**Push Box Game**

**20181566 강동우 20181709 홍재연**

**NCurses(team.cpp)**

**기능**

-wasd 키를 통해 조작(w:up a:left s:down d:right)

-wasd 키를 누르고 엔터키를 누르면 입력된 키에 해당하는 방향에 따라 0으로 표시된 캐릭터가 이동

-상자를 목적지에 도달할 수 없게 밀었을 경우 q버튼을 누르고 엔터키를 누르면 처음부터 다시 시도할 수 있도록 제작(이 때 step과 push횟수는 0으로 초기화됩니다.)

-캐릭터의 이동방향에 벽이 있거나 상자가 두 개 이상 연속으로 있으면 캐릭터가 움직이지 않고, 이동방향에 상자가 한 개 있으면서 그 상자 반대편에 또 다른 상자나 벽으로 막혀있지 않으면 상자도 캐릭터와 함께 이동 방향으로 한 칸 움직인다.

- step 횟수(움직인 횟수)와 push 횟수(상자를 밀어낸 횟수)를 화면에 보여주고 모든 상자가 목적지에 도달하면 무작위 키를 입력하여 현재 단계의 맵을 끝내고 다음 map으로 넘어가서 다시 게임을 시작한다.

**Source Code**

#include <iostream>

#include <ncurses.h>

#include <stdlib.h>

#include <stdio.h>

#include <string>

using namespace std;

int main(){

int map = 0; //1번째 맵 다음으로 바뀌려면 map++

int size = 100;

int row = 0;

int col = 0;

int cpos = 0;//현재 캐릭터의 위치

int tries = 0;

int push = 0;//캐릭터가 움직인 횟수

char direct[10];//화살표 키 입력

string position[] = {"34","74","57","58","66"};//캐릭터의 최초 위치, 꾸준한 수정필요, 맵이 종료될시 다음 인덱스의 위치로 넘어감

string answer[] = {"4444444444441111444444130114444413001444441302144444112011144441020014444100001444410011144441111444",

"4444444444444444444444111114444410001444441333144444122211444410000144441000014444111111444444444444",

"4444444444444444444444444444444111111114413000001441032220144130000014411111001444444111144444444444",

"4444444444444444444444111111144410000014441032301444102320144410323014441023201444100000144411111114",

"4444444444444444444441111444444100111144410000011411011000141303100211100010220110031000011111111111"};//각 레벨 맵의 구조

string start\_answer = answer[0];

start\_answer = start\_answer[1];

initscr();

keypad(stdscr, TRUE);

curs\_set(0);

noecho();

resize\_term(200,200);

start\_color();

init\_pair(1, COLOR\_WHITE,COLOR\_BLACK);//0

init\_pair(2, COLOR\_WHITE,COLOR\_YELLOW);//1

init\_pair(3, COLOR\_WHITE,COLOR\_RED);//2

init\_pair(4, COLOR\_WHITE,COLOR\_BLUE);//3 목적지

init\_pair(5, COLOR\_WHITE,COLOR\_GREEN);//4

for(map=0; map<5; map++){

string reanswer = answer[map];/\*q버튼을 눌렀을때 원상복구를 하기위해 초기문자열을 따로 저장\*/

string repos = position[map];/\*q버튼을 눌렀을때 원상복구를 하기위해 캐릭터의 최초의 위치를 따로 저장\*/

int level = map+1;

start\_answer = answer[map];

start\_answer = start\_answer[1];

int step = 0;//push 횟수

push = 0;

row = 0;

col = 0;

int count = 0; // 목적지의 갯수

for (int i = 0;i<size;i++){

if (answer[map][i] == '3'){

count += 1;//목적지의 갯수를 저장

}

}

clear();

mvprintw(50,30,"press the q button to try again");

mvprintw(51,30,"tries: %d",push);

mvprintw(52,30,"pushes: %d",step);

mvprintw(53,30,"level: %d",level);

refresh();

for (int i = 0;i<size;i++){//초기화면 출력과정

WINDOW \*win1;

char b = answer[map][i];

int a = b-'0';

win1 = newwin(5,5,5\*row,5\*col);

wbkgd(win1, COLOR\_PAIR(a+1));//번호에따른 팔레트 배치

wattron(win1, COLOR\_PAIR(a+1));

if (row == (int)((char)position[map][0]-'0')&&col == (int)((char)position[map][1]-'0')){

