오픈소스 프로그래밍

(GLSO215002)

팀 프로젝트 보고서

팀 7 – 블랙카우

김진규

김현학

윤성환

이수진

목차

1. 계획 및 구상

2. 분담

3. 개발

4. 게임 진행과정

5. 소감

**1. 계획 및 구상**

프로젝트가 공고된 이후 무엇을 만들지 회의한 결과 무엇이든 간에 게임을 만드는 편이 좋겠다고 의견이 모였다. 어떤 게임을 개발할지, 어떤 게임을 개발할 수 있는지에 대하여 3일정도 토론한 결과, 기존에 출시 되어있는 “포트리스”의 게임방식을 가진 게임을 개발하게 되었다.



<게임 포트리스의 한 장면>

유닉스 기반 터미널용 프로그램을 개발하는 것이 목적이기 때문에 GUI를 이용할 수 없기에 최대한 문자로 식별가능한 프로그램을 만들기로 결정하였다.

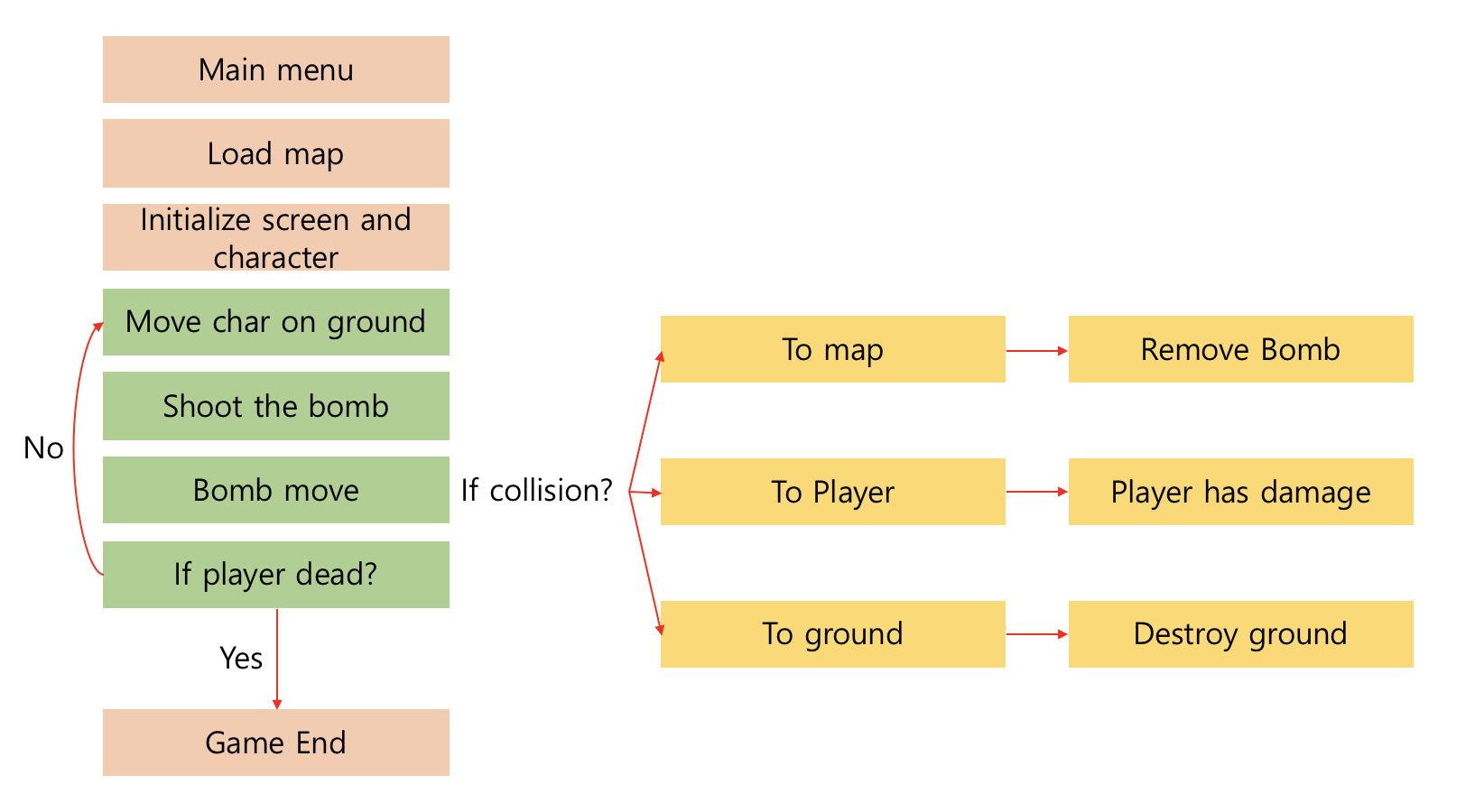
개발 기간은 시험기간을 고려하여 1달이 채 안될 것으로 예상하여 가급적 미리 개발을 끝내고자 하였다.

