

Order Rush

게임 프로그래밍

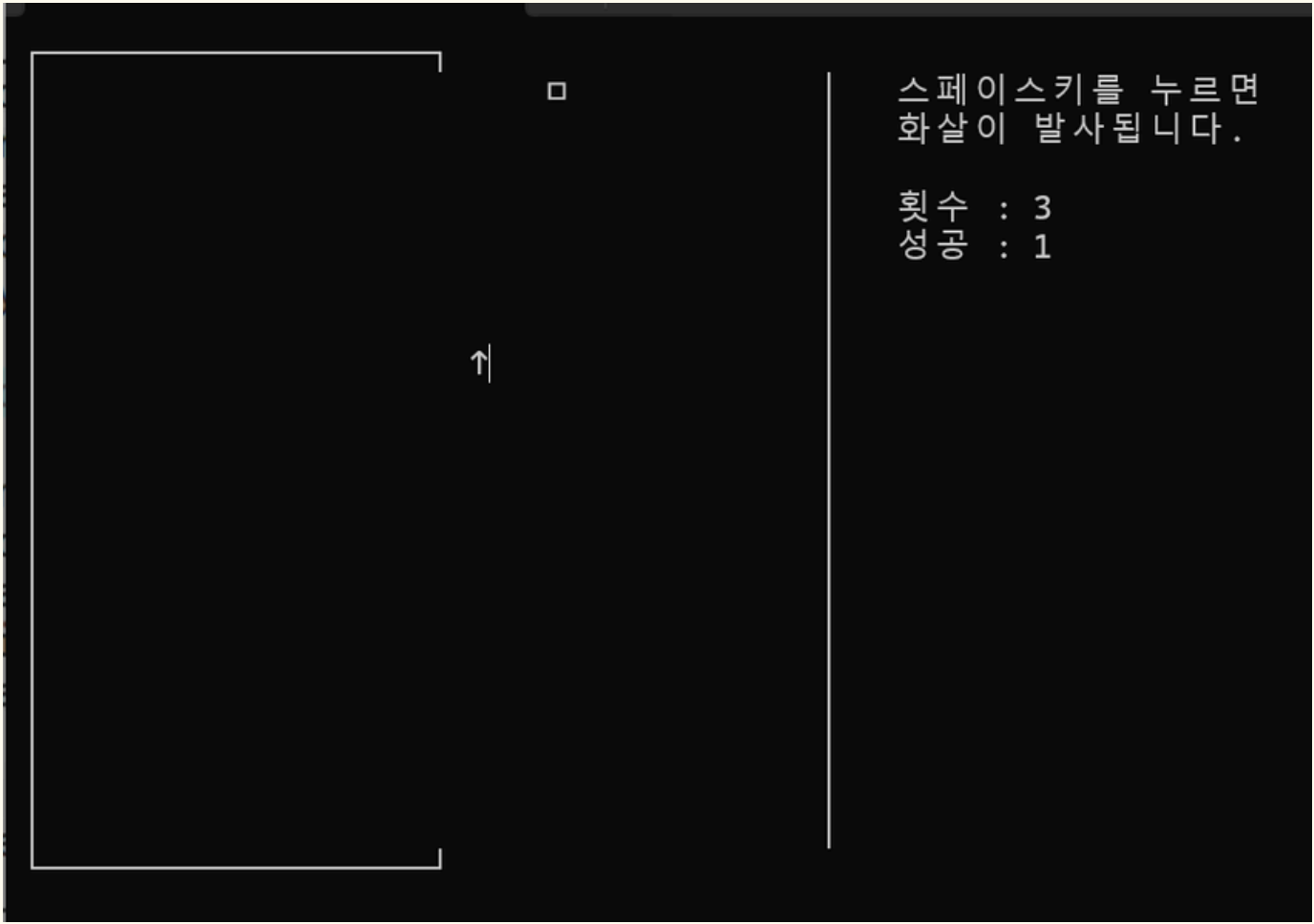
기존 게임의 문제점

8-5-1.cpp

말 타며 화살쏘기
말을 타고 이동하면서
목표물(□)을 맞추는 게임입니다.
화살은 스페이스키로 발사합니다.
아무키나 누르면 시작합니다. |

처음 화면

게임 제목, 간단한 설명이 적혀있지만 단조로움



게임 화면

불친절한 게임 화면, 불친절한 UI, 부족한 콘텐츠

새로운 게임 제작



Order Rush를 소개합니다!

