



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE CIENCIAS

**SUPERFICIES INTERACTIVAS COMO APOYO AL TRABAJO
COLABORATIVO EN EL AULA DE PRIMARIA**

**REPORTE DE
SERVICIO SOCIAL**

**QUE PARA OBTENER EL TÍTULO DE:
LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN**

P R E S E N T A:

OTTO ISRAEL SÁNCHEZ PADILLA



**TUTOR:
DR. FERNANDO GAMBOA RODRÍGUEZ**

2011



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

1. Datos del alumno.

Apellido paterno: Sánchez
Apellido materno: Padilla
Nombre(s): Otto Israel
Teléfono: 53701627
Universidad: Universidad Nacional Autónoma de México
Facultad: Facultad de Ciencias
Carrera: Ciencias de la Computación
Número de cuenta: 302767434

2. Datos del tutor.

Grado: Dr.
Nombre(s): Fernando
Apellido paterno: Gamboa
Apellido materno: Rodríguez

3. Datos del sinodal 1.

Grado: Dr.
Nombre(s): José
Apellido paterno: Galaviz
Apellido materno: Casas

4. Datos del sinodal 2.

Grado: M. en C.
Nombre(s): Gustavo
Apellido paterno: De la Cruz
Apellido materno: Martínez

5. Datos del sinodal 3.

Grado: M. en C.
Nombre(s): María Guadalupe Elena
Apellido paterno: Ibargüengoitia
Apellido materno: González

6. Datos del sinodal 4.

Grado: L. en C. C.
Nombre(s): Francisco Lorenzo
Apellido paterno: Solsona
Apellido materno: Cruz

7. Datos del trabajo escrito.

Título: Superficies Interactivas como Apoyo al Trabajo Colaborativo en el Aula de Primaria
Subtítulo:
Número de páginas: 68 p.
Año: 2011

Superficies Interactivas como Apoyo al Trabajo Colaborativo en el Aula de Primaria

Otto Israel Sánchez Padilla

Agradecimientos

Quisiera agradecerle a mi familia, mis amigos, mis sinodales y a todas las personas que hicieron posible este proyecto.

Índice general

I Fundamentos	7
1. Introducción	9
2. Desarrollo de la tecnología multitouch	11
3. Funcionamiento y Componentes de las Superficies Multitouch	15
3.1. Iluminación infrarroja	17
3.2. Cámara infrarroja	17
3.3. Métodos de construcción de superficies multitouch .	19
3.3.1. Frustración de la Reflexión Interna Total	19
3.3.2. Iluminación Difusa de la Superficie	21
3.3.3. Iluminación Difusa	22
3.3.4. LLP Laser Light Plane (Plano de Luz de Láser) .	24
3.3.5. LED LP (Plano de Luz de LEDs)	25
3.4. Herramientas para el seguimiento de dedos y fiduciar- rios (TUIO Trackers)	27
3.4.1. reactIVision	27
3.4.2. Community Core Vision	27
3.4.3. Touché	28
3.4.4. Movid	28
3.4.5. Surface Tracker	28
3.5. Protocolo TUIO	29
3.5.1. Estructura del cliente TUIO	29
3.6. Frameworks TUIO para desarrollo de aplicaciones mul- titouch	30

II	Desarrollo del proyecto	33
4.	Superficies aumentadas como apoyo al trabajo colaborativo en el aula: Una nueva propuesta de interacción	35
4.1.	Objetivos y Metas	35
5.	Diseño y Desarrollo de la Superficie interactiva	39
5.1.	Diseño de la superficie	39
5.2.	Hardware	41
5.3.	Software Utilizado	44
5.3.1.	reactIVision	44
5.3.2.	Processing (Java)	44
5.4.	Software desarrollado para el proyecto	45
5.4.1.	Framework	45
5.4.2.	Diseño de la aplicación colaborativa para niños "Comparación Grupal de Fracciones Equivalentes"	51
5.4.3.	Otras actividades	57
5.5.	Pruebas con alumnos en la escuela primaria "Galileo Galilei"	57
III	Conclusiones	61
6.	Conclusiones y Trabajos Futuros	63
6.1.	Conclusiones	63
6.2.	Trabajos Futuros	64

Parte I

Fundamentos

Capítulo 1

Introducción

Multitouch se refiere al conjunto de técnicas de interacción que permite a los usuarios controlar aplicaciones gráficas con varios dedos. Los dispositivos Multitouch constan de una superficie táctil (pantalla, mesa, pared, etc.) así como del software que reconoce múltiples y simultáneos puntos de toque. A diferencia de las pantallas táctiles (touchscreen), las interfaces multitouch permiten el uso de varios dedos al mismo tiempo dentro de una aplicación.

En los últimos años, la utilización de técnicas y dispositivos basados en interfaces Multitouch se ha incrementado (teléfonos móviles, computadoras, cámaras fotográficas, etc). Sin embargo su uso con fines didácticos y educativos no se ha explotado de manera suficiente, quedando como una importante veta y campo fértil para desarrollar aplicaciones que apoyen los procesos de enseñanza-aprendizaje. En efecto, las interfaces Multitouch (dispositivos que utilizan alguna técnica multitouch para su manipulación) permiten generar sistemas en la que varios usuarios (alumnos) participen de manera simultánea y colaborativa en la solución o análisis de un problema, lo que hace la propuesta sumamente interesante en el ámbito educativo.

Consecuentemente, el objetivo de este documento es señalar algunos aspectos básicos y generales de la historia en el desarrollo de estas técnicas, explicar su funcionamiento general y en particular, presentar la forma en que participé en el proyecto desarrollado dentro del Centro de Ciencias Aplicadas y Desarrollo Tecnológico, en el que se decidió probar esta tecnología, con fines educativos.

Capítulo 2

Desarrollo de la tecnología multitouch

Estos son algunos de los eventos más importantes que dieron lugar al desarrollo de esta tecnología: [7]

- Las interfaces multitouch tienen sus orígenes con el desarrollo de las pantallas táctiles. Dichas pantallas se empezaron a desarrollar a finales de los años sesentas por parte de IBM y durante los años setentas varias compañías comenzaron a fabricar este tipo de pantallas para controlar algunos dispositivos electrónicos, la mayoría de las veces utilizando sensores capacitivos. Una de las primeras pantallas táctiles más famosas fue la PLATO IV Touch Screen Terminal, desarrollada en la Universidad de Illinois con fines educativos.
- No fue sino hasta 1982 cuando se creó la primera pantalla multitouch para uso humano llamada Flexible Machine Interface. Fue desarrollada en la Universidad de Toronto por parte del Input Research Group. Esta interfaz consistía en un panel de vidrio y una cámara colocada detrás de éste. Cuando un dedo tocaba el panel, la cámara detectaba dichos puntos (las sombras) interpretándola como un evento de entrada. Con este dispositivo se podían dibujar imágenes con varios dedos a la vez.
- En 1983 fue creado el Video Place/Video Desk, sistema basa-

do en una visión que seguía las manos y los dedos. Dado que sólo seguía las manos, en sí no detectaba cuando se hacía contacto con la superficie, sino que se enfocaba en interpretar la posición de las manos y los gestos de éstas para detonar ciertos eventos, como mover objetos y escalarlos entre otras cosas.

- En 1984 Bob Boie, de Bell Labs, desarrolló una pantalla multitouch que consistía en un conjunto de sensores capacitivos transparentes colocados sobre un CRT (tubo de rayos catódicos). Con este dispositivo se podían manipular objetos gráficos con una buena respuesta.
- Ya en 1985, el Input Research Group de la Universidad de Toronto desarrolló una Tablet Multitouch, capaz de detectar varios dedos a la vez. Basada en sensores capacitivos; era mucho más delgada que los sistemas basados en visión desarrollados previamente.
- En 1991 Pierre Wellner desarrolló el Digital Desk, sistema donde se utilizaba una proyección sobre una superficie y con ayuda de técnicas ópticas y acústicas para detectar la posición de las manos y de los dedos. En este sistema se podían ver ya gestos famosos de las superficies multitouch como el escalamiento de objetos con 2 dedos entre otras cosas.
- A principios de la década de los noventa se desarrollaron varios dispositivos multitouch, como lo fueron el Flip Keyboard de 1992, que era un pad integrado a un teclado con el cual se podían controlar algunas aplicaciones y el Simon, desarrollado por IBM y Bell South, el cual era un teléfono con pantalla táctil.
- Posteriormente fueron desarrollados otros dispositivos como el Active Desk (1995), un escritorio con una proyección trasera, donde se manipulaban los objetos con una pluma y con una mano. También fue creado el T3 (1997), una Tablet que permitía manipular gráficos utilizando las manos.

- En 1998 fue creado el Fingerworks, una Tablet con capacidad multitouch que soportaba una gran biblioteca de gestos multitouch.
- En 2002 se creó el Smart Skin por parte de los laboratorios de Sony; este sistema era una arquitectura para crear superficies interactivas sensibles a las manos y a los gestos de los dedos. Funcionaba con un sensor que era capaz de detectar la posición de las manos, determinar su forma y calcular la distancia entre estas y la superficie.
- En 2003 y 2004 JazzMutant desarrolló el Lemur, un controlador musical con una pantalla con capacidades multitouch, aunque no fue hasta 2005 que presentaron la versión final de su producto.
- También en el 2003, un equipo de investigación de la Universidad Pompeu Fabra en Barcelona empezaron a desarrollar la Reactable, un instrumento musical electrónico donde el usuario puede controlar el sistema utilizando tanto sus dedos así como objetos tangibles. Fue presentada en un concierto público por primera vez en 2005.
- En 2004 y 2005 se crearon varios sistemas multitouch, como el TouchLight, PlayAnywhere y Tactiva entre otros.
- En 2006 es creado el Natural User Interface Group (NUI Group), comunidad con el propósito de investigar y crear "técnicas de sensado de máquina" para beneficiar aplicaciones artísticas, comerciales y educativas. NUI Group ofrece un ambiente colaborativo para desarrolladores interesados en aprender y compartir nuevos métodos y conceptos de interacción humano-computadora. Entre estos se encuentran la realidad aumentada, el reconocimiento de voz, la escritura y gestos, las tecnologías multitouch, la visión computacional y la visualización de la información.
- En el 2007 fueron creados 2 de los productos más famosos con capacidades multitouch: El Apple iPhone, teléfono con

una pantalla capacitiva e interfaz táctil, y la Microsoft Surface, que es una superficie multitouch que permite a varios usuarios manipular contenido digital sin necesidad de un teclado o ratón.

