MAKE
SCHOOL

C++

An Introduction

MAKE SCHOOL

C++ IS

Compiled

Object-oriented

Cross-platform

Fast

FILES

Header

.h extension

Defines a class'
interface

Source

.cpp extension

Defines a class'
implementation

INTERFACE

Describes what a class is.

What does it inherit from?

What are its public methods?

What are its public data members?

```
#ifndef __GameOfLife__Grid__
#define __GameOfLife__Grid__

#include "cocos2d.h"
#include "Creature.h"

class Grid : public cocos2d::Node
{
public:
    CREATE_FUNC(Grid);
    bool init() override;
    void onEnter() override;
    void evolveStep();
    int getGenerationCount();
    int getPopulationCount();

protected:
    int generationCount;
    int populationCount;
    float cellWidth;
    float cellHeight;
    cocos2d::Vector<Creature*> gridArray;

    void setupGrid();
    void setupTouchHandling();
    void updateNeighborCount();
    void updateCreatures();
    Creature* creatureForTouchLocation(cocos2d::Vec2 touchLocation);
    bool isValidIndex(int row, int col);
    int indexForRowColumn(int row, int col);
};

#endif /* defined(__GameOfLife__Grid__) */
```

header guard

class definition

public member functions

protected member variables

protected member functions

```
#ifndef __GameOfLife__Grid__
#define __GameOfLife__Grid__

#include "cocos2d.h"
#include "Creature.h"

class Grid : public cocos2d::Node
{
public:
    CREATE_FUNC(Grid);
    bool init() override;
    void onEnter() override;
    void evolveStep();
    int getGenerationCount();
    int getPopulationCount();

protected:
    int generationCount;
    int populationCount;
    float cellWidth;
    float cellHeight;
    cocos2d::Vector<Creature*> gridArray;

    void setupGrid();
    void setupTouchHandling();
    void updateNeighborCount();
    void updateCreatures();
    Creature* creatureForTouchLocation(cocos2d::Vec2 touchLocation);
    bool isValidIndex(int row, int col);
    int indexForRowColumn(int row, int col);
};

#endif /* defined(__GameOfLife__Grid__) */
```

header guard

Place all header code
between header guards

Header guards prevent the same
header from being included twice

```
#ifndef __GameOfLife__Grid__
#define __GameOfLife__Grid__

#include "cocos2d.h"
#include "Creature.h"

class Grid : public cocos2d::Node
{
public:
    CREATE_FUNC(Grid);
    bool init() override;
    void onEnter() override;
    void evolveStep();
    int getGenerationCount();
    int getPopulationCount();

protected:
    int generationCount;
    int populationCount;
    float cellWidth;
    float cellHeight;
    cocos2d::Vector<Creature*> gridArray;

    void setupGrid();
    void setupTouchHandling();
    void updateNeighborCount();
    void updateCreatures();
    Creature* creatureForTouchLocation(cocos2d::Vec2 touchLocation);
    bool isValidIndex(int row, int col);
    int indexForRowColumn(int row, int col);
};

#endif /* defined(__GameOfLife__Grid__) */
```

`class` definition

Defines class name

Defines what the class inherits from

```
#define __GameOfLife__Grid__

#include "cocos2d.h"
#include "Creature.h"

class Grid : public cocos2d::Node
{
public:
    CREATE_FUNC(Grid);
    bool init() override;
    void onEnter() override;
    void evolveStep();
    int getGenerationCount();
    int getPopulationCount();

protected:
    int generationCount;
    int populationCount;
    float cellWidth;
    float cellHeight;
    cocos2d::Vector<Creature*> gridArray;

    void setupGrid();
    void setupTouchHandling();
    void updateNeighborCount();
    void updateCreatures();
    Creature* creatureForTouchLocation(cocos2d::Vec2 touchLocation);
    bool isValidIndex(int row, int col);
    int indexForRowColumn(int row, int col);
};
```

public member functions

These are the functions that can be called by other classes to interact with this class

```
#ifndef __GameOfLife__Grid__
#define __GameOfLife__Grid__

#include "cocos2d.h"
#include "Creature.h"

class Grid : public cocos2d::Node
{
public:
    CREATE_FUNC(Grid);
    bool init() override;
    void onEnter() override;
    void evolveStep();
    int getGenerationCount();
    int getPopulationCount();

protected:
    int generationCount;
    int populationCount;
    float cellWidth;
    float cellHeight;
    cocos2d::Vector<Creature*> gridArray;

    void setupGrid();
    void setupTouchHandling();
    void updateNeighborCount();
    void updateCreatures();
    Creature* creatureForTouchLocation(cocos2d::Vec2 touchLocation);
    bool isValidIndex(int row, int col);
    int indexForRowColumn(int row, int col);
};

#endif /* defined(__GameOfLife__Grid__) */
```

