Diagrama de Paquete

Los paquetes Principales del OpticalMarketing :

* InteractionPackage
* CapturePackage
* PreprocessingPackage
* SkeletonPackage
* DataBasePackage
* WebPackage

Se relacionan de la siguiente forma, el paquete de interacción que define el comportamiento como se va realizar la interacción entre las personas y la publicidad, tiene Acceso sobre el paquete captura, este mismo posee el comportamiento de capturar de imágenes, detección movimientos, generando un cambio de estado, para que el paquete Skeleton pueda detectar personas, sus gestos, la posturas y sus movimiento. Así mismo poder realizar la interacción que generara Stream (Flujos de Datos) al paquete de Pre procesamiento de Imágenes y también al paquete Web. Por otro lado el paquete de Pre procesamiento de imágenes que filtras las imágenes controla al paquete de Segmentación como se realizara este comportamiento. La Segmentacion separa la imagen en distinta áreas en función de las características que tenga esa parte de la imagen, para ser tratado de la forma correcta

La comunicación entre el paquete Web y el Paquete de Interacción se realiza mediante Paquete de conexión que se encarga de realizar el nexo entre ambos paquetes, manipulando el Sistema de Archivos (FileSystem) y la Base de Datos. En el siguiente Diagrama figuran los paquetes principales y de soporte.



Diagrama de Clase Del Paquete ColorImageStream

Para Ejecutar la captura de una imagen con el sensor Kinect, se definen dos clases: ColorImageStream y ColorImageFrame. La ColorImageStream, se comporta a bajo nivel con el sensor Kinect definiendo cada cuánto tiempo va tomar frames(cuadros de imágenes), resolución lo va realizar, con que foco, el nivel de luz en función de estos parámetros, realiza la captura con la cámara RGB(Cámara que posee el sensor Kinect) ; y los define dentro de un Stream (flujo de datos) para luego ser Procesado y definido por la clase ColorImageFrame, esta clase transforma ese flujo de datos en una imagen, definiendo el formato : la cantidad de pixeles, el tamaño de la imagen(alto,ancho),



Diagrama de Clases del paquete Capture

En este diagrama se define el comportamiento que necesita para Iniciar la captura de datos (imagen o video).Esta basado con el sensor Kinect.

Para Destacar en este Diagrama son tres clases importante. Primero la interface Device, con esta interface, se describe que propiedades y que estado posee los dispositivos que se utilicen para la captura, se necesita conocer si está conectado el dispositivo, ¿Qué hacer cuando se conecta el dispositivo? y también ¿Cuándo se desconecta el dispositivo? .Segunda clase Kinect, que hereda de Device, además de lo mencionado anteriormente, es importante identificar la Kinect por el hecho que en una aplicación puede haber más de un sensor Kinect conectado, por eso posee un id, también de cada uno de ellos, conocer su estado, en caso de algún error de poder informarlo para que pueda ser tratado, ya que el dispositivo Kinect opera como una máquina de estado, además se identifica cada uno de los controladores(RGBCamera, Microphone, IRProyector, IRCamera, Motor, Accelerometer). Por último tenemos Capture, que va describir que comportamiento se va lograr cuando se inicie la captura que hacer con ese flujo de datos (Stream): definir el formato, en función de estos se definieron dos tipos de captura video e imagen.



Diagrama de clases del paquete FileSystem

Este Diagrama muestra el comportamiento del acceso al File System (Sistema de archivo) de la terminal donde se corre la aplicación. Para ellos definimos un controlador (FileSystemManager) que posee una colección de todos los archivos a los cual va acceder o guardar del File System. Cada archivo(FileSystemElement) para ser accedido necesita un atributo elementUrl de FileSystemElement que es la dirección del directorio donde se encuentra el archivo o va ser guardado el archivo. Principalmente el paquete FileSystem es usado para imágenes.



