**Survival Client Software Develop**

[1. Revision History 2](#_Toc25166)

[2. Introduction 2](#_Toc20708)

[3. Software Framework 3](#_Toc8481)

[4. Protocol layer 4](#_Toc6918)

[5. Policy layer 5](#_Toc9447)

[6. Project & Build 6](#_Toc25046)

[7. Program arguments 7](#_Toc2789)

[8. AutoTest 10](#_Toc10951)

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Author | Description |
| V1.0 | 2020/11/9 | 万志朋 | Initial version |
| V1.0.1 | 2020/11/10 | 万志朋 | typography |

## Introduction

代码赌场预赛题目叫做《Survival》，讲述的是8个吃豆人在充斥着幽灵的果园里靠吃水果和发射炮弹来保护自己，尽量生存的故事，《Survival》是一个多玩家同时在线的游戏，参赛队员实现一个客户端来控制游戏中的角色，以吃水果来积分，同时可发射炮弹来伤害其它游戏角色，包括幽灵和其它玩家控制的游戏角色。



本文档描述客户端设计与实现。

## Software Framework

**Policy**

**Protocol**

* **Protocol layer**

实现与服务端交互，完成连接、心跳、地图等数据获取与解析等交互功能；

* **Policy layer**

实现生存与积分策略，对地图分析，向Protocol层设置下一步（方向与是否开炮）。

## 

## Protocol layer

设计***server\_obj***结构体对象，用于描述服务端所有信息，包括socket、key、地图、位置、积分与关键字等。详细信息请查看头文件（survival\_protocol/survival\_protocol.h）。

主目录：survival\_protocol

server\_obj对象操作函数说明：

/\* 初始化server\_obj对象 \*/

*int server\_obj\_init(struct server\_obj \*s\_obj);*

/\* 与服务端建立连接与key认证 \*/

*int server\_connect(struct server\_obj \*s\_obj);*

/\* 关闭socket \*/

*int server\_close(struct server\_obj \*s\_obj);*

/\* 与服务端交互 \*/

*int server\_interactive(struct server\_obj \*s\_obj);*

/\* 用于policy层获取地图数据 \*/

*struct map\_obj \*get\_map\_obj(struct server\_obj \*s\_obj);*

/\* 用于policy层设置下一步action \*/

*int set\_action(struct server\_obj \*s\_obj, char \*action);*

## 

## Policy layer

设计***map\_obj***结构体对象，用于从Protocol层获取地图等信息，如下描述，

struct map\_obj {

const char (\*map)[MAP\_X\_MAX][MAP\_Y\_MAX];

/\* map size \*/

int map\_size\_h;

int map\_size\_v;

int player\_me;

int player\_count;

struct coord location[PLAYER\_COUNT\_MAX];

int score[PLAYER\_COUNT\_MAX];

};

头文件描述（common.h）。

主目录：survival\_policy

Policy入口：

client.cpp

extern int policy\_do\_wzp(struct map\_obj \*, char \*);

int (\*policy\_do)(struct map\_obj \*, char \*) = policy\_do\_wzp;

policy\_do赋值新函数更换策略。

策略函数示例：

int policy\_do\_wzp(struct map\_obj \*m\_obj, char \*action)

｛

action[0] = DOWN; /\* downward \*/

action[1] = LAUNCH\_NO; /\* Don't fire \*/

return 0;

｝

## Project & Build

采用跨平台CMake构建工具（自动生成Makefile文件）。

目录说明：

* build

在build目录中执行（source build.sh）命令实现构建。

* survival\_protocol

protocol层主目录。

* survival\_policy

policy层主目录。

* main.c

client入口文件

* client.c

client主源文件

* common.h

公共主头文件

* test

GoogleTest测试框架目录

## Program arguments

./client --ip <ip address> --port <port> [--logfile log.txt --showmap]

参数ip/port是必选项，用于连接服务端。

logfile/showmap可选项，用于将log导入到指定文件，与显示终端简易地图。

（./client --ip 172.17.0.2 --port 9527）示例

<i>ip: 172.17.0.2.

<i>port: 9527.

<i>send data >>(15ba4a49c8234c93af0b76ec7074e61>>

<i>recv data <<[OK]<<

<i>key verify success.

<i>connect server.

<i>heart beat...

<i>send data >>(H)>>

<i>recv data <<[OK]<<

<i>heartbeat success.

<i>heart beat...

<i>send data >>(H)>>

<i>recv data <<[OK]<<

<i>heartbeat success.

<i>heart beat...

<i>send data >>(H)>>

<i>recv data <<[START 4 25]<<

<i>index: 4, map size: 25 x 25.

<i>send data >>(READY)>>

<i>game start.

<i>gaming...

<i>recv data <<[MAP e7 2511524a235514w332341115<<

<i>token is {e7}.

<i>location: 350 86 189 14 [7] 279 597.

<i>score: 0 0 0 0 [0] 0 0.

<i>map:vvvvvvvv

2542512a352445d5343512222

2923594492412111151411394

5235231591594552153141313

2134244549412939131222914

45331G1234592243431292333

1292454919222543153521334

3935951193913149151144995

5925453599423394195245443

2413359134914311412542353

5151224513419142533239415

2223193153324325435431242

19523353539a5953944515224

1135499359442154149549231

4344412223453152192924233

2333191w212529a9954393521

5493922195412523945549133

1992534922415334459444343

1345321145495115511329951

5351529213515231423593131

4994339322143212255355222

2954111529553259545515945

3424253524444935513235124

4923199291591135545429123

3424w51434419133245294s34

2919525144453114132232111

<i>map:^^^^^^^^

<i>send data >>(e7d )>>

<i>recv data <<[OK]<<

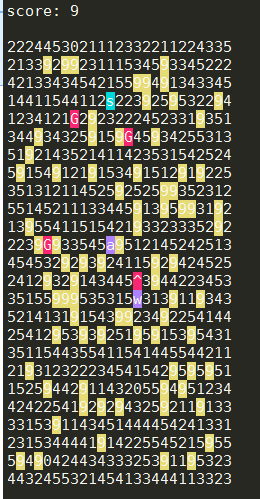
<i>me(4/7): L = 7, S = 0, next = {d }.

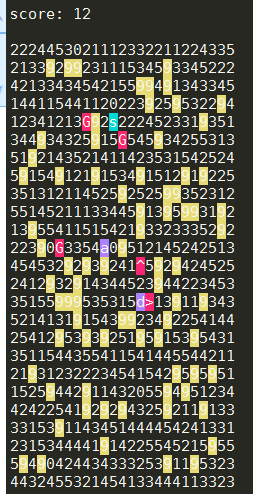
<i>My info: 0 [7,0].

<i>update action to {s }.

<i>gaming...

（./client --ip 172.17.0.2 --port 9527 --logfile log.txt --showmap）示例





## AutoTest

采用GoogleTest框架实现，待开发。