이 게임은 총 1단계부터 61단계까지 있으며 8X8바둑판 위에서 게임이 시작됩니다.

N단계에 N+3개의 목표물이 있고

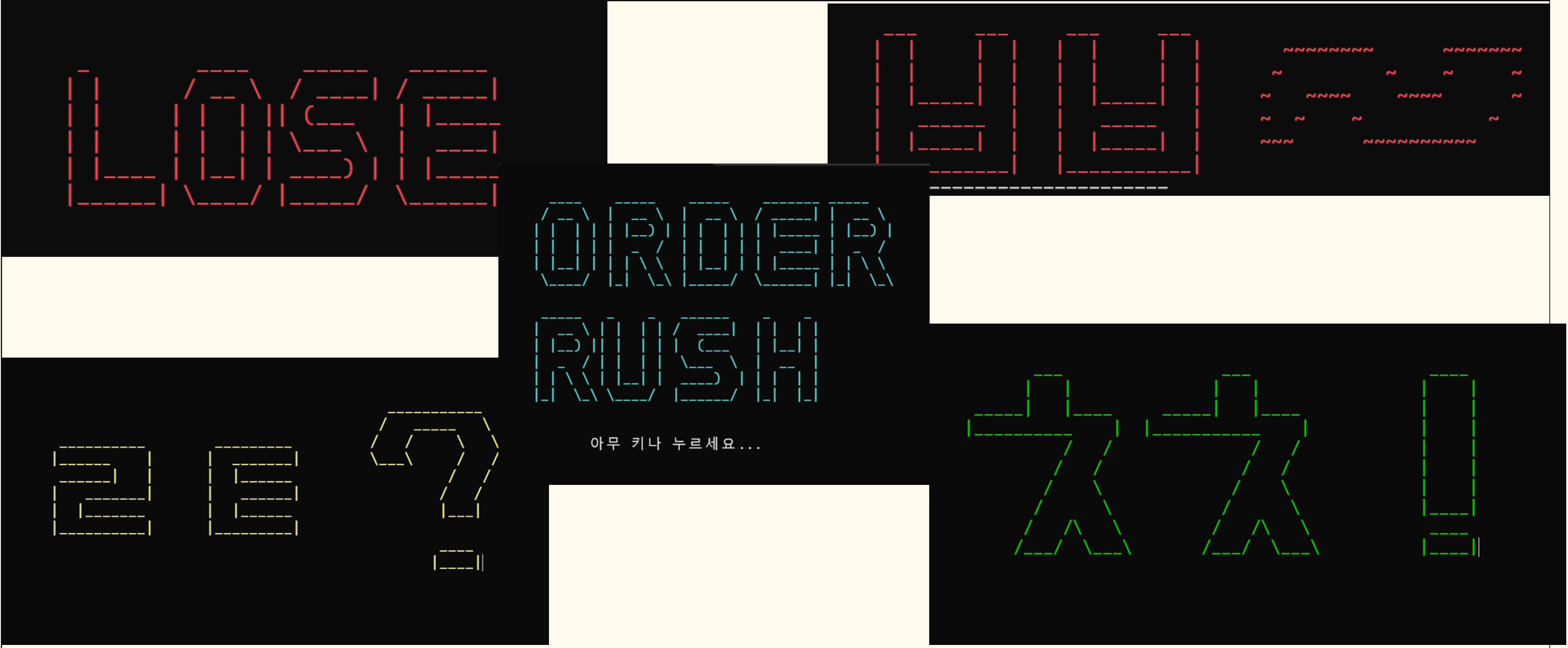
목표물에는 1~N+3의 숫자가 부여되어있습니다.

