

목 차

I. 프로그래밍	1~42
I - i. 학교 프로젝트	1~13
I - ii. 학원 프로젝트	14~42
I - iii. 팀단위 프로젝트	43~63
II. 대외 활동	64~72
II - i. 멘토링	64~65
II - ii. 현장실습	66~66
II - iii. 연구실	67~67
II - iiiii. 인턴생활	68~69
II - iiiiii. 동아리	70~71
II - iiiiii. 포트폴리오 동영상 링크	72~72

I - i. 학교 프로젝트 - 1

수강과목	Window API	담당교수	송인희 교수님
수강학기	2015년 1학기		
프로젝트명	손오공의 하루(드래곤볼)		
주요내용	윈도우 상에서의 프로그래밍 기법을 배우고 윈도우 환경의 생성 및 관리, 메시지 처리기법, GD(graphic Device Interface)사용, 비트맵 그래픽, 파일처리, 멀티쓰레딩 등의 Win32 API에 대한 내용을 배우며 윈도우 응용 기술을 게임에 적용하여 프로그램 하는 방법 들을 배워 2D 게임의 프로그래밍 과정을 이해한다.		

1) LogoScene



2) Skill1



3) Skill12



4) Skill13



5) Boss1



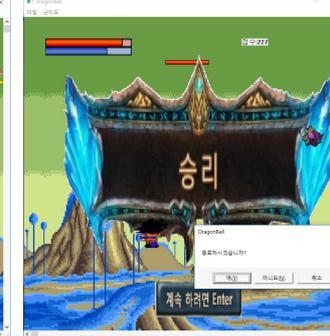
6) Boss2



7) Boss3



8) Ending



결과	이 프로젝트를 수행하면서 2D 게임의 프로그래밍에 대한 전반적인 흐름을 알 수 있었고, 좀 더 깊은 내용을 이해해보고 싶은 욕심이 생기게 되었다. 비록 큰 작품은 아니었지만 스스로의 힘으로 프로그래밍을 완성한 '첫 번째 게임' 이라는것에 만족감을 느끼며 창의력과 아이디어를 엿볼 수 있는 기회가 될 수 있었다.
----	---

윈도우API 소스 코드(C++언어)

1) 구충돌 소스 코드 일부

```
#define BSIZE 30 // 반지름

double LengthPts(int x1, int y1, int x2, int y2) {
    return(sqrt((float)((x2 - x1)*(x2 - x1) + (y2 - y1)*(y2 - y1))));
}

BOOL meet_Circle(int x, int y, int mx, int my)
{
    if (LengthPts(x, y, mx, my) <= BSIZE + 10) return TRUE;
    else return FALSE;
}
```

2) 더블버퍼링 소스 코드 일부

```
GetClientRect(hwnd, &rectView);
hdc = BeginPaint(hwnd, &ps);
memIdc = CreateCompatibleDC(hdc);
// hBit1에는 배경과 텍스트가 출력된 비트맵이 저장, memIdc에 설정
oldBit1 = (HBITMAP)SelectObject(memIdc, hBit1);
// memIdc에 있는 내용을 hdc에 부러준다.
BitBlt(hdc, 0, 0, 1024, 768, memIdc, 0, 0, SRCOPY);
SelectObject(memIdc, oldBit1);
DeleteDC(mem2dc);
```

3) 플레이어 스킬 소스 코드

```
if (GetAsyncKeyState('Q') & 0x8000)
{
    /*PlaySound(MAKEINTRESOURCE(IDR_WAVE8), q_hInst, SND_RESOURCE | SND_ASYNC);
    */if (Mp > 1)
    {
        state = 5;
        Mp--;
        bool isMissileAddAble;
        isMissileAddAble = FALSE;
        for (int i = 0; i < MAX_MISSILE; ++i)
        {
            if (Missiles[i] == NULL)
            {
                isMissileAddAble = TRUE;
                Missiles[i] = new Missile(rect, left, rect.top);
                break;
            }
        }
    }
}
```

4) 보스몬스터 미사일 발사 소스 코드

```
bool isBossMissileAddAble;
isBossMissileAddAble = FALSE;
for (int i = 0; i < MAX_MISSILE; ++i)
{
    if (BossMissiles[i] == NULL)
    {
        isBossMissileAddAble = TRUE;
        BossMissiles[i] = new Missile(BossMissilesMove.left, BossMissilesMove.top);
        break;
    }
}
BossMissilesMove.left = rand() % 730;
break;
```

5) 게임Update로직 소스 코드 일부

```
case WM_TIMER:
switch (wParam)
{
case 1:
{
    if (!GameOver && BossMCheck >= 3)
    {
        GameOver = TRUE;
        KillTimer(hwnd, 2);
        KillTimer(hwnd, 3);
        KillTimer(hwnd, 4);
        KillTimer(hwnd, 5);
    }
    if (GetAsyncKeyState(VK_LEFT) & 0x8000) {
        state = 4;
        rect.left -= 15;
        rect.right -= 15;
        if (rect.left < -20)
            rect.left = -20;
        InvalidateRect(hwnd, NULL, FALSE);
    }
    if (GetAsyncKeyState(VK_RIGHT) & 0x8000) {
        state = 2;
        rect.left += 15;
        rect.right += 15;

        if (rect.right > 699)
        {
            rect.right = 700;
            rect.left = 699;
        }
    }
}
```

6) 몬스터 클래스 멤버 함수 변수 소스 코드

```
class Monster
{
private:
    int m_nPosX;
    int m_nPosY;
    int m_hp;
public:
    Monster(int x, int y, int hp) : m_nPosX(x), m_nPosY(y), m_hp(hp)
    {
    }
    int GetX() { return m_nPosX; }
    int GetY() { return m_nPosY; }
    int GetHP() { return m_hp; }
    void DamageSet()
    {
        m_hp--;
    }
    bool Update()
    {
        m_nPosY += minionvelocity;
        if (m_nPosY > 800)
            return TRUE;
        return FALSE;
    }
    ~Monster(){ jumsucheck++; }
};
```

I - i. 학교 프로젝트 - 2

수강과목	Window API	담당교수	송인희 교수님
수강학기	2015년 1학기		
프로젝트명	물고기키우기		
주요내용	TransBlit함수를 이용하여 애니메이션 효과를 주고 구충돌을 이용하여 물고기 키우기 게임을 구현하였다. 플레이어 물고기를 제외한 물고기들은 윈도우상 좌우끝에서 Y값은 랜덤으로 스폰되며 자신보다 하위단계의 물고기만 먹으며 성장하는 게임이다.		

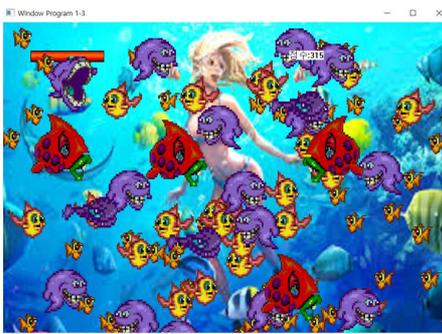
1) 초기 화면



2) 물고기를 먹는 장면



3) 1단계 진화



4) 2단계 진화



5) 3단계 진화



6) 최종 진화



결과

Window API를 통해 만든 2번째 게임으로 C++의 상속구조에 대해 이해할 수 있었다. 이를 통해 가상함수와 함수 오버라이딩에 대해 이해하였고, 그래픽적인 요소를 처리하는데 어떠한 함수가 필요한지에 대해서 알 수 있었고, 이를 효과적으로 활용하기 위해선 수학적 개념이 많이 필요하다는 사실을 알 수 있었다.

I - i. 학교 프로젝트 - 3

수강과목	Window API	담당교수	송인희 교수님
수강학기	2015년 1학기		
프로젝트명	포켓몬스터		
주요내용	ChildWindow를 이용하여 UI를 만들고 Scene전환이라는 기법을 사용하여 다른맵을 이동하며 클래스 상속을 통해 Monster라는 하위 자식Class내부를 구현하였다. 마우스 충돌을 하였으며 윈도우 좌표계상에 비충돌처리를 구현하였다.		

1) 초기 화면



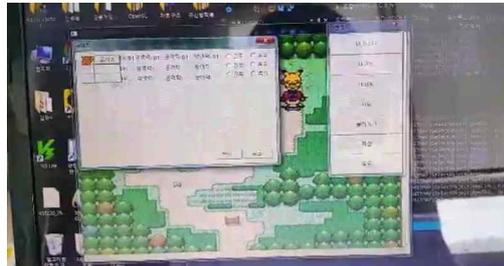
2) 마을 내부



3) 마을 외부



4) 도감 및 상태창(UI)



5) 배틀 이펙트



6) Battle Scene



7) 병원



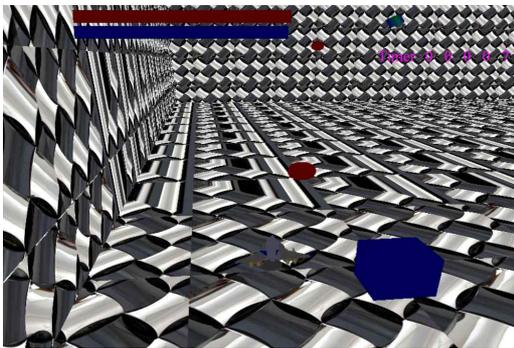
결과

API 환경에서 실제 포켓몬스터 인게임과 UI를 비슷하게 구현하였고 맵이동도 해보았으나 C++상속 구조에 대한 이해도가 부족하여 몬스터의 개수를 많이 늘리는데 한계가 있었음에 아쉬움이 들었다.

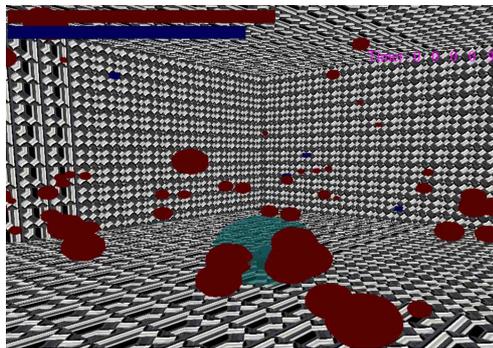
I - i. 학교 프로젝트 - 4

수강과목	컴퓨터 그래픽스	담당교수	송인희 교수님
수강학기	2015년 2학기		
프로젝트명	3차원 총알피하기		
주요내용	그래픽스의 기본 입출력, 그래픽 환경 설정, 2차원과 3차원에서의 물체의 표현 및 변환 등 그래픽스의 기본적인 개념과 특성, 그리고 알고리즘들에 대하여 학습한다. 또한 색상의 표현 및 렌더링 기법에 관하여 배운다. 이게임에서는 4개의 아이템이있다. HP,MP,무적,분쇄 아이템이있는데 분쇄 아이템을 먹었을 경우 모든 총알들이 분쇄되며 객체를 삭제한다.		

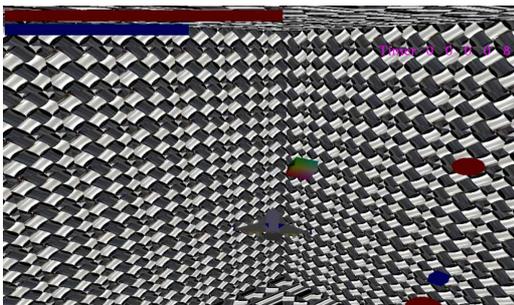
1) 초기화면



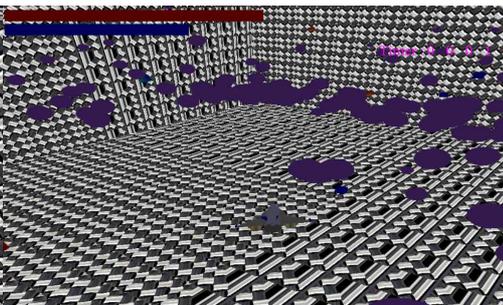
2) 방어막 아이템



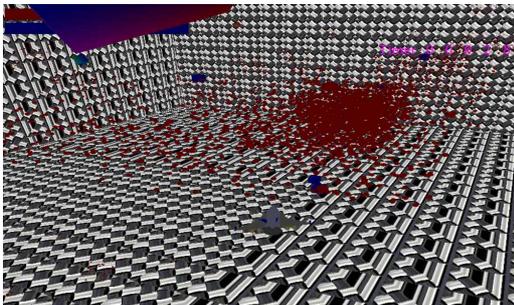
2) 분쇄 아이템



3) 분쇄아이템 발동직전



4) 분쇄아이템 발동직후



5) 게임종료



결과

OpenGL 코드가 처음엔 이해가 안 되어 조금 힘들었지만 결국, 그 원리를 이해하여 프로그래밍을 해보니 그리 어렵지 않게 구현 할 수 있었다. 이를 통해 3차원 객체가 돌아가는 원리에 대해서 이해 할 수 있어서 좋은 기회가 되었고, OpenGL뿐만 아니라 DirectX도 프로그래밍을 해봐야겠다는 생각이 들었다.

OpenGL 소스 코드(C++언어)

1) 초기화 소스 코드

```
void main(int argc, char *argv[]) { // 초기화 함수들

    srand((unsigned)time(NULL));
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA | GLUT_DEPTH); // 디스플레이 모드 설정
    glutInitWindowSize(100, 100);
    glutInitWindowPosition(WINDOWS_SIZE_WIDTH, WINDOWS_SIZE_HEIGHT);
    glutCreateWindow("그래픽스 실습");
    SetupRC();

    // 콜백 함수 호출
    glutDisplayFunc(drawScene);
    glutReshapeFunc(Reshape);
    glutSpecialFunc(SpecialKeyboard);
    glutKeyboardFunc(Keyboard);
    glutTimerFunc(FRAME_60FPS, TimerFuncIdh, 1);
    glutMainLoop();
}
```

2) 3차원 맵 생성 코드

```
for (int i = 0; i < 4; ++i)
{
    for (int j = 0; j < MapSize; ++j)
    {
        glColor3ub(255, 255, 0);
        glBegin(GL_POLYGON);
        glVertex3f(Map[i + 1][j].x, Map[i][j].y, Map[i][j].z);
        glVertex3f(Map[i + 1][j].x, Map[i][j].y, Map[i][j + 1].z);
        glVertex3f(Map[i][j].x, Map[i][j].y, Map[i][j + 1].z);
        glVertex3f(Map[i][j].x, Map[i][j].y, Map[i][j].z);
        glEnd();

        glColor3ub(255, 0, 125);
        glBegin(GL_POLYGON);
        glVertex3f(Map[i][j].x + 10, Map[i][j].y, Map[i][j].z);
        glVertex3f(Map[i][j].x + 10, Map[i][j].y, Map[i][j + 1].z);
        glVertex3f(Map[i][j].x, Map[i][j].y, Map[i][j + 1].z);
        glVertex3f(Map[i][j].x, Map[i][j].y, Map[i][j].z);
        glEnd();
    }
}
```

3) 플레이어와 총알간의 충돌체크 코드

```
//미사일과 몬스터 충돌체크
for (int i = 0; i < MAX_MISSILE; ++i)
{
    for (int j = 0; j < MAX_MONSTER; ++j)
    {
        if (Missiles[i] != NULL)
        {
            if (Monsters[j] != NULL)
            {
                if (meet_Circle(Missiles[i]->GetX(), Missiles[i]->GetZ(), Monsters[j]->GetX(), Monsters[j]->GetZ()))
                {
                    if (meet_Circle(Missiles[i]->GetX(), Missiles[i]->GetY(), Monsters[j]->GetX(), Monsters[j]->GetY()))
                    {
                        Monsters[j]->DamageSet();
                        if (Monsters[j]->GetHP() == 0)
                        {
                            delete Monsters[j];
                            Monsters[j] = NULL;
                        }
                        delete Missiles[i];
                        Missiles[i] = NULL;
                    }
                }
            }
        }
    }
}
```

4) 몬스터생성 코드

```
//몬스터생성
bool isMonsterAddAble;
isMonsterAddAble = FALSE;
for (int i = 0; i < MAX_MONSTER; ++i)
{
    if (Monsters[i] == NULL)
    {
        isMonsterAddAble = TRUE;
        Monsters[i] = new Monster(jangemove.moveX, jangemove.moveY, jangemove.moveZ, 2);
        break;
    }
}
```

5) 배경 텍스처 설정 코드

```
//텍스처 설정 정의를 한다. — (3)
glTexImage2D(GL_TEXTURE_2D, 0, 3, 736, 459, 0, GL_BGR_EXT, GL_UNSIGNED_BYTE, pBytes[0]);
//텍스처 파라미터 설정 — (4)
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);
// 나머지 n-1개의 텍스처에도 (1) ~ (4)까지의 과정을 진행하여 텍스처를 설정한다.
// 텍스처 모드 설정
glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_COLOR, GL_MODULATE);
// 텍스처 매핑 활성화
glEnable(GL_TEXTURE_2D);
// 텍스처를 객체에 맵핑
//glBindTexture(GL_TEXTURE_2D, name[0]);
glBegin(GL_QUADS);
```

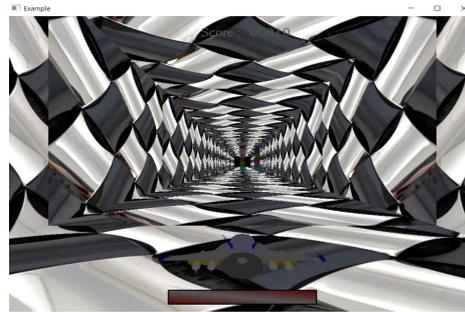
I - i. 학교 프로젝트 - 5

수강과목	OpenGL(컴퓨터그래픽스)	담당교수	송인희 교수님
수강학기	2015년 2학기		
프로젝트명	Block Breaker		
주요내용	1인칭 시점으로 3D 공간을 빠른 속도로 앞으로 나가게 됩니다. 눈 앞에 나타나는 장애물을 좌우 움직이면서 피해 나가야 한다. 가속이 붙으면서 점점 빨라지는 높은 속도감과 장애물을 피해가는 긴장감을 느끼는 게임이다. 아이템은 총 3가지이며 HP,방어막,무적 아이템이있다.		

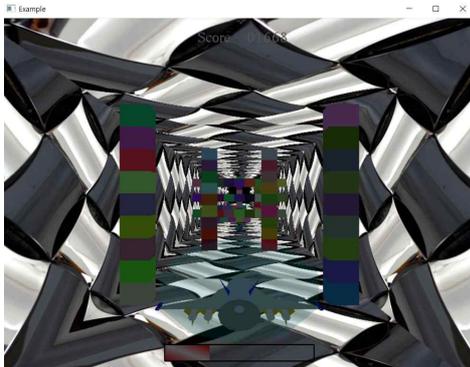
1) 로고 화면



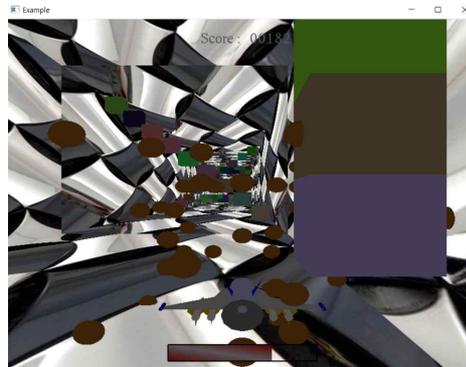
2) 초기 화면



3) 장애물 패턴



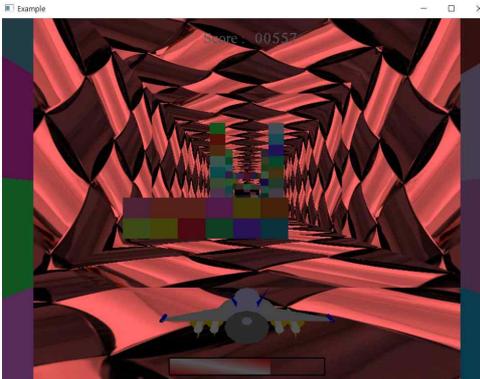
4) 충돌시 파티클



5) 방어막 아이템



6) 무적 아이템



결과

처음으로 팀단위 프로젝트이며 3차원에서의 물체 표현 및 개념을 학습했고 색상의 표현기법인 렌더링 기법에 대해 이해하였다. 2명에서 만든 프로젝트이며 나는 플레이어, 충돌체크,아이템, 파티클 등등을 만들었으며 팀작업할 때 역할분담이 중요하다는 것을 깨닫게 해준 프로젝트였다.

