



공개SW 프로젝트 BAOS 게임 제작

2조

2015112119 컴퓨터공학과 이현재

2015112084 컴퓨터공학과 조재권

2015112121 컴퓨터공학과 최성준



BAOS Online Game

TODAY's Agenda

01 본론1
클라이언트 부분 상황 보고

02 본론2
네트워크 부분 상황 보고

03 계획
다음 주 계획 보고



클라이언트 부분 클래스 다이어그램

Sprite.h

좌표를 받아 **이미지**를 출력



Texture.h

출력한 이미지에 **텍스처**를 출력



Camera.h

유저 화면에 출력할 **화면**



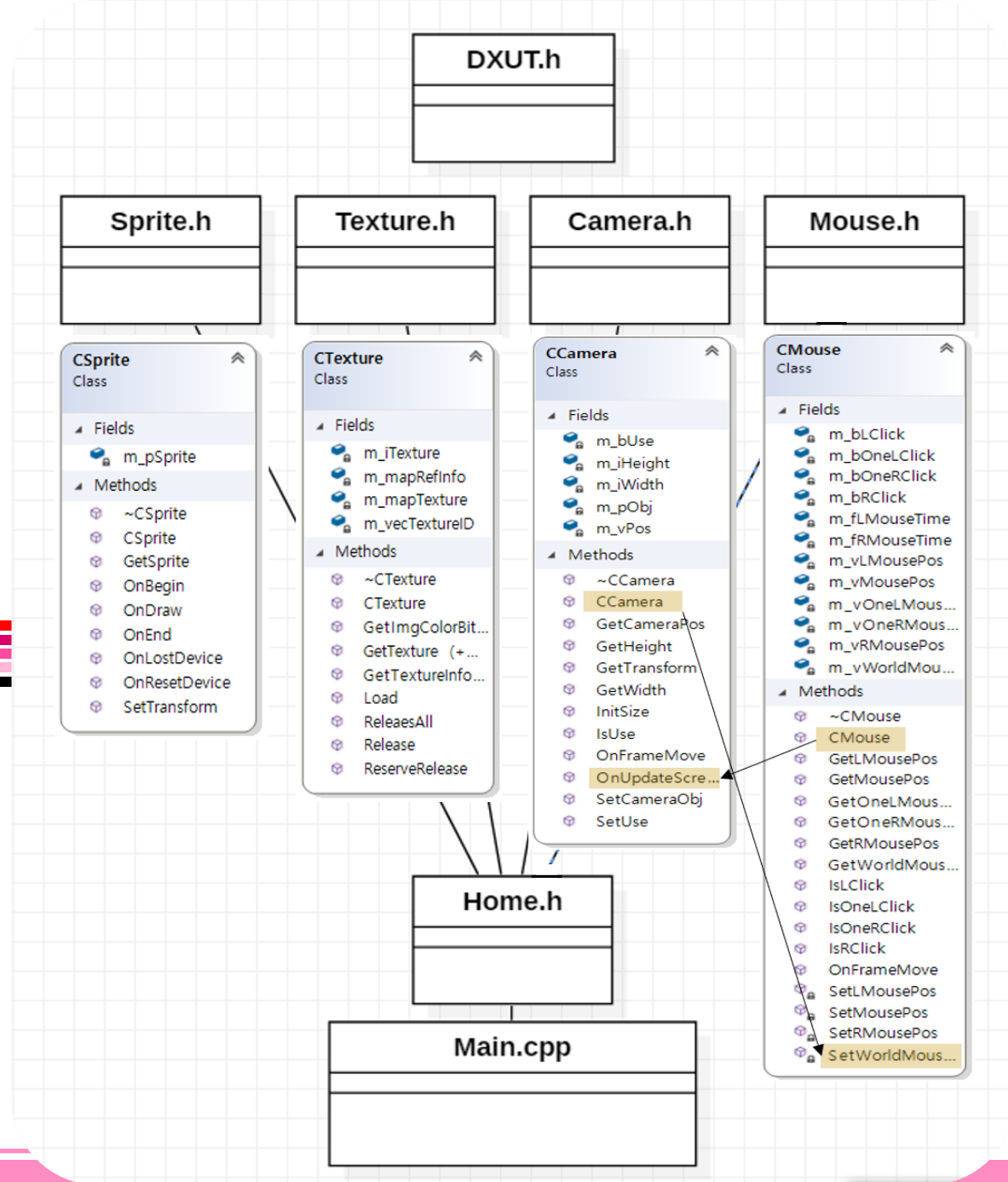
Mouse.h

시스템 **마우스의 위치**와 인게임의
위치를 일치시킴



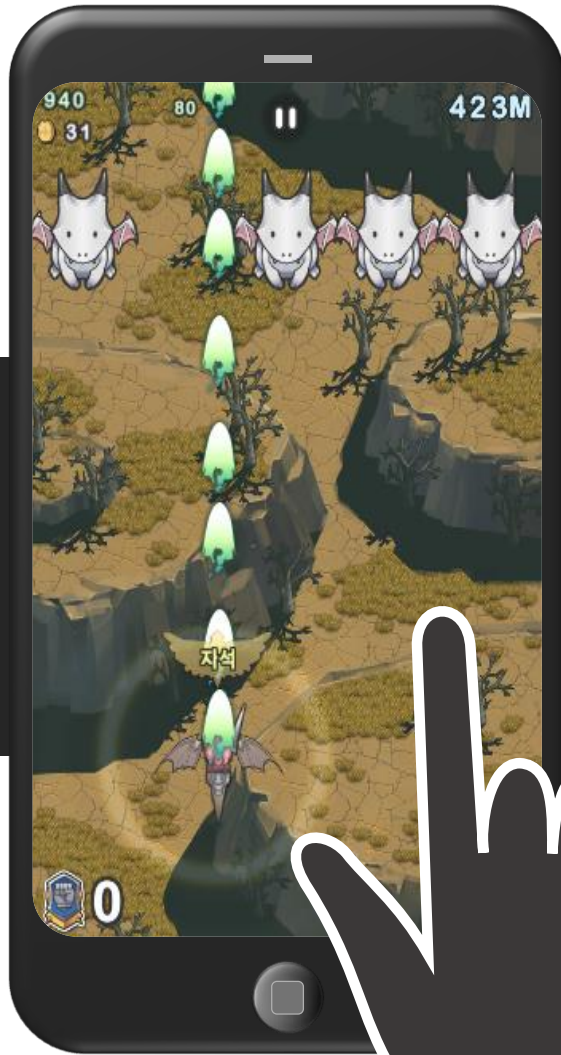
Debug.h

게임내 **실시간 변경사항 적용**
테스트 용도

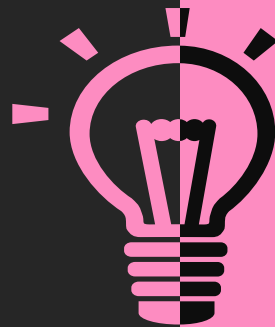




클라이언트 부분 클래스 기능 소개 - CCamera Class



사물 고정
배경 움직임



배경 고정
카메라 움직임



```
*****
주석 작성자: 최성준 2019.04.11 목요일 작성
주석 수정: 조재권 2019.04.13 토요일 수정
*****/
```

```
void CCamera::OnFrameMove( float fElapsedTime )
{
    if( KEYDOWN( VK_SHIFT ) ) // 키보드의 Shift키가 눌렸는지 확인 systemCall
        m_vPos = *m_pObj->GetPos(); // D3DXVECTOR2 2차원 객체가 카메라 위치를 받아온다.
    else
        OnUpdateScreen( fElapsedTime );
        // 컴퓨터 성능에 따라 게임속도가 달라지지 않도록 주기적으로 화면 갱신

    m_vPos.x = max( m_vPos.x, m_iWidth * 0.5f ); // 0.5를 곱해서 가로 중심 유지
