

PROGRAMACIÓN AVANZADA

Práctica de Laboratorio:

SVIM

Objetivos

Los siguientes son objetivos que el alumno cumplirá en este trabajo práctico:

- Integrar los conceptos aprendidos en Programación Avanzada.
- Formar parte de un proyecto de desarrollo de software de baja complejidad.
- Trabajar en equipo.

Requisitos

Los siguientes son requisitos necesarios para el alumno en este trabajo práctico:

- Asistir activamente a las clases de teoría, coloquios y problemas durante el cursado de la asignatura.
- Contar con una PC para el desarrollo de la aplicación, en la cual deberá instalar el software QtCreator, preferentemente la versión 5.5 o alguna superior. En los sistemas Windows, se deberá usar el compilador MinGW. La versión de sistemas Linux tiene incluida el compilador GNU g++.
- Enrolarse en el Campus de la FI-UNER: www.bioingenieria.edu.ar/campus y matricularse en la asignatura Programación Avanzada. Aquí se tratarán las consultas referidas al desarrollo del trabajo práctico y se publicarán las novedades del cursado.

Metodología de trabajo

El trabajo consiste en el diseño y desarrollo de un sistema de baja complejidad, el cual consta de una serie de subsistemas. El docente del laboratorio irá presentando cada parte del sistema a medida que se avanza en el cursado. En cada instancia explicará las características técnicas y mejores prácticas a tener en cuenta.

Los alumnos formarán grupos de no más de tres personas para trabajar en el desarrollo del sistema. El grupo podrá consultar sus dudas durante las horas de cursado de las clases de laboratorio, en los horarios de consulta publicadas o a través del campus de la facultad. El plagio no será admitido. La colaboración sólo está permitida dentro del grupo de trabajo. Los docentes guiarán a los alumnos a la resolución de las actividades. Todos los alumnos de un grupo deberán conocer íntegramente lo que han desarrollado y lo deberán defender en las sucesivas publicaciones.

La aprobación del trabajo práctico será dada una vez que se aprueben todos los puntos del trabajo. Para la aprobación de cada uno de ellos el grupo de alumnos deberá realizar una defensa oral explicando lo desarrollado al docente de la práctica y mostrando que el sistema funciona de la forma esperada. En estas instancias deberá estar presente todo el grupo. En los casos en los que un grupo se atrase con la presentación de algún punto podrán presentar el punto demorado en instancias posteriores. Sin embargo, no podrán presentar más de dos puntos por instancia de evaluación.

SVIM

En este trabajo práctico se presenta un proyecto de desarrollo de software del cual los alumnos y docentes formarán parte. Adjunto a esta guía se entrega el documento de especificación de requerimientos del sistema (ERS) y el documento de desafíos.

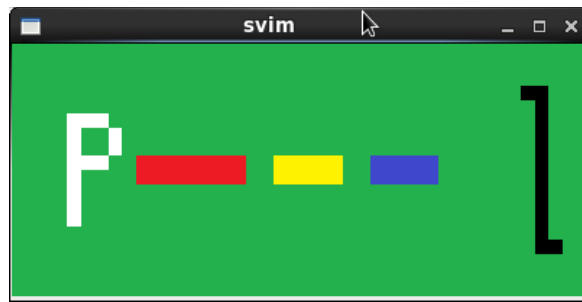
El documento de ERS contiene la descripción del sistema principal y todas las funcionalidades que debe tener. Éste debe ser utilizado por los alumnos para realizar el diseño detallado de la solución. El documento de desafíos contiene la descripción de los diferentes subsistemas adicionales que se pueden implementar. El grupo elegirá uno de ellos como subsistema final y lo desarrollará por completo.

Trabajo a presentar

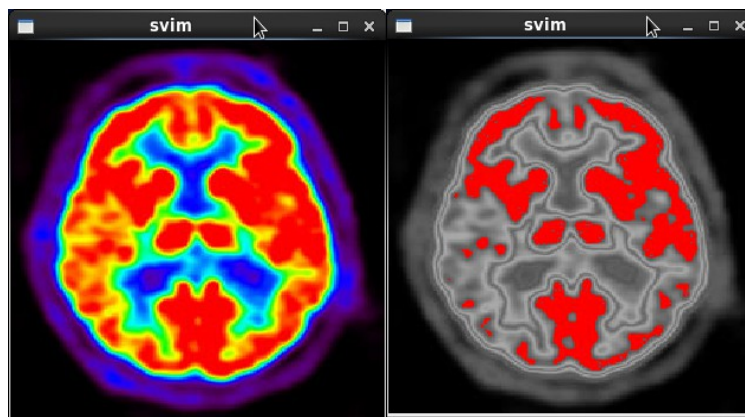
El trabajo de laboratorio está dividido en diferentes módulos, los cuales los alumnos deberán ir desarrollando y presentando a medida que avancen las clases. Para que un módulo pueda evaluarse deberán estar aprobados los módulos anteriores.

