**Lab Exercise 24- Component-onCompleted in QML**

In this exercise, we'll create a QML application that consists of a colored rectangle. Once the rectangle is fully loaded, it will smoothly animate its color from light blue to light green over a few seconds.

Create a QML file named main.qml and add the following code:

import QtQuick 2.0

import QtQuick.Controls 2.15

Rectangle {

width: 400

height: 400

color: "lightblue"

SequentialAnimation {

id: colorAnimation

running: true

loops: Animation.Infinite

ColorAnimation {

target: myRect

property: "color"

from: "lightblue"

to: "lightgreen"

duration: 2000

}

}

Rectangle {

id: myRect

width: 200

height: 200

anchors.centerIn: parent

color: "lightblue"

Component.onCompleted: {

colorAnimation.start()

}

}

}

In this exercise, we create a complex QML application that includes a rectangle. The SequentialAnimation element smoothly animates the color of the rectangle from light blue to light green over a period of 2000 milliseconds (2 seconds) and loops indefinitely. The Component.onCompleted signal is used to start the color animation once the rectangle is fully loaded.

To run the QML file, you can use the following Python script:

import sys

from PyQt5.QtWidgets import QApplication

from PyQt5.QtQuick import QQuickView

from PyQt5.QtCore import QUrl

if \_\_name\_\_ == '\_\_main\_\_':

app = QApplication(sys.argv)

view = QQuickView()

url = QUrl.fromLocalFile('main.qml') # Provide the path to your QML file here

view.setSource(url)

view.show()

sys.exit(app.exec\_())

Make sure to replace 'main.qml' with the actual path to your QML file. When you run the Python script, it will display the QML application, and the color of the rectangle will smoothly animate from light blue to light green over 2 seconds once it is fully loaded. This exercise demonstrates a more complex use of the Component.onCompleted signal along with animation in QML.