

**Universidad Católica**  
**“Nuestra Señora**  
**de la**  
**Asunción”**

**Trabajo Práctico**  
**de Teoría y Aplicación de la**  
**Informática**

**Tema: “*CamSpace*”**

**Juan José Jiménez Franco**

**2008**

# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Paradigm Shift</b>	
2.1 CamTrax – La Compañía	4
2.2 ¿Qué es CamSpace?	5
2.3 Características y funcionamiento del Software	5
2.4 Modo de Uso	
2.5 Detalles de la Implementación	10
2.6 Consideraciones para un mejor funcionamiento	10
<b>3. Juegos compatibles actualmente</b>	
3.1 Juegos oficiales del sitio	11
3.2 Juegos configurados por usuarios	11
<b>4. CamSpace Vs. Otras propuestas</b>	<b>12</b>
<b>5. Conclusión</b>	<b>14</b>
<b>6. Bibliografía</b>	<b>15</b>

# 1. Introducción

Corría la década de los 60, cuando Douglas Engelbart y Bill English, en el Instituto de Investigación de la Universidad de Stanford, diseñaron el primer prototipo del periférico patentado bajo el nombre de “*X-Y Position Indicator for a Display System*”, mejor conocido mundialmente como *mouse*. Un tiempo más tarde, en la misma región (Silicon Valley y Palo Alto - California), el *mouse* iba a ser mejorado e impulsado comercialmente en los laboratorios de la compañía Xerox. Con su aparición, se logró también dar el paso definitivo a la aparición de los primeros entornos o interfaces gráficas de usuario.

La invención del *mouse* no fue un hecho trivial ni mucho menos una coincidencia, sino que surgió dentro de un proyecto importante que buscaba aumentar la participación del humano en un sistema informático, mejorando la comunicación entre el hombre y la máquina. De esta manera, en Diciembre de 1968 se presentó al público el primer modelo oficial de *mouse* en una presentación que sirvió también para mostrar el primer sistema con interfaz gráfica de ventanas que más tarde se adoptaría por todos los sistemas operativos modernos.



El tener un mecanismo para apuntar una posición en la pantalla de la máquina es, a esta altura, casi indispensable en ciertas aplicaciones, en especial las que tienen entorno gráfico, como un *Web Browser* o algunos videojuegos. Tanto que hoy día el contar con un mouse, nos resulta tan natural y necesario que pasa desapercibido. En diciembre de 2008, el *mouse* cumple oficialmente 40 años. A pesar de que su modelo original fue fabricado con madera y contaba con un aspecto arcaico y de gran tamaño, actualmente su funcionamiento básico sigue siendo el mismo, y al parecer, luego de cuatro décadas no parece haber otro periférico capaz de reemplazar al *mouse* en todos sus usos. No obstante, desde hace ya unos años existe esa intención.

Grupos de investigación, Universidades y grandes corporaciones alrededor del mundo buscan un nuevo mecanismo que reemplace al *mouse* en todas las aplicaciones que hoy día se lo necesita. Utilización de cámaras o parlantes, *touch screens*, *Multi-touch screens*, apuntadores con laser, *Eye-tracking*, son algunos de los mecanismos que están siendo desarrollados para conseguir el objetivo. En este trabajo, nos centraremos específicamente en la propuesta que encuentro más prometedora. El proyecto se denomina ***CamSpace***.

En las líneas siguientes, se expondrán las características de esta tecnología desarrollada por la empresa ***CamTrax Technologies***, así como también sus principales aplicaciones, usos, la manera en que se está extendiendo y otros datos de interés. También se dará un

vistazo a las otras tecnologías del estado del arte que comparten la misma finalidad con *CamSpace*, de manera a justificar mi elección de este proyecto frente al resto.

## 2. Paradigm Shift

### 2.1 CamTrax – La Compañía

CamTrax Technologies es una empresa israelí fundada en marzo de 2007 por Yaron Tanne, actual jefe ejecutivo y principal desarrollador de la compañía. Sin embargo, la idea que desembocó finalmente en CamSpace, viene pensada por el fundador desde hace un tiempo atrás.



Tanne ya llevaba 3 años desarrollando la tecnología que impulsa a CamSpace en su propio departamento ubicado en Tel Aviv. Inicialmente el proyecto era programado y solventado solamente por él.

