN 28

FIGURE 2.4
SCHEMATIC OF EXPERIMENTAL SEQUENCE 30

FIGURE 3.1
%-INCORRECT BY GROUP AND BY PHASE 35

FIGURE 3.2
SIGNIFICANT FRIEDMAN RANKING OF SETS
BY %-INCORRECT FOR ALL FOUR GROUPS 38

FIGURE 3.3
%-INCORRECT BY TRANSFORMATION TYPE AND BY PHASE 39

FIGURE 3.4
%-INCORRECT COMPARISONS
BY TRANSFORMATION TYPE IN THE TRAINING PHASE 41

FIGURE 3.5
%-INCORRECT COMPARISONS
BY TRANSFORMATION TYPE IN THE TESTING PHASE 42
LIST OF TABLES

TABLE 3.1
GROUPS’ COMPOSITION AS USED IN THE ANALYSIS 34

TABLE 3.2
RELATIVE TARGET DIRECTION COMPARISONS BY GROUPS
(KRUSKAL-WALLIS TEST) IN THE TESTING PHASE 44

TABLE 4.1
RESPONSES TO THE QUESTION: “WHICH PHASE DID YOU FIND EASIER?” 49
1. INTRODUCTION

The present experiment focuses on how the observer’s point-of-view – Inside versus Outside an environment – and the rotation mode he adopts – Array- versus Self-rotation – influence his ability to transform spatial mental models, which are crucial to spatial orientation and spatial memory skills. This has become a particularly important issue for astronauts, who are susceptible to disorientation in weightlessness.

1.1 Transformation of Mental Models to Resolve Disorientation

When people encounter a novel environment, they apparently encode a mental representation of it that is stored in memory. The exact nature of this mental representation is not known, but psychologists have made several hypotheses. It may be like a mental image that has the same properties as visual perception (Kosslyn, 1980; Shepard and Podgorny, 1978; Farah, 1988). It is more and more argued, however, that people conceptualize the novel environment in order to elaborate a more abstract mental model (Tversky, 1991; McNamara, 1986). This particularly includes some landmarks and the relations among them.

Some situations require us to imagine an environment from a different perspective without physically moving. That is the case when we find ourselves suddenly disoriented, and must recover a sense of place and/or direction quickly. If, for example, you enter a familiar department store through the wrong door, you have to imagine the spatial layout of the store from this unexpected perspective before you can recover your accustomed sense of familiarity. Since all the information needed to reorient cannot generally be accessed visually,
people rely on their mental models of the environment. This may be true even when they are able to inspect the whole scene visually (Bryant, Tversky and Lanca, 1996). In those situations where the relationship to the environment is unknown, the mental model in its original form acquired during coding is not useful unless it can be manipulated. A novel environment is apparently encoded in a perspective-dependent way (Diwadkar and McNamara, 1997; Franklin and Tversky, 1990; Gaunet and Berthoz, 1999), just as complex objects are visually remembered with respect to one or more canonical perspectives (Tarr, 1995; Tarr and Pinker, 1989; Tarr, Williams et al., 1998). To imagine the same environment from a different perspective requires, therefore, that its mental model be transformed. The present experiment focuses on different ways to execute this cognitive process, which is called, in general, transformation, computation or updating.

Astronauts reported that the transformation of mental models and spatial orientation were harder in orbit. Because they are not accustomed to the larger possibilities of rotation available in weightlessness, they are likely to find themselves suddenly disoriented after an unusual body rotation. On Mir, sense of orientation took at least a month to become “natural and instinctive” (Richards et al., 2002). The study of such a general spatial cognition issue as orientation in weightlessness, with three possible axes of rotation, may also help us to understand better the unconscious mechanisms that we use to orient on Earth.

A mental model of an environment is not a static representation. We transform it continuously to adapt to new perspectives and keep track of our orientation. The present experiment aims to study different strategies we may adopt to perform this transformation. We hypothesize that, as in orthogonal geometric transformations, the transformations of a mental model can be formally decomposed into translations, rotations and reflections. Reflections play no role in our experiment. The component of translation is particularly significant when changing one’s point-of-view with respect to the environment. Two main points-of-view are distinguished: it is always possible to consider ourselves “Outside” a given environment, or embedded “Inside” another of larger scale. Rotation is characterized by two different modes, Array-rotation versus Self-rotation. In everyday life, we interpret changing visual images sometimes as a sign that we are moving, sometimes as a sign that the visual environment around us is moving. To describe relative motion, we can imagine either that
the environment is rotating while we are stationary (Array-rotation) or that we are rotating relative to the stationary environment (Self-rotation). Combinations of point-of-view – Inside versus Outside – with rotation mode – Array- versus Self-rotation – define four different types of transformation that could be used to update spatial mental models: Outside/Array-Rotation, Outside/Self-Rotation, Inside/Array-Rotation and Inside/Self-Rotation, as illustrated in Figure 1.1. The present experiment explores the extent to which each of these transformation types can be learned. It also studies specifically how changes of rotation mode affect subjects’ ability to reorient.

Figure 1.1
The Four Transformation Types.
1.2 Effect of Point-of-View

As a part of organizing an environment, subjects usually memorize its key features as a set of visual landmarks (Siegel and White, 1975). One task often studied in spatial cognition requires that the subject learn the landmarks’ spatial configuration in some canonical orientation (often upright). The subject then has to imagine that he is facing a new, specified orientation (defined by a certain viewing direction and a posture), and is finally asked to recall which landmark is located in a particular direction relative to his body axes.

Many variants of this task have been described in the research literature, and some have dealt particularly with the effect of point-of-view, Inside versus Outside the environment. When written narratives described an environment from a specific point-of-view, subjects naturally adopted that specified or directly implied point-of-view to build their spatial model (Inside: Franklin and Tversky, Experiment b, 1990; Outside: Bryant, Tversky and Franklin, Experiment 1, 1992). When the narrative specified no particular point-of-view, subjects selected a single point-of-view to comprehend the scene. Different points-of-view, however, could be adopted in different situations (Bryant et al., Experiment 4, 1992). In case of a narrative written in the third-person, the subjects were found to spontaneously adopt the Inside point-of-view of the narrator (Bryant et al., Experiment 2, 1992; Bryant, Tversky and Lanca, 1996). By contrast, Nigro and Neisser (1983) found that, when recollecting events, subjects tended to take an external point-of-view and described themselves in the scene, rather than adopt the actual point-of-view they had personally experienced. Once again, this demonstrated that no single point-of-view was preferred for all settings.

Although humans are easily able to adopt either point-of-view, their ability to perform a spatial task may depend on which one they choose. In Bryant et al.’s studies, subjects responded faster overall to questions with the Outside than with the Inside point-of-view. Huttenlocher and Presson, however, found the reverse (1979, Experiment 1). Bryant et al. reasoned that spatial relationships were easier to keep in mind when all the objects were concentrated in the viewer’s frontal view (Outside case) than when some were not in his field of vision (Inside case). By contrast, Huttenlocher and Presson argued that the directions of the objects were less distinguishable in the Outside case.
As mentioned, subjects tend to choose a single point-of-view when thinking about a scene in a static situation. When navigating, however, the motion of the viewer through the environment requires changes in point-of-view in order to assimilate the evolving spatial relationships. On Earth, for example, we are able to take alternatively an Inside and an Outside point-of-view without difficulty. This capacity may arise from the hierarchical structure of our spatial mental models (Davis, 1981; Stevens and Coupe, 1978). For instance, we usually consider that each room of a dwelling is embedded in a building embedded in a block, which is embedded in a city. Thus, we are used to continuously abandoning the Inside of an environment for its Outside, or the reverse. By contrast, the spatial structure of a space station and its surroundings are less neatly nested. The Outside of the station, for example, cannot be easily conceived as the Inside of another more global environment. Besides, the different modules astronauts have to navigate through are juxtaposed and not embedded within one another. This may explain in part why switching point-of-view in orbit seems to cause disorientation.

The point-of-view (Inside versus Outside) could be a determining factor for retrieving spatial information and processing it (McDonald and Pelligrino, 1993). Switching from one point-of-view to the other, therefore, may raise orientation difficulties in environments where there is no clearly nested spatial organization.

1.3 Effect of Rotation Mode

The previously described terrestrial spatial task, used to study the effect of point-of-view, has been used also by some researchers to assess the effect of rotation mode, with a 2D array. The new orientation could be specified either in terms of a rotated environment – “Imagine the array were rotated 90°” – or in terms of rotated viewer’s body – “Imagine you were rotated 90°”. In the Self-rotation mode, one imagines that one is moving relative to the environment. By contrast, in the Array-rotation mode, one imagines that the environment is moving relative to oneself. A good deal of research has sought to determine which of these two rotation modes, Array- or Self-, is easier to handle.
The kind of information asked of the subjects affects the relative difficulty of imagined Self- and Array-rotation (Huttenlocher and Presson, 1979; Presson, 1982; McMullen and Jolicoeur, 1990; Wraga et al., 1999). Subjects answered an “item” question – ‘Which picture is on your right?’ – better using Self-rotation (Presson, 1982). “Appearance” questions – ‘How does the whole array look?’ – were handled better with Array-rotation (Presson, 1982; Piaget and Inhelder, 1967). The same Array-rotation advantage holds for “position” questions – ‘Where is the flower?’ (Huttenlocher and Presson, 1979; Presson, 1982; McIntyre and Pick, 1976). Wraga, Creem and Proffitt (1999; 2000), however, argued that the Array-rotation advantage found for appearance and position questions may be based on an experimental bias. When the subject is supposed to mentally rotate the array to answer these two kinds of question, a circular translation of a single object would usually be sufficient to figure out the answer. The actual rotation of the entire array (‘holistic rotation’) would not be necessary. This would lead to the observed Array-rotation advantage since imagined translations of objects are thought to be easier to perform than imagined holistic rotations (Easton and Sholl, 1995; Rieser, 1989). Finally, if holistic rotation were actually performed, Self-rotation should lead to better performance than Array-rotation for any type of question.

The axis of imaginary rotation also affects performance. The gravitational axis is the most salient of the environment axes, and this feature, combined with properties of body axes, accounts for the subjects’ shorter response times to locate objects placed in directions other than right and left (e.g., Bryant, Tversky and Franklin, 1992). Most studies have used 2D arrays and have dealt with imagined rotations about the gravitational vertical (Huttenlocher and Presson, 1979; Amorim and Stuchi, 1997; Wraga and al., 2000). In these, Self-rotation tasks were faster and more accurate than Array-rotation tasks on questions like ‘Rotate 90°, which picture is on your right?’(item question). Only a few experiments looked at alternative axes of rotation. Carpenter and Proffitt (in press) found that the Self-rotation advantage disappeared when the imaginary rotational tasks were performed in the coronal and sagittal planes, but it was not clear whether this result was observed because the movements could not be physically performed, or because the geometric relationship between the viewer’s principal axis and array was not orthogonal. Creem, Wraga and Proffitt (2001) investigated further and showed that the orthogonality of the relationship between
the viewer’s principal axis and the array was the key factor leading to Self-rotation advantage. This finding holds for conflicting physical and imagined postures. Thus, Self-rotation seems to be easier than Array-rotation when the rotation is performed about one’s principal axis, regardless of gravity.

Wraga and al. (2000) have shown that the Self-rotation advantage (about the gravitational axis) holds for different arrays (a single-object or a four-object array). Does this result extend to a 3D array? Bryant, Franklin and Tversky used 3D arrays in many experiments involving imagined Self-rotation only. To our knowledge, a 3D array like the one used in the present experiment has not been used previously to study the effect of rotation mode in three dimensions. Less natural body-environment relations, in a 1-g but not a 0-g sense, are explored.

In weightlessness, problems of spatial orientation often arise from difficulties in visualizing and describing one rotation mode or the other. This has to do with the mental challenge of inter-relating different possible spatial reference frames. For example, the verticals of the modules on Mir were not always co-aligned and that made it difficult for astronauts to visualize one module while they were inside another. During Extra-Vehicular Activity (EVA), safely maneuvering the long robot arm fixed to the cargo-bay of the Shuttle requires the astronauts to specify in their verbal commands which reference frame they consider stationary and which they consider moving.

Humans exhibit a degree of flexibility as to which rotation mode is selected (Rock and Heimer, 1975). Both the Array- and the Self-rotation modes have been found to contribute to the human capacity to update an array across different viewpoints (Hummel, 1994; Simons and Wang, 1998; Tarr, 1995; Tarr and Pinker, 1989; Wraga et al., 2000). All these studies assumed that, whatever the experimental setting, every subject was able at any time to adopt successively one rotation mode or the other, as soon as he was instructed to change. It was never specifically verified, however, that the subjects followed those instructions faithfully. In certain situations, there may have been uncontrolled learning effects over time. The present experiment addresses this point: How does orientation training in one rotation mode affect subsequent performance in another one?
1.4 Overview of the Present Experiment

Because there is no way to simulate the ISS and 0-g on the ground, studies of human performance on three-dimensional orientation and navigation in space have been performed using virtual reality (Oman et al., 2000; 2002; Richards, 2000; Richards et al., 2002; Marquez, 2001; Aoki et al., 2000). The present experiment extends the findings of two earlier experiments (Oman et al., 2000; 2002; Richards, 2000; Richards et al., 2002) that used the same paradigm and helped define the current design. The present experiment explores the effects of point-of-view and of a change in rotation mode on a subject’s three-dimensional spatial abilities.

A first series of experiments (Oman et al., 2002) studied the ability of human subjects to learn to mentally orient relative to an array of six objects around them. Subjects were tested in a cubic room, either real or virtual, analogous to the interior of a space station module. They were able to learn to orient from two different viewing directions, in any roll orientation. The subject’s physical posture with respect to gravity had a negligible effect. Comparable results were obtained using a physical display and a head mounted virtual reality display of the type used in the present experiment. A second series of experiments (Richards, 2000; Richards et al., 2002) used a similar virtual cubic room and showed that teaching generic memorization strategies and providing practice with mental imagery could help subjects learn to orient and improve long-term spatial memory. In both studies, task performance correlated significantly with conventional paper-and-pencil test measures of ability to mentally rotate 2D and 3D objects (Eckstrom et al., 1976).

The environment used in the present experiment was a virtual 4-foot x 4-foot x 4-foot cubic room with black walls, identical to that used by Richards (2000). Visual landmarks – pictures of different familiar animals – were positioned at the center of each wall. The computer could rotate the whole room. This preserved the spatial relationships among the landmarks, but changed the orientation of the array relative to the subject from trial to trial. The orientation task was similar to that used by Richards et al. (2002), and is analogous to the task confronting an astronaut who becomes disoriented inside a six-sided node module on a space station. Astronauts presumably recognize a few familiar landmarks in the
environment around them, infer their orientation by transforming their mental model of the environment, and then confirm that it predicts the direction to other landmarks. In our experiment, we presented the subjects with a pair of visual landmarks (one in front, and a second below) and asked them to infer their orientation, and deduce the relative direction to a third unseen “target-landmark”. (In earlier experiments, Oman et al. (2002) told their subjects what their orientation was relative to a “baseline” (canonical) orientation, rather than asking them to deduce it). The subjects’ orientation was kept constant for the first few trials so they could memorize the spatial arrangement of the landmarks as seen from a baseline orientation, and develop a spatial mental model. The subjects’ orientation was then changed, so they could learn to do the task from arbitrary (modulo 90 degrees) orientations relative to the environment. Their response time and pointing accuracy were monitored during the learning process.

Our principal scientific objective in the present experiments was to compare the relative difficulty of learning to spatially transform a 6-landmark array using a Self-rotation mode as compared to an Array-rotation mode. A second objective was to see whether subjects could switch modes rapidly if we required them to. Hence, the core of the experiment consisted of a “training phase”, during which subjects learned to orient using one rotation mode, followed by a “testing phase”, where they were asked to switch rotation mode. They were retested with the same landmark array and we referred to this second sequence of trials as the “testing” phase, because the subjects did not have visual feedback confirmation of target-landmark direction, as they did during the training phase. A third experimental objective was to assess the effect of point-of-view, Inside or Outside, from which the environment was perceived. Since training required naïve subjects, we could not employ a crossover experiment design. Instead, we blocked the design by point-of view, and assigned two different groups of subjects to each one. Each of the four groups began training using the Self- or Array-rotation mode, and changed mode from the training to the testing phase. We balanced the four groups of subjects for mental rotation abilities using the Cube Rotation Test (Eckstrom, 1976), since in prior studies this paper-and-pencil test score correlated significantly with subject performance.
2. **METHOD**

2.1 **Subjects**

Forty-two subjects, most of them MIT students, were tested. (As explained in section 3.1, thirteen subjects were subsequently eliminated based on non-compliance with instructions, or outlier performance). Subjects denied any positive history of visual, vestibular or auditory disease in a screening questionnaire. The four experimental groups were balanced according to the subjects’ pre-experiment scores on the Cube Rotation Test, as noted earlier. The mean Cube Rotation Test scores varied between groups by no more than 3.5% (the average score by group varied between 27.2 and 28.1). There were approximately equal numbers of males and females in each group.

2.2 **Virtual Reality Display System**

The virtual display system consisted of a head-mounted display (HMD) (Kaiser Proview 80, 65 deg.* 50 deg. field-of-view per eye, 640*480*3 resolution, 100% binocular overlap). The visual scene was rendered in color stereo by a graphics-accelerator-equipped PC workstation (30 Hz. average scene update rate). The workstation stabilized the HMD image using angular orientation data provided by an inertial head tracking system (IS-600 Mark 2, Intersense), operating in 3D (rotational) tracking mode. Additional details are available in Richards (2000). Subjects sat erect in a chair throughout the experiment.
2.3 Virtual Environment

The 3D environment was intentionally complex, so that learning would not be instantaneous and the time course of performance during learning could be traced. The environment simulated a cubic room, four-feet-long on each side with small color 2D landmark icons centered on each of its black walls. Each icon consisted of four identical copies of an easily recognizable and nameable animal picture (lion, giraffe, turtle, butterfly, fish, or parrot), as seen in Figure 2.1. The four identical pictures were oriented in different directions to create a symmetric icon, so that the subjects could not use pointer rules (e.g., ‘the head of the butterfly points towards the fish’) to help them specify or locate any other icon.

