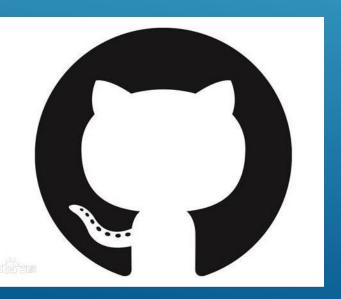
Project Collision

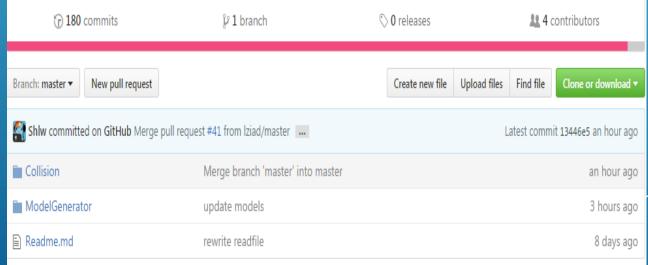
----Shlw

Github

GitHub 是一个面向开源及私有软件项目的托管平台,因为只支持 Git 作为唯一的版本库格式进行托管,故名 GitHub。

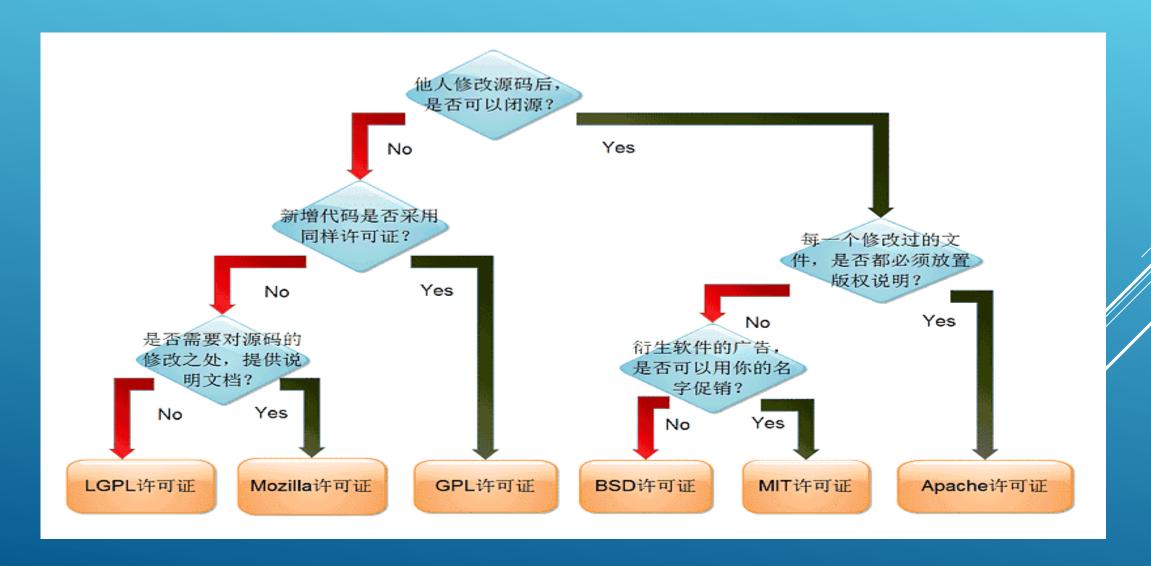
托管代码 + 阅读其他开源项目 + 团队开发





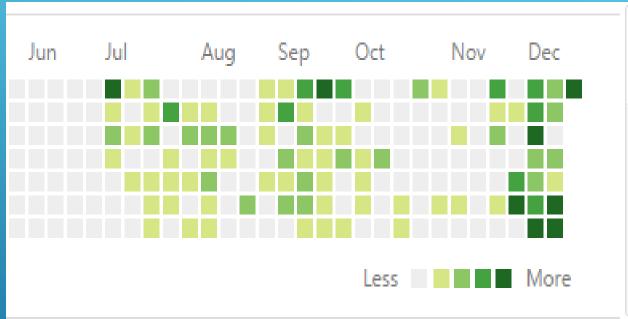


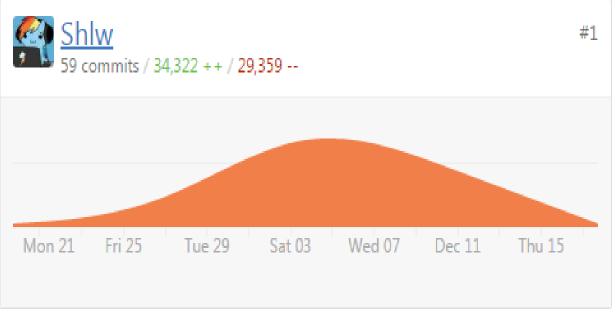
开源协议



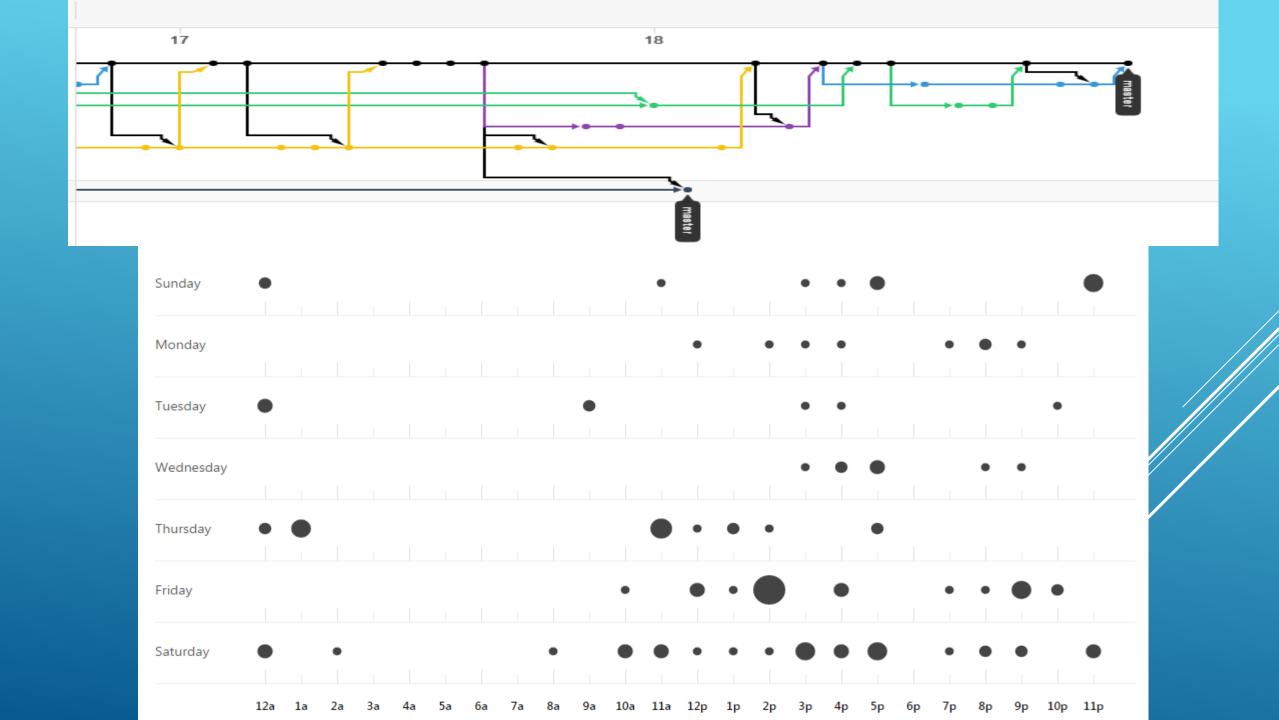
Which Open Source License? Scripting What kind of HTML! -Language code? Library Are you Real-time multi-threaded crazy? C++ Indexer A plugin Which Scripting Language? No. NO!!! What should Perl users do? Python PHP Acknowledge Pav me my work Creative GFDL Python Artistic Commons Digest, adopt, and obey the tenets of Free Software PHP Ah, how crazy How's the are you? weather in Slight Redmond? Family Peanut Butter History Hula Hoops It's called GNU/Crazy Emacs are Fine, why do LGPL People, too! you ask? AGPL But I'm GPL v2 from Sun! GPL v3 How do you like Ms-RL your license? Well-written Damn The Man! Not Anymore Short Apache v2 For what? Public Domain Fine. Oracle. Eclipse But There is Complicated only XUL Which side of the CDDL Mississippi do you live **EPL** on? MPL West East Content: @dbentley Lucent Public BSD 4-clause BSD License Graffle: @therealfitz

Github





https://github.com/Shlw/Stupid_OpenGL



使用方法

- 1.网页版
- 2.Desktop
- 3.Git命令行

一些名词:

Release

Pull Request Repository Fork

Branch
Master
Star
Contributor

Git命令行 🔷

git config -global user.name(user.email) git init git pull origin master git status git add files git commit -m "message" git push origin master(other branch) git differ file git log git reset -hard HEAD^(HEAD~100或hash码)

Git命令行 🚸

```
git checkout - file
git remote add origin http://github.com/......
git clone http://github.com/.....
git branch newlocalbranch
git checkout branch
git merge branch
回到master分支解决conflicts
<<<HEAD表示主分支
>>>branch表示次分支
```

Git命令行 🔷

远程分支 git push origin --delete branch git push origin :branch

本地分支 git branch -d branch

Git命令行 🍑

多人协作工作模式:

- 1.试图用git push origin branch-name推送自己的修改.
 - 2.如果推送失败,则因为远程分支比你的本地更新
- 早,需要先用git pull试图合并。
 - 3.如果合并有冲突,则需要解决冲突,并在本地提

交。

4.再用git push origin branch-name推送。

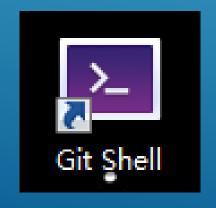
Git命令行 🔷

更多命令学习:

http://blog.jobbole.com/78960/

• • • • • • •

网上教程很多,可以自主学习©







- Collision
- ModelGenerator
- Readme.md

3D Model

- CubeGen.cpp
- CuboidGen.cpp
- EllipsoidGen.cpp
- Readme.md
- SphereGen.cpp
- regenerate.cpp

- Models
- Pictures
- Sounds
- CMakeLists.txt
- Readme.md
- SoundEff.txt
- Texture.txt
- bgm.wav
- collision.cpp
- display.cpp
- draw.cpp
- event.cpp
- game.cpp
- global.cpp
- global.hpp
- update.cpp

- Modelcube.txt
- Modelcuboid.txt
- Modelellipsoid.txt
- Modelsphere.txt
- guai.jpg
- huaji.jpg
- jingku.jpg
- pen.jpg
- weiqu.jpg
- a barehead.wav
- bluecat.wav
- curator_laugh.wav
- duang.wav
- helloeveryone.wav
- icome.wav
- jinkela.wav
- lanlanlu.wav
- proud.wav
- wa.wav
- a zhazha.wav
- zhazha_ex.wav

Class

```
// point class represents the still point in local coordinate system
                                                                        // triangle class represents the still triangle(cone) in local coordinate system
class Point{
                                                                        class Triangle{
public:
                                                                        public:
   int nFlag;
                                                                            PPoint pppVertex[3];
   PVec4 vpCoordinate;
                                                                             PVec4 vpNormalVector;
   PVec4 vpColor;
   PVec2 vpTexture;
                                                                            Triangle();
                                                                            Triangle(PPoint a, PPoint b, PPoint c);
   Point();
                                                                            Triangle(PPoint a, PPoint b, PPoint c, PVec4 v);
   Point(PPoint example);
                                                                            ~Triangle();
                                                                                                // object class contains
   Point(
                                                                                                // the matrix(mFrame) to transform between local and global coordinate system
       float x,float y,float z,
                                                                            void Draw();
                                                                                                // and several indispensable descriptive aspect of the object
       float r,float g,float b,float alpha
                                                                                                 class Object{
   );
                                                                                                public:
    ~Point();
                                                                                                    int nModelType;
                       // model class represents the still model in local coordinate system
                                                                                                     PMat4 mpFrame;
   void DrawVertex();
                        class Model{
                                                                                                    PVec3 vpSpeed;
                        public:
                                                                                                     PVec3 vpAngularMomentum;
                            int nLength; // the number of triangular cones
                                                                                                    float fMomentInertia;
                            PTriangle* tppCone; // the point array of triangle pointers
                            float fVolume, fElastic, fMass, fMaxRadius;
                                                                                                    Object();
                            glm::mat3* mMomentOfInertia;
                                                                                                    Object(int model, float vx=0, float vy=0, float vz=0, float mx=0, float my=0, float mz=0);
                                                                                                    ~Object();
                            Model();
                            ~Model();
                                                                                                    PTriangle IsInside(PVec4 tp, PVec3 vdir=NULL);
                                                                                                    //ymw changed tp from PPoint to PVec4
                            void Draw();
                       };
                                                                                                    void Draw(int index);
                                                                                                    void Update();
```