기말 Trem Project

202001485 김기수

목차)

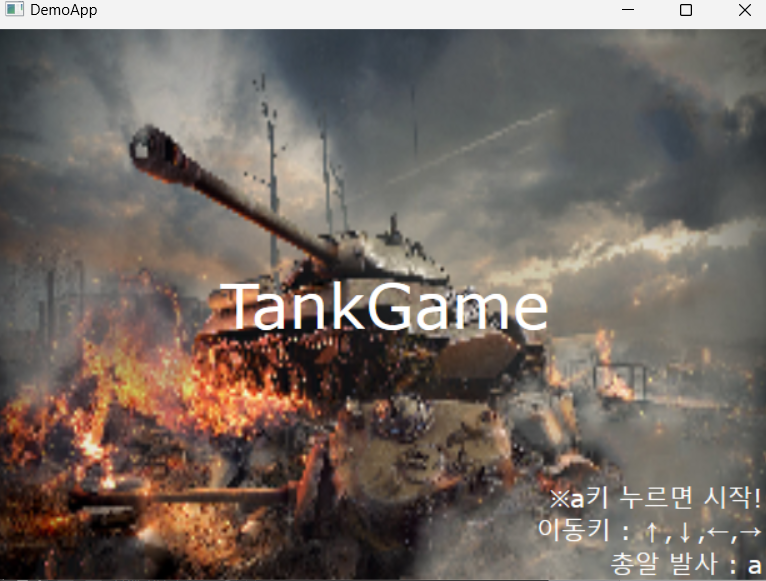
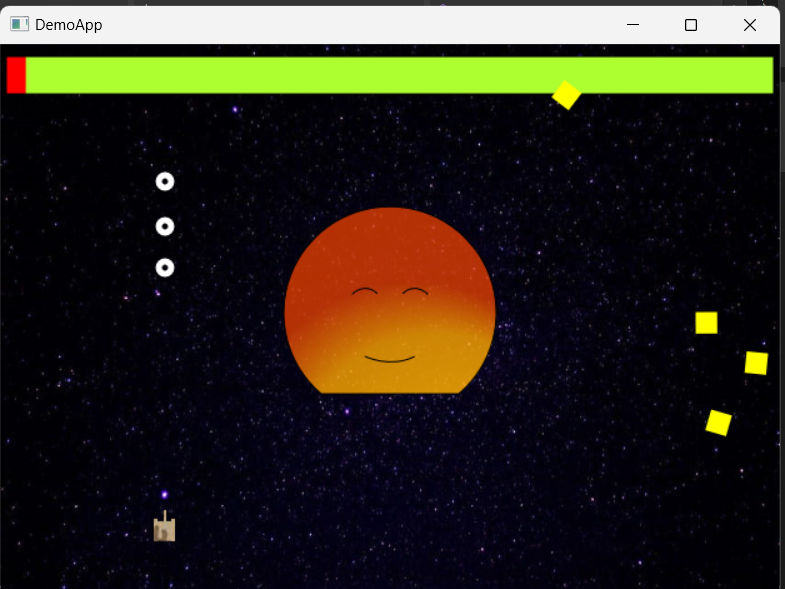
1. 게임기획
2. 디자인
3. 코드 설명
4. 실행 화면
5. 게임기획

간단한 탱크 게임을 ID21을 이용해서 만들어보자. 게임시작화면에서 A키를 누르면 게임화면으로 들어가게 되고 중앙에 보스 몹이 존재한다. 탱크는 위에서 내려다 보는 구조로 총알을 발사 할 수 있다. 보스 몹인 태양의 체력 바가 있어 체력에 따른 난이도 상승과 표졍변화가 존재한다. 태양이 내뿜는 장애물들을 피해 태양을 제압하면 게임에서 승리하고 R키를 통해 첫 화면으로 돌아가게한다.

1. 디자인

시작 화면에서는 게임다운 분위기를 내기 위해 웅장한 BGM과 화면에 사용 설명 키가 보이게 설정해 두었다.

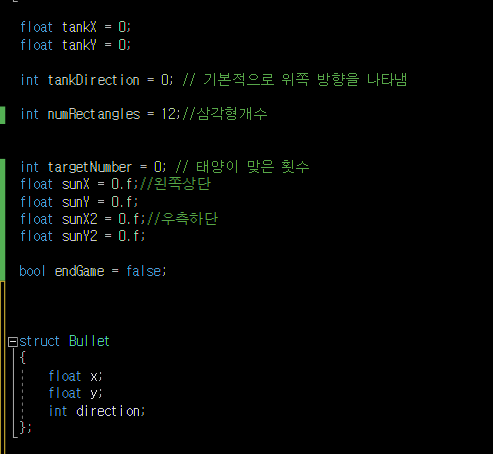
게임 화면에서 디자인을 그릴 때 장애물들과 내가 발사 한 총알을 눈에 잘 들어오게 하기 위해 배경을 우주으로 사진으로 설정 하기로 마음먹었다. 그에 따라 총알은 흰색, 중앙 보스몹은 태양을 그라데이션으로 칠했고 태양이 발사하는 장애물은 노란색으로 설정해 두었다. 보스몹의 체력 바도 초록색으로 해 가시성이 좋게 설정했다. 그리고 체력 바에 따른 태양의 표정 변화도 확인 할 수 있다.



