

Warming-up 프로그램 2

2023-2 컴퓨터 그래픽스

5. 카드 짝 맞추기 게임 만들기

- 4x4 크기의 격자 보드를 만든다. 격자 칸을 *로 표시한다. (다른 방법을 적용해도 무방함)
- 8개의 다른 대문자를 각각 2개씩 보드에 무작위로 배치한다.
 - 보드 위쪽에 a b c d, 좌측에 1 2 3 4를 표시하여 칸의 위치를 알 수 있게 한다.
 - 2개의 격자를 선택한다. 선택된 보드 칸에 o를 그려 선택된 것을 알려준다. 엔터키를 치면 선택된 칸의 문자가 보여진다.
 - 두 문자가 같으면 그 칸에는 문자가 계속 그려지고, 문자가 다르면 다시 가려진다. 문자의 색상 변경한다.
 - 특정 횟수만큼 진행할 수 있게 하고, 점수를 배점하여 출력한다. (횟수, 점수 배점은 각자 정해보기)
 - 명령어: r – 게임을 리셋하고 다시 시작한다.
- 보드칸 선택 방법
 - 행렬의 번호를 입력 받아 선택하기
 - 1행1열이 기본으로 선택되어 있고, 특정 키보드를 입력 받아 칸을 이동한다.

	a	b	c	d
1	*	*	*	*
2	*	*	*	*
3	*	*	*	*
4	*	*	*	*

input card 1: a1
input card 2: c2

	a	b	c	d
1	A	*	*	*
2	*	*	C	*
3	*	*	*	*
4	*	*	*	*

	a	b	c	d
1	*	*	*	*
2	*	*	*	*
3	*	*	*	*
4	*	*	*	*

input card 1: a1
input card 2: b4

	a	b	c	d
1	A	*	*	*
2	*	*	*	*
3	*	*	*	*
4	*	A	*	*

5. 카드 짝 맞추기 게임 만들기

- 콘솔에서 색 출력하기

```
#include <windows.h>
```

```
SetConsoleTextAttribute (GetStdHandle(STD_OUTPUT_HANDLE), attribute_color); //--attribute_color; WORD 타입으로 색상을 나타내는 숫자
```

- 콘솔에서 제공해주는 색상



- 사용 예)

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <windows.h>
```

```
int main()
```

```
{
```

```
    int i;
```

```
    for (i = 0; i < 16; i++)
```

```
    {
```

```
        unsigned short text = i;
```

```
        SetConsoleTextAttribute (GetStdHandle(STD_OUTPUT_HANDLE), text);
```

```
        printf("setColor (%d)\n", text);
```

```
    }
```

```
    return 0;
```

```
}
```

```
setColor (1)
setColor (2)
setColor (3)
setColor (4)
setColor (5)
setColor (6)
setColor (7)
setColor (8)
setColor (9)
setColor (10)
setColor (11)
setColor (12)
setColor (13)
setColor (14)
setColor (15)
```

<결과 화면>

6. 움직이는 도형 그리기

- 화면에 30x30크기의 보드를 그린다 (30x30의 각 칸에 .을 찍어 보드를 표현한다. 보드 크기는 변경 가능함). 사용자에게 사각형의 두 개의 좌표값 (x1, y1) (x2, y2) 값을 입력받는다. 명령어에 따라 도형을 이동, 확대/축소, 한 좌표 측으로의 확대/축소를 그린다.
- 입력 명령어 (명령어 키보드 변경 가능):
 - x/X: x축 우측/좌측으로 한 칸 이동 (가장자리에 도달해도 계속 이동하고, 보드 밖으로 나가는 부분은 반대편에 나타난다.)
 - y/Y: y축 아래쪽/위쪽으로 한 칸 이동 (가장자리에 도달해도 계속 이동하고, 보드 밖으로 나가는 부분은 반대편에 나타난다.)
 - s/S: x와 y 모두 한 칸 축소/확대 (범위를 넘어가는 축은 변경 안됨)
 - i/j: x축 한 칸 확대/축소
 - k/l: y축 한 칸 확대/축소
 - R: 모든 것을 리셋하고 다시 좌표값을 입력받는다.

출력 예)

(샘플로 10x10 보드로 나타냈음)

input coord value: 2 2 5 6 //--- (2, 2) (5, 6) 좌표값으로 이루어진 사각형)

6. 움직이는 도형 그리기

input order: x //--> 우측으로 한 칸 이동

.
.	.	0	0	0	0	.	.
.	.	0	0	0	0	.	.
.	.	0	0	0	0	.	.
.	.	0	0	0	0	.	.
.	.	0	0	0	0	.	.
.
.
.
.

input order: s //-- 한 칸 축소

[illegible]

7. 경로 만들고 경로 따라 도형 이동하기

- 50x50 크기의 2차원 배열을 만든다. 좌측 상단에서 우측 하단으로 가는 한 줄의 경로를 만든 후 화면에 출력한다.
 - 0은 길, 1은 벽 (혹은 반대로)으로 표시한다.
- 이때, 경로는 4방향인 좌/우/상/하로 연결되고, 다음의 조건에 맞게 경로를 만든다.
 - 조건 1) 경로는 한쪽 방향으로 5칸 이상 계속 이동할 수 없다.
 - 조건 2) 경로는 좌우상하로 최소한 1번 이상 방향을 전환한 적이 있어야 한다.
- 키보드 명령에 의해 경로를 따라 이동하는 도형을 그린다. (키보드 명령어는 다른 것을 사용해도 무관함)
 - enter키: 새로운 경로를 만든다.
 - r: 임의의 객체 (문자)가 경로 시작칸에 나타난다. (다른 색상으로 그리기)
 - +/-: 나타난 객체를 경로 따라 앞으로/뒤로 이동한다.
 - q: 프로그램 종료

출력 예) // 경로 생성 (샘플로 10X10으로 나타냈음. 0: 경로, 1: 벽) // r 키를 눌러 객체 생성

```
0 0 1 1 1 1 1 1 1 1
1 0 0 0 1 1 1 1 1 1
1 1 1 0 1 0 0 0 1 1
1 1 1 0 1 0 1 0 1 1
1 1 1 0 0 0 1 0 1 1
1 1 1 1 1 1 1 0 1 1
1 1 1 1 1 1 0 0 1 1
1 1 1 1 0 0 0 1 1 1
1 1 1 1 0 1 1 1 1 1
1 1 1 1 0 0 0 0 0 0
```

```
* 0 1 1 1 1 1 1 1 1 1
1 0 0 0 1 1 1 1 1 1
1 1 1 0 1 0 0 0 1 1
1 1 1 0 1 0 1 0 1 1
1 1 1 0 0 0 1 0 1 1
1 1 1 1 1 1 1 0 1 1
1 1 1 1 1 1 0 0 1 1
1 1 1 1 0 0 0 1 1 1
1 1 1 1 0 1 1 1 1 1
1 1 1 1 0 0 0 0 0 0
```

// + 키를 눌러 앞으로 이동

```
0 * 1 1 1 1 1 1 1 1 1 1
1 0 0 0 1 1 1 1 1 1
1 1 1 0 1 0 0 0 1 1
1 1 1 0 1 0 1 0 1 1
1 1 1 0 0 0 1 0 1 1
1 1 1 1 1 1 1 0 1 1
1 1 1 1 1 1 0 0 1 1
1 1 1 1 0 0 0 1 1 1
1 1 1 1 0 1 1 1 1 1
1 1 1 1 0 0 0 0 0 0
```