**UNIVERSIDAD TECONOLÓGICA NACIONAL FACULTAD REGIONAL CÓRDOBA**

19/06/2012

**Proyecto Final Optical Marketing SPRINT 1**

**Resumen Primer Entrega**

**Grupo 4 Carlos Kapica 51482 Rodrigo Liberal 51658 Julián Peker 51395**

# Índice

Introducción…………………………………………………………………………………………………3

Objetivos del SPRINT……………………………………………………………………………….…...4

User Stories a realizar del Product Backlog……………………………………………………4

Estimación y prioridades……………………………………………………………………………….4

Tiempo Real Invertido y porcentaje realizado……………………………………………….4

User Story: Investigar, testear características de librerías de Computer Vision.................................................................................................................5

User Story: Investigar testear sobre librerías y Drivers para MS Kinect………….6

User Story: Investigar y desarrollar prototipos sobre librerías de captura y filtrado de imágenes……………………………………………………………………………………..7

Conclusión………………………………………………………………….…………………………………9

# Introducción

En este informe quedarán plasmados los resultados del primer SPRINT y los inconvenientes que pudieran haberse generado.

El equipo de desarrollo ha comenzado a capitalizar los primeros conocimientos y ello se evidencia en resultados tanto de documentos como en las primeras líneas de código.

**Objetivos del Sprint**: Captura y procesamiento inicial de imágenes incluyendo la toma de decisiones de las principales librerías y SDK a utilizar.

User Stories a realizar del Product Backlog:

|  |  |  |
| --- | --- | --- |
| **AssetType** | **Name** | ID |
| Story | Investigar, testear características de librerías de Computer Vision | S-01001 |
| Story | Investigar testear sobre librerías y Drivers para MS Kinect. | S-01002 |
| Story | Investigar y desarrollar prototipos sobre librerías de captura y filtrado de imágenes. | S-01003 |
| Story | Desarrollar, documentar e implementar el módulo de captura de imágenes. | S-01004 |

Estimación y prioridades:

|  |  |  |  |
| --- | --- | --- | --- |
| **AssetType** | **Name** | Estimate | Priority |
| Story | Investigar, testear características de librerías de Computer Vision | 24 | High |
| Story | Investigar testear sobre librerías y Drivers para MS Kinect. | 8 | High |
| Story | Investigar y desarrollar prototipos sobre librerías de captura y filtrado de imágenes. | 16 | Medium |
| Story | Desarrollar, documentar e implementar el módulo de captura de imágenes. | 64 | High |

Tiempo Real Invertido y porcentaje realizado:

|  |  |  |  |
| --- | --- | --- | --- |
| **AssetType** | **Name** | Real | %Acomp |
| Story | Investigar, testear características de librerías de Computer Vision. | 22 | 100 |
| Story | Investigar testear sobre librerías y Drivers para MS Kinect. | 9 | 100 |
| Story | Investigar y desarrollar prototipos sobre librerías de captura y filtrado de imágenes. | 28 | 70 |
| Story | Desarrollar, documentar e implementar el módulo de captura de imágenes. | 60 | 85 |

**User Story: Investigar, testear características de librerías de Computer Vision**

**ID:** S-01001

**Nombre:** Investigar, testear características de librerías de Computer Vision.

**Objetivo:** Conocer sobre las librerías de Computer Vision, e identificar cuáles son las más funcionales y seleccionar la que más contemple los requerimientos del proyecto.

**Cumplimiento del objetivo:** el objetivo fue cumplido exitosamente y a término. Como resultado se seleccionó la librería OpenCV ya que es aquella provee la mayor de las funcionalidades requeridas para el proyecto y posee una buena documentación para sustentar gran cantidad de pruebas.

**Desarrollo involucrado:** investigación a través de la web de las librerías de Computer Vision y selección de la librería de OpenCV. El detalle de selección

**Breve descripción de la Librería de visión por computador seleccionada:**

OpenCV es una [librería](http://es.wikipedia.org/wiki/Biblioteca_(programaci%C3%B3n)) libre de visión artificial originalmente desarrollada por [Intel](http://es.wikipedia.org/wiki/Intel_Corporation). Desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en infinidad de aplicaciones. Desde sistemas de seguridad con detección de movimiento, hasta aplicativos de control de procesos donde se requiere reconocimiento de objetos.

