Warming-up 프로그램 - 2

2022-2 컴퓨터 그래픽스

4. 사각형과 사각형 간의 충돌 체크

- 비교할 사각형과 사각형의 좌표값을 [800, 600] 사이의 랜덤한 값으로 정한다.
  - 각각의 도형은 두 개의 좌표값으로 표시하고, 각 좌표값은 x와 y의 두 개의 정수값으로 표시한다.
  - 즉, 사각형의 (x1, y1) (x2, y2) 값과 두 번째 사각형의 (x3, y3) (x4, y4) 값을 랜덤하게 설정한다.
- 키보드 명령을 사용하여 도형의 좌표값을 이동시킨 후, 두 도형의 충돌 체크 여부를 출력한다.
  - 키보드 명령어 wasd: 사각형을 위/좌/하/우 측으로 일정 길이만큼 이동 (두 사각형에 번갈아서 적용됨)
    - 이동 값은 구현하는 사람이 설정함: 10픽셀 이상으로 설정하도록 함
  - 도형의 좌표는 범위를 벗어나지 않는다. 범위를 벗어나면 에러메시지를 출력하고 이동하지 않는다.

예) (이동 Command: wasd)

Input Shape Coordinates value:

Rect 1: (100, 100) (300, 400)

Rect 2: (350, 400) (500, 500)

Command: d //--- 1번 사각형을 오른쪽으로 이동시킨다: x 좌표값을 50 이동

Rect 1: (150, 100) (350, 400)

Rect 2: (350, 410) (500, 500)

Command: k //--- 2번 사각형을 아래쪽으로 이동시킨다: y 좌표값을 -60 이동

Rect 1: (150, 100) (350, 400)

Rect 2: (350, 350) (500, 440)

Command: d //--- 1번 사각형을 오른쪽으로 이동시킨다: x 좌표값을 50 이동

Rect 1: (200, 100) (400, 400)

Rect 2: (350, 350) (500, 440) Rectangle 1 & Rectangle 2 collide!!

5. 저장 리스트 만들기 (구조체 데이터 사용)

- 점 (x, y, z) 데이터 값을 저장하는 리스트를 만든다. (점 데이터는 각각 정수이고, 구조체를 사용하도록 한다.)
- 최대 10개의 점 데이터를 저장하도록 한다.
- 리스트에 데이터를 입력하거나 삭제하고 출력하는 명령어를 실행한다.
  - 각 명령어를 입력 받으면 결과 리스트를 다음페이지의 그림과 같이 항상 10개의 항목을 가진 리스트로 (인덱스와 데이터 값) 출력한다.

- 구현 함수 프로토타입과 명령어:
  - + x y z: 리스트의 맨 위에 입력 (x, y, z: 숫자)
  - -: 리스트의 맨 위에서 삭제한다.
  - e x y z: 리스트의 맨 아래에 입력 (명령어 +와 반대의 위치, 리스트에 저장된 데이터값이 위로 올라간다.)
  - d: 리스트의 맨 아래에서 삭제한다. (리스트에서 삭제된 칸이 비어있다.)
  - l: 리스트에 저장된 점의 개수를 출력한다.
  - c: 리스트를 비운다. 리스트를 비운 후 다시 입력하면 0번부터 저장된다.
  - m: 원점에서 가장 먼 거리에 있는 점의 좌표값을 출력한다.
  - n: 원점에서 가장 가까운 거리에 있는 점의 좌표값을 출력한다.
  - s: 원점과의 거리를 정렬하여 오름차순 (또는 내림차순)으로 정렬하여 출력한다. 인덱스 0번부터 빈 칸없이 저장하여 출력한다. 다시 s를 누르면 원래의 인덱스 위치에 출력한다.
  - q: 프로그램을 종료한다.

@@ 리스트에서 맨 위(인덱스 9번)까지 차고 아래칸(인덱스 0번)이 비어있으면 다음 데이터 입력할 때는 0번에 입력된다. 즉, 10개의 칸을 다 채울 수 있어야 함.