- En los últimos años la popularidad de los dispositivos multitouch se ha incrementado, muchos teléfonos han integrado esta tecnología e incluso los sistemas operativos ya cuentan con capacidad multitouch.

Capítulo 3

Funcionamiento y Componentes de las Superficies Multitouch

Cabe resaltar que para efectos del desarrollo del presente informe, solo se hará énfasis en las superficies basadas en el uso de dispositivos ópticos, ya que fue la plataforma que el grupo de trabajo de CCADET decidió utilizar en el desarrollo del proyecto, aunque se debe mencionar que, además de esta, existen algunas otras técnicas, como son:

- **Superficies Capacitivas** Constan de un panel capacitivo cubierto por una cubierta transparente conductora. Ya que el cuerpo humano es un conductor de electricidad, al tocar la superficie se efectúa un cambio en el campo electrostático que se puede medir como un cambio en la capacitancia. Con esto un controlador puede determinar la posición donde se efectuó el contacto dentro de la superficie.
- **Superficies Resistivas** Estas superficies constan de varias capas, entre las que destacan 2 capas conductoras de electricidad separadas ligeramente. Cuando un objeto hace contacto con la superficie, dichas capas hacen contacto haciendo un cambio en la corriente eléctrica lo cual es interpretado como una entrada por parte del controlador.

- **Sensores de Proximidad** Como su nombre lo indica, en este caso se utilizan sensores de proximidad para detectar la ubicación de los dedos u objetos que hagan contacto con la superficie.
- **Sensores Acústicos** Este método utiliza ondas de ultrasonido que pasan a través del panel. Cuando el panel es tocado, una parte de dichas ondas es absorbido. El cambio en las ondas puede ser interpretado como un toque y ser procesado por el controlador en cuestión.

El principio básico de las superficies que utilizan algún dispositivo óptico consiste en tener una superficie iluminada con luces infrarrojas, de modo que, cuando un dedo u objeto se encuentra sobre la superficie, este se ilumina generando un "blobs" u objetos brillantes luminiscentes (Figura 3.1). Dichos blobs pueden ser captados por una cámara infrarroja, y con ayuda de ciertos programas y protocolos (que se explicarán más adelante), analizan las imágenes y generan información útil para las aplicaciones multitouch.



Figura 3.1: Blobs detectados como la presencia de dedos dentro del Community Core Vision

La construcción de este tipo de superficies requiere de varios aspectos a considerar. Estos se explican a continuación.

3.1. Iluminación infrarroja

La luz infrarroja es un tipo de radiación electromagnética con una longitud de onda mayor a la de la luz visible, pero menor que las microondas. A su vez, la luz infrarroja se puede categorizar en Infrarrojo Cercano (NIR por sus siglas en inglés), Infrarrojo Medio (MIR) e Infrarrojo Lejano (FIR)¹.

Gran parte de las cámaras (sus sensores) son sensibles al Infrarrojo Cercano (de 700nm a 1000nm), por lo cual se les incorpora un filtro que bloquea dicha parte del espectro de luz. Si quitamos dicho filtro e incorporamos uno que elimine la luz visible, se obtiene una cámara infrarroja.

El principal motivo por el cual se usa iluminación infrarroja es para distinguir entre las imágenes que se proyectan sobre la superficie y la imagen que debe tomar la cámara para detectar la ubicación de los dedos y fiduciaros.

Para la construcción de estas superficies multitouch, es necesario incorporar una fuente de iluminación infrarroja. Los LEDs infrarrojos son la fuente de iluminación más común, aunque también se pueden utilizar algún otro tipo de emisor infrarrojo como láseres, dependiendo del método que se decida utilizar (más adelante se explicarán brevemente algunos de ellos).

Es importante verificar que los elementos que se decida utilizar como fuente de iluminación estén dentro de los rangos del espectro del Infrarrojo Cercano, con el fin que la cámara utilizada sea capaz de percibir los elementos sensados por la fuente de iluminación.

3.2. Cámara infrarroja

La cámara nos va a permitir detectar la ubicación de los dedos y de los fiduciaros sobre la superficie. Hay varios aspectos que deben ser considerados al elegir la cámara:

¹Existen varias categorizaciones de la radiación infrarroja, como lo son la del ISO 20473, la de la División de Astronomía entre muchas otras por lo cual es complicado elegir una categorización en particular. Para este caso, tomaremos la clasificación más general que se incluye en varios documentos

- **Framerate.** Esto nos indica la velocidad de la cámara, es decir la frecuencia con la que la cámara captura las imágenes y se mide en fps (frames per second) o cuadros por segundo. La mayor parte de las cámaras tienen una velocidad de 15 a 30 fps, aunque algunas llegan a 60 fps, dependiendo de la empresa que las fabrique. Entre más alto framerate tenga la cámara, mejores resultados se tendrán en la detección de objetos.
- **Resolución.** Este es otro aspecto que se debe tomar en cuenta, ya que entre mayor resolución tenga la cámara, se tendrá mayor precisión sobre la superficie. Muchas de las cámaras no especializadas ofrecen una resolución de 320 x 240 píxeles, lo que es suficiente para superficies pequeñas, pero si se desea fabricar una superficie de mayor tamaño, es recomendable utilizar una cámara con mayor resolución (640 x 480 píxeles o mayor).
- **Interfaz.** Básicamente existen 2 tipos de interfaces para cámaras: USB y Firewire. Generalmente se recomienda usar la interfaz Firewire ya que tiene menor latencia lo que contribuye a tener un sistema con mejor respuesta.
- **Lente.** En este punto hay que tener en cuenta 2 aspectos importantes: Si el filtro infrarrojo de la cámara está integrado a la lente, es necesario reemplazarla por una lente sin dicho filtro y además se debe verificar que el campo de visión de la cámara sea suficiente para abarcar toda la superficie, en caso contrario se puede adquirir un gran angular (si la montura de la cámara lo permite) para obtener el campo de visión necesario.
- **Sensor.** Al momento de elegir la cámara, es necesario verificar si la cámara es capaz de sensar la luz infrarroja. Gran parte de los fabricantes incluyen dicha información del sensor, lo que nos puede indicar la efectividad que tendrá la cámara.

Casi cualquier cámara puede servir para este propósito, pero algunas es necesario modificarlas. Casi todas las cámaras web cuen-

tan con un filtro infrarrojo (excepto cámaras de seguridad o de propósitos específicos) para bloquear esta luz y dejar pasar sólo la luz visible. Lo que se busca es lo contrario, de modo que utilicemos la luz infrarroja como medio de iluminación de los dedos y objetos que se coloquen sobre la superficie y que la cámara detecte únicamente (de preferencia) los mismos.

Muchas cámaras pueden captar la luz infrarroja sin modificación alguna, pero para tener un mejor desempeño, se les hacen estas adecuaciones.

3.3. Métodos de construcción de superficies multitouch

Existen distintas técnicas que permiten construir una superficie Multitouch. A continuación se habla un poco de los métodos más conocidos y desarrollados por miembros del NUI Group.

3.3.1. Frustración de la Reflexión Interna Total

La Frustración de la Reflexión Interna Total (FTIR por sus siglas en inglés) es un método que utiliza como principio el fenómeno del mismo nombre. Este fenómeno se produce cuando un rayo de luz, atravesando un medio de índice de refracción más grande que el índice de refracción en el que este se encuentra, se refracta de tal modo que no es capaz de atravesar la superficie entre ambos medios reflejándose completamente (Figura 3.2).

Con esto se busca es iluminar el interior de la superficie (frecuentemente se utiliza acrílico) con luz infrarroja, quedando dichos rayos atrapados en el interior del acrílico como lo señala este principio, de modo que cuando existe contacto con un dedo sobre la superficie, la reflexión interna se interrumpe dispersando los rayos en ese punto y permitiendo que la cámara infrarroja detecte un "blob".

Para esto, se requieren de los siguientes materiales:

- **Acrílico.** Se debe utilizar acrílico como base de la superficie. Según información del NUI Group, se recomienda que tenga

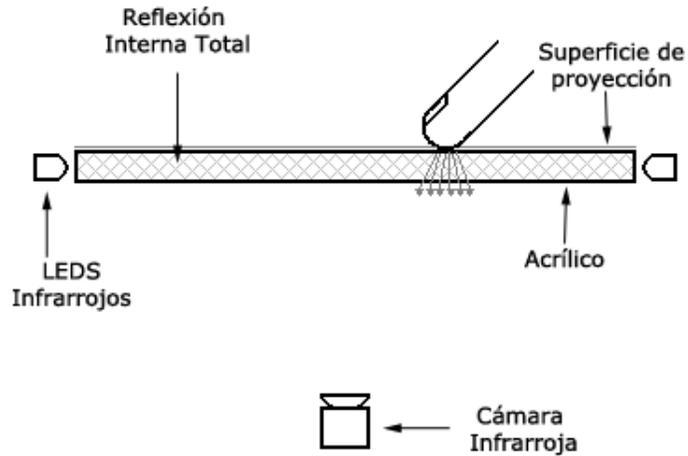


Figura 3.2: Frustración de la Reflexión Interna Total

un grosor grande (1 cm) para evitar que se doble por el peso y además se debe pulir todo el costado del acrílico para mejorar la iluminación.

