

공개SW 프로젝트

BAOS 게임 제작

2조

2015112119 컴퓨터공학과 이현재

2015112084 컴퓨터공학과 조재권

2015112121 컴퓨터공학과 최성준

BAOS

Online Game

01

본론

클라이언트 진행사항 보고

02

계획

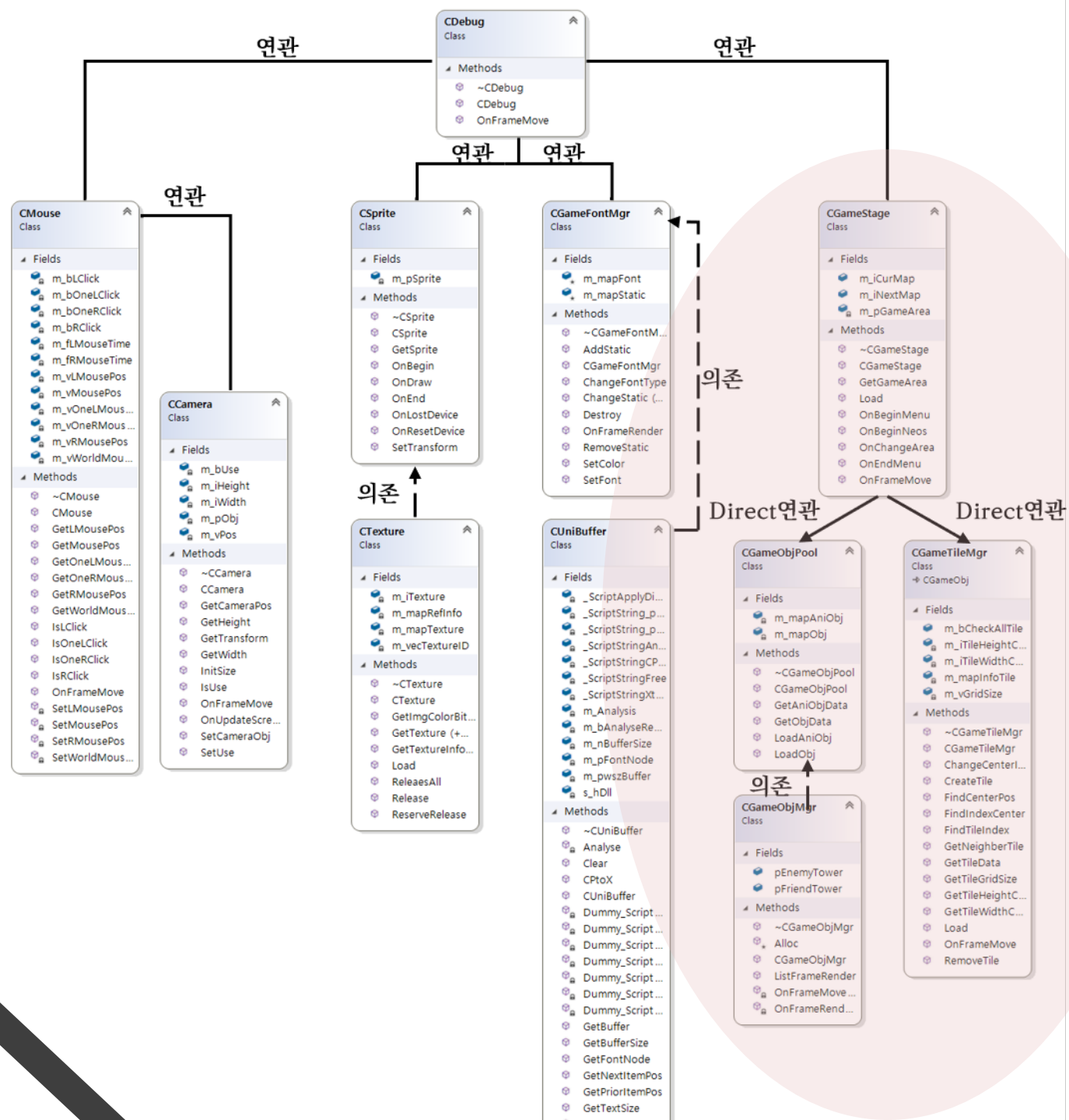
다음 주 계획 보고



클라이언트 진행상황 클래스 다이어그램



- 게임을 진행하기 위한 스테이지를 구현
- 게임 내 등장하는 모든 오브젝트를 관리





클라이언트 진행상황

클래스 기능소개 - CGameObjPool Class

“

Make object pool

오브젝트와 애니메이션을
미리 로드 해놓는다.

Mapping 구조로 로드 속도 감소.

