Table of the Gods: Development of a Multi-Touch App for Museum

Geneviève Lucet[*1], Víctor Godoy[2]

Institute of Aesthetic Research, National Autonomous of Mexico, Mexico City, Mexico

*[1]genevieve.lucet@gmail.com; [2]godoy.vic@gmail.com

***Abstract*- We describe the development of an application accomplished in a museum context to explain the varied representations, traits, and connections between some of the deities from the Mesoamerican pantheon. In order to attract a younger audience, a multi-user interface was implemented, with an intuitive interaction system based on graphic information and touch-screen technology.**

We explain the installed app, its conceptual design and its implementation with the Unity engine. The development has two main parts: the programming of the objects as intelligent agents with random and anti-collision movement; and the multi-touch screen programming, meaning the detection and interpretation of different kinds of touches and gestures and the changes they produce in objects behavior.

Keywords- Interface; Multitouch; Museum Application; Behavior

I. INTRODUCTION

For a long time, museums were regarded as places where pieces or collections valued through artistic criteria were exhibited to the public. This idea has changed in the second half of the 20th century, in the hopes that they become useful tools in the achievement of educational tasks towards a wide but predominantly young public [1]. Another goal is to improve the population's knowledge by motivating self-study and solving it interest about a wide range of topics.

Museums dedicated to the Hispanic cultures have also evolved, as evidenced by "The National Museum of Anthropology in Mexico" which was designed with a clear educational purpose. The founding of "The Museum of Tlatelolco" by the UNAM (National Autonomous University of Mexico) and INAH (National Institute of Anthropology and History) in 2011, attempted to integrate the multidisciplinary knowledge of history, anthropology, arts, science, and technology to show that the areas of science and humanities are integrated and combined in order to enrich and expand our approach to the past.

It is within this context that we developed a system to introduce some deities from the Mesoamerican pantheon. It's a complicated issue, there is a multitude of deities with varying degrees of importance and they are subject to changes, such as their names and representation according to the historical period and the local culture. Research on this subject has provided important insights for understanding the meaning, relevance and roles of prehispanic deities. We now know that they represent concepts, so the complexity of their definition and the links established between them can be measured by the sophistication of the Mesoamerican way of thinking. For the Museum, we simplified the information to include only a subset of deities and only some of their attributes.

In order to express this intricate way of thinking in an appealing way, we looked for a solution that has the following characteristics:

- Visual: communication through images is preferred by young people.
- Dynamic: to catch the users' attention and curiosity.
- Intuitive: with an accessible interface requiring little time to get used to.
- Interactive: the user becomes pro-active, thus increasing the cognitive process [2].
- Multi-user: to promote the dynamic participation of a group of people.

These requirements are common in many exhibitions, and various technological solutions are often used for enhanced educational effects; these include multimedia, systems built specifically and VR technologies [3].

II. CONCEPT DESIGN

In order to better explain Mesoamerican deities, we opted for dynamic imagery, representing them in motion, combined with a tactile system that allows several simultaneous users to select icons, drag them, and display information. The installation was designed as a horizontal board around which several people can interact at the same time (Fig. 1).

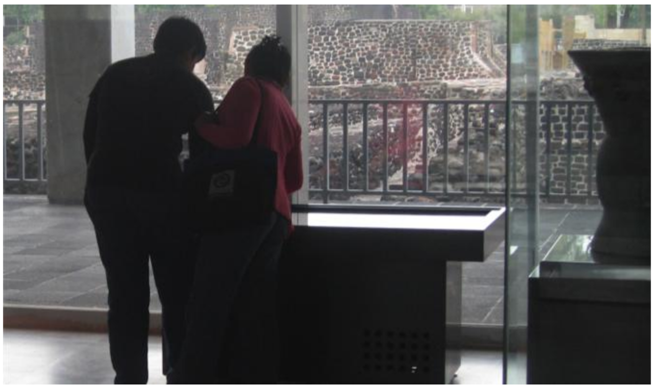

Fig. 1 The "Table of the Gods" installed in the Museum of Tlatelolco

The information about the deities was structured in a simple way: nine major deities manifest in motion on a background, where ten suns, eight stars and three comets randomly appear, soaring through and generating the contextual environment (Fig. 2).

