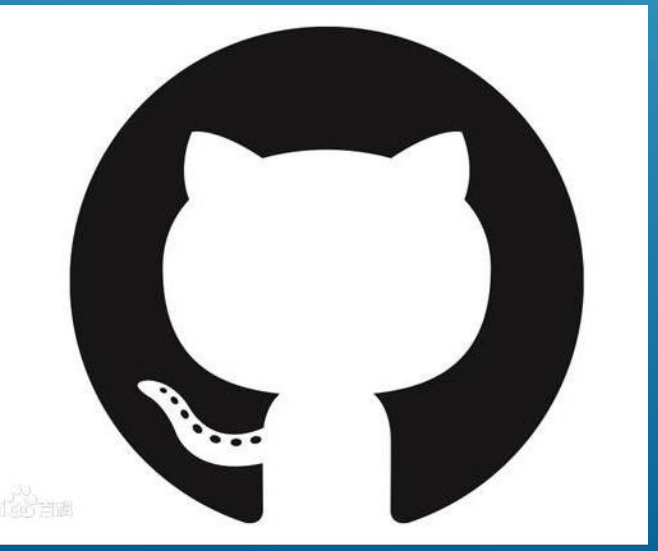
Project Collision

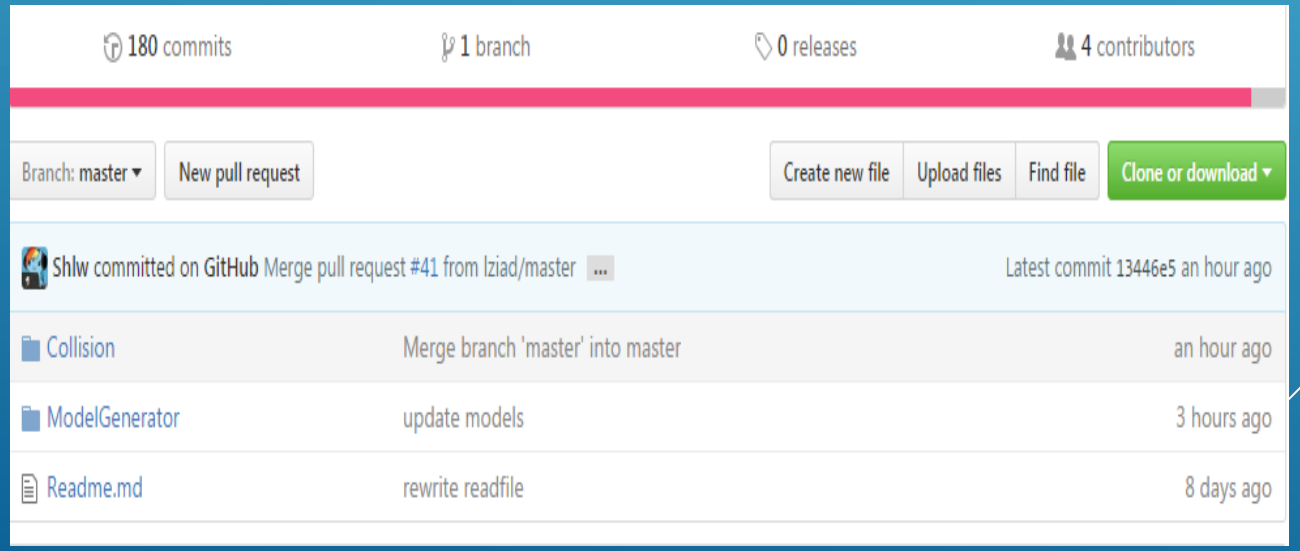
-----Shlw