- **Baffle.** Se necesita un baffle o una cobertura sobre los LEDs para evitar que su iluminación indirecta influya sobre la cámara.
- **Difusor.** Se debe colocar un difusor sobre la superficie para que la cámara solo capte los dedos y no otros objetos que pudieran encontrarse sobre la superficie.
- **Capa adaptable (Compliant layer).** Dado que para lograr la reflexión interna total se debe hacer un buen contacto con la superficie, se debe colocar una capa que mejore el contacto de los dedos con la superficie. Dicha capa debe tener un índice de refracción mayor al del acrílico, pero a su vez debe empalmarse con la superficie al momento de hacer contacto con ella. El recurso más común es recubrir la superficie con una delgada capa de silicón o un material similar.

De acuerdo con el NUI Group este método resulta bastante efectivo, pero tiene la desventaja de que la colocación de los LEDs puede resultar algo complicada, además requiere del uso de acrílico con la capa de silicón para un funcionamiento adecuado (no se puede usar vidrio que es más barato) y dado que la iluminación se realiza en el interior del acrílico, con este método no se pueden utilizar objetos con marcas fiduciaras.

3.3.2. Iluminación Difusa de la Superficie

Este método es muy similar al anterior, salvo que en este caso se debe utilizar un acrílico especial, con lo que se elimina el uso de la capa adaptable de silicón. Dicho acrílico tiene en su interior pequeñas partículas que actúan como pequeños espejos, por lo que cuando el acrílico es iluminado por los costados, la luz infrarroja se extiende a través de toda la superficie. Al acrílico utilizado en este método se le llama "plexiglass endlighen" (Figura 3.3).

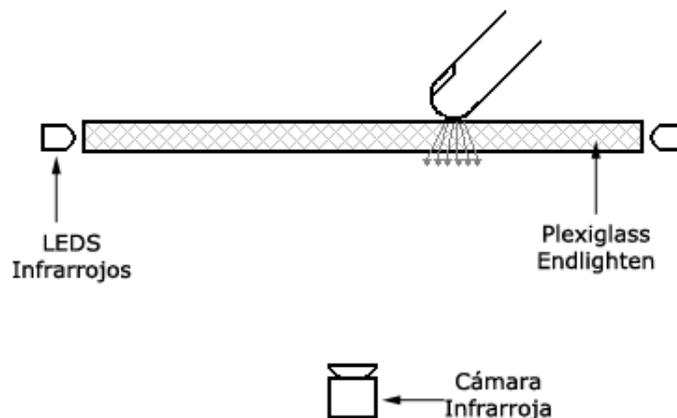


Figura 3.3: Iluminación Difusa de la Superficie

Utilizando este método, se tiene la ventaja de que ya no se requiere la capa de silicón (capa adaptable) utilizada en la Frustra-

ción de la Reflexión Interna Total, y dada la naturaleza del acrílico utilizado, si se pueden utilizar tanto objetos con marcas fiduciaras como dedos. Sin embargo el acrílico que se requiere resulta más caro y difícil de conseguir, y las dimensiones del mismo pueden estar sujetas a las disposiciones del fabricante.

3.3.3. Iluminación Difusa

Este método tiene 2 modos principales: la Iluminación Frontal Difusa y la Iluminación Trasera Difusa, ambos modos buscan lo mismo, que es utilizar el contraste generado por los dedos que toquen la superficie.

■ Iluminación Frontal Difusa

En este caso se utiliza la luz visible (ambiental) y una capa difusora sobre la superficie. Cuando un objeto o un dedo toca la superficie, se genera una sombra que la cámara es capaz de captar (Figura 3.4).

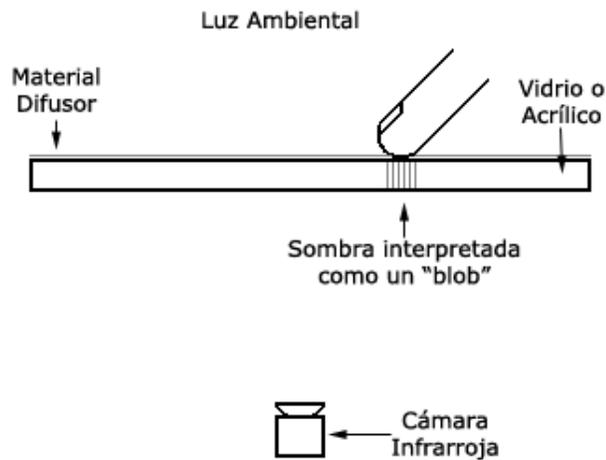


Figura 3.4: Iluminación Frontal Difusa

■ Iluminación Trasera Difusa

En este caso es necesario iluminar la superficie desde la parte trasera de la misma. Aquí también es necesario colocar una capa difusora sobre la superficie, de modo que cuando un objeto toca la superficie, éste refleja más luz de la que refleja la capa difusora o los objetos en el fondo. Esta capa difusora generalmente es utilizada también como superficie de proyección (Figura 3.5).

El principal problema que se presenta es lograr la iluminación uniforme de la superficie, lo cual se puede lograr cambiando la posición de las luces utilizadas.

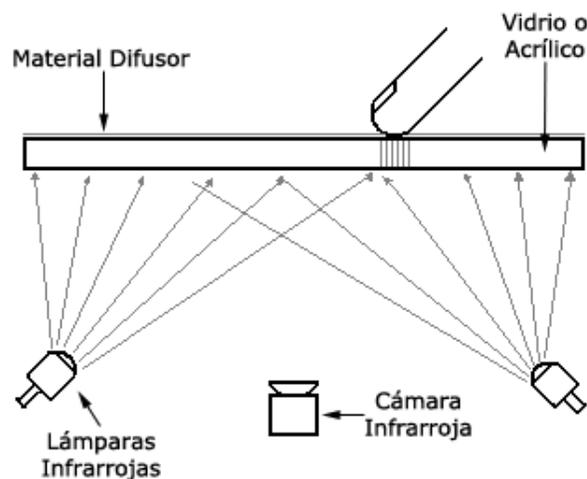


Figura 3.5: Iluminación Trasera Difusa

Tanto la Iluminación Frontal Difusa como la Iluminación Trasera Difusa tienen la ventaja de que no es necesario utilizar forzosamente acrílico (se puede usar vidrio o algún otro material), pero los resultados dependen mucho de la iluminación ambiental y de algunas otras fuentes de iluminación, por lo que es muy complicado lograr la iluminación uniforme de la superficie, lo que puede

dar lugar a la detección de falsos "blobs"(inexistentes) o no lograr el contraste necesario para que el software correspondiente detecte los dedos.

3.3.4. LLP Laser Light Plane (Plano de Luz de Láser)

Al igual que en los métodos anteriores, lo que se busca es iluminar la superficie con luz infrarroja, sólo que en este caso se utilizan láseres infrarrojos como fuente de iluminación (Figura 3.6). Cada láser que se utilice debe contar con un generador de línea, los cuales constan generalmente de una serie de cilindros de cristal o de plástico que se colocan justo enfrente del láser con el fin de transformar el haz de luz del láser en un plano de luz. Los generadores de línea más comunes son de 60° , 89° y 120° . El tipo de generador de línea utilizado depende mucho de las dimensiones de la superficie, de su forma y de cuántos láseres se decida integrar a la superficie. Dado que las superficies más comunes son rectangulares, la configuración más común consiste en utilizar 4 láseres (uno en cada esquina) con un generador de línea de 89° en cada uno de ellos. Los láseres utilizados más comúnmente son de 780nm y 940nm, aunque también es común encontrar láseres de 850nm. También dependiendo de la longitud de onda de los láseres elegidos se puede colocar un filtro con la misma longitud de onda en la cámara para lograr mejores resultados.

Para lograr un funcionamiento adecuado se requiere que el plano de luz generado esté justo por encima de la superficie (se recomienda que esté aproximadamente 1mm sobre la superficie), de modo que cuando un dedo u otro objeto esté en contacto con la superficie, sea iluminado por medio de los láseres.

La utilización de este método lleva consigo cierto riesgo, dado que una manipulación inadecuada de los láseres pueden dañar la vista sin darnos cuenta de ello ya que el ojo humano es incapaz de percibir la luz infrarroja (los reflejos indirectos de los láseres infrarrojos no causan daños). Por eso es altamente recomendable utilizar lentes de seguridad especiales al momento de instalar los láseres.

Otra medida de seguridad que ocasionalmente se puede utilizar

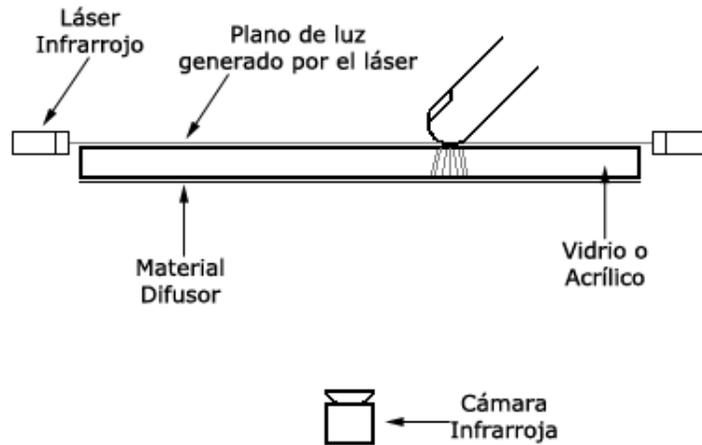


Figura 3.6: Laser Light Plane

(dependiendo de la cámara que se utilice) consiste en sincronizar el pulso del láser con la cámara, de modo de que el láser se prenda únicamente cuando la cámara esté tomando un cuadro.

