C언어를 이용한 오목 게임

-개인 프로젝트-

Omok Game Used C

2020년 8월 23일

선린 인터넷 고등학교

정보 보호 과

Unifox 12기

임도현

**목 차**

**제 1 장 서 론 .......................................................... 03**

제 1 절 프로그램 소개 ........................................................... 03

제 2 절 개발 동기 ……………………………………………........... 03

**제 2 장 내 용 ………………………………………….... 04**

제 1 절 프로그램 구조 ………………………………………….… 04

제 2 절 프로그램 구현 방법 ……………………………….….. 06

제 3 절 프로그램 기능 추가 및 개선 ……………....….… 08

**제 3 장 결 론 ……………………………………….…… 08**

제 1 절 느낀 점 ………………………………………………....……. 08

**제 4 장 참고 문헌 ...................................................... 09**

**제 1 장 서 론**

제 1 절 프로그램 소개

오목은 바둑판에 두 사람이 번갈아 돌을 놓아 가로나 세로, 대각선으로 다섯 개의 연속된 돌을 놓으면 이기는 게임이다. 19x19 크기의 바둑판에서 하기도 하나 정식 경기는 보통 15x15 크기의 판을 이용한다.

하지만 이 프로그램에서는 19x19 판을 사용하여 경기를 진행한다

제 2 절 개발 동기

선린에 들어오기 전 여러 프로그래밍 언어를 접해봤지만 c언어로 제대로된 프로그램을 만들어본 경험이 없었다. 그래서 이번 기회에 과거에 한번 만들어 봐야지 생각하며 적어뒀던 “만들고 싶은 프로그램” 리스트 중에서 많이 해봤던 오목이 가장 눈에 뛰어서 만들겠다고 결정했습니다.

**제 2 장 내 용**

제 1 절 프로그램 구조

이 프로그램의 큰 틀

* 이 프로그램은 2명이서 플레이를 해야하므로 키보드가 서로 붙어 있으면 불편하다고 느꼈기 때문에 W-A-S-D spacebar 와 8-4-5-6 0번으로 바둑돌을 움직인다.
* Spacebar 나 0번 으로 바둑돌을 놓으면 가로세로 대각선을 바둑돌이 놓일 때 마다 검사한다.

세부사항

* 바둑돌이 움직였을 경우 그 전 위치에 놓여있던 바둑돌 또는 바둑판을 다시 구현
* 이미 놓여있던 바둑돌을 체크하기 편하게 하기 위해 좌표와 배열을 일치시켜 줌
* 플레이 시간을 계산하기 위해 멀티스레드를 사용함

제 2 절 프로그램 구현 방법

1. 5줄 체크 알고리즘 //기본배열은 0으로 초기화 되어있으며 흑돌은1 백돌은

- 가로,세로:

돌이 놓인 가로 또는 세로줄을 처음부터 끝까지 검사하며 배열에 돌이 놓여질 때 색상을 의미하는 stone\_turn 과 같은 색상이 연속으로 5개가 있으면 끝나도록 만들어주었습니다.

//여기서 stone\_turn 은 흑돌과 백돌의 순서를 구분하기 위한 전역 변수로

//돌이 놓일 때 마다 1또는 2로 바뀜.

int i, j = 0, k = 1, count = 0;

int a[20] = { 0, }; //5줄 체크할때 사용하는 배열 초기화

for (i = 0; i <= 5; i++) a[i] = 0; j = 0; //5줄 체크할때 사용하는 배열 초기화

for (i = 1; i <= 19; i++) { //세로 체크

for (j = 0; j <= 4; j++) { //

a[j] = Omok\_Board[i + j][y];

}

for (j = 0; j <= 4; j++) { //

if (a[j] == stone\_turn) count++;

}

if (count == 5) return 1;

for (j = 0; j < 5; j++) a[i] = 0;

count = 0;

}

- 대각선 ↘

돌이 놓인 위치에서 대각선 왼쪽 위 첫번째 x좌표를 계산 즉 (x좌표 - y좌표,0) 여기서부터 x와 y를 1식 증가시켜 x의 좌표가 19이하가 될 때 까지 반복시키고 가로,세로와 같은 방법으로 연속된 돌

을 체크할 배열을 만들어 검사했습니다

k = 1; for (i = 0; i <= 5; i++) a[i] = 0; j = 0;//5줄 체크할때 사용하는 배열 초기화

for (i = x + y - 1; i >= 1; i--) { //대각선↗ 체크

if (i > 0) {

for (j = 0; j <= 4; j++) { //

a[j] = Omok\_Board[k + j][i - j];

}

for (j = 0; j <= 4; j++) { //

if (a[j] == stone\_turn) count++;

}

if (count == 5) return 1;

for (j = 0; j < 5; j++) a[j] = 0;

count = 0;

}

k++;

}

- 대각선 ↗

돌이 놓인 위치에서 왼쪽 아래 대각선 x=0 일때 y좌표를 계산하면 (0,x+y-1)이 되고

여기서부터 x를 1씩 증가하고 y를 1씩 감소 시켜주면서 1증가 감소 할 때 앞에 배열에서 미리 앞에 5개를 검사하고 다시 1을 증가하고 이 방법을 반복

k = 1; for (i = 0; i <= 5; i++) a[i] = 0; j = 0;//5줄 체크할때 사용하는 배열 초기화

for (i = x + y - 1; i >= 1; i--) { //대각선↗ 체크

if (i > 0) {

for (j = 0; j <= 4; j++) { //

a[j] = Omok\_Board[k + j][i - j];