A continuación se presentan cada uno de los módulos del trabajo:

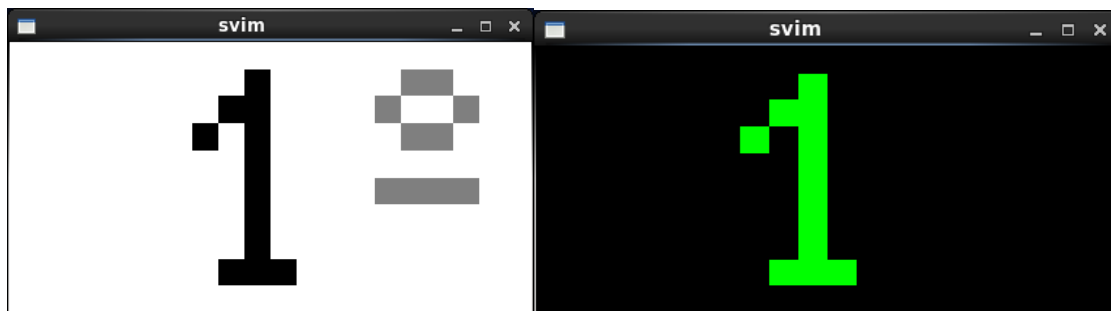
- Módulo 1: Diseño del sistema
 - El diseño del sistema deberá realizarse con la metodología CRC, obteniendo un diseño de alto nivel. Se deberán confeccionar las tarjetas CRC y los diagramas de herencia, contención y colaboración.
 - Se deberá realizar la lista de casos de uso y a partir de ella dos escenarios particulares para dos casos de uso diferentes.
- Módulo 2: Imagen
 - Se deberán implementar las clases que modelen a la imagen en memoria.
 - Se deberá utilizar sobrecarga de operadores para los pixeles.
 - Implementar una interfaz de consola para conocer y modificar los atributos de la imagen.
- Módulo 3: Archivos
 - Implementar las clases relacionadas con la lectura y escritura de imágenes en formato MIF. La lectura y escritura de los metadatos no deberá implementarse; se realizará en un módulo posterior.
 - Adicionar a la interfaz la carga y guardado de una imagen.
 - Verificar que la extensión del archivo sea la correcta (MIF) utilizando objetos strings.
 - Verificar la correcta carga de la imagen en memoria utilizando las siguientes imágenes patrón:
 - *patt01.mif*: Tiene 20 columnas y 10 filas; 20 x 10 metros de dimensión física; un único metadato de código *tipo* y valor *Patron*.
 - *patt02.mif*: Tiene 42 columnas y 18 filas; 42 x 18 milímetros de dimensión física; no tiene metadatos (por lo tanto ese renglón está vacío).
 - *patt03.mif*: Tiene 50 columnas y 35 filas; 50 x 35 nanómetros de dimensión física; dos metadatos: *tipo* - *Patron* y *nombre* - *Rectangulos*.
- Módulo 4: Graficación
 - Implementar las clases que brinden al sistema la posibilidad de visualizar una imagen determinada utilizando las librerías de OpenGL y Qt.
 - Agregar a la interfaz la opción de visualizar la imagen cargada en memoria.
 - Verificar la visualización con la imagen *patt02.mif*, en la cual se debe ver correctamente la letra "P". Los colores que deben verse son: blanco, negro, rojo, verde, amarillo y azul, como se observa en la siguiente imagen.



- Módulo 5: Procesamiento
 - Implementar las clases encargadas del procesamiento de la imagen con los algoritmos propuestos.
 - Anexar a la interfaz las opciones para procesar una imagen con estos algoritmos.
 - Verificar el filtro de color con la imagen *pet01.mif* usando un color rojo puro. En la siguiente imagen se ejemplifica esto.



- Verificar el detector de región con la imagen *patt01.mif*. En la fila 8 y columna 10 debe quedar sólo el número 1 de color verde y fondo negro, como se muestra abajo.



- Módulo 6: Metadatos
 - Implementar las clases relacionadas con la carga, visualización y modificación de metadatos.
 - Use el contenedor map de la STL para la gestión de metadatos en memoria.
- Módulo 7: Desafío
 - Implementar las clases relacionadas al desafío elegido.
 - La interfaz podrá modificarse para que se adapte al desafío.