Recientemente, gracias a empresas con más trayectoria que apoyaron a CamTrax, la empresa ha conseguido de mano de ellas la suma de 200,000US\$ para solventar las futuras investigaciones. Desde poco después de su fundación oficial, las oficinas de CamTrax se sitúan en Ramat Gan, Israel.



El principal producto de CamTrax es, justamente, el expuesto en el presente documento. El nombre comercial del mismo es CamSpace y su objetivo es proporcionar una nueva forma de apuntar la pantalla de una computadora. Una de las áreas que más motivaron al equipo, sin dudas es la de los juegos, pero apuntan de la misma manera a otros tipos de aplicaciones. Sin más preámbulos, entraremos a detallar a continuación las características que distinguen al software.

## 2.2 ¿Qué es CamSpace?

*“Paradigm Shift is the term used to describe a **change in basic assumptions** within the ruling theory of science.”*

Gran parte de los fanáticos y profesionales en el área de los videojuegos, podrían pensar que el futuro de los controles para consolas o computadoras yace en diseños únicos, costosos y quizás con hardware y software complejos. CamTrax parece ir totalmente al lado opuesto: hacer un software gratuito que te permita jugar cualquier juego necesitando únicamente algún objeto que tengas a mano y una cámara Web. Recientemente, en septiembre de 2008, lanzaron la primera versión beta disponible para bajar gratuitamente de CamSpace.

“Any game, any WebCam”, (en español: “Cualquier juego, cualquier cámara Web”) es el slogan del lanzamiento de CamSpace, pero los juegos fueron solamente la primera gran motivación. Esta tecnología ofrece una nueva manera de interactuar con el sistema operativo de la máquina, permitiendo que prácticamente cualquier objeto (tus propios dedos, una botella, una regla, un zapato, etc.) sea un dispositivo de entrada.



## 2.3 Características y funcionamiento del Software

Esta tecnología provee una solución exclusivamente en software que permite el *tracking*, o rastreo, de un máximo de cuatro objetos comunes del hogar en tiempo real únicamente a través de una cámara Web estándar. Este rastreo no tendría sentido si no

se hiciera con una considerable precisión y con un buen tiempo de respuesta. Afortunadamente, en la mayoría de los casos ocurre esto con un nivel muy bajo de consumo de CPU. En secciones posteriores se detallará más extensamente de que depende el correcto funcionamiento del software.

Básicamente, CamSpace consta de dos partes:

- Un dispositivo virtual junto con su controlador; y
- Un agente que corre localmente para hacer el *tracking*.

El agente, que corre en segundo plano, es el encargado de hacer el *tracking* de los ejes  $X$ ,  $Y$  y  $Z$ . Para comenzar a usar uno o más objetos como dispositivos de *input*, CamSpace necesita configurarse correctamente, calibrando las posiciones de los objetos que se utilizaran. Para esto, requiere tres pasos:

1. **Reconocimiento de los patrones existentes en el entorno** capturado por la *Web Cam*. El objeto a ser rastreado tiene que estar fuera del campo de muestreo de la cámara.
2. Una vez que CamSpace reconoce el ambiente, se pedirá al usuario que introduzca dentro del rango de visión, a los objetos que serán rastreados. En este momento, el agente buscará aquellos objetos que salieron fuera del patrón encontrado en la primera fase. Aquí se realiza la **calibración de los objetos reconocidos** por el agente.
3. Una vez reconocido los objetos que hay que seguir, el puntero del *mouse* se **bloqueará** en alguna posición respecto a los objetos. Esta posición depende de la cantidad de objetos usados. A partir de ese momento, el movimiento de los mismos serán los que determinaran la posición del puntero dentro del *display*.



Una vez terminados estos pasos, el *tracking* se hace automáticamente a través de lo capturado por la cámara Web, emulando al mouse, teclado, joystick o algún otro tipo de dispositivo. El dispositivo virtual mencionado renglones más arriba es reconocido por el sistema operativo como un dispositivo más activo en este momento.



Queda algo más que todavía no he mencionado que distingue aún más a CamSpace frente a otras propuestas similares y que, sobre todo, es de gran interés en el campo de la programación. CamSpace no solo es un software de usuario convencional, sino que se lo puede considerar como una plataforma abierta en la que programadores o usuarios pueden utilizar el CamSpace API para crear las propias emulaciones o controladores para juegos en el lenguaje de preferencia.

En la siguiente sección, se presentará más detalles sobre la utilización de CamSpace dentro de aplicaciones y se darán más detalles sobre el API.

