**[네모 로직]**

**6팀 프로젝트 완료 보고서**

6팀 조 원:

김민영(2017111502)

김예린(2017113030)

김주희(2016118155)

송유나(2017110821)

목 차

1. 서론\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1.1 프로젝트 명

1.2 주제 선정 및 프로젝트 소개

1.3 프로젝트 기간

1.4 개발 도구 및 환경

2. 본론\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2.1 기본 시나리오

2.2 구조

2.3 사용 header 및 library

2.4 함수

2.5 기능

2.6 게임 동작 결과

3. 결론\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

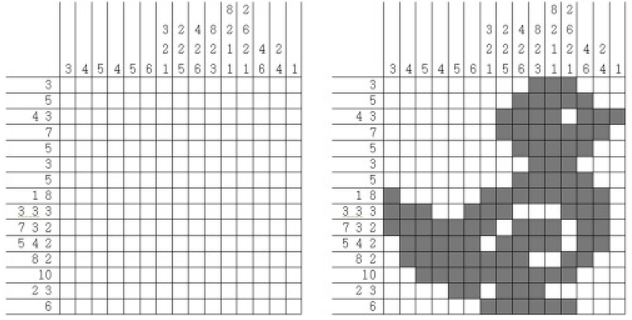
3.1 교재 및 참고 사이트

3.2 추후 계획

**서론\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

* 프로젝트 명: 네모 로직 게임
* 주제 선정 및 프로젝트 소개

초반에는 Curses Library를 이용한 Coloring Book 프로그램을 주제로 선정하였습니다. 그런데 도안을 뿌려주는 과정에서 overlap되는 현상이 발생하여 비슷한 네모 로직 게임으로 선정을 하였습니다.



네모 로직은 위와 같이 가로, 세로에 나열된 숫자들을 단서로 그림을 완성하는 퍼즐 게임입니다. 이 게임을 지금까지 배운 수업 내용 바탕으로 콘솔 게임으로 제작하였습니다.

