# 实验目的

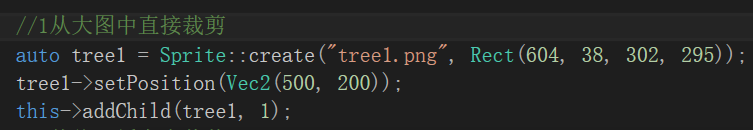
根据课程讲义完成一个精灵加载的工程项目，具体效果见ppt最后的阶段项目HERO。

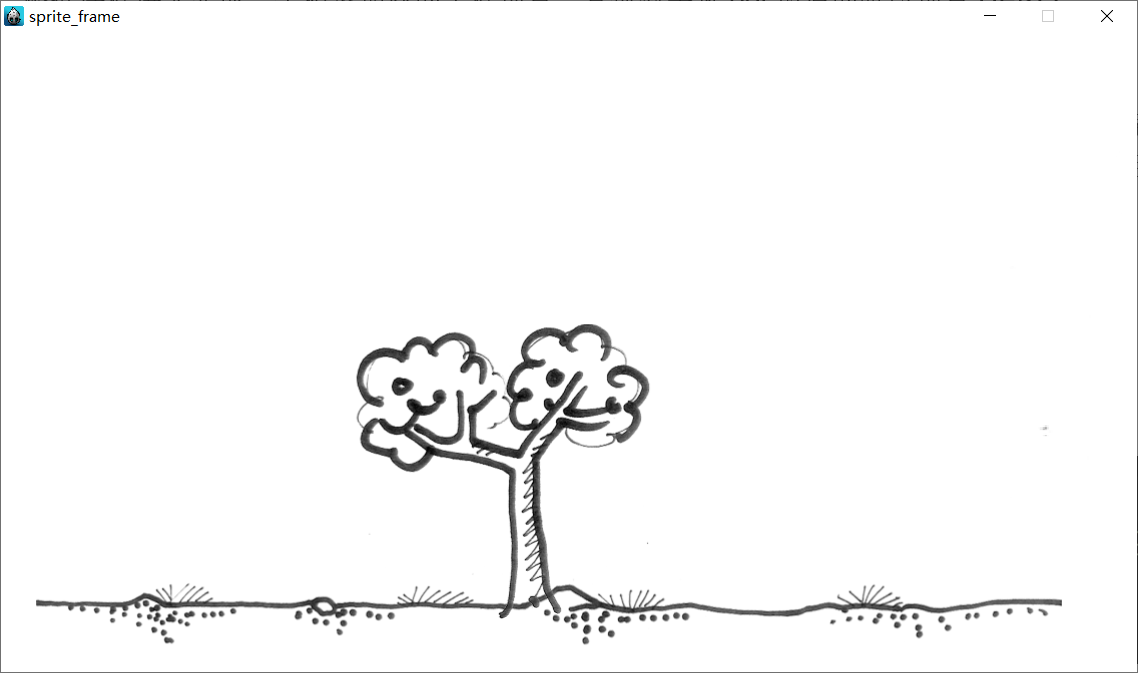
# 实验内容

要求上交实验报告及相应的项目截图。精灵的加载方式要包括：图片加载、精灵帧缓存（从精灵帧、精灵帧名字加载）等各种方式。并要求使用textPacker工具（项目中不一定要使用自己用textPacker生成的资源，但是实验报告中需包含textPacker工具的实际使用截图和总结）