## 2.4 Modo de Uso

Actualmente la versión disponible para descarga es para Windows XP o Vista. Según el sitio Web, las versiones para Mac y Linux ya están terminadas en su gran parte, solamente falta estabilizarlas. O sea que, finalmente según la publicidad de CamTrax, para comenzar a usar CamSpace es necesario contar con una máquina con Windows, una Web Cam estándar y algunos objetos arbitrarios.

Bajando un poco más a lo real y concreto, y dejando de lado por un momento la publicidad encontrada en los sitios Web oficiales de CamTrax y CamSpace, centrémonos en como realmente se ejecuta una aplicación mientras se está emulando un *mouse* o un teclado.



Si bien la propaganda de CamSpace dice “*Any game... Any WebCam*”, en realidad no es algo tan trivial ni directo. Cuando alguien esta utilizando una aplicación cualquiera, ésta tiene un conjunto de posibles valores de *input* esperados. Estos valores son determinados por los dispositivos de entrada requeridos para correrla: el *mouse* no solamente tiene un puntero en el *display*, sino también tiene 2 o 3 botones, un *scroll button* y en ocasiones otros más; el teclado tiene claramente muchas teclas; y un *joystick* tiene otra serie de botones. Con todas estas posibilidades, al parecer tenemos dos problemas con el método proveído por CamSpace. En primer lugar, el gran conjunto de valores posibles de entrada parece ser muy grande y con hacer el *tracking* de los tres ejes de coordenadas de objetos arbitrarios lo único que se consigue es emular la funcionalidad del puntero del *mouse*.



Afortunadamente, este software nos ofrece mucho más que eso. CamSpace tiene una gran precisión con respecto a los ejes  $X$  e  $Y$ , y se comporta lo suficientemente bien con respecto a la profundidad capturada. Junto con esta alta precisión y la interfaz proveída por la API de CamSpace, uno puede emular el clic derecho de un *mouse* con un movimiento hacia atrás del objeto rastreado, por ejemplo; o bien, se puede decir que cada vez que rote con respecto al eje  $Y$  el índice, siendo éste rastreado por el agente, sea equivalente a pulsar la tecla *ctrl*. Como un *tracking* puede tener hasta cuatro objetos, haciendo una correcta configuración con ellos se puede tener una emulación bastante buena para casi cualquier juego que no tenga un gran dominio de valores para el *input*.

Solucionado el primer problema del dominio de valores de entrada, queda por solucionar un posible segundo. Ya sabemos que tenemos una manera de emular las entradas esperadas por una aplicación, pero aún no sabemos cómo hacer el “mapeo” entre éstos valores y los patrones rastreados por el agente. Aquí es donde el CamSpace API cobra importancia.

Como se había mencionado previamente, este API hace de CamSpace una plataforma que ofrece funciones que permiten a cada usuario herramientas para crear sus propias emulaciones para el juego que quiera. Entre las herramientas de la interfaz se incluyen funciones para acceder a la cámara Web, cantidad, posición, estado y propiedades de los objetos rastreados, opciones de color y otros datos útiles. También es posible acceder a los *buffers* que maneja el agente junto con los tiempos de rastreo.

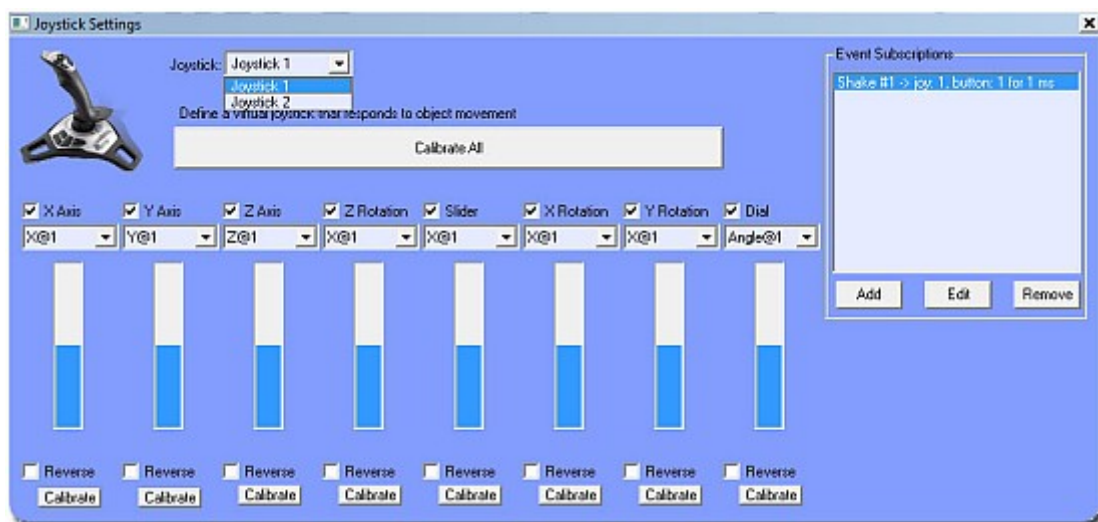
Gracias a esto, uno puede programar la manera en que se hará aquel mapeo mencionado párrafos anteriores. Por ejemplo, para jugar el Solitario Clásico de Windows, se podría configurar el software para que espere dos objetos para ser rastreados. Estos dos objetos se colocarían en la punta de dos dedos de la mano del usuario, los cuales nos servirían para “agarrar” las cartas en el juego. De hecho, este juego fue uno de los primeros con que se comenzó a probar el software para evaluarlo. Posteriormente se volvió bastante



