

REPORT

보고서 작성 서약서

1. 나는 타학생의 보고서를 복사(Copy)하지 않았습니다.
2. 나는 타학생의 보고서를 인터넷에서 다운로드 하여 대체하지 않았습니다.
3. 나는 타인에게 보고서 제출 전에 보고서를 보여주지 않았습니다.
4. 보고서 제출 기한을 준수하였습니다.

나는 보고서 작성시 위법 행위를 하지 않고,
성.균.인으로서 나의 명예를 지킬 것을 약속합니다.

과 목: 문제해결방법

담당교수: 서익성

학 과: 반도체시스템공학과

학 번: 2018314788

이 름: 오해성

제출일: 2020.06.13

서론

이 보고서에서는 프로그래밍 과제로 개발한 게임에 관해 다룹니다 .

본론에서는 게임의 플레이 적인 요소들을 먼저 설명한 후 세부 구현에 대하여 설명합니다.
전자에서는 게임의 전체적인 흐름과 규칙, 클리어 조건에 대해서만 설명하며 플레이 영상 링크를 포함할 것입니다.

후자에서는 크게 게임 인터페이스,플레이어,몬스터,탄환 이 네가지 항목에 대해서
어떻게 문제를 해결하고 어떠한 알고리즘을 사용하였는지에 대하여 다룰 것입니다.

결론에서는 수정하지 못한 버그들에 대해서 설명하며 게임을 어떻게 개선시킬 수 있을지에
대한 생각을 서술할 것입니다.

- 게임은 **c++**로 개발되었으며 **ncurses** 라이브러리의 도움을 받아 터미널에서 작동하도록 설계되었습니다.

본론

1. 게임 플레이에 대해

이 게임은 qix 류 게임 일명 땅따먹기 게임과 슈팅게임을 섞은 듯한 게임입니다.

게임을 시작하면 c 키를 눌러 코인을 넣고 게임을 시작하게 됩니다.

게임에는 플레이어와 3 종류의 몬스터 9 마리가 존재하는데 플레이어가 모든 몬스터를 잡으면 게임에서 승리하게 됩니다.

플레이어가 이동 시 남는 선으로 몬스터를 감싸게 되면 몬스터의 hp 가 감소하고 몬스터의 hp 가 0 이 되면 몬스터는 사라지게 됩니다.

몬스터의 hp 의 따라 몬스터의 색깔이 구별되며 이는 빨강 > 노랑 > 파랑 의 순으로 체력이 1 씩 증가합니다.

플레이어는 몬스터에게 닿거나 몬스터의 탄환에 맞게 되면 게임에서 패배하고 코인이 감소되어 코인이 0 이 되는 순간 코인을 더 넣어주어야 합니다.

몬스터와 몬스터의 탄환은 플레이어가 생성한 벽에 가로막히기 때문에 이를 적절히 이용하여 탄환과 몬스터를 피하면서 모든 몬스터를 잡아내야 합니다.

플레이어의 이동은 키보드로 제어하며 이는 방향을 제어하는 것이기 때문에 키보드 입력이 없어도 진행방향으로 계속 진행합니다.

게임에서 사용하는 키들은 코인 키(c)와 방향키와 엔터키가 전부입니다.

다만 과제 채점을 돕기 위한 무적키와 난이도 하향키가 존재합니다.

게임의 플로우를 나타내보면 다음과 같습니다.