Github

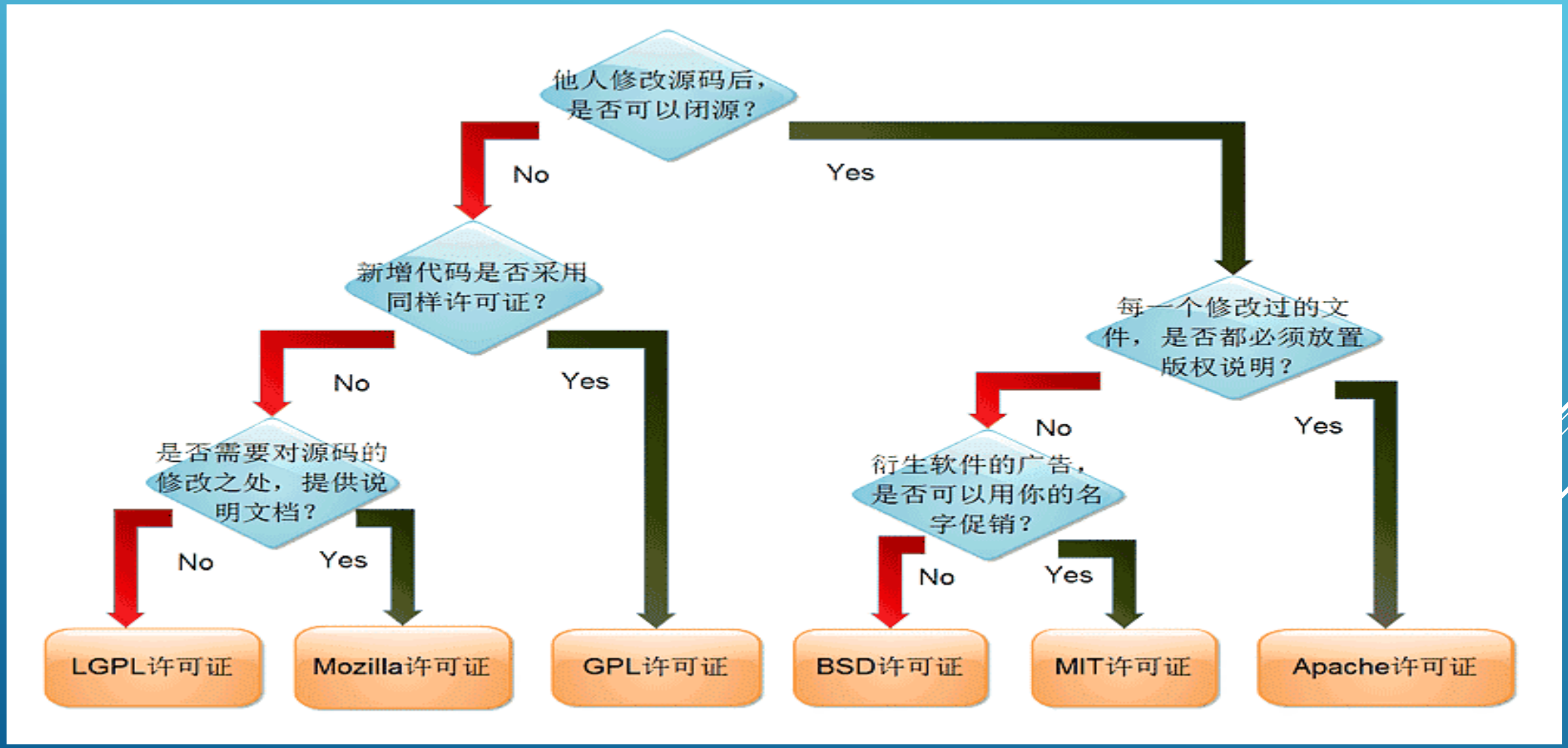
GitHub 是一个面向开源及私有软件项目的托管平台，
因为只支持 Git 作为唯一的版本库格式进行托管，
故名 GitHub。