시작화면 게임화면

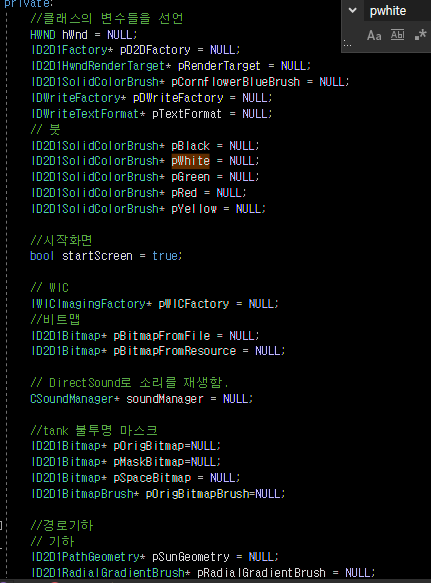
1. 코드 설명

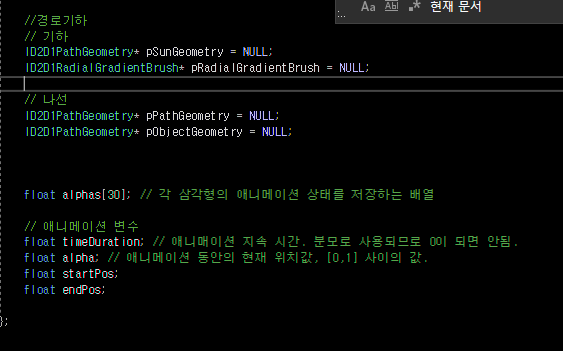
1)전역 변수



순서대로 탱크의 좌표, 탱크가 향하고 잇는 방향, 태양이 내뿜는 장애물의 개수, 태양이 총알을 맞은 횟수, 태양의 좌표를 저장하는 함수, 태양을 잡았을 때 끝난 게임을 표시하기 위한 bool함수가 있다. 그리고 총알의 하나하나의 좌표,방향을 저장하는 구조체 Bullet이 있다. 이것을 추후에 벡터를 통해 push.back을 해주는 것을 확인 할 수 있을 것이다.

2)demoApp 초기 변수들

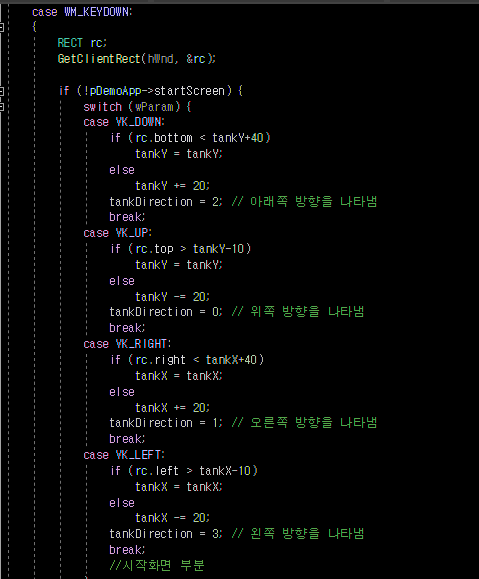
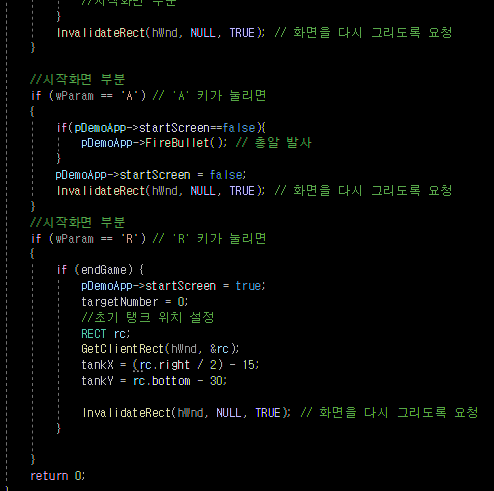




배경과 개체들을 비트맵을 통해 만들었고, soundManger을 통해 소리를 불러왔다. 또한 탱크의 모양 불투명 마스크로 만들어 색을 입히면 탱크를 찍어낼 수 있게 만들었다. 경로기하를 통해 태양을 구현했고 태양이 내뿜는 장애물을 나선이란 경로기하를 사용해 애니메이션에 이용했다.