5. 저장 리스트 만들기 (구조체 데이터 사용)

실행 예)

+ 0 1 0	→ 리스트 1번 출력, (리스트 각 항목 옆에 length 출력)
+ 0 1 1	→ 리스트 2번 출력, (리스트 각 항목 옆에 length 출력)
-	→ 리스트 3번 출력, (리스트 각 항목 옆에 length 출력)
e 1 1 1	→ 리스트 4번 출력, (리스트 각 항목 옆에 length 출력)
e 1 0 1	→ 리스트 5번 출력, (리스트 각 항목 옆에 length 출력)
+ 1 1 0	→ 리스트 6번 출력, (리스트 각 항목 옆에 length 출력)
d	→ 리스트 7번 출력, (리스트 각 항목 옆에 length 출력)
s	→ 리스트 8번 출력, (리스트 각 항목 옆에 length 출력)
m	→ (1, 1, 1) 출력
n	→ (0, 1, 0) 출력
l	→ 리스트 길이: 3

1

9	
8	
7	
6	
5	
4	
3	
2	
1	
0	0 1 0

2

9	
8	
7	
6	
5	
4	
3	
2	
1	0 1 1
0	0 1 0

3

9	
8	
7	
6	
5	
4	
3	
2	
1	
0	0 1 0

4

9	
8	
7	
6	
5	
4	
3	
2	
1	0 1 0
0	1 1 1

5

9	
8	
7	
6	
5	
4	
3	
2	0 1 0
1	1 1 1
0	1 0 1

6

9	
8	
7	
6	
5	
4	
3	1 1 0
2	0 1 0
1	1 1 1
0	1 0 1

7

9	
8	
7	
6	
5	
4	
3	1 1 0
2	0 1 0
1	1 1 1
0	

8

9	
8	
7	
6	
5	
4	
3	
2	1 1 1
1	1 1 0
0	0 1 0

6. 경로 만들기

- 30x30 크기의 2차원 평면을 만든다.
- 시작점 (0, 0)에서 대각선에 놓여있는 끝점 (29, 29)까지 랜덤한 경로를 설정하여 출력한다.
  - 길이 아닌 곳은 0으로 표시하고, 경로는 시작점을 1로 출력하고 연결되는 경로를 1보다 큰 정수를 순서대로 출력한다.
  - 이때, 길은 4방향인 좌/우/상/하로 연결되고, 다음의 조건에 맞게 길을 만든다.

    조건 1) 경로는 한 방향으로 최대 8칸 계속 이동할 수 있다.

    조건 2) 경로는 좌우상하로 최소한 1번 이상 방향을 전환한 적이 있어야 한다.

    조건 3) 한번 지나갔던 길을 다시 지나갈 수 있다. 한번 이상 거친 경로는 최종적으로 지나가는 순서의 정수값 (1, 2, 3, ...)으로 출력한다.

    출력할 때 포맷에 맞춰서 출력한다.
  - 명령어: r – 경로를 다시 만든다. r/l – 만든 경로를 모두 한 칸 우측/좌측으로 옮긴다. q – 프로그램을 종료한다.

예) (샘플로 10X10으로 나타냈음)

1	**2**	0	0	0	0	0	0	0	0
0	**3**	**4**	**5**	0	0	0	0	0	0
0	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	0
0	**27**	0	**7**	0	0	0	0	**18**	0
0	**28**	0	**8**	**9**	**10**	0	0	**17**	0
0	**29**	0	0	0	**11**	0	0	*16*	0
0	**30**	**31**	0	0	**12**	**13**	**14**	**15**	0
0	0	**32**	0	0	0	0	0	0	0
0	0	**33**	0	0	0	0	**40**	**41**	**42**
0	0	**34**	**35**	**36**	**37**	**38**	**39**	0	**43**

명령어: r 입력하여 모든칸이 한 칸 우측으로 이동

0	**1**	**2**	0	0	0	0	0	0	0
0	0	**3**	**4**	**5**	0	0	0	0	0
0	0	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**
0	0	**27**	0	**7**	0	0	0	0	**18**
0	0	**28**	0	**8**	**9**	**10**	0	0	**17**
0	0	**29**	0	0	0	**11**	0	0	*16*
0	0	**30**	**31**	0	0	**12**	**13**	**14**	**15**
0	0	0	**32**	0	0	0	0	0	0
42	0	0	**33**	0	0	0	0	**40**	**41**
43	0	0	**34**	**35**	**36**	**37**	**38**	**39**	0

7. 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 위에 a b c d, 좌측에 1 2 3 4를 표시하여 칸의 위치를 알 수 있게 한다.
  - 2개의 격자를 선택한다. 선택된 보드 칸에 o을 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 입력받아 선택하기
  - 1행1열이 기본으로 선택되어 있고, 특정 키보드를 입력받아 칸을 이동한다.

예)

	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	*	*

7. 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;
}
```