The six-icon array could be rotated as a whole in increments of 90° about every principal axis by the computer software – a different such rotation was used in each trial. The relative spatial relationships among the six icons remained constant throughout the experiment. Images of the room as seen from the Inside and Outside points-of-view are shown in Figure 2.2a and Figure 2.2b. The outline of the walls of the virtual room was always displayed. Figures 2.2a and 2.2b show the array in the “baseline” orientation used during the beginning of the training phase, where the orientation of the array was kept constant so the subjects could learn it. Subjects tested Inside the virtual environment could not see the icon immediately behind them. For subjects tested Outside, the nearest wall was
removed since it would have obstructed the view of landmarks inside. Thus, subjects could only see five icons at most. The unseen icon, placed on the inside of the wall that was located inside-behind or outside-nearest in the baseline orientation, is shown in the Figure 2.2b inset.

![Figure 2.2a](image)

**Figure 2.2a**
Inside View of the Environment in the Baseline Orientation.

![Figure 2.2b](image)

**Figure 2.2b**
Outside View of the Environment in the Baseline Orientation.
2.4 Design

Both point-of-view and rotation mode effects were potentially of interest, and their cross- (interaction) effects were studied in this between-groups design. (Naïve subjects had to be tested to be able to study learning trends, so the subjects could not serve as their own control.). It was impractical to test the twelve large groups that would have been needed to define all the potential effects. Informal pilot experiments suggested that changing point-of-view from Inside to Outside was relatively easy, but it was hard (sometimes almost impossible) to change both point-of-view and rotation mode at the same time. Thus, as shown in Figure 2.3, we blocked the experiment by point-of-view (Inside or Outside), and manipulated rotation mode (Array- or Self-) between the training and testing phase. As a result, only four groups were tested, leaving intergroup artifacts and cross-effects between factors possible. The four experimental groups were:

1. Inside/Array-rotation training – Self-rotation testing (IAS)
2. Inside/Self-rotation training – Array-rotation testing (ISA)
3. Outside/Array-rotation training – Self-rotation testing (OAS)
4. Outside/Self-rotation training – Array-rotation testing (OSA).

Each of the four experimental groups took one single point-of-view, Inside or Outside the environment, and successively adopted the two rotation modes (Array- and Self-), one mode in training, the other in testing. (Figure 2.3).
Figure 2.3
Experimental Design.
2.5 Procedure

2.5.1 Initial Cube Rotation Testing and Instructions to Subjects

At the beginning of each experimental session, the subject completed the Cube Rotation Test, was assigned to a group, and then was asked to read a first set of written instructions, prepared as a slide presentation (Powerpoint) and presented on a computer screen (Appendix B). Since each group began with a different point-of-view and training rotation mode, the instructions were tailored for each group’s task, as detailed in the next section. Special attention was given to terminology and to pictures illustrating the rotation (of themselves or of the array) that the subjects were to imagine. For the Self-rotation mode, subjects were taught to imagine the room as stationary and that they were rotating, whereas for Array-rotation, they were taught to imagine themselves as stationary, and that the room was rotating. The timeline for each trial and for the first, training phase was described. However, subjects were not told that they would be asked to change rotation mode until after the training phase was completed. Before the training phase began, subjects donned the HMD and performed three trials in a practice environment different from the experimental one. This allowed the subject to become accustomed to the head-mounted display and the basic nature of the task.

2.5.2 Procedure for Training Phase

The training phase was divided into three parts, as shown in Figure 2.5 below. The first two were considered “environmental familiarization”, and allowed the subject to learn the relative locations of the icons while the orientation of the environment was held fixed (first 4 trials), or when the viewing direction was fixed (next 8 trials). Thereafter, the orientation of the environment was varied randomly (24 trials). During the first four trials, the experimenter asked the subjects to describe the room orally, and corrected any mistakes. This staged approach to training had been successfully employed by Richards et al. (2002).
The order of orientations in the twenty-four random trials was balanced over icons in the front position and target directions relative to the body. Randomization of orientation presentations has recently been shown to enhance learning of this type of spatial task, relative to a completely blocked presentation of orientations (Tubre and Shebilske, personal communication), and is favored by training theorists (e.g., Bjork, 1994).

During the first four trials, the array was oriented in a fixed “baseline” orientation (as shown in Figure 2.2b). Depending on their training-phase rotation-mode assignment, subjects then had to imagine that they or the environment could rotate (adopting a Self- or an Array-rotation mode) relative to the baseline orientation they had memorized in the first four trials. Once subjects were familiar with the arrangement of the icons, seeing any two icons allowed them, in principle, to deduce their relative orientation to the environment, regardless of whether they were Inside or Outside it. In each trial, first the target-icon that the subject was to locate was shown for 2 seconds at the middle of the screen. Next, the icons in front and below the subject were shown for 3 seconds. The subject was asked to deduce his relative orientation, using the rotation mode that he had been instructed to handle, and to indicate the relative direction to the unseen target-icon relative to body-axes.
by pressing the “Above”, “Behind”, “Right” or “Left” buttons on a masked computer keyboard, mounted in the subject’s transverse plane. “Behind” was not appropriate for subjects indicating the near front wall in the Outside condition, and “Front” was used instead. After the front/below icons appeared, the subject had 9 seconds to respond. The front/below icons were visible only for the first 3 seconds of this interval. Then, the entire environment (all five icons) was shown for an additional 8 seconds so that the subject could verify his answer and reinforce his spatial knowledge of the array. This feedback, essential for learning and provided during the training phase, was omitted during testing.

After 24 randomized trials were completed, subjects were asked to complete a first questionnaire (Appendix C). The purpose of the questionnaire was to verify which transformation type the subject had actually used, and to learn more about the way the subjects learned to do the task – information that possibly could be used to improve the design of future experiments. The whole questionnaire consisted of approximately a dozen open-ended and numerically-scaled questions.

2.5.3 Procedure for Testing Phase

At the end of the training phase, the subjects were given a short break and then asked to read a second set of instructions describing the testing phase. They were told to change rotation mode, and that the task and environment would be the same as before. They were told also that, since the same environment would be used, they would not have the initial 12 familiarization trials, and that they would not be able to see the entire array at the end of each trial to confirm their answer. The instructions described the new rotation mode and the experimenter offered to answer any questions on the difference between the two modes. Testing was then conducted, using the exact same sequence of 24 randomized trials as in the training phase. Finally, the subjects completed a second questionnaire, again intended to identify which type of rotation mode they had actually used.

In summary, the entire experiment consisted of 36 trials in the training phase (12 familiarization and 24 standard), followed by 24 standard trials in the testing phase (as
schematized previously in Figure 2.5). Data from the first 12 familiarization trials of the training phase – where the subject could be coached – was expected to be relatively variable, and was not formally analyzed. For purposes of analysis, the 24 standard trials common to each phase were divided into 3 sets of 8 (Set 1, 2 and 3 in training, Set 4, 5 and 6 in testing). In our design, since two icons were presented as cues from which subjects would infer their orientation, subjects had a 75% chance of incorrectly predicting the target-icon direction – not 80% as in our earlier study (Oman et al., 2000). As a result, we averaged data across sets of 8 trials, rather than 4, to smooth the effects of quantization of the %-Incorrect data. Averaging across a larger number of trials (e.g., 12) would have given us too few averages to observe the time course of learning.

2.6 Data Analysis

In contrast to Bryant, Tversky and Lanca’s study of terrestrial spatial transformations, which focused only on response time as the experimental measure, we used %-Incorrect, averaged over each set or each phase, as the principal measure of spatial ability. The subjects made many errors and part of our goal was to track the improvement in accuracy over the experiment. The quantized %-Incorrect measure was imperfectly matched to statistical tests requiring normally distributed data. Therefore we used nonparametric Friedman, Sign and Kruskal-Wallis Tests to assess differences in %-Incorrect within and contrasts between groups. Learning trends were analyzed via the Page Test, which detects monotonic decreasing or increasing trends. On the other hand, for the response time data analysis we used repeated measures ANOVA, and paired and two-sample t-tests to evaluate differences in mean RT within groups, and contrasts between groups. Statistical analysis was performed using (Systat v10.0, SPSS, Inc.; StatXact-4, Cytel, Inc.) packages.
3. RESULTS

3.1 Subjects Excluded from Statistical Analysis

A subject was excluded if he clearly was unable to follow the instructions or his response was identified as a clear outlier. According to their answers to the questionnaire, ten subjects did not use the strategy we had prescribed for them, but used an “avatar” strategy instead, and imagined themselves to be represented by a token figure within the room. While this was an interesting strategy and should be researched further, it was not the focus of the present experiments, so we eliminated those subjects from the analysis. Also, three subjects showed outlier performance in the sense that they were able to respond to fewer than five trials in the testing phase. One clearly lacked motivation, the other two became quickly tired and discouraged. They were, therefore, also omitted from the analysis. Table 3.1 shows the composition of each of the four experimental groups before and after applying exclusion criteria.
3.2 Experimental Variables

Subjects’ performance on the spatial task was measured by %-Incorrect and by mean Response Time (RT). Although we expected subjects to take more time to respond when they were uncertain of their answer, mean RT did not vary as much as %-Incorrect. Most of the observed significant effects concerned %-Incorrect, while the corresponding effects on mean RT were attenuated or insignificant. Because the three-dimensional experimental task was cognitively demanding, it is possible that the subjects concentrated on improving their accuracy, and lacked the mastery of the task that speeding up their performance might have required. For this reason, our analysis focused on the %-Incorrect measurements.

Each of the four experimental groups was defined uniquely by two variables: the point-of-view (Inside or Outside) from which they perceived the environment during the entire experiment, and the rotation mode sequence they used, i.e., one mode during training (Self- or Array-rotation) and the other during testing (respectively Array- or Self-). We studied particularly the effects of two within-subject variables, the direction to the target relative to the subject’s body and direction to the target relative to gravity (defined in the baseline orientation).
3.3 Main Results

There was no significant effect of gender on %-Incorrect and it did not interact significantly with point-of-view, rotation mode, or group.

Figure 3.1 shows the %-Incorrect by group averaged across all trials in the training and the testing phases.

![Figure 3.1](image)

Figure 3.1
%-Incorrect by Group and by Phase.

Of the two Inside groups, Group IAS made significantly fewer errors than Group ISA both in the training (Kruskal-Wallis, Kruskal-Wallis Parameter KWP=5.16, df=1, p=0.023) and testing phase (Kruskal-Wallis, KWP=4.43, df=1, p=0.035). In each of the two Inside groups, there was a significant monotonic decrease in %-Incorrect over all 6 sets of trials (Page Test for Group IAS, pa(x)=-2.08, p=0.014; for Group ISA, pa(x)=-1.71,
p=0.033). For Group ISA, this trend was also significant for the 3 testing sets alone (Page Test, pa(x)=-1.9, p=0.025).

Transfer of learning, a within-group effect, occurs when previous learning facilitates new learning on a different task (Cormier and Hagman, 1987). In this experiment, we could evaluate the possible transfer of learning from one rotation mode to the other between the training and the testing phase. When the data from all 3 sets of each phase (training vs. testing) were compared, we found no significant difference between the phases for either of the Inside groups: we could not show an effect of manipulating the rotation mode. However, when the data from the first set of each phase were excluded – this tactic presumably would suppress any “new rotation mode” effect of suddenly being confronted with a new rotation mode to learn – the difference between the training (%Incorrect=0.39) and testing (%-Incorrect=0.23) phase for Group IAS approached but did not quite achieve significance (One-sided Sign Test, p=0.06). There were fewer errors in the testing phase, as we would expect from the advantage provided by prior efficient training, even in a different rotation mode. Thus, though the data suggested that a “new mode” effect may exist, it could not be formally demonstrated. Also, it is important to consider the context of the testing trials: the subjects did not have the 12 familiarization trials at the beginning of the testing phase that they had at the beginning of training. That they were apparently able to adopt a second mode without a statistically significant decrease in performance suggests that most subjects’ performance had asymptoted during training, and that performance was not greatly decreased by the change in rotation mode imposed during testing.

To assess any “switch” effect of changing abruptly from one rotation mode to the other, we examined the difference in %-Incorrect between the last training set (Set 3) and the first testing set (Set 4). Group ISA’s subjects made more (but not quite significantly more) errors in Set 4 (mean=0.59) than in Set 3 (mean=0.46) (One-sided Sign Test, p=0.06). As with the “new mode” effect, the data suggested the possibility of a “switch” effect, but it was not quite statistically significant.

The two other groups (Group OSA and Group OAS) were provided with an Outside point-of-view of the environment. We found no significant difference in %-Incorrect comparing these two groups, in either the training or the testing phase. The
performance of groups clearly improved slightly, at least during the training phase. For Group OSA, the %-Incorrect decreased monotonically and significantly over all 6 sets of trials (training through testing) (Page Test, $p(a(x))=-2.53$, $p=0.003$). The 3 training sets for Group OSA showed a significant decreasing trend (Page Test, $p(a(x))=-2.50$, $p=0.004$).

With respect to transfer of learning, we could not show a significant effect of manipulation of the rotation mode between the training and testing phase for the Outside groups, as for the Inside groups. Group OSA’s subjects, however, performed their best in the testing phase (their %-Incorrect was equal to 0.37 in testing vs. 0.48 in training) and this difference was almost significant by a Sign test (One-sided Sign Test, $p=0.06$). Group OAS’s subjects made more errors in Set 4 (%-Incorrect=0.56) than in Set 3 (%-Incorrect=0.37), and this difference was also almost significant (One-sided Sign Test, $p=0.06$).

Comparing individual groups to examine the effects of point-of-view during testing: during Self-rotation testing, those subjects tested Inside the virtual environment made significantly fewer errors than those tested Outside (Kruskal-Wallis Test, Group OAS vs. Group IAS, KWP=4.9, df=1, $p=0.027$). However, no such effect of point-of-view was found during Array-rotation testing.

Looking across combined groups at the effects of rotation mode during training: the combined groups who trained using an Array-rotation strategy made significantly fewer errors than those who used a Self-rotation training (Kruskal-Wallis, Groups OAS and IAS vs. Groups OSA and ISA, KWP=3.82, df=1, $p=0.05$).

### 3.4 Evidence for Similarly Skilled Groups

Four arguments suggest that the observed effects were not due to random sampling effects, which might have produced differences in inherent spatial abilities of the four groups of subjects. First, the design explicitly tried, in advance, to suppress such effects. Subjects were selected from a relatively homogeneous population of university students and were assigned to groups so that the mean score on the Cube Rotation Test would be almost identical for all groups. As noted earlier, Cube Rotation Test scores for the groups differed...
by only 3.5%. Second, if there were significantly more subjects with unusually strong or weak skills in one of the groups, we would expect the variance in %-Incorrect of that group to be greater. In fact, the groups showed similar variances. Third, %-Incorrect performance of all four experimental groups were significantly concordant across the 6 experimental sets (Friedman, FP=20.03, df=5, p=0.001) and the rank pattern was consistent with the general learning trend observed within phases, as shown in Figure 3.2 below. In the Friedman test, each subject’s scores (%-Incorrect) over the six sets are ranked in order of size. The Friedman test assesses the degree to which the sums of those individual ranks for the various sets are different from what would be expected if the ranks were randomly assigned. Fourth, there were no significant differences in performance (by Kruskal-Wallis Test) between any pair of groups in the final training set.

Figure 3.2
Significant Friedman Ranking of Sets by %-Incorrect for all four Groups.
3.5 Effect of Training for each Transformation Type

In view of the above arguments, we made the assumption that the observed effects are consequences of experimental conditions, and not of inherent differences among the four different groups of subjects. Under this assumption, we were able to compare results obtained with the same transformation type, i.e., the same point-of-view and rotation mode, even though the data came from two different subject groups. For example, one group adopted the given rotation mode during training (with feedback) and the other adopted it during testing (without feedback). Figure 3.3 presents the same data as Figure 3.1, but by mental model transformation type and not by subject group:

![Figure 3.3](image-url)

Figure 3.3
%-Incorrect by Transformation Type and by Phase.
Group IAS’s subjects made significantly fewer errors when using the Self-rotation mode in testing than Group ISA’s subjects did using the same mode in training, with no prior experience (Kruskal-Wallis, KWP=7.78, df=1, p=0.005). An opposite effect was observed for the Array-rotation mode: Group ISA, using the Array-rotation mode in testing, made more errors than Group IAS did in training with the same mode (Kruskal-Wallis, KWP=6.65, df=1, p=0.01). However, that result was significant only for the very first set of each phase, and could therefore be confounded with two separate potential surprise effects if they exist – one for entering the experiment, and the other for changing to a new rotation mode. No significant differences between phases or sets were found for the two Outside groups.

Figure 3.4 and Figure 3.5 illustrate the previous results in more detail. Figure 3.4 shows the Kruskal-Wallis ranking of the four possible transformation types in the training phase, for the first set (KWP=10.09, df=3, p=0.018). (No significant differences by Kruskal-Wallis were found for Set2, Set3, or all the sets together). This suggests that, although the four groups started the experiment delivering different %-Incorrect in Set 1, their performances at the end of the training in Set 3 were similar because their different learning rates compensated for the initial differences. Moreover, this ranking indicates also that Array-rotation is easy Inside only initially – it has the best, i.e., lower rank – perhaps because it is somehow more natural.
Figure 3.4

% Incorrect Comparisons by Transformation Type in the Training Phase.

(Significant Kruskall-Wallis Test for Set 1)

Figure 3.5 shows the Kruskal-Wallis ranking of the four transformation types in the testing phase, which was significant for all three sets taken together (KWP=8.25, df=3, p=0.041). Note that comparing training with testing, the Inside transformation types exchanged ranks: During Inside training, Array-rotation gave best performance, and Self-rotation training was worst. In the testing phase, their ranks reversed, so that Self-rotation testing now yielded best performance. The Outside transformation types occupied the middle ranks, in both training and testing. This suggests that for the Outside subjects, the rotation mode given first had little net effect on the final % Incorrect. Apart from point-of-view, the Array-rotation mode ranked better (i.e., lower) in the training than in the testing phase. The opposite result was observed for the Self-rotation mode, which ranked better in testing than in training.
3.6 Effect of Target Direction Relative to Body Axes

Bryant, Franklin, or Tversky et al. frequently studied orientation using narratively described or real object arrays in terrestrial settings, but focused on response time as a function of body posture. When the observer (whether actually observing or imagining) was upright, he located objects fastest if they were Above or Below him. When supine, he was fastest when the targets were Ahead or Behind. Finding targets that were Left or Right of him was slowest among the three target axes for both postures.