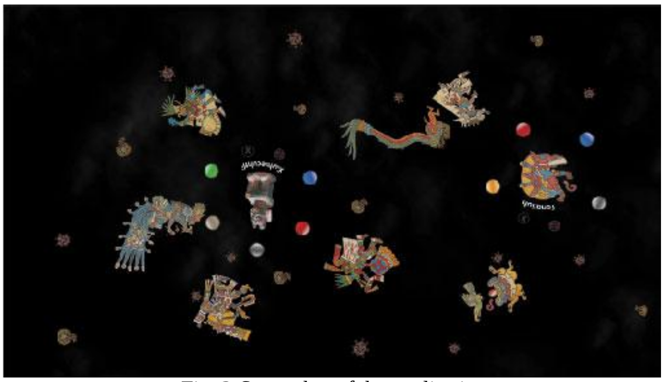

Fig. 2 Screenshot of the application

When the user touches, drags, or rotate one of the deities; its name is displayed, as well as a series of options in color circles around the deity. When touched, these options replace the central image with an alternate one and, depending on the god, may display a different representation; or show its characteristic attributes or connections to other deities; other options linked to each god image allow to change the language or to return to the original image.

III. BACKGROUND

The "Fiery Pool: The Maya and the Mythic Sea" sets a precedent to this work. Developed for the Peabody Museum in Essex in 2010 to exhibit the Mayan world view regarding water and its related deities and ancestors. This setup uses a projector and peripheral infrared sensors [4], it's a technical solution that requires a great deal of calibration, which complicates maintenance, so we opted for the more stable choice of a touchscreen device.

Touchscreen technology development started back in 1972 and IBM was the first company to market it, based on infrared technology, it contained 16 receptors within the screen. In 1984, the first multi-touch screen was further developed. Bill Buxton published research on human-computer interaction in 1986 [5][6] would mark this section of computer graphics developments for the following years. Research by the Mitsubishi Research Labs in 2001 revealed advances in complex gestures interpretation, which were followed by the works of Jefferson Han in 2005 and the creation of Perceptive Pixel Co [7]. At the same time, John Elias and Wayne Westerman developed multi-touch surfaces and the patents were bought by Apple Inc., who integrated all of these advances in the iPhone, in 2007. Thirty-five years had passed since the beginning of the research about computer-human interface, it had been necessary to overcome the different screen deficiencies, find stable and affordable technological solutions, a few years later, multitouch screen would be completely installed in our lifestyle.

Multi-touch systems recognize numerous points of contact and interpret them as an interface, meaning that many users can manipulate the display simultaneously with varied and independent events. These systems also record the pressure, speed and direction each point is sending, opening a variety of options that provide full interactivity through natural user gestures.

This feature represents an important change in application programming methods because the interpretation of multiple variables in pressure, gestures, and simultaneous events have to be taken into account.

There are several unresolved issues in multi-touch systems. The first is tied to traditional Operating Systems which are designed to work with a single point of interaction, so applications have been developed in order to capture the movement, interpret it, and communicate this information to the program, there are still, however, limitations in their potential as each manufacturer develops its own solution. The second issue resides in the lack of standardization in the language of gesture-based communication, although the iPad's and iPhone's popularity has promoted a more unified form of use.

IV. SOLUTION

The development must integrate solutions to: application-screen communication protocols; intelligent agent integration; object response to input and gesture; and multi-tactile recognition and interpretation.

A. *Multi-Touch Communication Protocols*

We opted for a 50' plasma screen, on top of which a PqLabs 32 touches Multi-tactile screen was mounted, so numerous users can interact at the same time and use various fingers to add up to 32 points of contact. The PqLabs screen is a transceiver kind frame consisting of two LEDs that emit an infrared beam and two photodiodes that receive it, being able to detect an area where the beam is interrupted by an object.

