

# Manual de Usuario

## Programa de Dibujo de Circunferencia por Bresenham

El programa que has descargado te permite dibujar una circunferencia utilizando el algoritmo de Bresenham en una ventana gráfica. El programa es altamente configurable y puedes especificar el radio de la circunferencia, así como las coordenadas del centro del círculo al ejecutarlo desde la línea de comandos. A continuación, se describen los pasos para utilizar el programa:

### Compilación Utilizando CMake:

#### 1. Requisitos Previos:

- Asegúrate de tener GLFW, GLEW y Boost instalados en tu sistema. Estas bibliotecas son necesarias para ejecutar el programa. Se recomienda tener vcpkg instalado para administrar las bibliotecas, o utilizar Linux.
- Tener CMake instalado en tu sistema.

#### 2. Configuración del Proyecto con CMake:

- Crea un directorio de compilación en el directorio principal del proyecto:

```
mkdir build
cd build
```

- Ejecuta CMake para configurar el proyecto. Asegúrate de incluir las dependencias necesarias, como GLFW, GLEW y Boost.Program\_options en tu archivo `CMakeLists.txt`. A continuación, se muestra un ejemplo de cómo podría verse la configuración en `CMakeLists.txt`:

```
find_package(OpenGL REQUIRED)
find_package(GLEW REQUIRED)
find_package(GLFW3 REQUIRED)
find_package(Boost REQUIRED COMPONENTS program_options)

# ... otras configuraciones ...
```

```
target_link_libraries(mi_programa ${OPENGL_LIBRARIES} GLEW::GLEW glfw
Boost::program_options)
```

### 3. Configura CMake para utilizar vcpkg (Solo Windows).

- Instala los paquetes necesarios.

```
./vcpkg install glfw3:x64-windows glfw3:x86-windows glew:x64-windows glew:x86-
windows glm:x64-windows glm:x86-windows boost
```

- Configura CMake para que utilice vcpkg.

```
-DCMAKE_TOOLCHAIN_FILE=C:/dev/vcpkg/scripts/buildsystems/vcpkg.cmake
```

- Genera los archivos de construcción:

```
cmake ..
```

### 4. Compilación con CMake:

- Una vez que la configuración se haya completado con éxito, compila el proyecto utilizando el comando `make` (o el equivalente en tu sistema operativo):

```
make
```

### 5. Ejecución del Programa:

- Para ejecutar el programa, utiliza el siguiente formato de línea de comandos:

```
./mi_programa [opciones]
```

## Compilación Utilizando Visual Studio (Sin CMake):

### 1. Requisitos Previos:

- Asegúrate de tener GLFW, GLEW y Boost.Program\_options instalados en tu sistema. Estas bibliotecas son necesarias para ejecutar el programa.
- Tener Visual Studio instalado en tu sistema.

### 2. Configuración del Proyecto en Visual Studio:

- Abre Visual Studio y crea un nuevo proyecto de C++.
- Agrega los archivos fuente del proyecto a tu solución.
- Configura las rutas de inclusión y las bibliotecas necesarias para GLFW, GLEW y Boost.Program\_options en las propiedades del proyecto.

### 3. Compilación en Visual Studio:

- Compila el proyecto en Visual Studio. Asegúrate de configurar la compilación para tu sistema y arquitectura específicos.

### 4. Ejecución del Programa:

- Una vez compilado, ejecuta el programa directamente desde Visual Studio o utilizando la línea de comandos. Utiliza el siguiente formato de línea de comandos:

```
mi_programa.exe [opciones]
```

### Opciones Disponibles:

- Puedes proporcionar las siguientes opciones al programa:
  - `--help`: Muestra las instrucciones de uso del programa y la lista de opciones disponibles.
  - `--x`: Define la coordenada X del centro del círculo (valor entero). Por defecto, es 0.
  - `--y`: Define la coordenada Y del centro del círculo (valor entero). Por defecto, es 0.
  - `--radio` o `-r`: Define el radio del círculo (valor entero). Por defecto, el radio se calcula para que el círculo ocupe 3/4 de la ventana.

### Ejemplos de Uso:

- Para dibujar un círculo con un radio de 100 y el centro en (50, -30), puedes ejecutar:

```
./mi_programa --x 50 --y -30 --radio 100
```

### Interacción con el Programa:

- Una vez que el programa se ejecute, se abrirá una ventana gráfica.

- Puedes cerrar la ventana en cualquier momento presionando la tecla "ESC" en tu teclado.