A comparable effect of relative target direction on %-Incorrect was seen in the present experiment results, even though we used a slightly different form of question. We measured direction accuracy, and asked for the location of a given icon, not for the identity of the icon located in a given location, as in the Tversky lab’s experiments. The Friedman
ranking of relative target directions showed significant concordance over the subjects of the four groups, in training as in testing (Friedman, training, FP=12.75, df=3, p=0.05; testing, FP=15.78, df=3, p=0.001). The Above/Behind (A/B) and Right/Left (R/L) target direction categories stood out: subjects located icons that were Above or Behind them with more consistent accuracy than icons that were on their Right or Left, which agrees with the results of Bryant, Franklin and Tversky, and Richards (2000) (%-Incorrect averaged across all training trials for A/B targets=0.39, for R/L targets=0.52; %-Incorrect averaged across all testing trials for A/B targets=0.30, for R/L targets=0.49).

Performance improvement across the successive sets of the experiment depended on group and relative target direction. The %-Incorrect for the A/B targets showed a significant decreasing trend (improvement in accuracy) over the 6 sets of both phases, but only for the Outside groups (Page Test, Group OSA, pa(x)=-2.55, p=0.002; Group OAS, pa(x)=-1.77, p=0.026). By contrast, the R/L targets showed a significant decreasing trend for the Inside groups (Page Test, Group IAS, pa(x)=-2.78, p<0.001; Group ISA, pa(x)=-1.76, p=0.023).

If we compare accuracy by direction category in the training phase with that in the testing phase to study transfer of learning, Group OSA was found to improve significantly in the testing phase for A/B targets (One-sided Sign Test, p=0.016), and Group OAS showed a non-significant tendency in the same sense. For the R/L targets, Group IAS improved significantly from training to testing (One-sided Sign Test, p=0.008). Group OAS, however, had almost significantly and surprisingly greater R/L accuracy in training than in testing (One-sided Sign Test, p=0.06).

Significant differences among groups were found for both target categories in the testing phase as seen in Table 3.2 (but not in the training phase). Separately, Group OAS performed best and Group ISA worst on A/B targets; Group IAS performed best and Group OAS worst on R/L targets.
### Table 3.2
Relative Target Direction Comparisons by Groups (Kruskal-Wallis Test) in the Testing Phase.

<table>
<thead>
<tr>
<th>Group</th>
<th>%-Incorrect Ranks by Relative Target Direction</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Above/Behind</td>
<td>Right/Left</td>
</tr>
<tr>
<td>OAS</td>
<td>86</td>
<td>133</td>
</tr>
<tr>
<td>OSA</td>
<td>100</td>
<td>115.5</td>
</tr>
<tr>
<td>IAS</td>
<td>95</td>
<td>63.5</td>
</tr>
<tr>
<td>ISA</td>
<td>154</td>
<td>123</td>
</tr>
</tbody>
</table>

KWP=8.2, df=3, p=0.042
KWP=8.03, df=3, p=0.045

#### 3.7 Effect of Target Direction Relative to Gravity

In the baseline orientation, the ceiling-floor icons were “turtle”-“lion”; east-west, “fish”-“parrot”; and north-south, “butterfly”-“giraffe”. These pairs could be assumed to define the axes of the subject’s cognitive model of the array relative to gravity. Arguably, the natural properties of the environment make the gravitational (ceiling-floor) axis the most efficient spatial organizer because of its asymmetry (Braine et al., 1987; Jolicoeur, 1985; Bryant et al., 1996). We might expect this salience to be replicated in our experimental environment, so that “turtle” and “lion” – the baseline ceiling and floor icons – may be easier to track, regardless of imagined relative body orientation. No group, however, showed a significant main effect of environmental (as opposed to body) axis per se, or any cross-effect of environmental axis with point-of-view or rotation mode. However, the “turtle” icon (baseline ceiling, and on the gravitational axis) was the one most accurately located among the 6 icons. The Friedman Test indicated significant concordance among the subjects in the overall ranking of icons in training (FP=34.99, df=5, p<0.001), as in testing (FP=13.86,
df=5, p=0.017), the best rank being attributed to the “turtle” icon (baseline ceiling) in both phases. If we compare accuracy in locating the two baseline orientation gravitational axis icons, the ceiling (“turtle”) had the advantage over the floor (“lion”) in both phases (Two-Sided Sign Test, training, p=0.001, %-Incorrect for ceiling icon=0.23, for floor icon=0.51; testing, p=0.021, %-Incorrect for ceiling icon=0.29, for floor icon=0.48).
4. DISCUSSION

The four groups of subjects performed the experimental orientation task with the same (Inside or Outside) point-of-view in training and testing, but a different rotation mode in each of the two phases. The point-of-view was imposed by the display and 26 of 29 subjects reported via questionnaire that they never adopted (mentally) a point-of-view different from the one instructed. By contrast, the rotation mode, which is based on an internal choice and often unconsciously executed, could apparently not be imposed as easily. The subject was instructed (but not forced) to adopt one rotation mode in training, and then to change and adopt the other mode in testing. A number of subjects could not carry out our instructions consistently, which led us to exclude 13 of the 42 from the final analysis. It is noteworthy that more than half of subjects excluded from the analysis because they used an “avatar” strategy had been previously assigned to Group IAS. Perhaps Array-rotation training experience led the subjects to apply a similar array-rotation cognitive process to a token figure of themselves in the subsequent testing phase because they found it easier, instead of imagining a Self-rotation of themselves.

4.1 Differences Among Individuals

Although the groups were balanced for natural aptitude by using subjects’ Cube Rotation Test scores, we looked for but found no significant correlation between Cube Rotation scores and a subject’s ability to locate a target-icon or to improve. In our earlier studies (Oman et al., 2000; 2002; Richards, 2000; Richards et al., 2002), Cube Rotation Test
scores proved to be reliable predictors of group performance. In the present experiment, however, it is possible that the subjects’ performance data were noisier, since our design included more variables (point-of-view and rotation mode). This may prevent us from finding any correlation between the test scores and subjects’ accuracy on our task. We noted that the Cube Rotation Test was completed at the very beginning of all experiment sessions. Arguably the nature of the cube rotation task implied an Array-rotation mode with an Outside point-of-view, so subjects had practice with this transformation type. That additional experience could have influenced their subsequent performance and might have contributed to the Array-rotation advantage found during the training phase of the experiment. Although no correlation could be demonstrated between task performance measures and Cube Rotation Test scores, we found no hint in our statistical analysis that the observed effects could be due to subtle peculiarities in the groups’ sampling. As a result, we believe that they are due exclusively to the experimental treatment and part of our interpretation is based on this inference.

4.2 Learning Within Training and Testing

The spatial task prescribed for subjects in this experiment was difficult. First, the subjects had to keep in mind the three-dimensional array in its baseline orientation. Second, they had to imagine that either they themselves or the array could rotate about multiple axes. Third, the subjects were not told their orientation relative to the virtual room (as in Oman et al., 2002), but had to infer it from the two icons presented in front and below them. As a result, subjects made errors, particularly in the early stages of training. Despite these obstacles, most subjects learned to handle their assigned transformation type to do the task within 36 trials (including the familiarization) in the first, training phase of the experiment, confirming the conclusion reached by Richards et al. (2002) for the Inside-Self-rotation case, and extending it to the other cases. All groups reduced their % Incorrect to about 35% in the last training set (where random choice would have given 75% Incorrect). This final performance was slightly below earlier results obtained for the Inside-Self rotation case using the same experimental environment (Richards, 2000; Richards et al., 2002). The orientation
task was harder in the present experiment because the subjects’ spatial thinking were more constrained. They were required to perform mental rotation only, and were not free to use whatever strategy they desired.

There was, for every group, a general decrease in %-Incorrect in the training phase, most notably for Group OSA. The same was true of the testing phase, where the decrease in %-Incorrect was most notable in Group IAS. In the questionnaire, 12 out of 29 subjects claimed that they did not miss the feedback much when it was withdrawn during testing. This suggests once again that the subjects had already learned the environment before entering testing and that the switch to the other rotation mode was graceful.

4.3 Change in Rotation Mode and Transfer of Learning

In no group did we observe a dramatic increase in the %-Incorrect when the subjects were asked suddenly (between Set 3 and Set 4) to change rotation mode. Nevertheless, the increase was most important for Group OAS and Group ISA (as shown by Sign Test between Set 3 and 4 in section 3.3). This suggests that most subjects were able to change rotation mode between the training and the testing phase with relative ease. Nevertheless, 10 of 29 subjects considered that changing rotation mode in the middle of the experiment was hard. Some subjects (18 out of 29, according to their answer to the question: “Did you switch rotation mode within the testing phase?”) managed to use the prescribed second rotation mode as soon as the testing phase began, and continued to use it throughout testing. The remaining 11 – 9 from Group OSA and Group IAS – said they changed rotation mode more gradually, and sometimes used the wrong one in testing. This could help explain the relative smooth transition observed between the training and testing phases for these two groups. (Page Tests for Group OSA and Group IAS across all 6 sets were significant,

---

\[a\] The question was “Did you miss having the feedback?”. On a scale ranking from (0=“By no means”) to (10=“Definitely”), 12 subjects gave ratings \(\leq 3\).

\[b\] “How would you describe the change of rotation mode that was required between the first and the second phase?”. On a scale ranking from (0=“Trivial”) to (10=“Impossible”), 10 subjects gave ratings \(\geq 6\).
showing that improving trends continued across the change in rotation mode. Sign Tests between Set 3 and Set 4 were not).

Since changing rotation mode did not seem to decrease the subjects’ performance during testing in a major way, even without 12 familiarization trials and without feedback, it is reasonable to hypothesize that transfer of learning from the training to the testing phase occurred: learning to orient using one rotation mode in training may help subjects perform in testing using the other mode (while the point-of-view was preserved). The subjects of Group IAS and Group OSA achieved a relatively higher level of accuracy in the second mode (than in the first), which could suggest that they may “learn how to learn”. This transfer may explain why we observed learning in the testing phase, even without feedback (as mentioned in the previous section). There was no such transfer in evidence, however, for the two other groups, particularly for Group OAS. All this was confirmed by the subjects themselves in the questionnaire (Table 4.1). Overall, almost twice as many subjects reported finding the testing phase rather than the training phase easier.

<table>
<thead>
<tr>
<th>Group</th>
<th>Number of Subjects who found the Training Phase Easier</th>
<th>Number of Subjects who found the Testing Phase Easier</th>
<th>Number of Subjects who had no Preference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group OAS</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Group OSA</td>
<td>3</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>Group IAS</td>
<td>2</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Group ISA</td>
<td>2</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>All Groups</td>
<td>10 (34%)</td>
<td>17 (59%)</td>
<td>2 (7%)</td>
</tr>
</tbody>
</table>

Table 4.1
Responses to the Question: “Which phase did you find easier?”.
4.4 Training Effect

We believe that the four experimental groups had similar average spatial skills. Hence, we estimated the effect of training by comparing the accuracy achieved by two different groups with the same transformation type, one using it during training, and the other using it during testing (Figure 3.3). That comparison measures the improvement provided by training in the other rotation mode. We found a strong training effect when the subjects were given an Inside point-of-view. The Inside Self-rotation mode showed substantially better accuracy as a result of Inside Array-rotation training. The effect is particularly interesting because there was no feedback during the testing phase, during which the improvement was observed. The Array-rotation mode showed nearly the reverse result – lower, but not significantly lower accuracy over all 3 sets – after Self-rotation training than they exhibited without it. That Self-rotation training appeared to interfere with learning.

No clear training effect was found when the subjects were asked to use an Outside point-of-view. Presumably this indicates that the Outside point-of-view is difficult, particularly in Self-rotation mode as suggested by some results mentioned previously.

4.5 Point-of-View and Rotation Mode Effects

In the training phase with naïve subjects, we found that Array-rotation conferred greater accuracy than Self-rotation. This finding was particularly strong for the groups with an Inside point-of-view of the environment, which may be surprising since in everyday life, we presumably do Self-rotation (Huttenlocher and Presson, 1979; Bryant et al., 1996) and it is usually impossible to physically rotate the environment around us. Yet this argument pertains to rotation about the body’s principal axis only and cannot be applied to our experiment, where Self-rotation had to be imagined about every axis. Our earlier study (Oman et al., 2002) using a similar virtual environment specified no strategy to do the orientation task and subjects were found to adopt the Self-rotation mode more often than
the Array-rotation mode. However, no firm conclusion can be drawn as to subject intrinsic preference for a specific rotation mode from the earlier experiments, since subjects may have been biased to employ Self-rotation, either through instructions (‘Imagine you have been tilted left-shoulder down.’, Richards, 2000; Richards et al., 2002) or through the procedure used to specify their orientation (in terms of imaginary body roll orientation with a clock hand, Oman et al., 2000; 2002). In the experiments reported here, administration of the Cube Rotation Test immediately prior to training may have introduced a bias in favor of Array-rotation strategies.

The advantage we observed for the Array-rotation mode agrees with results of other studies using the same type of “position” question – e.g., “Where are the parrots located?” Wraga, Creem and Proffitt (1999; 2000), however, argued that this finding may be biased since the subjects in the previously mentioned experiments would not really rotate the entire array in the Array-rotation mode. The introduction section provides further details. In the present experiment though, the subjects could not locate the target-icon by transforming it alone. They had also to transform at least two other icons at the same time – those presented in front and below which served as cues from which they could infer the new orientation of the room. These three icons were in different planes and the subjects would have to rotate the set of three together in order to place the target-icon correctly. Thus, Wraga, Creem and Proffitt’s argument advocating a “hidden” Self-rotation advantage does not apply to this experiment.

The array was three-dimensional and subjects could mentally rotate it or themselves about more than one axis of rotation. As a consequence, in most trials, the orientation task required mental rotations about other axes than the viewer principal axis only. This was shown to lead to an Array-rotation advantage (as mentioned in the Introduction section). This may explain why our subjects made fewer errors in the Array-rotation mode than in the Self-rotation mode. In any case, the Self- over Array-rotation advantage reported by Wraga et al. (2000) for certain types of array did not generalize to our more complex and realistic 3D array.
In the testing phase with a Self-rotation mode, the subjects who were Inside the virtual room made significantly fewer errors than those who were Outside (section 3.3). This finding supports Huttenlocher and Presson’s, rather than Bryant, Tversky and Franklin’s explanation for the effect of point-of-view. Huttenlocher argued that the target directions were more separated when inside, since the targets are located in direction separated from every other by the same, clear and uncluttered 90°. In the Outside case, the six possible directions of the icons were concentrated in the subjects’ narrow field-of-view, which was more confusing.

4.6 Effects of Relative Target Direction

We saw a strong effect of target direction relative to the body, as in earlier studies (Bryant et al., 1992; 1999; Franklin et al., 1990; 1992; Richards, 2000). This may be related to the fact that the subjects had to indicate the relative direction to the target-icon in body-axes and there appeared to be natural response anisotropies related to body symmetries. More errors were made when the target-icon was on the subject’s Right or Left than when it was Above or Behind him. To explain this, we must first note that the Right/Left targets correspond to a weak body asymmetry, whereas the asymmetries of the body’s head/feet and front/behind axes are strong (Franklin et al., 1990; Bryant et al., 1992). Second, in the questionnaire, 19 out of 29 subjects reported that they had memorized opposite pairs of icons on their own initiative. This strategy made finding the Above/Behind targets easier, since the subject was initially shown the icons ahead and below to deduce his orientation relative to the environment. If the target shown was the one opposite either the “ahead” or the “below” icon, subjects who memorized the pairs could locate it automatically using a pairs rule, and thus did not need to perform a mental rotation.

Learning effects depended on the target direction category (Above/Behind or Right/Left) (section 3.6). For Right/Left targets, the improvement was greater for the groups with an Inside point-of-view, which is consistent with our previous finding of an Inside point-of-view advantage. By contrast, for target-icons in the Above/Behind direction (remembering that the “Behind” targets are actually “Front” for the Outside groups), the
improvement was greater for the Outside point-of-view. Although the Front direction defined the same physical wall of the virtual room for the Outside groups as Behind did for the Inside groups, that wall was visually accessible to the Outside groups, but not to the Inside groups. Hence learning the Behind target was much harder for the Inside groups than learning the Front target was for the Outside groups. This explains why the Inside point-of-view advantage disappeared for the Above/Behind (Front) targets.

We found also a partial effect of target direction relative to gravity. When considered together, the pair of icons defining the gravitational axis in the baseline orientation (“turtle” and “lion”) was not more accurately located than those on the two other environmental axes. Nevertheless, subjects made fewer errors when pointing to the “turtle” – which was on the ceiling in the baseline orientation. This bias could have simply been due to a stimulus preference, i.e., something about the icon that made it particularly easy to remember or localize. For example, the “turtle” icon was yellow and its contrast with the black surroundings may make it more distinguishable. Or perhaps subjects used a mnemonic, such as “turtle on the top” to remember where it was. Alternatively, it is possible that subjects gave priority to the gravitational axis to organize their spatial knowledge of the baseline orientation, as suggested by other studies (Bryant et al., 1996).
5.

CONCLUSIONS

Space station architectures are generally not designed to assure that all the modules have their internal visual verticals co-aligned. In these circumstances, astronauts often lose track of their orientation relative to the environment and are unable to visualize the orientation of adjacent modules. To ascertain their own orientation, they need to be able to transform their mental models of the modules and their associated landmarks efficiently, and to do this about all three axes of space, whereas on Earth only rotations about a single axis aligned with gravity are required. The present experiment studied subjects’ orientation ability when performing this three-dimensional transformation process in different rotation modes (Array- vs. Self-), and how changing from one to the other may impact the accuracy of their spatial judgements. The objective was also to estimate the effect of point-of-view from which the subjects perceived the environment. Looking at all these effects from the more general perspective of spatial cognition in three dimensions was of particular interest.