Alpha[]는 삼각형 하나하나의 애니메이션 상태를 저장하기 위해 만들었다.(초깃값 0.f)

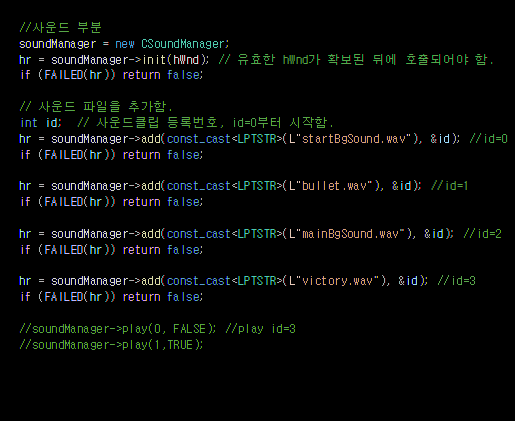
3)키보드 눌렀을 때 이벤트 처리

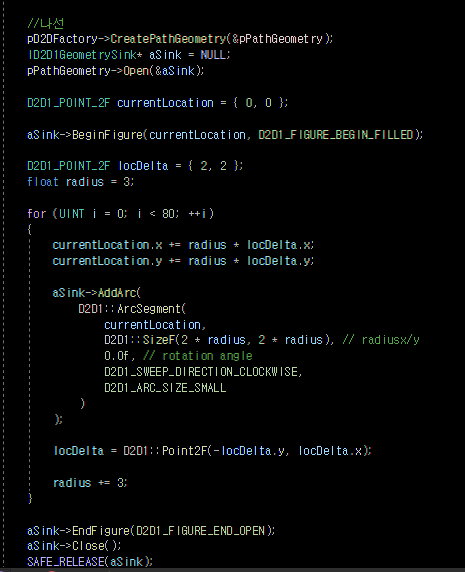


우선 방향키를 눌렀을 때 상하좌우로 움직여야 하기 때문에 tankX,tankY를 조절했고 탱크의 방향을 설정해 주었다.

그리고 A키가 눌렸을 때 총알 발사하는 부분을 구현 해주었고 함수를 통해 가시성을 높였다. 만약 게임이 종료되면 R을 눌러 처음화면으로 가게 하였고 그에 따른 탱크의 위치, 맞은 개수, 사운드 등을 초깃값 으로 설정했다.

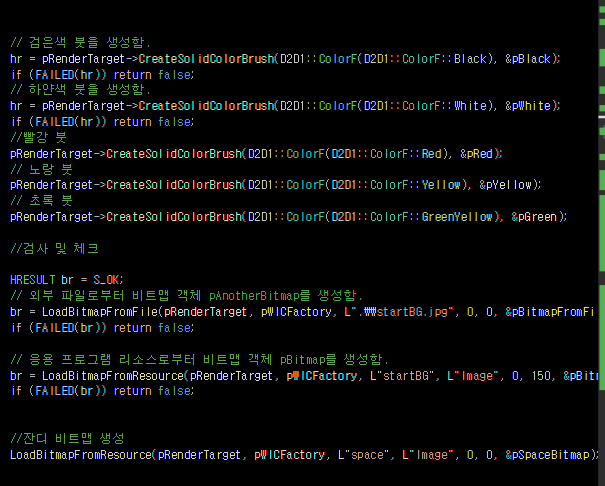
4) createAppResource

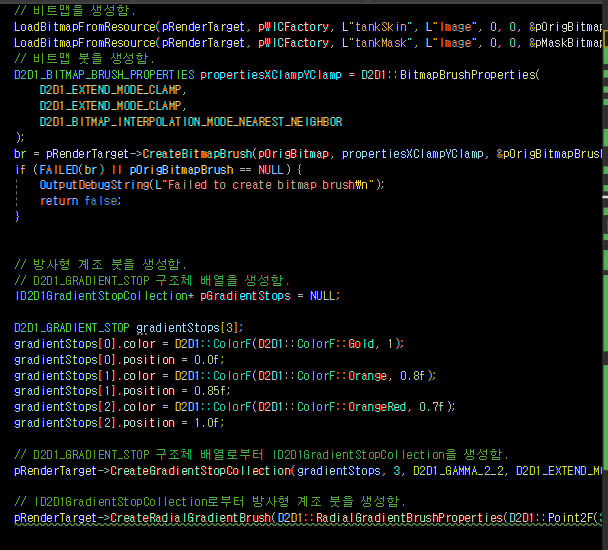




시작화면, 총알, 게임화면, 승리 사운드를 넣어주었고, 나선의 경로기하를 만들어주었다.