Diagrama de Clase del paquete HTTPConection

En este Diagrama se detalla el comportamiento de conexión entre un puesto y el servidorWeb. Se define una interface de conexión (ConnectionPackage:IConnection), con esto se logra conocer si se logra la conexión ,como se comporta y los estados. La HttpConnection es un protocolo que se utiliza para estableces un dialogo (paso de mensajes) por ejemplo conectarse, mandar una solicitud, recibir una repuesta. Se define un controlador ManagerConnection que va controlar la conexión entre el servidor y el puesto, qué hacer con lo que recibe del servidor. La funcionalidad principal del paquete HTTPConnection es solicitarte al servidor las imágenes (estáticas cargada por el usuario) que luego son utilizada para el catalogo



Diagrama de clases del paquete Interacción

Este diagrama describe el comportamiento de la interacción de las publicidades.

Se define tres tipos de publicidades: Catalog, AdvertStatistics (publicidad estatica) y InteractiveGame(juego interactivo con fin publicitario). . La Clase Advert es la publicidad tiene definido un tipo de publicidad y tiene asociado un estado. Los distintos estados de las publicidades que se definieron en el puesto son:

* En ejecución.
* En error.
* Activo.
* Desactivado.
* Fuera de Servicio.

Utilizando el patrón State definiendo la interface AdvertState. Se define un controlador AdvertManager, que describe el comportamiento de todos las publicidades en el puesto, por ejemplo saber que publicidad se va ejecutar, en el caso del catalogo cargar las imágenes que muestra por pantalla, ordenar las publicidades, y otras funcionalidades. Por último como mencione anteriormente de cargar imágenes al catalogo definimos una clase ConnectionAdvertList que tiene una lista de los distintos elementos que necesita de otro paquete, esta clase tiene definido la funcionalidad de conectarse con el paquete HTTPConnection.



Diagrama de Clases del paquete Preprocessing

En este diagrama se detalla el preprocesamiento de una imagen aplicando diferentes algoritmos.

Se Definen tres clases importantes : Matrix, Filter y Preprocessor.

La interfaz Filter se utiliza para aplicar distintos tipos de filtro aplicar a la imagen: filtros de bordes, filtro de paso bajo y filtro de paso alto. En el filtro de detección bordes tenemos distintos algoritmos de inteligencia artificial: Sobel, Sharr, Umbral y Canny. En el filtro de paso alto tenemos el algoritmo Laplaciano. Por último en el filtro de paso bajo tenemos Gaussiano, Mediana y Simple.

La Interfaz matrix se define como se trata la imagen, cuyos índices de filas y columnas identifican un punto en la imagen y el correspondiente elemento de matriz identifica el valor de gris o RGB en ese punto.

La clase Preprocessor, recibe la imagen digital capturada, luego convierte esa imagen en una matriz, evalúa esa matriz; en función de esa evaluación filtra la imagen aplicando un determinado filtro.



Diagrama de Clase del paquete Skeleton

Este diagrama describe, el comportamiento de la detección, seguimiento de movimientos, posturas, gestos de una persona. Se destacan tres clases fundamentales: Skeleton – Joint – SkeletonStream.

La clase Skeleton describe a la persona. Posee un Id(TrackingId) por que el sensor Kinect puede detectar hasta 6 personas; dos personas en estado Tracked y las restante en estado PositionOnly, como mencionamos anteriormente posee un estado que puede ser Tracked, es decir, detectada , que puede seguir todos sus movimiento, que reconoce las articulaciones del cuerpo, el otro estado es PositionOnly, que reconoce que hay una persona pero no puede hacer un seguimiento de sus movimientos por que ya hay dos persona que están siendo seguida y por ultimo tenemos un estado no NotTracked, que no detecta la presencia de una persona; Skeleton posee una colección de la clase Joint(articulaciones del cuerpo), cada Joint posee un estado, con un tipo(cabeza,mano,pie,etc) y una posición donde se encuentra , así mismo el Skeleton también tiene una posición en el espacio.En función de esta información se realizan los algoritmos de detección de movimiento, posturas y gestos

La clase SkeletonStream recupera desde la Kinect, los datos de las personas en flujo de datos, posee un array de Skeleton que es de tamaño constante, por el hecho que detecta hasta 6 personas.



