

# TCS215 オペレーティングシステム演習

## 作成プログラムの機能の概要

このプログラムは山梨大学コンピュータ理工学科 オペレーティングシステム演習にて作成したネットワーク対戦型ゲームプログラム（以下「本プログラム」と記す）である。

本プログラムは授業内にて提示された以下の仕様を満たす(表 1)。また、これらの仕様に対応する授業評価点も表中に示す。

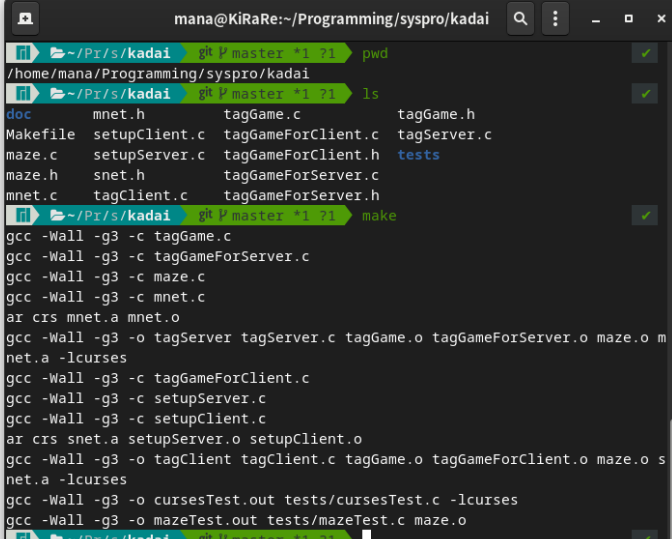
表 1: 本プログラムが満たす仕様と授業評価点

本プログラムが満たす仕様	仕様に対する授業評価点
ネットワーク対戦型ゲーム	50 点
障害物あり追跡ゲーム	60 点
CURSES マルチウィンドウ	+15 点
シューティング機構	+15 点
多人数接続	+α 点
自動生成迷路	+α 点

## 操作方法

## コンパイル

本プログラムでは make コマンドを使用しプログラムをコンパイルする。(図 1)



```
mana@KiRaRe:~/Programming/syspro/kadai
~/Pr/s/kadai $ pwd
/home/mana/Programming/syspro/kadai
~/Pr/s/kadai $ ls
doc      mnet.h      tagGame.c      tagGame.h
Makefile setupClient.c tagGameForClient.c tagServer.c
maze.c   setupServer.c tagGameForClient.h tests
maze.h   snet.h       tagGameForServer.c
mnet.c   tagClient.c  tagGameForServer.h
~/Pr/s/kadai $ make
gcc -Wall -g3 -c tagGame.c
gcc -Wall -g3 -c tagGameForServer.c
gcc -Wall -g3 -c maze.c
gcc -Wall -g3 -c mnet.c
ar crs mnet.a mnet.o
gcc -Wall -g3 -o tagServer tagServer.c tagGame.o tagGameForServer.o maze.o mnet.a -lcurses
gcc -Wall -g3 -c tagGameForClient.c
gcc -Wall -g3 -c setupServer.c
gcc -Wall -g3 -c setupClient.c
ar crs snet.a setupServer.o setupClient.o
gcc -Wall -g3 -o tagClient tagClient.c tagGame.o tagGameForClient.o maze.o snet.a -lcurses
gcc -Wall -g3 -o cursesTest.out tests/cursesTest.c -lcurses
gcc -Wall -g3 -o mazeTest.out tests/mazeTest.c maze.o
```

図 1 make コマンドを用いてコンパイルしたときの様子

## 実行

コンパイルして得られた実行ファイルはターミナル上にて動かす。サーバー側の実行ファイルである”tagServer”を実行してからクライアント側の実行ファイルである”tagClient”を実行する。(図 2)