게임은 하나의 기기에서 2명의 플레이어가 번갈아 가며 플레이하는 방식이며, 적의 체력을 모두 깎거나 상대를 바닥 아래로 떨어뜨리면 승리한다. 바닥들은 모두 폭탄에 맞을 시 폭탄의 범위만큼 파괴된다.

이런 기본적인 요소를 고려하여 게임을 개발하였다.

작성하는 모든 코드는 GitHub를 통하여 관리하고 이는 전체 공개 되어있다.

GitHub Repository Link : <https://github.com/crihit/knu_blackcow>



<게임의 전반적인 시나리오를 그린 구상도>

**2. 분담**

큰 틀의 개발이 끝날 때 마다 앞으로 더 개발해야 할 것을 토론하고 그것을 분담하여 개발하였다.

분담하였다고는 하나, 큰 틀을 하나씩 개발해 나가면서 타인이 맡은 파트에서 자신이 필요하다고 여기거나 개선해 그것을 설명함으로써 프로젝트를 개선해 나갔기 때문에 절대적인 역할분담은 아니다.

아래의 내용은 전체 프로젝트를 진행하며 맡았던 역할에 대하여 기록한 것이다.

김진규(GitHub name : solorsystem1) - 포탄 포물선궤도와 캐릭터 등의 핵심 오브젝트의 구현. 카메라 효과 구현. 맵 디자인.