托管代码 + 阅读其他开源项目 + 团队开发

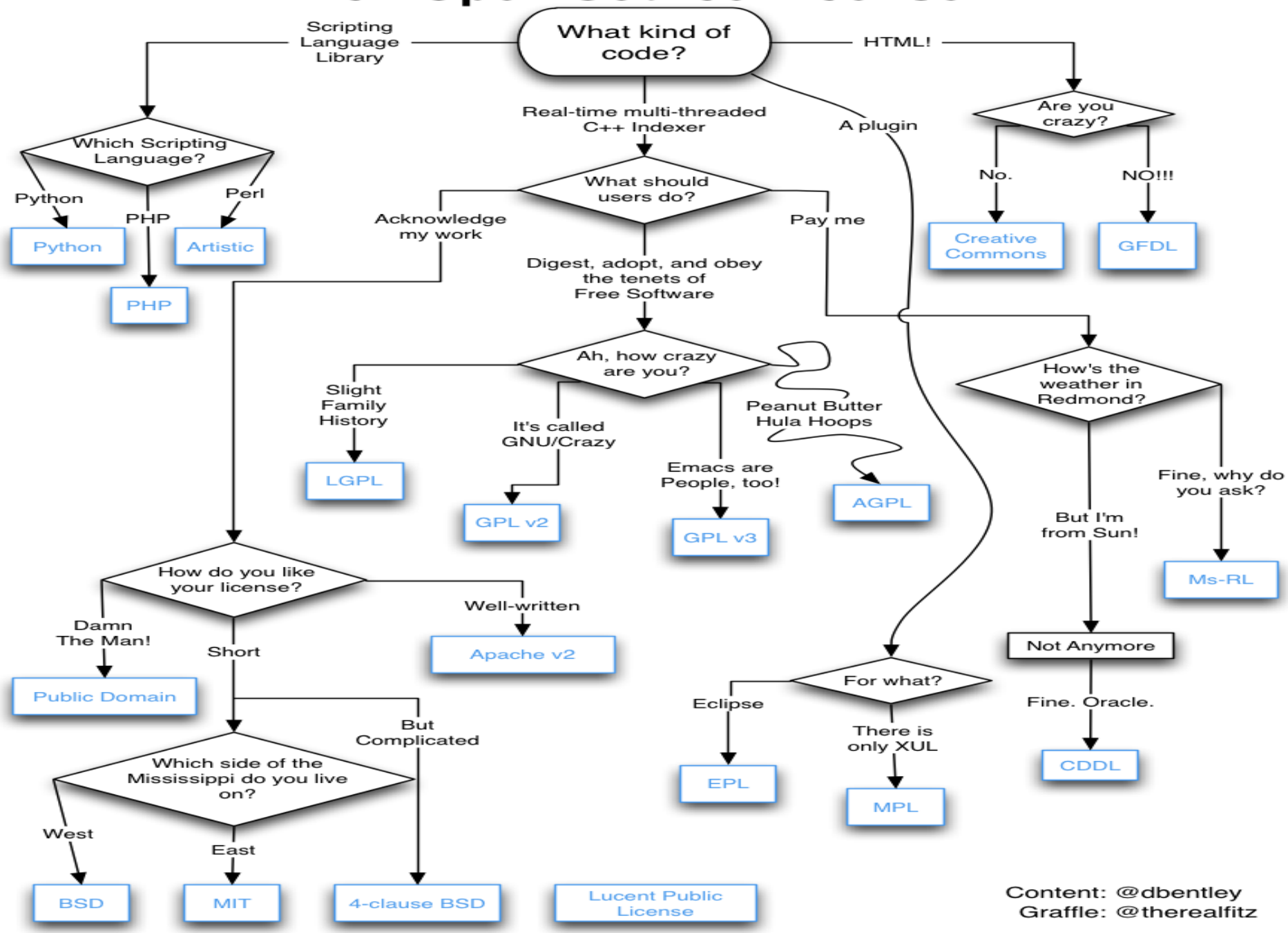




开源协议

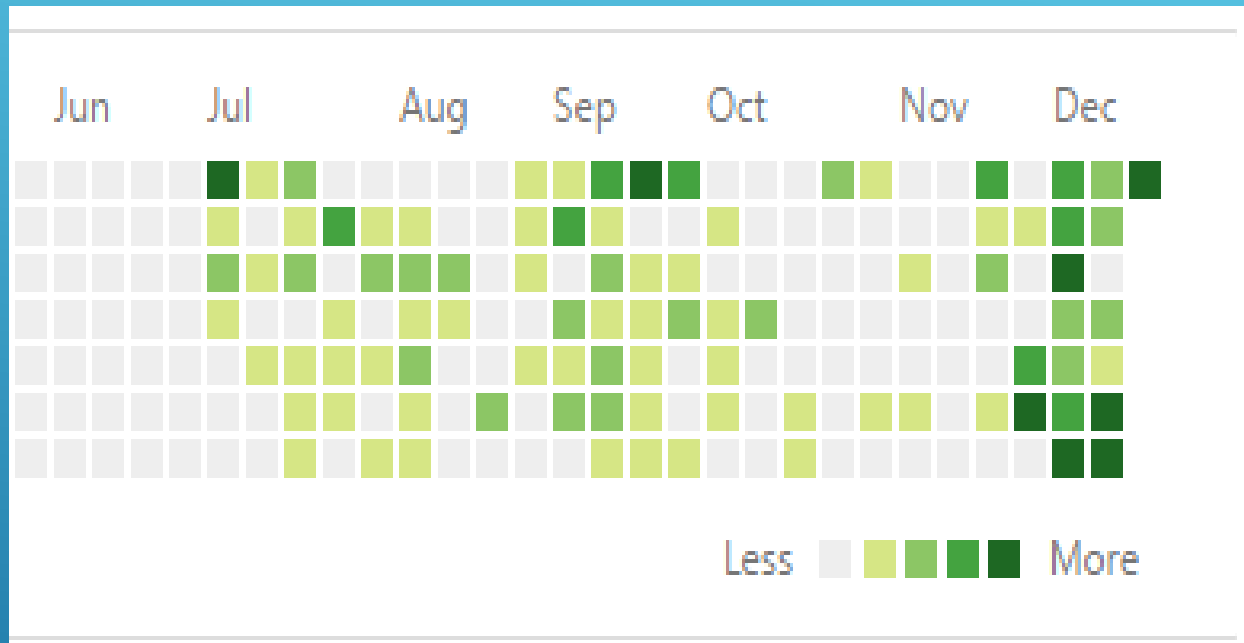

他人修改源码后，是否可以闭源？
No
Yes
新增代码是否采用同样许可证？
No
Yes
是否需要对源码的修改之处，提供说明文档？
No