Data from 29 subjects was analyzed. Results from 13 other subjects were set aside, because they did not adequately follow the instructions, or because their responses were obvious outliers. Since the experiments required naïve subjects, a between-groups design was adopted. Subject groups were closely balanced in terms of 3D figure rotation ability, and we could find no strong evidence that they differed markedly in terms of inherent spatial skills. Nonetheless, some of our conclusions are based on the assumption that the groups performed in similar fashion. In the future, our conclusions could be prospectively confirmed by repeating all or part of the experiment using additional control groups that have no rotation mode manipulation between training and testing.
Our analysis focused on the percentage of incorrect target indications. Although response time has also been used as the dependent measure in some earlier studies of single-axis spatial transformations, response-time results in the present experiment showed few clear trends. Previous research using single-axis tasks probably focused on response time because these tasks are relatively easy, and subjects make few incorrect responses. We take our task to be more difficult, making percent incorrect a more relevant measure for it.

Our interpretation of the effects of transformation type (section 3.5) is partly based on the assumption that the four experimental groups had similar inherent abilities. Evidence supporting this assumption is reviewed in Section 3.4. Subjects were able to learn how to handle their assigned rotation mode with their given point-of-view in the training phase (with feedback) with relative ease, but the learning rate varied between groups. In training where the subjects had no prior experience, the orientation task was most accurately performed by subjects using the Array-rotation mode. This finding indicates that Self-rotation is not always the easiest mode to adopt in a given situation. The geometry of the array (three-dimensional in the present experiment), the axes of imagined rotation (all possible), and our prior administration of a Cube Rotation Test which required Array rotation skills could be factors that led to the observed Array-rotation advantage in the present experiment.

Subjects were flexible and managed to change rotation mode between the training and testing phases with relatively little difficulty in most cases. It is not entirely clear, however, that the transition between the two modes was abrupt (as the instructions require) or whether the subjects switched strategies gradually. Transfer from one rotation mode to the other occurred successfully, except possibly for the Outside-point-of-view subjects who were trained in Array-rotation mode and then tested using Self-rotation mode. Therefore, not only our spatial mental models, but also the processes we use to transform them can be adapted as needed. It is also notable that many subjects could not resist looking for ways to do the task other than those we specified, e.g., using other rules or an “avatar” strategy (as described in section 3.1). These observations demonstrate the autonomy and richness of the human cognitive processes in dealing with complex spatial tasks. Does an “avatar” strategy improve performance? This should certainly be investigated in future experiments.
Training effects apparently were large only for the group of subjects who were trained and tested Inside. After training in Array-rotation mode, Self-rotation testing became easy, whereas subjects found Self-rotation training initially difficult, and their subsequent Array-rotation testing performance was poor. This suggests that the cognitive processes involved in learning to imagine Array- and Self-rotation may not be totally independent, as is often assumed. In the case of Inside point-of-view, at least, there seems to be a preferred order of exposure to the rotation modes: perhaps mastering Self-rotation with an Inside point-of-view involves a higher-level cognitive process that is facilitated by handling Array-rotation first. The majority of our subjects found Array-rotation initially easier when training Inside. Perhaps learning the environment using an “easy” rotation mode “bootstraps” the subject’s ability to learn a more cognitively complex Self-rotation task. At least in this study, the subjects who experienced the modes in the reverse order had poorer performance.

We demonstrated a clear effect of target direction relative to the body. As in previous studies, all our subjects located target-icons Above and Behind them more accurately than those on their Right and Left. The natural left-right reflection symmetry of the body, and the fact that many subjects memorized opposite pairs of targets may have both contributed to this effect. We noted that the improvement for each direction category (Above/Behind and Right/Left) was not similar for all groups. Also, there was an apparent effect of target direction relative to gravity, as defined by the baseline orientation. Subjects responded more accurately to the ceiling target. We cannot be sure if this is truly a gravity effect, or if it occurred simply because this icon was more visually salient or memorable.

Other interesting questions remain. The present experiment was blocked by point-of-view, with rotation mode as the major experimentally manipulated variable. The experiment could be repeated using the converse design. Also, since astronauts might, arguably, use Self-rotation when Inside a module – but use Array-rotation when visualizing modules other than the one they were in – it is important ultimately to do experiments where both point-of-view and rotation mode covary, i.e., to test along the diagonal of our Figure 2.3 experiment design. One would also verify whether the results are substantially different if the virtual environment simulated a realistic room, such as an office, or an
interior like that of the International Space Station, where the objects and their arrangement were more salient, and so subjects didn’t have to work so hard to memorize them. Ultimately, this basic experiment paradigm could then be adapted and extended to become a tool for comparing of the impact of different architectures and interior designs on spatial orientation ability.

Looking at the experiment’s results from the perspective of designing efficient orientation training, although this experiment focused on the effects of manipulation of rotation mode, it seems likely that both rotation mode and point-of-view have potentially important effects on performance. Any spatial memory training program, therefore, should probably give crewmembers experience transforming mental models with all combinations of rotation mode and point-of-view, and not concentrate on just one. Astronauts are currently trained on Earth inside mock-ups of the modules and presumably adopt the Inside/Self-rotation mode (about their body’s yaw axis). Our results indicate that when the transformation task is three-dimensional, Inside/Self-rotation is relatively difficult, and subjects did not subsequently perform well using Inside/Array-rotation mode. Our results suggest that training astronauts to do spatial transformations in Array-rotation mode may improve their ability to perform three-dimensional Self-rotation transformations when they reach orbit.
REFERENCES


APPENDIX A

Programming Code

Program code was written in Python (v. 2) and VRUT (release 2.2).

Included in this appendix are programs for the training and testing phases of the two groups for which the display provided an Inside point-of-view to the environment (Group ISA and Group IAS).

For the other two groups with an Outside point-of-view (Group OSA and Group OAS), the code was identical except for the parameter that determines the position of the observer’s head, at the beginning of the program:

```
vrut.translate(vrut.HEAD_POS, 0, 0, -1.7)
```

[instead of `vrut.translate(vrut.HEAD_POS, 0, 0, -0.6)` in the Inside case.]

Manipulation of the rotation mode between the two phases does not appear in the code as it concerns a mental process executed by the subject.

TRAINING PHASE for the Inside Groups

#Choose which icon set to train on first:
#'1' for ANIMALS1 (CROSS) set first
#'2' for ANIMALS2 (SQUARE) set first
ICONORDER = 1

#Choose which stimulus file to use
FILENAME = 'TrainingH.txt'
PRACTICE = 'PracticeBe.txt'
#Enter subject’s name in quotes here
SUBJECT = "

# 1 - HMD/tracker, 2 - CONSOLE/no tracker
HMD = 2

import vrut
import os
from string import *
import win32api
import time
from whrandom import random
from random import choice

if HMD == 1:
vrut.go(vrut.STEREO | vrut.HMD)
print 'adding sensor------------------------'
s = vrut.addsensor('intersense')
vrut.tracker()
else:
    vrut.go(vrut.CONSOLE)

vrut.setfov(60, 1.333)
vrut.setipd(0.06)

# Put eyepoint inside node against back surface
vrut.eyeheight(0)
vrut.translate(vrut.HEAD_POS, 0, 0, -.6)

#**********************************************
# Load geometry
#**********************************************

# Order in which you want the Icon sets to be presented.
if ICONORDER == 1:
    first = vrut.addchild('../models/animals1.wrl')
    second = vrut.addchild('../models/animals2.wrl')
    frame2 = vrut.addchild('../models/ww_box.wrl')
    frame1 = vrut.addchild('../models/ww_box.wrl')
elif ICONORDER == 2:
    first = vrut.addchild('../models/animals2.wrl')
    second = vrut.addchild('../models/animals1.wrl')
    frame1 = vrut.addchild('../models/ww_box.wrl')
    frame2 = vrut.addchild('../models/ww_box.wrl')

#**********************************************
# vrut.override()
# frame1.alpha(1.0)
# frame2.alpha(1.0)

mask = vrut.addchild('../models/maskADJ.wrl')
cardBegin = vrut.addchild('../models/ww_trialBegin.wrl')
practiceIcons = vrut.addchild('../models/ww_practice.wrl')
switchCard = vrut.addchild('../models/switch.wrl')
calibrateBegin = vrut.addchild('../models/calibrateCard.wrl')
calibrateFinish = vrut.addchild('../models/calibrateFinish.wrl')
practiceBegin = vrut.addchild('../models/practiceBegin.wrl')

#---------------------------------------------

Arrows = []
Arrows.append(vrut.addchild('../models/arrow.wrl'))
Arrows.append(vrut.addchild('../models/arrow.wrl'))
Arrows.append(vrut.addchild('../models/arrow.wrl'))
Arrows.append(vrut.addchild('../models/arrow.wrl'))
Arrows.append(vrut.addchild('../models/arrow.wrl'))
Arrows.append(vrut.addchild('../models/arrow.wrl'))

#---------------------------------------------

iPracCard = []
iPracCard.append(vrut.addchild('../models/picture_card.wrl'))
iPracCard.append(vrut.addchild('../models/picture_card.wrl'))
iPracCard.append(vrut.addchild('../models/picture_card.wrl'))
iPracCard.append(vrut.addchild('../models/picture_card.wrl'))
iPracCard.append(vrut.addchild('../models/picture_card.wrl'))
iPracCard.append(vrut.addchild('../models/picture_card.wrl'))

iPracTex = []
iPracTex.append(vrut.addtexture('../practiceIcons/diamonds.jpg')) # layer 0
iPracTex.append(vrut.addtexture('../practiceIcons/star.jpg')) # layer 1
# Layer 2
iPracTex.append(vrut.addtexture('../practiceIcons/hearts.jpg'))
# Layer 3
iPracTex.append(vrut.addtexture('../practiceIcons/pound.jpg'))
# Layer 4
iPracTex.append(vrut.addtexture('../practiceIcons/spades.jpg'))
# Layer 5
iPracTex.append(vrut.addtexture('../practiceIcons/clubs.jpg'))

# Cards for Experiment Icons
iCard = []
iCard.append(vrut.addchild('../models/picture_card.wrl'))
iCard.append(vrut.addchild('../models/picture_card.wrl'))
iCard.append(vrut.addchild('../models/picture_card.wrl'))
iCard.append(vrut.addchild('../models/picture_card.wrl'))
iCard.append(vrut.addchild('../models/picture_card.wrl'))
iCard.append(vrut.addchild('../models/picture_card.wrl'))
iCard.append(vrut.addchild('../models/picture_card.wrl'))
iCard.append(vrut.addchild('../models/picture_card.wrl'))
iCard.append(vrut.addchild('../models/picture_card.wrl'))
iCard.append(vrut.addchild('../models/picture_card.wrl'))
iCard.append(vrut.addchild('../models/picture_card.wrl'))

iCard2 = []
iCard2.append(vrut.addchild('../models/picture_card.wrl'))
iCard2.append(vrut.addchild('../models/picture_card.wrl'))
iCard2.append(vrut.addchild('../models/picture_card.wrl'))
iCard2.append(vrut.addchild('../models/picture_card.wrl'))
iCard2.append(vrut.addchild('../models/picture_card.wrl'))
iCard2.append(vrut.addchild('../models/picture_card.wrl'))

# Target Cards for Animals 1 and 2
iTex = []
iTex.append(vrut.addtexture('../nonpolarized/TARGfish.jpg'))  # Surface 0
iTex.append(vrut.addtexture('../nonpolarized/TARGturtles.jpg'))  # Surface 1
iTex.append(vrut.addtexture('../nonpolarized/TARGparrots.jpg'))  # Surface 2
iTex.append(vrut.addtexture('../nonpolarized/TARGlions.jpg'))  # Surface 3
iTex.append(vrut.addtexture('../nonpolarized/TARGbutterflies.jpg'))  # Surface 4
iTex.append(vrut.addtexture('../nonpolarized/TARGgiraffes.jpg'))  # Surface 5

# Target, Break, and End Cards
iTex.append(vrut.addtexture('../nonpolarized/targbackground.jpg'))  # Target Card
iTex.append(vrut.addtexture('../textures/break.jpg'))  # Break Card
iTex.append(vrut.addtexture('../textures/practiceEnd.jpg'))  # End of First Session Card
iTex.append(vrut.addtexture('../textures/end1.jpg'))  # End Card
iTex.append(vrut.addtexture('../textures/end2.jpg'))  # End Card

# The Natural Ordering of Icon Sets (i.e. Animals 1 first, Animals 2 second):
if ICONORDER == 1:
    for i in range(0, 6):
        iCard[i].texture(iTex[i], 'card')
        iCard2[i].texture(iTex[i+6], 'card')
else:
    for i in range(0, 6):
for i in range(0, 6):
    iPracCard[i].texture(iPracTex[i], 'card')
    iCard[i].scale(1.5, 1.5, 0)
    iCard2[i].scale(1.5, 1.5, 0)
    iPracCard[i].scale(1.5, 1.5, 0)

for i in range(6, 11):
    iCard[i].texture(iTex[i+6], 'card')

#***************************************
# Positioning of instructional objects
#***************************************
for i in range(0, 6):
    Arrows[i].scale(.5, .5, .5)
    # surface on right
    Arrows[0].rotate(0,1,0, -170)
    Arrows[0].translate(.25,0,0)
    # surface above
    Arrows[1].rotate(0,0,1, -90)
    Arrows[1].translate(0,.25,0)
    # surface on left
    Arrows[2].rotate(0,1,0, -10)
    Arrows[2].translate(-.25,0,.1)
    # surface below
    Arrows[3].rotate(0,0,1, 90)
    Arrows[3].translate(0,-.25,0)
    # surface straight ahead
    Arrows[4].rotate(0,1,0, 80)
    Arrows[4].translate(.15,0,.25)
    # surface behind
    Arrows[5].rotate(0,1,0, -90)
    Arrows[5].translate(.02,0,-.25)
    cardBegin.translate(0, 0, -.01)
    switchCard.translate(0,0,-.01)
    calibrateBegin.translate(0,0,-.01)
    calibrateFinish.translate(0,0,-.01)
    practiceBegin.translate(0,0,-.01)
    iCard[6].translate(0, 0, .27)
    cardBegin.scale(2,2,1)
    switchCard.scale(2,2,1)
    calibrateBegin.scale(2,2,1)
    calibrateFinish.scale(2,2,1)
    practiceBegin.scale(2,2,1)

for i in range(7, 11):
    iCard[i].scale(2, 2, 0)
    iCard[6].scale(2.5, 2.5, 0)

#*****************************************************************************
#Hide all the geometry and icons
#*****************************************************************************
frame1.curtain(vrtu.CLOSE)
frame2.curtain(vrtu.CLOSE)
first.curtain(vrtu.CLOSE)
second.curtain(vrtu.CLOSE)

practicecons.curtain(vrtu.CLOSE)

cardBegin.curtain(vrtu.CLOSE)
switchCard.curtain(vrtu.CLOSE)
calibrateBegin.curtain(vrtu.CLOSE)
calibrateFinish.curtain(vrtu.CLOSE)
practiceBegin.curtain(vrtu.CLOSE)
mask.curtain(vrtu.CLOSE)

for group in Arrows:
    group.curtain(vrtu.CLOSE)
for card in iCard:
    card.curtain(vrtu.CLOSE)
for card in iCard2:
    card.translate(0, 0, .25)
    card.curtain(vrtu.CLOSE)
for card in iPracCard:
    card.translate(0, 0, .25)
    card.curtain(vrtu.CLOSE)

#***************************************************************************
#Timer flags & conditional variables
#***************************************************************************

BLAST_FACTOR = 1.0
NO_TASK = 0
START_EXP = 1
START_TRIAL = 2
SHOW_STIMULUS = 3
SHOW_TARGET = 4
MEMORY_TASK = 5
SEARCH_TASK = 6
END = 7
TAKE_BREAK = 8
START_TRAINING = 9
VIEW_NODE = 10
CALIFORNIA = 11
CALIBRATION = 12
LIMBO = 13
ORIENT_NODE = 14
SWITCH = 15
SWITCH2 = 16
SWITCH3 = 17
SWITCH4 = 18
SWITCH5 = 19
SWITCH6 = 20
SWITCH7 = 21
REAL_EXPT = 22
MOCK_EXPT = 23
START_PRACTICE = 24
LIMBO2 = 25
LIMBO3 = 26
BEFORE = 27
MEMORY_TASK1 = 28

TRUE = 1
FALSE = 0

# Time constants
STIMULUS_TIME  = 4.0 *BLAST_FACTOR
TARGET_TIME  = 3.0 *BLAST_FACTOR
MEMORY_TIME  = 8.0 *BLAST_FACTOR
SEARCH_TIME  = 9.0 *BLAST_FACTOR

# Numbers in () correspond to the baseline orientation defined by the analysis
# convention laid out by Alan Natapoff.

#Convention
R = (0)
C = (1)
L = (2)
F = (3)
O = (4)
E = (5)
Z = (7) #break card

#******************************
#Other variables
#******************************

#counter for icon sets
ICONSET = 1

#Calibration array for arrow display
#calibrate [0, 5, 1, 2]
calibrate = [0, 2, 1, 2, 0, 5, 1, 5, 0, 1, 2, 5, 0, 1, 2, 5, 0, 1, 2, 5]

# Trial-by-trial specifications:
currentTrial = 0
rotate = 0

# Stuff to record reaction times
startMemory = 0
startSearch = 0
dTime   = 0
counter   = 0
clicker   = 0
trialNum = 0
goNextTrial = TRUE

#******************************
#Read Stimulus File for Practice Trials
#******************************

def InitializePractice():
    global fileP
    global allEntryP
    global allAngleP
    global allTargetP
    global allFloorP
    global allDirectionP
    global ms

    fileP = open(PRACTICE, 'r')
    print 'opened practice file: ', PRACTICE
    #for reading

    #Practice stim file
allP = fileP.readlines()
allEntryP = []
allAngleP = []
allTargetP = []
allFloorP = []
allDirectionP = []

for i in range(0, len(allP)):
    accessP = allP[i]
    sP = split(accessP)
    allEntryP.append(eval(sP[0]))
    allAngleP.append(atoi(sP[1]))
    allTargetP.append(eval(sP[2]))
    allFloorP.append(eval(sP[3]))
    allDirectionP.append(atoi(sP[4]))

fileP.close()
vrut.watch('Initialize Practice...')