OpenGL 소스 코드(C++언어)

1) 장애물 초기화 소스 코드

```
for (int j = 0; j < 1000; ++j)
{
    glBegin(GL_LINE_STRIP);
    for (float i = 0; i <= 2 * PI + 0.1; i += 0.1)
    {
        glColor3f(circle[j].r, circle[j].g, circle[j].b);
        glVertex3f(250 * cos(i), 250 * sin(i), -j * 10);
    }
    glEnd();
}
for (int i = 0; i < 3; ++i)
{
    glRotatef(-10 * (3 - i), 0, 0, 1);
    glTranslatef(60 * (3 - i) - 30, 3 * (3 - i), 0);
    for (int j = 0; j < 50; ++j)
    {
        glBegin(GL_POLYGON);
        glRotatef(30, 0, 0, 1);
        glVertex3f(-190 + (i * 35), -60, 330 - (j * 70));
        glVertex3f(-190 + (i * 35), -60, 390 - (j * 70));
        glVertex3f(-160 + (i * 35), -60, 390 - (j * 70));
        glVertex3f(-160 + (i * 35), -60, 330 - (j * 70));
        glEnd();
    }
}
```

3) 카메라 조명 셋팅 소스 코드

```
//----- 화면 회전 -----
glTranslatef(cam.moveX, cam.moveY, cam.moveZ);
glRotatef(cam.angleX, 1, 0, 0);
glRotatef(cam.angleY, 0, 1, 0);
glRotatef(cam.angleZ, 0, 0, 1);

//----- 조명 셋팅 -----
glLightfv(GL_LIGHT0, GL_AMBIENT, AmbientLight);
glLightfv(GL_LIGHT0, GL_DIFFUSE, DiffuseLight);
glLightfv(GL_LIGHT0, GL_SPECULAR, SpecularLight);
glLightfv(GL_LIGHT0, GL_POSITION, lightPos);
glEnable(GL_LIGHT0);
//-----
```

5) 점수 UI출력 소스코드

```
void jumsu()
{
    for (int j = 0; j < 3; ++j)
    {
        for (int i = 0; i < 9; ++i)
        {
            glPushMatrix();
            {
                glTranslatef(50 * j, 0, 0);
                glColor3f(1, 0, 0);
                glRasterPos3f(-270, 210, 0);
                int len = (int)strlen(string[j][i]);
                for (int k = 0; k < len; k++)
                    glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18, string[j][i][k]);
            }
            glPopMatrix();
        }
    }
}
```

2) 플레이어 기체 생성 코드 일부

```
// 미사일 꼬리 날개
glPushMatrix();
{
    glColor3f(1.0, 0.8, 0.0);
    glBegin(GL_QUADS);
    // 윗날개
    glVertex3f(0, 20, 8);
    glVertex3f(0, 8, 15);
    glVertex3f(0, 8, 0);
    glVertex3f(0, 20, 0);
    // 왼쪽 날개
    glVertex3f(-6, -5, 15);
    glVertex3f(-16, -16, 8);
    glVertex3f(-16, -16, 0);
    glVertex3f(-6, -5, 0);
    // 오른쪽 날개
    glVertex3f(6, -5, 15);
    glVertex3f(16, -16, 8);
    glVertex3f(16, -16, 0);
    glVertex3f(6, -5, 0);
    glEnd();
}
glPopMatrix();
```

4) 뷰행렬,원근투영행렬 갱신 소스 코드

```
GLvoid Reshape(int w, int h)
{
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    //glOrtho(0, 900, 0, 600, -1, 1);
    gluPerspective(60, w / h, 1, 3000);
    gluLookAt(0, 0, 500, 0, 0, 0, 1, 0);
    glMatrixMode(GL_MODELVIEW);
}
```

I - i. 학교 프로젝트 - 6

수강과목	2DGameProgramming(Python)	담당교수	이대현 교수님
수강학기	2015년 2학기		
프로젝트명	좀비vs식물		
주요내용	2D 게임 개발에 있어서 꼭 필요한 요소인 스프라이트, 애니메이션, 충돌 검사, 사운드 구현, 스크롤링과 타이핑, 마우스 및 키보드 입력 처리, 게임 프레임워크 등의 구현 기법을 실습을 통해서 익히고, 이를 기반으로 직접 2D 게임을 제작해봄으로써 게임 개발의 기초를 완성한다. Python 언어와 pico2d 라이브러리를 이용하여 객체지향적인 방법으로 개발한다.		

1) 로고 화면



2) 포탄쏘는UI



4) 상점 무기Upgrade



4) 중간보스



5) 최종보스



6) 엔딩화면



결과

파이썬 언어에 대해 처음엔 어색했지만 2D게임에 대한 기본적인 게임 로직과 이미지 처리, 충돌처리, 스크롤링, 타이핑, 애니메이션, 사운드 등등을 배우며 전반적인 게임 프레임워크에 대한 공부가 되었다. 새로운 언어와 라이브러리인 Python, pico2d를 사용함으로써 게임을 꼭 c++으로만 개발하지 않아도 되는 것을 깨달았다. python에 장점인 문법이 간단하고 프로그래밍을 할 때 다른언어보다 직관적인 것을 깨달았다.

Python 소스 코드

1) 좀비 클래스 소스 코드

```
class Zombie:
    image = None
    def __init__(self):
        self.x, self.y = 790, random.randint(50,550)
        self.frame = 0
        self.jombie1 = load_image('좀비1.png')
        self.dir = 1
    def update(self):
        self.frame = (self.frame + 1) % 8
        self.x += self.dir*0.3
    def draw(self):
        self.jombie1.clip_draw(self.frame + 95, 370, 100, 110, self.x, self.y)

global x, y
def handle_events():
    events = get_events()
    for event in events:
        if event.type == SDL_QUIT:
            game_framework.quit()
        elif event.type == SDL_KEYDOWN and event.key == SDLK_ESCAPE:
            game_framework.change_state(title_state)
```

2) 식물 랜더링 소스 코드 일부

```
def draw(self):
    global i, j
    self.image.draw(400, 300)
    self.gume.clip_draw(0, 0, 570, 69, self.gumex*30, self.gumey)
    self.plant.clip_draw(0, 250, 50, 50, self.plantx, self.planty+self.plantsize)
    self.plant.clip_draw(50, 250, 50, 50, self.plantx1, self.planty1+self.plantsize1)
    self.plant.clip_draw(250, 250, 50, 50, self.plantx2, self.planty2+self.plantsize2)
    self.plant.clip_draw(300, 250, 50, 50, self.plantx3, self.planty3+self.plantsize3)
    self.plant.clip_draw(0, 0, 50, 50, self.plantx4, self.planty4+self.plantsize4)
    self.plant.clip_draw(0, 200, 50, 50, self.plantx5, self.planty5+self.plantsize5)
    self.erase.clip_draw(0, 0, 95, 58, self.plantx6, self.planty6+self.plantsize6)
    self.menu.clip_draw(0, 0, 110, 26, self.plantx7, self.planty7+self.plantsize7)

if (self.state!=0 and self.state!=6 and self.state!=7):
    for i in range(0,9):
        for j in range(0,5):
            self.plantspace.draw(60+80*(i), 80+100*(j))

if (self.state==1):
    self.myplant.clip_draw(100, 180, 100, 100, x, y)
elif (self.state==2):
    self.myplant.clip_draw(0, 180, 100, 100, x, y)
elif (self.state==3):
    self.myplant.clip_draw(300, 180, 100, 100, x, y)
elif (self.state==4):
    self.myplant.clip_draw(350, 0, 100, 100, x, y)
elif (self.state==5):
    self.myplant.clip_draw(400, 180, 100, 100, x, y)
```

3) 식물 설치 소스 코드

```
def update(self):
    global x, y
    events = get_events()
    for event in events:
        if event.type == SDL_MOUSEMOTION:
            x, y = event.x, 600 - event.y
            if (x>self.plantx*25 and x<self.plantx*25 and y>self.planty*25 and y<self.planty*25):
                self.state=1
            elif (x>self.plantx1*25 and x<self.plantx1*25 and y>self.planty1*25 and y<self.planty1*25):
                self.state=2
            elif (x>self.plantx2*25 and x<self.plantx2*25 and y>self.planty2*25 and y<self.planty2*25):
                self.state=3
            elif (x>self.plantx3*25 and x<self.plantx3*25 and y>self.planty3*25 and y<self.planty3*25):
                self.state=4
            elif (x>self.plantx4*25 and x<self.plantx4*25 and y>self.planty4*25 and y<self.planty4*25):
                self.state=5
            elif (x>self.plantx5*25 and x<self.plantx5*25 and y>self.planty5*25 and y<self.planty5*25):
                self.state=0
            elif (x>self.plantx6*25 and x<self.plantx6*25 and y>self.planty6*25 and y<self.planty6*25):
                self.state=6
            elif (x>self.plantx7*25 and x<self.plantx7*25 and y>self.planty7*25 and y<self.planty7*25):
                self.state=7
        else:
            self.plantsize=0
            self.plantsize1=0
            self.plantsize2=0
            self.plantsize3=0
            self.plantsize4=0
            self.plantsize5=0
            self.plantsize6=0
            self.plantsize7=0
```

3) 초기화 소스 코드

```
def __init__(self):
    self.gumex, self.gumey, self.gumesize = 310,565,0
    self.changstate=0
    self.plantx, self.planty, self.plantsize= 130,565,0
    self.plantx1, self.planty1, self.plantsize1= 190,565,0
    self.plantx2, self.planty2, self.plantsize2= 250,565,0
    self.plantx3, self.planty3, self.plantsize3= 310,565,0
    self.plantx4, self.planty4, self.plantsize4= 370,565,0
    self.plantx5, self.planty5, self.plantsize5= 430,565,0
    self.plantx6, self.planty6, self.plantsize6= 620,565,0
    self.plantx7, self.planty7, self.plantsize7= 790,580,0
    self.click=0
    self.state=0
    self.image = load_image('맹.png')
    self.gume = load_image('구매창.png')
    self.plant = load_image('구매식물.png')
    self.myplant = load_image('타워.png')
    self.plantspace = load_image('식물자리.png')
    self.erase = load_image('파괴아이콘.png')
    self.menu = load_image('메뉴창.png')
    self.jumsu = load_image('숫자.png')
```

4) 프레임워크 상태 소스 코드

```
class GameState:
    def __init__(self, state):
        self.enter = state.enter
        self.exit = state.exit
        self.pause = state.pause
        self.resume = state.resume
        self.handle_events = state.handle_events
        self.update = state.update
        self.draw = state.draw

    def quit():
        global running
        running = False

    def run(start_state):
        global running, stack
        running = True
        stack = [start_state]
        start_state.enter()
        while (running):
            stack[-1].handle_events()
            stack[-1].update()
            stack[-1].draw()
            # repeatedly delete the top of the stack
            while (len(stack) > 0):
                stack[-1].exit()
                stack.pop()
        |
    def test_game_framework():
        start_state = TestGameState('StartState')
        run(start_state)
```

5) 프레임워크 변경 소스 코드

```
def change_state(state):
    global stack
    pop_state()
    stack.append(state)
    state.enter()

def push_state(state):
    global stack
    if (len(stack) > 0):
        stack[-1].pause()
    stack.append(state)
    state.enter()

def pop_state():
    global stack
    if (len(stack) > 0):
        # execute the current state's exit function
        stack[-1].exit()
        # remove the current state
        stack.pop()

    # execute resume function of the previous state
    if (len(stack) > 0):
        stack[-1].resume()
```


유니티 소스 코드(C#)

1) 총알 충돌 소스 코드 일부

```
void OnCollisionEnter(Collision collision)
{
    //탄환제거하기.
    Destroy(this.gameObject);

    if (collision.gameObject.tag == "Enemy")
    {
        //Enemy스크립트에 접근하기.
        Enemy enemy = collision.gameObject.GetComponent<Enemy>();

        //Enemy 스크립트의 ES가 Die가 아닌 경우 Hurt 함수 실행.
        if (enemy.ES != EnemyState.Die)
        {
            enemy.Hurt(Power);
        }
    }
}
```

2) 플레이어 애니메이션 소스 코드 일부

```
void AnimationUpdate()
{
    if (PS == PlayerState.Idle)
    {
        animation.CrossFade(Idle_Ani.name, 0.2f);
    }
    else if (PS == PlayerState.Walk)
    {
        animation.CrossFade(Walk_Ani.name, 0.2f);
    }
    else if (PS == PlayerState.Run)
    {
        animation.CrossFade(Run_Ani.name, 0.2f);
    }
    else if (PS == PlayerState.Attack)
    {
        animation.CrossFade(Idle_Ani.name, 0.2f);
    }
    else if (PS == PlayerState.Dead)
    {
        animation.CrossFade(Idle_Ani.name, 0.2f);
    }
}
```

3) 포톤 네트워크 연결 소스 코드 일부

```
if (PhotonNetwork.connectionStateDetailed != PeerState.Joined)
{
    // only use PhotonNetwork.Instantiate while in a room.
    return;
}

switch (InstantiateType)
{
    case 0:
        PhotonNetwork.Instantiate(Prefab.name, InputToEvent.inputHitPos + new Vector3(0, 5f, 0), Quaternion.identity, 0);
        break;
    case 1:
        PhotonNetwork.InstantiateSceneObject(Prefab.name, InputToEvent.inputHitPos + new Vector3(0, 5f, 0), Quaternion.identity, 0, null);
        break;
}
```

5) 궁게이지 소스 코드 일부

```
float Dvalue = m_Value.AsFloat - TransitoryValue;
if (Dvalue > 0)
{
    TransitoryValue += ProgressSpeed * Time.deltaTime;
    if (TransitoryValue > m_Value.AsFloat)
        TransitoryValue = m_Value.AsFloat;
}
else if (Dvalue < 0)
{
    TransitoryValue -= ProgressSpeed * Time.deltaTime;
    if (TransitoryValue < m_Value.AsFloat)
        TransitoryValue = m_Value.AsFloat;
}
```

6) 몬스터 move 소스 코드

```
void DistanceCheck()
{
    if (Vector3.Distance(Player.position, transform.position) >= FindRange && tome == false)
    {
        ES = EnemyState.Idle;
        anim.SetBool("Run", false);
        Speed = 0;
    }
    else
    {
        ES = EnemyState.Move;
        anim.SetBool("Run", true);
        Speed = MoveSpeed;
    }
}
```

7) 아이템 충돌 소스 코드

```
if (collision.gameObject.tag == "Player_Red")
{
    Debug.Log("HP먹음!");
    PC.hp += hpup;
    HC.wait = true;
    if (PC.Max_hp < PC.hp)
    {
        PC.hp = PC.Max_hp;
    }
    PC.LifeBar.sliderValue = PC.hp / PC.Max_hp;
}
```

8) 플레이어 move 소스 코드

```
void Update ()
{
    if (Input.GetKey(KeyCode.LeftArrow) ||
        Input.GetKey(KeyCode.RightArrow) ||
        Input.GetKey(KeyCode.UpArrow) ||
        Input.GetKey(KeyCode.DownArrow))
    {
        float xx = Input.GetAxisRaw("Vertical");
        float zz = Input.GetAxisRaw("Horizontal");
        lookDirection = xx * Vector3.forward + zz * Vector3.right;

        this.transform.rotation = Quaternion.LookRotation(lookDirection);
        this.transform.Translate(Vector3.forward * MoveSpeed * Time.deltaTime);
    }
}
```

I - ii. 학원 프로젝트 - 1

수강과목	WINAPI(C++)	담당선생님	이승준
수강학기	2017.03.01.~ 2017.04.01		
프로젝트명	메이플스토리		
주요내용	이번 프로젝트를 위해서 싱글톤패턴, 추상팩토리패턴, mediate패턴을 사용하였고 비트맵이미지처리, Line충돌, CScene전환, 몬스터패턴, 충돌체크, UI, 퀘스트 등등을 C++로 구현하였다.		

1) 로고화면



2) 상점과 아이템시창



3) 스킬 이펙트



4) 충돌체크시 데미지 이펙트



5) Level Up



6) 충돌박스



결과

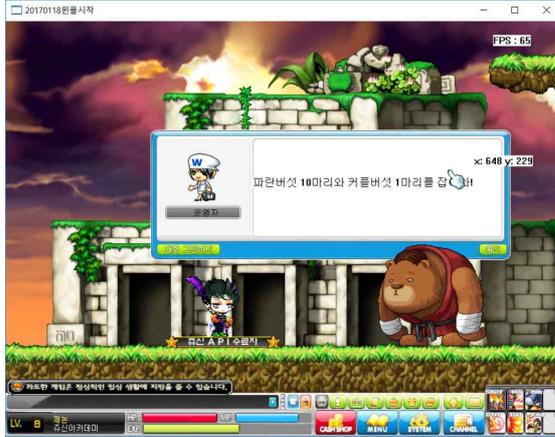
2D게임에 필요한 프레임 워크와 디자인패턴중 싱글톤,미디에이트,팩토리 패턴 3가지를 배웠다. 프로그래밍에 대한 두려움이 없어지고 보다 방대한 코드를 효율적으로 구현하는 법을 터득했다.

메이플스토리 스크린샷

7) 궁극기 스킬



8) Quest



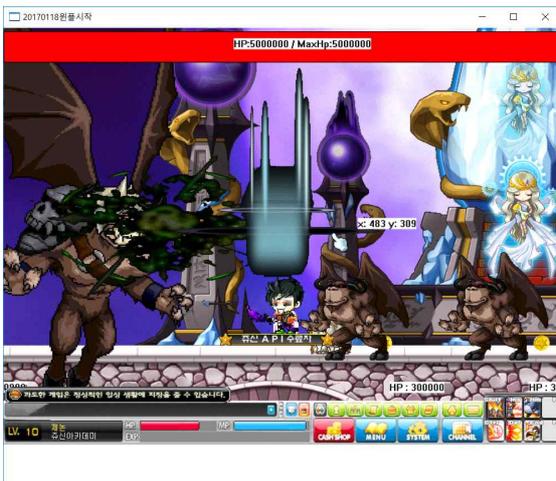
9) 대형 몬스터



10) Quest



11) 중간 보스



12) 최종 보스



API 소스 코드(C++언어)

1) Mediate Pattern 소스코드 일부

```
for(int i = 0; i < OBJ_end; ++i)
{
    list<CObj+>::iterator iter = m_ObjList[i].begin();
    OBJITER iter_end = m_ObjList[i].end();

    for(iter; iter != iter_end; )
    {
        int iResult = (*iter)->Update();
        if(iResult == 1)
        {
            delete *iter;
            iter = m_ObjList[i].erase(iter);
            iter_end = m_ObjList[i].end();
        }
        else if(iResult == 99)
            return 0;
        else
            ++iter;
    }
}
```

2) Mediate Pattern 소스코드 일부

```
for(int i = 0; i < RENDER_end; ++i)
{
    int iCount = m_vecRender[i].size();

    for(int j = 0; j < iCount; ++j)
    {
        m_vecRender[i][j]->Render(_dc);
        // i : Type
        // j : 객체 수
    }
    m_vecRender[i].clear();
}
```

3) Y Sorting 소스코드 일부

```
for(int i = 0; i < OBJ_end; ++i)
{
    list<CObj+>::iterator iter = m_ObjList[i].begin();
    OBJITER iter_end = m_ObjList[i].end();

    for(iter; iter != iter_end; ++iter)
    {
        m_vecRender[ (*iter)->GetRenderType() ].push_back(*iter);
    }
}
sort(m_vecRender[RENDER_WORLD OBJ].begin(),
     m_vecRender[RENDER_WORLD OBJ].end(),
     CObjMgr::Compare );
bool CObjMgr::Compare( CObj* pDest, CObj* pSour )
{
    return (pDest->GetInfo()->fY + pDest->GetInfo()->fCY
           <
           pSour->GetInfo()->fY + pSour->GetInfo()->fCY);
}
```

4) SingleTon Pattern 소스코드 일부

```
class CObj;
typedef list<CObj+>::iterator OBJITER;

#define DECLARE_SINGLETON(Classname) \
private: \
static Classname* m_pInstance; \
public: \
static Classname* GetInstance(void) \
{ \
    if(m_pInstance == NULL) \
    { \
        m_pInstance = new Classname; \
    } \
    return m_pInstance; \
} \
void DestroyInstance(void) \
{ \
    if(m_pInstance != NULL) \
    { \
        delete m_pInstance; \
        m_pInstance=NULL; \
    } \
} \
private: \
Classname(Classname& rObj){}; \
Classname& operator =(Classname& rObj){};

#define IMPLEMENT_SINGLETON(Classname) \
Classname* Classname::m_pInstance = NULL;
```

API 소스 코드(C++언어)

5) 알파 블렌딩 소스코드

```
if(m_tInfo.vPos.x < m_tInfo.fCX/2)
    m_tInfo.vPos.x = m_tInfo.fCX/2;
if(m_tInfo.vPos.x > m_tInfo.fCX/2)
    m_tInfo.vPos.x = m_tInfo.fCX/2;

if(m_tInfo.vPos.y > m_tInfo.fCY/2-65)
    m_tInfo.vPos.y = m_tInfo.fCY/2-65;

if(m_tInfo.vPos.y < m_tInfo.fCY/2+15)
    m_tInfo.vPos.y = m_tInfo.fCY/2+15;
// 오른쪽...
if(m_tInfo.vPos.x + m_ptScroll.x > WINCX / 2)
{
    float fSpeed = (m_tInfo.vPos.x + m_ptScroll.x) - WINCX / 2;
    if(m_ptScroll.x < WINCX - x)
    {
        m_tInfo.vPos.x -= WINCX - x;
        g_fScroll = WINCX - x;
        m_ptScroll.x = WINCX - x;
    }
}
//왼쪽..
if(m_tInfo.vPos.x + m_ptScroll.x < WINCX / 2 || m_tInfo.fX + m_ptScroll.x < WINCX / 2)
{
    float fSpeed = WINCX / 2 - (m_tInfo.vPos.x + m_ptScroll.x);
    if(m_ptScroll.x > 0)
    {
        g_fScroll = 0;
        m_ptScroll.x = 0;
        m_tInfo.vPos.x = 0;
    }
}
}
```

