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.