Open CV es multiplataforma, existiendo versiones para [GNU/Linux](http://es.wikipedia.org/wiki/GNU/Linux), [Mac OS X](http://es.wikipedia.org/wiki/Mac_OS_X) y [Windows](http://es.wikipedia.org/wiki/Microsoft_Windows). Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos, calibración de cámaras, procesamiento de imágenes y representación de imágenes.

Esta desarrollada en código C y C++ optimizados, aprovechando además las capacidades que proveen los procesadores multinúcleo.

**Documentos generados:** DAR: OpenCVDAR

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Escala 0-100 | | Escala 0-100 | Escala 0-100 | Escala 0-100 | Escala 0-100 | | | Escala 0-100 | | Escala 0-100 | | Escala 0-100 | | Escala 0-100 | | Escala 0-100 | |  |
|  |  | |  |  |  |  | | |  | |  | |  | |  | |  | |  |
| **Alternativas/**  **Factores** | Facil Instalacion | | Fácil Configuración | Adaptabilidad con Otras Librerías | Bajo Costo | Docu  mentación | | | Compatibilidad Multi  plataforma | | Funcionalidades Implementadas | | Formatos de entrada | | Representacion Matricial | | Lenguajes de Implementacion Soportados | | **Total** |
| **OpenCV** | 90 | | 100 | 90 | 100 | 95 | | | 100 | | 80 | | 100 | | 100 | | 100 | | **955** |
| **Camelia** | 70 | | 40 | 65 | 100 | 40 | | | 50 | | 70 | | 50 | | 70 | | 60 | | **615** |
| **BLEPO** | 60 | | 50 | 80 | 100 | 50 | | | 50 | | 50 | | 80 | | 40 | | 60 | | **620** |
| **CVD** | 70 | | 70 | 40 | 100 | 30 | | | 50 | | 50 | | 45 | | 40 | | 60 | | **555** |
|  |  | |  |  |  |  | | |  | |  | |  | |  | |  | |  |
| **El proposito principal de este DAR es proveer una ayuda y una vision general de las distintas librerias que pueden utilizarse para la edicion,modificacion,procesamiento,mejoramiento y utilizacion de las imagenes capturadas. Teniendo en cuenta que los analisis son subjetivos y estan bajo el criterio de los analizaores, el nivel de exactitud en los resultados puede ser difuso. Aún asi se tomaron los criterios expuestos como los mas influyentes para obtener los mejores resultados.** | | | | | | | | | | | | | | | | | | | |
| **Resultado Final: Se desea utilizar OpenCV.** | |  | |  |  | |  |  | |  | |  | |  | |  | |  | |
|  |  |  | |  |  | |  |  | |  | |  | |  | |  | |  | |

**Pruebas realizadas:**

* Instalación y configuración de OpenCV bajo Linux y Windows.
* Configuración de entorno de desarrollo Netbeans y Visual Studio con OpenCV.
* Ejecución de ejemplos de prueba de OpenCV.

**User Story: Investigar testear sobre librerías y Drivers para MS Kinect.**

**ID:** S-01002

**Nombre:** Investigar testear sobre librerías y Drivers para MS Kinect.

**Objetivo:** investigar cuales controladores son los mejores que facilitan un desarrollo ágil y funcional empleando a MS Kinect.

**Cumplimiento del objetivo:** el objetivo fue cumplido exitosamente y con un pequeño desfasaje. Como resultado se seleccionó los controladores Kinect SDK, ya que son aquellos proveen los controladores con gran cantidad funcionalidad para realizar la implementación del proyecto y además posee una gran documentación.

**Desarrollo involucrado:** investigación a través de la web de los controladores de MS Kinect y selección de Kinect SDK.

**Breve descripción de los controladores para MS Kinect seleccionados:**

Kinect SDK  son controladores y [librería](http://es.wikipedia.org/wiki/Biblioteca_(programaci%C3%B3n))s desarrolladas por Microsoft para controlar MS Kinect. La misma puede integrarse con el Framework .NET 4.0 en adelante.

Kinect SDK solo funciona en [Windows](http://es.wikipedia.org/wiki/Microsoft_Windows). Contiene gran cantidad de clases de acceso al firmware interno de la Kinect y una gran cantidad de funciones para desarrollar aplicaciones donde es empleada la Kinect sobre MS Windows.

