

print\_map()とmain文を変更した。print\_mapは、現在のx,y座標になった際にマスではなく矢印を表示する。main文は、ifの分岐を追加した。変数として、現在地のx,y座標に加えて、一手前のx,y座標を管理するものを追加した。マップの範囲外であることを確認するために現在の座標を移動させるが、同時にpre\_pに座標を記憶しておき、範囲外であった場合はpre\_pの座標に戻るアルゴリズムになっている。同時にその方向には進めない内容のメッセージを出力する。スタート地点と、ゴール地点に移動した場合も、if文で管理しており、文字を出力している。図1はスタート地点に戻ってしまった際の実行結果である。下方向から移動してきたため、矢印は上方向になっており、メッセージを正しく表示することができている。図2は範囲外に出ないようにするメッセージを表示する実行結果である。図3はゴールした際にメッセージを表示し、プログラムを終了する様子を示している。

1	void print_map(point p,int dir){
2	int col,row;
3	for(row = 0;row < MAP_SIZE;row++){
4	for(col = 0;col < MAP_SIZE;col++){
5	if(col == p.x && row == p.y){
6	printf("%s",arrw_disp[dir]);}
7	else{printf("%s",chip_disp[map[row][col]]);}
8	}
9	puts("");
10	}

算譜1 迷路を表示するコード

1	if(cur_p.x > INPUT_MAX    cur_p.x < INPUT_MIN    cur_p.y > INPUT_MAX    cur_p.y < INPUT_MIN    map[cur_p.y][cur_p.x] == WALL){
2	puts("その方向には進めません！");
3	cur_p.x = pre_p.x;cur_p.y = pre_p.y;
4	}
5	if(map[cur_p.y][cur_p.x] == STRT){puts("スタート地点に戻ってしまいました。 . . ");}
6	if(map[cur_p.y][cur_p.x] == GOAL){
7	puts("ゴールに到達しました！");
8	game_end = 1;}

算譜2 状況に応じて文字を出力するコード

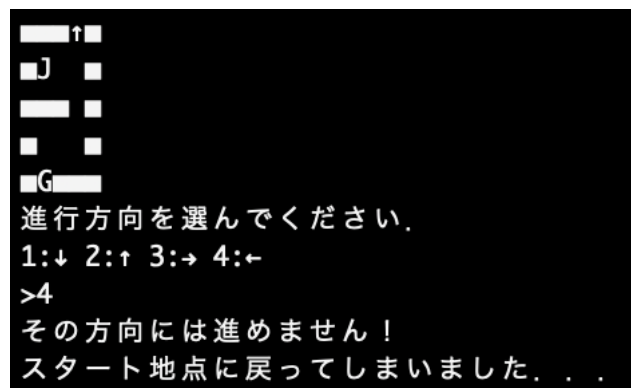


図1 スタート地点に戻ったことを表示するキャプチャ

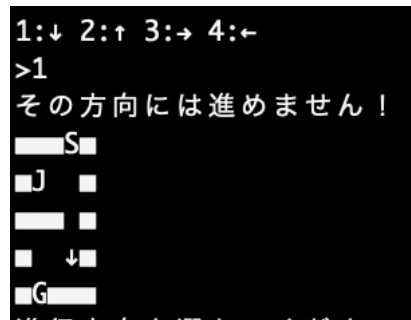


図2 範囲外に出ないようにしているキャプチャ



図3 ゴール画面を表示するキャプチャ