6) Line 충돌 소스코드 일부

```
LINE* pLine = NULL;

for(list<LINE>::iterator iter = m_pLineList->begin();
    iter != m_pLineList->end(); ++iter)
{
    if((*iter)->tLPoint.fX < m_tInfo.vPos.x && m_tInfo.vPos.x < (*iter)->trPoint.fX)
    {
        pLine = (*iter);
        break;
    }
}
if(pLine == NULL)
{
    m_tInfo.vPos.y += 10.f;
    return;
}
float fWidth = pLine->trPoint.fX - pLine->tLPoint.fX;
float fHeight = pLine->trPoint.fY - pLine->tLPoint.fY;
float fGradient = fHeight / fWidth; //기울기
float fY = fGradient * (m_tInfo.vPos.x - pLine->tLPoint.fX) + pLine->tLPoint.fY;
if(m_bJump != true && m_bUp == false)
{
    m_tInfo.vPos.y = fY - m_tInfo.fCY/2;
}
else
{
    if(m_tInfo.vPos.y > fY)
    {
        m_bJump = false;
        m_fJumpAcc = 0.f;
        m_tInfo.vPos.y = fY;
    }
}
}
```

7) 비트맵 이미지 출력 소스코드

```
pBitmap = CBitmapMgr::GetInstance()->FindImage(m_pName);
TransparentBit(_dc,
    int(m_tInfo.fX - m_tInfo.fCX) + (int)g_fScroll,
    int(m_tInfo.fY - m_tInfo.fCY) + (int)m_ptScroll.y,
    int(m_tInfo.fCX),
    int(m_tInfo.fCY),
    pBitmap->GetMemDC(),
    int(m_tInfo.fCX * m_tFrame.iFrameStart),
    int(m_tInfo.fCY * m_tFrame.iScene),
    int(m_tInfo.fCX),
    int(m_tInfo.fCY),
    RGB(0, 255, 0));
```

API 소스 코드(C++언어)

8) 인벤 객체 추가 구현 소스코드

```
if(PtInRect(&rect2, ptMouse))
{
    if(GetAsyncKeyState(VK_LBUTTON))
    {
        CSoundMgr::GetInstance()->PlayEffectSound(L"click.MP3");
        if(bTemp)
            return;
        bTemp = true;
        //포션이 인벤이 존재하는지 검사하자.
        ppInven = ((CPlayer*)m_pPlayer)->GetInven();
        ppRender = ((CPlayerUI14*)m_pPlayer)->GetInven();

        while(*ppInven != NULL)
        {
            if((*ppInven)->GetItem()->strname== m_pitem[0]->GetItem()->strname )
            {
                //아이템이 중복.
                ++(*ppInven)->GetItem()->iCount;
                (*iter_begin)->GetInfo()->iMoney -= m_pitem[0]->GetItem()->iPrice;
                return ;
            }
            ++ppInven;
        }
        while(*ppRender != NULL)
        {
            if((*ppRender)->GetItem()->strname== m_pitem[0]->GetItem()->strname )
            {
                //아이템이 중복.
                ++(*ppRender)->GetItem()->iCount;
                return ;
            }
            ++ppRender;
        }
        //포션이 없을 새물계 추가해준다.
        pitem = new Citem;
        pitem->SetItem( *m_pitem[0]->GetItem() );
        pitem->SetType( m_pitem[0]->GetType() );

        *ppInven = pitem;
        *ppRender = pitem;
    }
}
```

9) 숫자 이펙트 간격조절 소스코드

```
int D1000000= D1000000, D100000, D10000, D1000, D100, D10, D1;
D1000000= m_HitNum/1000000;
D100000= m_HitNum/100000%10;
D10000= m_HitNum /10000%10;
D1000 = m_HitNum/1000%10;
D100= m_HitNum/100%10;
D10= m_HitNum/10%10;
D1= m_HitNum %10;
if(D10 !=0 || D100 !=0 || D1000 !=0 || D10000 !=0 || D100000 !=0 || D1000000 !=0)
{
    pBitmap = CBitmapMgr::GetInstance()->FindImage(m_pName);
    TransparentBit(_dc,
        int(m_tInfo.fx - m_tInfo.fcX) + (int)g_fScroll + 50,
        int(m_tInfo.fy - m_tInfo.fcY) + (int)m_ptScroll.y,
        int(m_tInfo.fcX),
        int(m_tInfo.fcY),
        pBitmap->GetMemDC(),
        int(m_tInfo.fcX * D10),
        int(m_tInfo.fcY * m_tFrame.iScene),
        int(m_tInfo.fcX),
        int(m_tInfo.fcY),
        RGB(0, 0, 0));
}
```

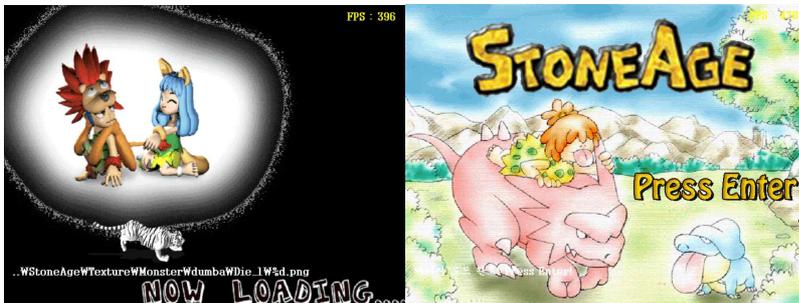
10) 미사일 Move 소스코드

```
if(m_bMissileRight==false)
{
    m_tInfo.fx += cosf(To_Radian(m_fAngle)) * m_fVeleocity;
    m_tInfo.fy += -sinf(m_fAngle * PI / 180) * m_fVeleocity;
}
else
{
    m_tInfo.fx -= cosf(To_Radian(m_fAngle)) * m_fVeleocity;
    m_tInfo.fy -= -sinf(m_fAngle * PI / 180) * m_fVeleocity;
}
```

I - ii. 학원과제 - 2

프로젝트	DirectX2D,MFC	지도선생님	이승준
기간	2017년 4월~2016년 6월 (대략 40일)	언어	c++
제목	스톤에이지		
주요내용	MFC Tool을 이용해 타일에 속성을 부여하여 save한후 Client에서 Load해 객체가 이동가능한 타일을 구분하였다.전체 11만개의 타일로 퀘스트,움직일 수 있는 타일, 아이템, 이동가능한 타일로 구분하였다. PathFinder라는 기능을 구현해 리소스의 경로를 보다 쉽게 관리하였고 쓰레드를 이용해 로딩창을 구현하였다. Astar알고리즘으로 플레이어를 움직였고 턴제 RPG답게 각자의 턴마다 공격,방어,스킬,도망 등을 구현하였다. 알파값에따라 Fade Out을 처리하여 화면전환을 보다 자연스럽게 하였다. Global변수 하나를 두어 총 17개의 퀘스트를 구현하였다. 이펙트는 브리지패턴을 이용하였고 UI는 옵저버패턴을 사용하였다.		

1) 쓰레드사용 리소스 로딩 2) 로고화면



3) 캐릭터 선택창 4) 캐릭터 생성창



5) InGame 6) 펫트창



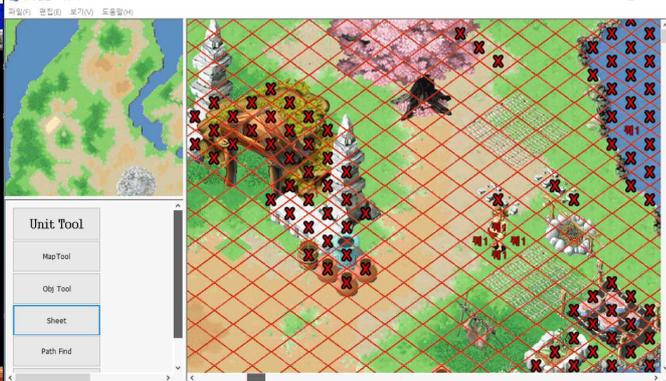
결과 Tool이란걸 처음 사용한 프로젝트이다. Client에서 구현하는것보다 쉽게말해 속성을 부여하여 해당 구조체들을 저장하고 Client에서 불러와 사용하는 방식인데 확실히 맵이넓거나 유닛이 많다면 효율적이라는걸 깨닫는 프로젝트였다. Astar라는 알고리즘도 배웠으며 내 프로젝트 중에서 가장 모작이 잘된 게임같아서 뿌듯하였다. Tool을통해 기획자나 그래픽디자이너와 소통이 편해질수있다는걸 느낄수 있었다..

스톤에이지 스크린샷

7) 미니맵



8) MFC를 이용한 각종 툴



9) 전투Scene



10) 화면전환시 이펙트



11) 피격 이펙트



12) 퀘스트 진행



스톤에이지 스크린샷

13) 펫트탑승



14) 탑승문서획득



15) 배 탑승



16) 펫트 획득



17) 버스 탑승



18) 퀘스트 진행



스톤에이지 스크린샷

19) 스킬 이펙트



20) 경험치 획득



21) 맵들을 이용한 못가는 타일



22) 전투Scene



23) 퀘스트 진행



24) 상점



스톤에이지 스크린샷

25) 마을 내부



26) 객체 알파테스팅



27) 플레이어 스킬



28) 스킬이펙트



29) 전투Scene



30) mfc를 이용한 못가는 타일들



스톤에이지 스크린샷

31) 중간 보스



32) 퀘스트 완료



33) 변산아이템 사용



34) 마을 외각



35) 최종 보스



36) Ending Scene



Direct2D, MFC 소스 코드(C++언어)

1) Factory 패턴 소스코드

```
template<typename T>
class CFactory abstract //추상 클래스 : abstract
{
public:
    static CObj* Create(void)
    {
        CObj* pObj = new T;
        pObj->Initialize();
        return pObj;
    }

    static CObj* Create(float _fx, float _fy)
    {
        CObj* pObj = new T;
        pObj->Initialize();
        pObj->SetPos(_fx, _fy);
        return pObj;
    }

    static CObj* Create(D3DXVECTOR3 vPos)
    {
        CObj* pObj = new T;
        pObj->Initialize();
        pObj->SetPos(vPos);
        return pObj;
    }
};
```

2) Observer 패턴 객체 추가 소스코드

```
void CDataSubject::AddData(int iMessage, void* pData)
{
    if(pData != NULL)
    {
        map<int, list<void*>>::iterator iter;
        iter = m_mapData.find(iMessage);
        if(iter == m_mapData.end())
            m_mapData.insert( make_pair(iMessage, new list<void*>() ) );
        m_mapData[iMessage]->push_back(pData);
        Notify(iMessage, pData);
    }
}
```

3) Observer 패턴 리스트에서 찾는 소스코드

```
void CUIObserver::Update( int iMessage, void* pData )
{
    list<void*>*** ppDataList = CDataSubject::GetInstance()->GetInstance()->GetDataList( iMessage);
    if(ppDataList ==NULL)
        return;

    list<void*>::iterator iter;
    iter = find((*ppDataList)->begin(), (*ppDataList)->end(), pData);

    switch( iMessage )
    {
    case MESSAGE_PLAYER_DATA:
        m_tData = *((UNIT_DATA*)(*iter));
        break;

    case MESSAGE_PLAYER_MATRIX:
        m_matPlayerMatrix = *((D3DXMATRIX*)(*iter));
        break;
    }
}
```

4) Observer 패턴 정보를 리스트에서 갱신하는 소스코드

```
void CSubject::Notify(int iMessage, void* pData /*= NULL*/)
{
    list<CObserver*>::iterator iter = m_ObserverList.begin();

    for(iter; iter != m_ObserverList.end(); ++iter)
    {
        (*iter)->Update( iMessage, pData);
    }
}
```

Direct2D, MFC 소스 코드(C++언어)

5) Bridge패턴 Render하는 소스코드

```
void CEffectBridge::Render(void)
{
    const TEX_INFO* pEffectTex =
        CTextureMgr::GetInstance()->GetTexture(
            m_pObject->GetObjKey().c_str(),
            m_wstrStateKey.c_str(),
            static_cast<int>(m_tFrame.fFrame) );

    if(pEffectTex == NULL)
        return;

    float fX = pEffectTex->ImageInfo.Width / 2.f;
    float fY = pEffectTex->ImageInfo.Height / 2.f;

    CDevice::GetInstance()->GetSprite()->SetTransform(
        &m_pObject->GetInfo()->matWorld
    );

    CDevice::GetInstance()->GetSprite()->Draw(
        pEffectTex->pTexture,
        NULL,
        &D3DXVECTOR3(fX, fY, 0.f),
        NULL,
        D3DCOLOR_ARGB(50, 255, 255, 255)
    );
}
```

6) Bridge패턴 알파블렌딩 소스코드

```
void CEffect_Buff::Render(void)
{
    CDevice::GetInstance()->Render_End();
    CDevice::GetInstance()->Render_Begin();

    CDevice::GetInstance()->GetDevice()->SetRenderState(D3DRS_BLENDDOP, D3DBLENDDOP_ADD);
    CDevice::GetInstance()->GetDevice()->SetRenderState(D3DRS_SRCBLEND, D3DBLEND_ONE);
    CDevice::GetInstance()->GetDevice()->SetRenderState(D3DRS_DESTBLEND, D3DBLEND_ONE);
    CEffectBridge::Render();
}
```

7) Bridge패턴 matrix정보를 부여하는 소스코드

```
void CEffect_Buff::WorldMatrix(INFO& rInfo)
{
    D3DXMATRIX matRotZ, matTrans, matScale, matParent;

    D3DXMatrixRotationZ(&matRotZ, D3DXToRadian(m_fRotateAngle) );
    D3DXMatrixTranslation(&matTrans, rInfo.vPos.x, rInfo.vPos.y, 0.f);
    D3DXMatrixScaling(&matScale, 1.f, 1.f, 1.f);

    matParent = dynamic_cast<CPlayer*>(CObjMgr::GetInstance()->FindGameObject<CPlayer>()->GetInfo()->matWorld;

    m_fRotateAngle += 100 * GET_TIME;

    rInfo.matWorld = matScale * matRotZ * matTrans * matParent;
}
```

8) MapTool에서 Picking하는 소스코드

```
//내적 피킹
bool CBackground::Picking(const D3DXVECTOR3& vPos
    , const int& iIndex)
{
    float fSlope = (TILECY / 2.f) / (TILECX / 2.f);
    //포인트를 구한다.
    D3DXVECTOR3 vPoint[4] =
    {
        D3DXVECTOR3( m_vecTile[iIndex]->vPos.x,
            m_vecTile[iIndex]->vPos.y - TILECY / 2.f,
            0.f),

        D3DXVECTOR3( m_vecTile[iIndex]->vPos.x + TILECX / 2.f,
            m_vecTile[iIndex]->vPos.y,
            0.f),

        D3DXVECTOR3( m_vecTile[iIndex]->vPos.x,
            m_vecTile[iIndex]->vPos.y + TILECY / 2.f,
            0.f),

        D3DXVECTOR3( m_vecTile[iIndex]->vPos.x - TILECX / 2.f,
            m_vecTile[iIndex]->vPos.y,
            0.f)
    };

    //방향 벡터
    D3DXVECTOR3 vDir[4] =
    {
        vPoint[1] - vPoint[0],
        vPoint[2] - vPoint[1],
        vPoint[3] - vPoint[2],
        vPoint[0] - vPoint[3]
    };

    //법선벡터를 만들어 보자.
    D3DXVECTOR3 vNormal[4] =
    {
        D3DXVECTOR3(vDir[0].y, -vDir[0].x, 0.f),
        D3DXVECTOR3(vDir[1].y, -vDir[1].x, 0.f),
        D3DXVECTOR3(vDir[2].y, -vDir[2].x, 0.f),
        D3DXVECTOR3(vDir[3].y, -vDir[3].x, 0.f),
    };
}
```

Direct2D, MFC 소스 코드(C++언어)

9) 내적을이용한 Picking 구현 소스코드

```
//법선벡터들을 단위벡터로 변경을 한다.
for(int i = 0; i < 4; ++i)
    D3DXVec3Normalize(&vNormal[i], &vNormal[i]);

for(int i = 0; i < 4; ++i)
{
    D3DXVECTOR3 vTemp = vPos - vPoint[i];

    float fDot = D3DXVec3Dot(&vTemp, &vNormal[i]);

    if(fDot > 0.f)
        return false;

    //내적 결과가 양수면 타일 외부이고
    //음수면 타일 내부이다.
}
return true;
```

11) MFC에서 저장하는 소스코드

```
void CMapTool::OnBnClickedButtonMapsave()
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    CMainFrame* pMainFrame = ((CMainFrame*)AfxGetMainWnd());
    CTool73View* pMainView = pMainFrame->GetMainView();
    CBackground* pBackground = pMainView->GetBackground();
    vector<TILE*> pVecTile = pBackground->GetTile();
    DWORD dwByte = 0;

    HANDLE hFile = CreateFile(L"..\\Data\\MapData.dat",
        GENERIC_WRITE,
        0, 0,
        CREATE_ALWAYS,
        FILE_ATTRIBUTE_NORMAL,
        NULL);

    for(size_t i = 0; i < pVecTile->size(); ++i)
    {
        WriteFile(hFile, (*pVecTile)[i], sizeof(TILE), &dwByte, NULL);
    }

    CloseHandle(hFile);
    //MSG_BOX(L"저장 성공!!");
}
```

10) MFC에서 로드하는 소스코드

```
void CMapTool::OnBnClickedButtonMapload()
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    CMainFrame* pMainFrame = ((CMainFrame*)AfxGetMainWnd());
    CTool73View* pMainView = pMainFrame->GetMainView();
    CBackground* pBackground = pMainView->GetBackground();
    vector<TILE*> pVecTile = pBackground->GetTile();
    DWORD dwByte = 0;

    HANDLE hFile = CreateFile(L"..\\Data\\MapData.dat",
        GENERIC_READ,
        0, 0,
        OPEN_EXISTING,
        FILE_ATTRIBUTE_NORMAL,
        NULL);
    for(size_t i = 0; i < pVecTile->size(); ++i)
    {
        safe_delete((*pVecTile)[i]);
    }
    pVecTile->clear();
    while(true)
    {
        TILE* pTile = new TILE;
        ReadFile(hFile, pTile, sizeof(TILE), &dwByte, NULL);
        if(dwByte == 0)
        {
            safe_delete(pTile);
            break;
        }
        pVecTile->push_back(pTile);
    }
    CloseHandle(hFile);
    pMainView->Invalidate();
}
```

12) Clnet에서 로드하는 소스코드

```
void CBackground::LoadTile(void)
{
    DWORD dwByte = 0;
    HANDLE hFile = CreateFile(L"..\\Data\\MapData.dat", GENERIC_READ,
        0, NULL, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL);

    for(size_t i = 0; i < m_vecTile.size(); ++i)
    {
        delete m_vecTile[i];
    }
    m_vecTile.clear();
    while(true)
    {
        TILE* pTile = new TILE;
        ReadFile(hFile, pTile, sizeof(TILE), &dwByte, NULL);
        if(dwByte == 0)
        {
            delete pTile;
            break;
        }
        m_vecTile.push_back(pTile);
    }
    CloseHandle(hFile);
}
```

Direct2D, MFC 소스 코드(C++언어)

13) Astar알고리즘중 노드만드는 소스코드

```
NODE* AStar::MakeNode(int iIndex, NODE* pParent, const vector<TILE*> pTile)
{
    NODE* pNode = new NODE;
    pNode->iIndex = iIndex;
    pNode->pParent = pParent;

    D3DXVECTOR3 vDistance = (*pTile)[iIndex]->vPos -
        (*pTile)[pParent->iIndex]->vPos;

    //노드와 노드 사이(인접한) 거리.....
    float fPCost = D3DXVec3Length(&vDistance);

    vDistance = (*pTile)[iIndex]->vPos - (*pTile)[m_iGoalIndex]->vPos;

    //가려는 노드에서 끝 인덱스까지의 거리...
    float fGCost = D3DXVec3Length(&vDistance);

    //최종 비용을 구함...
    pNode->fCost = fPCost + fGCost;

    return pNode;
}
```

14) BestList만드는 소스코드

```
//왼쪽 위 타일...
iIndex = pParent->iIndex -
    (TILEX - (pParent->iIndex / TILEX) % 2) - 1;

if(pParent->iIndex % TILEX != 0 &&
    pParent->iIndex >= TILEX + 2 &&
    (*pVecTile)[iIndex]->byOption == 0 &&
    CheckList(iIndex))
{
    pNode = MakeNode(iIndex, pParent, pVecTile);
    m_OpenList.push_back(pNode);
}

m_OpenList.sort(ComPare);

list<NODE*>::iterator iter = m_OpenList.begin();
pParent = *iter;

m_CloseList.push_back(*iter);
m_OpenList.erase(iter);

if(pParent->iIndex == m_iGoalIndex)
{
    while(true)
    {
        m_BestList.push_back(pParent->iIndex);

        //계속 역추적...
        pParent = pParent->pParent;

        if(pParent->iIndex == m_iStartIndex)
            break;
    }

    m_BestList.reverse();

    break;
}
```

