每日总结

# 完成的工作

上午，解决FCollada环境配置问题，google code的Wiki使用已经比较熟练，目录雏形建立。下午去公司总结回顾，决定了下一周的Task。

# 遇到的问题

上周FCollada配置中出现的一些诡异的错误，发现不是配置问题，而是我们的一个类的定义与VS的一个定义冲突。

解决方式，重名名我们的定义。

# 下一天的工作计划

上午研究骨骼动画的层次结构，下午实现架构。

# 经验和感想

对于类的命名，最好加入项目前缀，以防止与公共库冲突。

**从公司获取的Best Practice**

\* Code Review

\* Daily Meeting

让别人了解你做了什么，你也可以从其他人那里获得灵感。

遇到难题尽早的提出来，比自己花大量的时间解决更有价值。我们是团队开发，不是个人舞台。

\* 类的封装

尽可能少得暴露类的内部数据

\* const

尽可能多的使用const以防止你的数据被意外修改。

\* 注释

高级语言是给人看的，好的注释不但可以让别人看懂你的代码，也可以帮你回忆起你的算法与设计。

\* 头文件编写规则

\* 复杂的实现，经常改动的实现，放在cpp中，可以加快代码生成。

\* 尽量减少头文件对其他文件的引用，以防止循环依赖。

\* 一个头文件只放一个类，可以通过文件名就了解内容。

\* 宏#define

尽量使用static const Type 代替，因为#define不做类型检查，而且可能会出现优先级错误

\* new and delete

小心使用new，清楚如何传递与使用，以及谁负责释放。内存泄露将导致很多未知的错误。

\* 代码健康度 assert

关键代码尽量加上assert，可以帮助调试，追踪bug。

\* vector

为了性能以及空间节约，尽量不用vector。

\* 类视图

一张简单明确的类视图要比几页的说明更容易让人明白你的设计。