Yes
每一个修改过的文件，是否都必须放置版权说明？
No
Yes
衍生软件的广告，是否可以用你的名字促销？
No
Yes
LGPL许可证
Mozilla许可证
GPL许可证
BSD许可证
MIT许可证
Apache许可证

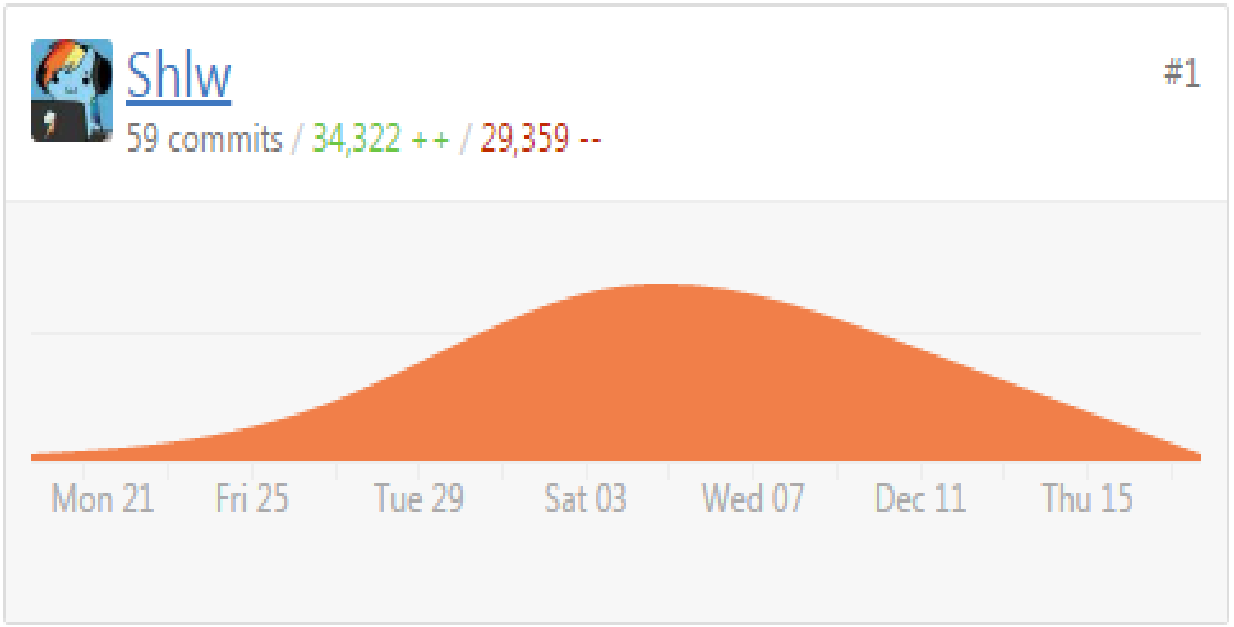
Which Open Source License?