* 프로젝트 기간: 2018. 11. 07 ~ 2018. 12. 20
* 개발 환경 및 도구

**언어:** C언어

**IDE:** vim editor

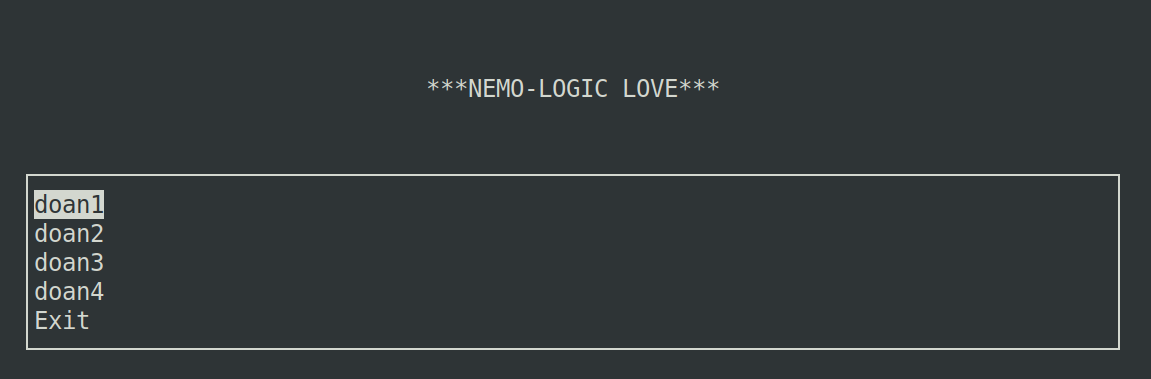
**O/S:** Ubuntu

**소스 코드 관리:** Git(GitHub)

**본론\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

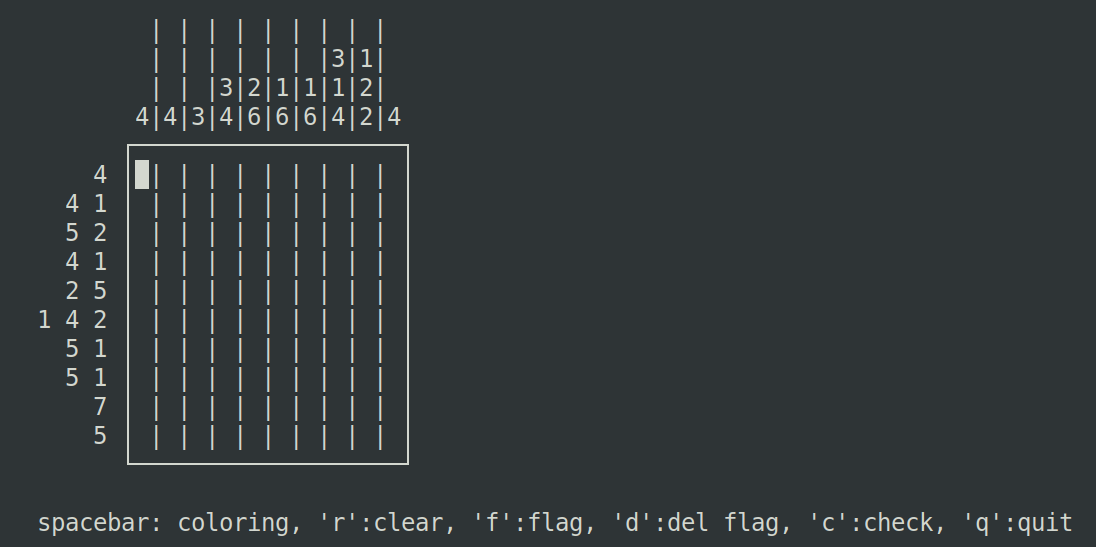
* 기본 시나리오

(1)



메뉴 화면에서 원하는 도안을 선택한다.

(2)



가로, 세로 숫자들 바탕으로 Space bar 키로 색칠해 나간다. 마지막 ‘c’ 키로 답을 체크한다.

* 구조

|  |  |
| --- | --- |
| **파일명** | **내용** |
| testing.c | 게임 시작화면 |
| coloring.h | coloring.c의 함수 선언 및 변수 선언 |
| coloring.c | 게임 화면 |
| doan1 ~ doan4 | 가로, 세로 숫자 담긴 텍스트 파일 |
| a\_doan1 ~ a\_doan4 | 답이 담긴 텍스트 파일 |

* 사용 header 및 library

|  |  |
| --- | --- |
| **이름** | **내용** |
| <stdio.h> | 기본 c언어 문법 |
| <curses.h> | 정답 확인을 위해 답 데이터 담을 때 필요 |
| <string.h> | 문자열관련 라이브러리 |
| <menu.h> | 메뉴 생성하는 라이브러리 |
| <stdlib.h> | 종료 코드 |

* 함수

|  |  |
| --- | --- |
| **이름** | **내용** |
| Start | 게임 시작함수 (가로/세로 숫자 출력, 키 조작) |
| insertAnswer | 정답 확인을 위해 답 데이터 담는 함수 |
| checkAnswer | 정답 확인하는 함수 |

* 기능

1. 메뉴 화면

\* 주요 변수

|  |  |
| --- | --- |
| **이름** | **기능** |
| menuwin | WINDOW 객체, 메뉴 박스 생성할 공간 |
| munu | 메뉴 목록 이름 모은 배열 |

|  |
| --- |
| //WINDOW 객체 생성  WINDOW \* menuwin=newwin(7, dlug-12, szer-9, 6);  box(menuwin, 0, 0);  refresh();  wrefresh(menuwin);  keypad(menuwin, TRUE);  //메뉴 내용 생성  char \*menu[] = {  "doan1",  "doan2",  "doan3",  "doan4",  "Exit",  };    //메뉴 출력  for(i = 0; i < 5; i++) {  if(i == zaznacz)  wattron(menuwin, A\_REVERSE);  mvwprintw(menuwin, i+1, 1, menu[i]);  if (i == zaznacz)  wattroff(menuwin, A\_REVERSE);  }  //커서 지우기  curs\_set(0);  //아래는 Key에 따른 기능 |

1. 게임 화면

\* 주요 변수

|  |  |
| --- | --- |
| **이름** | **기능** |
| design d[10][20]; | **구조체 design**는 색칠과 flag 여부를 알리는 check변수와 답이 담긴 answer변수로 구성되어 있다.  **d[10][20]** 색칠하는 공간 |
| x, y | 현재 커서 위치 |
| row\_x / row\_y  col\_x / col\_y  q / p | 가로, 세로 숫자 출력을 위해 사용하는 변수들 |
| Test | init\_pair 함수의 첫 번째 인자, init\_pair 호출할 때마다 test를 1씩 증가  (첫 번째 인자를 같은 수를 넣으니 색칠한 공간들이 생성  컬러로 바뀌는 현상이 나타남, test 변수로 값을 다르게 주  어 다른 공간 영향 가지 않게 하였다.) |

\* ↓,↑,←,→ 키를 누를 때 도안 공간에만 움직이도록 체크한다.

