

Snake

制作者 Wang Jiannan

Contents

1	贪吃蛇	1
2	类索引	3
2.1	类列表	3
3	文件索引	5
3.1	文件列表	5
4	类说明	7
4.1	MainWindow类 参考	7
4.1.1	构造及析构函数说明	8
4.1.1.1	MainWindow()	8
4.1.1.2	~MainWindow()	8
4.1.2	成员函数说明	8
4.1.2.1	endStatus	8
4.1.2.2	gameStatus()	9
4.1.2.3	keyPressEvent()	9
4.1.2.4	load	9
4.1.2.5	mousePressEvent()	9
4.1.2.6	newGame()	9
4.1.2.7	paintEvent()	9
4.1.2.8	pause	10
4.1.2.9	pauseStatus()	10
4.1.2.10	play	10
4.1.2.11	readyStatus()	10

4.1.2.12	restart	10
4.1.2.13	resume	10
4.1.2.14	savefile	11
4.1.2.15	timerEvent()	11
4.1.3	类成员变量说明	11
4.1.3.1	painter	11
4.1.3.2	snake	11
4.1.3.3	status	11
4.1.3.4	ui	11
4.2	Painter类 参考	12
4.2.1	构造及析构造函数说明	12
4.2.1.1	Painter()	12
4.2.2	成员函数说明	12
4.2.2.1	background()	12
4.2.2.2	game()	12
4.2.3	类成员变量说明	13
4.2.3.1	base	13
4.3	Snake类 参考	13
4.3.1	构造及析构造函数说明	14
4.3.1.1	Snake()	14
4.3.1.2	~Snake()	14
4.3.2	成员函数说明	14
4.3.2.1	death	14
4.3.2.2	find() [1/2]	15
4.3.2.3	find() [2/2]	15
4.3.2.4	genFood()	15
4.3.2.5	getLen()	15
4.3.2.6	grow()	15
4.3.2.7	initGame()	16
4.3.2.8	initMap()	16
4.3.2.9	move()	16
4.3.2.10	operator()()	16
4.3.2.11	readFile()	17
4.3.2.12	refreshTimer	17
4.3.2.13	reverse()	17
4.3.2.14	saveFile()	17
4.3.3	类成员变量说明	17
4.3.3.1	cnt	18
4.3.3.2	length	18
4.3.3.3	map	18
4.3.3.4	res	18

5 文件说明	19
5.1 Snake/main.cpp 文件参考	19
5.2 Snake/mainwindow.cpp 文件参考	19
5.3 Snake/mainwindow.h 文件参考	19
5.3.1 详细描述	20
5.4 Snake/painter.cpp 文件参考	20
5.5 Snake/painter.h 文件参考	20
5.5.1 详细描述	20
5.6 Snake/snake.cpp 文件参考	21
5.6.1 函数说明	21
5.6.1.1 random() [1/2]	21
5.6.1.2 random() [2/2]	21
5.7 Snake/snake.h 文件参考	21
5.7.1 详细描述	22
索引	23

Chapter 1

贪吃蛇

Project	贪吃蛇
Author	Wang Jiannan

Chapter 2

类索引

2.1 类列表

这里列出了所有类、结构、联合以及接口定义等，并附带简要说明:

MainWindow	7
Painter	12
Snake	13

Chapter 3

文件索引

3.1 文件列表

这里列出了所有文件，并附带简要说明：

Snake/main.cpp	
项目主函数文件	19
Snake/mainwindow.cpp	19
Snake/mainwindow.h	
主要管理整个游戏的逻辑	19
Snake/painter.cpp	20
Snake/painter.h	
负责绘制游戏地图	20
Snake/snake.cpp	21
Snake/snake.h	
负责蛇在游戏地图上的活动	21

Chapter 4

类说明

4.1 MainWindow类 参考

`#include <mainwindow.h>`

类 `MainWindow` 继承关系图:

`MainWindow` 的协作图:

Public 成员函数

- `MainWindow (QWidget *parent=nullptr)`
初始化
- `~MainWindow ()`

Protected 成员函数

- `void keyPressEvent (QKeyEvent *ev)`
处理键盘事件
- `void paintEvent (QPaintEvent *ev)`
处理绘制事件
- `void mousePressEvent (QMouseEvent *ev)`
处理鼠标事件
- `void timerEvent (QTimerEvent *ev)`
处理时间事件

Private 槽

- `void pause ()`
暂停事件
- `void play ()`
开始游戏事件
- `void load ()`
加载事件
- `void restart ()`
重新游戏事件
- `void resume ()`
继续游戏事件
- `void savefile ()`
保存游戏事件
- `void endStatus ()`
终止状态

