class

1. map
2. snake

files

1. main
2. LOC.h
3. map.cpp, map.h
4. snake.cpp, snake.h

Lv.1

LOC.h:

opeartor[](int idx): - LOC[1]은 y값, LOC[0]은 x값으로 접근

map class:

member variables:

size : 최소 21x 21

LOC \*imwall - 테두리에 있지 않은 immune wall들의 좌표 모임

LOC \*wall - 테두리에 있지 않은 wall들의 좌표 모임

const int IMWALL - immune wall의 칼라 pair 해당 정수

const int WALL - wall의 칼라 pair 해당 정수

member functions:

main.cpp:

displayUpdate(WINDOW\* display, Map& map):

화면 전체를 업데이트 한다 - 배경, 벽돌, 뱀, 아이템, 게이트의 좌표들을 다 다시 printw한다. - 속도면에서 문제가 없는가? 그냥 변한 것만 하면 되는 거 아닌가?

init\_config(Map& map):

화면의 초기설정을 한다, 서브 윈도우 설정, color\_pair 설정 등

Lv.2

todo:

* snake 이동:
  + head 방향으로 일정시간(0.5초) 틱
  + 방향키로(혹은 wasd) 누르면 head의 진행 방향 바꿈
  + tail 방향으로는 이동방향 바꿀수 없음 -> 실패
* 실패조건:
  + body 통과x
  + wall 통과x
  + tail 방향으로는 이동방향 바꿀수 없음 -> 실패

화면이 업데이트 되는 동안 입력을 받야아 하므로, therad가 필요함.

getch()는 입력 받을 때까지 기다리므로,,

Snake class:

memeber variables:

LOC head - head의 좌표

vector<LOC> body - body의 좌표들 모임

char opposite - head이동 방향과 반대의 키

LOC inc - head의 증/감 값

int life - 뱀의 목숨, (뱀의 길이 - 2), 0이되면 죽고, game failed 되면 역시 0이된다.

memeber functions:

// 생성자

Snake(LOC& head, int length)

// 실패 조건 검사

bool isDumpedBody(): head가 body와 겹쳐지는지 확인, -> 움직이는것은 head 하나이므로 이것만 판단해도 충분

bool isDumpedWall(Map& map): head가 map의 wall혹은 imwall과 충돌하는지 확인

bool pressedOps(char key) - key가 opposite와 같으면 true, 아니면 false 반환

// 좌표값 이동

void move() -0.5초 마다 뱀의 좌표값을 옮기는 함수, body의 맨끝 원소를 pop하고, body에 head의 LOC을 push 한다. 그리고, head는 key에 따라 하나한 좌표 증가시킨다.

void setDirection - key를 입력받아 뱀의 머리 방향을 설정하는 함수.

main.cpp:

std::thread tick(&Snake::move) - key를 입력받고, 화면을 업데이트하느 ㄴ건, main thread가 하고 뱀이 0.5초 간격으로 좌표를 이동 한다.

displayUpdate(WINDOW\* display, Map& map, Snake& snake): display에 map, snake를 printw하고, touch, refresh한다.

LOC.h:

opeartor=(LOC& diff): LOC 객체의 값을 = 연산자로 바꾸기 위함.

operator==(LOC& ops): LOC이 서로 같은지 확인.

문제점 - thread는 클래스의 static함수나 전역함수 만 받을 수 있다… 그래서 다른 방법이 필요하다 .

<https://linux.die.net/man/3/nodelay> nodelay를 쓰자.