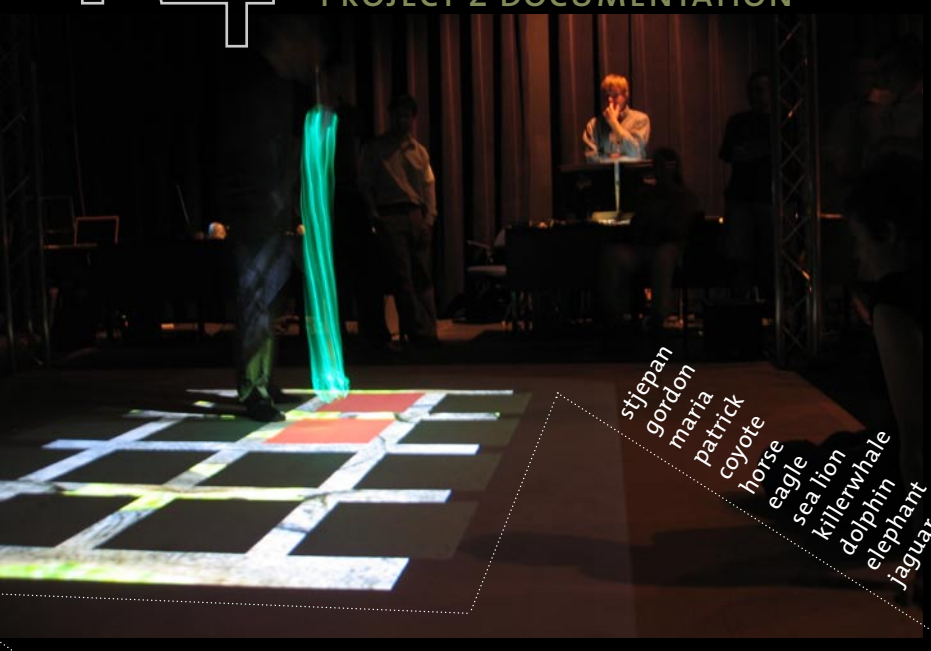
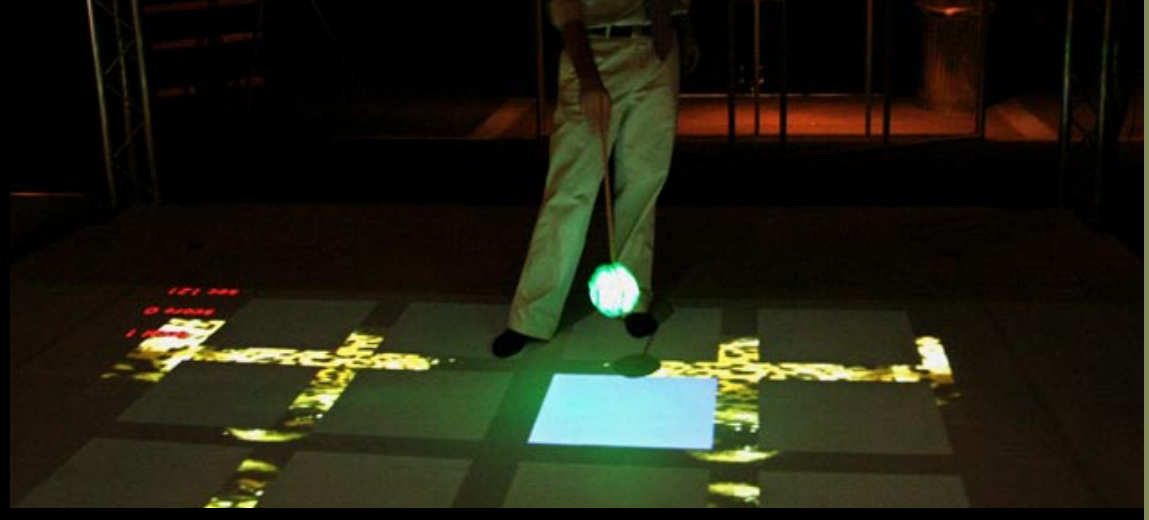
F4

PROJECT 2 DOCUMENTATION



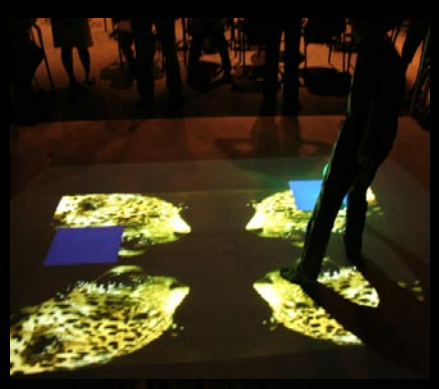


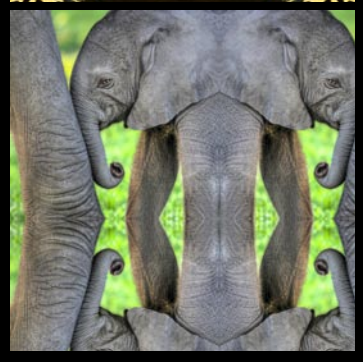
Our motivation in this project was to create a unique and user-friendly interface for exploring sights and sounds in SMALLab.