15) 플레이어가 Astar로 움직이는 소스코드

```
void CPlayer::AStarMove(void)
{
    g_aster=true;
    if(m_dwState==STATE_WALK)
        MouseAngle();
    list<int*> pBestList = AStar::GetInstance()->GetBestList();
    if(pBestList->empty())
    {
        g_aster=false;
        return;
    }
    CBackground* pBack = (CBackground*)CObjMgr::GetInstance()->GetObjList(OBJ_BACKGROUND)->front();
    const vector<TILE*> pTile = pBack->GetTile();
    int iMoveIndex = pBestList->front();

    m_tInfo.vDir = (*pTile)[iMoveIndex]->vPos - m_tInfo.vPos/+m_vScroll+;
    float fDistance = D3DXVec3Length(&m_tInfo.vDir);
    D3DXVec3Normalize(&m_tInfo.vDir, &m_tInfo.vDir);
    m_tInfo.vPos += m_tInfo.vDir * m_fSpeed * GET_TIME *100 ;
    m_vScroll -= m_tInfo.vDir * m_fSpeed * GET_TIME *100 ;

    if(fDistance < 5.f)
    {
        pBestList->pop_front();
    }
}
```

Direct2D, MFC 소스 코드(C++언어)

16) matrix정보로 텍스처 렌더 소스코드

```
D3DXMatrixIdentity(&m_tInfo.matWorld );
D3DXMatrixScaling(&matScale, 1.0f* (m_DeadTime+2), 0.7f* (m_DeadTime+2), 1.0f);
D3DXMatrixRotationZ(&matRotZ, D3DXToRadian(0));
D3DXMatrixTranslation(&matTrans, m_tInfo.vPos.x, m_tInfo.vPos.y-150, 0);
m_tInfo.matWorld = matScale * matRotZ * matTrans;
CDevice::GetInstance()->GetSprite()->SetTransform(&m_tInfo.matWorld);
pTexInfo
= CTextureMgr::GetInstance()->GetTexture( m_wstrObjKey.c_str()
, m_wstrStateKey.c_str() , int(m_tFrame.fFrame));
if(pTexInfo == NULL)
return;
fX = float(pTexInfo->ImageInfo.Width / 2);
fY = float(pTexInfo->ImageInfo.Height / 2);
CDevice::GetInstance()->GetSprite()->Draw(
pTexInfo->pTexture,
NULL,
&D3DXVECTOR3(fX, fY, 0),
NULL,
D3DCOLOR_ARGB(255, 255, 255, 255));
```

17) 전투 Class Update()소스코드

```
if(turn ==1)
{
    if(randenemy2==0)
    {
        MonsterFightupdate1();
    }
    else if(randenemy2==1)
    {
        MonsterFightPet1();
        if(CMon1->GetInfo()->iHp<=0)
            randenemy2=0;
    }
}
if(m_bAttack[0]==false &&
m_bAttack[1]==false&&
m_bAttack[2]==false&&
m_bAttack[3]==false&&
m_bAttack[4]==false)
SelectMonster();
```

18) 전투 Class Update()소스코드

```
if(CMon1->GetInfo()->SkillNum==2)
{
    PetDefense();
}
if(CMon1->GetInfo()->SkillNum!=0)
{
    list<CObj*>::iterator iter_begin1 = CObjMgr::GetInstance()->GetObjList()[OBJ_UI_FIGHT].begin();
    list<CObj*>::iterator iter_begin1_end = CObjMgr::GetInstance()->GetObjList()[OBJ_UI_FIGHT].end();
    for (; iter_begin1 != iter_begin1_end; ++iter_begin1)
    {
        //rtti에서 삭제
        if(typeid(*iter_begin1)==typeid(CUI9))
            (*iter_begin1)->SetDead();
        if(typeid(*iter_begin1)==typeid(CUI5_5))
            (*iter_begin1)->SetDead();
    }
}
if(myturn==0)
PlayerFightUpdate();
if(myturn==1)
PetAttack();
if(myturn==2)
PlayerSkill();
if(turn ==0)
{
    if(randenemy1==0)
    {
        MonsterFightupdate();
    }
    else if(randenemy1==1)
    {
        MonsterFightPet();
        if(CMon1->GetInfo()->iHp<=0)
            randenemy1=0;
    }
}
```

19) 몬스터가 플레이어 따라다니는 소스코드

```
list<CObj*>::iterator iter_begin = CObjMgr::GetInstance()->GetObjList()[OBJ_PLAYER].begin();

if(g_fight==false)
{
    m_dwState = (*iter_begin)->GetState();
    m_dwState1 = (*iter_begin)->GetState();
}

m_tInfo.vDir = (*iter_begin)->GetInfo()->vPos - m_tInfo.vPos;
float fDistance = D3DXVec3Length(&m_tInfo.vDir);
D3DXVec3Normalize(&m_tInfo.vDir, &m_tInfo.vDir);
```

I - ii. 학원과제 - 3

프로젝트	DirectX9,3D프로그래밍,MFC	지도선생님	김남희
기간	2017년 7월~2017 8월 (1개월)	언어	C++
제목	블레이드 앤 소울		
주요내용	셰이더를 통해 알파값조절, 맵툴을이용해 맵배치, 네비게이션 툴을이용해 네비게이션 Cell 배치, 액션카메라, 컴포넌트패턴, 내부구현을 동적라이브러리화(DLL사용),메쉬,애니메이션 로드, 부위파괴, AABB,OBB충돌체크, 조명, 소드트레일, 보스패턴, 범프맵핑, 군집알고리즘, 직교투영 함수를 이용한 UI등등을 구현하였다.		

1) 로고창



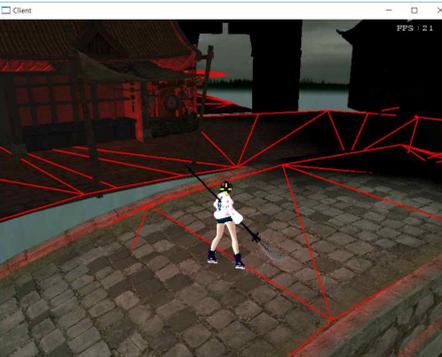
2) InGame



3) NPC



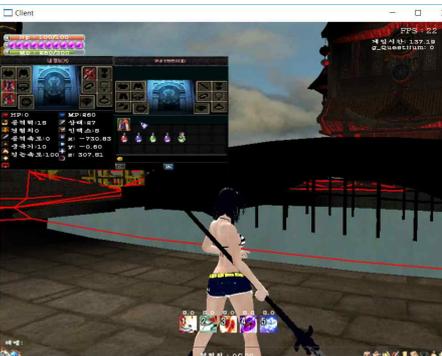
4) MFC를 이용한 네비메쉬



5) 조명



6) 인벤토리창

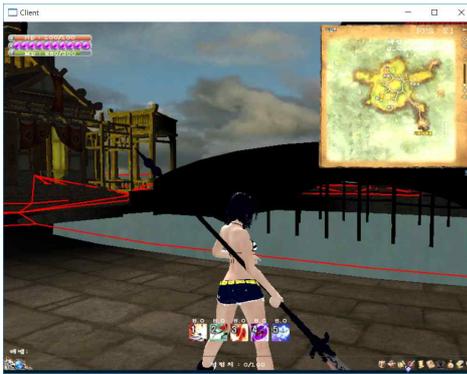


결과

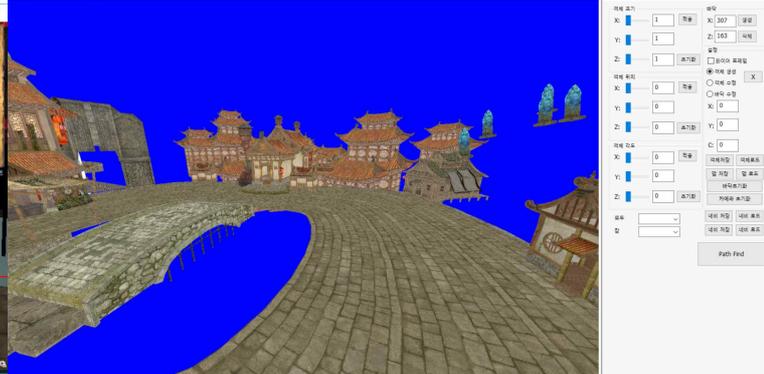
렌더링 파이프라인에 대해 이해하였고 애니메이션, 메쉬로딩, OBB충돌, 카메라구현, 파티클 등등을 구현하면서 게임회사에서 왜 게임을 팀단위로 만드는지 몸소느꼈다. 그중 컴포넌트패턴은 정말 혁신적이었다. Shader를 사용함으로써 조명연산, 범프맵핑 등 디바이스차원의 렌더링을 제어할 수 있게 되었다.

블레이드 엔 소울 스크린샷

7) 미니맵



8) MFC를 이용한 맵틀 및 네비게이션틀



9) 스킬창



10) 전투화면 및 이펙트(소드트레일)



11) 중간 보스



12) 플레이어 스킬 이펙트



블레이드 엔 소울 스크린샷

13) 중간 보스



14) 플레이어 스킬이펙트



15) 피격 이펙트



16) 액션카메라



17) 보스 스킬 파티클



18) 피격 이펙트

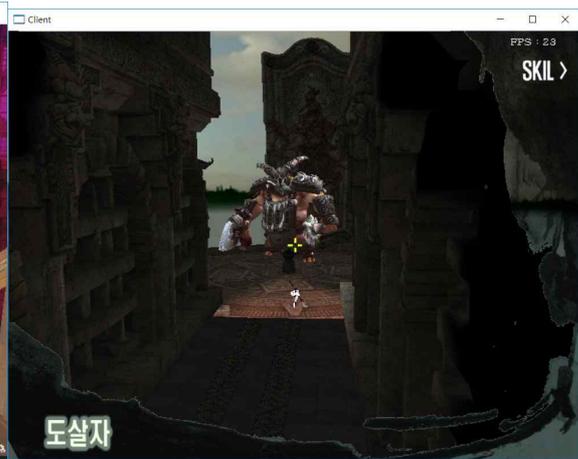


블레이드 엔 소울 스크린샷

19) 퀘스트 UI



20) 액션 카메라



21) 중간 보스



22) 보스 스킬 파티클



23) NPC

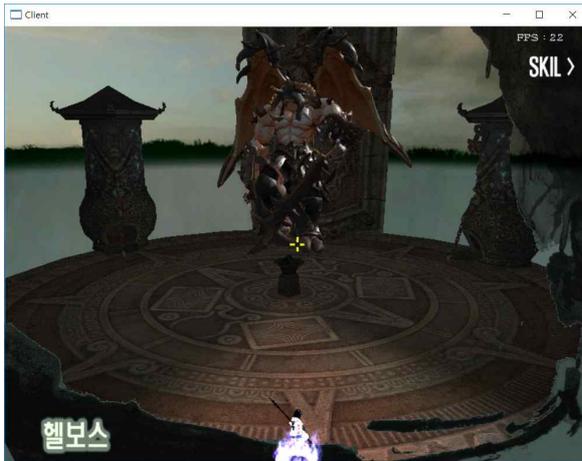


24) 전투Scene



블레이드 엔 소울 스크린샷

25) 중간 보스



26) 중간보스 스킬 파티클



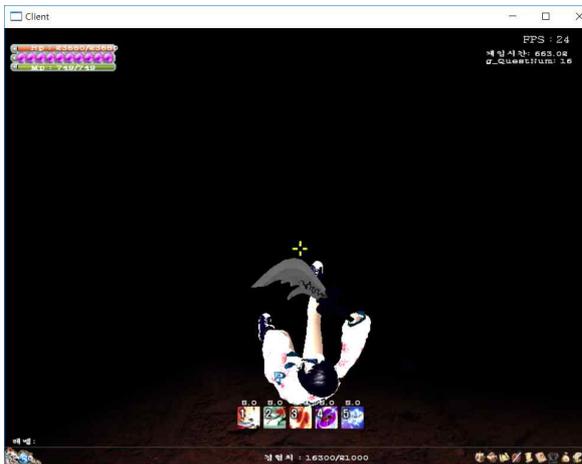
27) 파티사냥



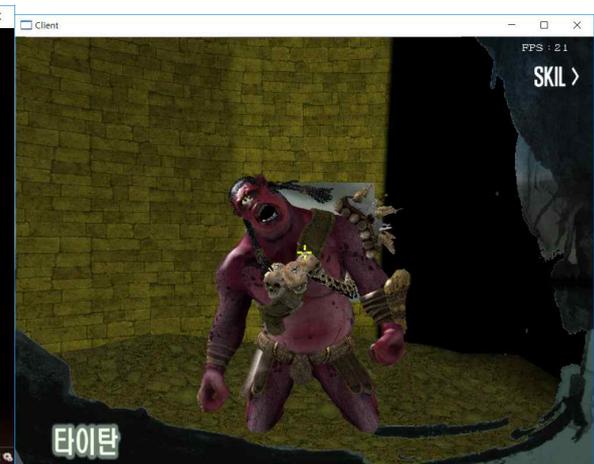
28) 조명처리



29) 낙하 애니메이션



30) 최종 보스 액션카메라



블레이드 엔 소울 스크린샷

31) 최종 보스



32) 변신 이펙트



33) 피격 이펙트



34) 플레이어 스킬 파티클



35) 보스 스킬 파티클



36) 보스 스킬 이펙트



블레이드 엔 소울 스크린샷

37) 보스 스킬 이펙트



38) 플레이어스킬 이펙트



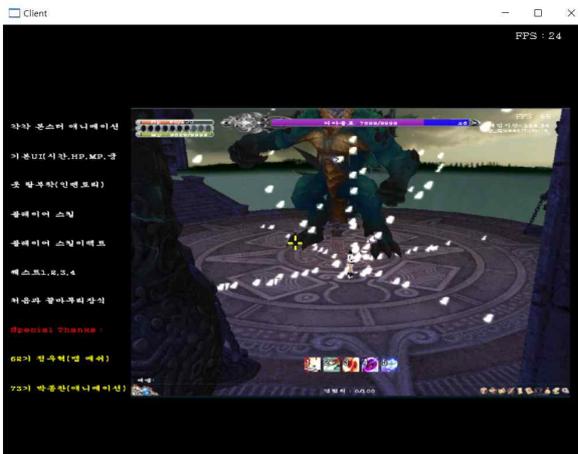
39) 보스공격 애니메이션



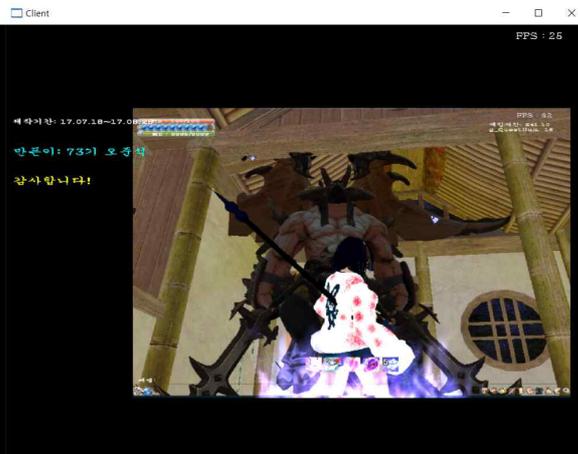
40) 보스 죽는 애니메이션



41) EndingScene



42) EndingScene



Direct3D 소스 코드(C++언어)

1) 컴포넌트 패턴을 위한 소스코드 2) Layer단위로 객체 관리 소스코드

```
#ifndef Component_h_
#define Component_h_

#include "Engine_Include.h"

BEGIN(Engine)

class ENGINE_DLL CComponent
{
protected:
    explicit CComponent(void);

public:
    virtual ~CComponent(void);

public:
    virtual void Update(void) {}
    virtual DWORD Release(void) PURE;
};

END

#endif // Component_h_

HRESULT Engine::CLayer::AddObject(const wstring& wstrObjKey, CGameObject* pGameObject)
{
    if(pGameObject)
    {
        MAPOBJLIST::iterator iter = m_mapObjList.Find(wstrObjKey);
        if(iter == m_mapObjList.end())
        {
            m_mapObjList[wstrObjKey] = OBJECLIST();
            m_mapObjList[wstrObjKey].push_back(pGameObject);
        }
        return S_OK;
    }
}

void Engine::CLayer::Update(void)
{
    MAPOBJLIST::iterator iter = m_mapObjList.begin();
    MAPOBJLIST::iterator iter_end = m_mapObjList.end();

    for (; iter != iter_end; ++iter)
    {
        OBJECLIST::iterator iterList = iter->second.begin();
        OBJECLIST::iterator iterList_end = iter->second.end();

        for (; iterList != iterList_end; )
        {
            if((*iterList)->Update()==1)
            {
                delete *iterList;
                iterList = iter->second.erase(iterList);
            }
            else
                ++iterList;
        }
    }
}
}
```

3) 맵틀에서 Mesh삭제 소스코드

```
void CBackGround::MeshDelete()
{
    // 메쉬 삭제
    D3DXMATRIX matWorld, matView, matProj;
    D3DXMatrixIdentity(&matWorld);
    m_pDevice->GetTransform(D3DTS_VIEW, &matView);
    m_pDevice->GetTransform(D3DTS_PROJECTION, &matProj);

    WORD dwCount = 0;
    WORD dwTriCheck = 0;
    float fDist = 0.f;
    int iDestIndex = -1;
    float fDestDist = 99999.f;
    vector<Engine::CStaticMesh*>::iterator iter = m_vecMesh.begin();
    vector<Engine::CStaticMesh*>::iterator iter_end = m_vecMesh.end();

    for(iter; iter != iter_end; ++iter)
    {
        m_pMouseCol->PickMesh(&dwTriCheck
            , &fDist
            , &(*iter)->GetInfo()->m_vPos
            , m_vp
            , g_hWnd
            , &((*iter)->GetInfo()->m_matWorld)
            , &matView
            , &matProj);
    }
    if(fDist!=0)
    {
        vector<Engine::CStaticMesh*>::iterator iter = m_vecMesh.end();
        iter--;

        for(int i = 0; i < iDestIndex; ++i)
        {
            ++iter;
        }
        Safe_Delete(*iter);
        m_vecMesh.erase(iter);
    }
}
```

Direct3D 소스 코드(C++언어)

4) 맵틀에서 Mesh추가 소스코드

```
if( NULL != m_pTerrain )
{
    m_pInfo->m_vPos.z+=500;
    m_pInfo->m_vPos.y-=50;
    vDestPos =m_pInfo->m_vPos;
    {
        Engine::CComponent*      pComponent;
        Engine::CStaticMesh*     pBuffer;
        switch(m_dwMeshNum)
        {
            case 0:
                pComponent= Engine::Get_ResourceMgr()->CloneResource(Engine::RESOURCE_STATIC, L"0");
                break;
            case 1:
                pComponent= Engine::Get_ResourceMgr()->CloneResource(Engine::RESOURCE_STATIC, L"1");
                break;
        }
        pBuffer = dynamic_cast<Engine::CStaticMesh*>(pComponent);

        NULL_CHECK(pBuffer);
        pBuffer->SetvPosInfo(vDestPos);
        pBuffer->SetvSizeInfo(m_vSize);
        pBuffer->SetAngleInfo(m_fAngle[Engine::ANGLE_X], m_fAngle[Engine::ANGLE_Y], m_fAngle[Engine::ANGLE_Z]);
        pBuffer->SetMeshNum(m_dwMeshNum);
        m_vecMesh.push_back(pBuffer);
    }
}
```

5) 네비메쉬 틀에서 네비메쉬 저장 소스코드

```
void CBackGround::SaveNavMesh()
{
    time_t timer = time(NULL);
    tm t;
    localtime_s(&t, &timer);

    // 현재 시간으로 저장
    TCHAR szFileName[64];
    sprintf(szFileName, L"../Map/RealNavi.dat", t.tm_mon + 1, t.tm_mday, t.tm_hour, t.tm_min, t.tm_sec);
    HANDLE hFile = CreateFile(szFileName, GENERIC_WRITE, NULL, NULL, CREATE_ALWAYS, FILE_ATTRIBUTE_NORMAL, NULL);
    DWORD dwByte;
    unsigned int i = 0;
    unsigned int j = 0;
    size_t iSize = m_vecSaveMesh.size();
    while (iSize > 0)
    {
        m_vecSaveMesh.pop_back();
        iSize--;
    }

    int iSaveSize = iSize / 3;

    WriteFile(hFile, &iSaveSize, sizeof(int), &dwByte, NULL); // 전체 갯수

    while ( (i+2) < iSize)
    {
        Engine::NAVESH NavMesh;

        if(m_vecSaveOption.size() > j)
            NavMesh.iOption = m_vecSaveOption[j++];
        NavMesh.vecNavMesh[0] = m_vecSaveMesh[i++];
        NavMesh.vecNavMesh[1] = m_vecSaveMesh[i++];
        NavMesh.vecNavMesh[2] = m_vecSaveMesh[i++];

        WriteFile(hFile, &NavMesh, sizeof(NavMesh), &dwByte, NULL);
    }
    CloseHandle(hFile);
}
```