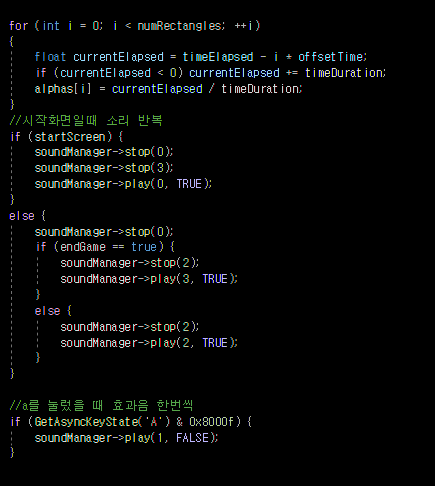
5) createDeviceResource





브러쉬들을 설정했고, 우주 배경, 시작화면 배경을 비트맵으로 만들고 탱크를 불투명 마스크로 만들기위해 tankMask를 불러오는 것을 확인 할 수 있다. 그리고 태양을 위한 방사형 계조 붓을 생성해서 그라데이션 효과를 줄려고 하는 것을 확인 할 수 있다.

6) update

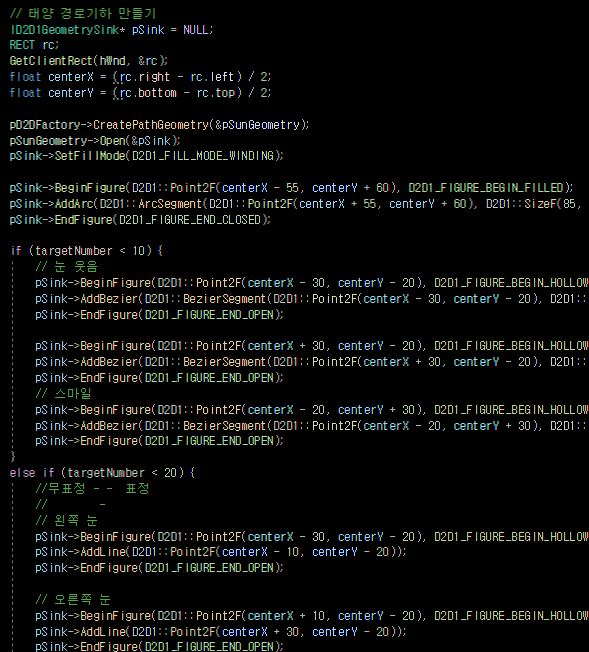


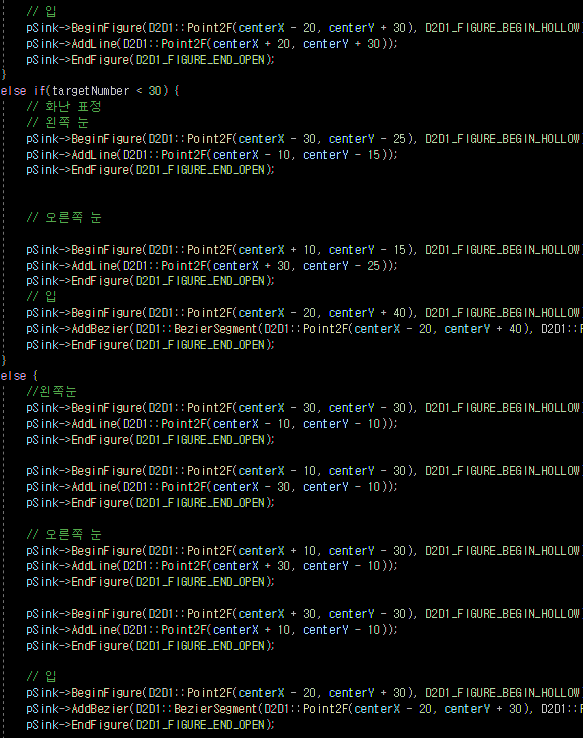
장애물 사각형의 애니매이션 부분을 설정해주고 사운드 설정을 해주는 것을 알 수 있다. 또한 a키를 눌렀을 때 효과음을 나는 것을 update에서 처리를 해 주었다.

