Qt Animation Framework 分析与设计

作者：moriv4

日期：2020-6-27

联系：985811440@qq.com

目录

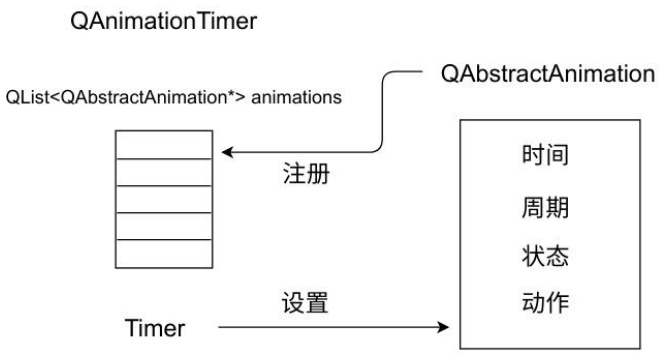
1. 简介

这篇文章简单介绍 Qt 动画框架的原理，阅读本文前先要看 Qt 动画文档并且熟练使用 Qt 动画功能。

2. QAbstractAnimation

抽象动画，Qt 动画框架 API 中的抽象基类。它定义了动画最基础的数据类型，例如“播放-暂停-终止”状态和动作，循环次数，当前的时间和周期等。

设计思想：



时间与周期	动画和时间是有关联的，所以这个类里面有一系列和时间有关的成员变量和函数。 当前时间 currentTime，动画总时间 duration，周期 loop。还有这些变量有关的读取、设置函数。
注册机制	每一个动画都有自己的计时器吗？不是的，从性能的角度考虑，它们共用一个全局计时器效果会更好。这个计时器“驱动”它下面的各个动画，每隔 delta 时间，调用 setCurrentTime()设置动画的当前时间。setCurrentTime()又调用 updateCurrentTime()完成动画的具体动作。 当动画初始化或者状态变化的时候，要向 QAnimationTimer 注册或者取消注册。 QAnimationTimer 控制各个动画的时间。
状态切换	开始、暂停、恢复、停止。不同状态之间的切换要修改时间、周期等相关成员变量，还要向 QAnimationTimer 注册或者取消注册。

2.1. 动画的时间驱动

动画是由时间驱动的。计时器每隔一段时间为动画设置当前时间，动画在每个时间点做不同的动作，于是形成了动态的效果。

时钟脉冲 ==> 系统计时器 ==> Qt Timer 等相关类 ==> QUnifiedTimer ==> QAnimationTimer ==> QAbstractAnimation setCurrentTime() ==> QAbstractAnimation updateCurrentTime()

QAnimationTimer 是单例类，它里面有一个动画指针列表。动画播放的时候会向 QAnimationTimer 注册，每隔一段时间 QAnimationTimer 遍历列表中的所有动画，为它们设置时间（setCurrentTime）。动画被设置时间后会调整自己的时间、周期相关参数，然后调用 updateCurrentTime() 做出不同动作。

继承 QAbstractAnimation 体验一下它的功能

```
class testAnimation : public QAbstractAnimation {

public:

    virtual int duration() const override
    {
        return 1000;
    }

    virtual ~testAnimation() override {}

protected:

    virtual void updateCurrentTime(int currentTime) override
    {
        qDebug() << currentTime;
```

```
    }

};

Widget::Widget(QWidget *parent)
    : QWidget(parent)
{
    testAnimation *tt = new testAnimation;
    tt->setLoopCount(3);
     tt->start();
}
```