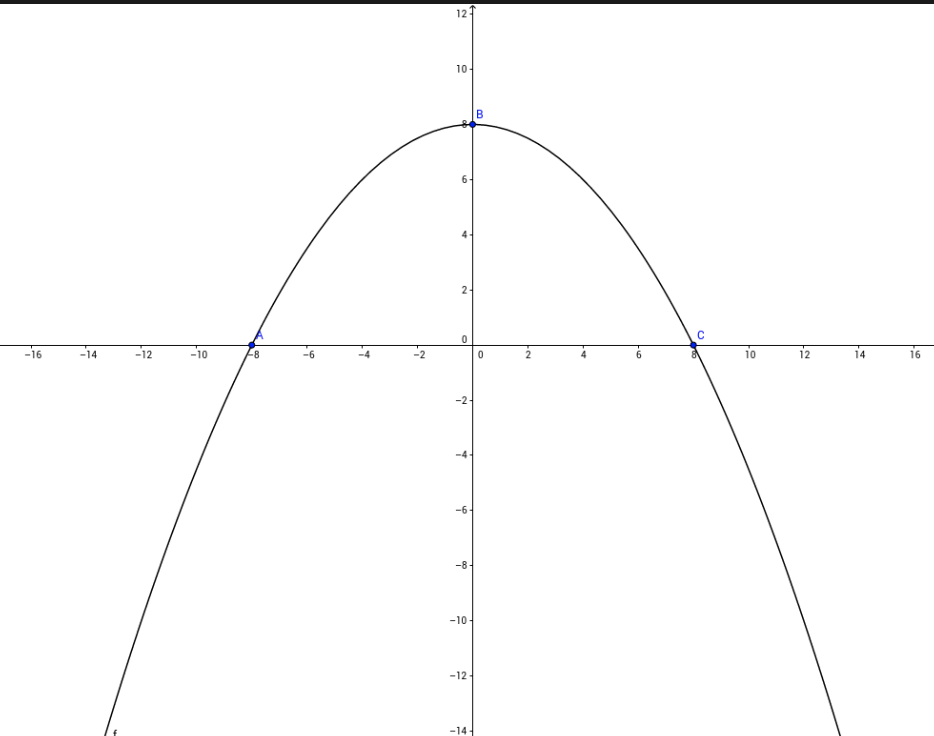
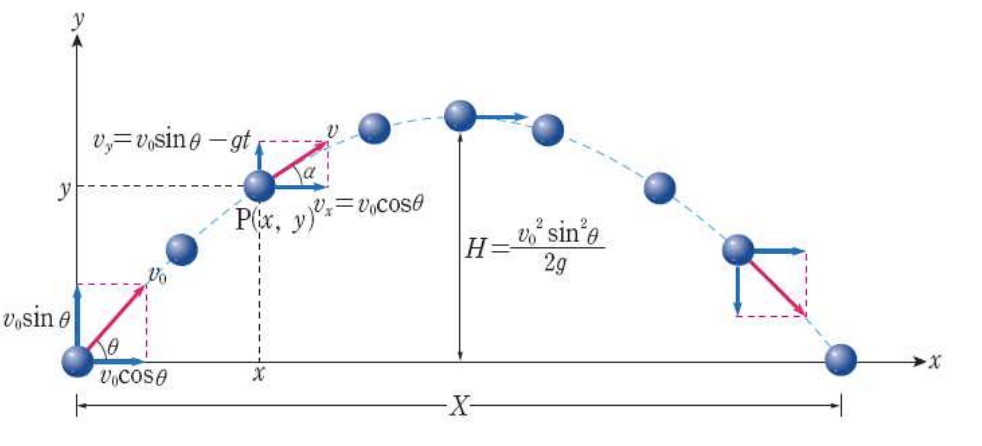
김현학(GitHub name : crihit) - 게임의 UI구현. 파일 입출력 관리.

윤성환(GitHub name : SUNGWHANYOON) - 포탄의 피격효과 구현. 맵 디자인.

이수진(GitHub name : LeesuJIN3419) - 게임 캐릭터 위치 표시 구현

**3. 개발**

1. 포물선



포트리스를 플레이 할 때 인풋 받는 값은 각도와 세기 2가지이다. 이 2개의 인풋 값을 의 형태로 바꿔주는 작업이 필요하다. 왜냐하면 포물선은 x축 방향으로는 등속도 운동이므로 x값이 일정하게 변함에 따라 y값이 변해가는 것을 mvaddch()함수와 sleep()함수로 나타내면 쉽게 나타낼 수 있기 때문이다.

위 그림에서 보는 것처럼 각도가 θ이고 세기가 속도에 비례하므로 (비례상수 생략) v라고 할 때 높이는가 된다. 그리고 이차함수 a(x-b)(x+b)에서의 높이는 x=0일때이므로 이다.(a<0)

즉 =이라는 식이 하나 완성된다.

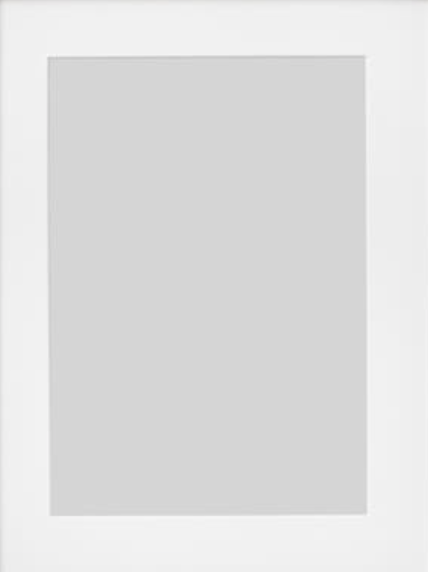
이차함수 a(x-b)(x+b)에서 x=-b일 때 미분하면 기울기가 나오는데 이때의 값은

-2ab이고 포물선에서 시작할 때의 기울기는 tan의 값이 나온다.