//n번째 단계에서의 캐릭터위치

cpos = i;

mvwprintw(win1,1,1,"0");

}

wrefresh(win1);

col++;

if(col == 10){//한줄을 모두 출력했을 때

row++;

col = 0;

}

delwin(win1);

}

refresh();

scanw("%s",direct);//(wasd키중 키 하나를 누르고 엔터키) 이 과정을 반복

while (count != 0){

count = 0;

if (direct[0] == 'q'){//처음부터 다시 시도

answer[map] = reanswer;

position[map] = repos;

map--;//for문의 마지막 map++ 무력화

break;

}

if (direct[0] == 's'){//캐릭터가 아래로

push++;

if(answer[map][cpos+10] == '1'){}

else if (answer[map][cpos+10] == '2'){

if (answer[map][cpos+20] == '0'){

char temp = answer[map][cpos+20];

answer[map][cpos+20] = '2';

answer[map][cpos+10] = temp;

cpos += 10;//아래로 한 칸

step += 1;

position[map][0] += 1;//이 부분 한 맵이 종료될때 마다 [0][0] -> [1][0] ->[2][0] 이런식으로 넘어감

}

else if (answer[map][cpos+20] == '3'){

answer[map][cpos+20] = '2';

if (start\_answer[cpos+10] == '3'){//초기단계에서 목적지였을 때

answer[map][cpos+10] = '3';

}

else{

answer[map][cpos+10] = '0';

}

cpos += 10;

step += 1;

position[map][0] += 1;//맵 넘어갈때마다 수정

}

else {}

}

else{

cpos += 10;

position[map][0] += 1;//맵 넘어갈때마다 수정

}

}

else if (direct[0] == 'w'){//위로

push++;

if (answer[map][cpos-10] == '1'){}

else if (answer[map][cpos-10] == '2'){

if (answer[map][cpos-20] == '0'){

char temp = answer[map][cpos-20];

answer[map][cpos-20] = '2';

answer[map][cpos-10] = temp;

cpos -= 10;//위로 한 칸

step += 1;

position[map][0] -= 1;//맵 넘어갈때마다 수정

}

else if (answer[map][cpos-20] == '3'){

answer[map][cpos-20] = '2';

if (start\_answer[cpos-10] == '3'){//초기에 목적지였을 때

answer[map][cpos-10] = '3';

}

else{

answer[map][cpos-10] = '0';

}

cpos -= 10;

step += 1;

position[map][0] -= 1;//맵 넘어갈때마다 수정

}

else{

}

}

else{

cpos -= 10;

position[map][0] -= 1;//맵 넘어갈때마다 수정

}

}

else if (direct[0] == 'a'){//왼쪽

push++;

if (answer[map][cpos-1] == '1'){

}

else if (answer[map][cpos-1] == '2'){

if (answer[map][cpos-2] == '0'){

char temp = answer[map][cpos-2];

answer[map][cpos-2] = '2';

answer[map][cpos-1] = temp;

cpos -= 1;//왼쪽으로 한 칸

step += 1;

position[map][1] -= 1;//맵 넘어갈때마다 수정

}

else if (answer[map][cpos-2] == '3'){

answer[map][cpos-2] = '2';

if (start\_answer[cpos-1] == '3'){//초기에 목적지였을 때

answer[map][cpos-1] = '3';

}

else{

answer[map][cpos-1] = '0';

}

cpos -= 1;

step += 1;

position[map][1] -= 1;//맵 넘어갈때마다 수정

}

else{

}

}

else{

cpos -= 1;

position[map][1] -= 1;//맵 넘어갈때마다 수정

}

}

else if (direct[0] == 'd'){//오른쪽

push++;

if (answer[map][cpos+1] == '1'){

}

else if (answer[map][cpos+1] == '2'){

if (answer[map][cpos+2] == '0'){

char temp = answer[map][cpos+2];

answer[map][cpos+2] = '2';

answer[map][cpos+1] = temp;

cpos += 1;//오른쪽으로 한 칸

step += 1;

position[map][1] += 1;//맵 넘어갈때마다 수정

}

else if (answer[map][cpos+2] == '3'){

answer[map][cpos+2] = '2';

if (start\_answer[cpos+1] == '3'){//초기에 목적지였을 때

answer[map][cpos+1] = '3';