#************************************
#Timer for (Dummy) Training Exercises
#************************************

def DummyTraining(timer):
    global TASK
    global ms
    global startTime
    global clicker
    global stimulus

    TASK = timer
    vru.close(task: %d, %d(TASK))

    if timer == START_TRAINING:
        frame1.curtain(vru.OPEN)
        cardBegin.curtain(vru.OPEN)
    elif timer == VIEW_NODE:
        practiceIcons.curtain(vru.OPEN)
    elif timer == SWITCH:
        switchCard.curtain(vru.OPEN)
    elif timer == LIMBO:
        switchCard.curtain(vru.CLOSE)
        cardBegin.curtain(vru.OPEN)
    elif timer == ORIENT_NODE:
        vru.rotate(practiceIcons, vru.ZAXIS, 90)
        practiceIcons.curtain(vru.OPEN)
    elif timer == SWITCH2:
        switchCard.curtain(vru.OPEN)
        vru.rotate(practiceIcons, vru.ZAXIS, -90)
    elif timer == LIMBO2:
        switchCard.curtain(vru.CLOSE)
        calibrateBegin.curtain(vru.OPEN)
    elif timer == CALIBRATION:
        if len(calibrate) > 0:
            startTime = time.time()
            stimulus = choice(calibrate)
            Arrows[stimulus].curtain(vru.OPEN)
        else:
            vru.starttimer(MOCK_EXPT, .1)
elif timer == MOCK_EXPT:
    vrut.watch('ca y est')
calibrateFinish.curtain(vrut.OPEN)

# Subject Controls for Dummy Exercises
# ****************************************
def DummyKey(key):
global counter
global clicker
global direction
global calibrate

eTime = time.time()

if key == '1' and HMD == 1:
    ms.reset()
    win32api.Sleep(200)
    vrut.watch('tracker has been reset')

if key == ' ':
    if TASK == START_TRAINING:
        cardBegin.curtain(vrut.CLOSE)
        vrut.starttimer(VIEW_NODE, .1)
    elif TASK == VIEW_NODE:
        practiceIcons.curtain(vrut.CLOSE)
        vrut.starttimer(SWITCH, .1)
    elif TASK == SWITCH:
        switchCard.curtain(vrut.CLOSE)
        vrut.starttimer(LIMBO, .1)
    elif TASK == LIMBO:
        cardBegin.curtain(vrut.CLOSE)
        vrut.starttimer(ORIENT_NODE, .1)
    elif TASK == ORIENT_NODE:
        practiceIcons.curtain(vrut.CLOSE)
        vrut.starttimer(SWITCH2, .1)
    elif TASK == SWITCH2:
        switchCard.curtain(vrut.CLOSE)
        vrut.starttimer(LIMBO2, .1)
    elif TASK == LIMBO2:
        calibrateBegin.curtain(vrut.CLOSE)
        vrut.starttimer(CALIBRATION, .1)
    elif TASK == MOCK_EXPT:
        calibrateFinish.curtain(vrut.CLOSE)
        vrut.callback(vrut.TIMER_EVENT, 'PracticeTimer')
        vrut.callback(vrut.KEYBOARD_EVENT, 'PracticeKey')
        vrut.starttimer(START_PRACTICE, .1)
    else:
        return

if TASK == CALIBRATION:
    if (key == '5' and stimulus == 5) or (key == '4' and stimulus == 2):
        Arrows[stimulus].curtain(vrut.CLOSE)
    elif (key == '6' and stimulus == 0) or (key == '8' and stimulus == 1):
        Arrows[stimulus].curtain(vrut.CLOSE)
    else:
return

if (endTime - startTime) < 2.0:
calibrate.remove(stimulus)

vrut.starttimer(CALIBRATION, .1)

if TASK == START_TRAINING:
    if key == 'a':
        cardBegin.curtain(vrut.CLOSE)
        vrut.starttimer(MOCK_EXPT, .1)
    elif key == 'e':
        cardBegin.curtain(vrut.CLOSE)
        vrut.callback(vrut.KEYBOARD_EVENT, 'ExptKey')
        vrut.callback(vrut.TIMER_EVENT, 'ExptTimer')
        vrut.starttimer(START_TRIAL, .1)

#*******************************************************************************
#Timer for PRACTICE Trials
#*******************************************************************************
def PracticeTrials(timer):
    global TASK
    global currentTrial
    global trialNum
    global goNextTrial
    global counter
    global showTargetP
    global showEntryP
    global showRotateP
    global showFloorP
    global showDirectionP
    global rotRoom
    global rotate

    TASK = timer
    vrut.watch('task: %d' % (TASK))

    if timer == START_PRACTICE:
        InitializePractice()
        switchCard.curtain(vrut.CLOSE)
        practiceBegin.curtain(vrut.OPEN)
    elif timer == START_TRIAL:
        counter = 0
        showTargetP = allTargetP[currentTrial]
        showFloorP = allFloorP[currentTrial]
        showEntryP = allEntryP[currentTrial]
        showDirectionP = allDirectionP[currentTrial]

        if showEntryP == Z:
            TakeBreak()
        else:

            # show begin card
            cardBegin.curtain(vrut.OPEN)

            trialNum = trialNum + 1

            vrut.watch('entry: %d' % (showEntryP))
            vrut.watch('target: %d' % (showTargetP))
            vrut.watch('floor: %d' % (showFloorP))
            vrut.watch('direction: %d' % (showDirectionP))

            if showEntryP == F:
                vrut.rotate(practiceIcons, vrut.XAXIS, 90)
            elif showEntryP == O:
                vrut.rotate(practiceIcons, vrut.YAXIS, 180)
            elif showEntryP == L:
vrut.rotate(practiceIcons, vrut.YAXIS, -90)

elif showEntryP == R:
    vrut.rotate(practiceIcons, vrut.YAXIS, 90)
elif showEntryP == C:
    vrut.rotate(practiceIcons, vrut.XAXIS, -90)

elif timer == SHOW_TARGET:
    iCard[6].curtain(vrut.OPEN)
    iPracCard[showTargetP].curtain(vrut.OPEN)
    vrut.starttimer(SHOW_STIMULUS, TARGET_TIME)

elif timer == SHOW_STIMULUS:
    iPracCard[showTargetP].curtain(vrut.CLOSE)
    iCard[6].curtain(vrut.CLOSE)
    showRotateP = allAngleP[rotate]
    if showRotateP == 3:
        showRotateP = -90
    elif showRotateP == 9:
        showRotateP = 90
    elif showRotateP == 6:
        showRotateP = 180
    rotRoom = -showRotateP
    if rotate < len(allAngleP):
        rotate = rotate + 1
    vrut.rotate(practiceIcons, vrut.ZAXIS, rotRoom)
    mask.curtain(vrut.OPEN)
    practiceIcons.curtain(vrut.OPEN)
    Arrows[3].curtain(vrut.OPEN)
    Arrows[4].curtain(vrut.OPEN)
    vrut.starttimer(MEMORY_TASK, STIMULUS_TIME)

elif timer == MEMORY_TASK:
    Arrows[3].curtain(vrut.CLOSE)
    Arrows[4].curtain(vrut.CLOSE)
    practiceIcons.curtain(vrut.CLOSE)
    mask.curtain(vrut.CLOSE)
    vrut.starttimer(SEARCH_TASK, MEMORY_TIME)

elif timer == SEARCH_TASK:
    practiceIcons.curtain(vrut.OPEN)
    counter = counter + 1
    if counter < (SEARCH_TIME*SEARCH_TIME):
        vrut.starttimer(SEARCH_TASK, 1/SEARCH_TIME)
    else:
        vrut.starttimer(END, .1)

elif timer == END:
    vrut.watch('End of a Trial')
    practiceIcons.curtain(vrut.OPEN)
    vrut.rotate(practiceIcons, vrut.ZAXIS, -rotRoom)

    if showEntryP == F:
        vrut.rotate(practiceIcons, vrut.XAXIS, -90)
    elif showEntryP == O:
        vrut.rotate(practiceIcons, vrut.YAXIS, 180)
    elif showEntryP == L:
        vrut.rotate(practiceIcons, vrut.YAXIS, 90)
    elif showEntryP == R:
        vrut.rotate(practiceIcons, vrut.YAXIS, -90)
    elif showEntryP == C:
        vrut.rotate(practiceIcons, vrut.XAXIS, 90)

    currentTrial = currentTrial + 1

    if currentTrial < len(allTargetP):
goNextTrial = TRUE
vru.starttimer(START_TRIAL, .1)
else:
iCard[8].curtain(vru.OPEN)
rotate = 0
currentTrial = 0
trialNum = 0
vru.callback(vru.TIMER_EVENT, 'ExptTimer')
vru.callback(vru.KEYBOARD_EVENT, 'ExptKey')
vru.starttimer(BEFORE, 2.0)

TASK == NO_TASK

# Subject inputs for PRACTICE Trials
#******************************************************************************

def PracticeKey(key):
    global goNextTrial
    global currentTrial
    global rotate
    global counter
    global ms

    if key == '1' and HMD == 1:
        ms.reset()
        win32api.Sleep(200)
        vru.watch('tracker has been reset')

    if TASK == START_PRACTICE:
        if key == ' ':  
            practiceBegin.curtain(vru.CLOSE)
            vru.starttimer(START_TRIAL, .1)
        elif key == 'a':
            practiceBegin.curtain(vru.CLOSE)
            currentTrial = 4
            vru.starttimer(START_TRIAL, .1)
        elif key == 'b':
            practiceBegin.curtain(vru.CLOSE)
            vru.starttimer(START_TRIAL, .1)

    elif TASK == START_TRIAL:
        if key == ' ':  
            cardBegin.curtain(vru.CLOSE)
            vru.starttimer(START_TRIAL, .1)

    elif (TASK==MEMORY_TASK or TASK==SHOW_STIMULUS):
        if (key=='4' or key=='5' or key=='6' or key=='8'):
            win32api.Beep(1300, 90)

    elif TASK == SEARCH_TASK:
        if key == '0':
            counter = SEARCH_TIME*SEARCH_TIME
            win32api.Beep(1300, 90)

    else:
        return

#******************************************************************************
# Read Stimulus File and Open Data File for Experimental Trials
#******************************************************************************
def InitializeExp():

global file
global data
global allEntry
global allAngle
global allTarget
global allFloor
global allDirection
global ms

file = open(FILENAME,'r')
print 'opened stim file: ', FILENAME
#'r' for reading
data = open(SUBJECT,'a')
print 'created output file: ', SUBJECT
#'a' for append
data.write('%CALIFORNIA GROUP' + '
')
data.write('%Subject Name:' + SUBJECT +'
')
data.write('%Stimulus File:' + FILENAME +'
'+'
')
data.write('%Columns ='+'
')
data.write('%S'+'	'+'SET'+'	'+'t#'+'	'+'C'+'	'+'I'+'	'+'RT'+'	'+'	')
data.write('Ent'+'	'+'Targ'+'	'+'Rot'+'	'+'Dir'+'
'+'
')
vrut.watch('opening: '+SUBJECT)

#Training stimulus file
all = file.readlines()
allEntry  = []
allAngle  = []
allTarget = []
allFloor  = []
allDirection = []
for i in range(0, len(all)):
    access = all[i]
    s = split(access)
    allEntry.append(eval(s[0]))
    allAngle.append(atoi(s[1]))
    allTarget.append(eval(s[2]))
    allFloor.append(eval(s[3]))
    allDirection.append(atoi(s[4]))

file.close()
vrut.watch('Initialize Experiment...')

#*******************************************************************************
#               Timer for Experimental Trials
#*******************************************************************************
def ExptTimer(timer):

global TASK
global ms
global currentTrial
global showTarget
global showEntry
global showRotate
global showFloor
global showDirection
global rotRoom
global rotate
global trialNum
global startMemory
global startSearch
global goNextTrial
global counter
global ICONSET
global ans
global buckle

TASK = timer

if timer == BEFORE:
    if ICONSET == 1:
        iCard[8].curtain(vrut.OPEN)
    else:
        iCard[9].curtain(vrut.OPEN)

elif timer == START_TRIAL:
    if currentTrial == 0 and ICONSET == 1:
        InitializeExp()

        counter = 0
        ans = 0
        buckle = 0

        vrut.watch('currentTrial = %d' %currentTrial)

        showTarget = allTarget[currentTrial]
        showFloor = allFloor[currentTrial]
        showEntry = allEntry[currentTrial]
        showDirection = allDirection[currentTrial]

    if showEntry == Z:
        TakeBreak()
    else:
        if ICONSET == 1:
            frame1.curtain(vrut.OPEN)
        else:
            frame1.curtain(vrut.CLOSE)
            frame2.curtain(vrut.OPEN)
            cardBegin.curtain(vrut.OPEN)

        trialNum = trialNum + 1

        vrut.watch('entry: %d' %showEntry)
        vrut.watch('target: %d' %showTarget)
        vrut.watch('floor: %d' %showFloor)
        vrut.watch('trial: %d' %trialNum)

    if ICONSET == 1:
        if showEntry == F:
            vrut.rotate(first, vrut.XAXIS, 90)
        elif showEntry == O:
            vrut.rotate(first, vrut.YAXIS, 180)
        elif showEntry == L:
            vrut.rotate(first, vrut.YAXIS, -90)
        elif showEntry == R:
            vrut.rotate(first, vrut.YAXIS, 90)
        elif showEntry == C:
            vrut.rotate(first, vrut.XAXIS, -90)
    else:
        if showEntry == F:
            vrut.rotate(second, vrut.XAXIS, 90)
        elif showEntry == O:
            vrut.rotate(second, vrut.YAXIS, 180)
        elif showEntry == L:
            vrut.rotate(second, vrut.YAXIS, -90)
        elif showEntry == R:
            vrut.rotate(second, vrut.YAXIS, 90)
        elif showEntry == C:
            vrut.rotate(second, vrut.XAXIS, -90)
elif timer == SHOW_TARGET:
    iCard[6].curtain(vrut.OPEN)
    if ICONSET == 1:
        iCard[showTarget].translate(0, 0, .25)
        iCard[showTarget].curtain(vrut.OPEN)
    else:
        iCard2[showTarget].curtain(vrut.OPEN)
    vrt.starttimer(SHOW_STIMULUS, TARGET_TIME)

elif timer == SHOW_STIMULUS:
    iCard[showTarget].curtain(vrut.CLOSE)
    iCard2[showTarget].curtain(vrut.CLOSE)
    iCard[6].curtain(vrut.CLOSE)
    startMemory = time.time()
    showRotate = allAngle[rotate]
    vrt.watch('rotate = %d' %(rotate))
    if showRotate == 3:
        showRotate = -90
    elif showRotate == 9:
        showRotate = 90
    elif showRotate == 6:
        showRotate = 180
    rotRoom = -showRotate
    if rotate < len(allAngle):
        rotate = rotate + 1
    mask.curtain(vrut.OPEN)
    Arrows[3].curtain(vrut.OPEN)
    Arrows[4].curtain(vrut.OPEN)
    if ICONSET == 1:
        vrt.rotate(first, vrut.ZAXIS, rotRoom)
        first.curtain(vrut.OPEN)
    else:
        vrt.rotate(second, vrut.ZAXIS, rotRoom)
        second.curtain(vrut.OPEN)
    vrt.starttimer(MEMORY_TASK, STIMULUS_TIME)

elif timer == MEMORY_TASK:
    if ans == 1:
        vrt.starttimer(SEARCH_TASK, 1/SEARCH_TIME)
    else:
        if ICONSET == 1:
            first.curtain(vrut.CLOSE)
        else:
            second.curtain(vrut.CLOSE)
        mask.curtain(vrut.CLOSE)
        Arrows[3].curtain(vrut.CLOSE)
        Arrows[4].curtain(vrut.CLOSE)
        vrt.starttimer(MEMORY_TASK1, 1/SEARCH_TIME)

elif timer == MEMORY_TASK1:
    buckle = buckle + 1
    if buckle < (MEMORY_TIME*MEMORY_TIME):
        vrt.starttimer(MEMORY_TASK1, 1/SEARCH_TIME)
    else:
        vrt.starttimer(SEARCH_TASK)

elif timer == SEARCH_TASK:
    if ans == 1:
        mask.curtain(vrut.CLOSE)
        Arrows[3].curtain(vrut.CLOSE)
        Arrows[4].curtain(vrut.CLOSE)
    if ICONSET == 1:
        first.curtain(vrut.OPEN)
else:
    second.curtain(vrut.OPEN)
    if counter == 0:
        startSearch = time.time()
        counter = counter + 1
    if counter < (SEARCH_TIME*SEARCH_TIME):
        vrut.starttimer(SEARCH_TASK, 1/SEARCH_TIME)
    else:
        vrut.starttimer(END)
elif timer == END:
    vrut.watch('End of a Trial')
if ICONSET== 1:
    first.curtain(vrut.CLOSE)
    vrut.rotate(first, vrut.ZAXIS, -rotRoom)
    if showEntry == F:
        vrut.rotate(first, vrut.XAXIS, -90)
    elif showEntry == O:
        vrut.rotate(first, vrut.XAXIS, -90)
    elif showEntry == L:
        vrut.rotate(first, vrut.YAXIS, 180)
    elif showEntry == R:
        vrut.rotate(first, vrut.YAXIS, -90)
    elif showEntry == C:
        vrut.rotate(first, vrut.XAXIS, 90)
else:
    second.curtain(vrut.CLOSE)
    vrut.rotate(second, vrut.ZAXIS, -rotRoom)
    if showEntry == F:
        vrut.rotate(second, vrut.XAXIS, -90)
    elif showEntry == O:
        vrut.rotate(second, vrut.YAXIS, 180)
    elif showEntry == L:
        vrut.rotate(second, vrut.YAXIS, 90)
    elif showEntry == R:
        vrut.rotate(second, vrut.YAXIS, -90)
    elif showEntry == C:
        vrut.rotate(second, vrut.XAXIS, 90)

currentTrial = currentTrial + 1
if currentTrial < len(allTarget):
    goNextTrial = TRUE
    vrut.starttimer(START_TRIAL, .1)
else:
iCard[9].curtain(vrut.OPEN)
    print 'Its all over folks!'
    if ICONSET == 1:
        ICONSET = ICONSET + 1
        currentTrial = 0
        rotate = 0
        trialNum = 0
        data.write("%End of first session for " + SUBJECT + "n"
        data.write("%Beginning of second session for " + SUBJECT + "n"
        data.flush()
        vrut.callback(vrut.KEYBOARD_EVENT, 'ExptKey')
        vrut.callback(vrut.TIMER_EVENT, 'ExptTimer')
        vrut.starttimer(BEFORE, .1)
    else:
iCard[10].curtain(vrut.OPEN)