Este método generalmente da buenos resultados, la instalación es sencilla, pero no se pueden utilizar fiduciales y en algunos casos se puede presentar oclusión de algunos dedos, dependiendo de la disposición y la cantidad de láseres utilizados.

3.3.5. LED LP (Plano de Luz de LEDs)

Aquí la configuración utilizada es muy parecida a la usada en la Frustración de la Reflexión Interna Total. Se cuenta con un arreglo de LEDs, pero en este caso, la luz infrarroja no viaja a través de la superficie sino justo encima de ella.

Al igual que en el método anterior, lo que se busca es generar un plano de luz pero en este caso se utilizan un arreglo de LEDs infrarrojos para lograrlo (Figura 3.7).

Dado que los LEDs en sí no generan un plano de luz, sino su área de iluminación es más amplia, se tiene el problema de que

se van a iluminar objetos que estén encima de la superficie sin tocarla. Estos problemas se pueden solucionar utilizando filtros en los programas encargados del seguimiento de los dedos y de los objetos, así como moviendo los valores de los umbrales de los mismos. Esta solución se puede considerar poco elegante, pero si se efectúa correctamente los resultados pueden ser buenos.

También es recomendado utilizar un bisel o cobertura de los LEDs para evitar en lo posible la iluminación de objetos por encima de la superficie y tratar de iluminar solo los dedos que estén en contacto con ella.

Al igual que en el LLP, aquí no se pueden detectar objetos con marcas fiduciaras

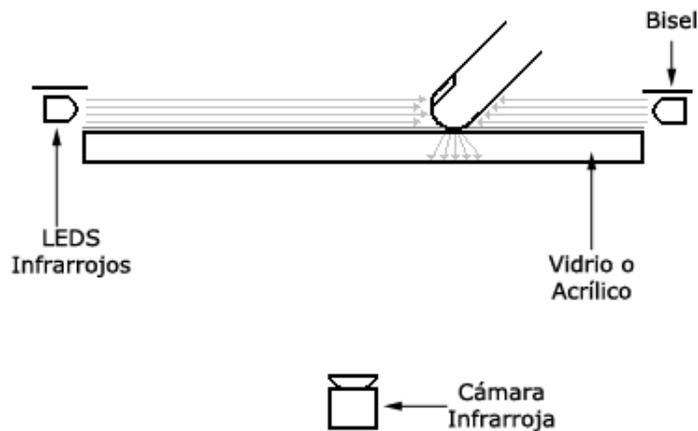


Figura 3.7: LED Light Plane

Como podemos ver, cada método tiene sus ventajas y desventajas, y dependiendo de los recursos y presupuestos que se tenga, se puede elegir un método con el cual construir nuestra superficie.

3.4. Herramientas para el seguimiento de dedos y fiduciaros (TUIO Trackers)

Hasta ahora solo se ha explicado un poco sobre los métodos más comunes para la construcción de las superficies, pero eso no basta. Se necesitan además un conjunto de aplicaciones y herramientas que interpreten y envíen toda la información generada sobre la superficie. Para esto existen varias herramientas encargadas del seguimiento o "trackeo" tanto de los dedos como de los fiduciaros. A continuación se mencionan algunas de estas herramientas.

3.4.1. reactIVision

reactIVision es un framework de visión computacional multi-plataforma y de código libre para el seguimiento rápido y robusto de marcas fiduciaras sobre objetos físicos, así como para el seguimiento de dedos. Fue diseñado principalmente como una herramienta para desarrollar rápidamente superficies multitouch [12].

Actualmente se encuentra la versión 1.4 disponible para Windows, MacOS X y Linux.

3.4.2. Community Core Vision

El CCV o también conocido como tbeta es una herramienta open source y multi-plataforma de visión computacional y de detección en máquinas. Ésta toma como entrada el video y tiene como salida datos del "seguimiento de los dedos", así como eventos que son utilizados dentro de las aplicaciones Multitouch. Puede interactuar con varias cámaras de video y conectarse a varias aplicaciones basadas en el protocolo TUIO, OSC y XML (Estos se explicarán más adelante). Soporta varias técnicas de iluminación como LLP, Frustración de la RIT, entre otros. Fue desarrollado y es mantenido por miembros del NUI Group [1].

3.4.3. Touché

Touché es una herramienta open source de seguimiento para superficies multitouch. Fue desarrollada para MacOS X Leopard utilizando algunas de sus tecnologías como son: Quicktime, Core Animation, Core Image y Accelerate Framework, así como bibliotecas open source como son libdc1394 y OpenCV. Consta de 2 partes principales: Una aplicación independiente encargada del seguimiento de los dedos y un framework que puede ser intergrado en cualquier aplicación que reciba la información del seguimiento de dedos [14].

3.4.4. Movid

Modular Open Vision Interaction Daemon (Movid) es otra herramienta de seguimiento multiplataforma que tiene como principales características poder ejecutarse como un servicio, se pueden habilitar hilos de ejecución así como poseer una API remota para ajustar todos los parámetros de la aplicación en tiempo real [5].

3.4.5. Surface Tracker

Surface Tracker es una aplicación de poca latencia (low latency) para el seguimiento de dedos en superficies Multitouch. Esta aplicación únicamente funciona únicamente con cámaras Optitrack, las cuales tienen buena velocidad (alrededor de 100 fps) utilizando solo de 3 a 5% del procesador [13].

Cabe resaltar que solo se mencionaron algunas de estas herramientas, pero existen algunas otras con características similares que están en desarrollo y/o que fueron creadas para alguna tecnología en particular (como Ortholumen, que fue diseñado para el seguimiento de plumas de luz).

También es importante señalar que existen algunos simuladores de superficies interactivas para poder desarrollar aplicaciones Multitouch sin necesidad de construir una superficie. Entre esos simuladores se encuentran el TUIO Simulator, SimTouch, QMTsim,

SimpleSimulator y el MultiPoint TUIO, que permite simular una superficie con múltiples ratones.

3.5. Protocolo TUIO

Las herramientas mencionadas en la sección anterior, tienen la función de leer las imágenes captadas por la cámara de video, analizarlas y a partir de ellas obtener información que será utilizada dentro de las aplicaciones Multitouch, como lo son posición de los dedos, tamaño de los "blobs" detectados, etc. Estas herramientas transmiten dicha información mediante un protocolo llamado TUIO.

TUIO es un framework abierto que define un protocolo común y un API para superficies Multitouch. El protocolo TUIO permite la transmisión de una descripción abstracta de las superficies interactivas, incluyendo los eventos multitouch y el estado de los objetos tangibles sobre la superficie. Este protocolo codifica la información generada por una aplicación de seguimiento de dedos y envía dicha información a cualquier aplicación capaz de decodificarla [11].

TUIO está basado en Open Sound Control (OSC), protocolo diseñado para la comunicación entre computadoras, sintetizadores de sonido y otros dispositivos multimedia, por lo cual puede ser fácilmente implementado en cualquier plataforma que soporte OSC.

3.5.1. Estructura del cliente TUIO

Actualmente existen clientes del protocolo TUIO para los lenguajes de programación más comunes. En dichos clientes se implementan algunas clases que se explican a continuación:

- **TuioPoint.** Esta clase sirve para manejar posiciones TUIO. Sirve como base para las clases TuioCursor y TuioObject.
- **TuioCursor.** Esta clase realiza el manejo de los cursores (dedos) que se detecten, tiene algunos métodos para obtener información como posición de los dedos, velocidad, la trayectoria que ha seguido entre otras prestaciones.

- **TuioObject.** Esta clase realiza el manejo de los objetos (marcas fiduciarias) que se detecten. Tiene algunos métodos para obtener la posición del objeto, velocidad, el identificador (ID) del objeto, velocidad y trayectoria entre otras cosas.
- **TuioTime.** Esta clase efectúa el manejo simple del tiempo transcurrido desde el inicio de la sesión en la aplicación.
- **TuioClient.** Esta es la clase principal del decodificador del protocolo TUIO, para que una aplicación sea capaz de recibir y decodificar los mensajes TUIO, es necesario crear una instancia del TuioClient. Tiene métodos asociados para conectarse al puerto asociado, desconectarse del mismo u obtener las listas de cursores (dedos) y objetos que se encuentran en cierto momento dentro de la aplicación.
- **TuioListener.** Este componente tiene la función de notificarle al TuioClient los eventos TUIO relevantes dentro de la aplicación, como lo son la aparición de un nuevo cursor, la aparición de un nuevo objeto (fiduciario), el movimiento de algún cursor o de un objeto y la desaparición de los mismos. Al desarrollar una aplicación Multitouch, es necesario implementar dichos métodos dentro de nuestro TuioListener para indicar qué acciones se deben realizar cuando se detecten cada uno de estos eventos.

Estas son las clases más comunes que le dan funcionamiento al cliente TUIO, dependiendo del lenguaje de programación y de la implementación, puede ser que existan algunas otras clases dentro de los clientes.

