本文档记录Unity中的一些可用编程技术和一些GUI下的组件和功能。

# 编程技术

## Coroutine协程

·Unity协程利用了yield和迭代器机制来实现，具体用法是相当于一个声明位置、执行频率均可自定义的更新过程（不必须在Update和FixedUpdate中），以下是MonoBehaviour中相关的方法：

Coroutine StartCoroutine(IEnumerator routine) //此方法也可以传递string参数

void StopCoroutine(IEnumerator routine)

void StopCoroutine(string methodName)

void StopCoroutine(Coroutine routine)

void StopAllCoroutines()

·默认情况下coroutine每个frame继续从上一个yield处执行，直到执行完毕方法或遇到yield break。

可以使用以下语句暂停本帧的协程执行：

yield return null

可以使用以下语句和对象来调配执行频率：

yield return new WaitForSeconds(x)

·WaitForSeconds受Time.timeScale影响，若需要使用WaitForSecondsRealtime。

·协程的开销和运行过程较复杂，只是在高级层面看起来简单流畅。一般情况下应该尽量避免嵌套调用协程以及反复重新调用同一协程（如果一个协程的运行会被下一个相同协程的调用打断，且存在多次反复的可能性），这时应考虑使用其他技术，或使用更复杂但性能更可靠的代码。

# 界面功能

## Animator/Animator Controller

·Animator Controller是控制动画的状态机，可以在图形界面用节点方式编辑。Animator组件需要Animator Controller。

·Parameters是控制状态机的参数，可以在GUI设置，脚本中获取和更改。

·BlendTree是一种State，这种State是由多个motion组合在一起构成的，每个motion对应一个动画clip。这样可以将多个动画（可能针对不同骨骼）混合到一起在同一状态。

·Transition状态过渡，由Parameters触发，也涉及到两个状态的动画之间的Blend。Exit Time指过渡必须在原状态经历一段时间后才允许检测与触发。