Está orientada para trabajar con placas de video y procesadores de mediano rendimiento. Las implementaciones de codificación pueden realizarse empleando los lenguajes C#, C++ y Visual Basic.

**Documentos generados:** DAR: KinectsSDKDAR

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **DAR SDK** | Escala  0-100 | | | Escala 0-100 | Escala 0-100 | Escala 0-100 | Escala 0-100 | | | Escala 0-100 | | Escala 0-100 | | Escala 0-100 | | Escala 0-100 | | Escala 0-100 | |  |
|  | | |  |  |  |  | | |  | |  | |  | |  | |  | |  |
| Alternativas\ Factores | Fácil Instalación | | | Fácil Configuración | Adaptabilidad con Otras Librerías | Bajo Costo | Documentación | | | Compatibilidad Multiplataforma | | Velocidad de captura | | Formatos de salida | | Procesamiento Concurrente | | Lenguajes de Implementacion Soportados | | **Total** |
| **MicrosoftSDK** | 100 | | | 100 | 55 | 40 | 100 | | | 40 | | 80 | | 90 | | 100 | | 40 | | **745** |
| **LibFreenect** | 30 | | | 45 | 70 | 100 | 40 | | | 80 | | 70 | | 70 | | 70 | | 65 | | **640** |
| **OpenNi** | 55 | | | 30 | 70 | 80 | 60 | | | 90 | | 90 | | 60 | | 90 | | 80 | | **705** |
|  |  | | |  |  |  |  | | |  | |  | |  | |  | |  | |  |
| **El proposito principal de este DAR es proveer una ayuda y una vision general de los distintos SDK que pueden utilizarse para la interaccion del equipo Kinect de Microsoft con las distintas plataformas y su nivel de satisfaccion. Teniendo en cuenta que los analisis son subjetivos y estan bajo el criterio de los analizaores, el nivel de exactitud en los resultados puede ser difuso. Aún asi se tomaron los criterios expuestos como los mas influyentes para obtener los mejores resultados.** | | | | | | | | | | | | | | | | | | | | |
| **Resultado Final: Se desea utilizar el SDK de MICROSOFT.** | | |  | |  |  | |  |  | |  | |  | |  | |  | |  | |
|  | |  |  | |  |  | |  |  | |  | |  | |  | |  | |  | |

**Pruebas realizadas:**

* Instalación y configuración de Kinect SDK Windows.
* Configuración de entorno de Visual Studio con Kinect SDK, DirectX 11 y XNA Studio.
* Ejecución de ejemplos de prueba de WPF con Kinect SDK.

**User Story: Investigar y desarrollar prototipos sobre librerías de captura y filtrado de imágenes.**

**ID:** S-01003

**Nombre:** Investigar y desarrollar prototipos sobre librerías de captura y filtrado de imágenes.

**Objetivo:** Conocer sobre las librerías de filtrado de Imagen provistas por OpenCV, para identificar cuáles formatos de captura y filtros de imagen son necesarios para realizar captura y preprosesamiento de imágenes.

**Cumplimiento del objetivo:** el objetivo no fue cumplido exitosamente aun. Se probaron tanto filtros bajos como altos para el mejoramiento de imágenes de la librería OpenCV, filtros para detección de bordes y formatos de representación de captura de imágenes. Se pueden en la actualidad aplicar ciertos filtros de mejoramiento de imágenes realizar obtención parcial de los bordes y realizar representación matricial de imágenes en RGB y escala de grises.

**Desarrollo involucrado:** investigación de filtros y métodos de captura de la documentación de OpenCV, desarrollo de pruebas en C++.

**Breve descripción de lo investigado:**

Los filtros de Blurs son filtros pasa bajo para eliminar el ruido de las imágenes.

Los filtros laplacianos , Canny, Sharr y Sobel son filtros pasa altos son para detectar variaciones de luminosidad.

Realiza captura de imágenes en RGB y escala de Gris de imágenes JPG.

**Documentos generados:**

Pruebas realizas: Pruebas de código en OpenCV.