    m_vPos.y = max( m_vPos.y, m_iHeight * 0.5f ); // 0.5를 곱해서 세로 중심 유지

    m_vPos.x = min( m_vPos.x, 2048.f - ( m_iWidth * 0.5f ) );
    m_vPos.y = min( m_vPos.y, 2048.f - ( m_iHeight * 0.5f ) );
}

void CCamera::OnUpdateScreen( float fElapsedTime )
{
    D3DXVECTOR2 vMouse = g_pMouse->GetMousePos();
    // 유저의 마우스가 화면 밖을 가리킬 경우 카메라를 이동시킨다.
    if( vMouse.x <= m_iWidth * 0.01 ) // 왼쪽 경계
        m_vPos.x -= 800 * fElapsedTime;

    if( vMouse.y <= m_iHeight * 0.01 ) // 위쪽 경계
        m_vPos.y -= 800 * fElapsedTime;

    if( vMouse.x >= m_iWidth * 0.99 ) // 오른쪽 경계
        m_vPos.x += 800 * fElapsedTime;

    if( vMouse.y >= m_iHeight * 0.99 ) // 아래쪽 경계
        m_vPos.y += 800 * fElapsedTime;
}

void CCamera::GetTransform( D3DXMATRIX * pmatTrans ) // D3DXMATRIX는 d3d9math 헤더파일 내의 행렬연산 구조체다.
{
    // 유저의 마우스가 화면 밖을 가리킬 경우 카메라를 이동시킨다. 이때, 카메라의 중심은 항상 화면의 중심이 된다.
    pmatTrans->_41 = (float)m_iWidth * 0.5f - m_vPos.x; // 0.5를 곱해서 가로 중심 유지
    pmatTrans->_42 = (float)m_iHeight * 0.5f - m_vPos.y; // 0.5를 곱해서 세로 중심 유지
    pmatTrans->_43 = 0.0f;
}
```