게임 시작

```
while (1)
{
    코인 입력
    if (enter) break;
}
```

게임 진입

```
while (1)
{
    화면 초기화
    몬스터 초기화
    탄환 초기화
    벽 초기화

    플레이어 생성
    몬스터 생성
```

```

while (1)
{
    플레이어 이동
    if (벽으로 가봤을 때)
    {
        벽 이미지 변경
        사용되지 않은 벽 초기화
        화면 리프레쉬
        500ms 딜레이
        내부 몬스터 판단
        if(몬스터) 체력감소 체력 0시 몬스터 제거
        사용된 벽 초기화
    }

    몬스터 랜덤 이동

    몬스터 랜덤 탄환 발사

    탄환 이동 처리(플레이어와 닿을 시 패배 판정)

    벽에 닿은 탄환 제거

    플레이어 출력
    몬스터 출력
    탄환 출력
    화면 리프레쉬

    몬스터 플레이어 닿을 시 패배 판정 처리

    if(몬스터 수 == 0) goto wingame;
}

if (coin)
{
    --coin;
    초기화
}
else
{
    9초간 코인 입력 다시받음
    if (coin) continue;
    break;
}
}

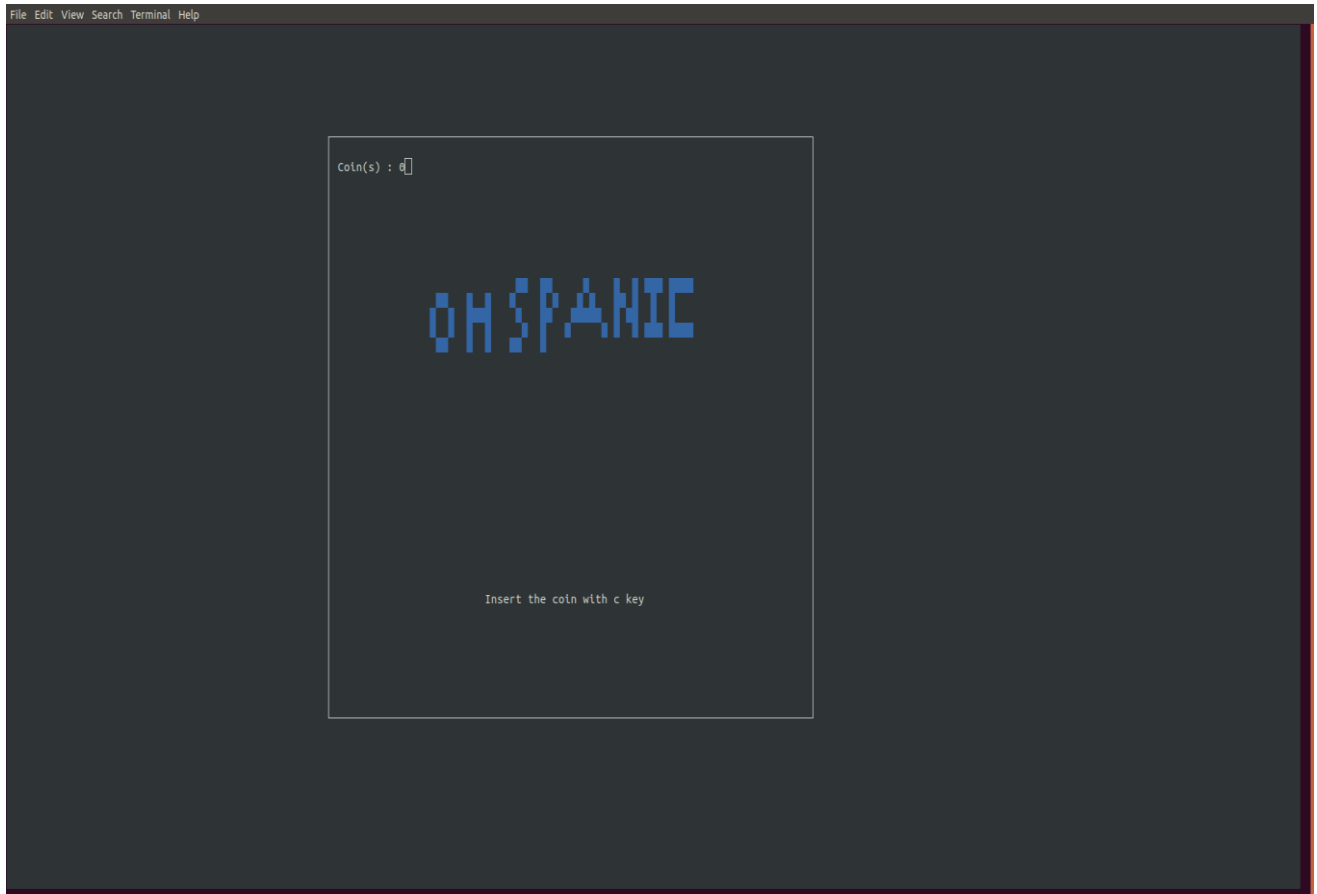
```

글로는 제대로 설명이 어려울 것 같아 직접 플레이 해보시거나 아래 링크에 영상을 보시는 것을 추천드립니다.

