



Juan Carlos Hermoso Quesada
41512475-M
Jose Luis Pedraza Roman
80108849-X
Metodología de Programación
2016/2017
Universidad de Granada

INFORME PRÁCTICA 3

Una de las dudas que nos ha tenido surgido durante el desarrollo de la práctica, es la utilidad del constructor público "Imagen" (construye una imagen negra de tamaño "filasxcolumnas") en el archivo "imagen.cpp", ya que según nuestra implementación, sería prácticamente la misma que el método público "void crear" (crea una imagen negra de tamaño "filasxcolumnas"). Esto no nos ha supuesto gran problema en el desarrollo de la práctica.

Un problema que nos hemos encontrado es que la imagen resultante recortada no es exactamente como debería y no sabemos a qué se debe esto.

Otro problema de despiste personal de novato fue que en un principio al terminar de implementar las funciones de imagen.cpp y al ejecutar su correspondiente makefile no nos generaba los archivos ejecutables debido a que no declaramos en un principio su llamada en el apartado all al principio del makefile. Una vez nos dimos cuenta, solucionamos este problema.

Vemos conveniente mostrar una serie de códigos candidatos que hemos ido generando para la consecución del apartado "arteASCII" hasta dar con el idóneo (ninguno de los siguientes lo es):

```
bool Imagen::aArteASCII(const char grises[], char arteASCII[], int
maxlength){ //maxlength: numero de caracteres maximo que puede
almacenar arteASCII
```

```
    int cuantos_g = strlen(grises);    //cuenta cuantos hay, numero
                                         //cardinal
```

```
    int contador_c = 0;
```

```
    int posicion;
```

```
    if (filas() * (columnas()+1) > maxlength){
```

```
        return false;
```

```
    }
```

```
    else{
```

```
        for(int i = 0; i < filas(); i++){
```

```
            for(int j= 0; j < columnas(); j++){
```

```

        posicion = (i*(ncolumnas)+j);

        arteASCII[contador_c]=grises[(getPos(posicion)*cuantos_g)/256];
//256 para que quepa el \0 final, que cierra la imagen

        contador_c++;

    }

}

return true;

}

}

```

```

bool Imagen::aArteASCII(const char grises[], char arteASCII[], int
maxlength){ //maxlength: numero de caracteres maximo que puede
almacenar arteASCII

    int cuantos_g = strlen(grises);    //cuenta cuantos hay, numero
cardinal

    int contador_c = 0;

    int posicion;

    if (filas() * (columnas()+1) > maxlength){

        return false;

    }

    else{

        for(int i = 0; i < filas(); i++){

            for(int j= 0; j <= columnas() + 1; j++){

                posicion = i * ncolumnas + j;

                arteASCII[contador_c]=grises[(getPos(posicion)*cuantos_g)/256];
//256 para que quepa el \0 final, que cierra la imagen

                contador_c++;

            }

            arteASCII[contador_c] = '\n';

```

```

    }

    arteASCII[contador_c + 1] = '\0';

}

return true;

```

```

bool Imagen::aArteASCII(const char grises[], char arteASCII[], int
maxlong){ //maxlong: numero de caracteres maximo que puede
           //almacenar arteASCII

int cuantos_g = strlen(grises);    //cuenta cuantos hay, numero
cardinal

int contador_c = 0;

if (filas() * (columnas()+1) > maxlong){

    return false;

}

else{

    for(int i = 0; i < filas(); i++){

        for(int j= 0; j <= columnas() + 1; j++){

            arteASCII[contador_c]=grises[(get(i,j)*cuantos_g)/256];    //256
para que quepa el \0 final, que cierra la imagen

            contador_c++;

        }

        arteASCII[contador_c] = '\n';

    }

    arteASCII[contador_c + 1] = '\0';

}

return true;

}

```

A la hora de visualizar en arte ASCII imágenes que no fuesen gio.pgm, teníamos un problema de ajuste de imagen, ya que los límites de imagen predeterminados en el array de arteASCII y el parámetro maxlong estaban pensados para gio.pgm en especial o para imágenes más pequeñas. Después se nos ocurrió aumentar esos valores a un millón y así pudimos imprimir en arte ASCII otras imágenes.

Para finalizar, el último inconveniente que nos hemos encontrado, aunque no afecta a la resolución del apartado arteASCII2, ha sido que a la hora de redirigir el programa al archivo de texto no imprime por pantalla ninguno de los mensajes, por lo cual el proceso de creación de una imagen en arte ASCII se realizaba un poco a ciegas.

Por lo demás, el desarrollo de la práctica ha transcurrido con normalidad y la resolución de los ejercicios finalmente ha terminado saliendo correctamente a base de estrujarnos la cabeza mi compañero y yo.