When the signal is interrupted by an obstacle, no matter the color, texture, shape or brightness of the object, the emitters recognize the (x,y) position on the screen and send the information via USB using TUIO or WM_TOUCH protocols, transmitting the interpretations of pressure and gestures from the tactile surface to the application. The TUIO protocol [8] (Table-top Tangible User Interface) recognizes an infinite amount of points of contact on the screen and transmits them to an application in OpenGL (Unix, Windows, and OSX) meanwhile WM_TOUCH only works with directX and just allows eight points of contact.

B. *The Virtual Space*

The application development and its integration to the interaction system was made possible by utilizing the Unity 3.0 engine, this program is being used more and more in the production of interactive material made for museums and the internet, since it offers developmental tools in multi-platform environments, as well as making possible the programming of object-oriented language scripts and enables the development of external extensions so as to meet specific needs, which has opened the possibility of making numerous plugins, several being open source. Its market acceptance is also a guarantee of continuity and updates for a few years.

This tool manages to integrate the graphic material in an organized manner. We associated behaviors to different objects and included interactive elements programed in C#. One group includes all the main components of the project: deities, stars, suns, comets, buttons, counters, virtual screen delimitations, information aspects, and animated background. Another group includes the elements that complement the environment and the system: camera, lights, and a multi-touch input manager.

Virtual Space Definition: each object moves in a random pattern within a determined space wider than the screen in order to let the object follows its course as it disappears and reenters the board again at another location. As the object detects this space's limits, it rotates and changes its course in the opposite direction. Simulated clouds appear in the background; this animation is calculated in real time with "billboard" particles, texturized with color.

C. Objects

Given their independent behavior and change in state, objects fall into three major categories: Deity-objects, Sun and Star-objects, and Comet-objects.

1) Properties and Traits:

All the elements found in the virtual space have an associated image to them. In addition, Deity-objects possess a name, buttons linked to secondary level information and language selection, and values set to simulate real-world-physics-dependable behavior.

2) Independent Behavior:

To program the object behavior and movement, we used Unitysteer [9,10], an open source solution developed by the Arges Systems, a consulting company in the development of games using Unity. It allows exerting control over generic objects' behavior required to be provided with independent movement, and it consists of a set of libraries based on OpenSteer and OpenSteerDotNet.

Except for the Comets, each of the objects that make up the application has the following properties (Fig. 3):

- Vehicle-like behavior, which allows establishing initial movement and defined speed.
- Random movement, which enables the object to have a soft, organic behavior and to simulate how it would behave in real world physics.
- Sonar or radar, which sets up each element's placement within the board, as well as detecting other objects or obstacles in proximity, in order to evade them, avoiding their piling up in just one spot. Each object takes into account the position as well as the direction of the other objects, calculates the trajectory and modifies its own route accordingly.
- Movement zone, which creates an allowed movement quadrant for each of the objects.

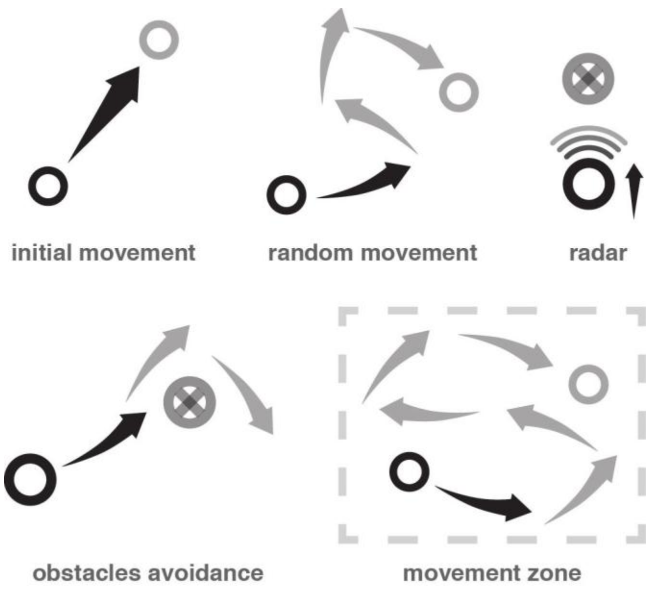

Fig. 3 Object behaviour

Sun-objects and Star-objects move in a separate layer, therefore avoiding collision with the Deity-objects. What distinguishes these two groups is a variation in speed.