popular entre los usuarios de CamSpace por lo eficaz y cómoda que es la forma de juego.

Pero naturalmente no todos los usuarios son capaces de programar la manera en que quieren que se haga el *tracking* y el mapeo para los juegos o aplicaciones que quieren utilizar. Por eso es que CamSpace permite cargar controladores para una emulación programados por otras personas para usar una aplicación propiamente dicha. Cada controlador especifica cuantos objetos son necesarios y la manera en que hay que moverlos para que sea efectiva la emulación.

La última versión de CamSpace viene incluida con un editor bastante amigable y sofisticado que permite configurar una emulación de una aplicación específica sin recurrir a la programación de más bajo nivel. Este editor es lo que faltaba al paquete para tener un alcance casi total para cualquier tipo de usuario que no le importa tomarse unos minutos para crear su configuración propia y comenzar a jugar el juego que quiera. Con este editor uno puede calibrar a su parecer la sensibilidad del movimiento sobre cada eje en particular, agregar eventos que se disparan cuando se mueva un objeto en alguna manera en especial o tocando alguna región en particular, inclusive escribir un *script* (en Lua) para que algo se comporte como uno desee.



En algunas secciones posteriores veremos que existe ya una gran variedad de juegos con controladores ya programados con muy buena respuesta y calidad. Los juegos van desde muy simples a algunos ya más complejos. Desde el sitio oficial de CamSpace y varios foros alrededor de la Web se pueden bajar los controladores para estos juegos.

El API no sólo ofrece la posibilidad de emular juegos o aplicaciones ya existentes. También te da la posibilidad de hacer aplicaciones propias desde cero. Se puede usar C/C++, Visual Basic, C#, Python o Delphi y recibir la información del *tracking* automáticamente desde el API.

## 2.5 Detalles de la Implementación

Lamentablemente, mucho no se sabe de la implementación propiamente dicha de CamSpace. Lo poco que Tanne ha revelado es que la gran mayoría de los algoritmos que utilizaron son básicamente algunos de los algoritmos de *Motion Track* de conocimiento público, pero, mejorado por su grupo. No se especificaron cuales fueron estas mejoras. Sin embargo, también señaló que la implementación requirió construir nuevos algoritmos desde cero.

## 2.6 Consideraciones para un mejor funcionamiento

Para un mejor *tracking*, se recomienda colocar la cámara en un espacio despejado, con buena iluminación y con pocos colores.



Los objetos rastreados deben ser preferentemente de la siguiente manera:

- Textura y color uniformes;
- Los colores de los objetos deben de hacer aunque sea un leve contraste con el ambiente;
- De forma regular;
- Mayor a 5mm de extensión;

## 3. Juegos compatibles actualmente

### 3.1 Juegos oficiales del sitio

En el sitio oficial de CamSpace figura una lista de una lista de controladores de juegos y aplicaciones disponibles para bajar. Bajando el controlador, uno simplemente necesita obtener los objetos necesarios para la emulación, abrir el controlador (de extensión ‘.camspace’) y ejecutar la aplicación para comenzar a usarla.