3.6. Frameworks TUIO para desarrollo de aplicaciones multitouch

En la actualidad se han desarrollado diversos frameworks que facilitan la elaboración de aplicaciones para las superficies Multitouch. Dichos frameworks ya incluyen algunas funcionalidades como crear objetos dentro de las aplicaciones (imágenes, figuras,

etc) y asociarles un comportamiento en específico dependiendo del evento TUIO que se detecte dentro del mismo.

Entre estos frameworks se encuentran:

- **MT4j.** (Multitouch for Java) hecho sobre Processing (Java).
- **PyMT.** Desarrollado en Python.
- **Grafiti.** Basado en el API TUIO de C#.
- **TangibleFramework.** Desarrollado sobre openFrameworks (C++).
- **uniTUIO y uniducial.** Soporte Multitouch y de fiduciarios para el motor de juegos Unity3D.
- **Miria.** Para MS Silverlight.

Quizá el más famoso de ellos es el MT4j, el cual cuenta con una documentación bastante completa.

También existen algunos frameworks con licencia comercial como **GestureWorks** (para Flash) y **So touch** (para Adobe AIR/AS3).

Parte II

Desarrollo del proyecto

Capítulo 4

Superficies aumentadas como apoyo al trabajo colaborativo en el aula: Una nueva propuesta de interacción

4.1. Objetivos y Metas

Desde hace décadas, el uso de las tecnologías de la información ha resultado trascendente en prácticamente todos los campos de la actividad humana. Sin embargo, cuando se analiza el efecto que dichas tecnologías han tenido en la educación, sus repercusiones resultan menos claras, ya que han podido modificar poco los esquemas educativos tradicionales. En este contexto, la literatura especializada señala al trabajo colaborativo como una estrategia que además de permitir un mejor aprendizaje en el estudiante, ayuda a desarrollar en ellos la capacidad de llevar a cabo trabajos de manera coordinada y organizada.

¿Cómo contravenir el paradigma de la computadora personal?, ¿Cómo ofrecer sistemas multiusuario con control distribuido, en que todos los usuarios puedan modificar el estado del sistema directamente?, ¿Cómo hacer sistemas con interfaces en dimensión humana, en los que un grupo de usuarios puedan debatir cómodamente el estado actual del sistema y sus posibles soluciones?.

En este proyecto desarrollado dentro del Centro de Ciencias Aplicadas y Desarrollo Tecnológico de la UNAM (CCADET) con fondos del CONACYT se buscó resolver dichas incógnitas. La meta principal del proyecto fue la construcción de una superficie interactiva y el desarrollo de aplicaciones específicas para la misma a fin de ser probadas por niños de 8 a 10 años estudiantes de Educación Primaria.

Dentro del proyecto se plantearon diversos objetivos:

- Desarrollar una tecnología que impacte de manera significativa la forma y el método en que los niños colaboran y aprenden
- Lograr un enfoque de mediano plazo que permita estudiar, analizar y caracterizar los aportes de dicha tecnología en el contexto educativo previo a su introducción masiva en la sociedad.
- Comparar las interacciones que se dan entre los usuarios al realizar colaborativamente tareas similares con y sin el uso de la tecnología.
- Analizar e identificar tareas susceptibles de ser desarrolladas de manera colaborativa por parte de los niños que utilicen la superficie interactiva.
- Analizar el comportamiento de los niños mientras utilizan la superficie interactiva.

Durante el desarrollo del proyecto, se tuvieron reuniones semanales en las que participaron especialistas del CCADET en la materia como el Dr. Fernando Gamboa y la Dra. Erika Martínez, los expertos en pedagogía de la Facultad de Estudios Superiores Iztacala, Luis Zarzosa y Ofelia Contreras, así como Ruth Briones de la Universidad Pedagógica Nacional.

En dichas reuniones se discutían diversos puntos como son: el diseño de la superficie; las posibles aplicaciones que se desarrollarían para la superficie, así como la viabilidad de las propuestas generadas.

Todo el trabajo desarrollado alrededor del proyecto se puede clasificar en 4 grupos: Diseño de la Superficie, Hardware utilizado, Software utilizado y Software desarrollado.

En el próximo capítulo hablaré de mi participación en este proyecto desarrollado dentro del CCADET.

Capítulo 5

Diseño y Desarrollo de la Superficie interactiva

5.1. Diseño de la superficie

Uno de los primeros puntos que se trató en las reuniones fue el diseño de la superficie.

La superficie que se quería construir debía cumplir con algunos puntos:

- **Portabilidad.** Se requería que la superficie fuera lo más portátil posible, ya que las pruebas se iban a realizar fuera de las instalaciones del CCADET, particularmente en una escuela primaria.
- **Usabilidad en el día.** Como se puede ver en varios videos, la mayoría de este tipo de superficies son utilizadas en ambientes con muy poca iluminación. Para este proyecto se requería que la superficie fuera utilizable en cualquier lugar independientemente de la iluminación ambiental.
- **Utilización como mesa.** Para el proyecto se deseaba que los usuarios se pudieran sentar alrededor de ella teniendo la posibilidad de poner las piernas debajo de la mesa. Se quería que más que una superficie sofisticada, fuera lo más parecido a una mesa.

- **Forma circular.** Se decidió que la superficie tuviera forma circular, ya que de este modo todos los usuarios tendrían la misma participación dentro de las actividades, evitando así darle mayor prioridad a los usuarios que se colocaran justo al frente de la proyección.
- **Altura ajustable.** La superficie debía poder cambiar su altura dependiendo de los usuarios que la utilicen.

Todos estos aspectos suponían algunas complicaciones que se tuvieron que resolver conforme el proyecto se desarrolló. La portabilidad fue uno de los puntos que costó más trabajo resolver. Para conseguirlo se requirieron los servicios de un diseñador industrial. Finalmente se optó por un sistema de piezas especiales de aluminio que se pueden ensamblar de distintas maneras, lo cual hizo que la superficie fuera modular y fuera más fácil de transportar que si se hubiera utilizado una estructura de madera como regularmente se ve en este tipo de superficies.

Como se describió en los capítulos anteriores, el desempeño de estos sistemas depende en gran parte de la iluminación que se tenga en el lugar donde se instale, pues entre más luz ambiental se tenga, más fuentes de luz infrarroja ajenas a las de la superficie serán detectadas, lo que puede conllevar a un desempeño más pobre del sistema. Por esto se decidió utilizar el método del Plano de Luz Láser (LLP) descrito en el punto 3.3.4 de este informe pues permite generar puntos suficientemente brillantes para ser discriminados incluso en un ambiente con mucha luz natural.

Dado que se quería que los usuarios se pudieran sentar alrededor de la mesa (Figura 5.1), la ubicación de los componentes no podía estar justo debajo de ella como regularmente se acostumbra. Para solucionar este punto se adaptó un módulo a un lado de la superficie donde se colocaron el CPU y el proyector. Con esto, el único componente importante que se tuvo que dejar debajo de la mesa fue la cámara, la cual se colocó justo en el centro de los postes que unen las patas de la superficie (Figura 5.2).

El diseño original de la superficie contemplaba usar un vidrio satinado para que sirviera al mismo tiempo como superficie y como medio de retroproyección, pero se observó que la proyección perdía



Figura 5.1: Ubicación final de los usuarios alrededor de la mesa

nitidez y que era muy brillante (lastimaba un poco la vista). Fue por esto que se decidió adquirir un papel especial, para utilizarlo como pantalla (papel Herculene) y otro vidrio para que dicho papel quedara entre dos vidrios y no presentara dobleces, arrugas u otras irregularidades en la superficie.

5.2. Hardware

Como se mencionó anteriormente, se decidió probar el método LLP (Laser Light Plane), por lo cual se adquirieron **4 láseres infrarrojos de 780nm**, los cuales fueron colocados en cada una de las patas de la mesa con su respectivo generador de línea (Figura 5.3).

Una de las principales preocupaciones era el uso de los láseres infrarrojos, dado que si no se manejan con cuidado, pueden dañar la vista sin darse cuenta uno de ello. Por eso mismo, ya que quedaron instalados, se solicitó apoyo al Grupo de Óptica del CCADET, para que efectuara mediciones de la radiación de los láseres y así se descartara cualquier posible peligro para los usuarios.

Al momento de elegir la cámara que se utilizaría en el proyecto, se tenían contempladas 2 opciones: la cámara del PlayStation 3 (PS3 Eye), cámara que ya se había utilizado en proyectos dentro del CCADET y que además es ampliamente recomendada por



Se ubica justo al centro de los postes

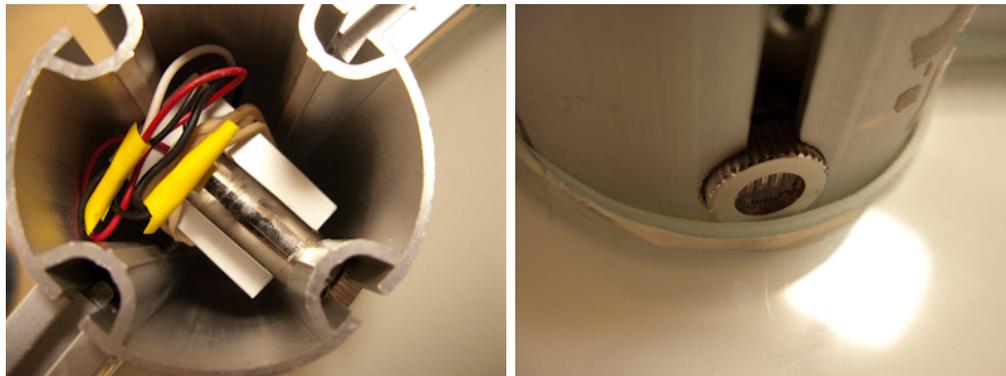
Otra toma

Figura 5.2: Ubicación de la cámara

miembros del NUI Group. Esta cámara tenía la ventaja de que se desempeña a una buena velocidad (60 fps) en una resolución buena (640 * 480), además de que era más económica que nuestra otra opción. La otra cámara que se contempló fue la Optitrack V100, creada por la empresa NaturalPoint que se especializa en crear dispositivos para capturar movimiento. La V100 tenía la ventaja de ser más rápida (100 fps) a la misma resolución, pero siendo considerablemente más cara.