}

else{

answer[map][cpos+1] = '0';

}

cpos += 1;

step += 1;

position[map][1] += 1;//맵 넘어갈때마다 수정

}

else{

}

}

else{

cpos += 1;

position[map][1] += 1;//맵 넘어갈때마다 수정

}

}

clear();

mvprintw(50,30,"press the q button to try again");

mvprintw(51,30,"tries: %d",push);

mvprintw(52,30,"pushes: %d", step);

mvprintw(53,30,"level: %d",level);

refresh();

row = 0;

col = 0;

for (int i = 0;i<size;i++){//초기화면 출력과정

WINDOW \*win1;

char b = answer[map][i];

int a = b-'0';

win1 = newwin(5,5,5\*row,5\*col);

wbkgd(win1, COLOR\_PAIR(a+1));//번호에따른 팔레트 배치

wattron(win1, COLOR\_PAIR(a+1));//

if (row == (int)((char)position[map][0]-'0')&&col == (int)((char)position[map][1]-'0')){

//n번쨰 단계에서의 캐릭터위치

cpos = i;

mvwprintw(win1,1,1,"0");

}

wrefresh(win1);

col++;

if(col == 10){//한줄을 모두 출력했을 때

row++;

col = 0;

}

delwin(win1);

}

refresh();

for (int i = 0;i<size;i++){

if (answer[map][i] == '3'){

count += 1;//입력된 키에 따라 변경된 맵의 남은 목적지 갯수 확인

}

}

scanw("%s", direct);

}

}

refresh();

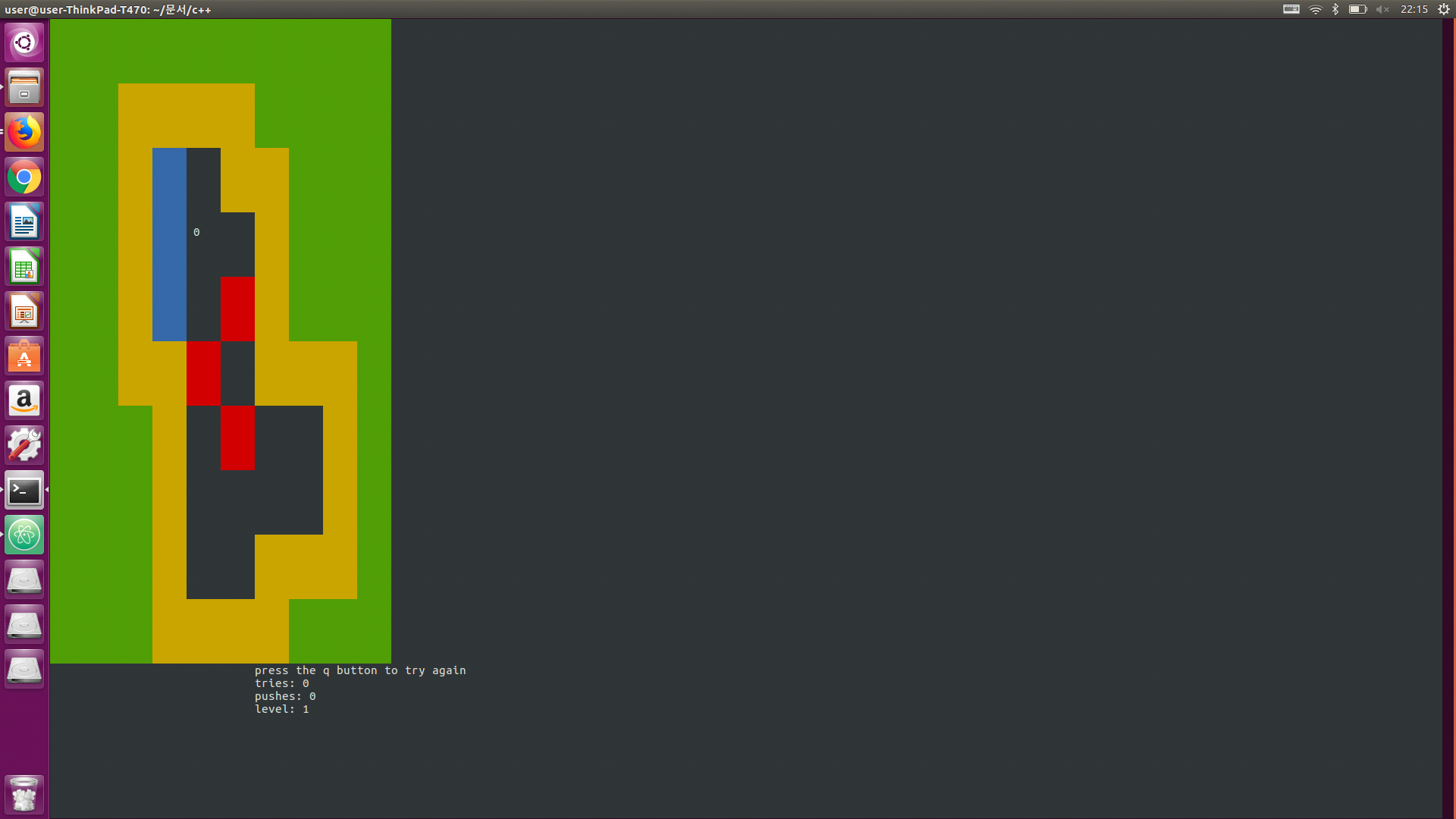
getch();

