文档说明

1. 说明
   1. 本项目名称为3dTorrent.本项目包含两大模块：3dTorrent和3d torrent tracker server.分别位于3dTorrent和3d torrent tracker server文件夹中。本项目需要部署两个服务器：一台为静态资源服务器，一台为信令服务器
   2. 本项目所有代码使用javascript编写，其中，服务器使用nodejs编写
   3. 本项目访问地址为http://smart3d.tongji.edu.cn/peer
   4. 本项目实现了人物在森林场景中的漫游行走，边走边下载其视点范围内的树木模型。这些模型来源于对等节点的P2P传输.项目截图如下：



1 3d torrent tracker server模块

1.1 这是本项目的信令服务器和tracker服务器实现

1.2 功能：

1.2.1 通过websocket和peer节点进行通信

1.2.2 当节点A上传文件资源R时，信令服务器记录”该节点A拥有的资源R”信息

1.2.3 当节点A请求下载文件资源R时，信令服务器搜索自身存储的记录，找到拥有该资源的若干节点信息并返回给请求节点A

1.2.4 当两个节点之间准备建立webRTC通道连接时，充当信令服务器

* 1. 实现方法：直接调用了NPM下的bittorrent-tracker模块

1. 3dTorrent模块

2.1 这是本项目的静态服务器和浏览器端实现

2.2 功能

2.2.1 实现服务器路由的配置

2.2.2 实现用户P2P上传资源的操作

2.2.3 实现用户P2P下载资源的操作

2.2.4 实现scenegraph的处理和webgl渲染

2.3 实现方法

2.3.1 场景的渲染和角色的控制的代码在peer.html和相关的js文件里

2.3.2 scenegraph的下载和解析的代码在peer.html中实现

2.3.3 P2P上传和下载的操作直接调用NPM下的webtorrent模块

2.4 各文件介绍

Server.js node服务器程序，监听80端口，实现路由功能

Peer.html 该页面是用户打开的页面，进入该页面后用户即开始进行漫游

sendPeer.html 该页面仅作上传初始资源之用

testP2P.html 该页面用于测试P2P文件传输的速率

package.json 该文件用于配置npm模块

node\_modules文件夹 该文件夹是npm依赖的模块所在的文件夹，这些npm模块都是由别人写好打包并放到npm上进行共享

public文件夹 该文件夹是存放静态文件的文件夹，其中包括

1. build文件夹：jquery和three.js库文件
2. data文件夹：存储scenegraph数据和树木模型数据
3. images文件夹：存储贴图文件
4. js文件夹：存储第三人称操作、模型加载器和帧率统计脚本
5. models文件夹：存储角色化身模型文件
6. skybox文件夹：存储天空盒文件
7. bundle.js：经过browserify打包过后的模块文件
8. matrix4.js: 实现四阶矩阵
9. matrixCalculating.js：进行矩阵计算的子线程
10. scene-scale.js: 定义场景大小和坐标操作
11. webtorrent.min.js 依赖的webtorrent模块的压缩版本
12. 流程

3.1 在实验室服务器上打开并运行server.js，在虚拟机上打开并运行tracker.js

3.2 首先配置一台机器，打开smart3d.tongji.edu.cn/send，作为最初始的资源提供节点。点击模型资源进行上传。每次上传的树木模型都会生成一个torrent对象，用以保存模型文件元数据。节点紧接着和tracker服务器通过websocket进行通信，将torrent信息交由tracker服务器保存。

3.3 任意用户登陆smart3d.tongji.edu.cn/peer，首先从服务器下载天空盒、地形等模型（这些模型较小，不通过P2P传输）进行渲染，之后从服务器下载scenegraph.json文件进行解析

3.4 通过解析，用户的内存中存储模型相关信息，包括每一个地形分块内有哪些模型。根据用户的初始坐标位置确定其视点范围内有哪些模型，分别对这些模型请求P2P下载

3.5 用户节点向tracker服务器发送请求，寻找哪些节点拥有该模型，并返回这些节点信息。用户节点与这些返回的节点建立webRTC连接，并建立起data channel.

3.6 模型文件被分割成若干细小的碎片。用户节点和若干节点建立webRTC连接并开始请求传输这些碎片。用户每次随机选择一个webRTC连接传输一个碎片，直到传输全部完成

3.7 用户获得文件的全部碎片后，将其组装完毕，并保存在内存中。同时渲染在浏览器上。

3.8 用户后续如果在漫游中遇到相同的树木，则无需下载模型，直接存内存中取出并加载和渲染即可