<https://www.youtube.com/watch?v=nizje-PgPsM>

2. 게임 세부 구현에 대해

2-1 유저 인터페이스의 구현



게임의 시작 시 마주하는 인터페이스입니다. 게임의 이름과 함께 c 키로 코인을 넣어주라는 문구가 나타납니다. c 키를 누르면 코인이 증가하며 코인이 있으면 엔터키를 눌러 게임을 진행하라는 문구가 나타납니다.

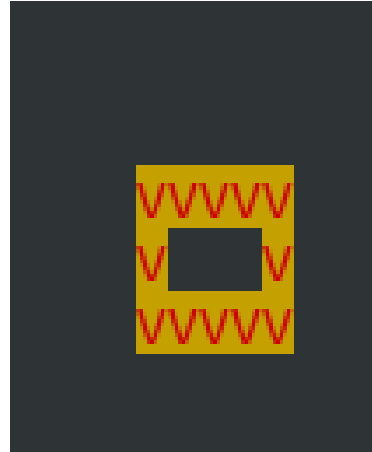
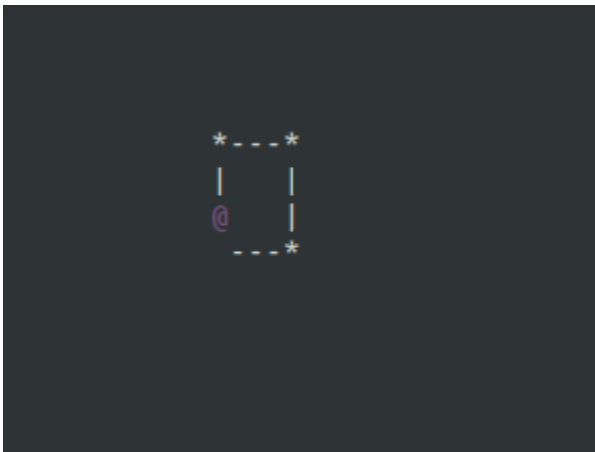
이는 사용자화면의 최대 y,x(세로축,가로축 값) 을 받아서 y/8,4/x 지점에 게임 창을 만든 후 내부에 코인을 나타내는 창과 제목을 나타내는 창 사용자에게 정보를 전달하는 창으로 나눠 총 네 개의 창을 사용하여 인터페이스를 구성하였고 제목은 배열로 만든 글자를 배경과 글자색이 모두 파란색인 color_pair 를 만들어 구현하였습니다.

위 정보를 띄운 후 while 루프를 구성해 코인을 원할 때까지 받은 뒤 엔터를 누르면 반복문을 빠져나와 게임으로 진행하게 만들었습니다.

후에 이 창에서 숫자 0 키를 누르면 무적이 되도록 구현하였고 7 키를 누르면 난이도를 낮추도록 구현하였습니다.

2-2 플레이어의 구현

클래스를 활용하여 플레이어를 객체로 만들고 이를 헤더에 구현한 뒤 플레이어 이동과 벽 등과 관련한 사항은 플레이어 헤더에 구현하였습니다.