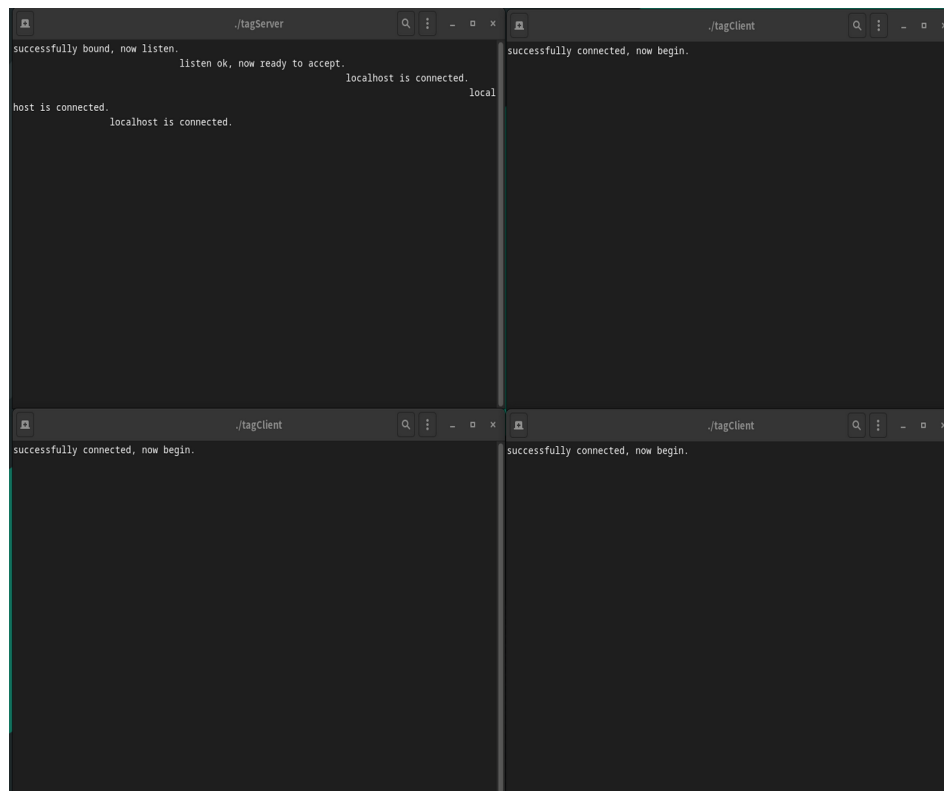


図 2 tagServer(左上)と tagClient(左上以外)を実行したときの様子

任意のタイミングでサーバー側のターミナルに任意のキーを入力するとゲームを開始する。(図 3)