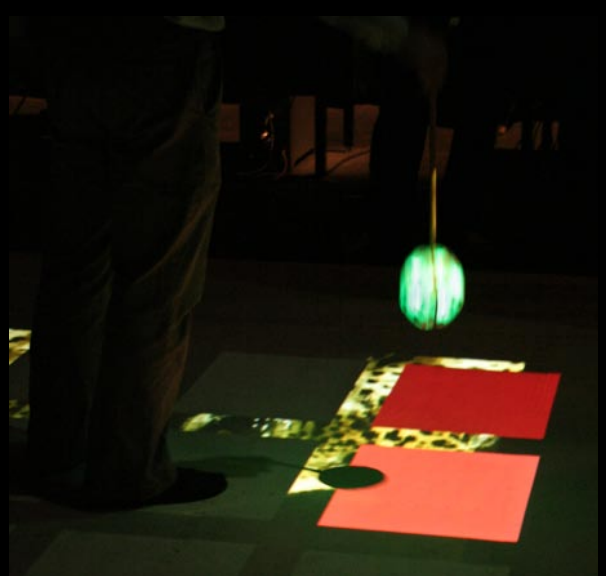
Our goal was the creation of a memory sound matching game in SMALLab, where the space was transformed into 16 "cards" evenly spaced in a four-by-four grid over the projection area. Each of the 16 cards had one of eight sound files attached to it, and the participant had to match the two cards that contained the same sound file. As the player successfully matched cards, those cards that were matched disappeared revealing a large background image in the bottom of the space. Additionally, we hoped to redefine the SMALLab glowing ball user-interface by attaching the ball to a stick, to provide the player with a "magic-wand-like" feeling while selecting cards.

The implementation of our project required the selection of a central theme for all of the sound files that were to be matched, and also the background images that are revealed as correct matches are achieved. We chose animals as our theme, because of the wide range of possible sounds and images available. Three images of animals were then edited into three collage-like images, and used as the backgrounds for the three levels of our memory game. The next step required the creation of a MAX/MSP render engine for conversion of the raw x-y coordinates coming from the tracking system into four-by-four grid coordinates, i.e., the player position in the space could be in only one of sixteen positions corresponding to the card locations. These grid points were then used as the input to a memory logic object, created using a custom MAX/MSP render engine built for this project. This render engine was responsible for playing the animal sound files, displaying the card and player location, and also for the calculation of all game logic. The game logic was responsible for allowing the game to be played in the following manner: (1) The player explores the space and the card that the ball is currently over will be lit up, (2) the player selects their first card by holding the ball lower than a two ft. height (z-axis) threshold, (3) the sound file corresponding to the card the player selected will play and that card will flash between red and blue to indicate that it was selected (4) the player repeats steps 1-3 for selecting their second card (5) if the two player selected cards contain the same sound file they will disappear, otherwise those cards will remain in play (6) the player repeats steps 1-5 until all matches are found and the background image becomes visible or time runs out.

The memory logic render engine was also responsible for level, time and scoring calculations. Our version of memory had three levels with different background images for each level. The player advances to the next level by matching all eight pairs of cards before a time limit. As the player advances to higher levels they are given less time to find all eight matches in the space, and the deck is reshuffled prior to the beginning of each level, i.e., the location of the cards in the space changes in a random fashion. Highly skilled players who advance past level 3 will the move into the fast action bonus round. The number of correct matches a player has obtained is displayed in the space as the "score", along with the current time and level. The player gains time for every correct match, and loses time for every incorrect match. If the player fails to clear a level in the allotted time, the deck is reshuffled and they are returned to level one.







Our contribution in this project was a novel way of using SMALLab in terms of game play and interface design. The height-based card selection is easily understood and facilitated by attaching the ball to the stick to create a magic wand. Furthermore, the memory sound matching game logic foundation we have created can have other uses such as matching pitch in music training, or matching words in language training.

One thing we noticed from the player interactions with our project was that the memory-interface design appeared to work very well, as everyone seemed to pick-up the game quickly. It appeared that some players had a hard time spatially associating sounds with their corresponding card location, and the use of positional audio could be a remedy for this, but we were not able to implement this due to time constraints. Implementing the memory game logic successfully proved difficult and required several iterations to fine-tune. Calibrating the four by four card location grid with the projector was also challenging, but ultimately successful as was the method of rotating background images as the player advanced/regressed in level. One problem with our interface is that when a single bad value from the visual tracking system with a z-coordinate below 0.2 is received, that causes the wrong card to be selected, which can be very frustrating for the player.

. Maria . Patrick . Stjepan . Gordon

ball modification, documentation, concept development

audio files and images processing, documentation, concept development

Max/Msp coding, written documentation, audio research, concept development

Max/Msp coding, project implementation, concept development