

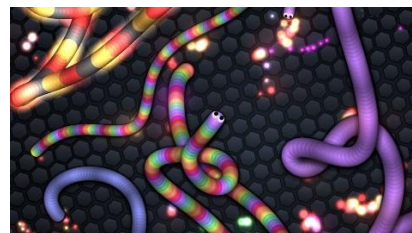
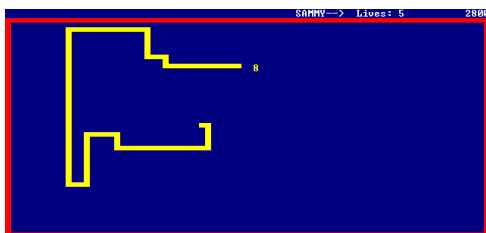
MFC 및 Linked List를 이용한 뱀 게임 구현

뱀 게임이란 1970 년대에 처음 나온 게임 장르로 아직까지도 수많은 변종들이 나오고 있는 인기 장르이다. 본 프로젝트에서는 MFC 와 자료구조의 일종인 Linked List를 이용하여 뱀 게임을 구현한다. MFC 를 이용함으로써 GUI 프로그래밍의 기초를 기르며, 뱀 게임의 기본적인 알고리즘 구현을 위하여 LinkedList 를 사용함으로써 자료 구조를 실제 문제에 적용하며 문제 해결력을 기른다.

일반적인 뱀 게임의 규칙은 다음과 같다.

규칙

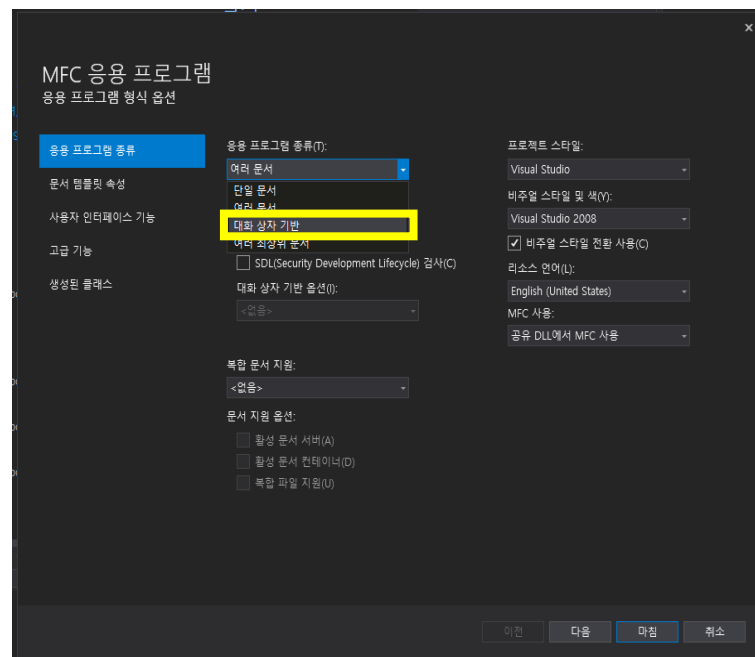
1. 사방이 막힌 네모난 공간에 플레이어가 조종하는 뱀 한 마리가 놓인다.
2. 뱀은 현재 머리가 향하고 있는 방향으로 멈추지 않고 이동한다. 플레이어의 조작으로는 머리가 진행하는 방향만 바꿀 수 있다.
3. 뱀이 벽이나 자신의 몸 일부에 머리를 부딪히면 죽는다.
4. 목적은 사방이 막힌 네모난 공간에 놓여있는 먹이를 모두 먹는 것이다. 한번에 모든 먹이가 놓이지 않고 하나의 먹이가 놓이며 해당 먹이를 먹을 경우 다른 위치에 먹이가 놓인다.
5. 먹이를 먹을 때 마다 뱀의 길이가 길어진다.
6. 먹이를 모두 먹어 뱀이 화면 전체를 차지하게 될 경우 게임은 끝난다.