6) 네비메쉬 틀에서 네비메쉬 로드 소스코드

```
void CBackGround::LoadNavMesh(void)
{
    DWORD dwByte = 0;
    Reset();
    // 기존 네비메쉬 그대로 두고 이어서 불러오기
    HANDLE hFile = CreateFile(L"../Map/RealNavi.dat", GENERIC_READ, NULL, NULL, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL);
    unsigned int iSize = 0;
    ReadFile(hFile, &iSize, sizeof(int), &dwByte, NULL);
    while (true)
    {
        Engine::NAVESH* pNavMesh = new Engine::NAVESH;
        ReadFile(hFile, pNavMesh, sizeof(Engine::NAVESH), &dwByte, NULL);

        if (dwByte == 0)
        {
            Engine::Safe_Delete(pNavMesh);
            break;
        }

        m_iOption = pNavMesh->iOption;
        CheckClockwise(pNavMesh->vecNavMesh);

        // 저장할 정보
        for(int i=0; i<3; ++i)
            m_vecSaveMesh.push_back(pNavMesh->vecNavMesh[i]);
        m_vecSaveOption.push_back(m_iOption);

        D3DXPLANE plane;
        // Del이 추가
        Engine::Get_NaviMgr()->AddDel((pNavMesh->vecNavMesh[0], pNavMesh->vecNavMesh[1], pNavMesh->vecNavMesh[2],
        Engine::Safe_Delete(pNavMesh);
    }
    CloseHandle(hFile);
}
```

Direct3D 소스 코드(C++언어)

7) 클라이언트에서 네비메쉬로드시 모두 시계방향으로 바꿔주는 소스코드

```
void CBackground::CheckClockWise(D3DXVECTOR3 vecNavMesh[3])
{
    D3DXVECTOR3 vec1 = vecNavMesh[1] - vecNavMesh[0];
    D3DXVECTOR3 vec2 = vecNavMesh[2] - vecNavMesh[1];

    D3DXVECTOR3 vecCross;
    D3DXVec3Cross(&vecCross, &vec1, &vec2);

    if (vecCross.y < 0)
    {
        D3DXVECTOR3 vecTemp = vecNavMesh[1];
        vecNavMesh[1] = vecNavMesh[2];
        vecNavMesh[2] = vecTemp;
    }
}
```

8) 플레이어가 네비메쉬 위를 움직이는 소스코드

```
DWORD Engine::CNavMgr::MoveOnNavMesh(D3DXVECTOR3* pPos, const D3DXVECTOR3* pDir, const DWORD& dwCurrentIdx, float& pTop)
{
    NEIGHBOR eNeighborID;
    DWORD dwNextIndex = dwCurrentIdx;

    if(m_vecNavMesh[dwCurrentIdx]->CheckPass(pPos, pDir, &eNeighborID))
    {
        CNavCell* pNeighbor = m_vecNavMesh[dwCurrentIdx]->GetNeighbor(eNeighborID);
        if(pNeighbor == NULL)
        {
            //출력이동백터
        }
        else
        {
            dwNextIndex = pNeighbor->GetIndex();

            if(pPos->y < pTop)
                pPos->y = (-pNeighbor->m_pPlane.a * pPos->x - pNeighbor->m_pPlane.c * pPos->z - pNeighbor->m_pPlane.d) / pNeighbor->m_pPlane.b;
            *pPos += *pDir;
        }
    }
    else
    {
        if(pPos->y < pTop)
            pPos->y = (-m_vecNavMesh[dwCurrentIdx]->m_pPlane.a * pPos->x - m_vecNavMesh[dwCurrentIdx]->m_pPlane.c * pPos->z - m_vecNavMesh[dwCurrentIdx]->m_pPlane.d) / m_vecNavMesh[dwCurrentIdx]->m_pPlane.b;
        *pPos += *pDir;
    }

    return dwNextIndex;
}
```

9) 네비메쉬 로드시 인접셀과 링크하는 소스코드

```
void Engine::CNavMgr::LinkCell(void)
{
    VECCELL::iterator iter = m_vecNavMesh.begin();
    if(m_vecNavMesh.end() == iter)
        return ;

    for( ; iter != m_vecNavMesh.end(); ++iter)
    {
        VECCELL::iterator iter_Target = m_vecNavMesh.begin();
        while(iter_Target != m_vecNavMesh.end())
        {
            if(iter == iter_Target)
            {
                ++iter_Target;
                continue;
            }

            if((+iter_Target)->ComparePoint((+iter)->GetPoint(POINT_A), (+iter)->GetPoint(POINT_B), (+iter)))
            {
                (+iter)->SetNeighbor(NEIGHBOR_AB, (+iter_Target));
            }

            else if((+iter_Target)->ComparePoint((+iter)->GetPoint(POINT_B), (+iter)->GetPoint(POINT_C), (+iter)))
            {
                (+iter)->SetNeighbor(NEIGHBOR_BC, (+iter_Target));
            }

            else if((+iter_Target)->ComparePoint((+iter)->GetPoint(POINT_C), (+iter)->GetPoint(POINT_A), (+iter)))
            {
                (+iter)->SetNeighbor(NEIGHBOR_CA, (+iter_Target));
            }

            ++iter_Target;
        }
    }
}
```

Direct3D 소스 코드(C++언어)

10) 정점셰이더에서 조명연산하는 소스코드

```
VS_OUT VS_MAIN(VS_IN In)
{
    VS_OUT    Out = (VS_OUT)0;

    Matrix    matWV, matWVP;

    matWV = mul(g_matWorld, g_matView);
    matWVP = mul(matWV, g_matProj);

    Out.vPos = mul(vector(In.vPos.xyz, 1.f), matWVP);
    Out.vTexUV = In.vTexUV;

    vector    vLightDirInv = normalize(g_vLightDir) * -1;
    vector    vWorldNormal = normalize(mul(vector(In.vNormal.xyz, 0.f), g_matWorld));

    Out.vShade = max(dot(vLightDirInv, vWorldNormal), 0.f) * (g_vLightDiffuse + g_vMtrlDiffuse) + (g_vLightAmbient + g_vMtrlAmbient);

    vector    vReflect = reflect(normalize(g_vLightDir), vWorldNormal);
    vector    vWorldPos = mul(vector(In.vPos.xyz, 1.f), g_matWorld);

    vector    vLook = vWorldPos - g_vCamPos;

    Out.vSpecular = pow(max(dot(normalize(vLook) * -1.f), normalize(vReflect)), 0.f), g_fPower) * (g_vLightSpecular + g_vMtrlSpecular);
    return Out;
}
```

11) 카메라 거리에따라 맵 객체들을 움직이는 소스코드

```
//플레이어 스킬이벤트 액션카메라!
const Engine::CComponent* pPlayerInfo = Engine::Get_Management()->GetComponent(
    CStage::LAYER_GAMELOGIC, L"Player", L"Transform");
const Engine::CComponent* pPlayerMesh = Engine::Get_Management()->GetComponent(
    CStage::LAYER_GAMELOGIC, L"Player", L"DynamicMesh");

//const D3DXMATRIX* m_pmatPlayerSkill;
m_pmatPlayerSkill = ((Engine::CDynamicMesh*)pPlayerMesh)->FindFrame("ValveBiped_Anim_Attachment_CAM_Origin");

if(fDistance <=700 && fDistance >= 50.f
    && m_pInfo->m_tPlayer.m_dwState!=STATE_START
    && m_pInfo->m_tPlayer.m_dwState!=STATE_DASH)
{
    if(m_pInfo->m_tPlayer.iRealLife<4)
    {
        m_pInfo->m_tPlayer.m_dwState=STATE_RUN;
        m_pInfo->m_vPos.x -= vDir.x * 2.f* m_fSpeed + Engine::Get_TimeMgr()->GetTime();
        m_pInfo->m_vPos.z -= vDir.z * 2.f* m_fSpeed + Engine::Get_TimeMgr()->GetTime();
    }
    else
    {
        m_pInfo->m_tPlayer.m_dwState=STATE_WALK;
        m_pInfo->m_vPos.x -= vDir.x * m_fSpeed + Engine::Get_TimeMgr()->GetTime();
        m_pInfo->m_vPos.z -= vDir.z * m_fSpeed + Engine::Get_TimeMgr()->GetTime();
    }
}
else if(fDistance < 50.f)
{
    m_pInfo->m_tPlayer.iFightSpeed+=Engine::Get_TimeMgr()->GetTime();
    if(m_pInfo->m_tPlayer.iFightSpeed>3.5f)
    {
        m_pInfo->m_tPlayer.m_dwState=STATE_STAND;
        m_pInfo->m_tPlayer.iFightSpeed=0.f;
        m_iSkill=rand() %3;
        switch(m_iSkill)
        {
            case 0: CSoundMgr::Get Instacne()->Player2SkillSound(L"sfx_bg_melee_s2_warn.ogg"); break;
            case 1: CSoundMgr::Get Instacne()->Player2SkillSound(L"sfx_bg_dam_hv03.ogg"); break;
            case 2: CSoundMgr::Get Instacne()->Player2SkillSound(L"sfx_bg_dam_hv04.ogg"); break;
        }
    }
    else
    {
        switch(m_iSkill)
        {
            case 0:
                m_pInfo->m_tPlayer.m_dwState=STATE_ATT;
                break;
        }
    }
}
```

12) 클라이언트에서 셰이더정보 Set하는 소스코드

```
LPD3DXEFFECT pEffect = m_pShader->GetEffect();
NULL_CHECK(pEffect);
D3DXMATRIX matView, matProj;
m_pDevice->GetTransform(D3DTS_VIEW, &matView);
m_pDevice->GetTransform(D3DTS_PROJECTION, &matProj);
pEffect->SetMatrix("g_matWorld", &m_pInfo->m_matWorld);
pEffect->SetMatrix("g_matView", &matView);
pEffect->SetMatrix("g_matProj", &matProj);
pEffect->SetTexture("g_BaseTexture", m_pTexture->GetTexture());
pEffect->SetFloat("g_fDetail", 1.f);
pEffect->SetFloat("g_fAlpha", m_fAlpha);
m_pShader->Begin_Shader();
m_pBuffer->Render(&m_pInfo->m_matWorld);
m_pShader->End_Shader();
```

Direct3D 소스 코드(C++언어)

13) Clinet에서 컴포넌트 붙여주는 소스코드

```
HRESULT CPlayer::AddComponent(void)
{
    Engine::CComponent* pComponent = NULL;

    //Transform
    pComponent = m_pInfo = Engine::CTransform::Create(g_vLook);
    NULL_CHECK_RETURN(pComponent, E_FAIL);
    m_mapComponent.insert(MAPCOMPONENT::value_type(L"Transform", pComponent));

    //Mesh
    pComponent = m_pResourceMgr->CloneResource(RESOURCE_STAGE, L"Player");
    m_pDynamicMesh = dynamic_cast<Engine::CDynamicMesh*>(pComponent);
    NULL_CHECK_RETURN(m_pDynamicMesh, E_FAIL);
    m_mapComponent.insert(MAPCOMPONENT::value_type(L"DynamicMesh", pComponent));

    //CollisionOBB
    pComponent = Engine::Get_CollisionMgr()->CloneColObject(Engine::CCollisionMgr::COLLISION_OBB);
    m_pCollisionOBB = dynamic_cast<Engine::CCollision_OBB*>(pComponent);
    NULL_CHECK_RETURN(pComponent, E_FAIL);
    m_mapComponent.insert(MAPCOMPONENT::value_type(L"Collision_OBB", pComponent));
    m_pCollisionOBB->SetColInfo(&m_pInfo->m_matWorld, m_pDynamicMesh);

    //CollisionAABB
    pComponent = Engine::Get_CollisionMgr()->CloneColObject(Engine::CCollisionMgr::COLLISION_AABB);
    m_pCollisionAABB = dynamic_cast<Engine::CCollision_AABB*>(pComponent);
    NULL_CHECK_RETURN(pComponent, E_FAIL);
    m_mapComponent.insert(MAPCOMPONENT::value_type(L"Collision_AABB", pComponent));
    m_pCollisionAABB->SetColInfo(&m_pInfo->m_matWorld, m_pDynamicMesh);

    //Shader
    pComponent = Engine::Get_ShaderMgr()->CloneShader(L"Shader_Mesh");
    m_pShader = dynamic_cast<Engine::CShader*>(pComponent);
    NULL_CHECK_RETURN(pComponent, E_FAIL);
    m_mapComponent.insert(MAPCOMPONENT::value_type(L"Shader", pComponent));
    return S_OK;
}
```

14) UI생성시 matrix정보를 직교투영화하는 소스코드

```
D3DXMATRIX matView,matProj,matOldView,matOldProj;
D3DXMatrixIdentity(&matView);
D3DXMatrixIdentity(&matProj);
m_pDevice->GetTransform(D3DTS_VIEW,&matOldView);
m_pDevice->GetTransform(D3DTS_PROJECTION,&matOldProj);
D3DXMatrixOrthoLH(&matProj,800.f,600.f,0.f,1.f);
m_pDevice->SetTransform(D3DTS_VIEW,&matView);
m_pDevice->SetTransform(D3DTS_PROJECTION,&matProj);
m_pInfo->m_matWorld._11+=400;
m_pInfo->m_matWorld._22+=300;
D3DXMATRIX matWorld,matTrans,matScale;
D3DXMatrixIdentity(&matWorld);
D3DXMatrixIdentity(&matTrans);
D3DXMatrixIdentity(&matScale);
```

15) 소드트레일 정점보간 소스코드

```
void CTrailEffect::PointMemorize(void)
{
    while(m_Pointlist.size() >= 20)
    {
        m_Pointlist.pop_front();
    }

    float fTime = Engine::Get_TimeMgr()->GetTime();
    m_fAccTime += fTime;

    if(0.01f < m_fAccTime)
    {
        D3DXVECTOR3 vPoint[2];
        vPoint[0] = D3DXVECTOR3(50.f, 0.f, 0.f);
        vPoint[1] = D3DXVECTOR3(20.f, 0.f, 0.f);
        for(int i = 0; i < 2; ++i)
        {
            D3DXVec3TransformCoord(&vPoint[i], &vPoint[i], m_pmatWeaponRef);
            D3DXVec3TransformCoord(&vPoint[i], &vPoint[i], m_pmatPlayerInfo);
            m_Pointlist.push_back(vPoint[i]);
        }

        m_fAccTime = 0.f;
    }
}
```

Direct3D 소스 코드(C++언어)

16) 플레이어 오른손 matrix정보에 무기 장착 소스코드

```
const Engine::CComponent* pPlayerInfo = Engine::Get_Management()->GetComponent(CStage::LAYER_GAMELOGIC, L"Player", L"Transform");
Engine::CComponent* pRealPlayer = NULL;
pRealPlayer =const_cast<Engine::CComponent*>(pPlayerInfo);

D3DXVec3TransformNormal(&m_pInfo->m_vDir, &g_vLook, &m_pInfo->m_matWorld);
if(m_pmatPlayerInfo == NULL || m_pmatWeaponRef == NULL)
{
    const Engine::CComponent* pPlayerInfo = Engine::Get_Management()->GetComponent(
        CStage::LAYER_GAMELOGIC, L"Player", L"Transform");
    const Engine::CComponent* pSphereInfo = Engine::Get_Management()->GetComponent(
        CStage::LAYER_GAMELOGIC, L"Sphere", L"Transform");
    const Engine::CComponent* pPlayerMesh = Engine::Get_Management()->GetComponent(
        CStage::LAYER_GAMELOGIC, L"Player", L"DynamicMesh");

    m_pmatPlayerInfo = &((static_cast<const Engine::CTransform*>(pPlayerInfo)->m_matWorld));
    m_pmatWeaponRef = ((Engine::CDynamicMesh*)pPlayerMesh)->FindFrame("ValveBiped_Anim_Attachment_RH");
    m_pSphereInfo = &((static_cast<const Engine::CTransform*>(pSphereInfo)->m_matWorld));
}
```

17) 플레이어 애니메이션 렌더링하는 소스코드

```
D3DXMatrixScaling(&matScale, m_pScale, m_pScale, m_pScale);
D3DXMatrixTranslation(&matTrans, x, y, z);
D3DXMatrixRotationX(&matRotX, m_pInfo->m_fAngle[Engine::ANGLE_X]);
D3DXMatrixRotationY(&matRotY, m_pInfo->m_fAngle[Engine::ANGLE_Y]);
D3DXMatrixRotationZ(&matRotZ, m_pInfo->m_fAngle[Engine::ANGLE_Z]);
m_pInfo->m_matWorld= matScale* matRotX * matRotY * matRotZ* matTrans;
m_pDynamicMesh->FrameMove(Engine::Get_TimeMgr()->GetTime());
```

18) 피격시 이펙트 생성하는 소스코드

```
pGameObject = CEffect::Create(m_pDevice, pRealPlayer, pRealMonster);
m_pLayer->AddObject(L"EffectNum", pGameObject);
m_mapLayer.insert(MAPLAYER::value_type(LAYER_EFFECT, m_pLayer));
dynamic_cast<Engine::CTransform*>(pRealPlayer)->m_tPlayer.m_dwStateCheck=0;

int randeffect =rand()&3;
switch(randeffect)
{
case 0:
    pGameObject = CEffectParticle::Create(m_pDevice, pRealPlayer, pRealMonster, 0);
    m_pLayer->AddObject(L"EffectNum", pGameObject);
    m_mapLayer.insert(MAPLAYER::value_type(LAYER_EFFECT, m_pLayer)); break;
case 1:
    pGameObject = CDiaEffect::Create(m_pDevice, pRealPlayer, pRealMonster, 0);
    m_pLayer->AddObject(L"EffectNum", pGameObject);
    m_mapLayer.insert(MAPLAYER::value_type(LAYER_EFFECT, m_pLayer)); break;
case 2:
    pGameObject = CDiaEffect1::Create(m_pDevice, pRealPlayer, pRealMonster, 0);
    m_pLayer->AddObject(L"EffectNum", pGameObject);
    m_mapLayer.insert(MAPLAYER::value_type(LAYER_EFFECT, m_pLayer)); break;
}
```

I-iii. 팀단위 프로젝트 - 1

프로젝트	DirectX9,TCP/IP,MFC,VS15,C++(11)	지도선생님	김남희
기간	2017년 6월~2017 7월 (약1개월)	언어	C++
제목	오버워치(서버연동)		
주요내용	TCP/IP 소켓프로그래밍을 이용한 4인용 서버게임, 맵틀에서 CUBE맵으로만 만든 맵과 충돌처리, 큐브캐릭터를 이용한 자연스러운 애니메이션, 화물을 움직이는 Quest와 각종 이벤트, AABB,OOB 추방향 정렬,회전 충돌체크, 각종 캐릭터마다 특징이있는 미사일, 몬스터,보스 패턴 알고리즘, 비트연산을 이용한 keyManager 등등		

1) 캐릭터 선택창



2) InGame



3)퀘스트UI



4) 파라의 궁극기



5)퀘스트 아이템



6) 옥상에서본 Range 타워



결과

3D게임에 처음으로 서버를 넣은 팀작업으로 TCP/IP에 대해 이해하였고 맵틀,애니메이션들을 3D로 작업하였다. 메쉬를 띄우기전단계로 3D에 대한 기본적인 렌더링파이프라인에 대해 복습하는 계기가되었고 카메라충돌, 키매니저 등 게임을 보다 원활히 할 수 있는 구조도 만들었다. 서버에 대한 막연한 두려움이 사라지는 중요한 팀작업이었다. 팀장으로써 팀원들의 의견을 수렴할줄알고 의사소통하는 법도 배운 프로젝트다.

