

## 22. 총알 위치 재설정 및 스테이지 배경 출력

```
void CPlayer::Fire()
{
    CObject* pBullet = CMoveObject::CreateCloneObj( "Bullet", "PlayerBullet", m_pPlayer );

    // Get middle point of right
    POSITION tPos;
    tPos.x = m_tPos.x + (1.f - m_tPivot.x) * m_tSize.x;
    tPos.y = m_tPos.y + (0.5f - m_tPivot.y) * m_tSize.y;

    pBullet->SetPos( tPos.x, tPos.y - pBullet->GetSize().y / 2.f );
}
```

Player.cpp

→ 피봇값이 그때 그때마다 달라짐.  
그래서 x는 1에서 빼주고, y는 가운데를 맞춰주기 위해 0.5에서 빼준만큼 기존 y값에 더해주는 것.

```
#pragma once
#include "StaticObject.h"

class CStage : public CStaticObject
{
private:
    friend class CObject;
    friend class CScene;

public:
    CStage();
    CStage( const CStage& stage );
    ~CStage();

public:
    virtual bool Init();
    virtual void Input( float fDeltaTime );
    virtual int Update( float fDeltaTime );
    virtual int LateUpdate( float fDeltaTime );
    virtual void Collision( float fDeltaTime );
    virtual void Render( HDC hDC, float fDeltaTime );
    virtual CStage* Clone();
};
```

CStage.h

```
#include "Stage.h"
#include "../Resources/Texture.h"
#include "../Core.h"
```

CStage.cpp

```
CStage::CStage()
{
}

CStage::CStage( const CStage& stage )
:
    CStaticObject( stage )
{
}

CStage::~CStage()
{
}

bool CStage::Init()
{
    SetPos( 0.f, 0.f );
    SetSize( 1280.f, 720.f );
    SetPivot( 0.0f, 0.0f );

    SetTexture( "Stage1", L"Stage1.bmp" );

    return true;
}

void CStage::Input( float fDeltaTime )
{
    CStaticObject::Input( fDeltaTime );
}

int CStage::Update( float fDeltaTime )
{
    CStaticObject::Update( fDeltaTime );
    return 0;
}

int CStage::LateUpdate( float fDeltaTime )
{
    CStaticObject::LateUpdate( fDeltaTime );
    return 0;
}

void CStage::Collision( float fDeltaTime )
{
    CStaticObject::Collision( fDeltaTime );
}

void CStage::Render( HDC hDC, float fDeltaTime )
{
    //CStaticObject::Render( hDC, fDeltaTime );

    if( m_pTexture )
    {
        BitBlt(
            hDC, m_tPos.x - m_tSize.x * m_tPivot.x,
            m_tPos.y - m_tSize.y * m_tPivot.y,
            GETRESOLUTION.iw, GETRESOLUTION.ih, m_pTexture->GetDC(),
            0, 0, SRCOPY );
    }
}

CStage* CStage::Clone()
{
    return new CStage(*this);
}
```

위치, 크기 설정  
텍스처 불러오기

화면 크기 만큼 출력

```
#include "Scene.h"
#include "Layer.h"
#include "../Object/Object.h"

unordered_map<string, class CObject*> CScene::m_mapPrototype;

CScene::CScene()
{
    CLayer* pLayer = CreateLayer( "UI", INT_MAX );
    pLayer = CreateLayer( "Default", 1);
    pLayer = CreateLayer( "Stage" );
}
```

Scene.cpp

스테이지 레이어 번호를 0. 제일 아래  
출력할 수 있도록 함.

```
bool CInGameScene::Init()
{
    // 부모 Scene 클래스의 초기화 함수를 호출해줌.
    if( !CScene::Init() )
    {
        return false;
    }

    CLayer* pLayer = FindLayer( "Default" );

    CPlayer* pPlayer = CObject::CreateObj<CPlayer>( "Player", pLayer );

    SAFE_RELEASE( pPlayer );

    CMinion* pMinion = CObject::CreateObj<CMinion>( "Minion", pLayer );

    SAFE_RELEASE( pMinion );

    // 총알 프로토타입을 만들어 준다.
    CBullet* pBullet = CScene::CreatePrototype<CBullet>( "Bullet" );

    pBullet->SetSize( 50.f, 50.f );

    SAFE_RELEASE( pBullet );

    // Stage
    CLayer* pStageLayer = FindLayer( "Stage" );
    CStage* pStage = CObject::CreateObj<CStage>( "Stage", pStageLayer );

    SAFE_RELEASE( pPlayer );

    return true;
}
```

InGameScene.cpp

스테이지 생성