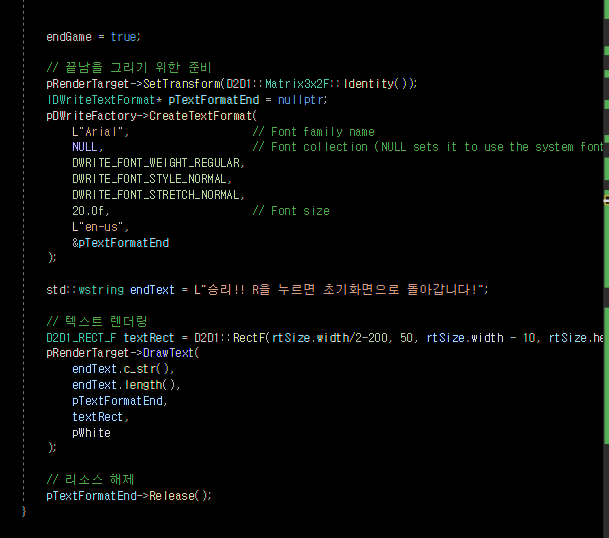
7) render



시작화면 일 때를 boolean을 통해 조건을 걸어주어 시작화면 게임화면을 구분지어 주었다. 게임화면을 먼저 살펴보면 처음에 우주 배경을 그리고 targetNumber>0 즉 한번이라도 태양이 맞으면 체력바가 표시 되도록 설정했다. 그리고 처음에 초록색으로 설정해주고 데미지를 받을수록 빨간색 사각형이 퍼센트 단위로 차 점점 초록색이 줄어드는 것처럼 보인다. targetNumber<31으로 해주었는데 이는 30번 맞으면 태양이 죽는 것이므로 그 이후는 체력 바가 변하지 않게 설정해주었다.



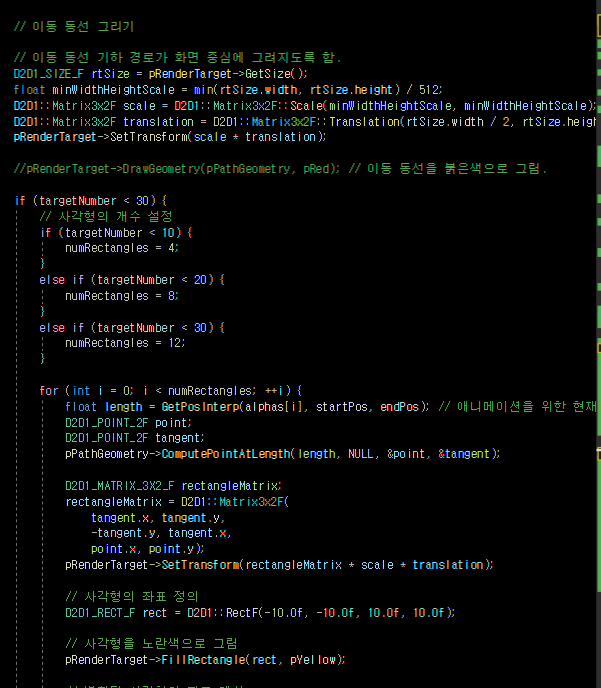


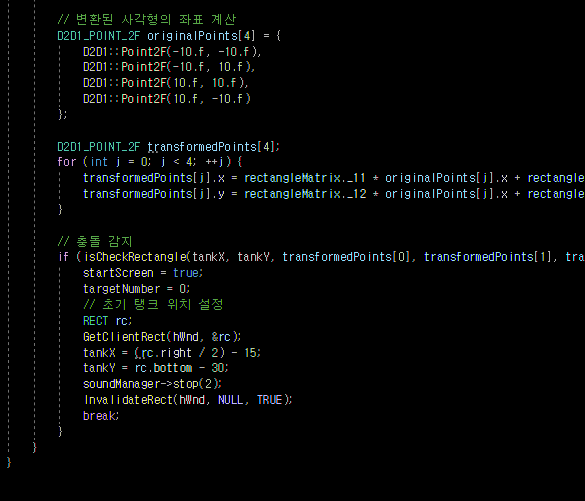


태양경로기하를 만들어 배경을 칠하고 targetNumber에 따라 이모티콘이 달라지는 것을 확인 할 수있다. 10보다 작을때는 웃음 10과 20사이일때는 20과 30 사이는 화남 그 이상이 되면 죽는 이모티콘을 설정 했다. 만약 태양을 처치하면 endgame을 true라고 설정을 해주고 승리에대한 텍스트가 나오게 설정한다.

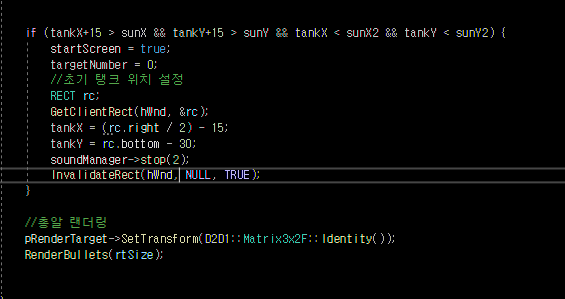


태양의 히트 박스를 정해 주었고 탱크가 키보드 방향에 따라 회전하는 것을 TransForm을 통해 구현하였다.