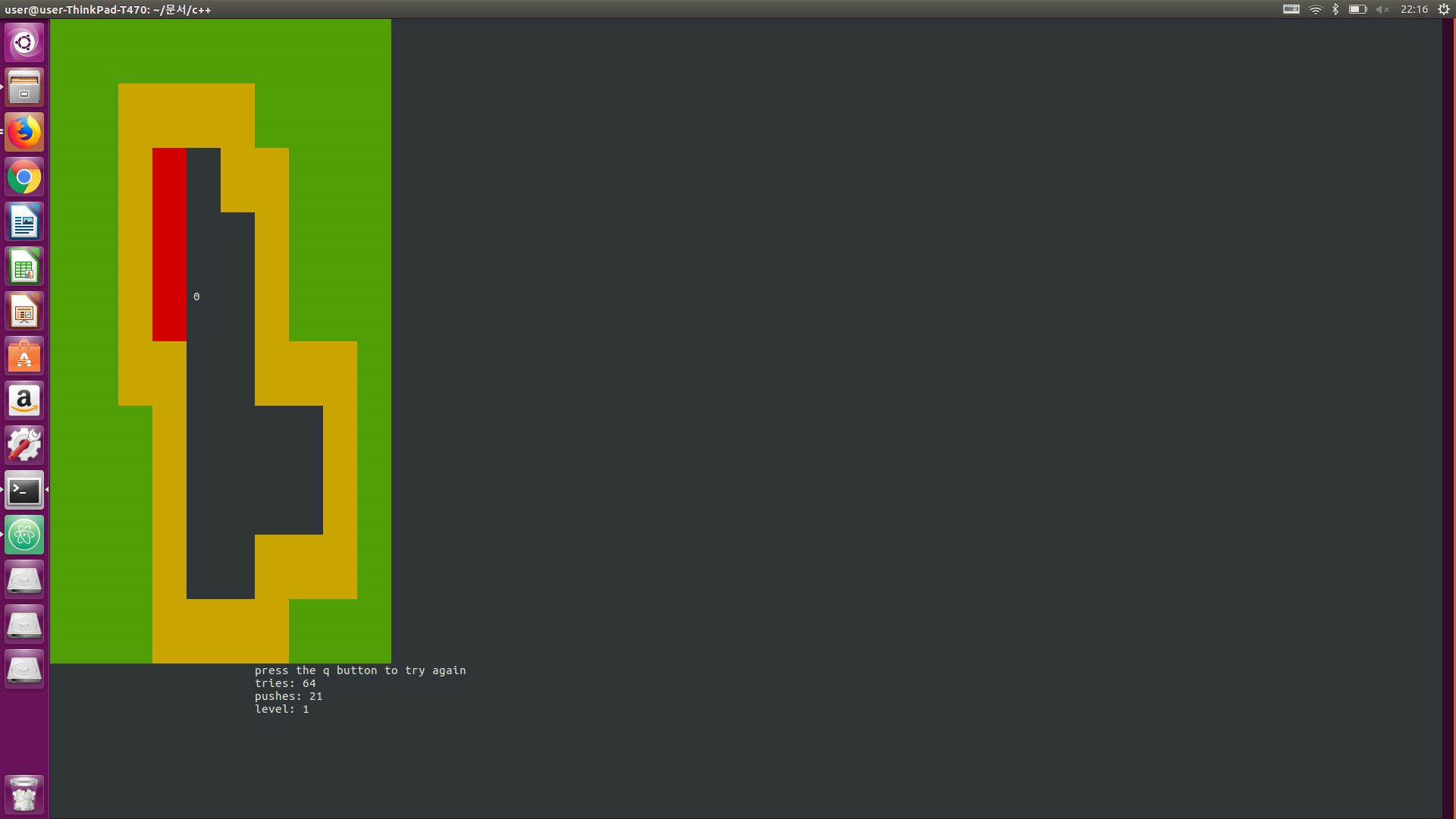
endwin();