아직 렌더링 된 오브젝트 &
애니메이션 미 구현

”



```
struct SobjData
{
    wstring m_wFile;
    D3DXVECTOR2 m_vScale;

    void Load( wstring wFile );
};

class CGameObjPool
{
private:
    // 각 오브젝트들을 미리 만들어 mapping 한다.
    map< wstring, SAniObjData > m_mapAniObj;
    map< wstring, SobjData > m_mapObj;
public:
    CGameObjPool(void);
    virtual ~CGameObjPool(void);
    // 애니메이션 오브젝트를 로드하는 함수
    void LoadAniObj(wstring wID, wstring wFile);
    // 일반 오브젝트를 로드하는 함수
    void LoadObj(wstring wID, wstring wFile);

    const SAniObjData* GetAniObjData(wstring wFile);
    const SobjData* GetObjData(wstring wFile);
};
```



클라이언트 진행상황

클래스 기능소개 - CGameObjMgr Class

```
enum
{
    EGAMEOBJ_TILE,           // 바닥 타일 오브젝트
    EGAMEOBJ_BEGINEFFECT,    // 이펙트 시작 오브젝트

    EGAMEOBJ_TOWER,          // 건축물 오브젝트
    EGAMEOBJ_PLAYER,         // 캐릭터 오브젝트
    EGAMEOBJ_CREATURE,       // 크리처(몬스터) 오브젝트

    EGAMEOBJ_EFFECT,         // 이펙트 종료 오브젝트

    EGAMEOBJ_HEALTH,         // 체력 관련 오브젝트
    EGAMEOBJ_DARK,           // 밝혀지지 않은 지도 부분 오브젝트
    EGAMEOBJ_UI,             // UI 관련 오브젝트
    EGAMEOBJ_MINIDARK        // 밝혀졌지만 회색 부분 오브젝트
};
```

“

게임에 등장하는
오브젝트들을 **생성**
명시적인 이름으로 구별

Set Objects

”

“

Manage Objects

화면에 등장하는 오브젝트들을 **관리**

벡터로 오브젝트들을 프레임에 따라 움직이게 한다.
오브젝트들을 순서대로 렌더링한다.

”

```
class CGameObjMgr           // 게임 오브젝트 매니저 클래스
{
private:
    void OnFrameMove( const vector< CGameObj* >& vecObj, float fElapsedTime );
    void OnFrameRender( const vector< CGameObj* >& vecObj );

protected:
    virtual CGameObj* Alloc( int iType );

public:
    CGameObj* pEnemyTower; // 적군 건축물 오브젝트 관리
    CGameObj* pFriendTower; // 아군 건축물 오브젝트 관리

public:
    CGameObjMgr(void);
    virtual ~CGameObjMgr(void);

    void OnFrameMove( float fElapsedTime );
    void OnFrameRender();
    void ListFrameRender();
};
```



클라이언트 진행상황

클래스 기능소개 - CGameTileMgr Class

```
struct STileData          // 타일 정보 구조체
{
    D3DXVECTOR2 m_vIndex;    // 타일 번호
    D3DXVECTOR2 m_vPos;      // 타일 위치
    D3DXVECTOR2 m_vIndexCenter; // 타일 중앙

    int m_iTileType;

    STileData():
    {
        m_vIndex( 0.0f, 0.0f ),
        m_vPos( 0.0f, 0.0f ),
        m_vIndexCenter( 0.0f, 0.0f ),
        m_iTileType( -1 )
    }
};
```

“

Set Tile

맵을 만들기 위한 타일 구조체 **생성**

지정된 좌표에 타일을 생성한다.

”