#******************************************************************************
#Subject inputs for Experimental Trials
#******************************************************************************

def ExptKey(key):
global data
global goNextTrial
global currentTrial
global rotate
global counter
global trialNum
global ans
global buckle

endTime = time.time()
#vrut.watch('Response Time: %d' %(endTime-startTime))
vrut.watch('task: %d' %(TASK))

if key =='1' and HMD == 1:
    ms.reset()
    win32api.Sleep(200)
    vrut.watch('tracker has been reset')

if TASK == BEFORE:
    if key == ' ':  
        if ICONSET == 1:
            iCard[8].curtain(vrut.CLOSE)
        else:
            iCard[9].curtain(vrut.OPEN)
        vrut.starttimer(START_TRIAL, .1)

elif TASK == START_TRIAL:

    #***************************************************************
    #Set trial to beginning of Stage 2 (2nd back surface)
    #***************************************************************
    if key == 'a':
        currentTrial = 21
        rotate = 21
        trialNum = 22
    if ICONSET == 1:
        if showEntry == F:
            vrut.rotate(first, vrut.XAXIS, -90)
        elif showEntry == O:
            vrut.rotate(first, vrut.YAXIS, 180)
        elif showEntry == L:
            vrut.rotate(first, vrut.YAXIS, 90)
        elif showEntry == R:
            vrut.rotate(first, vrut.YAXIS, -90)
        elif showEntry == C:
            vrut.rotate(first, vrut.XAXIS, 90)
        else:
            if showEntry == F:
                vrut.rotate(second, vrut.XAXIS, -90)
            elif showEntry == O:
                vrut.rotate(second, vrut.YAXIS, 180)
            elif showEntry == L:
                vrut.rotate(second, vrut.YAXIS, 90)
            elif showEntry == R:
                vrut.rotate(second, vrut.YAXIS, -90)
            elif showEntry == C:
                vrut.rotate(second, vrut.XAXIS, 90)

    vrut.starttimer(START_TRIAL, .1)

elif key == 'b':
    currentTrial = 10
    rotate = 10

    if ICONSET == 1:
        if showEntry == F:
            vrut.rotate(first, vrut.XAXIS, -90)
        elif showEntry == O:
            vrut.rotate(second, vrut.XAXIS, -90)
        elif showEntry == L:
            vrut.rotate(second, vrut.YAXIS, 180)
        elif showEntry == R:
            vrut.rotate(second, vrut.YAXIS, 90)
        elif showEntry == C:
            vrut.rotate(second, vrut.XAXIS, 90)

    vrut.starttimer(START_TRIAL, .1)
vrut.rotate(first, vrut.YAXIS, 180)
eelif showEntry == L:
    vrut.rotate(first, vrut.YAXIS, 90)
eelif showEntry == R:
    vrut.rotate(first, vrut.YAXIS, -90)
eelif showEntry == C:
    vrut.rotate(first, vrut.XAXIS, 90)
eelse:
e    if showEntry == F:
        vrut.rotate(second, vrut.XAXIS, -90)
eelif showEntry == O:
    vrut.rotate(second, vrut.YAXIS, 180)
eelif showEntry == L:
    vrut.rotate(second, vrut.YAXIS, 90)
eelif showEntry == R:
    vrut.rotate(second, vrut.YAXIS, -90)
eelif showEntry == C:
    vrut.rotate(second, vrut.XAXIS, 90)
vrut.starttimer(START_TRIAL, .1)
eelif key == ':':
    cardBegin.curtain(vrut.CLOSE)
    vrut.starttimer(SHOW_TARGET, .1)

eelif TASK == TAKE_BREAK:
e    if key == ':':
e        if rotate < len(allAngle):
            rotate = rotate + 1
        if currentTrial < len(allTarget):
            currentTrial = currentTrial + 1
        goNextTrial = TRUE
    # Hide the break card
    iCard[7].curtain(vrut.CLOSE)
    vrut.starttimer(START_TRIAL, .1)

eelif (TASK == SHOW_STIMULUS or TASK == MEMORY_TASK or TASK == MEMORY_TASK1):
e    if (key=='4' or key=='5' or key=='6' or key=='8'):
        ans = 1
        buckle = MEMORY_TIME*MEMORY_TIME
    vrut.watch('response registered')
    #sound a "beep"
    win32api.Beep(1300, 90)
    #write the data to file
    outdata = SUBJECT+'	'+str(ICONSET)+'	'+str(trialNum)+'	'+'1'+'	'+str(key)+'	'+str(endTime-startMemory)
    outdata = outdata+'	'+str(showEntry)+'	'+str(showTarget)+'	'+str(showRotate)+'	'+str(showDirection)+'
    data.write(outdata)
    data.flush()

eelif TASK == SEARCH_TASK:
e    if key=='0':
        counter = SEARCH_TIME*SEARCH_TIME
    outdata = SUBJECT+'	'+str(ICONSET)+'	'+str(trialNum)+'	'+'0'+'	'+str(key)+'	'+str(endTime-startSearch)+'	'
    outdata = outdata+'	'+str(showEntry)+'	'+str(showTarget)+'	'+str(showRotate)+'	'+str(showDirection)+'
    data.write(outdata)
    data.flush()
eelse:
    return

#*******************************************
#Break for Subject
#*******************************************

def TakeBreak():


global TASK

# show take break card
iCard[7].curtain(vrut.OPEN)
TASK = TAKE_BREAK

#*******************************************************************************
#*******************************************************************************
vrut.callback(vrut.KEYBOARD_EVENT, 'DummyKey')
vrut.callback(vrut.TIMER_EVENT, 'DummyTraining')
vrut.starttimer(START_TRAINING, 0.5)

#vrut.callback(vrut.KEYBOARD_EVENT, 'ExptKey')
#vrut.callback(vrut.TIMER_EVENT, 'ExptTimer')
#vrut.starttimer(START_TRIAL, 0.5)

TESTING PHASE for the Inside Groups
(no practice trials, no familiarization trials)

#Choose which icon set to train on first:
#"1" for ANIMALS1 (CROSS) set first
#"2" for ANIMALS2 (SQUARE) set first
ICONORDER = 1

#Choose which stimulus file to use
FILENAME = 'Exp1TrainingH.txt'
#Enter subject's name in quotes here
SUBJECT = "
# 1 - HMD/tracker, 2 - CONSOLE/no tracker
HMD = 2

import vrut
import os
import win32api
import time
from whrandom import random
from random import choice
if HMD == 1:
    vrut.go(vrut.STEREO | vrut.HMD)
    print 'adding sensor'
    ms = vrut.addsensor('intersense')
    vrut.tracker()
else:
    vrut.go(vrut.CONSOLE)
vrut.setfov(60, 1.333)
vrut.setipd(0.06)
#Put eyept inside node against back surface
vrut.eyeheight(0)
vrut.translate(vrut.HEAD_POS, 0, 0, -.6)

#*******************************************************************************
#Load geometry
#*******************************************************************************

#Order in which you want the Icon sets to be presented.
if ICONORDER == 1:
    first = vrut.addchild('../models/animals1.wrl')
    second = vrut.addchild('../models/animals2.wrl')
    frame2 = vrut.addchild('../models/ww_box.wrl')
    frame1 = vrut.addchild('../models/ww_box.wrl')
elif ICONORDER == 2:
    first = vrut.addchild('../models/animals2.wrl')
    second = vrut.addchild('../models/animals1.wrl')
    frame1 = vrut.addchild('../models/ww_box.wrl')
    frame2 = vrut.addchild('../models/ww_box.wrl')

#-----------------------------------

#practiceIcons = vrut.addchild('../models/PRACTICEbig.wrl')
#vrut.override()
#frame1.alpha(1.0)
#frame2.alpha(1.0)
mask = vrut.addchild('../models/maskADJ.wrl')
cardBegin = vrut.addchild('../models/ww_trialbegin.wrl')
switchCard = vrut.addchild('../models/switch.wrl')

Arrows = []
Arrows.append(vrut.addchild('../models/arrow.wrl'))
Arrows.append(vrut.addchild('../models/arrow.wrl'))
Arrows.append(vrut.addchild('../models/arrow.wrl'))
Arrows.append(vrut.addchild('../models/arrow.wrl'))
Arrows.append(vrut.addchild('../models/arrow.wrl'))
Arrows.append(vrut.addchild('../models/arrow.wrl'))

#-----------------------------------

#cards for EXPERIMENT ICONS
iCard = []
iCard.append(vrut.addchild('../models/picture_card.wrl'))
iCard.append(vrut.addchild('../models/picture_card.wrl'))
iCard.append(vrut.addchild('../models/picture_card.wrl'))
iCard.append(vrut.addchild('../models/picture_card.wrl'))
iCard.append(vrut.addchild('../models/picture_card.wrl'))
iCard.append(vrut.addchild('../models/picture_card.wrl'))
iCard.append(vrut.addchild('../models/picture_card.wrl'))
iCard.append(vrut.addchild('../models/picture_card.wrl'))
iCard.append(vrut.addchild('../models/picture_card.wrl'))
iCard.append(vrut.addchild('../models/picture_card.wrl'))
iCard.append(vrut.addchild('../models/picture_card.wrl'))
iCard.append(vrut.addchild('../models/picture_card.wrl'))
iCard.append(vrut.addchild('../models/picture_card.wrl'))
iCard2 = []
iCard2.append(vrut.addchild('../models/picture_card.wrl'))
iCard2.append(vrut.addchild('../models/picture_card.wrl'))
iCard2.append(vrut.addchild('../models/picture_card.wrl'))
iCard2.append(vrut.addchild('../models/picture_card.wrl'))
iCard2.append(vrut.addchild('../models/picture_card.wrl'))
iCard2.append(vrut.addchild('../models/picture_card.wrl'))
iCard2.append(vrut.addchild('../models/picture_card.wrl'))
iCard2.append(vrut.addchild('../models/picture_card.wrl'))
iCard2.append(vrut.addchild('../models/picture_card.wrl'))
iCard2.append(vrut.addchild('../models/picture_card.wrl'))
iCard2.append(vrut.addchild('../models/picture_card.wrl'))
iCard2.append(vrut.addchild('../models/picture_card.wrl'))
iCard2.append(vrut.addchild('../models/picture_card.wrl'))
iCard2.append(vrut.addchild('../models/picture_card.wrl'))

#-----------------------------------

iTex = []
#target cards for animals1:
    iTex[0-5]
iTex.append(vrut.addtexture('../nonpolarized/TARGfish.jpg')) # surface 0
iTex.append(vrut.addtexture('../nonpolarized/TARGturtles.jpg')) # surface 1
iTex.append(vrut.addtexture('../nonpolarized/TARGparrots.jpg')) # surface 2
iTex.append(vrut.addtexture('../nonpolarized/TARGlions.jpg')) # surface 3
iTex.append(vrut.addtexture('../nonpolarized/TARGbutterflies.jpg')) # surface 4
iTex.append(vrut.addtexture('../nonpolarized/TARGgiraffes.jpg')) # surface 5
#target cards for animals2
    iTex[6-11]
iTex.append(vrut.addtexture('../nonpolarized/TARGdeer.jpg')) # layer 0
iTex.append(vrut.addtexture('../nonpolarized/TARGfrogs.jpg')) # layer 1
iTex.append(vrut.addtexture('../nonpolarized/TARGsnakes.jpg')) # layer 2
iTex.append(vrut.addtexture('../nonpolarized/TARGbluebirds.jpg')) # layer 3
iTex.append(vrut.addtexture('../nonpolarized/TARGelephants.jpg')) # layer 4
iTex.append(vrut.addtexture('../nonpolarized/TARGroosters.jpg')) # layer 5
# Target, Break, and End cards:

```python
iTex.append(vrut.addtexture('../nonpolarized/targbackground.jpg'))   # target card
iTex.append(vrut.addtexture('../textures/break.jpg'))      # break card
iTex.append(vrut.addtexture('../textures/end1.jpg'))      # end of first session card
iTex.append(vrut.addtexture('../textures/end2.jpg'))      # end card
```

# The natural ordering of icon sets (i.e. animals1 first, animals2 second):

```python
if ICONORDER == 1:
    for i in range(0, 6):
        iCard[i].texture(iTex[i], 'card')
        iCard2[i].texture(iTex[i+6], 'card')
```

# The REVERSE ordering of icon sets (i.e. animals2 first, animals1 second):

```python
elif ICONORDER == 2:
    for i in range(0, 6):
        iCard[i].texture(iTex[i+6], 'card')
        iCard2[i].texture(iTex[i], 'card')
```

```python
for i in range(0, 6):
    iCard[i].scale(1.5, 1.5, 0)
    iCard2[i].scale(1.5, 1.5, 0)
```

```python
for i in range(6, 10):
    iCard[i].texture(iTex[i+6], 'card')
```

# Positioning of instructional objects

```python
for i in range(0, 6):
    Arrows[i].scale(.5, .5, .5)
    Arrows[i].rotate(0,1,0, -170)
    Arrows[i].translate(.25,0,0)
    Arrows[i].rotate(0,0,1, -90)
    Arrows[i].translate(0,.25,0)
    Arrows[i].rotate(0,1,0, -10)
    Arrows[i].translate(-.25,0,.1)
    Arrows[i].rotate(0,0,1, 90)
    Arrows[i].translate(0,-.25,0)
    Arrows[i].rotate(0,1,0, 80)
    Arrows[i].translate(.15,0,.25)
    Arrows[i].rotate(0,1,0, -90)
    Arrows[i].translate(.1,0,-.25)
    cardBegin.translate(0, 0, -.01)
    switchCard.translate(0,0,-.01)
    iCard[6].translate(0, 0, .27)
    cardBegin.scale(2,2,1)
    switchCard.scale(2,2,1)
    for i in range(7, 11):
        iCard[i].scale(2, 2, 0)
```

---

86
iCard[6].scale(2.5, 2.5, 0)

#******************************************************************************
#Hide all the geometry and icons
#******************************************************************************

frame1.curtain(vrut.CLOSE)
frame2.curtain(vrut.CLOSE)
first.curtain(vrut.CLOSE)
second.curtain(vrut.CLOSE)
cardBegin.curtain(vrut.CLOSE)
switchCard.curtain(vrut.CLOSE)
mask.curtain(vrut.CLOSE)

for group in Arrows:
    group.curtain(vrut.CLOSE)
for card in iCard:
    card.curtain(vrut.CLOSE)
for card in iCard2:
    card.translate(0, 0, .25)
    card.curtain(vrut.CLOSE)

#******************************************************************************
#timer flags & conditional variables
#******************************************************************************

BLAST_FACTOR = 1.0

NO_TASK = 0
START_EXP = 1
START_TRIAL = 2
SHOW_STIMULUS = 3
SHOW_TARGET = 4
MEMORY_TASK = 5
SEARCH_TASK = 6
END = 7
TAKE_BREAK = 8
START_TRAINING = 9
VIEW_NODE = 10
CALIFORNIA = 11
CALIBRATION = 12
LIMBO = 13
ORIENT_NODE = 14
SWITCH = 15
SWITCH2 = 16
SWITCH3 = 17
SWITCH4 = 18
SWITCH5 = 19
SWITCH6 = 20
SWITCH7 = 21
REAL_EXPT = 22
MOCK_EXPT = 23
START_PRACTICE = 24
LIMBO2 = 25
LIMBO3 = 26
BEFORE = 27
TRUE = 1
FALSE = 0

# Time constants
STIMULUS_TIME = 4.0 *BLAST_FACTOR
TARGET_TIME = 3.0 *BLAST_FACTOR
MEMORY_TIME = 8.0 *BLAST_FACTOR
SEARCH_TIME = 9.0 *BLAST_FACTOR
# Numbers in () correspond to the baseline orientation defined by the analysis
# convention laid out by Alan Natapoff.
R = (0)
C = (1)
L = (2)
F = (3)
O = (4)
E = (5)
Z = (7) # break card

# Other variables

# counter for icon sets
ICONSET = 1

# Trial-by-trial specifications:
currentTrial = 0
rotate = 0

# Stuff to record reaction times
startMemory = 0
startSearch = 0
endTime = 0

counter = 0
clicker = 0
trialNum = 0
goNextTrial = TRUE

def InitializeExp():
    global file
global data
global allEntry
global allAngle
global allTarget
global allFloor
global allDirection
global ms

    file = open(FILENAME,'r')
    print 'opened stim file: ', FILENAME
    data = open(SUBJECT,'a')
    print 'created output file: ', SUBJECT
    vrut.watch('opening: '+SUBJECT)

    # Training stimulus file
    all = file.readlines()
    allEntry = []

    for line in all:
        data.write(line)
    data.write('%CALIFORNIA GROUP' + '
')
    data.write('%Subject Name:' + SUBJECT +'
'+'
')
    data.write('%Stimulus File:' + FILENAME +'
'+'
')
    data.write('%Columns ='+'
')
    data.write('%S'+'	'+'SET'+'	'+'t#'+'	'+'C'+'	'+'I'+'	'+'RT'+'	'+'	')
    data.write('Ent'+'	'+'Targ'+'	'+'Rot'+'	'+'Dir'+'
'+'
')

    vrut.watch('opening: '+SUBJECT)
allAngle = []
allTarget = []
allFloor = []
allDirection = []

for i in range(0, len(all)):
    access = all[i]
    s = split(access)
    allEntry.append(eval(s[0]))
    allAngle.append(atoi(s[1]))
    allTarget.append(eval(s[2]))
    allFloor.append(eval(s[3]))
    allDirection.append(atoi(s[4]))

file.close()
vrut.watch('Initialize Experiment...')