즉 -2ab= tan이라는 두번째 식이 나오게 된다. 이 두식을 연립하여 풀게 되면

b=(k는 비례상수) a=tan/b 의 값이 나오게 된다. 이처럼 각도와 세기를 받아 형태의 2차함수를 유도하여 포물선을 스크린에 출력할 수 있게 된다.

1. 시점

위의 포물선 출력을 사용한다면 스크린의 크기보다 큰 맵의 경우 포물선이 스크린 밖으로 나가서 게임의 진행상황(폭탄이 맞춘 위치, 맵의 모습 등등)을 알 수 없게 된다. 즉 맵이 스크린보다 큰 상황일 때 폭탄 또는 캐릭터 중심으로 화면을 이동시켜야 한다. 즉 스크린이 맵의 가장 끝부분에 닿아서 더이상 시점이 움직일 수 없는 경우를 제외하고는 캐릭터를 중심으로 화면이 이동해야 한다. 즉 오른쪽 그림이 맵 전체라면 흰색부분은 화면을 움직이지 않고 포물선을 그리면 되지만 회색부분에서는 공이 이동하는 게 아니라 그만큼 화면을 움직여야 한다.

즉 이렇게 캐릭터 시점으로 화면을 움직이게 되면 스크린보다 큰 맵도 실행시킬 수 있게 된다.

1. keyboard input받기

termflags=fcntl(0,F\_GETFL); termflags|=O\_NDELAY; fcntl(0,F\_SETFL,termflags);를 사용하면 while문 안의 getchar를 딜레이없이 상시 받을 수 있게 터미널모드를 설정하게 되어 게임이 진행되는 동안 실시간으로 input을 받을 수 있다.

1. 부가 세부사항

각각의 플레이어의 struct 설정(체력, 이동거리제한, 폭탄의 종류, 폭탄방사 크기)

게임 종료조건 : 각각의 플레이어 체력이 0이하가 되거나 맵 아래로 떨어질 경우.

여러 개의 맵 파일과 플레이어를 게임 시작 시 게임 플레이어가 직접 설정할 수 있음.

폭탄이 땅에 닿으면 원 형태로 제거 -> 가로:세로=2:1 비율인 타원형태로 제거한다. 왜냐하면 터미널에서 글자 모양이 가로:세로=1:2이기에 타원형태로 제거해야 스크린에서는 원 형태로 제거된다.

포물선의 인풋 각도와 터미널에서 보이는 실제 각도를 최대한 비슷하게 보이도록 비례상수를 지정.

스크린 표시를 char 값에서 color함수를 사용하여 색을 칠한 char ‘ ‘값으로 대체.

1. 오류

각각의 맵 파일은 미리 txt파일로 저장을 해 두고 map.txt파일을 불러와서 게임을 하는 형식으로 진행하였다. 하지만 txt파일이 총 글자수가 대략적으로 70000개 이상 넘어갈 때 bus error가 발생한다.

터미널에서 수많은 글자가 빠른 속도로 바뀔 때 잔상이 남는 오류가 생긴다.

1. 더 개선할 점

인공지능을 넣어 1인플레이가 가능하게 하기

socket통신으로 다른 컴퓨터로 각각 플레이가 가능하게 하기

backtracking 또는 궤도를 바꿔 직선으로 날라가는 유도 탄 구현하기

**4. 게임 진행과정**

상하좌우를 ‘i j k l’로 조작하고 선택 : a ,메뉴 : s , 맵 조작 : d 로 조작한다.