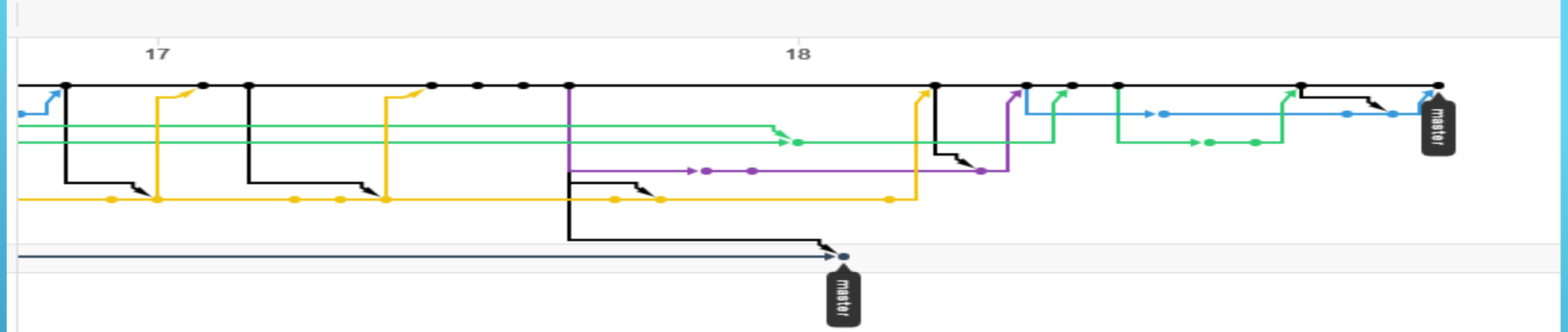
Content: @dbentley
Graffle: @therealfitz