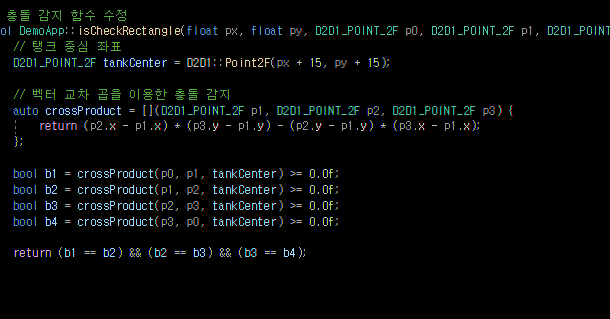
태양이 내뿜는 사각형 장애물의 대한 처리 부분이다. 애니매이션을 위해 아까 나선으로 만들어 현재 위치를 받아오고 그려준다. 사각형의 좌표를 휙득하여 충돌감지 함수로 넘어간다. 충돌이 감지되면 startScreen을 키고 초기값으로 설정해준다.



태양 히트박스안에 내 탱크가 들어갈 경우도 죽는 처리를 해주었다.

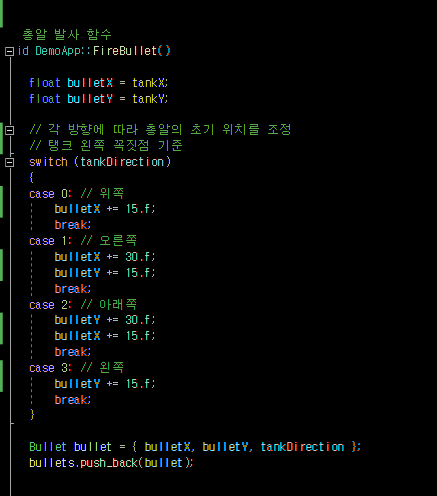
그리고 마지막으로 총알을 랜더링해준다.

7) 충돌 감지 함수



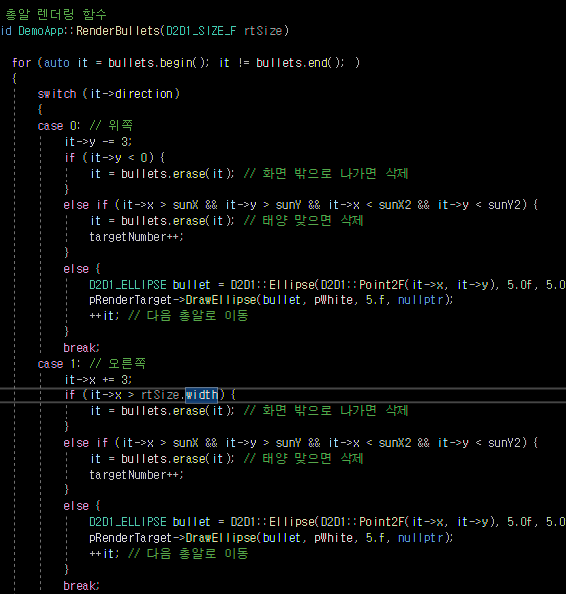
사각형의 계속 회전하기 때문에 더 정확한 충돌 감지를 위해서 사각형의 각 변을 검사하는 방식을 사용했다. 벡터 교차 곱을 사용해 검사를 한 부분이다.

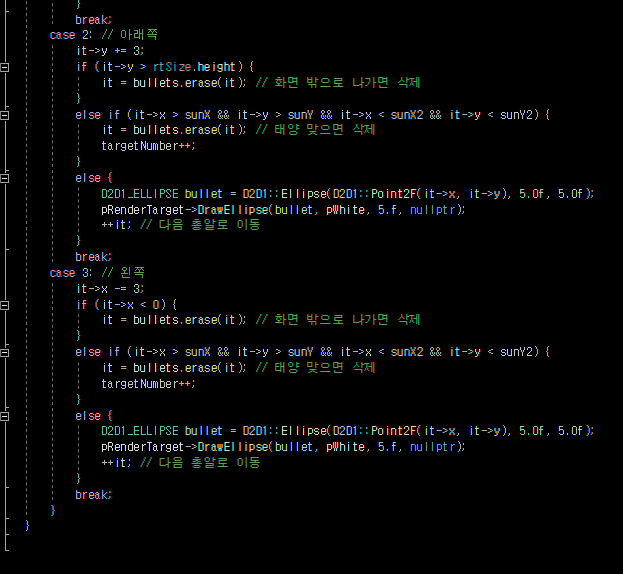
8) 총알 발사 함수



A키를 눌렀을 때 탱크 중앙이 아닌 탱크 이미지의 포탄 쪽에 나가게 하기 위해서 좌표를 약간 수정해 주고 bullet구조체에 push를 해준다.