Se probó inicialmente la PS3 Eye de PlayStation, pero se tuvieron muchos problemas para poder integrar el filtro infrarrojo, además de su óptica que no le permitía abarcar toda el área de la mesa. Tampoco fue posible adaptarle un lente gran angular. Por todo ello y con el fin de garantizar una mejor nitidez y resolución de imágenes, lo que redundaría en un desarrollo más exitoso del proyecto se decidió invertir un poco más y adquirir la cámara **Optitrack V100** (Figura 5.4).

Debido a las restricciones de espacio que se tuvieron, la cámara tuvo que quedar a muy poca distancia de la superficie de proyección, evitando así ver (con la cámara) toda la proyección. Para resolver esto, se adquirió un lente gran angular para la cámara, solucionando este problema, pero añadiendo dificultad al momento de calibrarla debido a la distorsión de la lente (sobre todo en el



Montura del laser

Generador de línea

Figura 5.3: Laser utilizado para iluminar la superficie

perímetro de la superficie).

El proyector que se utilizó fué el **NEC WT610**, que cuenta con una resolución nativa de 1024 * 768 píxeles y además es de "tiro corto" lo que le permite, a través de una serie de espejos hacer proyecciones sobre áreas grandes, a pesar de que se encuentre a poca distancia de la superficie de proyección (Figura 5.5).

Al momento de buscar el CPU, se buscaron principalmente 2 cosas: que tuviera dimensiones pequeñas para facilitar su transportación y colocación, y que tuviera un buen rendimiento, principalmente en cuestiones de gráficos. Finalmente se adquirió un CPU **Dell Inspiron 530 S** con las siguientes características:

- Procesador Intel Core 2 Duo a 2.66 GHz.
- 2.00 GB de memoria RAM.
- Disco Duro de 140 GB.
- Tarjeta de Video ATI Radeon X1300 de 128 MB.

Se eligió este CPU principalmente por tener un buen desempeño en cuestiones de gráficos en un CPU con dimensiones pequeñas (comparado con CPUs de las mismas capacidades y de otras marcas).



Figura 5.4: Optitrack V100 con un lente gran angular

5.3. Software Utilizado

5.3.1. reactIVision

Como se mencionó anteriormente, reactIVision es una aplicación encargada de analizar las imágenes obtenidas a través de una cámara, interpretar dicha información y enviarla a la aplicación Multitouch por medio del protocolo TUIO (Figura 5.7).

Se decidió utilizar reactIVision sobre las demás aplicaciones similares (CCV, etc.) principalmente por la calibración de la cámara. A diferencia de la mayoría de los programas encargados del seguimiento de dedos en los que la calibración se realiza colocando dedos en ciertos puntos clave de la superficie (esquinas, o el centro) (Figura 5.8), en reactIVision se debe mapear una cuadrícula proyectada sobre la superficie con la cuadrícula de calibración del reactIVision. Esto resultó de gran ayuda ya que al ser circular la superficie, la calibración utilizando como referencia las esquinas de la proyección hubiera sido más complicada.

5.3.2. Processing (Java)

Processing es un lenguaje de programación y entorno de desarrollo basado en Java. Es de código abierto y está enfocado en la



Figura 5.5: Proyector WT610 de tiro corto

enseñanza y desarrollo de proyectos multimedia. Puede correr en Mac, Windows y Linux. El principal motivo por el cual se eligió este entorno de desarrollo fue a las prestaciones que ofrece en el manejo de gráficos y a la facilidad al momento de depurar el código.

5.4. Software desarrollado para el proyecto

5.4.1. Framework

Ya que dentro del CCADET se tenían varios proyectos que involucraban la utilización de superficies Multitouch como pruebas de esta tecnología dentro de los laboratorios de ciencias del bachillerato, en el aula del futuro o este proyecto en específico, nos dimos a la tarea (un amigo que también realizó su servicio social en el CCADET y yo) de investigar sobre las herramientas que facilitarían el desarrollo de aplicaciones para este tipo de superficies. En aquel momento la mayoría de los frameworks que hoy se encuentran disponibles estaban en vías de desarrollo o tenían muy poca difusión, por lo cual decidimos desarrollar nuestro propio framework para el desarrollo de las aplicaciones requeridas por todos los proyectos.

Lo primero que tuvimos que decidir fue el lenguaje y el conjunto



Figura 5.6: Diseño final de la mesa

de herramientas con las cuales desarrollaríamos dicho framework, con lo cual nos quedamos con 3 opciones: utilizar C# con el conjunto de herramientas de XNA de Microsoft, usar C++ con openFrameworks o bien utilizar JAVA con las bibliotecas de Processing.

Después de analizar y evaluar las tres opciones, decidimos que sería conveniente que las aplicaciones se pudieran ejecutar en distintas plataformas, con lo cual eliminamos la posibilidad de utilizar XNA (eso nos obligaría a desarrollar aplicaciones únicamente para Windows).

Las 2 opciones restantes tienen prácticamente la misma funcionalidad, por lo que finalmente nos decidimos por utilizar JAVA y Processing ya que consideramos que sería más fácil depurar código que utilizando C++. Cabe resaltar que el desempeño no se ha visto afectado en absoluto al utilizar JAVA ya que el manejo de todos los gráficos es realizado por la tarjeta de video mediante el uso de OPENGL.

A continuación doy a conocer algunas de las principales características del Framework desarrollado:

Características del Framework desarrollado

- **Multiplataforma.** Las aplicaciones desarrolladas con el framework pueden ejecutarse tanto en Windows, Mac o Linux



Figura 5.7: Detección de dedos dentro de reactIVision

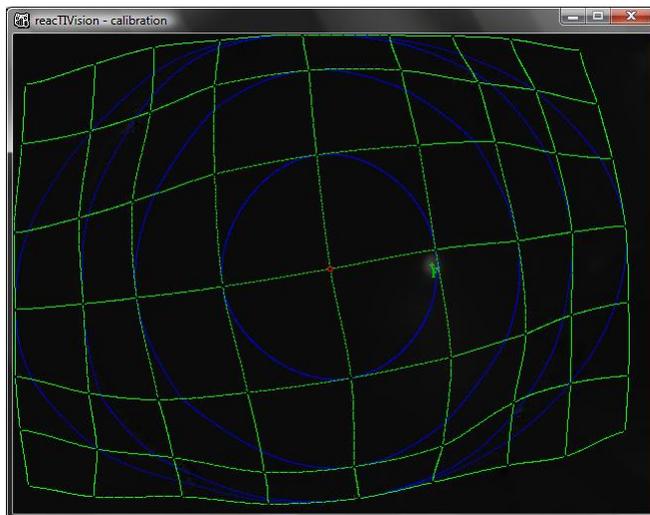


Figura 5.8: Calibración de la cámara

(exceptuando algunas bibliotecas como las de captura de video que tienen que adaptarse).

- **Aceleración gráfica.** Al desarrollar una aplicación se puede elegir entre utilizar JAVA2D u OPENGL, lo cual es una ventaja en caso de contar con una tarjeta de video adecuada.
- **Motor de física.** Se le integró un motor de física (JBox2D) que permite agregar efectos como gravedad o colisión entre objetos dentro de la aplicación (Figura 5.10).
- **Gestos personalizados.** A cada objeto creado dentro de una aplicación se le puede asociar un comportamiento personalizado (mover, rotar, escalar, etc) dependiendo de los movimientos de los dedos asociados al objeto.

También se le decidió incorporar una biblioteca de cálculos matemáticos rápidos, que sacrificando un poco de precisión, reducen el número de operaciones necesarias a la hora de hacer algunos cálculos como distancias, senos, cosenos, etc.

El framework desarrollado se puede dividir en 3 partes: Aplicaciones, módulos y objetos.

Aplicaciones

En la parte de aplicaciones se definieron e implementaron todas las clases que se utilizan dentro de una aplicación en particular. Más adelante se hablará de las aplicaciones desarrolladas para este proyecto.

Módulos

La parte de los módulos contiene las clases básicas que le dan estructura y funcionalidad básica a las aplicaciones. Entre los módulos se encuentran:

- **AssetManager.java** Esta clase se encarga del manejo de los archivos ajenos a la aplicación, por ejemplo, si se desea utilizar una imagen dentro de la aplicación, ésta debe ser agregada al AssetManager.

- **Canvas.java** En esta clase se especifican propiedades básicas de la zona de dibujo de la aplicación como resolución, coordenadas relativas a la zona de dibujo, así como la inicialización de la aplicación dependiendo del modo que se elija (JAVA2D u OPENGL). También se especifican los parámetros que se utilizarán en caso de que se use el motor de física, como fricción, gravedad, etc.
- **Delegates.java** En esta clase se define el comportamiento que va a tener un objeto determinado dado un evento en particular como un click, un drag o algún otro gesto. Cuando se determinan las propiedades de un objeto se puede elegir un comportamiento previamente definido, como que sea rotatable, escalable y movable, o se puede crear un comportamiento personalizado.
- **FastMath.java** Contiene algunas funciones matemáticas como funciones trigonométricas o de cálculo de distancias que efectúan dichos cálculos más rápidamente que utilizando las funciones integradas en JAVA sacrificando un poco de precisión.
- **FiducialDelegates.java** Aquí se definen los métodos que se deben sobrescribir para asignar acciones cuando se detecte algún fiduciario dentro de la aplicación.
- **Gesture.java** Esta clase se encarga del manejo de los gestos (movimientos de dedos) dentro de la aplicación. Aquí se puede tener acceso a propiedades de dichos gestos como son posición, ángulo y distancia entre los dedos que se encuentren asociados a un mismo objeto.
- **ModularMultitouchGesturedTuiListener.java** Esta clase implementa el TuiListener y se encarga de interpretar todos los eventos táctiles que ocurran en la aplicación. Aquí se asocian los dedos que se detecten en el tracker con los objetos dentro de la aplicación y con base en eso se crean los gestos correspondientes que pueden ser utilizados por los objetos.