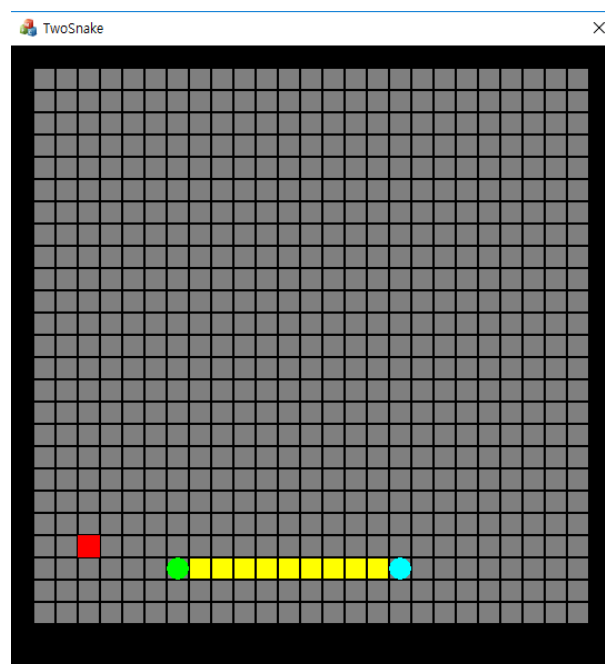
<그림 1. 다양한 뱀 게임의 종류 예시>

▶ Program implementation

프로그램은 MFC에서 지원하는 응용 프로그램 종류 중 다이얼로그(대화 상자) 기반으로 작성한다. 본 프로젝트에서 구현해야 할 프로그램의 형태는 그림 3. 과 같다.



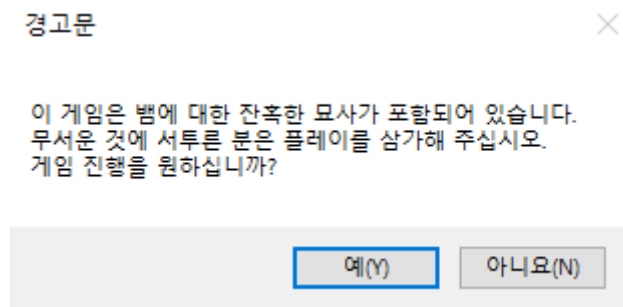
<그림 2. MFC 응용 프로그램 종류>



<그림 3. 최종 프로그램 형태 예시>

처음 프로그램 실행 시 게임 진행 여부에 대한 의사를 묻는 메시지 박스가 출력 된다.

메시지 박스의 내용과 형태는 그림 4. 와 같다. 사용자가 “아니요” 를 누르는 경우 즉시 프로그램을 종료하고 “예” 를 누르는 경우 게임이 실행된다.

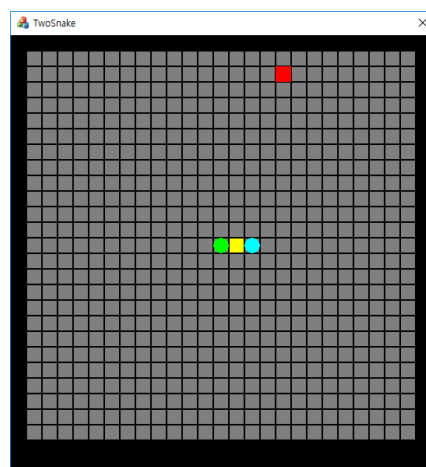


<그림 4. 프로그램 실행 시 출력되는 메시지 박스>

게임 실행 시 처음 나타나는 화면은 그림 5. 와 같다.

회색 사각형은 뱀이 움직일 수 있는 영역으로써, 영역의 범위는 25 x 25 (UI 상의 절대적인 크기가 아님을 유의) 에 해당하며 해당 영역을 벗어나는 경우 벽에 부딪힌 것으로 간주한다. (본 프로젝트에서는 이 영역을 map 이라고 칭함.)