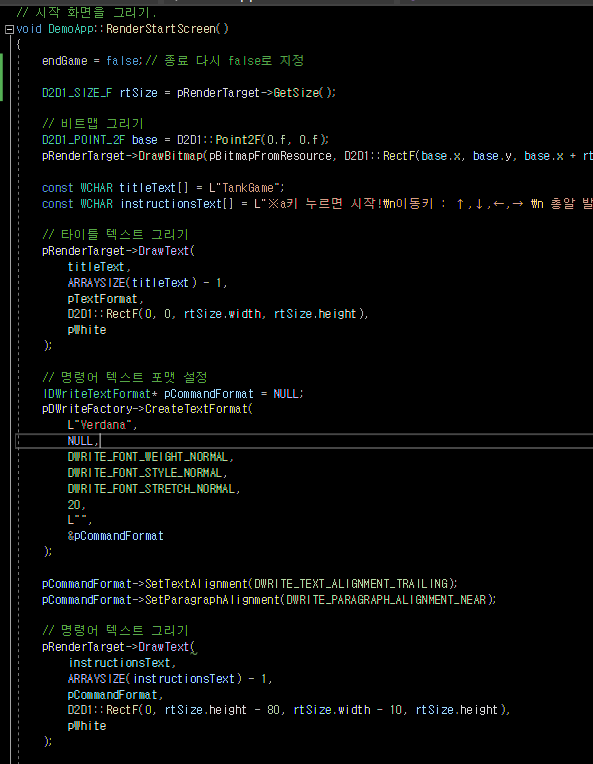
9)총알 랜더함수





Bullets의 벡터를 순회하면서 총알이 태양 맞거나 화면 밖으로 나가면 삭제를 해준다. 아니면 그화면에 그리도록한다. 태양에 맞으면 targetNumber이 증가되는 것을 확인 할 수 있다.

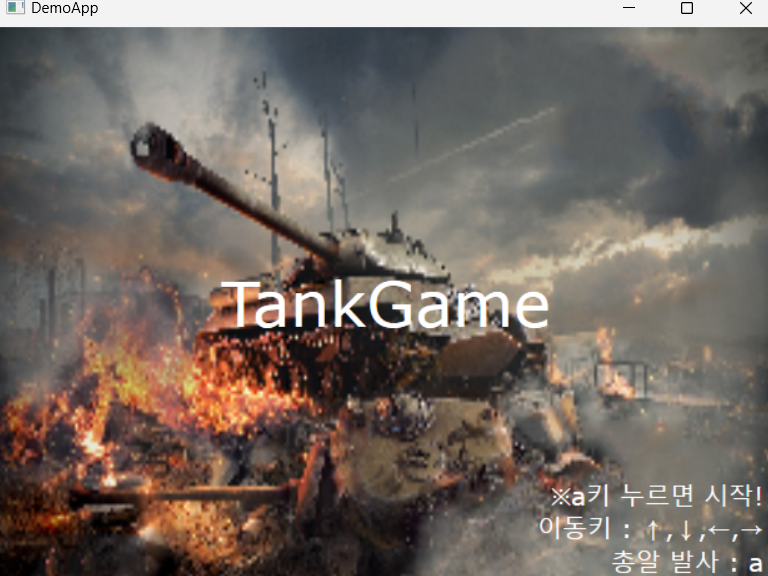
10)



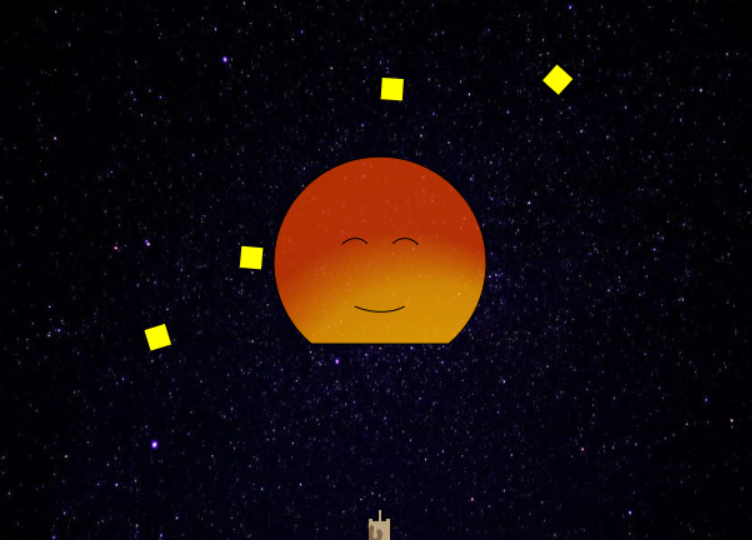
맨 처음 비트맵을 통해 배경을 불러오고 나머지는 텍스트를 지정해 준 것을 확인 할 수 있다.