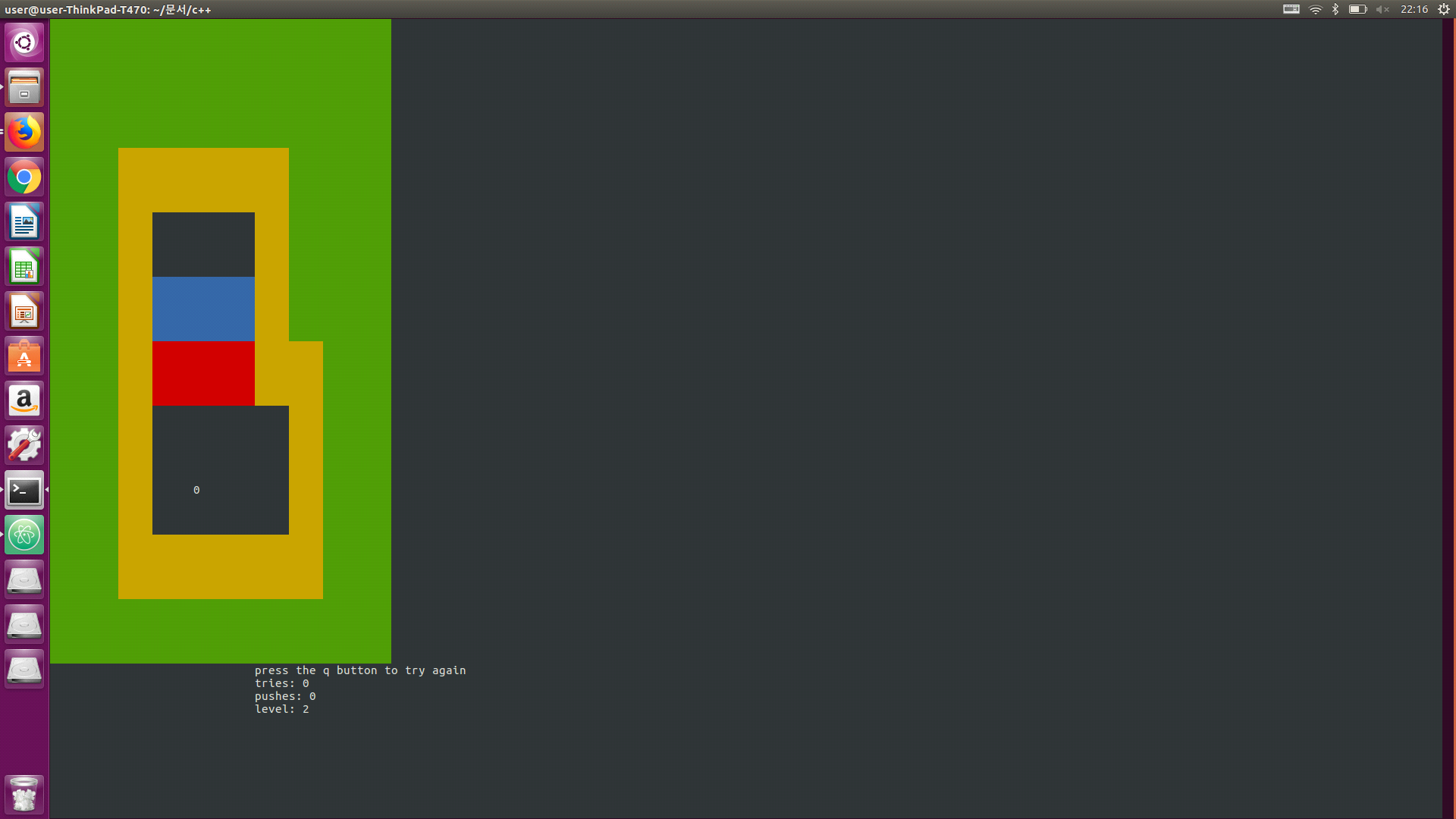
return 0;