게임이 실행되면 터미널을 nodealy\_mode모드로 설정하고 플레이어는 플레이 할 맵을 선택한다. 선택된 맵은 txt파일에서 read모드로 받아 map[][] 배열에 넣는다.

그 뒤 플레이어는 플레이할 캐릭터를 2개 고른다. 고른 캐릭터 struct는 각각 플레이어 struct에 저장된다.

mposX, mposY의 변수로 map배열에서 스크린에 보여주는 위치를 설정해주고 set\_mpos()함수로 중심이 되는 character를 기준으로 mposX, mposY값을 정한다. 캐릭터가 움직일 때마다 화면을 지속적으로 출력하고, b메뉴로 포탄의 종류를 교체 가능하다. a선택을 누르면 각도와 세기를 정한다. 각도와 세기를 정하여서 a버튼 선택을 누르면 parabola()함수를 실행하여 포물선을 그리면 포탄을 발포 한다.(이때 화면이동의 중심은 캐릭터에서 포탄으로 교체) 만약 맞은 캐릭터가 있다면 맞은 캐릭터의 체력 값을 포탄의 데미지만큼 감소시키고 다음 플레이어의 차례로 넘긴 후 한명이 죽거나 낙사 할 때까지 반복한다.

지원하는 캐릭터는 6개이고 맵은 3가지의 종류가 존재한다.

**5. 소감**

김진규 : 처음에 계획을 잡을 때 이것을 다 할 수 있을까 고민이었다. 하지만 하나 둘씩 계획대로 코드를 작성하다 보니 완성할 수 있게 되었다. 가끔은 오류가 생기고 어려워 고민스러울 때가 있었지만, 내가 작성한 코드가 터미널 상에서 잘 작동했을 때 정말 뿌듯했던 것 같다. 무엇보다 한 명의 팀원으로서 팀에 큰 기여를 했다는 것이 자랑스럽다.

김현학 : 많은 과목을 수강해오면서 많은 종류의 코드를 작성했었지만 이번처럼 팀 프로젝트로 코드를 작성해본 경험은 거의 없었다. 혼자 모든 코드를 개발한다면 그냥 이해하고 넘어갔을 함수나 변수 명 등을 공동 작업을 하기에 좀더 명확하게 쉽게 이해할 수 있도록 한번 더 고민하고 작성하였다.

특히 나는 수학을 잘 하지 못하는데 팀원들이 포물선에 관한 식을 도출하고 구현하는 것을 지켜보니 팀 프로젝트가 서로의 부족한 부분을 채워주고 더 나은 프로그램을 만들어 낼 수 있다는 것을 깨닫고 협업의 중요성에 대하여 이해하게 되었다.

윤성환 : 오픈소스가 나에게 너무 많이 어렵게 느껴졌기 때문에 팀프로젝트를 시작할 때 상당히 걱정했다. 하지만 다행이 팀에 도움이 되어서 정말 다행이라고 생각한다. 아무리 생각해도 우리 팀 결과물이 너무 괜찮은 것 같다. 내가 포함되어 있는 조가 이런 복잡한 게임을 만들었다는 게 대단하다고 느껴진다. 그리고 이 프로젝트를 성공적으로 마무리 지을 수 있어서 너무 기쁘다.

이수진 : 처음 팀을 짤 때부터 함께한 것이 아니라, 도중에 참여해서, 프로젝트에 그리 큰 기여를 하지 못 한 점을 안타깝게 생각한다. 나 역시 프로그래밍 과목에서 진행된 팀프로젝트는 처음인데, 다른 교양에서 해봤던 조별과제와는 진행도 구성도 모두 다른 것이 실감났다. 조원끼리 파트를 쉽게 구분할 수 있었던 교양 조별과제와는 달리, 팀프로젝트는 한 가지 목표에 대해 여러가지 일이 동시에 진행되어서, 조원 간의 소통과 의견교류가 중요하다는 것을 느낄 수 있었다.