COMPUTACIÓN GRÁFICA

FRAMEWORKS DE PROGRAMACIÓN V1.5

1. Introducción

El presente documento describe los diversos Frameworks implementados y distribuidos con el objetivo de simplificar la tarea de implementación de los conceptos estudiados a lo largo del curso de Computación Gráfica I de la Facultad de Ingeniería y Tecnologías de la Universidad Católica del Uruguay.

Los Frameworks se componen en su mayoría por una clase Canvas, la cual implementa la funcionalidad básica para crear un Lienzo (rectángulo dibujable) sobre el cual se define la operacin para pintar pixeles.

Las operaciones de pintado de Pixeles, denominadas *putpixel*, pasan a través de un cambio de coordenadas que posiciona el punto (0,0) en el centro del Canvas.

Todos los Frameworks incluyen un archivo adicional que muestra cómo instanciar un nuevo Canvas, colocarlo dentro de la ventana (en caso de que sea necesario) y pintar un píxel sobre él.

Los Frameworks para C y C++ incluyen, a su vez, abstracciones propias para representar colores. Tanto en estos Frameworks como en los de los demás lenguajes, los valores de los canales rojo, verde y azul de cada color se espera que sean valores en el intervalo [0-255].

Todo el código del cual se componen los Frameworks se encuentra disponible, por lo cual no duden en estudiarlo y familiarizarse con él, ya que deberán modificarlo a lo largo del curso para implementar los conceptos que se trabajen en clase.

Se dispone de versiones para los siguientes lenguajes: Java, C#, Python, C++ y C. Una vez decidido el lenguaje, dirigirse directamente a la sección correspondiente.

2. Actualizaciones

Versión	Cambios
1.5	Reescritura de la documentación en LaTeX.
	Mejoras a Frameworks C y C++.
1.4	Correción de errores menores al Framework C++.
1.3	Mejoras de Performance al Framework C#.
1.2	Mejoras de Performance al Framework C#.
1.1	Mejoras de Performance al Framework C#.
1.0	Versión inicial.

3. Framework Java

3.1. **Descripción.** La clase *Canvas* se encuentra implementada en el archivo *Canvas.java*. Ésta se encuentra implementada como un *JPanel* que debe ser agregado como "hijo" componente de un *JFrame*.

Se incluye un ejemplo en el archivo *Main.java*.

3.2. **Operaciones.**

(1) *public Canvas(int w, int h)* - Crear un nuevo *Canvas* de dimensiones w x h.
(2) *public void putpixel(int x, int y, Color c)* - Pintar un píxel sobre el Canvas en (x,y) de color *c*.
(3) *public void clear()* - Borrar el contenido del *Canvas*, reseteandolo a negro.
(4) *public void repaint()* - Refrescar el *Canvas*, mostrando los cambios realizados por las llamadas a putpixel. Es necesario llamarlo para mostrar nuevos pixels pintados.

4. Framework C#

4.1. **Descripción.** La clase *Canvas* se encuentra implementada en el archivo *Canvas.cs*. Esta se encuentra implementada como un *Panel* que debe ser agregado como "hijo" componente de un *Frame*.

Se incluye un ejemplo en el archivo *Form1.cs*.

4.2. **Operaciones.**

(1) *public Canvas(int w, int h)* - Crear un nuevo *Canvas* de dimensiones w x h.
(2) *public void putpixel(int x, int y, Color c)* - Pintar un píxel sobre el Canvas en (x,y) de color *c*.
(3) *public void clear()* - Borrar el contenido del *Canvas*, reseteandolo a negro.
(4) *public void repaint()* - Refrescar el *Canvas*, mostrando los cambios realizados por las llamadas a putpixel. Es necesario llamarlo para mostrar nuevos pixels pintados.

5. Framework Python

5.1. **Descripción.** La clase *Canvas* se encuentra implementada en el archivo *canvas.py*. Ésta implementa la lógica necesaria tanto para crear la ventana como para pintar píxeles sobre la misma.

Se incluye un ejemplo en el archivo *main.py*.

5.2. **Operaciones.**

(1) *public Canvas(int w, int h)* - Crear un nuevo *Canvas* de dimensiones w x h.
(2) *public void putpixel(int x, int y, Color c)* - Pintar un píxel sobre el Canvas en (x,y) de color *c*.
(3) *public void clear()* - Borrar el contenido del *Canvas*, reseteandolo a negro.
(4) *public void repaint()* - Refrescar el *Canvas*, mostrando los cambios realizados por las llamadas a putpixel. Es necesario llamarlo para mostrar nuevos pixels pintados.

6. Framework C++

7. Descripción

La clase *Canvas* se encuentra implementada en los archivos *canvas.h* y *canvas.cpp*. Ésta implementa la lógica necasaria tanto para crear la ventana como para pintar píxels sobre la misma.

También incluyen los archivos *color.h* y *color.cpp*, los cuales definen e implementan la clase *Color* para representar colores, y un ejemplo en el archivo *main.cpp*. A su vez, se incluye un archivo *Makefile* que muestra cómo compliar el proyecto sobre un ambiente UNIX/Linux.

7.1. **Operaciones.**

(1) *public Canvas(int w, int h)* - Crear un nuevo *Canvas* de dimensiones w x h.
(2) *public void putpixel(int x, int y, Color c)* - Pintar un píxel sobre el Canvas en (x,y) de color *c*.
(3) *public void clear()* - Borrar el contenido del *Canvas*, reseteandolo a negro.
(4) *public void repaint()* - Refrescar el *Canvas*, mostrando los cambios realizados por las llamadas a putpixel. Es necesario llamarlo para mostrar nuevos pixels pintados.

8. Framework C

8.1. **Descripción.** Debido a que C es un lenguaje estructurado, no se dispone de una clase *Canvas*, sino que el Framework se encuentra implementado en forma de funciones, dentro de los archivos *canvas.h* y *canvas.c*. Las funciones implementan la lógica necesaria tanto para crear la ventana como para pintar píxeles sobre la misma.

Así mismo, se incluye un tipo opaco *Color* (definido en *color.h*) por conveniencia y se incluye un archivo *Makefile* que muestra cómo compliar el proyecto sobre un ambiente UNIX/Linux.

8.2. **Funciones.**

(1) *void cg_init(int w, int h)* - Crear una ventana de dimensiones w x h.
(2) *void cg_putpixel(int x, int y, Color c)* - Pintar un píxel sobre el Canvas en (x,y) de color *c*.
(3) *public cg_clear()* - Borrar el contenido del *Canvas*, reseteandolo a negro.
(4) *void cg_repaint(void)* - Refrescar el *Canvas*, mostrando los cambios realizados por las llamadas a putpixel. Es necesario llamarlo para mostrar nuevos pixels pintados.
(5) *void cg_close(void)* - Liberar recursos.