1. 실행화면

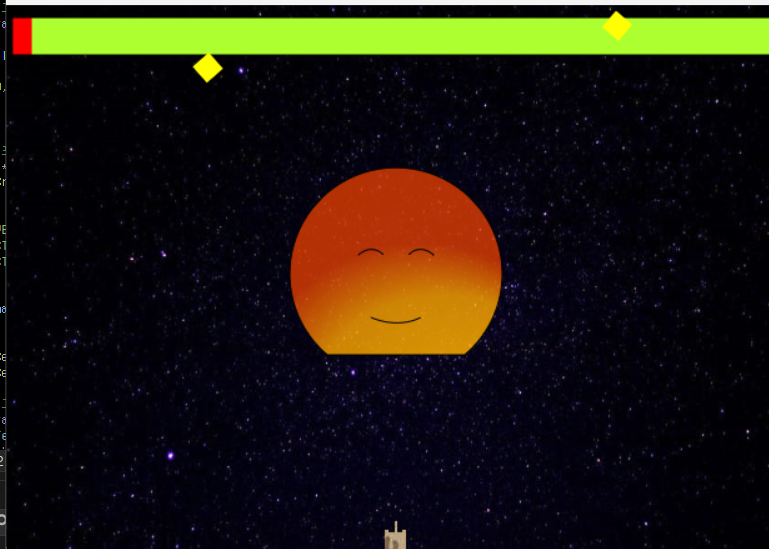
첫화면



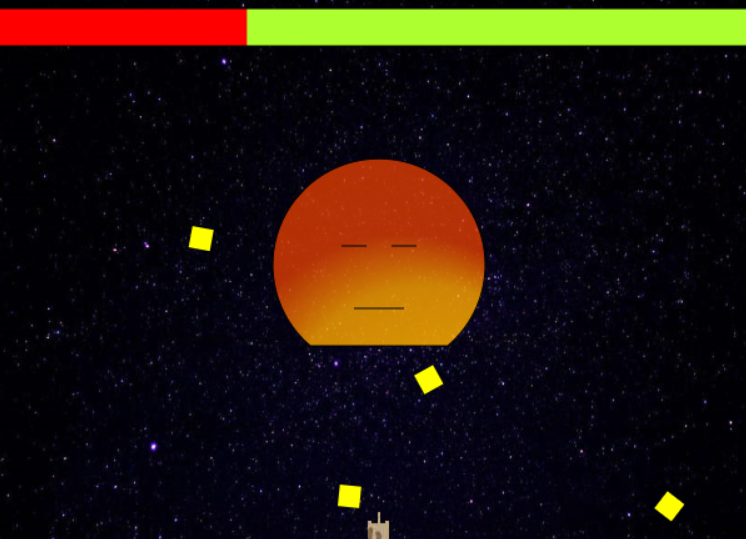
게임화면(초기)



태양을 한대 때림

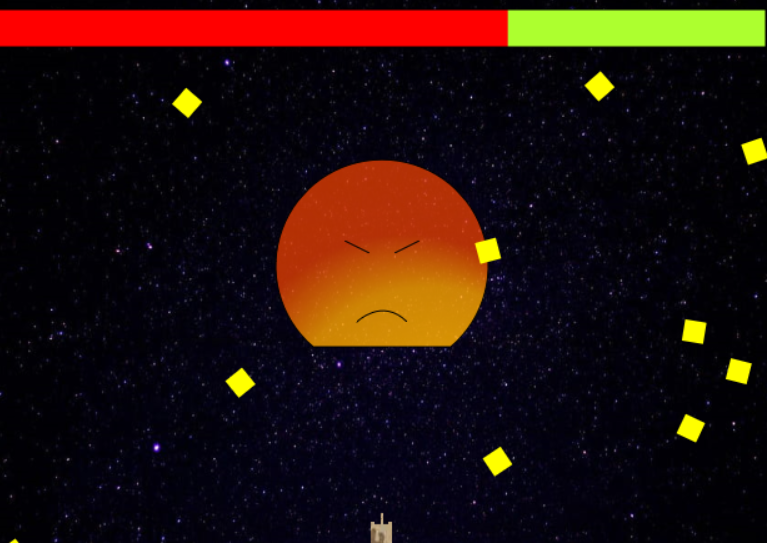


태양을 10번때림



표정과 사각형의 개수가 증가

태양을 20번 때림



표정과 사각형 개수 증가

태양을 30번 때림



태양이 죽고 r을 누르면 초기화면으로 돌아감