Private 成员函数

- void `readyStatus` ()
未开始状态
- void `gameStatus` ()
游戏状态
- void `pauseStatus` ()
暂停状态
- void `newGame` ()
开始一局新游戏

Private 属性

- `Ui::MainWindow` * `ui`
- `Painter` * `painter`
- `Snake` * `snake`
- int `status`

4.1.1 构造及析构函数说明

4.1.1.1 MainWindow()

```
MainWindow::MainWindow (
    QWidget * parent = nullptr )
```

初始化

各变量初始化，信号与槽的 connect

4.1.1.2 ~MainWindow()

```
MainWindow::~MainWindow ( )
```

4.1.2 成员函数说明

4.1.2.1 endStatus

```
void MainWindow::endStatus ( ) [private], [slot]
```

终止状态

4.1.2.2 gameStatus()

```
void MainWindow::gameStatus ( ) [private]
```

游戏状态

4.1.2.3 keyPressEvent()

```
void MainWindow::keyPressEvent (
    QKeyEvent * ev ) [protected]
```

处理键盘事件

处理方向键，控制蛇的移动

4.1.2.4 load

```
void MainWindow::load ( ) [private], [slot]
```

加载事件

调用 [Snake](#) 中 `readFile` 函数

4.1.2.5 mousePressEvent()

```
void MainWindow::mousePressEvent (
    QMouseEvent * ev ) [protected]
```

处理鼠标事件

在游戏地图内点击后，调用 [Snake](#) 中 `reverse` 函数

4.1.2.6 newGame()

```
void MainWindow::newGame ( ) [private]
```

开始一局新游戏

[Snake](#) 初始化，重新绘制游戏地图

4.1.2.7 paintEvent()

```
void MainWindow::paintEvent (
    QPaintEvent * ev ) [protected]
```

处理绘制事件

调用 [Painter](#) 中函数

4.1.2.8 pause

```
void MainWindow::pause ( ) [private], [slot]
```

暂停事件

调用 [Snake](#) 中 `readFile` 函数

4.1.2.9 pauseStatus()

```
void MainWindow::pauseStatus ( ) [private]
```

暂停状态

4.1.2.10 play

```
void MainWindow::play ( ) [private], [slot]
```

开始游戏事件

4.1.2.11 readyStatus()

```
void MainWindow::readyStatus ( ) [private]
```

未开始状态

4.1.2.12 restart

```
void MainWindow::restart ( ) [private], [slot]
```

重新游戏事件

4.1.2.13 resume

```
void MainWindow::resume ( ) [private], [slot]
```

继续游戏事件

4.1.2.14 savefile

```
void MainWindow::savefile ( ) [private], [slot]
```

保存游戏事件

调用 [Snake](#) 中 `saveFile` 函数

4.1.2.15 timerEvent()

```
void MainWindow::timerEvent (
    QTimerEvent * ev ) [protected]
```

处理时间事件

4.1.3 类成员变量说明

4.1.3.1 painter

```
Painter* MainWindow::painter [private]
```

4.1.3.2 snake

```
Snake* MainWindow::snake [private]
```

4.1.3.3 status

```
int MainWindow::status [private]
```

4.1.3.4 ui

```
Ui::MainWindow* MainWindow::ui [private]
```

该类的文档由以下文件生成:

- [Snake/mainwindow.h](#)
- [Snake/mainwindow.cpp](#)

4.2 Painter类 参考

```
#include <painter.h>
```

Public 成员函数

- `Painter ()`
- `void background (QPainter *p)`
绘制游戏地图背景
- `void game (QPainter *p, Snake *s)`
绘制游戏地图中当前游戏局面

静态 Public 属性

- `static const int base = 10`

4.2.1 构造及析构函数说明

4.2.1.1 Painter()

```
Painter::Painter ( )
```

4.2.2 成员函数说明

4.2.2.1 background()

```
void Painter::background (  
    QPainter * p )
```

绘制游戏地图背景

4.2.2.2 game()

```
void Painter::game (  
    QPainter * p,  
    Snake * s )
```

绘制游戏地图中当前游戏局面

绘制蛇、障碍、食物

4.2.3 类成员变量说明

4.2.3.1 base

```
const int Painter::base = 10 [static]
```

该类的文档由以下文件生成:

- Snake/painter.h
- Snake/painter.cpp

4.3 Snake类 参考

```
#include <snake.h>
```

类 Snake 继承关系图:

Snake 的协作图:

信号

- void `death` ()
蛇死亡, 进入终止状态
- void `refreshTimer` (int)
更新计时器

Public 成员函数

- `Snake` ()
- `~Snake` ()
- void `genFood` ()
随机生成食物
- void `initMap` ()
初始化地图
- void `reverse` (const int &, const int &)
指定位置进行反转 实现非障碍物与障碍物的转化
- void `initGame` ()
初始化游戏
- int `getLen` ()
获取蛇的长度
- int `operator()` (const int &x, const int &y)
获取 *map* 中的值
- void `grow` (const int &x, const int &y)
蛇向格点 (x,y) 伸长 若 *res* 为零, 则将蛇所在格点值全部减一
- void `move` (int dir)
操控蛇向四周移动
- void `saveFile` (QString filename)
保存游戏状态
- void `readFile` (QString filename)
读取游戏状态

Private 成员函数

- void **find** (int &x, int &y, int d)
在 *map* 内寻找一个指定大小元素
- std::vector< int * > **find** (int d, int lim)
在 *map* 内寻找所有大小元素 *lim* 与边界距离

Private 属性

- int **length**
蛇的长度
- int **res**
剩余增长长度
- int **cnt**
当前时间
- int **map** [40][40]
游戏地图

4.3.1 构造及析构函数说明

4.3.1.1 Snake()

```
Snake::Snake ( )
```

4.3.1.2 ~Snake()

```
Snake::~~Snake ( )
```

4.3.2 成员函数说明

4.3.2.1 death

```
void Snake::death ( ) [signal]
```

蛇死亡，进入终止状态

4.3.2.2 find() [1/2]

```
void Snake::find (
    int & x,
    int & y,
    int d ) [private]
```

在 map 内寻找一个指定大小元素

4.3.2.3 find() [2/2]

```
std::vector< int * > Snake::find (
    int d,
    int lim = 0 ) [private]
```

在 map 内寻找所有大小元素 lim 与边界距离

4.3.2.4 genFood()

```
void Snake::genFood ( )
```

随机生成食物

4.3.2.5 getLen()

```
int Snake::getLen ( )
```

获取蛇的长度

4.3.2.6 grow()

```
void Snake::grow (
    const int & x,
    const int & y )
```

蛇向格点 (x,y) 伸长 若 res 为零，则将蛇所在格点值全部减一

4.3.2.7 initGame()

```
void Snake::initGame ( )
```

初始化游戏

4.3.2.8 initMap()

```
void Snake::initMap ( )
```

初始化地图

4.3.2.9 move()

```
void Snake::move (
    int dir )
```

操控蛇向四周移动

参数

<i>dir</i>	移动方向，若非 0~3 则为默认移动方向
	<ul style="list-style-type: none">• 0 左• 1 上• 2 右• 3 下

4.3.2.10 operator>()

```
int Snake::operator() (
    const int & x,
    const int & y )
```

获取 map 中的值

返回

返回格点类型

返回值

-2	障碍
-1	食物
0	空白点
>0	蛇，其中蛇头值与 <code>length</code> 相等

4.3.2.11 readFile()

```
void Snake::readFile (
    QString filename )
```

读取游戏状态

4.3.2.12 refreshTimer

```
void Snake::refreshTimer (
    int ) [signal]
```

更新计时器

4.3.2.13 reverse()

```
void Snake::reverse (
    const int & x,
    const int & y )
```

指定位置进行反转 实现非障碍物与障碍物的转化

4.3.2.14 saveFile()

```
void Snake::saveFile (
    QString filename )
```

保存游戏状态

4.3.3 类成员变量说明

4.3.3.1 cnt

```
int Snake::cnt [private]
```

当前时间

4.3.3.2 length

```
int Snake::length [private]
```

蛇的长度

4.3.3.3 map

```
int Snake::map[40][40] [private]
```

游戏地图

4.3.3.4 res

```
int Snake::res [private]
```

剩余增长长度

该类的文档由以下文件生成:

- Snake/[snake.h](#)
- Snake/[snake.cpp](#)

Chapter 5

文件说明

5.1 Snake/main.cpp 文件参考

项目主函数文件

```
#include "mainwindow.h"
#include <QApplication>
#include <QPushButton>
#include <QTimer>
#include <QDebug>
```

main.cpp 的引用(Include)关系图:

5.2 Snake/mainwindow.cpp 文件参考

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
mainwindow.cpp 的引用(Include)关系图:
```

5.3 Snake/mainwindow.h 文件参考

主要管理整个游戏的逻辑