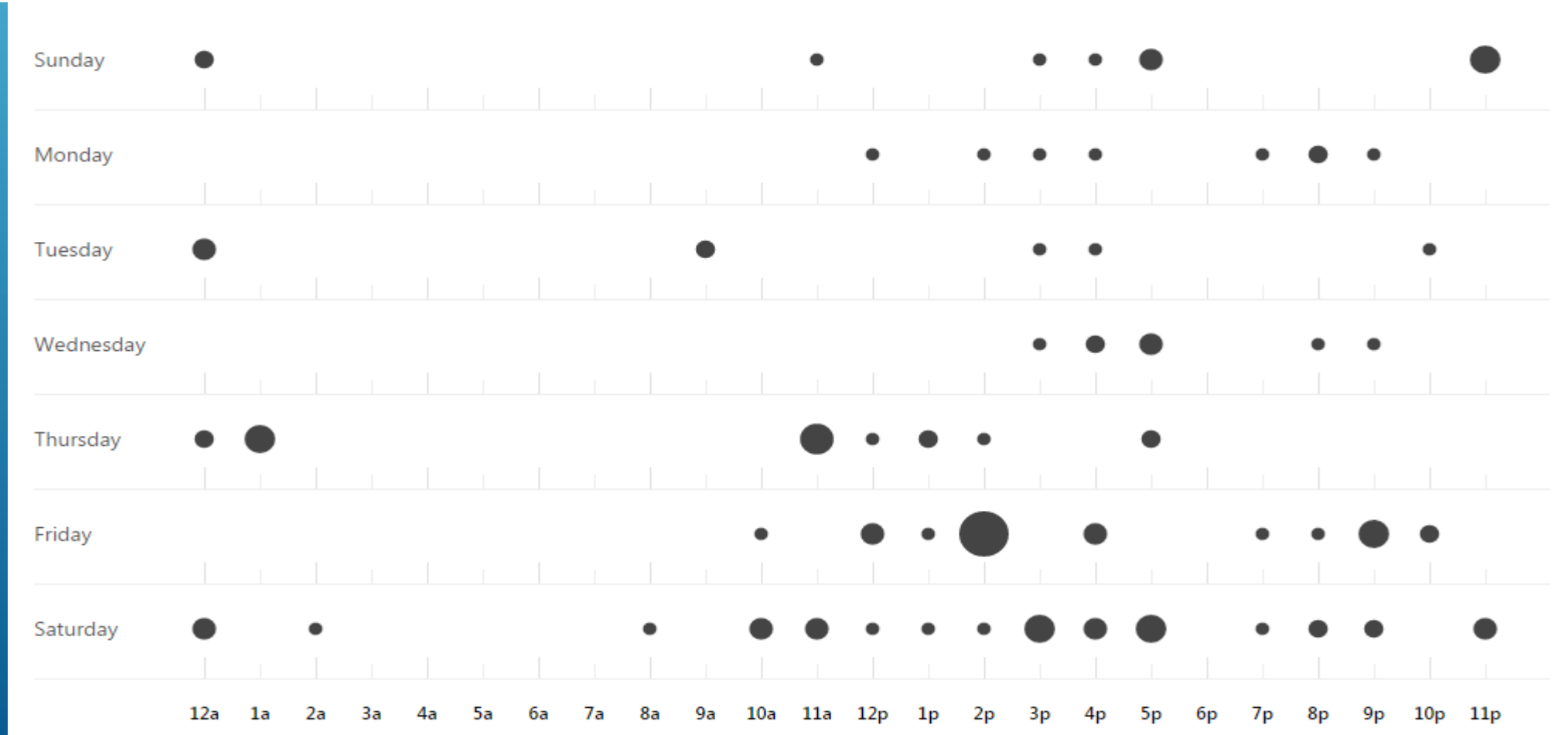
Github





https://github.com/Shlw/Stupid_OpenGL


17
18
master
master


Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
12a
1a
2a
3a
4a
5a
6a
7a
8a
9a
10a
11a
12p
1p
2p
3p
4p
5p
6p
7p
8p
9p
10p
11p

使用方法

1.网页版
2.Desktop
3.Git命令行

一些名词:

Pull Request
Repository
Fork
Release
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。

