

# 05. 총알 발사

```

1 // AssortrockWINAPI.cpp : Defines the entry point for the application.
2 //
3
4 #include "pch.h"
5 #include "framework.h"
6 #include "AssortrockWINAPI.h"
7 #include <list>
8
9 #define MAX_LOADSTRING 100
10
11 typedef struct _tagRectangle
12 {
13     float l, t, r, b;
14 }RECTANGLE, *PRECTANGLE;
15
16 // Global Variables:
17 HINSTANCE hInst; // current instance
18 WCHAR szTitle[MAX_LOADSTRING]; // The title bar text
19 WCHAR szWindowClass[MAX_LOADSTRING]; // the main window class name
20
21 HWND g_hWnd;
22 HDC g_hDC;
23 bool g_bLoop = true;
24 RECTANGLE g_tPlayerRC = { 100, 100, 200, 200 };
25
26 typedef struct _tagBullet
27 {
28     RECTANGLE rc;
29     float fDist;
30     float fLimitDist;
31 }BULLET, *PBULLET;
32
33 // 플레이어 총알 리스트
34 std::list<BULLET> g_PlayerBulletList;

```

총알 구조체

if( GetAsyncKeyState( VK\_SPACE ) & 0x8000 ) 스페이스 바 누르면 총알 생성.

```

{
    BULLET tBullet = { RECTANGLE{0.0f, 0.0f, 0.0f, 0.0f}, 0.0f, 0.0f };

    tBullet.rc.l = g_tPlayerRC.r;
    tBullet.rc.r = tBullet.rc.l + 50.f;
    tBullet.rc.t = (g_tPlayerRC.t + g_tPlayerRC.b) / 2.0f - 25.f;
    tBullet.rc.b = tBullet.rc.t + 50.f;
    tBullet.fDist = 0.f;
    tBullet.fLimitDist = 500.f;

    g_PlayerBulletList.push_back( tBullet );
}

Rectangle( g_hDC, g_tPlayerRC.l, g_tPlayerRC.t, g_tPlayerRC.r, g_tPlayerRC.b );

std::list<BULLET>::iterator iter;
std::list<BULLET>::iterator iterEnd = g_PlayerBulletList.end();

```

// 총알의 이동속도  
fSpeed = 600.f \* g\_fDeltaTime \* fTimeScale; <총알이동>

```

for( iter = g_PlayerBulletList.begin(); iter != iterEnd; )
{
    iter->rc.l += fSpeed;
    iter->rc.r += fSpeed;

    iter->fDist += fSpeed;

    if( iter->fDist >= iter->fLimitDist )
    {
        iter = g_PlayerBulletList.erase( iter );
        iterEnd = g_PlayerBulletList.end();
    }

    else if( iter->rc.l > clientRect.right )
    {
        iter = g_PlayerBulletList.erase( iter );
        iterEnd = g_PlayerBulletList.end();
    }

    else
    {
        ++iter;
    }
}

```

erase 사용시  
for문 구조

erase는 삭제 후  
삭제될 때의 위치  
다음 위치 iterator를  
반환함.

삭제된 뒤 end iterator 위치를  
갱신해 준다.

iterator 증가를 for문 위치에 늘림

erase 해서 iterator가 다음 위치로 이동한 상태에서  
해당 iterator를 스킵하고 다음으로 이동해버림. <총알 그리기>

```

for( iter = g_PlayerBulletList.begin(); iter != iterEnd; ++iter )
{
    Rectangle( g_hDC, iter->rc.l, iter->rc.t, iter->rc.r, iter->rc.b );
}

```

Rectangle( g\_hDC, 0, 0, 800, 600 ); 다 불러오고 한 번 덮어버림.