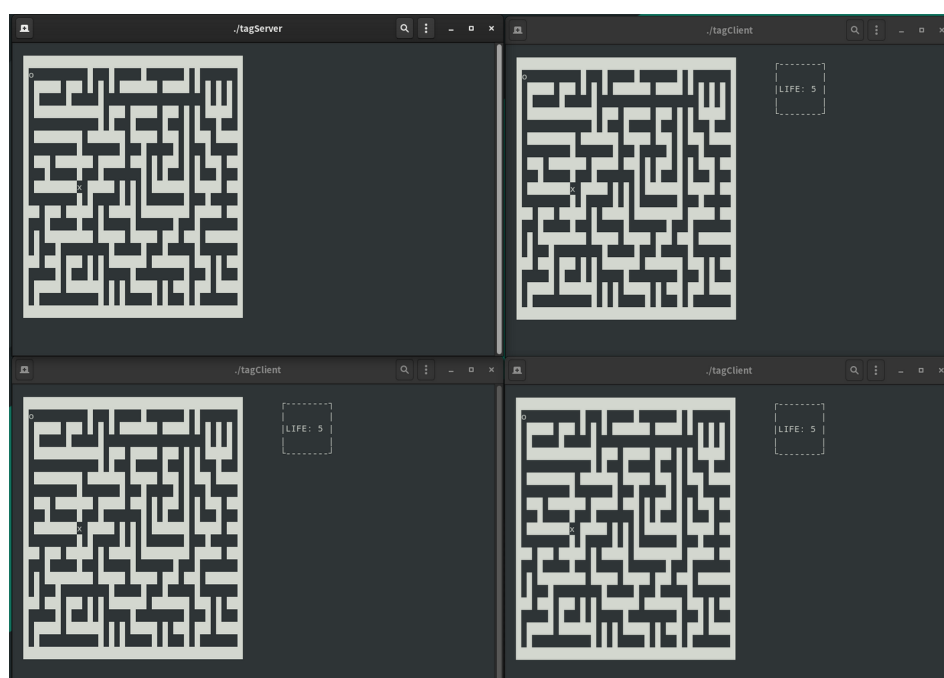


図 3 ゲーム開始したときの画面

# プログラムの内部構造の概要

本プログラムのディレクトリ構成を表 2 に示す。

表 2 本プログラムのディレクトリ構成

ファイル名/ディレクトリ名	ファイル or ディレクトリ	概要
.git	ディレクトリ	git のディレクトリ
.gitignore	ファイル	git の管理下に置かないファイルの定義
Makefile	ファイル	make コマンドを実行したときの動き
maze.c	ファイル	迷路生成のプログラム
maze.h	ファイル	迷路生成のヘッダファイル
mnet.c	ファイル	多重接続のプログラム
mnet.h	ファイル	多重接続のヘッダファイル
setupClient.c	ファイル	snet.h の実装
setupServer.c	ファイル	snet.h の実装
snet.h	ファイル	通信ライブラリのヘッダ
tagClient.c	ファイル	クライアントの main 関数が定義されているプログラム
tagGameForClient.c	ファイル	tagClient.c が使うプログラム
tagGameForClient.h	ファイル	tagClient.c が使うプログラムのヘッダ
tagGameForServer.c	ファイル	tagServer.c が使うプログラム
tagGameForServer.h	ファイル	tagServer.c が使うプログラムのヘッダ
tagGame.h	ファイル	サーバー、クライアント共通で使うプログラムのヘッダ
tagGame.c	ファイル	サーバー、クライアント共通で使うプログラムの
tagServer.c	ファイル	サーバーの main 関数が定義されているプログラム
tests	ディレクトリ	ライブラリを単体で動かす為に作成したディレクトリ。この中にテストプログラムが入る。メインの機能とは関係ない。

授業にて掲示されたサンプルプログラムでは”tagClient.c”と”tagServer.c”にてサーバー/クライアントの main 関数を定義し、それぞれが”tagGame”を使用した。しかし、これではサーバーとクライアントで処理を分けたいときにプログラムの改変が行いにくくなる。また、サーバーもしくはクライアントのどちらかしか使わない関数の一つのファイルにて定義されており、プログラムを確認するたびどちらのためのものなのか判断する必要があり、無駄に脳のリソースを割くことになる。

“tagGame”を元にした”tagGameForServer”と”tagGameForClient”を作成しこれらの問題を解決した。サーバーのみで使う関数を”tagGameForServer”に，クライアントのみで使う関数を”tagGameForClient”に定義し，サーバー/クライアントのメイン関数はこれらを使う。本当に共通で定義したほうがよいものを”tagGame”内で定義した。

## 工夫した点、アピールしたいポイント

### シューティングゲーム

サーバー側のキャラクターは弾丸を発射することができる。a/z/d/w キーを押すとそれぞれ右/下/左/上方向に弾丸が発射される。クライアントはこの弾丸に触れると LIFE が1減る。残機は5であり，0を下回るとゲーム終了となる。サーバー側は銃弾による攻撃を受けないので LIFE ウィンドウが表示されない。

### 迷路の自動生成

本プログラムを実行した際に現れる迷路は自動生成である。maze.h と maze.c にて実装されている。自動生成された迷路を図 4，図 5，図 6 に示す。

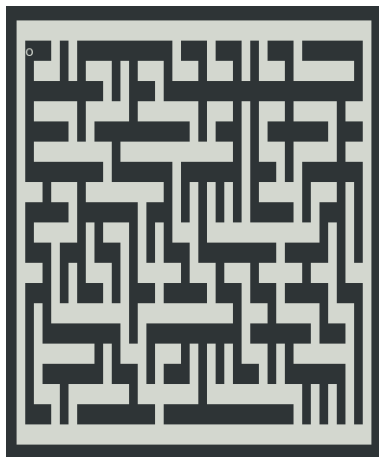


図 4 自動生成された迷路 1

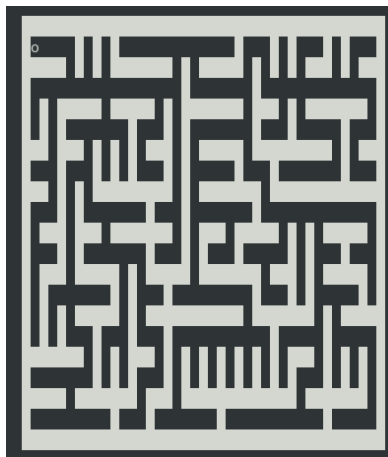


図 5 自動生成された迷路 2

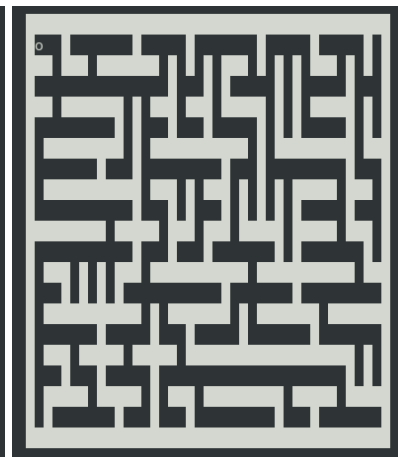


図 6 自動生成された迷路 3

### 多重接続

サーバーを実行してから任意のキーを押すまでクライアントの接続を受け付ける。クライアント数は0以上10未満を許容する。全クライアントは同じ位置からゲーム開始するが，独立して行動できる。クライアントのうち誰かが死ぬと，“WIN”もしくは“LOSE”というメッセージが表れその3秒後にゲーム全体が終了する。