- **ObjManager.java** Como su nombre lo dice, esta clase realiza el manejo de todos los objetos que se hayan declarado dentro de la aplicación. Contiene funciones para detectar si existe algún objeto en una posición, agregar o quitar objetos a la aplicación y se encarga también de hacer el llamado a los métodos `update()` y `draw()` de cada objeto.
- **ObjList.java** Esta es una clase auxiliar al `ObjManager`. Aquí se definen los métodos para utilizar la lista de objetos de la aplicación
- **Tween.java y TweenCallbacks.java** Estas clases sirven para efectuar animaciones dentro de la aplicación. Para definir una animación se puede definir un Tween, especificándose su duración y qué se va a hacer antes, durante y cuando finalice la misma.
- **AudioManager.java** Esta clase tiene funciones para manipular archivos de audio dentro de la aplicación.
- **VideoManager.java** Esta clase sirve para manipular video dentro de la aplicación, desde captura con una webcam, hasta la reproducción y navegación de un clip de video ya existente.

Objetos

La parte de los objetos se encarga de definir todos los objetos gráficos que se pueden utilizar dentro de las aplicaciones.

Algunas de las clases definidas en esta parte son:

- **Obj.java** Esta es la clase abstracta que define un objeto que se va a dibujar dentro de la aplicación. Contiene atributos como posición, escala, rotación, si está visible o no, así como métodos para mover, rotar o escalar dicho objeto. Cada objeto que se defina a partir de esta clase va a tener 2 métodos importantes: `draw()` y `update()`. Dentro del método `draw()` se especifica qué elementos del objeto se van a dibujar en cada cuadro de la aplicación y dentro del método `update()` se pueden hacer actualizaciones de las variables internas del objeto.

- **Container.java** Esta clase es similar a Obj.java (extiende a esta), pero agrega las funcionalidades de un contenedor, es decir, un contenedor puede tener dentro de sí varios objetos o incluso varios contenedores de modo que al manipular un contenedor dentro de la aplicación (moverlo, escalarlo, rotarlo, etc.), todos los objetos que se encuentren dentro del mismo van a ser afectados. Es una clase muy útil para crear objetos constituidos por varias partes dentro de una aplicación.
- **Rectangle.java, Circle.java, Shape.java** Permiten dibujar rectángulos, círculos y figuras dentro de la aplicación. Cada uno de estos objetos tiene propiedades como altura, anchura, color, canal alpha (transparencia) además de los atributos definidos dentro de Obj.java.
- **Image.java** Esta clase permite agregar imágenes a la aplicación (JPG o PNG).
- **Keyboard.java** Permite incorporar un teclado indicando el objeto que será el receptor de todos los eventos que ocurran en este.
- **Text.java** Esta clase permite desplegar cadenas de texto indicando color del texto, dimensiones e incluso el tipo de fuente que se desea utilizar.
- **MovieClip.java** Permite importar clips de video con métodos básicos para controlarlos como play, pause, stop, etc.
- **DrawingBoard.java** Clase para agregar una zona de dibujo (tipo pizarra) dentro de la aplicación.

5.4.2. **Diseño de la aplicación colaborativa para niños "Comparación Grupal de Fracciones Equivalentes"**

La primera aplicación que se eligió para utilizar la superficie consistió en una actividad para comprender las equivalencias entre



Figura 5.9: Detección de dedos dentro de una aplicación desarrollada con el framework

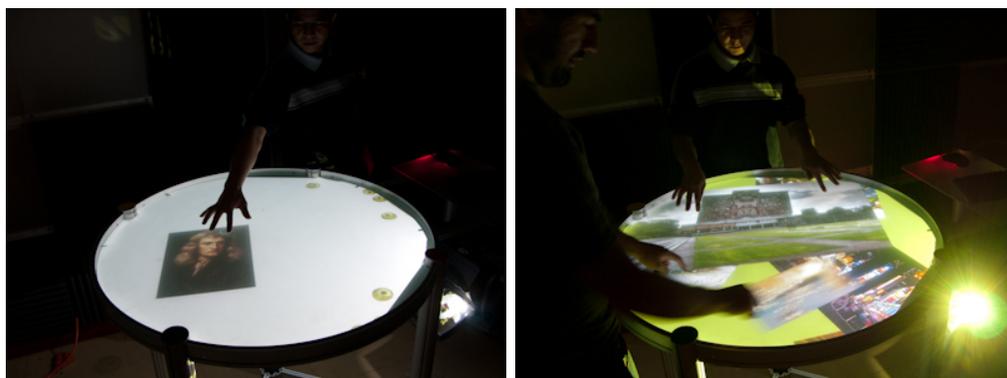
fracciones. Dicha actividad debía estar diseñada para ser utilizada por 4 usuarios simultáneamente. Después de varias reuniones donde se discutían las posibles opciones para su implementación, se llegó a la conclusión de implementar una actividad en la que se manipularan un conjunto de reglas para medir objetos y así determinar equivalencias.

También se decidió que la actividad sería dirigida por una persona, la cual decidiría en qué momento se podría cambiar de ejercicio de acuerdo al desempeño y la participación de los usuarios.

La actividad diseñada se dividió en 3 fases que se explican a continuación:

Fase 1

La primera fase de la actividad consiste en una serie de ejercicios en los cuales los usuarios (los alumnos de educación primaria) deben medir un conjunto de objetos y con base en los resultados obtenidos llenar una tabla de equivalencias. Para ello, en cada ejercicio aparece un objeto sobre la superficie así como un conjunto de 5 ó 6 reglas (dependiendo del ejercicio), con la misma longitud, pero cada una tiene un número distinto de divisiones o escalas (desde 3 hasta 12). También aparece una línea de referencia al lado del



Prueba del motor de física

Manipulación de objetos

Figura 5.10: Aplicaciones de prueba desarrolladas con el framework

objeto a medir, la cual se usa como guía para colocar las reglas y así medir los objetos.

Cuando se coloca una regla que mide exactamente al objeto, esta se "ilumina" indicando que es una regla correcta (Figura 5.12). Con base en todas las mediciones y observaciones realizadas por todos los participantes, se debe ir llenando una tabla de equivalencias. Esta tabla se llena a través de un generador de fracciones (algo similar a una calculadora), y en caso de que se coloque una fracción correcta dentro de la tabla, ésta aparecerá con color negro, pero si la fracción ingresada resulta incorrecta, ésta tendrá un color rojo.

Cuando se determine que se ha terminado con el ejercicio se puede cambiar de ejercicio presionando un botón dentro de la aplicación (Figura 5.12).

Fase 2

La fase 2 consiste en una serie de ejercicios donde se deben medir distancias entre objetos. En cada ejercicio aparece un par de objetos ubicados a una distancia dada (en forma de fracción) y 10 reglas divididas desde 3 hasta 12 partes iguales sobre la superficie.

En esta fase los participantes deben ponerse de acuerdo y defi-

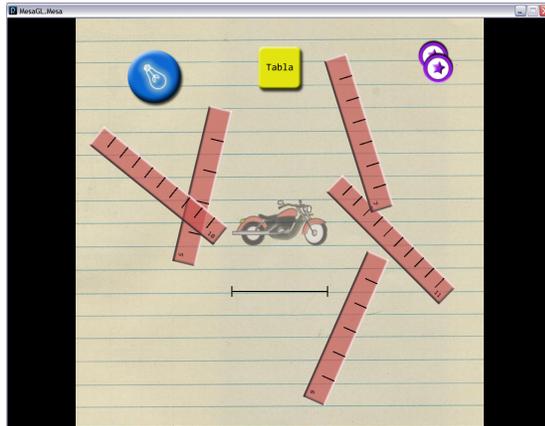
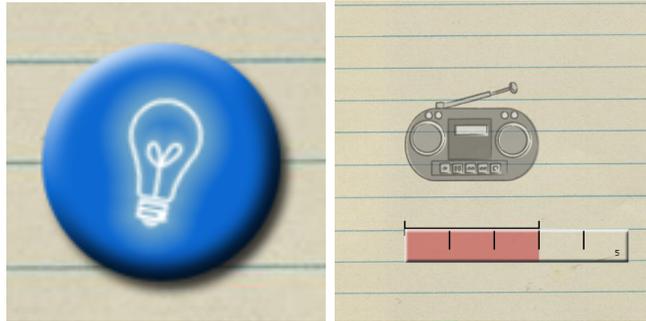


Figura 5.11: Vista general de la aplicación en la fase 1



Botón para cambiar de ejercicio Regla midiendo exactamente el objeto

Figura 5.12: Diversos componentes de la Fase 1



Manipulando las reglas



Generador de fracciones

Figura 5.13: Fase 1 de la actividad de las fracciones

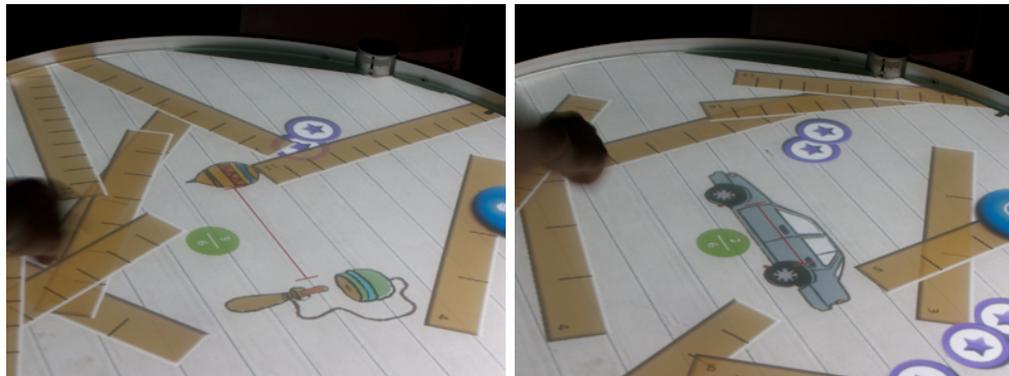
nir qué subconjunto de las reglas dadas podría medir de manera exacta la distancia entre los objetos conociendo de antemano la distancia entre ellos. Al igual que en el ejercicio anterior, cuando una regla que mide exactamente la distancia es colocada en la zona de medición, ésta se ilumina indicando que era una regla correcta.