오버워치 스크린샷

7)파라의 궁극기



8) 화물



9)Range타워



10)화물운송 퍼센트UI



11)보스STAGE



12) 리퍼의 평타공격



오버워치 스크린샷

13) 시작지점 아이템



14) 계단오르기



15) 리퍼와의 전투



16) 리퍼의 궁극기



17) 승리UI



18) 만드미



오버워치 스크린샷

19) 서버 연동시 캐릭터선택



20) 서버 연동시 InGame



21) 서버 연동시 같이 화물운송



Direct3D, TCP/IP 소스 코드(C++언어)

1) 서버 연동을 위한 구조체

```
typedef struct tagNetworkMember
{
    int    iNumber;
    BYTE   bNumber;
    D3DXVECTOR3 vPos;
    D3DXVECTOR3 vDir;
    int    iHp;
    int    iAttack;
    float   fAngleY;
    float   fAngleZ;
}ALL_MSG;
typedef struct tagNetworkMessage
{
    int    iClientNumber;
    ALL_MSG iMessage;
}NET_MSG;
typedef struct tagPlayerMessage
{
    ALL_MSG iMessage[4];
}P_MSG;
typedef struct tagiSelect
{
    int    iSelect;
    bool   bReady;
    bool   ChangeScene;
}P_SELECT;
```

2)소켓초기화

```
InitializeCriticalSection(&PlayerCS);
//----- 소켓 초기화
WSADATA data;
WSAStartup(MAKEWORD(2, 2), &data);
//----- Server 생성
SOCKET server = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
if( server == INVALID_SOCKET )
    ShowError("서버 생성 실패!");
//----- Server Bind
sockaddr_in addr = { 0 };
addr.sin_family = AF_INET;
addr.sin_addr.s_addr = htonl(INADDR_ANY);
addr.sin_port = htons(SERVERPORT);
if( bind(server, (sockaddr*)&addr, sizeof(addr)) == SOCKET_ERROR )
    ShowError("바인딩 실패!");
//----- Server listen
if( listen(server, SOMAXCONN) == SOCKET_ERROR )
    ShowError("듣기 실패!");
cout << "Server : waiting connection request." << endl;
```

3) 키매니저를위한 define

```
#define SERVERPORT 7000
#define CLIENT_EXIT 99
#define MOVE_FIRE 0x00000001
#define MOVE_UP 0x00000002
#define MOVE_DOWN 0x00000004
#define MOVE_LEFT 0x00000008
#define MOVE_RIGHT 0x00000010
#define MOVE_JUMP 0x00000020
#define MOVE_RIGHTA 0x00000040
#define MOVE_LEFTA 0x00000080
#define MOVE_ALL 0x000000FF
```

Direct3D, TCP/IP 소스 코드(C++언어)

4) 클라이언트 접속시 스레드 만드는 부분

```
sockaddr_in  clientAddr;
SOCKET      client;
int         addrlen = sizeof(clientAddr);
HANDLE      hThread;
while(1)
{
    //----- Client Access
    client = accept(server, (struct sockaddr*)&clientAddr, &addrlen);
    if( client < 0 )
        ShowError("Server : accept failed");
    cout << "Server : " << inet_ntoa(clientAddr.sin_addr) << " client connect." << endl;
    iClientCount++;

    recv(client, (char*)&iTemp, sizeof(int), 0);
    send(client, (char*)&iClientCount, sizeof(int), 0);
    m_ESCENE=SCENE_SELECT;
    hThread = CreateThread(NULL, 0, ProcessClient, (LPVOID)client, 0, NULL);
}
```

5) 스레드안에서 실제 값 주고받는 부분

```
while(iClientCount > 0)
{
    msg.iClientNumber = 0;
    msg.iMessage.iNumber = 0;
    msg.iMessage.vDir = D3DXVECTOR3(0, 0, 0);
    msg.iMessage.vPos = D3DXVECTOR3(0, 0, 0);
    msg.iMessage.bNumber=0;
    if(m_ESCENE==SCENE_STAGE)
    {
        retval = recv(client, (char*)&msg, sizeof(NET_MSG), 0);
        if( retval == SOCKET_ERROR )
        {
            cout << "수신 에러" << endl;
            break;
        }
        pMsg.iMessage[iClientNum - 1] = msg.iMessage;
        if( msg.iMessage.iNumber == CLIENT_EXIT ) // 클라이언트 접속 종료
            iClientCount--;
        retval = send(client, (char*)&pMsg, sizeof(P_MSG), 0);
        if( retval == SOCKET_ERROR )
        {
            cout << "송신 에러" << endl;
            break;
        }
    }
}
```

6) 몇 번째 접속 클라이언트에따라 다르게 객체 할당하는부분

```
pGameObject = CPlayer::Create(m_pDevice);
dynamic_cast<CPlayer*>(pGameObject) ->SetMap(dynamic_cast<CTerrain*>(pGameObjectmap) ->GetMap());
NULL_CHECK_RETURN(pGameObject, E_FAIL);
((CPlayer*)pGameObject) ->SetPlayerNumber(1);
((CPlayer*)pGameObject) ->SetSetLayer(pLayer);
pLayer ->AddObject(L"Player", pGameObject);
((CPlayer*)pGameObject) ->SetTerrainVtx(pTerrainVtx);
if( g_iClientNumber == (1) )
{
    m_pTarget = pGameObject ->GetComponent(L"Transform");
    m_pPUI = CPlayerUI::Create(CPlayerUI::PLAYERUI_STAGE1, pGameObject, pCargo, m_pDevice);
    if( g_iSelect.iMessage[0].iSelect==2) //파라
    {
        pGameObject = CMainUIImg::Create(m_pDevice, pGameObject, pCargo);
        NULL_CHECK_RETURN(pGameObject, E_FAIL);
        pLayer ->AddObject(L"MainUIImg", pGameObject);
    }
    if( g_iSelect.iMessage[0].iSelect==6) //라인
    {
        pGameObject = CMainUIImg1::Create(m_pDevice, pGameObject, pCargo);
        NULL_CHECK_RETURN(pGameObject, E_FAIL);
        pLayer ->AddObject(L"MainUIImg", pGameObject);
    }
    if( g_iSelect.iMessage[0].iSelect==8) //아나
    {
        pGameObject = CMainUIImg2::Create(m_pDevice, pGameObject, pCargo);
        NULL_CHECK_RETURN(pGameObject, E_FAIL);
        pLayer ->AddObject(L"MainUIImg", pGameObject);
    }
}
```

Direct3D, TCP/IP 소스 코드(C++언어)

7) 플레이 객체가 서버로부터 정보를 받고 갱신하는부분

```
int CPlayer::Update(void)
{
    g_iMsg.iMessage.bNumber =0;
    m_iPlayerNumber =1;
    if(m_tPlayer.iHp > m_tPlayer.iMaxHp)
        m_tPlayer.iHp = m_tPlayer.iMaxHp;
    if( g_iClientNumber == m_iPlayerNumber)
    {
        if(m_bMouse==true)
            MouseCheck();
        if( g_iSelect.iMessage[0].iSelect==2)
            Para();
        if( g_iSelect.iMessage[0].iSelect==6)
            Line();
        if(g_iSelect.iMessage[0].iSelect==8)
            Ana();
        KeyCheck();
    }
    else if(g_iClientNumber != m_iPlayerNumber)
    {
        MessageReceive();
    }
    else
        g_iMsg.iMessage.bNumber |= MOVE_DEAD;
}
```

8) 키값에따라 서버에 값을 전송하는 부분 9) 맵과 충돌체크하는 부분

```
if (GetAsyncKeyState('W'))
{
    g_iMsg.iMessage.bNumber |= MOVE_MOVE; // W
    m_pInfo->m_vPos += m_pInfo->m_vDir * m_fSpeed * fTime;
}
if (GetAsyncKeyState('S'))
{
    g_iMsg.iMessage.bNumber |= MOVE_MOVE; // S
    m_pInfo->m_vPos -= m_pInfo->m_vDir * m_fSpeed * fTime;
}
if( GetAsyncKeyState('R') )
{
    m_tPlayer.iBulletNum=m_tPlayer.iBulletMaxNum;
}
if( GetAsyncKeyState(VK_SPACE) )
{
    g_iMsg.iMessage.bNumber |= MOVE_JUMP;
    m_bJump=true;
}
g_iMsg.iMessage.fAngleY = m_pInfo->m_fAngle[Engine::ANGLE_Y];
g_iMsg.iMessage.fAngleZ = m_pInfo->m_fAngle[Engine::ANGLE_Z];
g_iMsg.iMessage.vPos = m_pInfo->m_vPos;
g_iMsg.iMessage.iHp = m_tPlayer.iHp;
g_iMsg.iMessage.iAttack = m_tPlayer.iAttack;
g_iMsg.iClientNumber = g_iClientNumber;
send(g_ClientSocket, (char*)&g_iMsg, sizeof(NET_MSG), 0);
recv(g_ClientSocket, (char*)&g_iGetMsg, sizeof(P_MSG), 0);
```

```
vector<Engine::CVIBuffer*>::iterator    iter = m_VecMap->begin();
vector<Engine::CVIBuffer*>::iterator    iter_end = m_VecMap->end();
for(; iter != iter_end; ++iter)
{
    float fx = (*iter)->GetInfo()->m_vSize.x;
    float fy = (*iter)->GetInfo()->m_vSize.y;
    float fz = (*iter)->GetInfo()->m_vSize.z;

    if(m_pInfo->m_vPos.x-1 < (*iter)->GetInfo()->m_vPos.x +fx&&
       m_pInfo->m_vPos.x+1 > (*iter)->GetInfo()->m_vPos.x &&
       m_pInfo->m_vPos.y-1 < (*iter)->GetInfo()->m_vPos.y +fy&&
       m_pInfo->m_vPos.y+1 > (*iter)->GetInfo()->m_vPos.y &&
       m_pInfo->m_vPos.z-1 < (*iter)->GetInfo()->m_vPos.z +fz&&
       m_pInfo->m_vPos.z+1 > (*iter)->GetInfo()->m_vPos.z )
    {
        if( m_fTop+1 < (*iter)->GetInfo()->m_vPos.y+(*iter)->GetInfo()->m_vSize.y
           )
        {
            if( GetAsyncKeyState('W') )
            {
                m_pInfo->m_vPos += m_pInfo->m_vDir* m_fSpeed * fTime;
            }
            if( GetAsyncKeyState('S') )
            {
                m_pInfo->m_vPos -= m_pInfo->m_vDir* m_fSpeed * fTime;
            }
        }
    }
}
```

Direct3D, TCP/IP 소스 코드(C++언어)

10) 맵과 충돌체크시 점프 부분

```
else if( m_bJump==false)
{
    if((*iter)->GetInfo()->m_vSize.x > (*iter)->GetInfo()->m_vSize.z)
        m_pInfo->m_vPos.z += float((*iter)->GetInfo()->m_vSize.z *0.1);
    else
        m_pInfo->m_vPos.x += float((*iter)->GetInfo()->m_vSize.x *0.1);
}
else if( m_pInfo->m_vPos.y-1 <(*iter)->GetInfo()->m_vPos.y +(*iter)->GetInfo()->m_vSize.y)
{
    if(m_pInfo->m_vPos.y-m_fTop < 4.f)
    {
        m_pInfo->m_vPos.y=(*iter)->GetInfo()->m_vPos.y + (*iter)->GetInfo()->m_vSize.y+1;
        m_fTop=(*iter)->GetInfo()->m_vPos.y + (*iter)->GetInfo()->m_vSize.y+1;
    }
}
```

11) aabb충돌체크시 플레이어 정보 등록부분

```
void CMonster::Collision_AABB( void )
{
    D3DXVECTOR3    vDestMin, vDestMax;
    const Engine::CComponent* pPlayerInfo = Engine::Get_Management()->GetComponent(CStage::LAYER_GAMELOGIC, L"Player", L"Transform")

    D3DXMATRIX     matAABB;
    D3DXMatrixIdentity(&matAABB);
    memcpy(&matAABB.m[3][0], &((Engine::CTransform*)pPlayerInfo)->m_matWorld.m[3][0], sizeof(D3DXVECTOR3));
    D3DXVec3TransformCoord(&vDestMin, &matAABB);
    D3DXVec3TransformCoord(&vDestMax, &matAABB);

    D3DXVECTOR3    v SourMin, v SourMax;
    if(Engine::CCollisionMgr::GetInstance()->Collision_AABB(&vDestMin, &vDestMax, &v SourMin, &v SourMax))
        m_bCollision = true;
    else
        m_bCollision = false;
}
```

12) 거리에 따른 몬스터 패턴

```
for(int i = 0 ;i<2;i++)
{
    vDir2 = vPlayer[i]->m_vPos - m_pInfo->m_vPos;
    if( fDistance >D3DXVec3Length(&vDir2))
    {
        vDir = vDir2;
        fDistance = D3DXVec3Length(&vDir2);
    }
}
D3DXVec3Normalize(&vDir, &vDir);
m_pInfo->m_fAngle[Engine::ANGLE_Y] = atan2(vDir.x,vDir.z);
iDelay++;
if(fDistance < 20.f)
{
    if(iDelay > 100)
    {
        CGameObject* pGameObj = new CMonsterBullet(m_pDevice,0);
        dynamic_cast<CMonsterBullet*>(pGameObj)->Initialize();
        dynamic_cast<CMonsterBullet*>(pGameObj)->SetPlayerPos(&m_pInfo->m_vPos, &vDir);
        m_vecBullet.push_back(dynamic_cast<CGameObject*>(pGameObj));
        iDelay=0;
    }
}
else if(fDistance <50)
{
    m_pInfo->m_vPos.x += vDir.x * m_fSpeed * Engine::Get_TimeMgr()->GetTime();
    m_pInfo->m_vPos.z += vDir.z * m_fSpeed * Engine::Get_TimeMgr()->GetTime();
}
```

13) 파라 공극기 발사 소스

```
if(paraPill==true&& paraPilltime<0.5f)
{
    paraPilltime=Engine::Get_TimeMgr()->GetTime();

    g_iMsg.iMessage.bNumber |= MOVE_SKILL3;
    g_iMsg.iMessage.vDir = m_pMouseCol->ShootBullet(&matView,&matProj);
    CGameObject* pGameObj = new CParaPill(m_pDevice);
    dynamic_cast<CParaPill*>(pGameObj)->Initialize();
    dynamic_cast<CParaPill*>(pGameObj)->SetPlayerPos(&m_pInfo->m_vPos, &vDir);
    //방향을 랜덤하자
    g_iMsg.iMessage.vDir.x = g_iMsg.iMessage.vDir.x + (-4*rand() %8)/10.f;
    g_iMsg.iMessage.vDir.y = g_iMsg.iMessage.vDir.y + (-4*rand() %8)/10.f;
    dynamic_cast<CParaPill*>(pGameObj)->SetPlayerPos(&g_iMsg.iMessage.vPos, &g_iMsg.iMessage.vDir);
    m_vecBullet.push_back(dynamic_cast<CGameObject*>(pGameObj));
}
else
{
    paraPill=false;
    paraPilltime=0.f;
}
```

Direct3D, TCP/IP 소스 코드(C++언어)

14) AABB충돌체크 확인코드

```
bool Engine::CCollisionMgr::Collision_AABB(
{
    float    fMin = 0.f;
    float    fMax = 0.f;

    fMin = max(pDestMin->x, pSourMin->x);
    fMax = min(pDestMax->x, pSourMax->x);
    if(fMin > fMax)
        return false;

    fMin = max(pDestMin->y, pSourMin->y);
    fMax = min(pDestMax->y, pSourMax->y);
    if(fMin > fMax)
        return false;

    fMin = max(pDestMin->z, pSourMin->z);
    fMax = min(pDestMax->z, pSourMax->z);
    if(fMin > fMax)
        return false;

    return true;
}
```

15) 화물이 저장된 벡터값에따라 움직이는 소스

```
if(m_bSetTruckStartPos == false && m_pvecNavigation->size() > 0)
{
    m_pInfo->m_vPos = m_pvecNavigation->front();
    m_bSetTruckStartPos = true;
}
D3DXVec3TransformNormal(&m_pInfo->m_vDir, &g_vLock, &m_pInfo->m_matWorld);
```

16) 수신버퍼 크기늘려주는 코드

```
//수신버퍼 크기를 두배로 늘린다.
getsockopt(client, SOL_SOCKET, SO_RCVBUF,
(char *)&optval, &optlen);
optval *= 2;
setsockopt(client, SOL_SOCKET, SO_RCVBUF,
(char *)&optval, sizeof(optval));
printf("수신 버퍼 크기(old)= %d바이트\n", optval);
```

17) 부모행렬을 곱함으로 미리 저장된 애니메이션 출력 부분

```
m_vFixFrame = m_vecMotionFix[i][m_iFrameCount+1];

m_pChest->GetBodyInfo()->m_vPos.x = m_vFixFrame.tChest.vPos.x;
m_pChest->GetBodyInfo()->m_vPos.y = m_vFixFrame.tChest.vPos.y;
m_pChest->GetBodyInfo()->m_vPos.z = m_vFixFrame.tChest.vPos.z;

m_pChest->GetBodyInfo()->m_vSize.x = m_vFixFrame.tChest.vSize.x * m_vSize.x;
m_pChest->GetBodyInfo()->m_vSize.y = m_vFixFrame.tChest.vSize.y * m_vSize.y;
m_pChest->GetBodyInfo()->m_vSize.z = m_vFixFrame.tChest.vSize.z * m_vSize.z;

m_pChest->GetBodyInfo()->m_fAngle[Engine::ANGLE_X] = m_vFixFrame.tChest.fAngle[Engine::ANGLE_X];
m_pChest->GetBodyInfo()->m_fAngle[Engine::ANGLE_Y] = m_vFixFrame.tChest.fAngle[Engine::ANGLE_Y];
m_pChest->GetBodyInfo()->m_fAngle[Engine::ANGLE_Z] = m_vFixFrame.tChest.fAngle[Engine::ANGLE_Z];

for(int j = 0; j < 11; ++j)
{
    m_pBodyParts[j]->GetBodyInfo()->m_vPos.x = m_vFixFrame.tBodyParts[j].vPos.x;
    m_pBodyParts[j]->GetBodyInfo()->m_vPos.y = m_vFixFrame.tBodyParts[j].vPos.y;
    m_pBodyParts[j]->GetBodyInfo()->m_vPos.z = m_vFixFrame.tBodyParts[j].vPos.z;

    m_pBodyParts[j]->GetBodyInfo()->m_vSize.x = m_vFixFrame.tBodyParts[j].vSize.x * m_vSize.x;
    m_pBodyParts[j]->GetBodyInfo()->m_vSize.y = m_vFixFrame.tBodyParts[j].vSize.y * m_vSize.y;
    m_pBodyParts[j]->GetBodyInfo()->m_vSize.z = m_vFixFrame.tBodyParts[j].vSize.z * m_vSize.z;

    m_pBodyParts[j]->GetBodyInfo()->m_fAngle[Engine::ANGLE_X] = m_vFixFrame.tBodyParts[j].fAngle[Engine::ANGLE_X];
    m_pBodyParts[j]->GetBodyInfo()->m_fAngle[Engine::ANGLE_Y] = m_vFixFrame.tBodyParts[j].fAngle[Engine::ANGLE_Y];
    m_pBodyParts[j]->GetBodyInfo()->m_fAngle[Engine::ANGLE_Z] = m_vFixFrame.tBodyParts[j].fAngle[Engine::ANGLE_Z];
}
```

I-iii. 팀단위 프로젝트 - 2

프로젝트	DirectX11, HLSL, MFC, C++(11), VS15	지도선생님	정의훈
기간	2017년 9월~2017 12월 (2개월)	언어	C++
제목	베트맨 아캄시티		
주요내용	애니메이션보간, 옵저버 패턴을 이용한 OBB충돌, 디퍼드셰이더를 이용한 조명연산 HLSL셰이더를이용한 노멀맵핑, 맵전체에 테셀레이션 적용 백터를 이용해 맵틀에서 제작한 맵객체 관리, 층단위 네비게이션메쉬, 직교투영을 이용한 UI, emissive를 이용한 자체발광간판 제작, 램버트 조명공식을 이용한 점,디렉션,스포트라이트 조명 맵의 깊이값을 이용해 플레이어 회전시 모션블러적용, StreamOut을 이용한 비 파티클, 파일입출력을 통한 액션카메라, 플레이어기준 90도 시야각안에 맞출수있는 지역을 향해 맞줄을쏘고 이동구현, DirectShow를이용한 동영상Loading, 거리에따른 사운드 크기 조절.		

1) 로고창



2) InGame



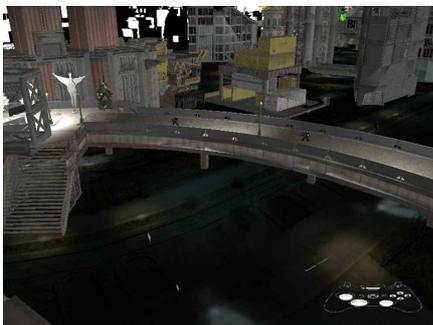
3)상공에서 바라본 맵



4)해상에서바라본 파도



5) 디퍼드이용한 조명



6) 동영상 및 UI



결과

DX11에 대해 이해하였다. 9과의 큰차이점은 크게 테셀레이션, 컴퓨팅셰이딩 등이있었다. 테셀레이션을 이용해 보다 세밀한 메쉬표현이 가능해졌고 컴퓨팅셰이딩을 통해 FPS상승이 올라갔다. 개인적으로 힘들었던점은 XMMATH 함수가 바로대입이 안되고 SetLoad함으로써 개발하는부분이 불편했는데 이또한 fps상승에 기여한다.

