CODE STYLE

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FILE

What this file includes.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef CLASS\_H

#define CLASS\_H

//Standard Library Related Stuff

#include <string>

#include <ctdargs>

//OGL Related Stuff

#include<OpenGL.h>

#include <Glew.h>

//Engine Related Stuff

#include “engine.h”

#include “utilities.h”

#define SOME\_METHOD(x) { Log << x; }

#define SAFE\_RELEASE(p) { if(p) { p->Release(); delete p; p = nullptr; } }

static const float PI = 3.145;

static const int MAX\_SIZE = 10;

bool g\_SomeFlag;

namespace BEngine

{

enum ECreateWindowResult

{

CREATE\_WINDOW\_RESULT\_SUCCEEDED,

CREATE\_WINDOW\_RESULT\_FAILED

};

enum class ECreateFileResult

{

STATUS\_OK,

STATUS\_FAIL

};

struct SSomeStruct

{

};

struct SSomeClasssConstrInfo {

};

//-------------------------

//SOME CLASS

//

//Description

//-----------------------

class CSomeClass

{

public:

CSomeClass(int val1, int val2){};

CSomeClass(int veryBigValueMemberName,

float veryHugeFloatValueName,

SSomeStruct& veryhugeStructName)

{};

virtual void update(float timeStep) = 0; //override

void addMethod(int val1, int val2);

protected:

void m\_method(string str);

private:

void m\_method(float r, float g, float b);

void m\_methodTwo();

bool m\_someVar;

Player\* m\_pPlayer;

};

namespace Graphics

{

}

}

#endif // CLASS\_H

CPP file

namespace BEngine

{

//-------------------------

//SOME CLASS METHODS START HERE

//-----------------------

void CSomeClass::addMethod(int val1, int val2)

{

return (val1 + val2);

}

}