Synchronize

마우스에 따라 움직이는 화면
마우스가 화면 경계에 닿으면
카메라도 따라서 움직임



Limitation

카메라의 움직임 **제한**
위치는 언제나 중앙 고정
성능 독립적 게임속도

성능 독립적
가능 구현의
중요성



총기 반동 \propto 프레임



로딩 속도 \propto 프레임

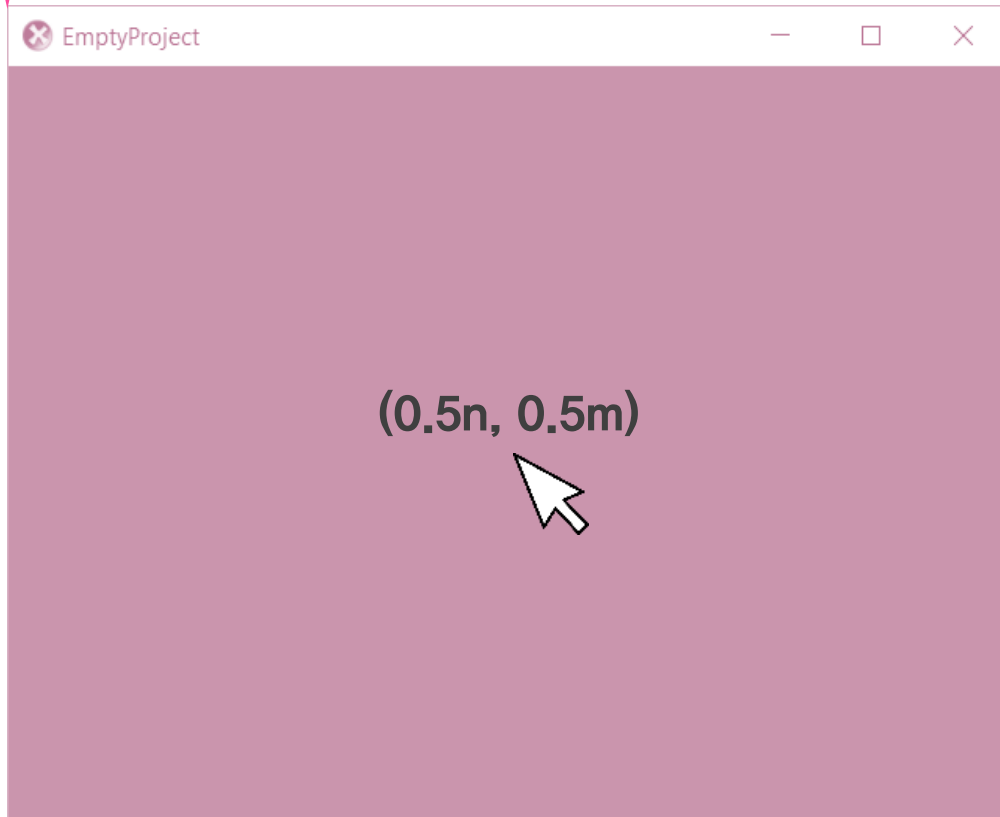
BIOHAZARD **RE:2**

칼 데미지 \propto 프레임



클라이언트 부분 클래스 기능 소개 - CMouse Class

(0, 0)