Comet-objects possess a simpler definition; they appear on the board sporadically and follow a lineal path as they travel across. They're located on a layer separated from the other two and their course is not modified when they encounter another object. They have set and random movement properties, and are limited by the virtual space boundaries. Their movement speed is the highest among the objects and their random movement is the minimum required to obtain harmonic traversing.

D. Interactivity

Touching an object stops its independent movement and displays its information. Gestures are translated into different instructions according to the amount of points of contact on the board, as well as their location and the distance between them. With this information, the application can determine the amount of users and how many points are used by each one.

For each user, a single touch is translated as a cursor, while multiple touches are interpreted as different independent cursors. The user can drag an object with a touch; this combines rotation and translation following the orientation of the finger's movement. Two fingers are used to rotate an object and the fingers' turning direction determine the object's rotational direction in regard to the object's rotational center (Fig. 4).

We used uniTUIO for the multi-touch gesture-based communication between the screen and Unity. UniTUIO is free software, developed in 2009 by Sandor Rozsa and Stephan Schlupek. It is a group of C# libraries used to interpret the inputs sent through the touch screen and into Unity. Unity was originally conceived to deal with single-user events, which not only affects the amount of users that can interact simultaneously, but also gesture possibilities and object dynamics. UniTIUO was specifically designed to resolve these problems, integrating multi-touch events to the object selection that Unity already possesses. It incorporates new input events: doTouchDown, doTouchUP, doTouchOutside, singleTouch, doubleTouch and manyTouch, being able to work with various simultaneous events, and making possible for an object to interact with several touches in an accurate way.

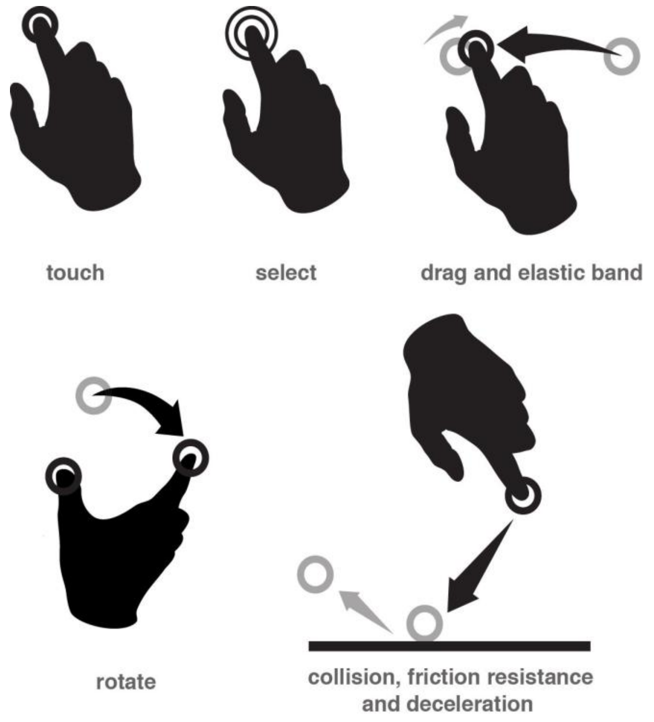

Fig. 4 Touch gesture

Given the interaction and object dynamics specifications this application has, we established direct communication with uniTUIO developers, so as to add physic properties to the objects and improve their response to input. The objects' movement speed, acceleration, friction, force absorption, and direction of drag properties was modified and adjusted. Likewise, two-finger object manipulation, multiple-pointer (cursor) object selection and dynamic movement response to different applied inputs was included.

The parameters are: mass, drag value, drag angle, drag distance, absorption and collision reduction and recoil. The mass value serves as a weight simulator enabling the object to behave as heavy or light. The drag value establishes friction resistance and deceleration of objects in motion. The drag angle defines the friction resistance to rotation. The objects have, in addition and so as to avoid rigidity when manipulated, absorption and recoil characteristics and their movement resembles that of an elastic band. Several values were defined to control the distance between the object drag and the point of contact, as well as the amount of collision reduction. When an object is grasped with a quick motion, it possesses an initial acceleration factor and when it is winnowed an inverse force is generated to reduce progressively its speed. In order for the objects to stay inside the virtual space, they have collision features that detect the screen's boundaries, and recoil properties so they can bounce off.