“

Manage Objects

타일을 만들기 위한 **관리** 클래스

”

```
class CGameTileMgr : public CGameObj
{
private:
    map< wstring, STileData > m_mapInfoTile; // 타일정보를 저장한 맵 배열

    D3DXVECTOR2 m_vGridSize; // 타일 사이즈
    int m_iTileWidthCount; // 타일 개수
    int m_iTileHeightCount; // 타일개수

public:
    CGameTileMgr(void);
    virtual ~CGameTileMgr(void);

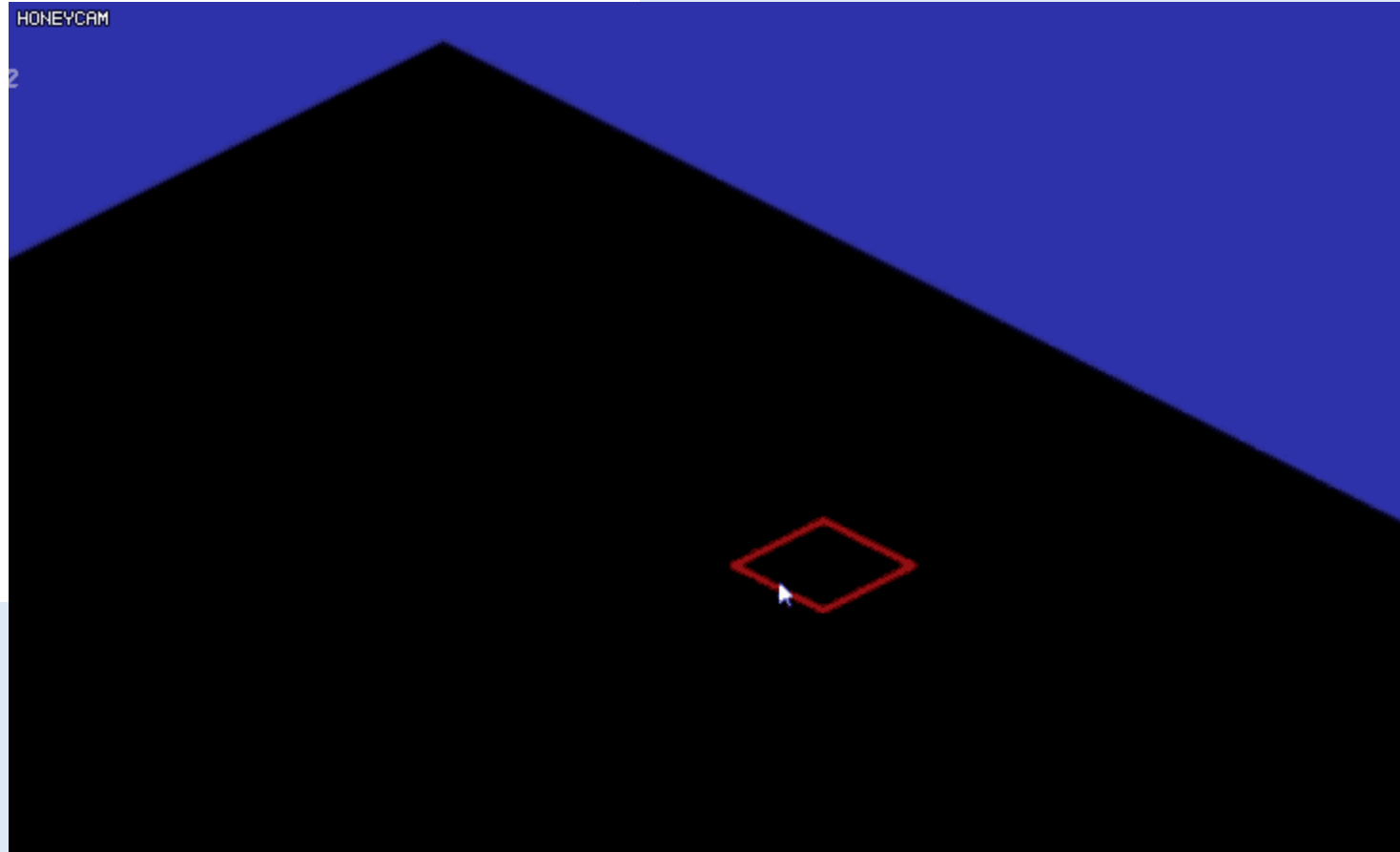
    bool m_bCheckAllTile; // 모든 타일을 불러왔는가?

    void Load( wstring wFile );
    virtual void OnFrameMove( float fElapsedTime );
```



클라이언트 진행상황

클래스 기능소개 - CGameTileMgr Class



디버깅 과정에서 리얼타임으로 타일과 오브젝트 **생성**



클라이언트 진행상황

클래스 기능소개 - CGameStage Class

“

Make object pool

맵을 만들기 위한 클래스

외부에서 맵을 가져오거나
직접 맵을 만들 수 있다

”

```
class CGameArea;

class CGameStage
{
private:
    CGameArea* m_pGameArea;    // 맵을 객체화

public:
    int m_iCurMap;             // 현재 맵 번호
    int m_iNextMap;            // 다음 맵 번호

public:
    CGameStage(void);
    virtual ~CGameStage(void);

    CGameArea* GetGameArea() { return m_pGameArea; }

    void Load();

    void OnFrameMove( float fElapsedTime );
    void OnChangeArea( int iMap );

public:
    static void OnBeginMenu( DWORD dwObj );
    static void OnEndMenu( DWORD dwObj );

    static void OnBeginNeos( DWORD dwObj );

};
```



다음 주 계획 보고

소개한 클래스들 제대로 구현 하기

게임 디자인 및 컨셉 구체화



엔진
개발

THANK YOU