

🔍 搜索

Qt设计工作室手册 > [动画技术概论](#)

# 动画技术概论



Qt设计工作室手册3.8.0  
Topics >

Qt Design Studio支持以下适合不同用途的动画技术：

- 适用于 2D 和 3D 的常见运动设计技术
- 屏幕到屏幕或状态到状态动画
- 数据驱动的 UI 逻辑动画

## 常见的运动设计技术

下表总结了 2D 和 3D 的常见运动设计技术及其典型用例。

技术	用例
时间轴动画	通过指定关键帧处的中间值进行线性插值，而不是立即更改为目标值。
缓动附加到关键帧的曲线	关键帧之间的非线性插值，使组件在动画结束时看起来加快速度、减慢速度或反弹。

## 基于时间轴和关键帧的动画

时间轴动画基于**关键帧**。在Qt Design Studio中，关键帧确定**组件**在特定时间的属性值。对属性进行动画处理使其值能够在中间值之间移动，而不是立即更改为目标值。

例如，可以将矩形的 y position 属性在动画开始时设置为 0，在动画结束时设置为 100。运行动画时，矩形在 y 轴上从位置 0 移动到 100。在动画中间，y 属性的值为 50，因为默认情况下关键帧是线性插值的。

## 缓动曲线

有时，您不希望线性移动，而是希望矩形在动画开始时移动得更快，在动画结束时移动得更慢。要实现此效果，您可以在开始帧和结束帧之间插入大量关键帧。为避免这种工作，您可以为关键帧之间的非线性插值指定**缓动曲线**。缓动曲线可以使组件在动画结束时看起来加快速度、减慢速度或反弹。

## 动画曲线

虽然缓动曲线适用于大多数简单的 UI 动画，但更复杂的 3D 动画需要多个关键帧，因此有必要同时可视化关键帧的值和插值。“**曲线**”视图可同时可视化属性的整个动画，并显示关键帧的有效值以及关键帧之间的插值。它还可以同时显示不同属性的动画，以便您可以并排查看 x 位置的动画和 y 位置的动画。

## 屏幕到屏幕或状态到状态动画

下表总结了用于在屏幕和 UI 状态之间导航的技术。

技术	用例
应用程序流	可以单击以模拟应用程序的用户体验的交互式原型。
状态之间的转换	使用基于关键帧的过渡时间线在 UI 的不同状态之间进行转换。您可以将缓动曲线应用于关键帧。

## 申请流程

您可以以示意图的形式设计应用程序，该示意图通过符号显示应用程序 UI 的所有重要组件及其互连。这将生成一个交互式原型，可以单击该原型以模拟应用程序的用户体验。代码在后台创建，可用作应用程序生产版本的基础。

有关更多信息，请参见[设计应用程序流](#)。

## 状态之间的转换

UI 旨在不同方案中呈现不同的 UI 配置，或修改其外观以响应用户交互。通常，会同时进行多项更改，以便可以看到 UI 在内部从一种状态更改为另一种状态。

这通常适用于 UI，无论其复杂性如何。照片查看器最初可能会在网格中显示图像，当单击图像时，将更改为详细状态，其中单个图像被展开，界面被更改以显示图像编辑的新选项。在刻度的另一端，当按下按钮时，它可能会变为按下状态，在该状态下，其颜色和位置被修改，使其看起来像是按下的。

任何组件都可以在不同的状态之间更改，以应用修改相关组件属性的更改集。每个状态可以呈现不同的配置，例如：

- 显示一些 UI 组件并隐藏其他组件。

- 执行新状态下所需的一些脚本。
- 更改特定组件的属性值。
- 显示不同的视图。

状态更改会引入突然的运动，您可以通过使用过渡来使其具有视觉吸引力。过渡是插入由状态更改引起的属性更改的动画类型。

在“过渡”中，可以设置每个属性过渡的开始帧、结束帧和持续时间。您还可以为每个动画设置缓动曲线以及整个过渡的最长持续时间。

## 数据驱动的 UI 逻辑动画

下表总结了用于使用后端的真实或模拟数据对 UI 逻辑进行动画处理的技术。

技术	用例
数据驱动的时间轴动画	使用后端的真实或模拟数据来控制运动。
编程属性动画	以编程方式插入属性值以创建平滑过渡。

### 数据驱动的时间轴动画

可以将属性值连接到数据后端以驱动时间轴动画。您可以从各种来源获取数据，例如数据模型、JavaScript 文件和后端服务。您还可以将 UI 连接到 Simulink，以便从 Simulink 模拟加载实时数据。

您可以将这些数据源连接到时间轴的当前帧，从而在后端更改当前帧属性时创建动画。

例如，您可以将速度值从后端连接到集群中的转速表刻度盘。当速度值从后端增加或减少时，它会将指针动画从时间轴的一端移动到另一端。

有关更多信息，请参见[模拟复杂体验](#)。

### 程序化动画

可以通过编程方式控制属性动画。属性动画是通过将动画组件绑定到组件实例的属性值来创建的，以随着时间的推移逐渐更改属性值。属性动画通过在属性值更改之间插值来应用平滑移动。它们提供时序控制，并通过缓动曲线实现不同的插值。

开发人员可以使用 `start()`、`stop()`、`resume()`、`pause()`、`restart()` 和 `complete()` 函数来控制属性动画的执行。

您可以创建 **组件 > 默认组件 > 动画** 中可用的预设动画组件的实例，以根据属性类型和所需行为创建动画。

有关动画组件及其属性的详细信息，请参阅[动画](#)。

元件	用例
属性动画	在属性值更改时应用动画。颜色和数字动画是用于特定目的的属性动画类型。
属性操作	在动画期间设置非动画属性值。
彩色动画	在颜色值更改时应用动画。
数字动画	在数值更改时应用动画。
并行动画	并行运行动画。

暂停动画	在连续动画中创建在指定持续时间内不会发生任何操作的步骤。
脚本操作	在动画期间执行 JavaScript。

[◀ 运动设计](#)

[创建时间轴动画 ▶](#)



联系我们

公司

关于我们  
投资者  
编辑部  
职业  
办公地点

发牌

条款和条件  
开源  
常见问题

支持

支持服务  
专业服务  
合作伙伴  
训练

对于客户

支持中心  
下载  
Qt登录  
联系我们  
客户成功案例

社区

为Qt做贡献  
论坛  
维基  
下载  
市场

© 2022 Qt公司

[反馈](#) [登录](#)