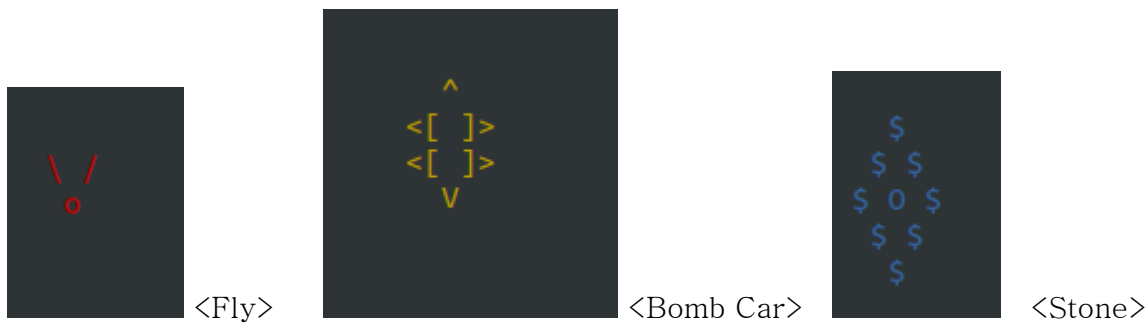


플레이어 이동과 관련해서는 먼저 `half_delay(n)`으로 설정 뒤 매 주기마다 사용자의 입력을 받으며 입력이 없을 시에는 계속하여 이동했던 방향으로 진행합니다. 입력이 있을 시에는 해당하는 방향으로 이동합니다. 다만 임의로 게임을 가속하는 것을 막기 위해 한방향으로 키를 연타하는 경우 이동이 진행되지 않도록 구현해 놓았고, 같은 지점을 두 번 방문할 때 벽이 닫힌 것으로 판정하기 때문에 방향을 반대로 돌려서 바로 전 지점으로 가는 것도 이동이 진행되지 않도록 구현하였습니다.

플레이어는 이동할 때 마다 지점에 벽을 생성하며 이 지점의 좌표를 스택에 넣어 벽이 닫혔을 시 실제로 닫힌 벽을 구성하는 벽이 어느 범위 인지 판단합니다.

세부적인 알고리즘은 이동 시마다 좌표를 스택에 넣고 방문 여부를 배열에 저장한 뒤 방문한 지점을 다시 방문하면 스택에서 그 지점을 만날 때 까지 빼고 여기 까지가 닫힌 벽을 구성하는 벽이라고 판단합니다. 나머지 벽들은 차례대로 초기화 해줍니다.

2-2 몬스터의 구현



3 개의 몬스터를 디자인하였으며 필요에 따라 몬스터를 새로 추가하는 것이 쉽도록 클래스로 구현하였고 공통된 정보로 작동합니다.

이 클래스들을 벡터 배열에 저장하여 몬스터 들의 동작을 관리합니다.

몬스터를 나타내는 정보는 몬스터의 좌표와 타입 체력 모양 등이 있으며 몬스터의 모양을 타입마다 배열로 구현해 놓았습니다.

Fly 는 대각선 방향으로 움직이며
Bomb Car 는 상하좌우로 움직이고
Stone 은 팔방향으로 움직입니다.

각각의 몬스터마다 공격방식이 다르며 이는 뒤에 탄환에 관한 설명에서 다루겠습니다.

몬스터는 랜덤하게 움직이며 seed 를 time 으로 받지만 time 이 대부분 겹치는 문제가 있어 임의의 상수 배열과 time 을 더한 값으로 난수를 생성해 이 난수로 이동 방향을 결정합니다. 이동할 때마다 att 라는 변수에 임의의 값들을 더해주면서 att 라는 변수가 해당 몬스터 타입의 설정된 값을 넘기면 attack 함수를 실행하여 탄환을 발사하는 방식입니다.

2-3 탄환의 구현

탄환은 몬스터 헤더의 attack 헤더에서 생성되는 구조체입니다.

각각의 탄환은 좌표와 방향 ctype 자료형의 캐릭터 그리고 속도 값을 가지고 있습니다.

추가로 Bomb Car 의 두번째 공격과 Stone 의 첫번째 공격은 탄환이 터지면서 또다른 탄환을 생성하므로 bomb 이라는 변수를 가지고 있습니다.

탄환도 몬스터와 같이 따로 벡터배열에 담아 탄환의 동작을 관리합니다.

Attack 함수를 보면 각각 몬스터의 타입마다 탄환 정보를 세팅하고 attv 벡터에 담는 것을 확인할 수 있습니다.

Bomb Car 와 Stone 은 각각 두가지 공격이 존재하는데 이 때문에 attack 함수가 실행될 때 마다 다른 동작이 진행됩니다.

Bomb Car 의 경우 가시를 날리는 듯한 공격이 t1 에 진행되며

t2,t3,t4 는 실제로 탄환을 발사하지 않고 몬스터의 이미지가 변형됩니다.

t5 가 됐을 때 일정 시간 후 사방으로 폭발하는 탄환을 발사하며 이후 다시 t1 동작으로 진행됩니다.