베트맨 아캄시티 스크린샷

7) 하강애니메이션



8) 플레이어 공격 애니메이션



9) 신호등 오브젝트



10) 탈 수 있는 자동차



11) 운전중



12) 운전중



베트맨 아캄시티 스크린샷

13) 디퍼드 조명,노말텍스처를 입힌 맵



14) 2Stage Boss



15) 디퍼드 조명,노말텍스처를 입힌 맵



DirectX11 소스 코드(C++언어)

1)플레이어기준 90도 시야각안에 맞출탈수있는 지역을 향해 빛줄을쏘고 이동구현

```
vector<XMVECTOR>::iterator iter = m_vGrappingPos.begin();
vector<XMVECTOR>::iterator iter_end = m_vGrappingPos.end();
for (iter; iter != iter_end; ++iter)
{
    float fDist = 0.f;
    XMVECTOR vecTempPos;
    XMVECTOR vTemp1 = XMLoadFloat3(m_pTransformCom->Get_Direction(Engine::CTransform::DIR_LOOK));
    XMVECTOR vTemp2 = XMLoadFloat3(vecPos) - XMLoadFloat3(&(*iter));
    vTemp1 = XMVector3Normalize(vTemp1);
    vTemp2 = XMVector3Normalize(vTemp2);
    float fDot = XMVectorGetX(XMVector3Dot(vTemp1, vTemp2));
    if (fDot > 0.5f)
    {
        vecTempPos = XMVector3Length(XMLoadFloat3(vecPos) - XMLoadFloat3(&(*iter)));
        fDist = XMVectorGetX(vecTempPos); //x값이 제일 짧은 진선 거리다.
        if (fDist > 3.f)
            m_mapPos.insert(make_pair(fDist, *iter)); //맵에 저장(자가 균형 트리)
    }
}
m_fShortestDist = m_mapPos.begin()->first;
m_vecDestinationPos = m_mapPos.begin()->second; //이걸 내시야각 에서만 검사
```

2)거리에따른 사운드 크기 조절

```
void CSoundMgr::MonsterSound(TCHAR * pSoundKey, float fDist)
{
    map<TCHAR*, FMOD_SOUND*>::iterator iter;

    iter = find_if(m_mapSound.begin(), m_mapSound.end(), Engine::CStringCMP(pSoundKey));

    if (iter == m_mapSound.end())
        return;

    FMOD_Sound_SetMode(iter->second, FMOD_LOOP_OFF);

    m_Result = FMOD_System_PlaySound(m_pSystem, FMOD_CHANNEL_REUSE, iter->second, 0, &m_pMonster);
    FMOD_Channel_SetVolume(m_pMonsterMent, 1 / fDist);
    ErrorCheck(m_Result);
}
```

3) 미리찍어둔 액션캠이 정보불러와서 움직이는코드

```
m_vEye=m_vecEye[i INDEX];
m_vDir = XMLoadFloat3(&_vec3(m_vecDir[i INDEX].x, m_vecDir[i INDEX].y, m_vecDir[i INDEX].z));
XMStoreFloat3(&m_vAt, XMLoadFloat3(&m_vEye) + m_vDir);
if (i INDEX < m_vecEye.size() && i INDEX < m_vecDir.size())
{
    i INDEX++;
}
```

4)맵의 Z값을 이용해 플레이어 회전시 모션블러적용

```
Blur = pin.vDir.xy;
for (int i = 1; i < 10; i++)
{
    BColor = g_DiffuseMap.Sample(samAnisotropic, pin.Tex + Blur * (float)i);
    BColor=g_DiffuseMap.Sample(samAnisotropic, pin.Tex);
    Out.ColorSpecInt += BColor;
}
Out.ColorSpecInt = curFramePixelVelocity;
Out.Veclotiy = pin.vDir;
Out.Normal = float4((pin.vNor), 1);
Out.Depth = vector(pin.vPosNew.z / pin.vPosNew.w, pin.vPosNew.w / 1000.0f, 0.f, 0.f);
return Out;
```

DirectX11 소스 코드(C++언어)

5) 램버트 조명공식을 이용한 스포트라이트 조명

```
void ComputeSpotLight1(Material mat, SpotLight L,
float3 normal, float3 toEye,
out float4 ambient,
out float4 diffuse,
out float4 spec)
{
    // Initialize Outputs
    ambient = float4(0.f, 0.f, 0.f, 0.f);
    diffuse = float4(0.f, 0.f, 0.f, 0.f);
    spec = float4(0.f, 0.f, 0.f, 0.f);
    // 빛 벡터는 광선이 나아가는 방향의 반대방향이다.
    float3 lightVec = -L.Direction;
    // Add ambient term : 주변광 항을 더한다.
    ambient = mat.Ambient * L.Ambient;
    // : 빛이 막히지 않고 표면에 도달한다는 가정 하에
    // 분산광 항과 반영광 항을 더한다.
    float diffuseFactor = dot(lightVec, normal);
    // 동적 분기를 피하기 위해 조건문을 펼친다.
    [flatten]
    if (diffuseFactor > 0.f)
    {
        float3 v = reflect(-lightVec, normal);
        float specFactor = pow(max(dot(v, toEye), 0.f), mat.Specular.w);
        diffuse = diffuseFactor * mat.Diffuse * L.Diffuse;
        spec = specFactor * mat.Specular * L.Specular;
    }
}
```

6) 직교투영을 이용한 UI

```
XMMATRIX mWorld, mView, mProj;
XMMATRIX I = XMMatrixIdentity();
mView = I;
mProj = I;
mProj = XMMatrixOrthographicLH(static_cast<float>(g_iClientCX), static_cast<float>(g_iClientCY), 0.f, 1.f);
XMMATRIX matScale, matRotZ, matTrans;
matScale = XMMatrixScaling(80.f, 170.f, 1.f);
matRotZ = XMMatrixRotationZ(XMConvertToRadians(-90));
matTrans = XMMatrixTranslation(0, 260.f, 0.f);
m_pTransformCom->m_matWorld = matScale * matRotZ * matTrans;
m_pMyGD1->pEffect->SetWorld(m_pTransformCom->m_matWorld);
m_pMyGD1->pEffect->SetView(mView);
m_pMyGD1->pEffect->SetProj(mProj);
m_pTextureCom->Render(m_iTex);
m_pBufferCom->Render(false);
```

7) 네비게이션메쉬 이웃일 때 Y값까지 구해서 움직이는코드

```
dwNextIndex = pNeighbor->GetIndex();
XMVECTOR VP = XMVectorFloat3(pPos);
XMVECTOR VD = XMVectorFloat3(pDir);
VP += VD;
XMStoreFloat3(pPos, VP);
pPos->y = (-pNeighbor->m_pPlane.x * pPos->x - pNeighbor->m_pPlane.z * pPos->z - pNeighbor->m_pPlane.w) / pNeighbor->m_pPlane.y;
```

8) 캐릭터 착지시 층단위 네비게이션메쉬 Y값 구하는코드

```
//지면에 도달했을 경우(네비메쉬 위로 고정)
if (pPlayerPos->y <= m_fNaviTopPosY && m_bFallState)
{
    m_bNaviMeshOnState = true;
    m_bLand = true;
    m_bJumpState = false; //점프 상태 초기화
    m_bFallState = false; //떨어지고 있는 상태 초기화
    m_fJumpAcc = 0.f; //가속도 초기화
    m_pTransformCom->m_vInformation[Engine::CTransform::INFO_POSITION].x = pPlayerPos->x;
    m_pTransformCom->m_vInformation[Engine::CTransform::INFO_POSITION].y = m_fNaviTopPosY;
    m_pTransformCom->m_vInformation[Engine::CTransform::INFO_POSITION].z = pPlayerPos->z;
}
```

DirectX11 소스 코드(C++언어)

9) 벡터를 이용해 맵틀에서 제작한 맵객체 렌더

```
float fDistance = 0.f;
XMVECTOR vCamPos, vMeshPos;
vCamPos = XMLoadFloat3(&g_vCamPos);
vector<Engine::CStaticMesh*>::iterator iter = m_VecMeshMap.begin();
vector<Engine::CStaticMesh*>::iterator iter_end = m_VecMeshMap.end();
XMVECTOR Pos;
XMVECTOR vCenter = XMLoadFloat3(&Pos);
for (iter; iter != iter_end; ++iter)
{
    vMeshPos = XMLoadFloat3((*iter)->GetInfo()->GetInformation(Engine::CTransform::INFO_POSITION));
    fDistance = XMVectorGetX(XMVector3Length((vCamPos) - (vMeshPos)));
    if (fDistance < 100.f || (*iter)->GetMeshNum()==131)
    {
        m_pMyGD1->pEffect->SetWorld((*iter)->GetInfo()->m_matWorld);
        m_pMyGD1->pEffect->SetWorldInverseTranspose(MathHelper::InverseTranspose((*iter)->GetInfo()->m_matWorld));
        (*iter)->RenderMesh(false);
    }
}
m_matLast = XMLoadFloat4x4(m_pMyGD1->pMatView) * XMLoadFloat4x4(m_pMyGD1->pMatProj);
```

10) 맵객체 셰이더에서 그릴때 테셀레이션 적용

```
// The domain shader is called for every vertex created by the tessellator.
// It is like the vertex shader after tessellation.
[domain("tri")]
DomainOut DS(PatchTess patchTess,
    float3 bary : SV_DomainLocation,
    const OutputPatch<HullOut, 3> tri)
{
    DomainOut dout;
    dout.vPos = bary.x * tri[0].vPos + bary.y*tri[1].vPos + bary.z * tri[2].vPos;
    dout.vTexUV = bary.x * tri[0].vTexUV + bary.y*tri[1].vTexUV + bary.z * tri[2].vTexUV;
    float4x4 matVP = mul(g_matView, g_matProj);
    float4x4 matWV, matWVP;
    matWV = mul(g_matWorld, g_matView);
    matWVP = mul(matWV, g_matProj);
    dout.vPosNew = mul(float4(dout.vPos, 1.0f), matWVP);
    return dout;
}
```

11) HLSL셰이더를이용한 노멀맵핑

```
for (int i = 0; i < gLightCount; ++i)
{
    float4 A, D, S;
    ComputeDirectionalLight(g_Material, g_DirLights[i], pin.vNor, toEyeW,
        A, D, S);
    ambient += A;
    diffuse += D;
    spec += S;
}
litColor = texColor * (ambient + diffuse) + spec;
litColor.a = g_Material.Diffuse.a * texColor.a;
return litColor;
```

DirectX11 소스 코드(C++언어)

12) 디퍼드쉐이더 실제값 계산하는 PS

```
PS_GBUFFER_OUT PackGBuffer(float3 BaseColor, float3 Normal, float SpecIntensity, float SpecPower)
{
    PS_GBUFFER_OUT Out;

    // Normalize the specular power
    float SpecPowerNorm = max(0.0001, (SpecPower - g_SpecPowerRange.x) / g_SpecPowerRange.y);

    // Pack all the data into the GBuffer structure
    Out.ColorSpecInt = float4(BaseColor.rgb, SpecIntensity);
    Out.Normal = float4(Normal * 0.5 + 0.5, 0.0);
    Out.SpecPow = float4(SpecPowerNorm, 0.0, 0.0, 0.0);

    return Out;
}

PS_GBUFFER_OUT RenderScenePS( VS_OUTPUT In )
{
    // Lookup mesh texture and modulate it with diffuse
    float3 DiffuseColor = DiffuseTexture.Sample( LinearSampler, In.UV );
    DiffuseColor *= DiffuseColor;

    return PackGBuffer(DiffuseColor, normalize(In.Normal), specIntensity, specExp);
}
```

13) 디퍼드쉐이더 렌더순서

```
void Engine::CRenderer::Render_GameObject(void)
{
    Engine::Get_GraphicDev()->Render_SetFirst();
    Engine::Get_GraphicDev()->Render_Begin();

    KEYON(bDebugOn, DIK_F1);

    ID3D11RenderTargetView* rt[5] = { Engine::Get_GraphicDev()->GetRenderTarget(), m_NormalRTV, m_SpecPowerRTV, m_VelocityRTV, m_DepthRTV };
    m_pMyGDI->pContext->OMSetRenderTargets(5, rt, Engine::Get_GraphicDev()->GetDepthSV());
    Render_Priority();
    Render_Deffered();
    Engine::Get_GraphicDev()->Render_USEDSV(m_DepthStencilDSV);
    Render_Alpha();
    Render_UI();

    if (bDebugOn)
        Render_Debug();

    m_LastSRV = m_ColorSpecIntensitySRV;
    Engine::Get_GraphicDev()->Render_End();
    Clear_RenderList();
}
```

14) 렌더타겟 생성하는 코드

```
void Engine::CRenderer::Render_Deffered(void)
{
    m_pMyGDI->pContext->OMGetDepthStencilState(&pPrevDepthState, &nPrevStencil);
    m_pMyGDI->pContext->PSSetSamplers(0, 1, &g_pSamLinear);

    PreRender(m_pMyGDI->pContext);

    Render_NoneAlpha();

    PostRender(m_pMyGDI->pContext);

    PrepareForUnpack(m_pMyGDI->pContext);
    ID3D11ShaderResourceView* arrViews[5] = { m_DepthStencilSRV, m_ColorSpecIntensitySRV, m_NormalSRV, m_SpecPowerSRV, m_VelocitySRV };

    m_pMyGDI->pContext->PSSetShaderResources(0, 5, arrViews);
    if (nullptr != m_pMyGDI->pMatView)
        m_Light->SetViewProj(XMLoadFloat4x4(m_pMyGDI->pMatView), XMLoadFloat4x4(m_pMyGDI->pMatProj));
    m_Light->DoLighting();

    m_pMyGDI->pContext->OMSetDepthStencilState(pPrevDepthState, nPrevStencil);

    ZeroMemory(arrViews, sizeof(arrViews));
    m_pMyGDI->pContext->PSSetShaderResources(0, 5, arrViews);
}
```

DirectX11 소스 코드(C++언어)

15) 충돌객체 옵저버 패턴에 등록하는 코드

```
int CObjectObserver::Update(int iMsg)
{
    const list<void*>* pDataList = Engine::Get_InfoSubject()->GetDataList(iMsg);
    if (pDataList == NULL)
        return -1;

    switch (iMsg)
    {
        case MSG_INFOLIST:
            m_pInfolist = (const list<Engine::CTransform*>*)pDataList;
            break;
        case MSG_PLAYERINFO:
            m_pPlayerInfo = S_CAST(const Engine::CTransform*, pDataList->front());
            break;
        case MSG_MONSTERLIST:
            m_pMonsterlist = (const list<Engine::CTransform*>*)pDataList;
            break;
        case MSG_BULLETLIST:
            m_pBulletlist = (const list<Engine::CTransform*>*)pDataList;
            break;
    }
}
```

16) 무게값을 기준으로 애니메이션보간

```
// Pos, normal and texture coordinates.
SkinnedVertex vertex(v.x, v.y, v.z, n.x, n.y, n.z, 0, 0, 1, t.x, t.y);

// Bone indices and weights.
for (size_t k = 0; k < vecWeights[i].boneIndices.size(); ++k)
{
    if (k < 4)
        vertex.BoneIndicesA[k] = vecWeights[i].boneIndices[k];
    else
        vertex.BoneIndicesB[k - 4] = vecWeights[i].boneIndices[k];
}

if (vecWeights[i].weights.size() > 8)
    int iA = 0;

vertex.WeightsA.x = vecWeights[i].weights.size() >= 1 ? vecWeights[i].weights[0] : 0;
vertex.WeightsA.y = vecWeights[i].weights.size() >= 2 ? vecWeights[i].weights[1] : 0;
vertex.WeightsA.z = vecWeights[i].weights.size() >= 3 ? vecWeights[i].weights[2] : 0;
vertex.WeightsA.w = vecWeights[i].weights.size() >= 4 ? vecWeights[i].weights[3] : 0;

vertex.WeightsB.x = vecWeights[i].weights.size() >= 5 ? vecWeights[i].weights[4] : 0;
vertex.WeightsB.y = vecWeights[i].weights.size() >= 6 ? vecWeights[i].weights[5] : 0;
vertex.WeightsB.z = vecWeights[i].weights.size() >= 7 ? vecWeights[i].weights[6] : 0;
vertex.WeightsB.w = vecWeights[i].weights.size() >= 8 ? vecWeights[i].weights[7] : 0;

vertices.push_back(vertex);
```

I -iii. 팀단위 프로젝트 - 3

프로젝트	DirectX11,IOCP,TCP/IP,HLSL,MFC (학교 졸업작품)	지도교수님	송인희
기간	2018년 3월~2018 5월 (2개월)	언어	C++
제목	배틀 건 (Battle Gun)		
주요내용	노말맵, 빛을 이용한 셰이더 출력. StreamOutPut을 이용한 파티클 랜더. 컴포넌트 패턴을 사용한 프레임워크. 스텐실버퍼를 이용한 셰도우맵핑,안개적용. ssao기법을 사용한 셰이더 출력. 업저버 패턴을 이용한 구충돌. IOCP를 활용한 네트워크 관리 시스템. 키매니저(비트연산)을 통한 Key입력. mfc맵틀을 이용한 네비게이션 메쉬 적용.		

1) 텍스처값만 적용



2) Color값만 적용



3)노말값을 이용한 노말맵핑



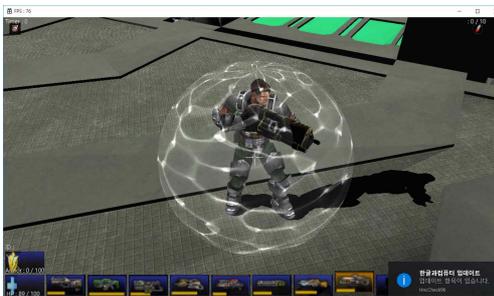
3)타격시 파티클,이펙트



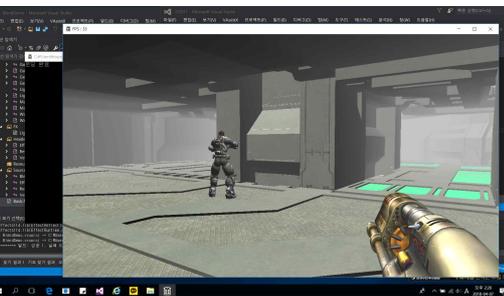
4)쉐도우 맵핑.



5)아이템 획득.



6)안개적용



결과

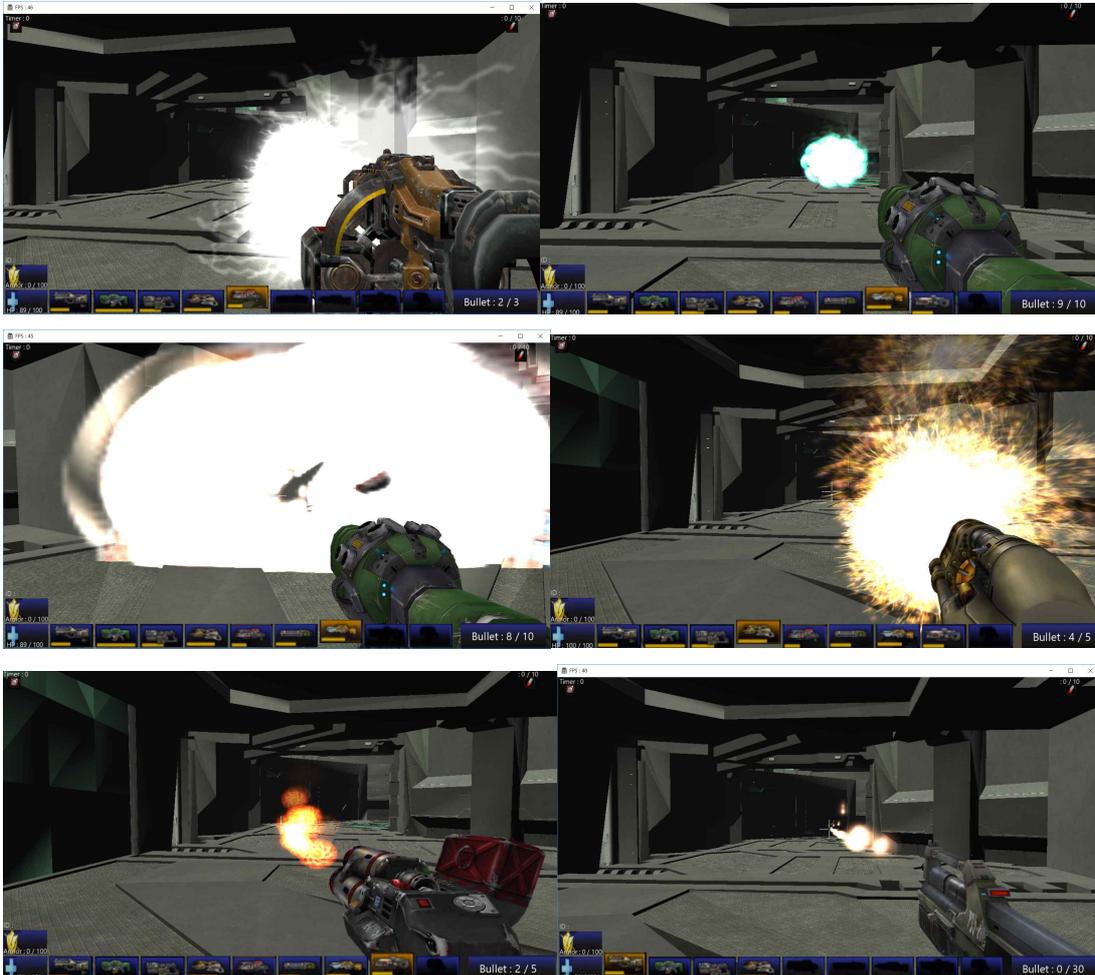
베트맨 아캄나이트에서 사용하지 못했던 셰이더 위주로 공부했으며 안개맵,그림자,SSAO등을 적용시켰다. IOCP에 기본적인 이해와 서버에대해 한층더 깊이 알게된 프로젝트이다. FPS에대한 기본적인 카메라,랜더링 파이프라인 개념등을 다시금 복습하게되는 계기도 되었고 Github를 이용하여 소스를 관리하였다.

배틀 건(Battle Gun) 스크린샷

7) 구충들을 이용한 충돌처리.