초록색 원은 뱀의 머리이고 노란색 사각형은 뱀의 몸이며 청록색 원은 뱀의 꼬리이다. 게임 실행 시 뱀의 머리는 map 의 정중앙에 위치 하며 몸은 머리에서 우측으로 한 칸에 떨어져서 위치하고 꼬리는 머리에서 우측으로 두 칸 떨어져서 위치한다. 이 때, 뱀의 머리, 몸, 꼬리는 Linked List 구조로 연결되어 있어야한다. 게임 실행 시 뱀의 움직이는 방향은 왼쪽으로 설정되며, 사용자가 키를 입력하는 경우 방향이 변한다. 움직이는 방향을 설정하는 키는 ‘W’, ‘A’, ‘S’, ‘D’ (소문자도 가능해야 함.) 로 ‘W’를 입력할 경우 위쪽, ‘A’를 입력할 경우 왼쪽, ‘S’를 입력할 경우 아래쪽,



<그림 5. 게임 실행 시 출력되는 초기 화면>

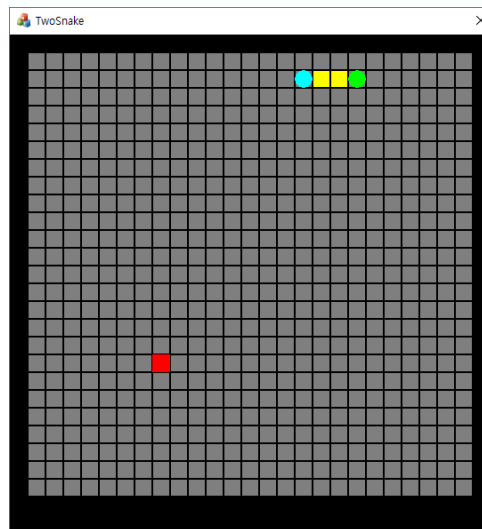
‘D’를 입력할 경우 오른쪽으로 방향이 전환된다. 이 때, 현재 움직이는 방향에서 반대 방향으로의 방향전환은 불가능하다.

예시)

1. 뱀이 위쪽 방향으로 움직이고 있었을 경우, 아래쪽 방향으로는 방향전환이 불가능함.
2. 뱀이 아래쪽 방향으로 움직이고 있었을 경우, 위쪽 방향으로는 방향전환이 불가능함.
3. 뱀이 왼쪽 방향으로 움직이고 있었을 경우, 오른쪽 방향으로는 방향전환이 불가능함.
4. 뱀이 오른쪽 방향으로 움직이고 있었을 경우, 왼쪽 방향으로는 방향전환이 불가능함.

뱀의 움직임을 화면에 출력할 때 업데이트 주기는 55ms 로 제한한다. 이는 MFC에서 제공하는 Timer 함수를 사용하여 구현하도록 한다.

빨간색 사각형은 뱀의 먹이이며, 뱀이 이를 먹을 경우(뱀의 머리가 해당 영역에 도달하는 경우) 뱀의 몸(노란색 사각형)의 길이가 1 증가하게 된다. 이 후 map 의 다른 위치에 먹이가 랜덤하게 놓이게 된다. 그림 6. 은 뱀이 첫 번째 먹이를 먹은 후의 상황에 대한 예시로 뱀이 먹이를 먹음으로써 몸의 길이가 1 증가되었으며, 또 다른 위치에 먹이가 놓임을 확인할 수 있다.



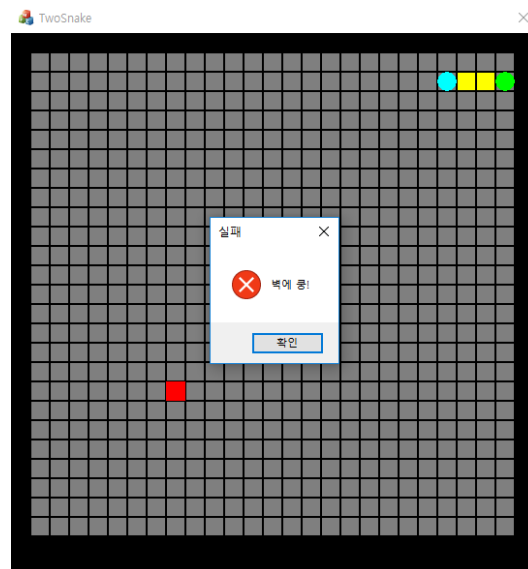
<그림 6. 첫 번째 먹이를 먹은 이후의 상황 예시>

게임이 종료되는 경우는 2 가지가 있다.