Diagrama de clases del paquete Web

Este diagrama describe todas las entidades para el manejo de las publicidades.

Las publicidades forman parte de una campaña, en una campaña pueden tener más de una publicidad que pertenece a un cliente, una campaña se puede ejecutar en más de un puesto publicitario.

La clase AdvertCampaign representa a la campaña, posee un estado, corresponde a un Customer (cliente) con su respectivo Invoice (factura),estimate( cuánto dura la campaña). La clase AdvertCampaignDetail posee una publicidad (Advert), cuando se emite y finaliza esa publicidad; y el precio de la publicidad. La clase Advert descrita anteriormente en el paquete Interaction Package es la publicidad que puede ser un Catalog(catalogo),InteractionGame(juego interactivo),video y publicidad estática. La clase Catalog tiene definido una fecha inicio, fecha fin, fecha actualización y a que cliente pertenece. La clase CatalogDetail posee un CommercialProducto (producto) y el orden que aparece en el Catalog. La clase CommercialProduct tiene definido la descripción del producto, la dirección donde está alojado la imagen en el servidor (urlPath), el nombre, los likes(son los me gusta que seleccionan las personas),views(son cuantas veces fue visto el producto) y un tipo(CommercialProductType).

Los AdvertHost son los puesto publicitarios donde se van emitir las publicidades.



Diagrama de estado de KinectStateMachine

En este diagrama se describe el funcionamiento de la máquina de estado del sensor Kinect, tratando todos los estados que puede tomar. La Maquina de estado del sensor Kinect se crea tomando el estado Initilizating, que es cuando ,el sensor kinect se conecta al host y se prepara para pasar el estado Connected, creando y habilitando los procesos que se necesiten en función del objetivo que tenga que cumplir el sensor Kinect ; por ejemplo: Realizar el seguimiento de una persona , se habilitan determinados procesos. En el estado Initilizating además de pasar a Connected, puede pasar a un estado NotPowered en el caso de que no reciba o se perdió la corriente eléctrica del sensor kinect, puede pasar al estado Error en caso que se produzca error porque no se pueda iniciar algún proceso por que falten parámetros no inicializado o por que la Kinect no es original, puede pasar al estado Disconnected en caso de que se pierda la conexión de Datos(USB) o Se evaluó que por ciertos parámetros o procesos no se puede conectar. En el Estado NotPowered, en este estado el sensor kinect no posee energía eléctrica se informa al usuario para que restablezca la conexión eléctrica para luego pasar al estado Initilizating, y si persiste la falta de no poder restablecerse la conexión eléctrica se pasa al estado Error informando de la situación. En el Estado Connected, ya están habilitado los servicios para ser utilizado, el sensor Kinect se encuentra listo para transmitir Stream(flujo de datos) , en este estado puede también pasar al estado NotPowered, se puede volver al estado Initilizating en caso de querer cambiar la configuración y habilitar otros servicios, en caso de haber fallas con algunos de los componentes del sensor(cámaras, motor, acelerómetro, micrófonos) pasa al estado Error y se pasa al estado Disconnected en caso de querer terminar la operación del sensor. En el estado Disconnected en este estado el sensor se encuentra desconectado por diferentes motivos (falla en los componente, sin señal de datos,se termino la operación del sensor Kinect), se puede pasar al estado Initilizating en caso de querer iniciar determinados servicios para realizar un correspondiente objetivo, se pasa al estado error en caso de querer informar algún error de por qué se llegó al estado Disconnected. En el estado Error, se informa todos los errores al usuario y luego se lo guarda a todos los error indicando el tipo de error con su identificación y prioridad.



