

当你在Unity中制作动画时，有两个类是关键的：

1. PlayableBehaviour:

- 这就像是你定义动画中角色的特殊行为的地方，比如开始播放时应该发生什么，停止播放时应该发生什么，等等。
- 想象你有一个角色在跳跃，你可以使用`PlayableBehaviour`来定义在跳跃开始或结束时发生的特殊事件。

2. Playable:

- 这是一个通用的动画播放单元。你可以把它想象成一个播放器。但是，它需要知道在播放时应该执行哪些特殊行为。
- `ScriptPlayable` 是一种特殊类型的 `Playable`，它允许你将自定义的 `PlayableBehaviour`（定义了特殊行为的地方）嵌入到播放器中。

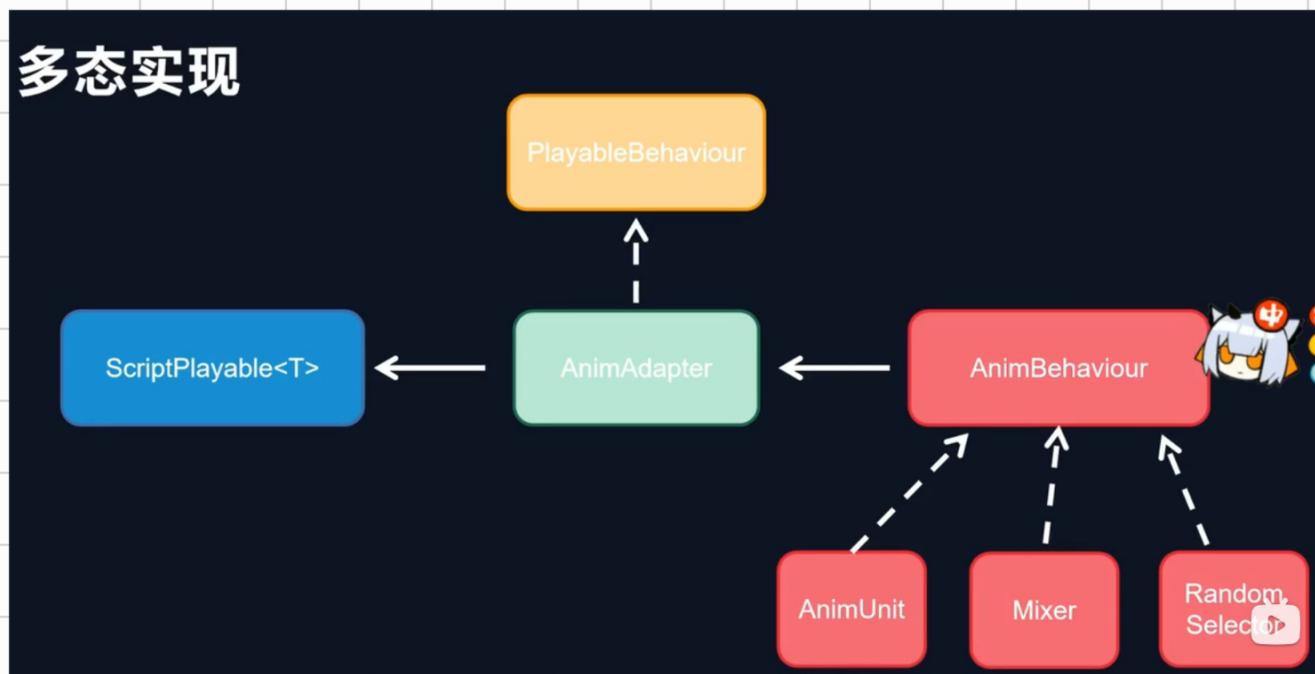
关系总结：

- `PlayableBehaviour` 是你定义特殊动画行为的地方。
- `Playable` 是一个通用的播放器，而 `ScriptPlayable` 允许你将自定义的特殊行为嵌入到这个播放器中，从而在播放动画时实现自定义的效果。