(0.5n, 0.5m)

(n, m)

시스템상 마우스의 좌표와
게임 스크린 내부에서 좌표를
일치시켜야 한다.
각종 발생 가능 오류를 막기 위해



클라이언트 부분 클래스 기능 소개 - CMouse Class

```
void CMouse::SetWorldMousePos()  
{  
    m_vWorldMousePos = m_vMousePos; // 월드 마우스 initialize  
    // 월드 마우스의 x좌표와 y좌표를 카메라의 좌표만큼 변경해준다.  
    m_vWorldMousePos.x += ( g_pCamera->GetCameraPos().x - ( g_pCamera->GetWidth() * 0.5f ) );  
    m_vWorldMousePos.y += ( g_pCamera->GetCameraPos().y - ( g_pCamera->GetHeight() * 0.5f ) );  
}  
  
// 오른쪽 마우스 버튼 세팅 멤버 함수  
void CMouse::SetRMousePos( float fElapsedTime )  
{  
    m_vOneRMousePos = NULLVECTOR2; // 마우스 좌표 초기화  
    m_bOneRClick = false; // 아직 클릭 X  
  
    m_fLMouseTime += fElapsedTime;  
  
    // 오른쪽 마우스 클릭 시간이 충분히 길다면  
    if((m_fRMouseTime > 4 * fElapsedTime)&& KEYDOWN( VK_RBUTTON ) )  
    {  
        m_vRMousePos = m_vWorldMousePos; // SetWorldMousePos()함수에서 설정한 캐릭터 좌표값 대입  
        m_fRMouseTime = 0; // 다음 입력을 위해 초기화.  
        m_bRClick = true; // 오른쪽 마우스 클릭 된거 확인.  
    }  
  
    // 오른쪽 마우스가 클릭 된 게 아닌 상태면  
    // 계속 flag를 true로 놔둘 수 없으니 (true가 계속되면 마우스 좌표를 계속 따라가니까)  
    else if( KEYDOWN( VK_RBUTTON ) == false )  
    {  
        if( m_bRClick == true )  
        {  
            m_bRClick = false; // false로 만들어줘야하고  
            m_bOneRClick = true; // 단타 입력 플래그 true로 만들  
  
            m_vOneRMousePos = m_vRMousePos; // 좌표 받아오기  
        }  
    }  
}
```

“

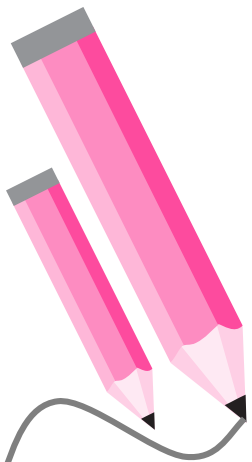
Set mouse

마우스 기능 구현
보편적인 마우스 기능
캐릭터의 좌표 이동 기능
긴 입력과 짧은 입력을 구분

”



클라이언트 부분 클래스 기능 소개 - CSprite Class



D3DXMATRIX 라는 행렬 이용
사용자의 입력키, 좌표에 따라
원하는 행동, 원하는 동작을 하는
이미지를 출력할 수 있다.



(0, 1)



(0, 2)



(0, 3)



(1, 0)



(1, 1)



(1, 2)



(1, 3)



(2, 0)



(2, 1)



(2, 2)



(2, 3)





클라이언트 부분 클래스 기능 소개 - CSprite Class

```
void CSprite::SetTransform( D3DXMATRIX *pmatTrans )
{
    if( m_pSprite == NULL )
        return ;

    m_pSprite->SetTransform( pmatTrans );           // 프로그램 종료로 스프라이트가 사라질 때까지 무한 loop!
}

void CSprite::OnDraw( LPDIRECT3DTEXTURE9 pTexture // LPDIRECT3DTEXTURE9: d3d9.h에 있는 텍스처
, const RECT * pRect                             // 관련 인터페이스 ID3DXSprite의 포인터 객체 - 확인하기
, float fX                                         // windef.h에 있는 텍스처 관련 사각형 구조체
, float fY )
{
    D3DXVECTOR3 vPos( fX, fY, 0.0f );              // 좌표 구하고
    m_pSprite->Draw( pTexture, pRect, &vPos, NULL, 0xffffffff );
    // 0xffffffff번 색깔로 해당 위치에 텍스처를 입혀서 그린다.
    // Draw함수도 ID3DXSprite에서 제공한다.
}
```