#*******************************************************************************
#  Timer for Experimental Trials
#*******************************************************************************

def ExptTimer(timer):
    global TASK
    global ms
    global currentTrial
    global showTarget
    global showEntry
    global showRotate
    global showFloor
    global showDirection
    global rotRoom
    global rotate
    global trialNum
    global startMemory
    global startSearch
    global goNextTrial
    global counter
    global ICONSET
    global ans
    global data
    global outdata

    TASK = timer

    if timer == BEFORE:
        if ICONSET == 1:
            iCard[8].curtain(vrut.OPEN)
        else:
            iCard[9].curtain(vrut.OPEN)
    elif timer == START_TRIAL:
        if currentTrial == 0 and ICONSET == 1:
            InitializeExp()
            counter = 0
            ans = 0
            vru.
            showTarget = allTarget[currentTrial]
            showFloor = allFloor[currentTrial]
            showEntry = allEntry[currentTrial]
            showDirection = allDirection[currentTrial]

            if showEntry == Z:
                TakeBreak()
            else:
frame1.curtain(vrut.OPEN)
frame2.curtain(vrut.CLOSE)
cardBegin.curtain(vrut.OPEN)

trialNum = trialNum + 1

vrut.watch('entry: %d' %showEntry)
vrut.watch('target: %d' %showTarget)
vrut.watch('floor: %d' %showFloor)
vrut.watch('trial: %d' %trialNum)

if ICONSET == 1:
    if showEntry == F:
        vruX.rotate(first, vrut.XAXIS, 90)
    elif showEntry == O:
        vruX.rotate(first, vrut.YAXIS, 180)
    elif showEntry == L:
        vruX.rotate(first, vrut.YAXIS, -90)
    elif showEntry == R:
        vruX.rotate(first, vrut.YAXIS, 90)
    elif showEntry == C:
        vruX.rotate(first, vrut.XAXIS, -90)
    else:
        if showEntry == F:
            vruX.rotate(second, vrut.XAXIS, 90)
        elif showEntry == O:
            vruX.rotate(second, vrut.YAXIS, 180)
        elif showEntry == L:
            vruX.rotate(second, vrut.YAXIS, -90)
        elif showEntry == R:
            vruX.rotate(second, vrut.YAXIS, 90)
        elif showEntry == C:
            vruX.rotate(second, vrut.XAXIS, -90)

elif timer == SHOW_TARGET:
    iCard[6].curtain(vrut.OPEN)
    if ICONSET == 1:
        iCard[showTarget].translate(0, 0, .25)
        iCard[showTarget].curtain(vrut.OPEN)
    else:
        iCard2[showTarget].curtain(vrut.OPEN)
    vruX.starttimer(SHOW_STIMULUS, TARGET_TIME)

elif timer == SHOW_STIMULUS:
    iCard[showTarget].curtain(vrut.CLOSE)
    iCard2[showTarget].curtain(vrut.CLOSE)
    iCard[6].curtain(vrut.CLOSE)
    startMemory = time.time()
    showRotate = allAngle[rotate]
    vruX.watch('rotate = %d' %rotate)
    if showRotate == 3:
        showRotate = -90
    elif showRotate == 9:
        showRotate = 90
    elif showRotate == 6:
        showRotate = 180
    rotRoom = -showRotate

    if rotate < len(allAngle):
        rotate = rotate + 1

    mask.curtain(vrut.OPEN)
    Arrows[3].curtain(vrut.OPEN)
    Arrows[4].curtain(vrut.OPEN)

if ICONSET == 1:
vrut.rotate(first, vrut.ZAXIS, rotRoom)
first.curtain(vrut.OPEN)
else:
vrut.rotate(second, vrut.ZAXIS, rotRoom)
second.curtain(vrut.OPEN)

vrut.starttimer(MEMORY_TASK, STIMULUS_TIME)

elif timer == MEMORY_TASK:
if ICONSET == 1:
first.curtain(vrut.CLOSE)
else:
second.curtain(vrut.CLOSE)
mask.curtain(vrut.CLOSE)
Arrows[3].curtain(vrut.CLOSE)
Arrows[4].curtain(vrut.CLOSE)

counter = counter + 1
if counter<(MEMORY_TIME*MEMORY_TIME):
vrut.starttimer(MEMORY_TASK, 1/SEARCH_TIME)
else:
vrut.starttimer(END)

elif timer == END:
vrut.watch('End of a Trial')
if ICONSET== 1:
vrut.rotate(first, vrut.ZAXIS, -rotRoom)
if showEntry == F:
vrut.rotate(first, vrut.XAXIS, -90)
elif showEntry == O:
vrut.rotate(first, vrut.YAXIS, 180)
elif showEntry == L:
vrut.rotate(first, vrut.YAXIS, 90)
elif showEntry == R:
vrut.rotate(first, vrut.YAXIS, -90)
elif showEntry == C:
vrut.rotate(first, vrut.XAXIS, 90)
else:
vrut.rotate(second, vrut.ZAXIS, -rotRoom)
if showEntry == F:
vrut.rotate(second, vrut.XAXIS, -90)
elif showEntry == O:
vrut.rotate(second, vrut.YAXIS, 180)
elif showEntry == L:
vrut.rotate(second, vrut.YAXIS, 90)
elif showEntry == R:
vrut.rotate(second, vrut.YAXIS, -90)
elif showEntry == C:
vrut.rotate(second, vrut.XAXIS, 90)
currentTrial = currentTrial + 1

if currentTrial < len(allTarget):
goNextTrial = TRUE
vrut.starttimer(START_TRIAL, .1)
else:
iCard[9].curtain(vrut.OPEN)
print 'Its all over folks!' if ICONSET == 1:
ICONSET = ICONSET + 1
currentTrial = 0
rotate = 0
trialNum = 0
data.write('%End of first session for ' + SUBJECT + '
' + '
')
data.write('%Beginning of second session for ' + SUBJECT + '
')
data.flush()
vrut.callback(vrut.KEYBOARD_EVENT, 'ExptKey')
vrut.callback(vrut.TIMER_EVENT, 'ExptTimer')
vrut.starttimer(BEFORE, .1)
else:
    iCard[10].curtain(vrut.OPEN)

#********************************************************************************
#Subject inputs for Experimental Trials
#********************************************************************************

def ExptKey(key):
    global data
global goNextTrial
global currentTrial
global rotate
global counter
global trialNum

time.sleep(0.1)
endTime = time.time()
#vrut.watch('Response Time: %d' %(endTime-startTime))
vrut.watch('task: %d' %(TASK))

if key == '1' and HMD == 1:
    ms.reset()
    win32api.Sleep(200)
    vrut.watch('tracker has been reset')

if TASK == BEFORE:
    if key == ' ':
        if ICONSET == 1:
            iCard[8].curtain(vrut.CLOSE)
        else:
            iCard[9].curtain(vrut.CLOSE)
    vrut.starttimer(START_TRIAL, .1)

elif TASK == START_TRIAL:
    #******************************************************************************
    #Set trial to beginning of Stage 2 (2nd back surface)
    #******************************************************************************
    if key == 'a':
        currentTrial = 21
        rotate = 21
        trialNum = 22

        if ICONSET == 1:
            if showEntry == F:
                vrut.rotate(first, vrut.XAXIS, -90)
            elif showEntry == O:
                vrut.rotate(first, vrut.YAXIS, 180)
            elif showEntry == L:
                vrut.rotate(second, vrut.YAXIS, 90)
            elif showEntry == R:
                vrut.rotate(second, vrut.YAXIS, -90)
            elif showEntry == C:
                vrut.rotate(second, vrut.XAXIS, 90)
            else:
                if showEntry == F:
                    vrut.rotate(second, vrut.XAXIS, -90)
                elif showEntry == O:
                    vrut.rotate(second, vrut.YAXIS, 180)
                elif showEntry == L:
                    vrut.rotate(second, vrut.YAXIS, 90)
                elif showEntry == R:
                    vrut.rotate(second, vrut.YAXIS, -90)
                elif showEntry == C:
                    vrut.rotate(second, vrut.XAXIS, 90)

    vrut.starttimer(START_TRIAL, .1)
elif key == 'b':
    currentTrial = 10
    rotate = 10

if ICONSET == 1:
    if showEntry == F:
        vrut.rotate(first, vrut.XAXIS, -90)
    elif showEntry == O:
        vrut.rotate(first, vrut.YAXIS, 180)
    elif showEntry == L:
        vrut.rotate(first, vrut.YAXIS, 90)
    elif showEntry == R:
        vrut.rotate(first, vrut.YAXIS, -90)
    elif showEntry == C:
        vrut.rotate(first, vrut.XAXIS, 90)
else:
    if showEntry == F:
        vrut.rotate(second, vrut.XAXIS, -90)
    elif showEntry == O:
        vrut.rotate(second, vrut.YAXIS, 180)
    elif showEntry == L:
        vrut.rotate(second, vrut.YAXIS, 90)
    elif showEntry == R:
        vrut.rotate(second, vrut.YAXIS, -90)
    elif showEntry == C:
        vrut.rotate(second, vrut.XAXIS, 90)

vrut.starttimer(START_TRIAL, .1)

elif key == ' ':  
    cardBegin.curtain(vrut.CLOSE)
    vrut.starttimer(SHOW_TARGET, .1)

elif TASK == TAKE_BREAK:
    if key == ' ':
        if rotate < len(allAngle):
            rotate = rotate + 1

    if currentTrial < len(allTarget):
        currentTrial = currentTrial + 1
        goNextTrial = TRUE

# Hide the break card
ifCard[7].curtain(vrut.CLOSE)
vrut.starttimer(START_TRIAL, .1)

elif (TASK == SHOW_STIMULUS or TASK == MEMORY_TASK):
    if (key=='4' or key=='5' or key=='6' or key=='8'):
        ans = 1
    vrut.watch('response registered')
    #sound a "beep"
    win32api.Beep(1300, 90)
    #write the data to file
    counter = MEMORY_TIME*MEMORY_TIME
    outdata = SUBJECT+'t'+str(ICONSET)+'t'+str(trialNum)+'t'+str(endTime-startMemory)
    outdata = outdata+'	'+str(showEntry)+'t'+str(showTarget)+'t'+str(showRotate)+'t'+str(showDirection)+'
    data.write(outdata)
    data.flush()

    # elif TASK == SEARCH_TASK:
    #    if key=='0':
    #        counter = SEARCH_TIME*SEARCH_TIME
    #        outdata = SUBJECT+'t'+str(ICONSET)+'t'+str(trialNum)+'t'+str(endTime-startSearch)+'t'
    #        outdata = outdata+'	'+str(showEntry)+'t'+str(showTarget)+'t'+str(showRotate)+'t'+str(showDirection)+'
    #        data.write(outdata)
    #        data.flush()
else:
    return

#*****************************************************************************
#     Break for Subject
#*****************************************************************************

def TakeBreak():
    global TASK

    # show take break card
    iCard[7].curtain(vrut.OPEN)
    TASK = TAKE_BREAK

    #*****************************************************************************
    #*****************************************************************************

    vrut.callback(vrut.KEYBOARD_EVENT, 'ExptKey')
    vrut.callback(vrut.TIMER_EVENT, 'ExptTimer')
    vrut.starttimer(START_TRIAL, 0.5)
APPENDIX B

Instructions

There were four sets of instructions, one for each group as we employed different direction terminologies. For brevity, only instructions for Group OAS and Group ISA are included in this appendix. The two other sets can be deduced easily.

Each set was composed of 22 slides. Instruction Set 1 (Figure 2.5) consisted of the first 17 slides, and were read at the beginning of the experiment, before the training phase trials. Instruction Set 2 consisted of the last 5 slides, and was read at the middle of the experiment, between the training and the testing phase trials.
1. Instructions for Group OAS

This group took an Outside point-of-view to the virtual environment and performed the orientation task with the Array-rotation mode first in training, then changed to the Self-rotation mode in testing.

1.1 Instruction Set 1 provided to subjects at the beginning of the experiment session (Group OAS)

These first instructions described the general trial timeline, the trial sequence of the training phase and explained in details how to handle the required training rotation mode.

---

**Slide 1**

3-Dimensional Spatial Learning
OUTSIDE
a Virtual Space Station Node

---

**Slide 2**

One astronaut recalled: “After I first boarded MIR, I decided to go back to the Shuttle, but discovered I didn’t know which way to go, since I hadn’t left behind any bread crumbs!”.

These navigation and orientation problems, specific to microgravity seem to be due to the station’s complex architecture. For instance, the former MIR consisted of six different modules, all connected together by a central hub called a ‘node’.

“Learning to recognize the correct hatch to go through is difficult”, astronauts say, “because you can be floating in practically any orientation..”
This experiment, sponsored by NASA, is designed to study how people learn to do an analogous task: learning the arrangement of six different objects, or pictures on the walls of a virtual cubic room.

Essentially, we will put you OUTSIDE this virtual room, as if you were about to enter it, and will show it to you from several different orientations. In that way, we will simulate the orientation conditions of weightlessness.

Your task will be to learn the arrangement of the pictures well enough so you can mentally visualize it (i.e. "see it in your mind’s eye") and predict where each picture is regardless of orientation.

---

**Slide 4**

First phase of the experiment:

You remain stationary.

You imagine that the ROOM is rotating in front of you.

Remember: in all the schemes of these instructions, YOU are the astronaut and you are looking at the scene through his/her eyes.

Notice that the walls are transparent. So, you can consider you are capable of X-ray vision, as if you could see through the walls of the rooms.
One example! Imagine you are standing in front of a small house trailer. A crane is in charge of picking up the trailer and rotating it.

Now, what would the kitchen look like from outside, if the trailer were rotated 180° about the vertical axis?

You remain stationary OUTSIDE.

---

We highly recommend that you find out and memorize the "BASELINE ORIENTATION" of the room.

So, imagine the virtual room can rotate about its three axes, as if you manipulated it in front of you, while you are standing stationary and upright.

Try to mentally visualize the rotating room in different orientations. Of course, it’s impossible to remember for each orientation the arrangement of all the 6 pictures.

Consequently, it is easier to get one’s bearings according to some single reference view that we will call the Baseline Orientation of the room.
Slide 7

This particular reference view will be the first orientation you master in the experiment.

Memorizing the Baseline Orientation requires that you first name the pictures and remember them in terms of their locations relative to the room:

1) What is on the CEILING?
2) What is on the FLOOR?
3) What is on the LEFT wall?
4) What is on the RIGHT wall?
5) What is on the FRONT wall?
6) What is on the BACK wall?

Slide 8

We’ll first allow you to observe a practice virtual-room similar to the one you’ll encounter in the experiment. These are NOT the pictures you will see in the real experiment. But so, you can get used to wearing the Head Mounted Display and looking around in a virtual environment.

Make sure you look around in all directions, and try to imagine that you’re actually standing upright in front of this virtual room.

When you see the environment for the first time, make a real attempt to establish a whole picture of this precise configuration of the pictures in your mind, as explained in the previous slide. This will be considered the Baseline Orientation of the practice room.
Let’s try visualizing the room being in a different orientation, NOT the Standard Orientation...

First, visualize the room in its Standard Orientation. Now, suppose the room has just been tilted 90 degrees counterclockwise from the Standard Orientation.

Can you “see in your mind’s eye” where the pictures will appear on the different walls? Take a look and find out. Let’s see if you’re right...

Visualize the ROOM ROTATING in front of you while you are standing upright and stationary outside it.

Name the 6 different pictures. Don’t hesitate to speak aloud.

---

Experimental Setup

**Step 1:**

Before each trial, you will see a card that says **BEGIN**, indicating that a new trial will begin when you press the spacebar.

Once you press the spacebar, you’ll see one of the pictures appear on a green background with the word **TARGET** written on it. It will disappear after 2 SECONDS. You need to remember this picture: it is the target that you will attempt to locate...

Be careful, it is in fact really easy to forget which target you’re looking for!
**Step 2:**

Next, **only 2** of the pictures in the room will be uncovered:
- the one **far in front**
- the one **below**

with arrows pointing to them. They will be visible for only **3 SECONDS**.

During this time, you should try to imagine how the room was rotated relative to you standing upright OUTSIDE as inferred from the 2 pictures you just saw. Try then to visualize where the target is.

You can make a response during this time if you already know the location of the target.

---

**Step 3:**

For the next **6 SECONDS**, all of the pictures will be covered up again. If you haven't made a response -- indicated where you think the target is located -- you should try to do that during these 6 seconds.

Try to respond as fast as possible without sacrificing accuracy.

In any case, there are only 4 possible target locations:

- **ORANGE** = response keys
- **GREEN** = terminate trial
- **GREY** = not used

"to wall above"  "to left wall"  "to front wall"  "to right wall"
Step 4: Finally...
For the last 8 SECONDS of the trial, all of the pictures will be uncovered, except the one closest to you (of course, you will be able to guess that one). In that way, you will see what the orientation of the room was and so, where the target picture was.

*Once you become very confident in your spatial memory, you can move on to the next trial before the 7 seconds is up by pressing the '0' key as shown in the diagram. Please, do not do this unless you really are confident.

Familiarization Exercise: To get familiar with the keys, you will try to match arrows that appear in the room with the corresponding response keys, as fast as possible.

Practice Trials: Then, you will do 3 practice trials outside the practice room you saw earlier. The trials will proceed automatically as described in Experimental Setup. This is mainly so you can get used to the timing of the trials. Don't worry too much about trying to memorize this practice configuration.
**Train Phase:**
You mentally visualize that the ROOM is rotating in front of you.
You remain stationary and upright.

---

**Trials**

**Trials 1 - 4** The room will be put in its Baseline Orientation. So, the orientation of the room will be the same in each of these trials. No matter of rotation at all.

This is when you should learn your Baseline Orientation.

**Trials 5 - 12** First, a new picture will hang up on the wall far in front of you. Then, the only difference among these 8 trials is that the simulated tilt orientation (roll) of the room will vary randomly.

**Possible Roll Angles**

**Trials 13 - 36** The room can be in any possible orientation; i.e. tilted by any angle while any picture is on the wall far in front of you.
Your ultimate goal for the last 24 trials is to be able to **mentally visualize the room** in front of you in any orientation.

Take as much time as you like. Concentrate on mentally visualizing and on memorizing the room between trials.