Algunos de los controladores más populares figurados en el sitio son los siguientes:

- Track Mania
- Solitario (Clásico y para Vista)
- Ludo Race
- Off Road Arena
- Flight Model Simulator
- Free Cell
- Mario Forever
- Chicken Rush
- Space Bubbles
- Google Earth
- Balance 3D
- Plasma Worm

### 3.2 Juegos configurados por usuarios

Los anteriores son solamente los juegos que figuran en el sitio oficial. Estos juegos son controladores que hicieron los desarrolladores registrados. Sin embargo, gracias al editor mencionado previamente, usuarios no registrados pueden configurar su emulación.

Esta lista de juegos y aplicaciones figura en sitios personales, foros, *youtube* y varios más. Como se podrá apreciar en la lista siguiente, los juegos son más complejos e interesantes:

- Unreal Tournament
- Nexuis
- Counter Strike
- Half Life 2
- Quake
- Need For Speed ProStreet
- Collin McRae Dirt
- Time Fortress 2

También hay una configuración que facilita la navegación en Internet, abrir y cerrar programas, jugar juegos en flash online, ver fotos, o utilizar aplicaciones simples que requieran solamente mover el Mouse y hacer *clicks*.



## 4. CamSpace Vs. Otras propuestas

Existen muchas otras propuestas actualmente en lo que refiere a maneras alternativas a las convencionales de interactuar con la máquina. Si bien no todas son totalmente similares a CamSpace, todas son comunes en lo que refiere a cambiar radicalmente la comunicación Humano-Computadora. A continuación una pequeña lista de algunas tecnologías del estado del arte:

- *The Multi Touch Screen Project*, proyecto liderado por Jeff Han de la Universidad de New York y fundador de la compañía *Perceptive Pixel*. Consiste en un *display* que es capaz de responder a estímulos en múltiples puntos del mismo. Es considerado un genio en este campo al presentar avances significativos en todo momento.
- *Microsoft Surface*, es una combinación de software y hardware: Windows Vista junto con una pantalla *multi-touch*. Lo que distingue a esta tecnología es que la pantalla además está equipada con varios sensores y cámaras que detectan objetos antes que inclusive toquen la pantalla. Algunos objetos, inclusive, no sólo son detectados, sino reconocidos, si es que tienen alguna clave que lo identifique.

- *Nintendo Wii*, si bien esto encaja más dentro del mundo de las consolas de videojuegos, es necesario incluir en esta lista por ser prácticamente la primera propuesta de producción masiva de interacción mediante el movimiento con un sistema. Es evidente la influencia que tuvo el “*wimote*” en el mundo de la tecnología.
- *Eye Motion Tracking*, esta técnica aún no tiene nada concreto, pero en la Universidad de Stanford hay un grupo de investigación trabajando en ello y señalan que en unos años será una realidad. Consiste en reemplazar el *Mouse* por una cámara de altísima precisión que haga *tracking* del movimiento de las pupilas. No hay mucha información sobre lo que han conseguido hasta el momento.
- *Eye Toy*, es una cámara diseñada por Logitech conjuntamente con Sony originalmente para Play Station 2. Esta cámara figura en la lista por tener una idea similar a la de CamSpace. Consiste en rastrear ciertos colores o detectar sonidos del medio para que interactúe con un juego. No todos los juegos eran compatibles, por lo que esta tecnología no tuvo mucho éxito. Posteriormente se dedicaron solamente a juegos infantiles.
- Existen también propuestas que consisten también en una Web Cam rastreando objetos para imitar al *mouse*. Pero difieren con CamSpace en el sentido que solo emulan la funcionalidad del puntero y los objetos rastreados tienen que ser específicos.

Estos son solamente algunos ejemplos. Existen muchas otras empresas trabajando en productos similares. A continuación se exponen algunas de las desventajas y ventajas frente a algunas de estas propuestas anteriores. También se hablará sobre algunos puntos débiles de CamSpace, así como también sus puntos sobresalientes en general.

Para comenzar con los puntos en contra de CamSpace, podemos citar el hecho de que el proyecto no tiene el respaldo de grandes empresas o universidades como algunas de las mencionadas anteriormente: Sony, UNY, Stanford, Microsoft, etc. Pero creo que este punto no lo pone tan en desventaja. Todo lo que han conseguido hasta ahora lo han hecho prácticamente solos. No obstante, el hecho de tener una gran compañía detrás de un proyecto, da algo de garantía al producto. CamSpace es un proyecto apenas conocido a nivel mundial, cosa que tal vez no hubiese pasado si tenía una firma importante. Por estas razones, este punto figura en esta sección.