“

Draw Images

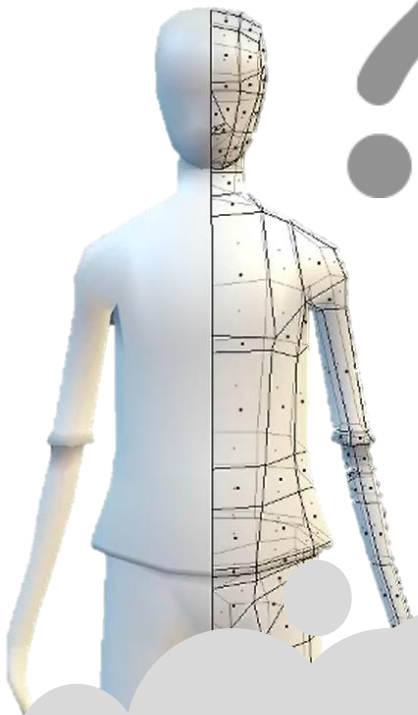
원하는 위치에 이미지 **출력**

”



클라이언트 부분

클래스 기능 소개 - **CTexture** Class



?



!



클라이언트 부분

클래스 기능 소개 - CTexture Class

```
int CTexture::Load( wstring wFile )
{
    map< wstring, STextureInfo* >::iterator mit = m_mapRefInfo.find( wFile );
    if( mit != m_mapRefInfo.end() )
    {
        ++mit->second->m_iRefCount;

        return mit->second->m_iTexture;
    }

    D3DXIMAGE_INFO TextureInfo;    // d3dx9tex.h 에서 정의된 이미지 정보 저장 구조체
    LPDIRECT3DTEXTURE9 pTexture;    // IDirect3DTexture9 인터페이스
    // 파일명에 해당하는 텍스처를 불러온다. 만일 S_OK 플래그가 아니라면 로드 오류이므로 비정상 종료 호출.
    if( D3DXCreateTextureFromFileEx( DXUTGetD3D9Device(), wFile.c_str(), D3DX_DEFAULT, D3DX_DEFAULT, 1,
        0, D3DFMT_A8R8G8B8, D3DPPOOL_MANAGED, D3DX_FILTER_NONE, D3DX_FILTER_NONE, 0, &TextureInfo, NULL,
        & pTexture ) != S_OK )
        return - 1;

    ++m_iTexture;    // 텍스처 개수 증감

    STextureInfo* pTextureInfo = new STextureInfo;    // 객체 동적 생성 후 초기화
    pTextureInfo->m_iTexture = m_iTexture;
    pTextureInfo->m_wFile = wFile;
    pTextureInfo->m_pTexture = pTexture;
    pTextureInfo->m_TextureInfo = TextureInfo;

    m_mapTexture.insert( make_pair( m_iTexture, pTextureInfo ) );    // 이미지에 텍스처를 삽입한다.
    m_mapRefInfo.insert( make_pair( wFile, pTextureInfo ) );

    return m_iTexture;
}

// Parameter 파일명에 해당하는 텍스처를 가져온다.
template<typename Type>
LPDIRECT3DTEXTURE9 CTexture::GetTexture(Type wFile)
{
    map< wstring, STextureInfo* >::iterator mit = m_mapRefInfo.find(wFile);
    if (mit == m_mapRefInfo.end())
        return NULL;

    return mit->second->m_pTexture;
}
```