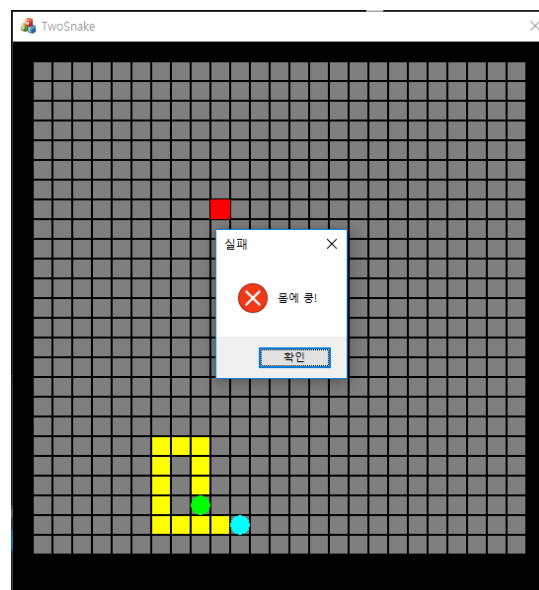
첫 번째는 그림 7. 과 같이 뱀의 머리가 벽에 부딪히는 상황으로 이 경우에 그림 7. 과 같은 메시지 박스(제목: 실패, 내용: 벽에 쿵!)를 출력하며 게임을 종료한다.

두 번째는 그림 8. 과 같이 뱀의 머리가 자신의 몸에 부딪히는 상황으로 이 경우에 그림 8. 과 같은 메시지 박스(제목: 실패, 내용: 몸에 쿵!)를 출력하며 게임을 종료한다.

*게임의 성공 조건은 뱀의 머리와 몸, 꼬리가 모든 map 을 차지하는 경우로써 성공 조건의 경우
평가시 확인이 어렵기 때문에 프로젝트 성적 평가 항목에서는 제외된다.



<그림 7. 뱀의 머리가 벽에 부딪힌 경우>



<그림 8. 뱀의 머리가 자신의 몸에 부딪힌 경우>

UI 의 세부적인 크기 제한은 없다. 뱀이 갈 수 있는 영역의 범위가 25x25 인 점과 뱀의 부위별
색상 및 모양, 먹이의 색상 및 모양이 프로젝트에서 제안된 바와 같아야 하며 경우에 따라 출력
되는 메시지 박스의 내용은 일치해야 한다.

게임의 동작 영상에 대한 예시는 아래의 링크에서 확인할 수 있습니다.

<https://youtu.be/uvydSx92Cy4>

▶제출 기한 및 제출 방법

- 2019년 6월 14일 23:59 까지 제출(U-campus 서버시간기준)

- 채점시 코드를 수정해야 하는 일이 없도록 할 것

- Software와 보고서 파일을 함께 제출

- 보고서는 아래의 내용을 포함한다.

Introduction/ Algorithm 및 Flowchart / / 결과 화면/고찰

소스코드는 보고서에 포함하지 않음

반드시 영문으로 각 항목을 모두 포함하여 작성

표지를 제외하고 20 페이지 이내로 작성

- 설계반_실습반_학번_한글이름_verX.zip

1_A_학번_이름_ver1.zip : 설계수강, 실습A반

0_C_학번_이름_ver1.zip :설계미수강,실습C반

1_0_학번_이름_ver1.zip : 설계수강, 실습미수강

설계만 듣는 경우 설계 과목에만 제출

설계, 실습 듣는 경우 실습 과목에만 제출

실습만 듣는 경우 실습 과목에 제출

◆ 프로젝트의 관한 질문사항은 아래 조교에게 문의