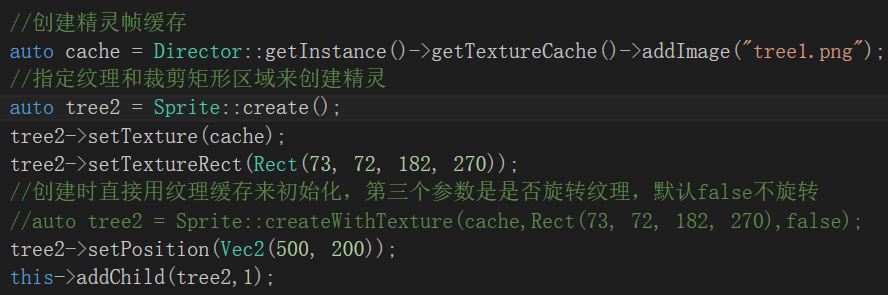
# 实验步骤

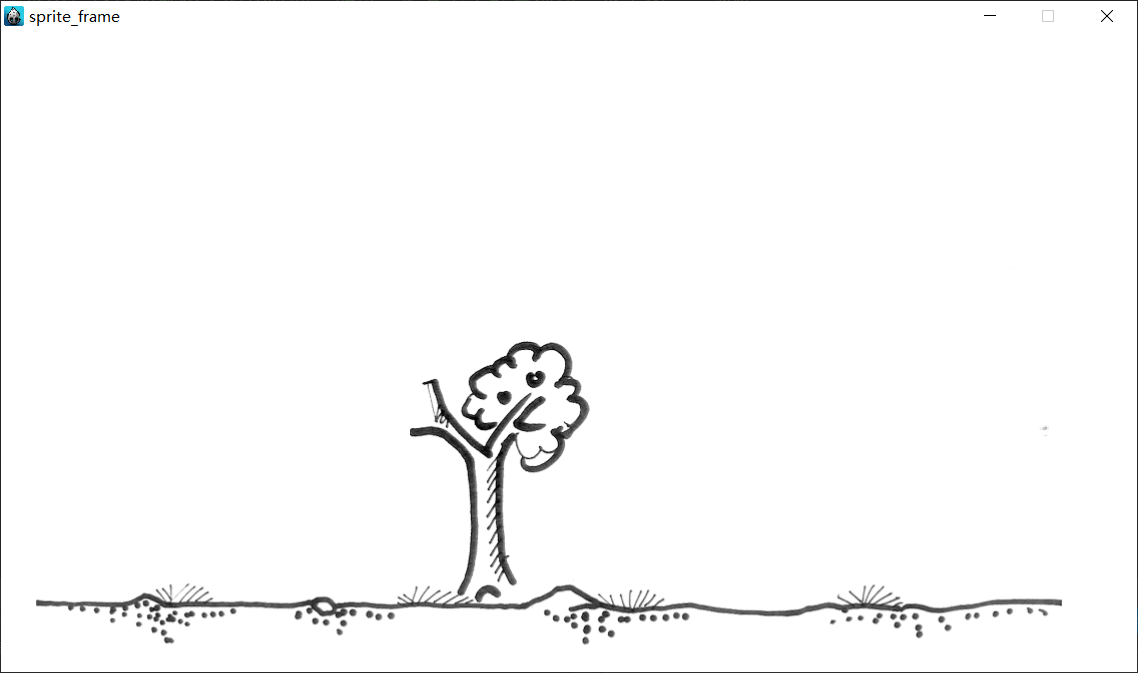
## 从大图直接裁切





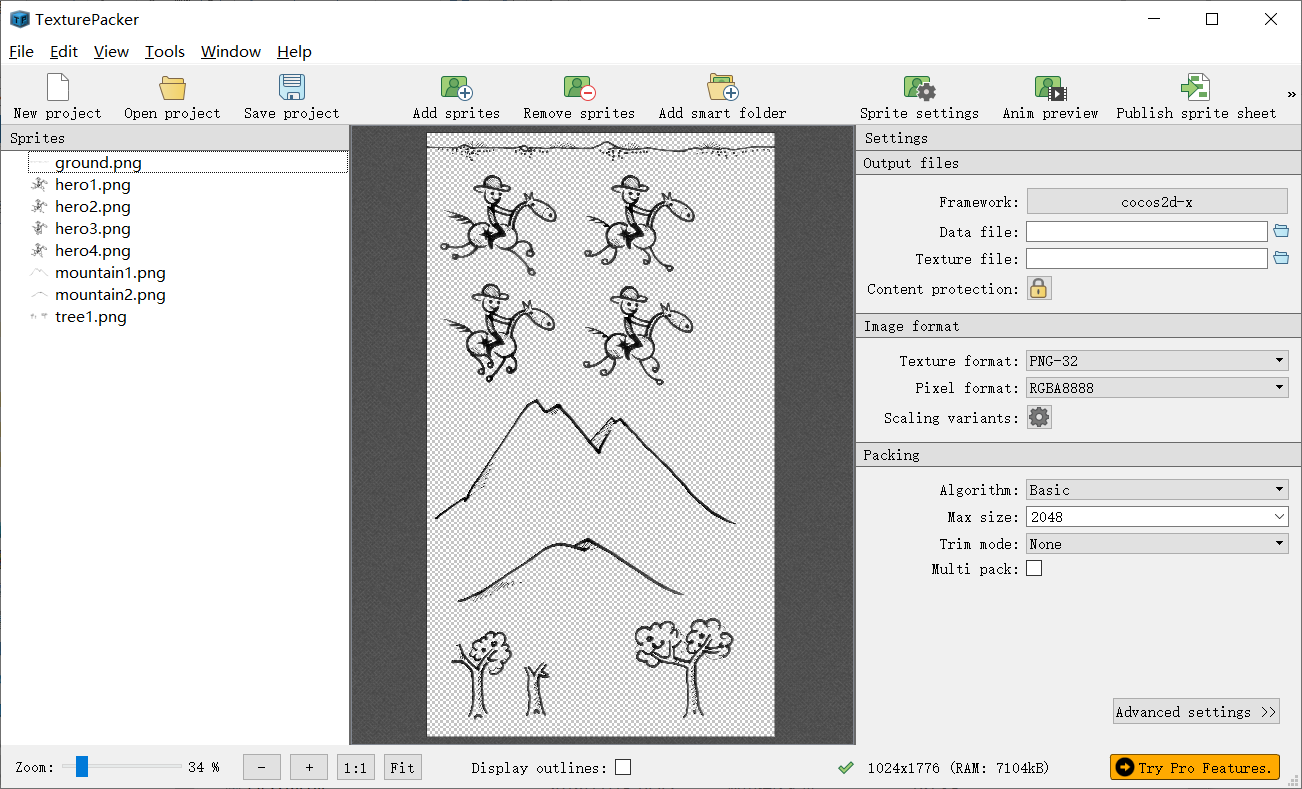
## 使用纹理缓存



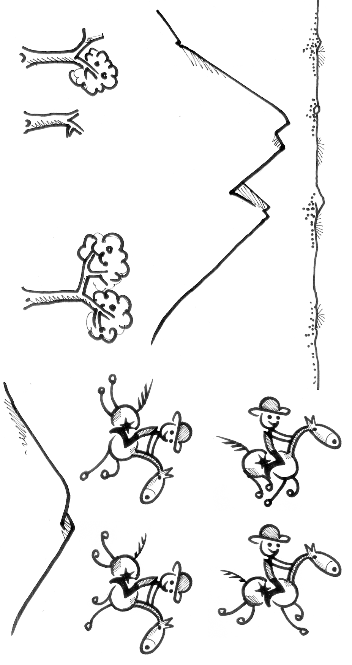


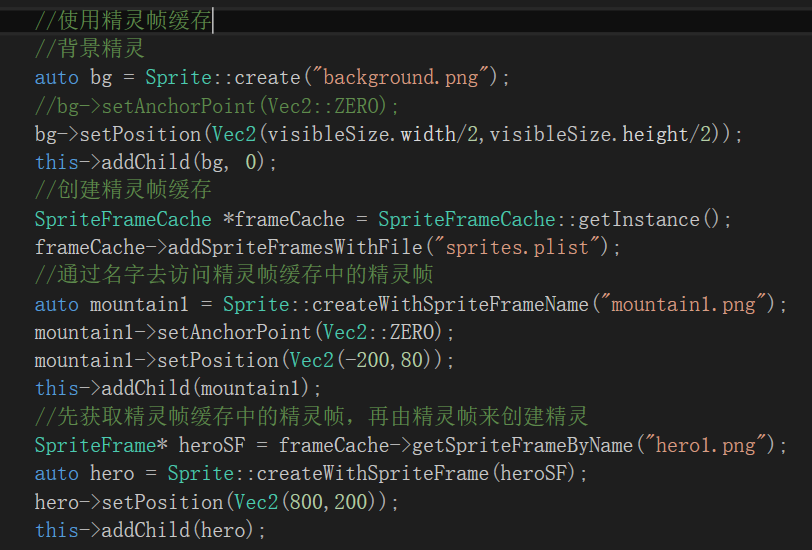