Por otro lado, la ventaja de ser un proyecto independiente es el costo. El software es gratis, una cámara Web se consigue en cualquier tienda por un precio muy accesible y los objetos a rastrear pueden elegirse arbitrariamente. La posibilidad de que usuarios normales puedan usar algo tan novedoso, es un punto muy favorable. Algunos de los proyectos antes mencionados, son prácticamente impagables para un usuario normal, muchos de esos apuntan a hoteles, restaurantes, etc.

Desde cierto punto de vista, el hecho que requiera configuración, podría ser una desventaja de CamSpace para algunos usuarios. A diferencia de otras tecnologías, este software no es uno que se instala y se usa directamente, sino que requiere configuración previa a cada uso. En el sentido que se pierde algo de tiempo para configurar, esto podría ser tomado como desventaja.

El hecho de ser configurable, es a mi parecer, también una ventaja. Esto permite que usuarios exigentes puedan configurar a su parecer el funcionamiento del *tracking* y de la emulación, dando más libertad al usuario. Obviamente, esto no ocurre con mucho de los otros proyectos.

Como ya se mencionó, CamSpace es una plataforma abierta. Muchos programadores alrededor del mundo están haciendo controladores, aplicaciones y juegos gracias a la interfaz proveída. Inclusive se puede acceder a la API para proyectos totalmente independientes a CamSpace. Esto acerca a la gente al proyecto. No ocurre lo mismo con los proyectos de Microsoft, Sony, etc.

El buen desempeño de CamSpace depende de qué tan bien funcione la Web Cam, la cual es de otra marca que no tiene nada que ver con CamTrax. Esta independencia entre software y hardware hasta cierto punto es buena, pero la aplicación no funcionará correctamente hasta que se revise ese punto. Otro punto débil que tiene el proyecto hasta este momento, es lo difícil que puede llegar a ser configurar correctamente el rastreo cuando no se alcanza lo esperado en cuanto a iluminación de la habitación, elección de los objetos o cosas similares. Pero el equipo de Tanne ya está trabajando para mejorar este aspecto.

## 5. Conclusión

Creo que la propuesta de CamTrax es bastante prometedora. Ofrece no sólo una forma radicalmente nueva de comunicarse con la máquina, sino también una interfaz para que uno decida la forma en la que se hará esta interacción. Según mi opinión, a parte del bajísimo costo del producto, es en este punto donde tal vez CamSpace le saca ventaja al resto de las propuestas presentadas en el documento: CamTrax involucra y alienta a la gente a ser parte de esto. Es cierto que el código del software no es libre, pero provee la interfaz necesaria para hacer a nuestra conveniencia la interacción. Inclusive me inclino a pensar que cuando salgan las versiones para Mac y Linux, crecerá con más rapidez la cantidad de personas que interesadas en el proyecto, habiendo cada vez más controladores para juegos y aplicaciones generales.

Lo conseguido por este proyecto hasta el momento, es para CamTrax Technologies tan sólo la punta del iceberg. No olvidemos que el propio *mouse* tardó más de 20 años en aceptarse completamente, junto con la salida comercial de la interfaz gráfica. CamSpace es capaz de entrar en varias áreas no sólo la del *gaming*. Puede ser muy útil para diseño y modelado 2D y 3D, programas para discapacitados, navegación en 3D, etc.

Es importante darse cuenta que gran parte del proyecto lo hicieron un grupo de menos de 5 personas y auto solventándose. La empresa recién está comenzando a percibir apoyo económico y ya lanzaron el primer beta estable con una muy buena respuesta de los usuarios. Creo que con un poco más de tiempo, CamSpace se podría convertir en una realidad a nivel mundial, con una respuesta mucho más masiva en sus posteriores versiones.



## 6. Bibliografía

- [www.camspace.com](http://www.camspace.com)
- [forum.camspace.com](http://forum.camspace.com)
- [wiki.camspace.com](http://wiki.camspace.com)
- [www.camtraxtechnologies.com](http://www.camtraxtechnologies.com)
- [www.wikipedia.org](http://www.wikipedia.org)
- [www.techcrunch.com](http://www.techcrunch.com)
- [www.engadget.com](http://www.engadget.com)
- [www.gamecyte.com](http://www.gamecyte.com)
- [www.youtube.com](http://www.youtube.com)
- [www.eyetoy.com](http://www.eyetoy.com)