**Pruebas realizadas:**

* Pruebas de bajo Linux realizadas exitosamente.
* Pruebas realizadas de: Laplaciano, Canny, Sharr, Sobel, threshold, Blurs, Matrices de grises y RGB sobre imágenes en formato JPG.

**User Story: Investigar y desarrollar prototipos sobre librerías de captura y filtrado de imágenes.**

**ID:** S-01004

**Nombre:** Desarrollar, documentar e implementar el módulo de captura de imágenes.

**Objetivo:** Desarrollar un módulo de captura que integre todas las funcionalidades investigadas.

**Cumplimiento del objetivo:** el objetivo fue prácticamente cumplido exitosamente. Se realizo un diagrama de clases de la mayoría del módulo de captura. Falta generar un intérprete de dialogo entre los lenguajes C# y C++ para finalizar el desarrollo de dicho módulo.

**Desarrollo involucrado:** desarrollo de diagrama de clases de Diseño del módulo de captura implementación de codificación en C# y C/C++.

**Documentos generados:**

Pruebas realizas: Diagrama de clases de Diseño del Módulo de Captura .Pruebas de código en OpenCV.

**Pruebas realizadas:**

* Pruebas de bajo Linux y Windows realizadas exitosamente.

Diagrama de Paquetes del Sistema (Version 0.1)



Diagrama de clases de diseño del módulo de Captura (Version 0.1)



Diagrama de clases de diseño del módulo de Preprosesamiento (Version 0.1)

* 

# Problemas e Inconvenientes

Los principales problemas del proyecto se dieron por una mala granulación de las actividades. Estas son muy genéricas y con un nivel de especificación muy bajo, lo cual puede generar ambigüedad y falta de comprensión en quien esté realizando dicha/s actividad/es. Estos problemas se evidenciaron al momento de documentar los avances dado que los revisores a cargo no comprendían específicamente la funcionalidad de varias User Stories y sus eventuales Technical Tasks. Para poder resolver dichos inconvenientes el equipo ha decidido realizar una mayor subdivisión de las tareas, de modo que estas puedan proveer mayor información y de la manera más clara posible.

Por otra parte la inconclusa finalización del primer SPRINT nos demuestra una falta de conocimiento en planificación y administración de los tiempos del proyecto, seguramente, también afectada por las causas mencionadas con anterioridad.

A saber:

En la Technical Task ”Investigar y desarrollar prototipos sobre librerías de captura y filtrado de imágenes” se incorporaron de forma desmedida el análisis de filtros que no realizaban grandes aportes en esta etapa del proyecto, lo cual produjo un desfasaje de los tiempos que se habían propuesto. El equipo ha decidido hacer foco en aquellos que realmente sirvan en las actuales circunstancias, tales como: CANNY(utilizado para la detección de bordes), GAUSSIANO, MEDIA y MEDIANA(para filtrado de ruidos impulsivos), con lo cual se realizaran los primeros avances en la detección de rostros y personas.

En la Technical Task “Desarrollar, documentar e implementar el módulo de captura de imágenes” no se realizaron solamente las tareas abocadas específicamente a ella, sino que el equipo al haberse encontrado con soluciones para futuras Technical Task, decidió seguir investigando sobre estas, lo cual hizo desvirtuar el foco de la misma y al momento de haber resuelto los objetivos principales, no haber finalizado el SPRINT.

# Conclusiones sobre el SPRINT 01

El Sprint01 del proyecto no fue concretado exitosamente, hubieron grandes retrasos en el desarrollo de la UserStory: Investigar y desarrollar prototipos sobre librerías de captura y filtrado de imágenes.

Para concretar eficientemente dicha UserStory se deberá incluir la misma en otro sprint del proyecto.

Por parte del módulo de captura, pudo desarrollarse en su gran mayoría, sin embargo es necesario mejorar aspectos como establecer un mecanismo de dialogo más eficiente entre las capturas realizadas y el preprocesamiento inicial de imágenes.

Para ello se buscarán alternativas como realizar el almacenamiento en base de datos de las capturas, a fin de poseer una mayor integridad de información y buscar un mecanismo de procesamiento más ágil.

Por último podemos destacar que las UserStories: Investigar, testear características de librerías de Computer Vision e Investigar y testear sobre librerías y Drivers para MS Kinect, fueron desarrolladas exitosamente y sin inconvenientes.