E. Information Switching

Only deity-objects image can switch to another image, touching one makes a set of 6-7 colored buttons to appear around it, associated information is obtained when selecting a button(Fig. 5). One button selects language (English – Spanish), another hides all other options and displays the image of the deity again while reverting to the independent movement properties. A hidden timer returns the object to independent movement after three minutes of inactivity.

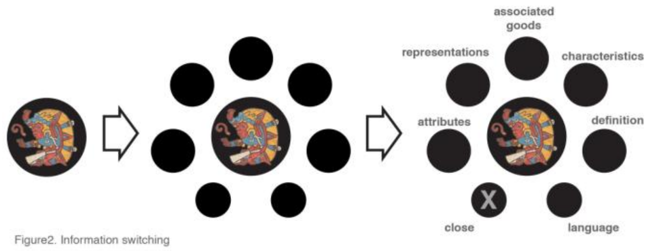

Fig. 5 Information switching

F. Independent Movement Changes

Selecting a Deity-object interrupts its movements and makes it obedient to the motions and directions given by the user. When selected, Sun-objects and Star-objects lose their independent movement and remain static until released. Comet-objects cannot be interacted with.

V. CONCLUSIONS

The board installed at "the Museum of Tlatelolco" accomplished its exhibition goal; users can watch the icons freely moving and this attracts their attention, and since the multi-touch technologies are becoming increasingly popular, when users approach, they immediately begin to interact intuitively with the application.

However, this simplicity conceals a great deal of care and complexity in the development of this application, as well as attention to details in the object behavior and responses to user's input interface.

The object's behavior organization and programming need a lot of care, because it is necessary to define the events that classify individual behaviors assigned to each object: object and option selection, independent movements, solid object dynamics, borders, obstacles, options display, information switches, language selection, and timers. It is important to establish priority events and tag sets, where every event and change can be stored in.

The integration of a multi-touch-screen goes beyond being able to manage multiple simultaneous events; it resides in the interaction implementing possibilities between objects and users' gestures. Not taking advantage of this feature would mean losing one of the most important aspects of this interface. The application design and development costs compensate the greater impact of its use.

REFERENCES

Orozco, G., Interactive museums as educational mediators, Electronic Journal Kinetic, vol. 26, Feb-June 2005, pp. 38-50.

Paris, S., Perspectives on Object–Centered Learning in Museums, Lawrence Elbaum Associates, Mahwah, N.J., 2002, pp.408.

Caulton, T., Hands-on Exhibitions: Managing Interactive Museums and Science Centres, Routledge, Nueva York,1998, pp.168.

LMG, Creatures of the Fiery Pool, Peabody Essex Museum, http://rlmg.com/project/creatures-of-the-fiery-pool-4, consulted on March 10th, 2013.

Buxton, W. & Myers, B., A study in two-handed input. In Proceeding of CHI ̕86 Conference on Human Factors in Computing System Boston Massachusetts. 1986, pp. 321–326.

Baecker, R. & Buxton, W., Readings in human-computer interaction: A multidisciplinary approach, Morgan Kaufman Publishers, Inc., Los Altos, CA, 1987, pp. 950.

Miette, T., Les technologies tactiles - Historique - L'origine des temps, http://www-igm.univ-mlv.fr/~dr/XPOSE2008/Les technologies tactiles/histo_origine.html, Xposés Systèmes et Réseaux, consulted on March 10th, 2013.

Kaltenbrunner, M., Bovermann, T., Bencina, R. & Costanza, E., TUIO A Protocol for Table-Top Tangible User Interfaces,http://opensoundcontrol.org/files/tuio_gw2005.pdf , 2005.

Reynolds, C. W., Flocks, herds and schools: a distributed behavioral model, Proceedings of the 14th annual conference on Computer graphics and interactive techniques, ACM, 1987, pp. 25–34, New York, USA.

Reynolds, C. W., Steering Behaviors For Autonomous Characters, Game Developers Conference, 1999, pp. 763–782, San Jose, California, USA.