## 使用精灵帧缓存

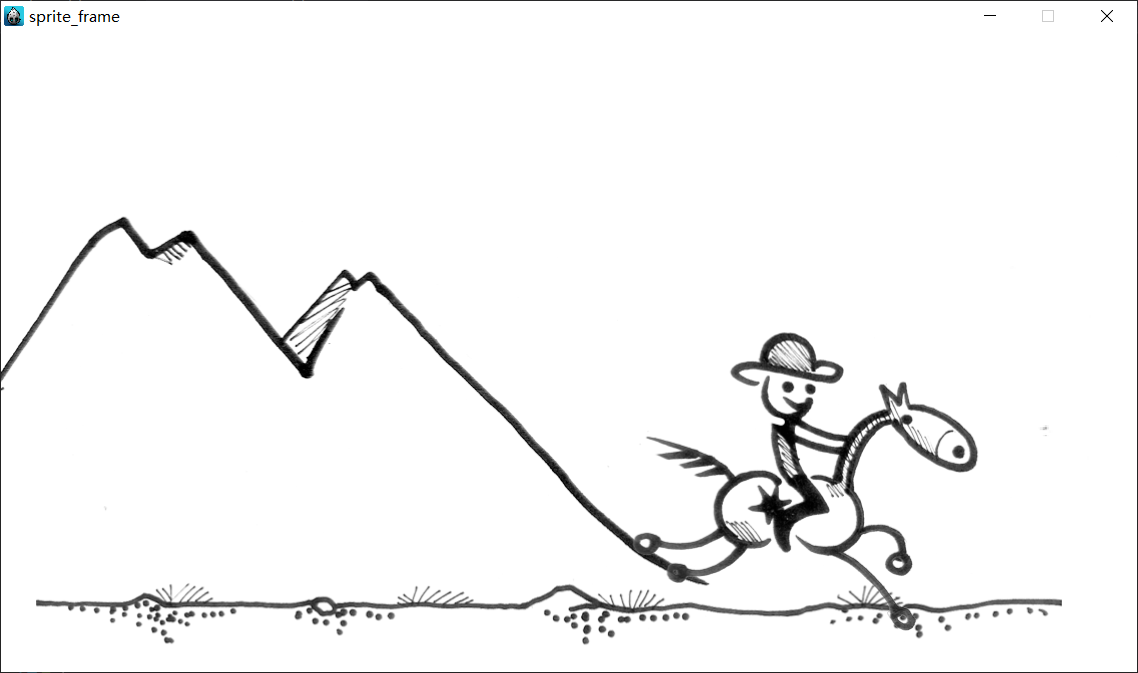
texturePacker的使用



生成的纹理图集sprites.png(这张图是前面生成的)







# 实验心得：

## 从大图直接裁切

一个精灵做一个图片会耗费比较多的IO时间

## 使用纹理缓存

一个精灵做一个图片会耗费比较多的IO时间，所以一般是若干精灵做成一个大图，一次性读入内存（创建纹理缓存)，然后裁剪出来,这种方法内存会耗费多一些。

把大图读入计算机内存中，然后利用纹理缓存反复使用，读入需要的部分存入大图的时候，一般有一个loading的过程。这个时候可以用logo来分散注意力在此过程中会启动一个子线程，进行加载和预处理，后面就比较流畅。

## 使用精灵帧缓存

单例设计模式，整个程序运行过程中只有一个精灵帧缓存。

Step1创建精灵帧缓存(texturePacker生成的大图)：将每个frame(精灵帧，小图)创建出来并放入cache中，以便后续可以反复使用。

Stpe2用精灵帧创建精灵

由于精灵帧缓存的加载过程是游戏启动时，因此该步骤会比较快。

实验评分：

**指导教师签字：**

**年 月 日**