Stone 의 경우 t1 에 일정 시간 이후 같은 방향으로 일직선으로 퍼지는 탄환이 발사되며

t2 에 몬스터의 중앙부분이 '!' 로 바뀌며 정지한 뒤 화면에 레이저가 발사될 영역을 깜빡이는 '!'로 표시해줍니다.

t3 에 마치 레이저와 같이 파란색 선이 발사되고 다시 t1 동작으로 진행됩니다.

결론

게임 개발이나 프로젝트를 경험해 본 적이 처음이라 과제를 수행할 수 있을까 막막했지만 작은 부분부터 해결해 나가니 게임을 완성할 수 있었습니다.
게임을 개발하는 것은 재밌는 부분이 많아서 많은 시간을 투자하게 된 것 같습니다.

게임을 개발하면서 수많은 버그들이 있었는데 대부분은 픽스하였지만 일부러 남겨둔 버그나 해결하지 못 한 버그들도 존재합니다. 여기서는 간단히 두가지 버그에 대해서 서술하려고합니다.
첫번째로 벽짓살이라는 버그가 존재합니다.

이는 사용자 입력이 없을 시 진행 방향으로 움직일 때 벽에 닿게 되면 위치가 변하지 않는데 이때 플레이어의 위치가 벽으로 처리되어 무적상태가 되는 버그입니다.
이를 벽에 닿을 때 해당 지점을 초기화 시켜주면 되기 때문에 픽스하는건 어려운 일은 아니지만

게임이 조금 어렵기 때문에 이를 테크닉으로 이용해도 무방하다 판단하여 남겨두었습니다.

두번째로는 해결방법을 찾지 못해서 남겨둔 버그인데
플레이 영상을 보시거나 직접 플레이 해보시면 몬스터가 벽 내부에 있는데도
처리되지 않는 경우를 확인할 수 있습니다.
몬스터가 도형의 내부에 있는지 외부에 있는지는 다각형 내부 판단 알고리즘인 반직선을 그어
교점의 개수가 홀수인지 짝수인지 판단하여 홀수라면 내부에 있다고 판단하는데
벽이 몬스터의 축과 평행하게 존재하는 경우 이를 교점 1 개로 처리하여야 하기 때문에
이렇게 하게 되면 외부에 있는데도 몬스터가 내부로 판단되는 경우가 생겨서
몬스터에 위치에서 4 방향으로 반직선을 그은 뒤 각각 교점이 홀수일 경우를 카운트해
3 이상이면 내부로 판단하도록 구현하였습니다.
이렇게 하면 외부에 있는 몬스터가 도형 내부에 있다고 판단하는 오류는 발생하지 않으나
내부에 있는 몬스터가 판정을 벗어나는 경우가 생깁니다. 확률적으로 이러한 경우가 플레이를 진행
할 수 없을 정도로 심각하진 않고 해결 할 만한 방법이 떠오르지 않아서 처리하지 못한 버그입니다.

게임의 난이도가 고정적인 측면과 몬스터를 잡을 때 마다 난이도가 내려가는 점이 이 게임의 단점입니다.
게임을 스테이지로 구성하거나 시간 지연에 따라 몬스터가 생성되는 방식으로 구현하였으면 조금 더
흥미로운 게임이 되었으리라고 생각합니다. 몬스터와 플레이어는 객체로 구현하였고 여러가지 판정이
되도록이면 동일한 논리로 작동되게 개발하였기 때문에 몬스터를 새로 디자인해서 추가하거나

보스 스테이지를 만드는 등 게임이 개선될 수 있는 방법이 다양합니다. 또한 터미널에서 사운드를 구현하는 것이 찾아봤지만 어려워서 마지막에 시스템콜로 엔딩곡을 넣는 정도로만 구현하였는데 추후에는 이러한 효과를 조금 더 잘 활용할 수 있는 window 라이브러리를 사용해서 게임을 개발해보고자 합니다. 마지막 과제에서 정말 배운 것이 많고 처음으로 개발을 해 볼 수 있어서 즐거웠습니다.