Diagrama de Estado de KinectStateMachine

En este diagrama se describe el funcionamiento de una publicidad detallando por todos los estados que pasa. En el estado Preparing, aquí el host (puesto publicitario) solicita a la base de datos que la información sobre la publicidad que va emitir, luego conociendo que imágenes y videos va necesitar , le solicita al servidor HTTP mediante una petición las imágenes y los videos para guardarlo ; y pasa al Estado Disclaiming. En el estado Disclaiming, el host detecta la presencia de una persona y informa que se va ejecutar una publicidad interactiva en la cual va extraer datos de esa interacción por cuestiones legales; si acepta se paso al Estado Snaping. En el estado Snaping se puede aplicar dos estrategias en función del patrón Estrategy: Algoritmo interactivo para anuncios o Algoritmo de monitoreo para rastrear el movimiento de las personas, En principio se encuentra en modo monitoreo se va ejecutar de 1 a n veces siempre que la persona se encuentra mayor a los 2 metros pero si se encuentra menor a los dos metros se ejecuta el algoritmo interactivo. Si se encuentra en el modo monitoreo se ejecuta una publicidad convencional y extrae datos de las personas pasa al estado Saving. Si se encuentra en el modo interactivo se ejecuta la publicidad interactiva pasa al estado Running. En el estado Running Se ejecuta la publicidad interactiva (puede ser el Catalogo), se captura todos los datos que se va generando en la interacción de la persona con la publicida luego pasa al estado Saving para guardar esa información. En el Estado Saving si se llega por el modo monitoreo se procesas las imágenes digitales y se guarda en la base de datos, si se llega por el modo interactivo se guarda la información generada en la base de datos, se puede volver al estado saving si se encuentra en el modo monitoreo para seguir extrayendo datos de las personas si no se puede terminar la interacción pasando al estado Closing. En el estado Closing se termino la publicidad y puede volver a iniciarse el ciclo si todo está en condiciones pasando al estado Preparing o sino terminar el ciclo e informar la situación.



Diagrama de Estado de Skeleton

Este diagrama describe el comportamiento de la captura de un Skeleton(persona) para poder comenzar el anuncio. En el estado Tracked se detecta la persona para poder realizar el seguimiento de movimientos, posturas gesto; y se inicia el anuncio y el cronometro, cuando el sensor pierde el seguimiento de la persona se pasa al estado Inferred. En el Estado Inferred se perdió momentáneamente el seguimiento de una persona, busca a la persona para pasar al estado Tracked en un determinado tiempo; pasado el tiempo se pasa al estado NotTracked. En el estado NotTracked,Se perdió la persona , no hay seguimiento de persona se determina el tiempo que estuvo en estado Tracked en función del cronometro, se guardan las estadísticas.



Diagrama de Despliegue de OpticalMarketing

En este diagrama de despliegue describe la arquitectura de OpticalMarketing.

Los AdvertHost son los clientes puesto publicitarios puede haber de uno a N. Está compuesto Contiene un ordenador con un 32 "o 42" pantalla LED. Conectado con un dispositivo Sensor Kinect, además una aplicación cliente con un framework NET 4.0 y Microsoft Kinect SDK.

Se Emiten anuncios con interacciones e información, se genera información y feedback la guarda en el DatabaseServer. Se solicita videos o imágenes al OpticalMarketingServer necesarios para emitir los anuncios. La información almacenada será consultada por nuestro cliente en los host CustomerClients.

OpticalMarketingServer, es un servidor centralizado que soporta aplicaciones distribuidas, entre ellas un servidor web y un servidor de base de datos.

WebServer: un servidor web que se encuentra en OpticalMarketingServer que opera para recibir consultas y transacciones usando HTTP, HTTPS y WS. Implementa la instancia de Internet Information Server (IIS).

DataBase Server.Base de datos centralizada de OpticalMarketing. Esta base implementa data mining bajo la plataforma Microsoft SQL Server. Está ubicada en OpticalMarketingServer.

Customer Cliente: Es un cliente que quiere contratar a través de un navegador Web, servicios de OpticalMarketing. El cliente puede administrar sus anuncios, ver gráficos e información de los anuncios emitidos en los AdvertHost, pagar los servicios brindados por OpticalMarketing atravez de los servicios PayPal

