**Clase QCamera**

La clase QCamera en Qt proporciona una interfaz para manejar cámaras de video o fotografía en una aplicación. Permite controlar dispositivos de cámara, ajustar configuraciones como la resolución, balance de blancos, exposición, y capturar imágenes o videos.

**Características Principales de QCamera**

* **Control de la cámara**: Permite manejar el encendido, apagado y los estados de la cámara.
* **Ajustes de imagen**: Configuración de resolución, balance de blancos, brillo, contraste, y otros parámetros relacionados con la imagen.
* **Captura de video**: Posibilidad de grabar videos y controlar los flujos de video en vivo.
* **Compatibilidad con múltiples cámaras**: En sistemas con múltiples cámaras, permite seleccionar cuál utilizar.
* **Control avanzado**: Soporte para funciones avanzadas como el enfoque automático, exposición, y zoom.

**Métodos Principales de QCamera**

**1. Control de la cámara**

* **QCamera(QObject \*parent = nullptr)**
  + **Descripción**: Crea una instancia de QCamera, seleccionando por defecto el dispositivo de cámara predeterminado.
  + **Ejemplo**:

cpp

Copiar código

QCamera \*camera = new QCamera;

* **void start()**
  + **Descripción**: Inicia la cámara y comienza a emitir datos de video.
  + **Ejemplo**:

cpp

Copiar código

camera->start();

* **void stop()**
  + **Descripción**: Detiene la cámara y deja de emitir datos de video.
  + **Ejemplo**:

cpp

Copiar código

camera->stop();

* **QCamera::Status status() const**
  + **Descripción**: Devuelve el estado actual de la cámara (por ejemplo, si está encendida, apagada, en inicialización, etc.).
  + **Ejemplo**:

cpp

Copiar código

if (camera->status() == QCamera::ActiveStatus) {

qDebug() << "La cámara está activa.";

}

**2. Ajustes de la Cámara**

* **void setViewfinder(QCameraViewfinder \*viewfinder)**
  + **Descripción**: Establece un visor (viewfinder) para mostrar el video en tiempo real.
  + **Ejemplo**:

cpp

Copiar código

camera->setViewfinder(viewfinder);

* **void setCaptureMode(QCamera::CaptureMode mode)**
  + **Descripción**: Establece el modo de captura de la cámara (fotografía, video, etc.).
  + **Ejemplo**:

cpp

Copiar código

camera->setCaptureMode(QCamera::CaptureStillImage);

* **QCamera::CaptureMode captureMode() const**
  + **Descripción**: Devuelve el modo de captura actual de la cámara.
  + **Ejemplo**:

cpp

Copiar código

if (camera->captureMode() == QCamera::CaptureStillImage) {

qDebug() << "Modo de captura: imagen fija.";

}

**3. Control de Configuraciones**

* **void setExposureCompensation(float ev)**
  + **Descripción**: Ajusta la compensación de exposición de la cámara.
  + **Ejemplo**:

cpp

Copiar código

camera->setExposureCompensation(1.0); // Aumenta la exposición

* **void focusToPoint(const QPointF &point)**
  + **Descripción**: Establece el punto de enfoque en la cámara.
  + **Ejemplo**:

cpp

Copiar código

QPointF focusPoint(0.5, 0.5); // Punto medio de la pantalla

camera->focusToPoint(focusPoint);

**4. Control del Zoom**

* **void zoomTo(qreal opticalZoom)**
  + **Descripción**: Realiza un zoom óptico a un valor especificado.
  + **Ejemplo**:

cpp

Copiar código

camera->zoomTo(2.0); // Zoom 2x

* **qreal maximumOpticalZoom() const**
  + **Descripción**: Devuelve el nivel máximo de zoom óptico soportado por la cámara.
  + **Ejemplo**:

cpp

Copiar código

qDebug() << "Zoom óptico máximo:" << camera->maximumOpticalZoom();

**Ejemplo Completo**

A continuación se muestra un ejemplo de cómo usar QCamera para capturar una imagen fija y mostrar el video en un QCameraViewfinder.

cpp

Copiar código

#include <QApplication>

#include <QCamera>

#include <QCameraViewfinder>

#include <QPushButton>

#include <QVBoxLayout>

#include <QWidget>

int main(int argc, char \*argv[]) {

QApplication app(argc, argv);

QWidget window;

QVBoxLayout \*layout = new QVBoxLayout;

QCamera \*camera = new QCamera;

QCameraViewfinder \*viewfinder = new QCameraViewfinder;

camera->setViewfinder(viewfinder);

QPushButton \*startButton = new QPushButton("Iniciar cámara");

QPushButton \*stopButton = new QPushButton("Detener cámara");

layout->addWidget(viewfinder);

layout->addWidget(startButton);

layout->addWidget(stopButton);

window.setLayout(layout);

QObject::connect(startButton, &QPushButton::clicked, [=]() {

camera->start();

});

QObject::connect(stopButton, &QPushButton::clicked, [=]() {

camera->stop();

});

window.show();

return app.exec();

}

**Ejercicios de Consolidación**

1. **Ejercicio 1: Captura de imagen fija**
   * Crea una aplicación que permita capturar imágenes fijas desde la cámara y guardarlas en un archivo local.
2. **Ejercicio 2: Ajuste de parámetros de imagen**
   * Implementa una interfaz que permita al usuario ajustar parámetros como brillo, contraste, exposición, y zoom, y ver los efectos en tiempo real en el visor.
3. **Ejercicio 3: Selección de cámara**
   * Crea un programa que permita seleccionar entre múltiples cámaras disponibles en el sistema y capturar video desde la cámara seleccionada.
4. **Ejercicio 4: Grabación de video**
   * Implementa una aplicación que permita grabar video desde la cámara y almacenarlo en un archivo.

Con esto, tienes una comprensión sólida de la clase QCamera y sus capacidades para manejar cámaras en Qt.