输出

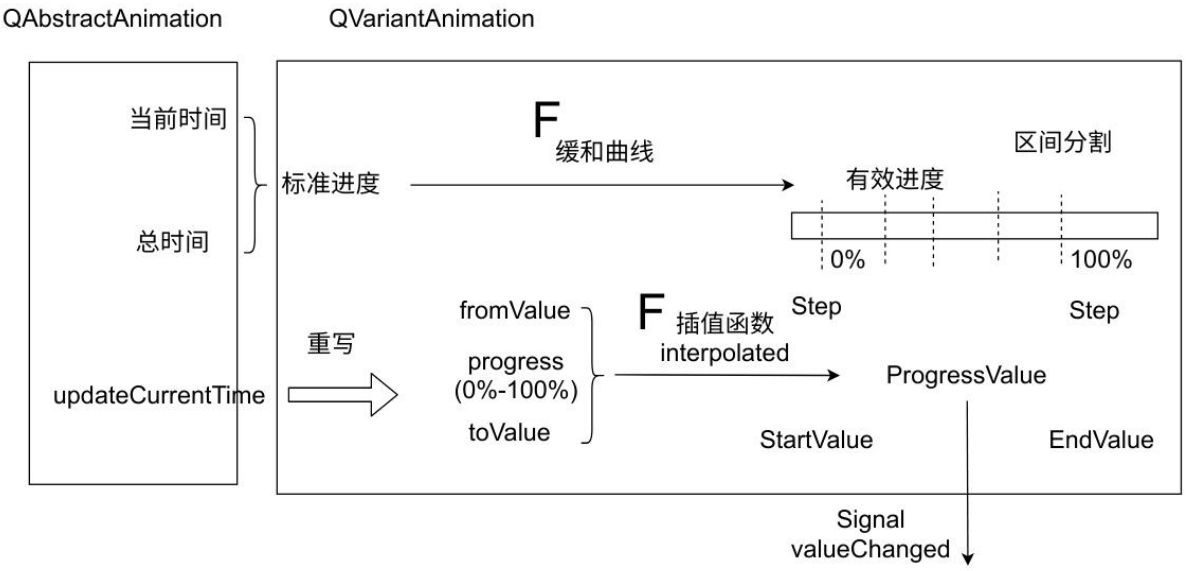
0
16
32
48
65

16ms 刷新一次，是 60Hz 的频率，刚好和显示器的刷新频率同步。

3. QVariantAnimation

变量动画，让变量的值随着时间的流逝而变化

设计思想：



标准进度	标准进度是 0-1 之间的实数，等于 当前时间 / 动画总时间
缓和曲线	类似一个函数，建立“标准进度”到“有效进度”的映射，例如线性曲线、圆锥曲线、弹簧曲线。 实例：篮球自由落体，在地上弹了几下。高度和时间的关系是一个缓和曲线。
有效进度	由标准进度经过缓和曲线变换得到，是动画插值时实际使用的进度
区间分割	把有效进度分割成几个区间，可以给每个区间设置初始值、结束值
值变化	映射，当前有效进度、区间、初始值、结束值 ==>> 当前值。 原版自带的插值函数只支持有限几种类型的变量。
插值函数	根据插值函数计算当前进度对应的值
值变化	上述功能配合的效果，变化时发射 valueChanged(const QVariant &value) ，如果要产生实际效果，需要关联信号到槽。

QVariantAnimation 继承 QAbstractAnimation，增加了上述功能，重写基类的 updateCurrentTime 纯虚函数，在里面调用 recalculateCurrentInterval 重新计算当前区间。先根据时间计算标准进度，再用缓和曲线映射为有效进度，对比一系列区间，确定区间后用 setCurrentValueForProgress 和 interpolated 函数插值。对外提供 valueChanged 信号和 updateCurrentValue 虚函数。

4. QPropertyAnimation

属性动画，让 QObject 的属性值随着时间的流逝而变化。

设计思想：

建立在 QVariantAnimation 的基础上，非常清晰简洁。初始化时设置 QObject 对象和属性名称，重写 updateCurrentValue 函数，函数修改指定对象的属性值。

5. QPauseAnimation

停顿动画

设计思想：

继承 QAbstractAnimation 类，实现 duration 相关函数，重写 updateCurrentTime 函数，函数体为空。什么都不做就是停顿效果。

6. QAnimationGroup

一组动画的容器。QAnimationGroup 继承 QAbstractAnimation，但没有实现 duration 和 updateCurrentTime 函数，所以它也是抽象类。它有一个 QList<QAbstractAnimation *> animations 动画列表，形成一组动画。成员函数是对动画列表这个私有成员变量进行增删查改等常规操作。

7. QParallelAnimationGroup

并发执行的动画组。当它开始播放的时候，容器里面所有的动画都开始播放。

设计思想：

重写 duration 和 updateCurrentTime。并行动画组持续的时间是列表动画中持续时间最长的一个。updateCurrentTime 函数中控制列表动画的并发执行、组的状态、动画的状态等。

8. QSequentialAnimationGroup

串行的动画组。当它开始播放的时候，容器里面所有的动画一个接一个播放。
原理和上面的类似。

完结 END