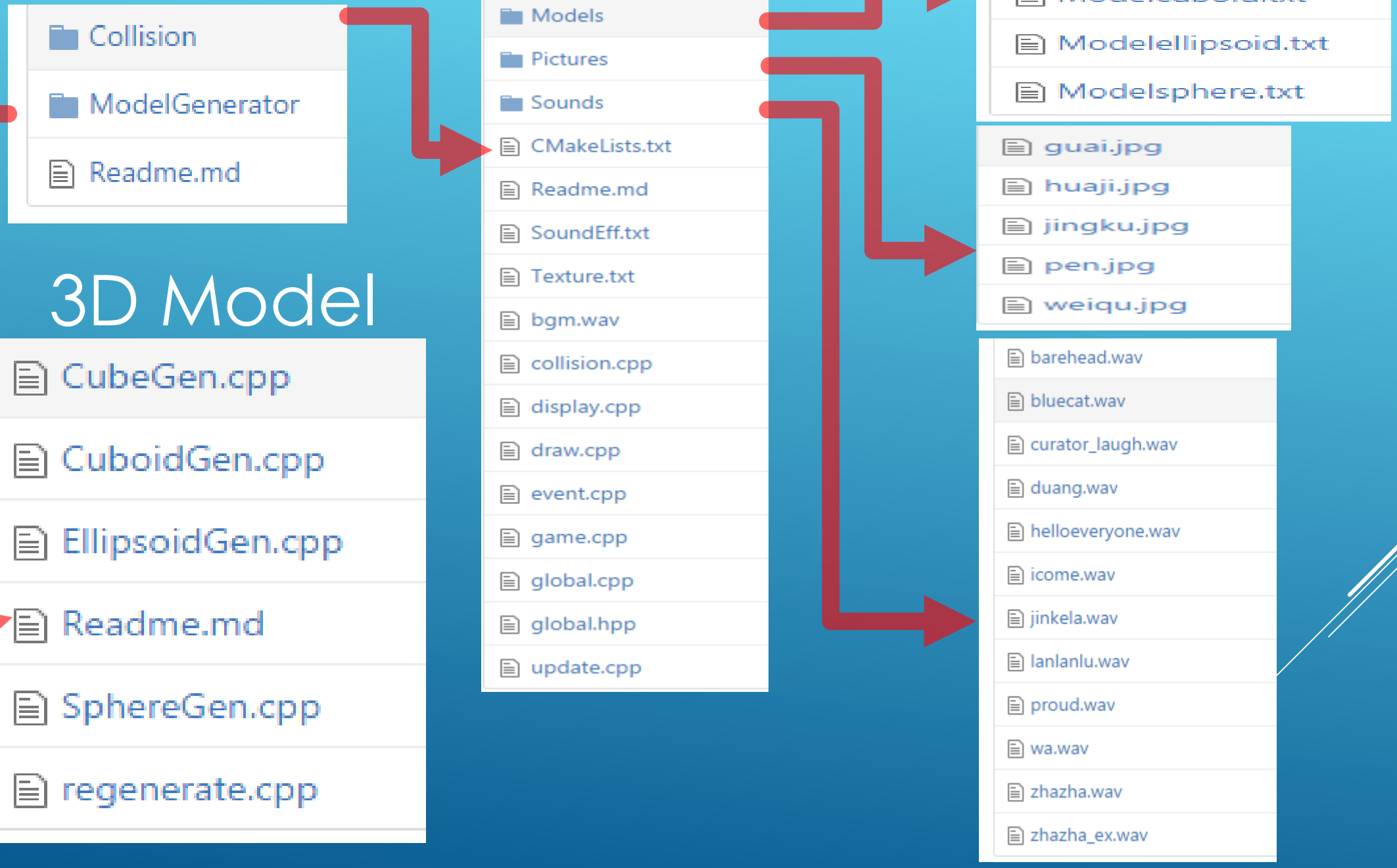
4.再用git push origin branch-name推送。

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☺



3D Model

CubeGen.cpp

CuboidGen.cpp

EllipsoidGen.cpp

Readme.md

SphereGen.cpp

regenerate.cpp

Class

```
// point class represents the still point in local coordinate system
class Point{
public:
    int nFlag;
    PVec4 vpCoordinate;
    PVec4 vpColor;
    PVec2 vpTexture;

    Point();
    Point(PPoint example);
    Point(
        float x,float y,float z,
        float r,float g,float b,float alpha
    );
    ~Point();

    void DrawVertex();
};
```

```
// triangle class represents the still triangle(cone) in local coordinate system
class Triangle{
public:
    PPoint pppVertex[3];
    PVec4 vpNormalVector;

    Triangle();
    Triangle(PPoint a,PPoint b,PPoint c);
    Triangle(PPoint a,PPoint b,PPoint c,PVec4 v);
    ~Triangle();

    void Draw();
};
```

```
// model class represents the still model in local coordinate system
class Model{
public:
    int nLength; // the number of triangular cones
    PTriangle* tppCone; // the point array of triangle pointers
    float fVolume,fElastic,fMass,fMaxRadius;
    glm::mat3* mMomentOfInertia;

    Model();
    ~Model();

    void Draw();
};
```

```
// object class contains
// the matrix(mFrame) to transform between local and global coordinate system
// and several indispensable descriptive aspect of the object
class Object{
public:
    int nModelType;
    PMat4 mpFrame;
    PVec3 vpSpeed;
    PVec3 vpAngularMomentum;
    float fMomentInertia;

    Object();
    Object(int model,float vx=0,float vy=0,float vz=0,float mx=0,float my=0,float mz=0);
    ~Object();

    PTriangle IsInside(PVec4 tp,PVec3 vdir=NULL);
    //ymw changed tp from PPoint to PVec4

    void Draw(int index);
    void Update();
};
```