(2)-1. 색칠/지우기 기능과 flag/지우기 기능

\* **space bar 키 누르면 색칠하고 지우는 기능**

d[y][x].check 값 1이면 색칠 상태

d[y][x].check 값 0이면 지운 상태

|  |
| --- |
| 만약 d[y][x].check가 0이면 즉, 지운 상태이면 {  // 지운 상태이므로 색칠 상태로 변경해 주고  // 글자 색상 검정, 배경 색상 흰색의 컬러를 설정한다.  d[y][x].check = 1;  init\_pair(test, COLOR\_BLACK, COLOR\_WHITE);  } d[y][x].check가 1이면 즉, 색칠 상태이면 {  // 색칠 상태이므로 지운 상태로 변경해 주고  // 글자 색상 흰색, 배경 색상 검정이 컬러를 설정한다.  d[y][x].check = 0;  init\_pair(test, COLOR\_WHITE, COLOR\_BLACK);  }  // 설정한 컬러 on하고 색칠/지움  attron(COLOR\_PAIR(test));  move(y, x);  addstr(" ");  // 설정한 컬러 off하고 refresh  attroff(COLOR\_PAIR(test));  refresh();  move(y, x);  refresh();  test++; |

\* **‘f’ 키 누르면 플래그 출력 기능**

d[y][x].check 값 -1이면 그 공간은 플래그 출력 상태

컬러 설정하고 출력한 다음 refresh는 위와 같음

\* **‘d’ 키 누르면 플래그 출력 상태인 모든 공간을 지우기 기능**

d[y][x].check 값 1인 공간을 d[y][x].check 값 0으로 바꿔 지운 상태로 변경

컬러 설정하고 출력한 다음 refresh는 위와 같음

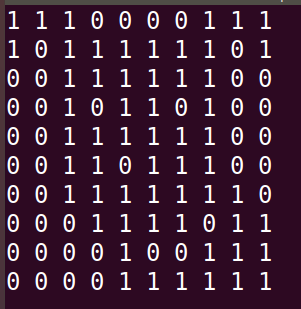
(2)-2. clear 기능 (**‘r’ 키)**

모든 공간을 d[y][x].check 값 0으로 바꿔 지운 상태로 변경

컬러 설정하고 출력한 다음 refresh한다.

(2)-3. 결과 check 기능 (**‘c’ 키)**

\* **insertAnswer 함수**



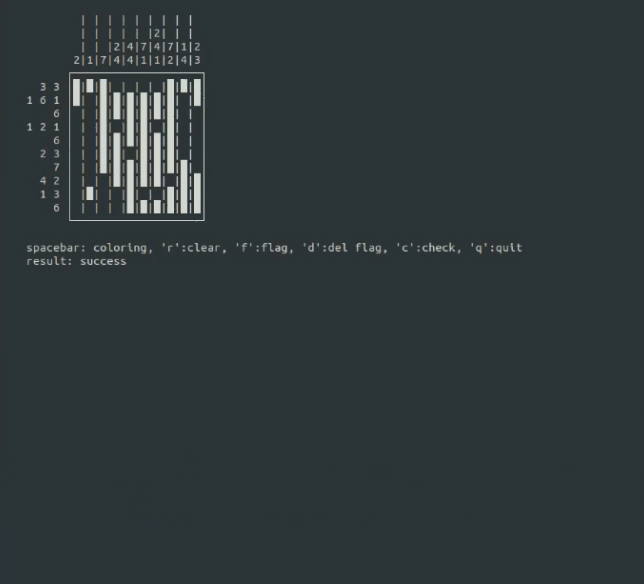
우선 답 체크위한 데이터는 0과 1로 구성된 답 txt 파일을 읽어와 d[y][x].answer에 넣는다.

\* **checkAnswer 함수**

d[y][x].check와 d[y][x].answer 값을 비교하여 모두 같으면 Success를 출력

다르면 fail을 출력한다.

* 게임 동작 결과



**결론**

* 교재 및 참고 사이트

Understanding Unix/Linux Programming, Bruce Molay, Prentice-Hall.

https://stackoverflow.com/questions/41625257/basic-ncurses-menu/41627095#41627095

* 추후 계획

1. 현재 구현한 게임은 도안이 10x10로 고정되어 있는 데 동적 크기인 도안으로 바꿀 예정입니다.
2. 답을 체크할 때 텍스트 파일로 값을 받아와 답을 체크합니다. 이 부분은 알고리즘을 짜서 체크할 예정입니다.