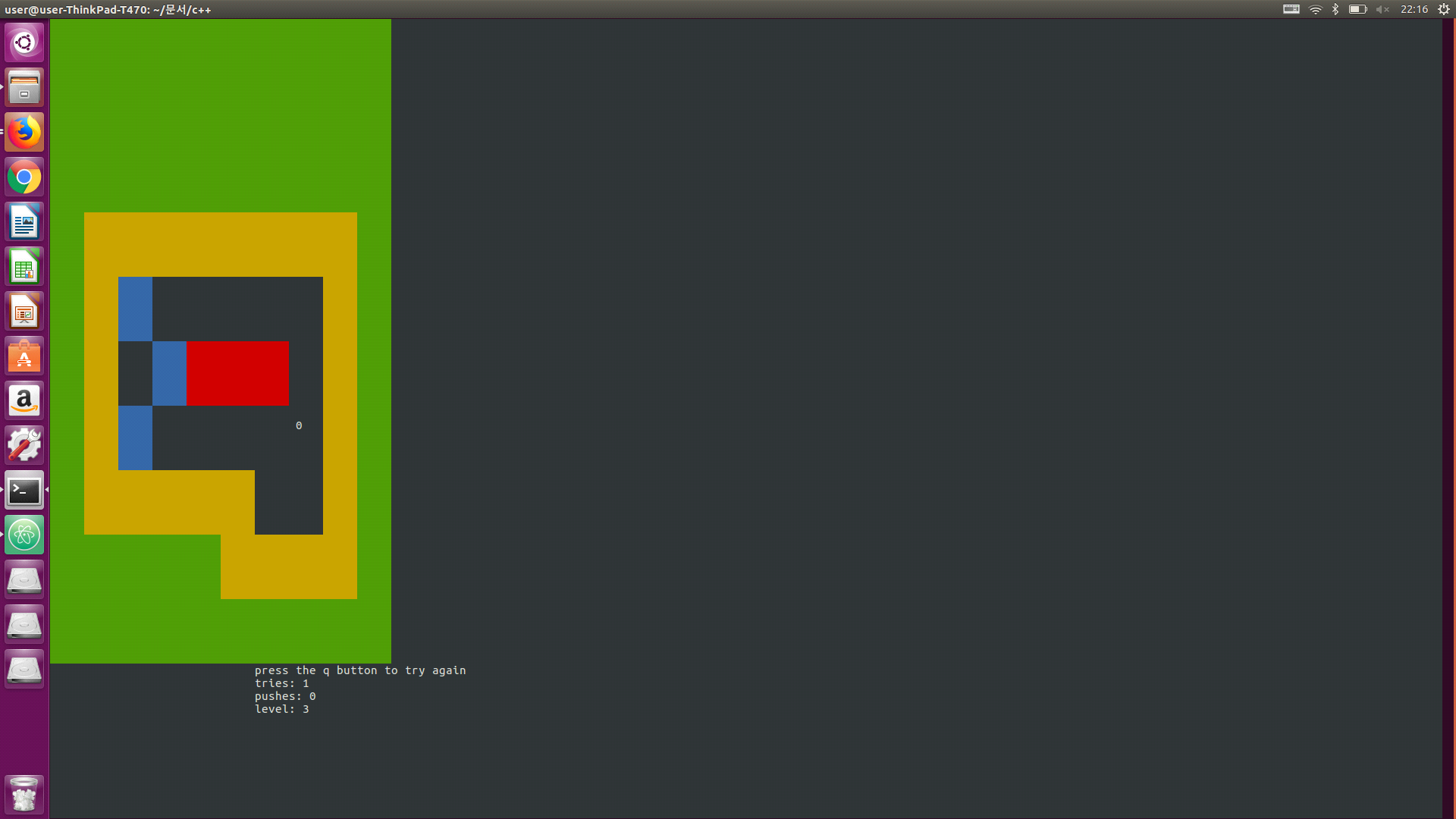
}

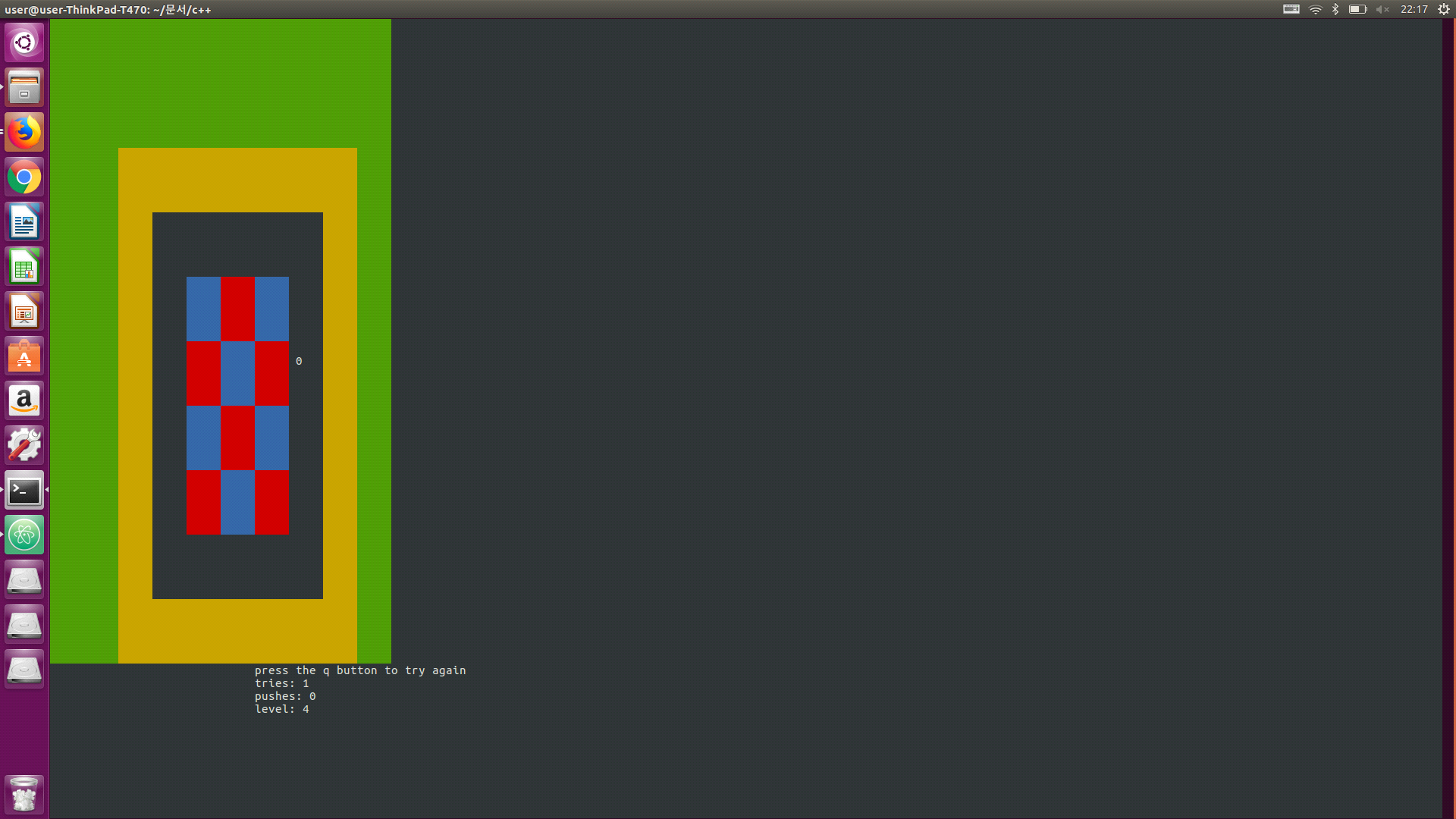
**실행화면**

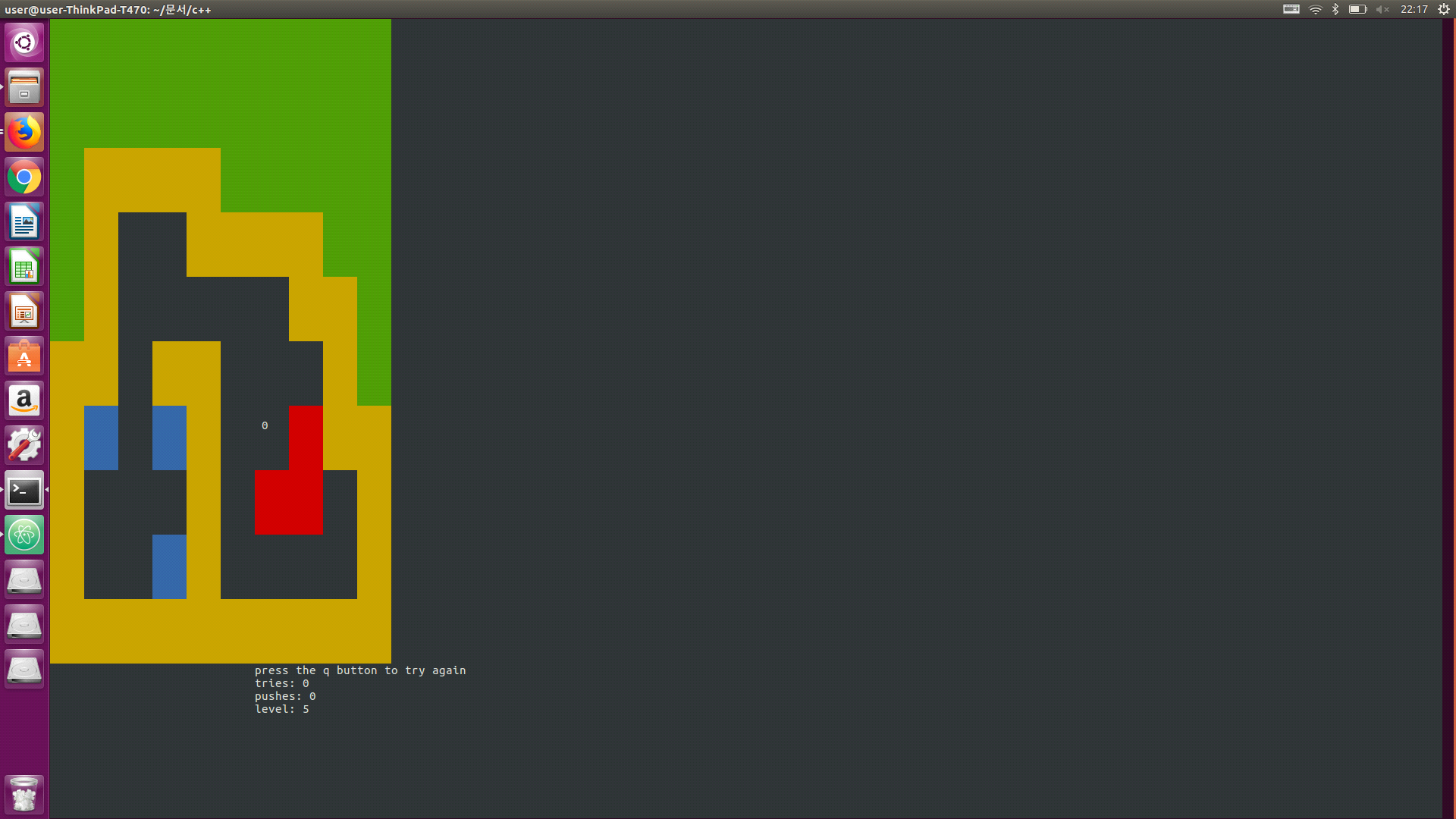






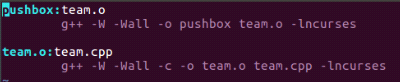






**Makefile**

**1. vi Makefile**



pushbox:team.o

g++ -W -Wall -o pushbox team.o -lncurses

team.o:team.cpp

g++ -W -Wall -c -o team.o team.cpp -lncurses

**team.cpp -> team.o -> pushbox**

**2. make**



**make 명령 입력->Makefile 찾음->pushbox 생성**

**3. ./pushbox**



team.cpp 실행