These variables are only accessible by this class and subclasses

Other classes can not access them

protected member variables

```
#define __GameOfLife__Grid__

#include "cocos2d.h"
#include "Creature.h"

class Grid : public cocos2d::Node
{
public:
    CREATE_FUNC(Grid);
    bool init() override;
    void onEnter() override;
    void evolveStep();
    int getGenerationCount();
    int getPopulationCount();

protected:
    int generationCount;
    int populationCount;
    float cellWidth;
    float cellHeight;
    cocos2d::Vector<Creature*> gridArray;

    void setupGrid();
    void setupTouchHandling();
    void updateNeighborCount();
    void updateCreatures();
    Creature* creatureForTouchLocation(cocos2d::Vec2 touchLocation);
    bool isValidIndex(int row, int col);
    int indexForRowColumn(int row, int col);
};
```

These functions are only accessible by this class and subclasses

Other classes can not access them

protected member functions

`public`

Any class can call these functions or access these variables

`protected`

Only this class and subclasses can call these functions or access these variables

`private`

Only this class can call these functions or access these variables

IMPLEMENTATION

Describes how the class works.

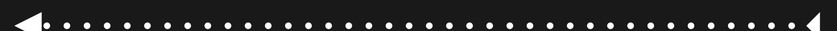
Contains implementations of member functions

Sets default values for member variables

```cpp
#include "Grid.h"

using namespace cocos2d;

bool Grid::init()
{
    if (! Node::init())
    {
        return false;
    }

    generationCount = 0;
    populationCount = 0;

    return true;
}

void Grid::onEnter()
{
    Node::onEnter();

    this->setupGrid();

    this->setupTouchHandling();
}
```

member function definitions

```
#include "Grid.h"

using namespace cocos2d;

bool Grid::init()
{
    if (! Node::init())
    {
        return false;
    }

    generationCount = 0;
    populationCount = 0;

    return true;
}

void Grid::onEnter()
{
    Node::onEnter();

    this->setupGrid();

    this->setupTouchHandling();
}
```



#include header

Must include own class header file

Ex. **Grid.cpp** must include **Grid.h**

MAKE SCHOOL

```cpp
#include "Grid.h"

using namespace cocos2d;

bool Grid::init()
{
    if (! Node::init())
    {
        return false;
    }

    generationCount = 0;
    populationCount = 0;

    return true;
}

void Grid::onEnter()
{
    Node::onEnter();

    this->setupGrid();

    this->setupTouchHandling();
}
```

namespace declarations

Any `using namepace` declarations go here

Do not put `using namespace` declarations in header files

MAKE SCHOOL

```cpp
#include "Grid.h"

using namespace cocos2d;

bool Grid::init()
{
    if (! Node::init())
    {
        return false;
    }

    generationCount = 0;
    populationCount = 0;

    return true;
}

void Grid::onEnter()
{
    Node::onEnter();

    this->setupGrid();

    this->setupTouchHandling();
}
```

Implementations for **public**, **private** and **protected** functions declared in header