}

for (j = 0; j <= 4; j++) { //

if (a[j] == stone\_turn) count++;

}

if (count == 5) return 1;

for (j = 0; j < 5; j++) a[j] = 0;

count = 0;

}

k++;

}

1. 돌 삭제

-돌이 움직일 때 그전 위치에 있던 돌을 지우고 바둑판으로 void rm\_stone(int x, int y) {

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), 6);

Gotoxy(x, y); // 현재 돌이 위치한 곳으로 이동

if (Omok\_Board[x][y] == 0) {

if (x == 1 && y == 1) // 그곳이 만약 1,1 (왼쪽 위 모서리)인 경우

printf("┌─");

else if (x == 19 && y == 1) { // 그곳이 만약 19,1 (오른쪽 위 모서리)인 경우

printf(" "); //버그 처리용도

Gotoxy(x, y);

printf("┐");

}

else if (x == 1 && y == 19) // 그곳이 만약 0,19 (왼쪽 아래 모서리)인 경우

printf("└─");

else if (x == 19 && y == 19) { // 그곳이 만약 19,19 (오른쪽 아래 모서리)인 경우

printf(" ");

Gotoxy(x, y);

printf("┘");

}

else if (x == 1) // 그곳이 만약 1,\* (왼쪽줄)인 경우

printf("├─");

else if (y == 1) // 그곳이 만약 \*,1 (윗줄)인 경우

printf("┬─");

else if (x == 19) { // 그곳이 만약 19,\* (오른쪽줄)인 경우

printf(" ");

Gotoxy(x, y);

printf("┤");

}

else if (y == 19) // 그곳이 만약 \*,19 (아랫줄)인 경우

printf("┴─");

else // 그 외의 경우

printf("┼─");

}else if(Omok\_Board[x][y]==1){

Put\_stone(1);

}else if(Omok\_Board[x][y] == 2){

Put\_stone(2);

이 부분을 통해 움직임을 입력 받고 한칸 이동하기전 x,y좌표를 rm\_stone으로 전송해서 바둑판을 다시 채우고 빠른 속도로 한칸 이동된 위치에 다시 바둑알을 놓음으로 돌이 움직임을 표현하며 바둑판을 유지할 수 있습니다

제 3 절 추가하고 싶은 프로그램 기능 추가 및 개선

1. 타이머

-이미 플레이 시간은 있지만 한사람이 둘 수 있는 시간을 제한하는 타이머를 만들고 싶다

2) 렌주룰

- 오목에는 흑돌 33금지, 6목 금지 등 여러 렌주룰이 있는데

이런 다양한 룰을 검사하는 알고리즘을 추가해보고 싶습니다

3)좀더 깔끔한 사용자 인터페이스 환경 제공

* 처음으로 만든 제대로 된 프로그램인지라 인터페이스 신경을 최대한 쓴다고 했지만 어렵네요….

**제 4 장 결 론**

오목 프로젝트는 나의 첫 번째 c언어 장문 코딩입니다. 처음 아무것도 없는 화면에 코드를 작성 한다는 게 그냥 막막하기만 했다. 하지만 “뭐 어쩌겠는가 그냥 부딪혀 봐야지” 라는 생각으로 코드를 작성했고

가장 어려웠던 부분은 5줄 체크하는 부분에서 대각선 왼쪽 오른쪽

2개를 작성할 때는 어떻게 해야 대각선을 검사할 수 있을지 한참 고민했습니다. 그림판에 그림을 그려보며 감을 익혔고 x와 y좌표를 보고 힌트를 얻어 완성했을 때 스스로 풀었다는 거에 한층 더 성장한 듯한 기분을 느꼈습니다. 2번째로 어려웠던 것은 처음 바둑판을 그릴 때 윈도우 창의 비율이 1:2 란 사실을 몰라서 한참 돌았습니다..

그래도 마지막을 앞두고 모든 기능들이 잘 작동하고 에러없이 사용자에게 빈화면이 아닌 진짜 오목을 플레이할 수 있게 만들었다는 생각에 안도감과 만족감은 극에 달했습니다. 코딩을 하면서 처음 느껴보는 기분으로 기억속에 각인 됬고 이제 오목 게임을 발판삼아 내가 원했던 프로그램을 스스로 진행하며 실력을 쌓을 것 입니다.

**제 5 장 참고문헌**

[1] c언어 콘솔에 글자 색 넣기

<http://blog.naver.com/PostView.nhn?blogId=wcwtmt&logNo=10172917569>

[2] c언어 gotoxy 사용, 움직임 표현

<https://www.youtube.com/watch?v=5CIMIneVp4k>

[3] c언어 \_getch( ) 함수

<http://blog.naver.com/PostView.nhn?blogId=tipsware&logNo=221242072724>

[4] c언어 콘솔창 크기설정

<http://egloos.zum.com/EireneHue/v/350607>

[5] c언어 멀티스레드

<https://m.blog.naver.com/PostView.nhn?blogId=chgy2131&logNo=221111771427&proxyReferer=https:%2F%2Fwww.google.com%2F>