Pay attention to the first 4 trials. They are crucial because they will establish the baseline orientation of the room. Don’t miss them!

You can make only one guess in each trial -- we’ll take your *first* guess and that’s all. So be sure of your answer before pressing a key.

**GOOD LUCK!**

Subjects stopped here and began the training phase.

**1.2 Instruction Set 2 provided to subjects before the testing phase began (Group OAS)**

These instructions were read after the subjects completed the questionnaire about the training phase and before the testing phase began. They described the trial sequence of the testing phase and explained quickly the second rotation mode.

**Slide 18**

**Trials continued...**

**Testing Phase:**
You mentally visualize that YOU are revolving around the room and rotating about yourself.

The room remains stationary.
Notice the strategy in this new part is different because you are trying to imagine not the room BUT yourself rotating. Anyway, what you have learned the first part should help.

We are interested in knowing how easily you can switch from one mental process to the other.

So, imagine:
- You can revolve along two circular paths around the cube.
- You can also rotate about yourself, as if you were doing a cartwheel (roll).

NO feedback will be provided in this part.

---

One example! Imagine you are standing in front of a small house trailer. You can climb on any wall, in any position, even upside down!

You are rotating outside around the stationary room.

Now, what would this kitchen look like from outside, if you were in a tilted orientation facing a new wall, as it happens in weightlessness?
Slide 21

Visualize YOURSELF ROTATING outside the stationary room (this simulates weightlessness!).

Name the 6 different pictures. Don’t hesitate to speak aloud.

Trials 1 - 24 You may be in any possible orientation outside the room, i.e. facing any wall while tilted by any angle (roll).

Slide 22

Your ultimate goal for the last 24 trials is to be able to mentally visualize yourself outside the room in any orientation.

Take as much time as you like. Concentrate on mentally visualizing your imagined body position and on memorizing the room between trials.

You can make only one guess in each trial -- we’ll take your first guess and that’s all. So be sure of your answer before pressing a key.

Thank you for your participation!
2. Instructions for Group ISA
This group took an Inside point-of-view to the virtual environment and performed the orientation task with the Self-rotation mode first in training, then changed to the Array-rotation mode in testing.

2.1 Instruction Set 1 provided to subjects at the beginning of the experiment session (Group ISA)

Slide 1

3-Dimensional Spatial Learning
INSIDE
a Virtual Space Station Node

Slide 2

One astronaut recalled: "After I first boarded MIR, I decided to go back to the Shuttle, but discovered I didn’t know which way to go, since I hadn’t left behind any bread crumbs!".

These navigation and orientation problems, specific to microgravity seem to be due to the station’s complex architecture. For instance, the former MIR consisted of six different modules, all connected together by a central hub called a ‘node’.

"Learning to recognize the correct hatch to go through is difficult", astronauts say, "because you can be floating in practically any orientation."
This experiment, sponsored by NASA, is designed to study how people learn to do an analogous task: learning the arrangement of six different objects, or pictures on the walls of a virtual cubic room.

Essentially, we will put you INSIDE this virtual room, and will show it to you from several different orientations. In that way, we will simulate the orientation conditions of weightlessness.

Your task will be to learn the arrangement of the pictures well enough so you can mentally visualize it (i.e. "see it in your mind’s eye") and predict where each picture is regardless of orientation.

First phase of the experiment:

The room remains stationary.

You imagine that you are rotating inside it.

Just imagine your point of view is changing.
One example! Imagine you are inside a bedroom and you have the skills of an acrobat hung at the end of the rope. So, you can make your body be in different orientations.

Now, what would this bedroom look like if you were in a tilted orientation facing a new wall, as it happens in weightlessness?

We highly recommend that you find out and memorize your "BASELINE ORIENTATION". So, imagine your body orientation is changing, as in weightlessness. You can rotate about yourself, more precisely about all your 3 body axes. For example, you can do a cartwheel (roll).

Try to mentally visualize yourself being in different orientations. Of course, it’s impossible to remember for each of your body orientations the arrangement of all the 6 pictures.

Consequently, it is easier to get one’s bearings according to some single reference view that we will call your Baseline Orientation. In other words, this is the appearance of the room while you are standing upright, so while there is NO rotation of your body.
This particular reference view will be the first orientation you master in the experiment.

Memorizing the Baseline Orientation requires that you first name the pictures and remember them in terms of their locations relative to your body:

1) What is **ABOVE** you?
2) What is **BELOW** you?
3) What is on your **LEFT**?
4) What is on your **RIGHT**?
5) What is in **FRONT** of you?
6) What is in your **BACK**?

We’ll first allow you to observe a practice virtual-room similar to the one you’ll encounter in the experiment. These are **NOT** the pictures you will see in the real experiment. But so, you can get used to wearing the Head Mounted Display and looking around in a virtual environment.

Make sure you look around in all directions, and try to imagine that you’re actually standing upright inside this virtual room.

When you see the environment for the first time, make a real attempt to establish a whole picture of this precise configuration of the pictures in your mind, as explained in the previous slide. This will be considered **your Baseline Orientation** for the practice room.
Let’s try visualizing yourself being in a different orientation, NOT the Baseline Orientation ...

First, visualize yourself in the Baseline Orientation. Now, suppose you’d just been tilted right-shoulder down (90 degrees clockwise) from the Baseline Orientation.

Can you “see in your mind’s eyes” where the pictures will appear relative to your body? Take a look and find out. Let’s see if you’re right...

Visualize YOURSELF ROTATING inside the stationary room (this simulates weightlessness!).

Name the 6 different pictures. Don’t hesitate to speak aloud.

---

**Experimental Setup**

**Step 1:**

Before each trial, you will see a card that says **Begin**, indicating that a new trial will begin when you press the spacebar.

Once you press the spacebar, you’ll see one of the 6 pictures appear on a green background with the word **TARGET** written on it. It will disappear after 2 SECONDS. You need to remember this picture: it is the target that you will attempt to locate...

Be careful, it is in fact really easy to forget which target you’re looking for!
Step 2:

Next, only 2 of the pictures in the room will be uncovered:
- the one in front
- the one below
with arrows pointing to them. They will be visible for only 3 SECONDS.

During this time, as inferred from the 2 pictures you just saw, you should try to imagine how you rotated relative to the Baseline Orientation, so what your new orientation is INSIDE the stationary room. Try then to visualize where the target is.

You can make a response during this time if you already know the location of the target.

Step 3:

For the next 6 SECONDS, all of the pictures will be covered up again. If you haven’t made a response -- indicated where you think the target is located -- you should try to do that during these 6 seconds.

Try to respond as fast as possible without sacrificing accuracy.

In any case, there are only 4 possible target locations:

ORANGE = response keys
GREEN = terminate trial
GREY = not used

“to wall above”
“to left wall”
“to behind wall”
“to right wall”
Step 4: Finally...

For the last 8 SECONDS of the trial, all of the pictures will be uncovered, except the one behind you (of course, you will be able to guess that one). In that way, you will see what the whole room looks like with this new orientation of your body and so, where the target picture was.

*Once you become very confident in your spatial memory, you can move on to the next trial before the 7 seconds is up by pressing the '0' key as shown in the diagram. Please, do not do this unless you really are confident.

Familiarization Exercise: To get familiar with the keys, you will try to match arrows that appear in the room with the corresponding response keys, as fast as possible.

Practice Trials: Then, you will do 3 practice trials inside the practice room you saw earlier. The trials will proceed automatically as described in Experimental Setup. This is mainly so you can get used to the timing of the trials. Don’t worry too much about trying to memorize this practice configuration.
Training Phase:
You mentally visualize that YOU are rotating about yourself inside the room.

The room remains stationary.

---

**Trials**

**Trials 1 - 4** You will be put in the Baseline Orientation. So, your orientation inside the room will be the same in each of these trials: you are standing upright. You always face the same direction, there is no matter of rotation at all.

*This is when you should learn your Baseline Orientation.*

**Trials 5 - 12** First, you will face a new direction in the room. Then, the only difference among these 8 trials is that the simulated lift orientation (roll) of your body will vary randomly.

**Possible Roll Angles**

**Trials 13 - 36** You can be in any possible orientation; i.e. facing any direction while tilted by any angle.
Your ultimate goal for the last 24 trials is to be able to mentally visualize yourself inside the room in any orientation.

Take as much time as you like. Concentrate on mentally visualizing your imagined body orientation and on memorizing the room between trials.

Pay attention to the first 4 trials. They are crucial because they will establish your Baseline Orientation. Don't miss them!

You can make only one guess in each trial -- we'll take your first guess and that's all. So be sure of your answer before pressing a key. GOOD LUCK!

2.2 Instruction Set 2 provided to subjects before the testing phase began (Group ISA)

Trials continued...

Testing Phase:
You mentally visualize that the ROOM is rotating all around you.

You remain stationary and upright inside the room.
Notice the strategy in this new part is different because you are trying to imagine not yourself but the room rotating. Anyway, what you have learned in the first part should help.

We are interested in knowing how easily you can switch from one mental process to the other.

So, imagine the virtual room can rotate about its three axes around you, as if you manipulated it, while you are standing stationary and upright.

NO feedback will be provided in this part.

---

One example! Imagine you are standing inside a small house trailer. A crane is in charge of picking up the trailer and rotating it.

Now, what would the bedroom look like from inside, if the trailer were rotated upside down?
Slide 21

**Visualize the ROOM ROTATING all around you while you are standing upright and stationary inside it.**

*Name the 6 different pictures. Don’t hesitate to speak aloud.*

**Trials 1 - 24** The room can be in any possible orientation, i.e., tilted by any angle (roll) while any picture is on the wall in front of you.

**Possible Roll Angles**

---

Slide 22

Your ultimate goal for the last 24 trials is to be able to **mentally visualize the room around you in any orientation.**

Take as much time as you like. Concentrate on mentally visualizing and on memorizing the room between trials.

You can make only one guess in each trial -- we’ll take your *first* guess and that’s all. So be sure of your answer before pressing a key.

**Thank you for your participation!**
APPENDIX C

Questionnaires

Included in this appendix are the two questionnaires to be completed by Group OAS’s subjects. This group took an Outside point-of-view to the virtual environment and performed the orientation task with the Array-rotation mode in training, then changed to the Self-rotation mode in testing.

The questionnaires given to subjects of Group ISA, Group IAS and Group OSA were similar, except that order of rotation mode, terminology and schemes were adapted for each group.

Questionnaire 1 was completed at the end of the training phase.

Questionnaire 2 was completed at the end of the testing phase. The first questions were similar to those of Questionnaire 1, but concerned the rotation mode adopted in the testing phase. The other questions asked about comparisons between phases, but also concerned various strategies that could not be mentioned in Questionnaire 1 (they might have influenced the subjects in the subsequent testing phase).

The experimenter was present to answer any questions the subject had about how to answer the questionnaires.
Questionnaire 1

*For some questions, we would like you to use a rating scale:*

*Example: Did you happen to be sick during the experiment?*

![Rating Scale]

1- *Which strategies did you use in this first phase when you had to imagine that the ROOM was rotating in front of you while you stood stationary and upright? How much percentage of the time?*

<table>
<thead>
<tr>
<th>Strategy Description</th>
<th>0</th>
<th>5</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>You referred to the Baseline orientation as a reference configuration.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>You referred to other specific orientations. Which ones?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>You used rules: you remembered opposite pairs.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>You used rules: you remembered triads (corner).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>You used rules: you remembered rings of pictures.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>You answered by chance.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

0: Never  
10: 100% of the time
<table>
<thead>
<tr>
<th>Mentally you made the room rotate while you remained stationary.</th>
<th>![0 5 10]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Was it a two-step strategy? Explain.</td>
<td>□ Yes □ No</td>
</tr>
<tr>
<td>Did you perceive the imagined motion of the room..</td>
<td>□ Yes □ No</td>
</tr>
<tr>
<td>• ..as continuous [the transition between one orientation and the other was smooth, as if you physically manipulated it and had seen the intermediate stages]?</td>
<td>□ Yes □ No</td>
</tr>
<tr>
<td>• ..as discrete [it had nothing to do with a physical motion, you just saw the room in one orientation, and then directly in another, as if there were missing stages]?</td>
<td>□ Yes □ No</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Mentally you made yourself rotate while the room remained stationary.</th>
<th>![0 5 10]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Was it a two-step strategy? Explain.</td>
<td>□ Yes □ No</td>
</tr>
<tr>
<td>Did you perceive the imagined motion of the room..</td>
<td>□ Yes □ No</td>
</tr>
<tr>
<td>• ..as continuous [the transition between one orientation and the other was smooth, as if you physically manipulated it and had seen the intermediate stages]?</td>
<td>□ Yes □ No</td>
</tr>
<tr>
<td>• ..as discrete [it had nothing to do with a physical motion, you just saw the room in one orientation, and then directly in another, as if there were missing stages]?</td>
<td>□ Yes □ No</td>
</tr>
</tbody>
</table>
2- How would you describe the task under each of the following configurations in the first phase? [You had to imagine that the room was rotating in front of you while you stood stationary and upright]

- Rotation of the room about the X-axis
- Rotation of the room about the Y-axis
- Rotation of the room about the Z-axis
- Target Above you
- Target in Front of you
- Target on your Left
- Target on your Right

Did you find an orientation of the room particularly hard to deal with? [As for example when ‘the giraffes were on the furthest wall]
Questionnaire 2

1- How would you describe the task under each of the following configurations in the second phase? [You had to imagine that you were revolving around the room and rotating about yourself outside the stationary room]

- Rotation of your body around the room in the vertical plane

- Rotation of your body around the room in the horizontal plane

- Rotation of your body about itself (Roll)

- Upright

- Upside-down

- Left/right shoulder down

- Target Above you

- Target in Front of you

- Target on your Left

- Target on your Right

0 : Trivial
10: Impossible
Did you find an orientation of your body particularly hard to deal with? [As for example when ‘you were behind the furthest wall, upside down’?]

2- Which strategies did you use in this second phase when you had to imagine that YOU were revolving around the room and rotating about yourself outside the stationary room? How much percentage of the time?

<table>
<thead>
<tr>
<th>Strategy Description</th>
<th>Percentage Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>You referred to the Baseline orientation as a reference configuration.</td>
<td>0: Never 10: 100% of the time</td>
</tr>
<tr>
<td>You referred to other specific orientations. Which ones?</td>
<td></td>
</tr>
<tr>
<td>You used rules: you remembered opposite pairs.</td>
<td></td>
</tr>
<tr>
<td>You used rules: you remembered triads (corner).</td>
<td></td>
</tr>
<tr>
<td>You used rules: you remembered rings of pictures.</td>
<td></td>
</tr>
<tr>
<td>You answered by chance.</td>
<td></td>
</tr>
</tbody>
</table>
Mentally you made the room rotate while you remained stationary.

Was it a two-step strategy? Explain.

Did you perceive the imagined motion of the room..
- ...as continuous [the transition between one orientation and the other was smooth, as if you physically manipulated it and had seen the intermediate stages]?
- ...as discrete [it had nothing to do with a physical motion, you just saw the room in one orientation, and then directly in another, as if there were missing stages]?

Mentally you made yourself rotate while the room remained stationary.

Was it a two-step strategy? Explain.

Did you perceive the imagined motion of the room..
- ...as continuous [the transition between one orientation and the other was smooth, as if you physically manipulated it and had seen the intermediate stages]?
- ...as discrete [it had nothing to do with a physical motion, you just saw the room in one orientation, and then directly in another, as if there were missing stages]?
3- Which of those two strategies did you find easier?

☐ The one in which you made the room rotate while you remained stationary.

☐ The one in which you made mentally yourself rotate while the room remained stationary

4-

<table>
<thead>
<tr>
<th></th>
<th>1st Phase: When you had to imagine that the ROOM was rotating in front of you while you stood stationary and upright.</th>
<th>2nd Phase: When you had to imagine that YOU were revolving around the room and rotating about yourself outside the stationary room.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Did you switch from one strategy to another? (between room rotating and you rotating)</td>
<td>0 5 10</td>
<td>0 5 10</td>
</tr>
<tr>
<td>Did you take an additional external point-of-view: were you like a spectator of the entire scene that included a moveable model of yourself and the room together? (Did you use a two-viewpoint strategy, like those of a cameraman and an actor?)</td>
<td>0 5 10</td>
<td>0 5 10</td>
</tr>
<tr>
<td>Did you happen to perceive yourself INSIDE the room?</td>
<td>0 5 10</td>
<td>0 5 10</td>
</tr>
<tr>
<td>Did you use other strategies: which ones?</td>
<td>0 5 10</td>
<td>0 5 10</td>
</tr>
</tbody>
</table>
5- *Which phase did you find easier?*

☐ The first phase: when you had to imagine that the room was rotating in front of you while you stood stationary and upright
☐ The second phase: when you had to imagine that you were revolving around the room and rotating about yourself outside the stationary room
☐ The 2 phases were similar

*Please, explain.*

6- *Do you think your accuracy improved in the second phase [relative to the first]?*

![Accuracy Improvement Scale]

7- *How would you describe the change of strategy that was required between the first and the second phase of the experiment?*

![Strategy Change Scale]
If you had faced the conditions in the reverse order..  

[Reminder:  
If in Phase 1: you had to imagine that you were revolving around the room and rotating about yourself outside the stationary room, with feedback
If in Phase 2: you had to imagine that the room was rotating in front of you while you stood stationary and upright, with no feedback]

..how do you think you would have described the experiment?

8- Did the feedback in the first phase help you?

Did you miss having it in the second phase?

9- Were you able to “mentally visualize” the virtual room with all its pictures and without the help of any rules? Did you have a complete mental image of the room in your mind..

.. in the first phase?

.. in the second phase?
10- During the trials, were you distracted by any features of the real external world (the lab, as opposed to the virtual environment)?

If Yes, which ones?

☐ Discomfort of helmet
☐ Sounds, noises
☐ Chair perception
☐ Tiredness
☐ Others

11- How many hours per week do you play video/computer games?

☐ 0
☐ 1 to 3
☐ 3 to 5
☐ 5 to 10
☐ 10 to 20
☐ more than 20
APPENDIX D

Graphs of %-Incorrect by Set for each Group

Change in Rotation Mode

Group IAS

Group ISA