시간내에 1부터 순차적으로 모든 숫자를 쏘면 끝나는 게임입니다.

총알은 N+4개의 총알이 있으며 숫자를 다 없애기전에 시간이 다 가거나 총알을 다 쓰면 게임이 끝납니다.

목표를 쏘 플레이어는 방향키로 움직이고, 스페이스바를 이용해 총알을 쏠수있습니다.

아스키 아트



UI개선

기존UI

스페이스키를 누르면
화살이 발사됩니다.

횟수 : 1
성공 : 0

VS

바꾼UI

Stage : 11
Next : 1
bullet : 15
Time : 32.9 s

Move: LEFT/RIGHT
Fire: SPACE
Quit: Q

컨텐츠 개선

1부터 순서대로 쏘면 됩니다!

오른쪽키, 왼쪽키를 사용해 움직일 수 있습니다.
ENTER를 누르면 시작합니다.

종료를 원하시면 **esc**를 누르시오.

Select Stage (1~61)

Stage: 1|

38	32	34	39	36	35	37	33
27	28	25	26	24	30	29	31
19	16	22	20	17	18	21	23
12	13		14	15	9	11	10
2	4	5	7	3	1	8	6

Stage : 36
Next : 1
bullet : 40
Time : 105.0 s

Move: 왼쪽키 / 오른쪽키
Fire: SPACE
Quit: Q

1 2 3 4 5 6 7 8

숫자 배치 알고리즘

```
/* ===== 배치 ===== */
void init_targets_monotone(int goals, int stage){
    for(int y=1;y<=H;y++) for(int x=1;x<=W;x++) board[y][x]=0;
    expectedNum=1;

    int prevY = H - (rand()%3); if(prevY<1) prevY = H;

    for(int num=1; num<=goals; ++num){
        int remaining = goals - (num - 1);
        int floorY = min_rows_needed(remaining);
        if(floorY > prevY) floorY = prevY;

        int yMin=floorY, yMax=prevY; if(yMax<yMin) yMax=yMin;
        int y = yMin + rand() % (yMax - yMin + 1);

        int x, tries=0;
        do{
            x = 1 + rand()%W;
            tries++;
            if(tries>64){
                if(y>yMin) y--;
                tries=0;
            }
        }while(board[y][x]!=0);

        board[y][x]=num;
        prevY=y;
    }

    playerX=1;
}
```

```
/* ===== 수용량 가드 ===== */
static int min_rows_needed(int remaining){
    return (remaining + W - 1) / W;
}
```

해당 배치는 init_targets_monotone()함수에서 실행되며 단조 하강(monotone decreasing)형태의 알고리즘 입니다.

숫자 배치 알고리즘은 y값(줄)을 먼저 정한 뒤, 해당 줄에서 x를 랜덤으로 돌려 빈칸을 찾아 숫자를 배치합니다. 1은 가장 큰 y(맨 아래줄)에 배치되고, 2는 그와 같은 줄이거나 더 위쪽 줄에 랜덤으로 배치됩니다.

이때 61단계의 경우 64개의 칸 모두 목표물이 나와야하기 때문에 수용량 가드가 필요합니다. 수용량 가드는 $64+7(\text{목표})/8(\text{최대행})$ 을 해서 나오는 값으로 y를 고정하기 때문에 1~8까지는 y = 8, 9~16까지는 y = 7....이렇게 64개의 칸을 채울 수 있습니다.

64	63	59	58	57	62	61	60
56	51	54	55	53	49	50	52
43	45	46	47	48	44	42	41
39	33	40	36	37	34	35	38
32	26	28	30	27	31	25	29
22	19	24	18	17	20	23	21
15	10	13	16	12	9	14	11
5	6	2	8	4	1	3	7
1	2	3	4	5	6	7	8

Q&A