“

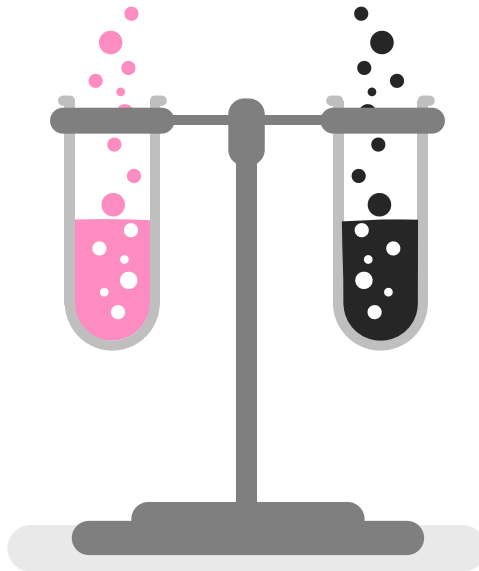
Insert Texture




이미지에 텍스처를 **씌운다**
렌더링 된 이미지에
텍스처를 씌워야
하나의 사물/캐릭터가 완성

”



네트워크 부분 네트워크 상황 보고



-  동기화(Synchronous) TCP
-  비동기화 TCP 서버
-  비동기 UDP echo 서버





네트워크 부분

네트워크 상황 보고

서버	클라이언트 1
<pre>C:\Users\cydph\source\repos\WP... 서버 시작... 클라이언트 접속 성공 SessionID: 0 클라이언트 로그인 성공 Name: 최성준 클라이언트 접속 성공 SessionID: 1 클라이언트 로그인 성공 Name: 조재권 클라이언트 접속 성공 SessionID: 2 클라이언트 로그인 성공 Name: 이현재</pre>	<pre>C:\Windows\system32\cmd.exe 서버 접속 성공 이름을 입력하세요!! 최성준 클라이언트 로그인 성공 ? : 1 Hello world 최성준: Hello world Hello world 최성준: Hello world 조재권: Good afternoon 이현재: Nice to meet you</pre>
클라이언트 2	클라이언트 3
<pre>C:\Windows\system32\cmd.exe 서버 접속 성공 이름을 입력하세요!! 조재권 클라이언트 로그인 성공 ? : 1 최성준: Hello world Good afternoon 조재권: Good afternoon 이현재: Nice to meet you</pre>	<pre>C:\Windows\system32\cmd.exe 서버 접속 성공 이름을 입력하세요!! 이현재 클라이언트 로그인 성공 ? : 1 최성준: Hello world 조재권: Good afternoon Nice to meet you 이현재: Nice to meet you</pre>



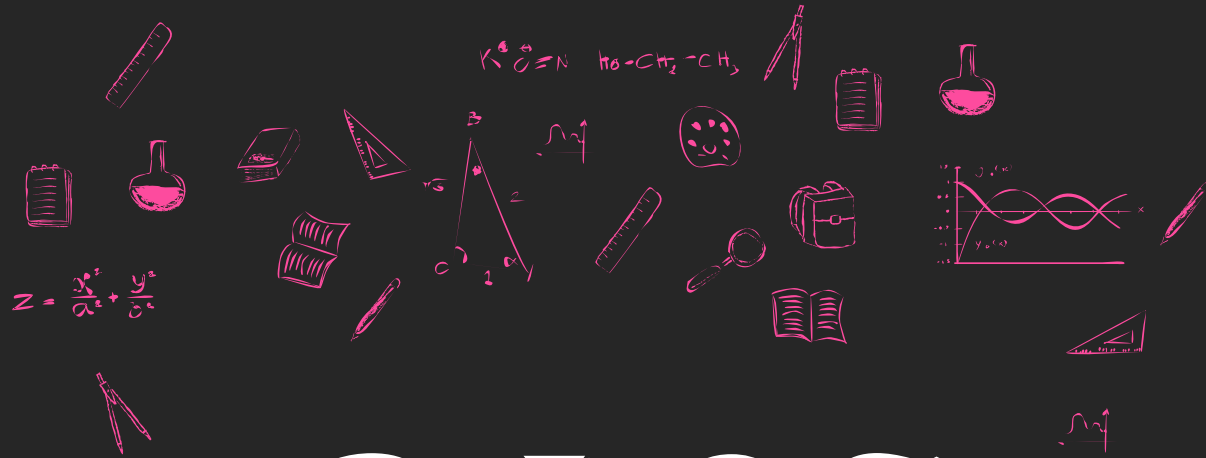
다음주 계획

01 클라이언트 클래스 추가 구현
Cdebug, FontMgr...etc

02 클라이언트와 네트워크부분의 가시적인 성과 보이기

03 서버 구축 공부, CGSF 공부

04 게임 컨셉 및 디자인에 대해 토의



BAOS

Thank You