member function definitions

C++ FUNCTION

```
bool Grid::isValidIndex(int row, int col)
```

class scope

return type

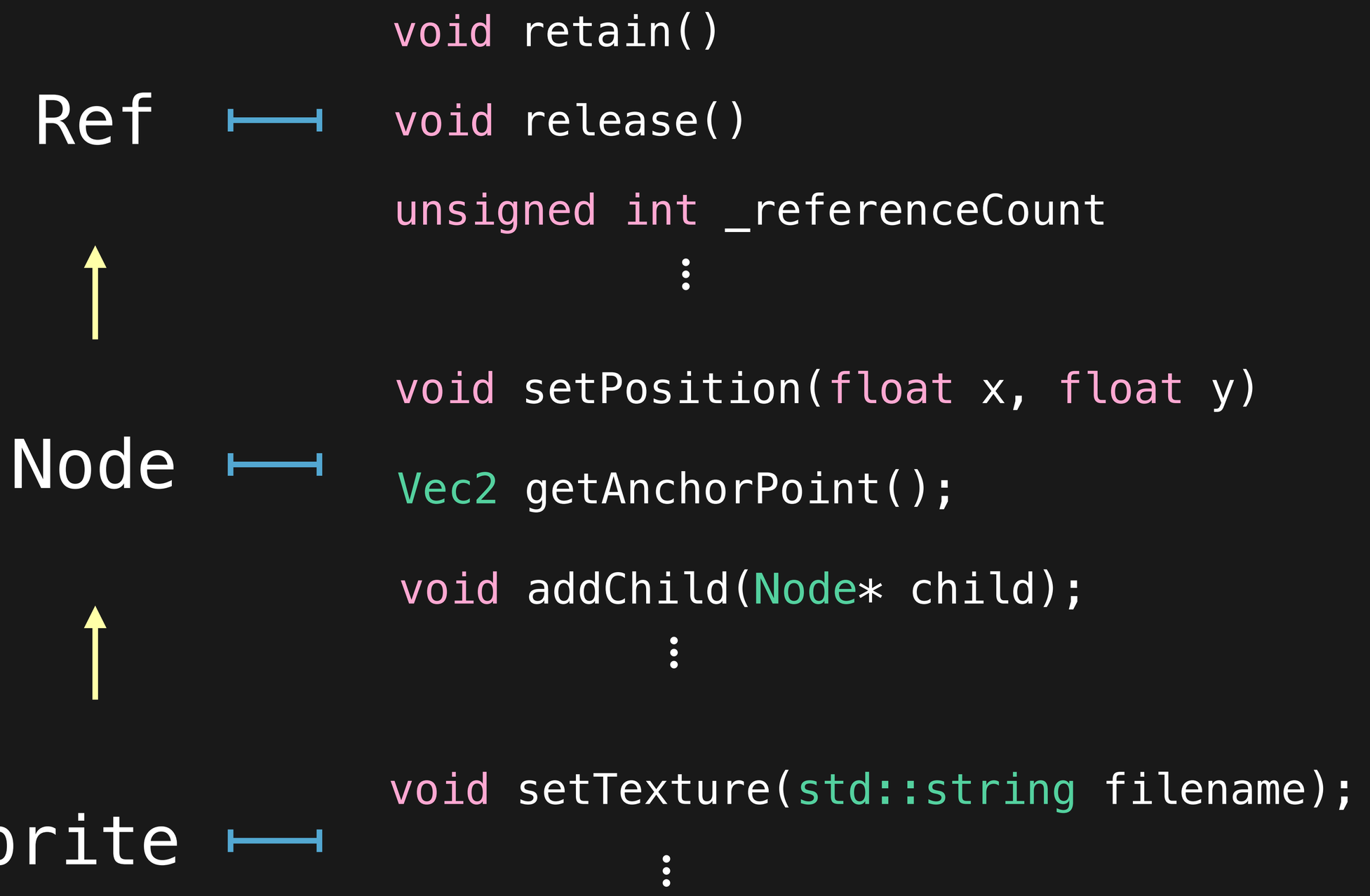
function name

parameters

C++ TYPES

	bits	range	example
`bool`	8	true, false	`true, false`
`char`	8	-127 to 127 0 to 255	`54, 'h'`
`float`	32	+/- 3.4e +/- 38 ~7 digit precision	`5.326f`
`double`	64	+/- 1.7e +/- 308 ~15 digit precision	`27.495`
`int`	32	-2,147,483,648 to 2,147,483,647	`34`

INHERITANCE


Ref
Node
Sprite
void retain()
void release()
unsigned int _referenceCount
void setPosition(float x, float y)
Vec2 getAnchorPoint();
void addChild(Node* child);
void setTexture(std::string filename);

ENCAPSULATION

`generationCount` and `populationCount` are *properties* of Grid

```
class Grid : public cocos2d::Node
{
public:
    int generationCount;
    int populationCount;
};
```

Why is this ***bad?***

MAKE SCHOOL

```
class Grid : public cocos2d::Node
{
public:
    int getGenerationCount();
   void setGenerationCount(int generationCount)

    int getPopulationCount();
   void setPopulationCount(int populationCount)

protected:
    int generationCount;
    int populationCount;

};
```

This is good because the implementation of **generationCount** and **populationCount** are hidden from other classes.

The value of **generationCount** and **populationCount** cannot be changed without **Grid** knowing about it.

MAKE
SCHOOL