En el momento en que se considere adecuado, se puede cambiar de ejercicio mediante un botón.

Fase 3

La última fase de la actividad consiste en una serie de ejercicios para comparar fracciones. En cada ejercicio de esta fase a los usuarios se les plantea un problema en el cual deben comparar un par de fracciones y en base a dicha comparación deben responder una pregunta.

Para responder cada pregunta, en la mesa aparecen objetos relacionados con la misma, así como un par de reglas correspondientes con las fracciones a comparar. Cuando los usuarios han realizado las mediciones correspondientes y ya han decidido su respuesta, podrán elegirla utilizando un botón. En caso de que la respuesta sea correcta, aparecerá una paloma de color verde sobre la superficie, en caso contrario aparece una cruz de color rojo. Momentos después de elegir la respuesta y de conocer si era correcta



Midiendo distancias entre objetos

Midiendo la distancia entre las llantas de un coche

Figura 5.14: Fase 2 de la actividad de las fracciones

o no, automáticamente se realiza el cambio de ejercicio, cambiándose todos los elementos sobre la superficie por nuevo elementos correspondientes con el nuevo problema.

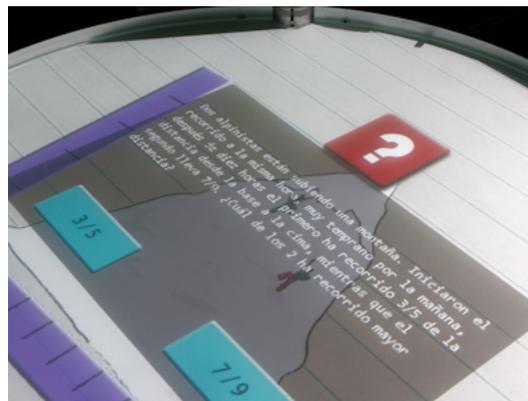


Figura 5.15: Fase 3 de la actividad de las fracciones

5.4.3. Otras actividades

Actualmente se siguen diseñando actividades para ser probadas con la superficie. Entre estas se encuentra una actividad que involucra la comprensión y construcción de textos de manera colaborativa.

5.5. Pruebas con alumnos en la escuela primaria "Galileo Galilei"

Una vez desarrolladas las aplicaciones, se buscó una escuela en la cual se permitiera que un grupo de estudiantes trabajara con la superficie y la sesión fuera filmada. Tras pláticas con el director de la escuela primaria Galileo Galilei, ubicada en Tenosique # 379 Esq. Sacalum, Col. Pedregal de San Nicolás México D.F., se nos permitió realizar las pruebas dentro de sus instalaciones.

Las pruebas se realizaron en 2 etapas, una sin tecnología y otra con el uso de tecnología. Ambas etapas siguieron el mismo protocolo: Se le pidió al director de la escuela que eligiera 4 alumnos al azar (distintos para cada etapa) de tercer grado de primaria, los cuales llevarían a cabo las tareas que la guía de la actividad (la pedagoga Ruth Briones) les pidiera.

En la primer etapa se probó la actividad sin el uso de la superficie, utilizando ilustraciones y reglas de papel para efectuar las tareas a observar. La prueba tuvo una duración aproximada de 2 horas.

Para la segunda etapa, se trasladó la superficie interactiva a la escuela un día antes de la prueba. En esta ocasión, los niños debían efectuar las tareas de medición de objetos y comparación de reglas utilizando los objetos proyectados sobre la mesa. La prueba tuvo una duración aproximada de 2 horas.

Ambas etapas fueron filmadas para el posterior análisis de los videos por parte de los especialistas.

Una vez realizadas las pruebas, los colaboradores del proyecto han realizado algunas observaciones con las cuales se han podido identificar posibles cambios favorables dentro de las aplicaciones y nuevas ideas para el desarrollo de las mismas.

Aunque no fue mi labor analizar los videos de dichas sesiones, en mi opinión los alumnos se mostraron más interesados y más participativos en las sesiones en las cuales se usó la superficie interactiva.



Figura 5.16: Pruebas en un ambiente iluminado



Figura 5.17: Participación de los usuarios en las pruebas



Figura 5.18: Usando las reglas para medir los objetos

Parte III

Conclusiones

Capítulo 6

Conclusiones y Trabajos Futuros

6.1. Conclusiones

Después de haber trabajado en este proyecto, estas son las conclusiones y reflexiones a las que he llegado.

- La construcción de este tipo de superficies se ha facilitado notablemente gracias a una gran cantidad de páginas y foros, en los cuales se explica con gran detalle el funcionamiento y el cómo desarrollar este tipo de superficies, para casi cualquier presupuesto.
- Por su interfaz innovadora, las aplicaciones para estas superficies llaman más la atención a los usuarios en comparación con aplicaciones similares desarrolladas para una computadora común.
- El desarrollo de las aplicaciones realizado durante mi participación en el proyecto me permitió poner en práctica diversos conocimientos adquiridos a lo largo de la carrera como el uso de estructuras de datos, herencia o el manejo de transformaciones (rotaciones, traslaciones de objetos) entre otras cosas, así como darles un destino definido y práctico.

- Mi participación en este proyecto me permitió interactuar con personas de diferentes disciplinas y carreras, lo que me dio una perspectiva distinta del mismo proyecto y no una visión exclusivamente determinada por mi formación.
- De lo observado en las evaluaciones, estas tecnologías parecen ser una herramienta efectiva de apoyo al proceso de enseñanza-aprendizaje de los niños, por lo que debería promoverse su desarrollo y explotación masiva. Del mismo modo, se deben seguir probando nuevas tecnologías (Kinect, etc.) ya que considero que tienen un gran potencial para su uso con fines educativos.

6.2. Trabajos Futuros

Aunque gran parte del trabajo ya se ha realizado, aún quedan algunos detalles en los que se puede seguir trabajando tanto para facilitar el uso de la superficie como para facilitar el desarrollo de aplicaciones para la misma. Estos son algunos puntos que considero se deben enfocar las personas que continúen desarrollando este proyecto:

- Desarrollar un modo de calibración más sencillo que el que se utiliza actualmente. Esto facilitaría entre muchas cosas el traslado de la superficie para efectuar pruebas.
- Por el momento, únicamente las aplicaciones utilizan la capacidad multitouch, el resto del trabajo (iniciar las aplicaciones, apagar el equipo, etc.) se efectúa utilizando el teclado y el ratón. Sería conveniente realizar todas las tareas desde la superficie y así eliminar por completo el uso del teclado y del ratón.
- En lo referente al framework desarrollado, quedan algunas tareas pendientes como la documentación adecuada del mismo para poderlo hacerlo público y así para permitir su enriquecimiento por parte de la comunidad.

- En lo referente a las aplicaciones desarrolladas, considero que sería conveniente integrar audio a las mismas para hacerlas más atractivas.

Bibliografía

- [1] Community core vision. <http://ccv.nuigroup.com/>.
- [2] Infrared (ir). <http://en.wikipedia.org/wiki/Infrared>.
- [3] Lemur controller. http://www.jazzmutant.com/lemur_overview.php.
- [4] Microsoft surface. <http://www.microsoft.com/surface/en/us/default.aspx>.
- [5] Movid. <http://movid.org/>.
- [6] mt4j. <http://www.mt4j.org/>.
- [7] Multi-touch systems that i have known and loved. <http://www.billbuxton.com/multitouchOverview.html>.
- [8] Nui group. <http://nuigroup.com/go/>.
- [9] Optitrack. <http://www.naturalpoint.com/optitrack/>.
- [10] Processing. <http://processing.org>.
- [11] Protocolo tuio. <http://tuio.org/>.
- [12] Reactivision. <http://reactivision.sourceforge.net/>.
- [13] Surface tracker. <http://www.future-instruments.net/fr/download.html>.
- [14] Touché. <http://gkaindl.com/software/touche>.

- [15] **Touchscreen.** [http://en.wikipedia.org/wiki/Touchscreen.](http://en.wikipedia.org/wiki/Touchscreen)
- [16] **AUTHORS, N. G.** *Multitouch Technologies.* 2009.
[http://nuicode.com/attachments/download/115/Multi-Touch_Technologies_v1.01.pdf.](http://nuicode.com/attachments/download/115/Multi-Touch_Technologies_v1.01.pdf)