总的来说

playable就是一张图上有很多时间轴，时间轴上有很多节点，可以定义节点中运行的内容，从而在不同时间完成一定的操作

多态实现

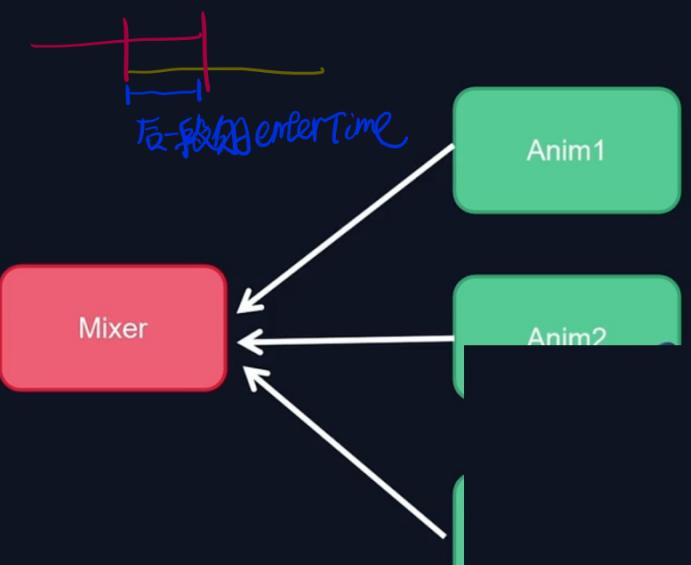


实现一个自己的Playable系统，使用适配器 AnimAdapter，以便于用户定制

Mixer

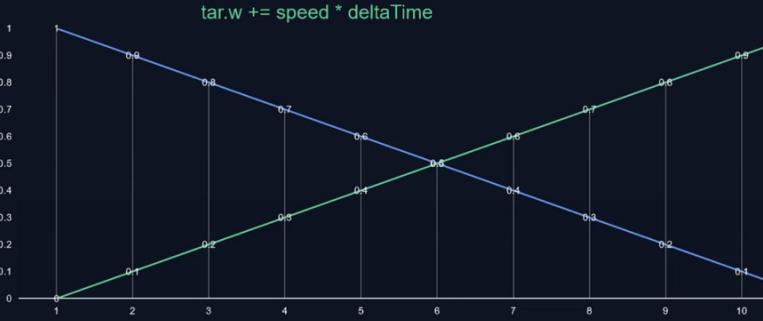
- 从当前状态向目标状态过渡
- 两个状态之间过渡需要一段时间 *enterTime*
- 调用方式：
- 通过 `TransitionTo(int index)`，传入目标状态

current: 0 $\xrightarrow[\text{anim1.enterTime}]{\text{TransitionTo(1)}}$ target: 1
current: 1 $\xrightarrow[\text{anim0.enterTime}]{\text{TransitionTo(0)}}$ target: 0



Mixer

```
speed = 1 / tar.enterTime
cur.w -= speed * deltaTime
tar.w += speed * deltaTime
```



$$\begin{aligned} \text{斜} &= \frac{1}{\text{entertime}} \\ w &+= \text{斜} \cdot \text{delta} \quad (1) \\ y &= \text{斜} \cdot x \quad (2) \\ &\text{on entertime} \end{aligned}$$

之前一直对(1),(2)这两个式子耿耿于怀
高觉得在unity中这样的线性方程好奇怪

$$w_{\text{tar}} = \left(\frac{w_0}{\text{斜}} + \text{delta} \right) \text{斜}$$

$$w_{\text{tar}} = (x_0 + \text{delta}) \text{斜} \Leftrightarrow y = x \cdot \text{斜}$$

Mixer

两个状态切换时，切换到第三个状态
且 当前状态权重 > 目标状态权重

```
speed = 1 / tar.enterTime
delSpeed = 2 / tar.enterTime
cur.w -= speed * deltaTime
del.w -= delSpeed * deltaTime
tar.w = 1 - cur.w - del.w
```

鸢尾零音 bil