```
#include <QMainWindow>
#include <QObject>
#include <QKeyEvent>
#include <QPainter>
#include <QToolBar>
#include "painter.h"
#include "snake.h"
```

mainwindow.h 的引用(Include)关系图: 此图展示该文件直接或间接的被哪些文件引用了:

类

- class `MainWindow`

命名空间

- [Ui](#)

5.3.1 详细描述

主要管理整个游戏的逻辑

作者

Wang Jiannan

日期

2020-8-24

5.4 Snake/painter.cpp 文件参考

`#include "painter.h"`
painter.cpp 的引用(Include)关系图:

5.5 Snake/painter.h 文件参考

负责绘制游戏地图

`#include <QWidget>`
`#include <QPainter>`
`#include "snake.h"`
painter.h 的引用(Include)关系图: 此图展示该文件直接或间接的被哪些文件引用了:

类

- class [Painter](#)

5.5.1 详细描述

负责绘制游戏地图

作者

Wang Jiannan

日期

2020-8-24

5.6 Snake/snake.cpp 文件参考

```
#include "snake.h"
#include <QDebug>
snake.cpp 的引用(Include)关系图:
```

函数

- int [random](#) (int a, int b)
- int [random](#) (int a)

5.6.1 函数说明

5.6.1.1 random() [1/2]

```
int random (
    int a,
    int b )
```

5.6.1.2 random() [2/2]

```
int random (
    int a )
```

5.7 Snake/snake.h 文件参考

负责蛇在游戏地图上的活动

```
#include <cstdlib>
#include <ctime>
#include <vector>
#include <QObject>
#include <algorithm>
#include <QString>
#include <QFile>
#include <QDataStream>
snake.h 的引用(Include)关系图: 此图展示该文件直接或间接的被哪些文件引用了:
```

类

- class [Snake](#)

5.7.1 详细描述

负责蛇在游戏地图上的活动

负责蛇的移动、成长、吃食物、判断是否蛇死亡，游戏开始蛇的随机生成、每次食物的随机生成

作者

Wang Jiannan

日期

2020-8-24

Index

- ~MainWindow
 - MainWindow, 8
- ~Snake
 - Snake, 14
- background
 - Painter, 12
- base
 - Painter, 13
- cnt
 - Snake, 17
- death
 - Snake, 14
- endStatus
 - MainWindow, 8
- find
 - Snake, 14, 15
- game
 - Painter, 12
- gameStatus
 - MainWindow, 8
- genFood
 - Snake, 15
- getLen
 - Snake, 15
- grow
 - Snake, 15
- initGame
 - Snake, 15
- initMap
 - Snake, 16
- keyPressEvent
 - MainWindow, 9
- length
 - Snake, 18
- load
 - MainWindow, 9
- MainWindow, 7
 - ~MainWindow, 8
 - endStatus, 8
 - gameStatus, 8
 - keyPressEvent, 9
 - load, 9
 - MainWindow, 8
 - mousePressEvent, 9
 - newGame, 9
 - paintEvent, 9
 - painter, 11
 - pause, 9
 - pauseStatus, 10
 - play, 10
 - readyStatus, 10
 - restart, 10
 - resume, 10
 - savefile, 10
 - snake, 11
 - status, 11
 - timerEvent, 11
 - ui, 11
- map
 - Snake, 18
- mousePressEvent
 - MainWindow, 9
- move
 - Snake, 16
- newGame
 - MainWindow, 9
- operator()
 - Snake, 16
- paintEvent
 - MainWindow, 9
- Painter, 12
 - background, 12
 - base, 13
 - game, 12
 - Painter, 12
- painter
 - MainWindow, 11
- pause
 - MainWindow, 9
- pauseStatus
 - MainWindow, 10
- play
 - MainWindow, 10
- random
 - snake.cpp, 21
- readFile
 - Snake, 17

- readyStatus
 - MainWindow, 10
- refreshTimer
 - Snake, 17
- res
 - Snake, 18
- restart
 - MainWindow, 10
- resume
 - MainWindow, 10
- reverse
 - Snake, 17
- saveFile
 - Snake, 17
- savefile
 - MainWindow, 10
- Snake, 13
 - ~Snake, 14
 - cnt, 17
 - death, 14
 - find, 14, 15
 - genFood, 15
 - getLen, 15
 - grow, 15
 - initGame, 15
 - initMap, 16
 - length, 18
 - map, 18
 - move, 16
 - operator(), 16
 - readFile, 17
 - refreshTimer, 17
 - res, 18
 - reverse, 17
 - saveFile, 17
 - Snake, 14
- snake
 - MainWindow, 11
- snake.cpp
 - random, 21
- Snake/main.cpp, 19
- Snake/mainwindow.cpp, 19
- Snake/mainwindow.h, 19
- Snake/painter.cpp, 20
- Snake/painter.h, 20
- Snake/snake.cpp, 21
- Snake/snake.h, 21
- status
 - MainWindow, 11
- timerEvent
 - MainWindow, 11
- ui
 - MainWindow, 11