8) 각종 총알 파티클



DirectX11 소스 코드(C++언어)

1) 조명과 노말값을 이용하여 픽셀셰이더에서 연산하는 코드

```
///  
// Lighting  
float3 normalMapSample = gNormalMap.Sample(sanLinear, pin.Tex).rgb;  
float3 bumpedNormalW = NormalSampleToWorldSpace(normalMapSample, pin.NormalW, pin.TangentW);  
float4 litColor = texColor;  
{  
    // Start with a sum of zero.  
    float4 ambient = float4(0.0f, 0.0f, 0.0f, 0.0f);  
    float4 diffuse = float4(0.0f, 0.0f, 0.0f, 0.0f);  
    float4 spec = float4(0.0f, 0.0f, 0.0f, 0.0f);  
  
    // Finish texture projection and sample SSAO map.  
    pin.SsaoPosH /= pin.SsaoPosH.w;  
    float ambientAccess = gSsaoMap.SampleLevel(sanLinear, pin.SsaoPosH.xy, 0.0f).r;  
  
    // Sum the light contribution from each light source.  
    [unroll]  
    for (int i = 0; i < gLightCount; ++i)  
    {  
        float4 A, D, S;  
        ComputeDirectionalLight(gMaterial, gDirLights[i], bumpedNormalW, toEye,  
            A, D, S);  
  
        ambient += ambientAccess*A;  
        diffuse += D;  
        spec += S;  
    }  
    // Modulate with late add. : 변조 후 가산  
    litColor = texColor * (ambient + diffuse) + spec;  
}
```

2) 알파블렌딩 셰이더 코드 3)안개적용하는 셰이더 코드

```
BlendState AdditiveBlending  
{  
    AlphaToCoverageEnable = FALSE;  
    BlendEnable[0] = TRUE;  
    SrcBlend = SRC_ALPHA;  
    DestBlend = ONE;  
    BlendOp = ADD;  
    SrcBlendAlpha = ZERO;  
    DestBlendAlpha = ZERO;  
    BlendOpAlpha = ADD;  
    RenderTargetWriteMask[0] = 0x0F;  
};  
  
if (gFogEnabled)  
{  
    float fogLerp = saturate((distToEye - gFogStart) / gFogRange);  
    // Blend the fog color and the lit color.  
    litColor = lerp(litColor, gFogColor, fogLerp);  
}
```

4) 플레이어의 그림자를 바닥에 적용시키는 코드

```
for (UINT p = 0; p < techDesc.Passes; ++p)  
{  
    ID3DX11EffectPass* pass = m_pMyGDI->pTech->GetPassByIndex(p);  
    m_pMyGDI->pContext->IASetVertexBuffers(0, 1, &m_pVB, &stride, &offset);  
    m_pMyGDI->pContext->IASetIndexBuffer(m_pIB, DXGI_FORMAT_R32_UINT, 0);  
  
    XMVECTOR shadowPlane = XMVectorSet(0.0f, 5.0f, 0.0f, 0.0f); // xz plane  
    XMVECTOR toMainLight = -XMloadFloat3(&mDirLights[0].Direction);  
    XMATRIX S = XMMatrixShadow(shadowPlane, toMainLight);  
    XMATRIX shadowOffsetY = XMMatrixTranslation(0.0f, 0.01f, 0.0f);  
  
    // Set per object constants.  
    XMATRIX world = matWorld*S*shadowOffsetY;  
    XMATRIX worldInvTranspose = MathHelper::InverseTranspose(world);  
    XMATRIX worldViewProj = world*view*proj;  
  
    m_pMyGDI->pEffect->SetWorld(world);  
    m_pMyGDI->pEffect->SetWorldInvTranspose(worldInvTranspose);  
    m_pMyGDI->pEffect->SetMaterial(mShadowMat);  
  
    m_pMyGDI->pContext->OMSetDepthStencilState(RenderStates::NoDoubleBlendDSS, 0);  
    pass->Apply(0, m_pMyGDI->pContext);  
    m_pMyGDI->pContext->DrawIndexed(m_iNumIndices, 0, 0);  
  
    // Restore default states.  
    m_pMyGDI->pContext->OMSetBlendState(0, blendFactor, 0xffffffff);  
    m_pMyGDI->pContext->OMSetDepthStencilState(0, 0);  
}
```

DirectX11 소스 코드(C++언어)

5) 지오메트리 셰이더에서 빌보드적용, Compute Triangle하는 부분.

```
// The draw GS just expands points into camera facing quads.
[maxvertexcount(4)]
void DrawGS(point VertexOut gin[1], inout TriangleStream<GeoOut> triStream)
{
    if (gin[0].Type != PT_EMITTER)
    {
        float3 look = normalize(gEyePosW.xyz - gin[0].PosW);
        float3 right = normalize(cross(float3(0, 1, 0), look));
        float3 up = cross(look, right);
        float halfWidth = 0.5f*gin[0].SizeW.x;
        float halfHeight = 0.5f*gin[0].SizeW.y;
        float4 v[4];
        v[0] = float4(gin[0].PosW + halfWidth*right - halfHeight*up, 1.0f);
        v[1] = float4(gin[0].PosW + halfWidth*right + halfHeight*up, 1.0f);
        v[2] = float4(gin[0].PosW - halfWidth*right - halfHeight*up, 1.0f);
        v[3] = float4(gin[0].PosW - halfWidth*right + halfHeight*up, 1.0f);
        // Transform quad vertices to world space and output them as a triangle strip
        GeoOut gout;
        [unroll]
        for (int i = 0; i < 4; ++i)
        {
            //gout.PosH = mul(v[i], gViewProj);
            gout.PosH = mul(v[i], gViewProj);
            gout.Tex = pQuadTex[i];
            gout.Color = gin[0].Color;
            triStream.Append(gout);
        }
    }
}
```

6) 플레이어에 필요한 자료를 컴포넌트화 하여 붙이는 부분.

```
HRESULT CCharacter::AddComponent(void)
{
    Engine::CComponent* pComponent = NULL;
    //Transform
    pComponent = m_pInfo = Engine::CTransform::Create(g_vLook);
    NULL_CHECK_RETURN(pComponent, E_FAIL);
    m_mapComponent.Insert(MAPCOMPONENT::value_type(L"Transform", pComponent));

    //Mesh
    pComponent = m_pResourceMgr->OneResource(RESOURCE_STAGE, L"DynamicMesh_HumanMaleA");
    m_pCharMesh = dynamic_cast<Engine::CDynamicMesh*>(pComponent);
    NULL_CHECK_RETURN(pComponent, E_FAIL);
    m_mapComponent.Insert(MAPCOMPONENT::value_type(L"DynamicMesh", pComponent));

    //SphereCol
    pComponent = m_pSphereCol = CSphereCol::Create(m_pMyGD1, 6.0f);
    NULL_CHECK_RETURN(pComponent, E_FAIL);
    m_pSphereCol->SetTargetInfo(m_pInfo);
    m_mapComponent.Insert(MAPCOMPONENT::value_type(L"SphereCol", pComponent));

    return S_OK;
}
```

7) 플레이어 정보를 찾아 아이템과 충돌처리 하는 부분 8)키매니저 변수정의

```
void CBulletMgr::ItemColltoSphere(XMFLOAT3* vPlayerPos, XMFLOAT3* vItemPos,
float fRadius, XMFLOAT3* vResult, int iIndex, int* pItemNum)
{
    int iHitIndex;
    ITEMGET_eItemTarget = ITEM_NONE;

    auto pMapUser = CUserMgr::GetInstance()->GetUserList();

    auto iter = pMapUser->begin();
    auto iter_end = pMapUser->end();

    for (iter; iter != iter_end; ++iter)
    {
        if (iter->second->GetUserInfo().iIndex == iIndex
            || iter->second->GetCharacter()->GetCharInfo().eState == CHARSTATE_DEAD)
            continue;

        if (CheckSphere(vPlayerPos, vItemPos,
            (iter->second->GetCharacter()->GetCharInfo()->GetColRadius())[0], fRadius, vResult))
        {
            if (*pItemNum == ITEM_IDLE) // 충돌하기전 상태값
                *pItemNum = ITEM_COLLISION; // 충돌후
            iHitIndex = iter->second->GetUserInfo().iIndex;
            eItemTarget = ITEM_SHIELD;
            //여기서 아이템 막혔다고 서버로 알려주면된다.
        }
    }
}
```

```
#define KEY_UP 0x01
#define KEY_DOWN 0x02
#define KEY_LEFT 0x04
#define KEY_RIGHT 0x08
#define KEY_SPACE 0x10
#define MOUSE_LEFT 0x20
#define MOUSE_RIGHT 0x40
#define WEAPON_CHANGE 0x80
```

II- i. 멘토링 - 1

활동명	후배사랑 멘토링	지도교수	임창주 교수님
주관기관	한국산업기술대학교 게임공학과, 교수학습지원센터	역할	멘티
활동기간	2015.03.01.~2015.11.30.(약 1년)		
주요내용	한 학기에 학과당 5명씩 멘티를 학과에서 선정후 선정된 멘티들은 지정된 멘토20명과 정기적으로 만나며 학교생활의 고충이나 미래 취업에 대한 조언등을 주면서 후배들의 학교생활을 도와준다.		

후배사랑 멘토제 운영방안

- 멘토 활동기간 : 2개학기(2015 학년도 1, 2학기)
- 멘토 교육 : 교수학습지원센터와 협조 사전 교육 실시
- 멘토 임명장 발급 : 선정된 멘토들에게 임명장 수여를 통해 소속감 및 책임감 부여
- 멘토 활동비 지급 : 멘토교육을 수료한 자에 한하여, 학기당 최대 300,000원의 멘토 활동비 지급
- 멘토 - 멘티 수기 공모전을 통한 포상
- 멘토와 멘티가 한그룹이 되어 정기적 만남(학기당 최소 2회 이상)을 실시, 멘토활동 보고서(첨부 2)를 작성하여 학부사무실로 제출.
- 학과(부)에서는 제출된 '멘토활동보고서'를 학과장 확인을 거쳐 교무처에 제출.
- 교무처는 학과(부) 통보내용에 기초하여 멘토활동비 지급.
- 멘토활동비 : 멘토활동 1회에 멘티 1인당, 1만원 지급.

멘토활동보고서

작성일자	2015.11.17		
멘토성명	오준석		
활동구분	멘토 활동		
활동일시	11 월 13 일	멘토링 장소	구이갸(고기집)
활동내용	15학년 후배들을 11명과 윤기들과 구이갸에 모여서 저녁식사를 같이하며 후배님들의 고민과 학교에대한 궁금증 및 공부에대한 방향과 관심사 군대가는 것등 개인적인 고민등을 이야기하였습니다. 그리고 다같이 고기를 먹고싶다고 하여 고기집에서 식사를 해서그런지 다들 기분이 좋아보였습니다. 다들 바쁘지만 이러한 계기라도 있어서 다같이 모여 좋았습니다. 앞으로 후배들과 자주 만나 조금이나마 도움이되는 선배가 되도록 준비를 많이해야겠다고 생각했습니다.		
참여멘티	총 11명, 성명 : 곽용석, 김덕규, 김명석, 김영환, 김용민, 김지웅, 문석준, 박건호, 박남계, 박정만, 장성운		
다음 활동계획	이번학기 마지막 멘토활동입니다.		
은행명	신한은행	계좌번호	110-334-082060
		예금주	오준석
		학과장	(인)



멘토링 활동 보고서

멘토활동보고서

작성일자	2015.09.17		
멘토성명	오준석		
활동구분	멘토 활동		
활동일시	09 월 17 일	멘토링 장소	TIP지하1층
활동내용	<p>15학년 후배님들 11명과 TIP건물 지하 1층에 모여서 가볍게 점심식사를 같이 하며 후배님들의 고민과 학교에대한 궁금증 및 공부에대한 방향을 의논하였고 한명한명씩 우리과에온 배경과 다짐을 들어보았습니다. 게임공학과라는 이름하나로 모여 각자 자신이 나아갈방향에대해 조금이나마 도움을 줬다는 뿌듯함이 넘치는 모임이었습니다. 앞으로 후배님들과 자주 만나 조금이나마 도움이되는 선배가 되도록 준비를 해야하겠다고 생각했습니다.</p>		
참여멘티	총 11명, 정명 : 고희석, 김덕규, 김병석, 김영환, 김용민, 김지용, 문석준, 박건호, 박남제, 박정만, 정성은		
다음 활동계획	다음 활동에는 후배님들이 좀더 좋아하는 피자나 치킨같은걸 같이 먹으며 담소를 나누고 고민도 상담할 예정입니다.		
은행명	신한은행	계좌번호	110-334-082060
	예금주	오준석	
	학과장	(인)	

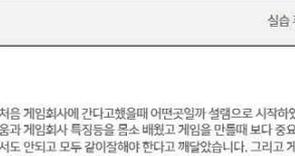
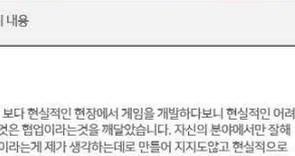
멘토활동보고서

작성일자	2015.05.22		
멘토성명	오준석		
활동구분	멘토 활동		
활동일시	05 월 22 일	멘토링 장소	총합관2층 그라운드디플
활동내용	<p>15학년 후배님들 12명과 총합관 2층 그라운드디플 3호실에 모여서 후배님들의 고민과 학교에대한 궁금증 및 공부에대한 방향을 의논하였고 한명한명씩 우리과에온 배경과 다짐을 들어보았습니다. 게임공학과라는 이름하나로 모여 각자 자신이 나아갈방향에대해 조금이나마 도움을 줬다는 뿌듯함이 넘치는 모임이었습니다. 앞으로 후배님들과 자주 만나 조금이나마 도움이되는 선배가 되도록 준비를 해야하겠다고 생각했습니다.</p> <p>우리학과에서 배우는 과목과 커리큘럼, 게임회사에대한 진路和 실상 및 취업과 학점등에대한 심도있는 TIP를준비하여 체계적인 수준에서 최대한 후배님들에게 도움이 되도록 설명하였습니다.</p>		
참여멘티	총 12명, 정명 : 고희석, 김덕규, 김병석, 김정은, 김영환, 김용민, 김지용, 문석준, 박가은, 박건호, 박남제, 박정만		
다음 활동계획	도서관에있는 그라운드디플하나를 빌려서 C언어질문과 1학년들이 어려워하는 문제나 프로그래밍 언어를 설명할것입니다.		
은행명	신한은행	계좌번호	110-334-082060
	예금주	오준석	
	학과장	(인)	



II-ii. 현장실습 - 2

실습기업명	(주)드림로스팅	지도교수	송인희 교수님
실습기간	2016~06.27 ~ 2016-08.19	기업담당자	배경환
실습개요	취업하기전 각 학과에 해당하는 회사에 파견하여 언제든 현장에 투입할 수 있는 능력을 배양.		
주요내용	유니티에 대한 전반적인 내용과 게임회사의 실질적인 개발분야와 어떻게 돈을 벌고 어떻게 마케팅하는지에 대해 몸서 체험하였다. 엔진의 편리함과 강력함 그리고 기획과 그래픽간 팀단위 회의내용이 실제 기업에서하는 방식을 몸소 체험해보니 커뮤니케이션의 중요성을 깨달았다.		

현장실습 증명서		현장실습 종합보고서																																														
<table border="1"> <tr><td>학과</td><td colspan="3">게임공학부</td></tr> <tr><td>학년</td><td colspan="3">2012181023</td></tr> <tr><td>성명</td><td colspan="3">오준석</td></tr> <tr><td>학기</td><td>2016년도 여름학기</td><td>교과목명</td><td>현장실습3</td></tr> <tr><td>실습기간</td><td colspan="3">2016-06-27~2016-08-19 (8주)</td></tr> <tr><td>실습기관</td><td colspan="3">(주)드림로스팅</td></tr> <tr><td>실습기관 주소</td><td colspan="3">서울 구로구 구로동 디지털로33길 55 301호</td></tr> <tr><td>업무역량</td><td colspan="3">유니티에 관련된 사버,오버월드, 초재기, 클라이언트, 동등 코드를 저승과 동영성 그리고 선 배터지 회사 프로그래머한테 질문하여 배웠고 전반적인 게임회사가 기획부터 그래픽 프로그래머까지 오는 분야가 어떻게 돌아가는지 현실성을 깨달았습니다.</td></tr> </table> <p>상기 학생은 한국산업기술대학교에서 주관하는 국내 현장실습 프로그램에 참여하여 이수함을 증명합니다.</p> <p>2017년 9월 17일</p> <p>한국산업기술대학교 총장 </p>		학과	게임공학부			학년	2012181023			성명	오준석			학기	2016년도 여름학기	교과목명	현장실습3	실습기간	2016-06-27~2016-08-19 (8주)			실습기관	(주)드림로스팅			실습기관 주소	서울 구로구 구로동 디지털로33길 55 301호			업무역량	유니티에 관련된 사버,오버월드, 초재기, 클라이언트, 동등 코드를 저승과 동영성 그리고 선 배터지 회사 프로그래머한테 질문하여 배웠고 전반적인 게임회사가 기획부터 그래픽 프로그래머까지 오는 분야가 어떻게 돌아가는지 현실성을 깨달았습니다.			<table border="1"> <tr><td>성명</td><td>오준석</td><td>학과</td><td>게임공학부</td></tr> <tr><td>실습기간</td><td>2016-06-27~2016-08-19</td><td>학년</td><td>2012181023</td></tr> <tr><td>실습기관명</td><td>(주)드림로스팅</td><td>기관 담당자</td><td>배경환</td></tr> </table>			성명	오준석	학과	게임공학부	실습기간	2016-06-27~2016-08-19	학년	2012181023	실습기관명	(주)드림로스팅	기관 담당자	배경환
학과	게임공학부																																															
학년	2012181023																																															
성명	오준석																																															
학기	2016년도 여름학기	교과목명	현장실습3																																													
실습기간	2016-06-27~2016-08-19 (8주)																																															
실습기관	(주)드림로스팅																																															
실습기관 주소	서울 구로구 구로동 디지털로33길 55 301호																																															
업무역량	유니티에 관련된 사버,오버월드, 초재기, 클라이언트, 동등 코드를 저승과 동영성 그리고 선 배터지 회사 프로그래머한테 질문하여 배웠고 전반적인 게임회사가 기획부터 그래픽 프로그래머까지 오는 분야가 어떻게 돌아가는지 현실성을 깨달았습니다.																																															
성명	오준석	학과	게임공학부																																													
실습기간	2016-06-27~2016-08-19	학년	2012181023																																													
실습기관명	(주)드림로스팅	기관 담당자	배경환																																													
<p>현장실습 후기</p> <table border="1"> <tr><td>학과</td><td>게임공학부</td><td>이름</td><td>오준석</td></tr> <tr><td>학년</td><td>3</td><td>학년</td><td>2012181023</td></tr> <tr><td>실습기간</td><td colspan="3">2016-06-27 ~ 2016-08-19 (8 주)</td></tr> </table>     <p>실습 후기 내용</p>		학과	게임공학부	이름	오준석	학년	3	학년	2012181023	실습기간	2016-06-27 ~ 2016-08-19 (8 주)			<p>실습개요 및 목표</p> <p>유니티에 대한 전반적인 내용과 게임회사의 현실과 미래에 대한 프로그래밍에 관한 모든것을 몸소 체험하고 깨달을려고합니다. 그리고 게임회사의 실질적인 개발분야와 어떻게 돈을 벌고 어떻게 사는지 현실적인 곳을 가서 몸소 체험할 것입니다.</p>																																		
학과	게임공학부	이름	오준석																																													
학년	3	학년	2012181023																																													
실습기간	2016-06-27 ~ 2016-08-19 (8 주)																																															
<p>실습내용</p>		<p>주마다 유니티에 관한 계획을 세우고 계획세운것을 토대로 교제와 동영상강의를 들으며 직접 프로젝트를 실습하고 모르는 부분은 학교 선배님이나 회사의 프로그래머들에게 물어봐서 배웠습니다.</p>																																														
<p>실습 전후의 차이점 (개선사항)</p>  		<p>보다 현실적인 코딩과 엔진의 편리함과 강력함 그리고 실제로 게임을 만들수있는 유니티 도구의 중요성을 몸소 실천하여 깨달았고 졸업작품을 위해는 다이어트로 하려고하였으나 유니티의 위대함과 편리성에 반하여 유니티로 하기로 졸업팀원들과 상의를 맞췄습니다. 그리고 다이어트 개념들도 꼭 같이 공부하기로 할것입니다.</p>																																														
<p>건의사항 및 예로서항</p>		<p>중간발표나 회사가 이동하는것에 따른 개별적 연락이 없어서 불편한점 빼고는 없습니다. 건의사항으로는 보다 많은 회사와 연관되어서후배님들의 회사 선택의 폭이 넓어졌으면 좋겠고 수당도 많이 받았으면합니다.</p>																																														
<p>처음 게임회사에 간다고했을때 어떤곳일까 싶으로 시작하였고 보다 현실적인 현장에서 게임을 개발하다보니 현실적인 어려움과 게임회사 특징등을 몸소 배웠고 게임을 만들때 보다 중요한것은 협업이라는것을 깨달았습니다. 자신의 분야에서만 잘해 서도 안되고 모두 같이잘해야 한다고 깨달았습니다. 그리고 게임이라는게 재가 생각하는대로 만들어 지지않고 현실적으로 구현할수있는 한계가 있다는것을 깨달았습니다. 그리고 사람관계도중요하고 보다 현실적이고 구체적인 계획서를 쓰고 개발에 임해야한다는것도 깨달았습니다. 가장 많이 깨달았던것은 프로그래밍이 만만하지않지만 엔진의 힘을 깨달았습니다. 좋은 경험은 한것같아 부도합니다. 이상입니다.</p>																																																

II-iii. 연구실 - 3

실습기업명	(주)엑티미디어(가상현실 유통과정논문 보조와 VR개발)	소속대학	한국산업기술대
실습기간	2016~09.22 ~ 2016-011.24	지도교수	이대현 교수님
주요내용	학과장이신 이대현교수님의 VR개발 연구실에서 VR연구논문 보조, VR게임을 실제 오컬러스에 연동하여 게임개발, 드론조작방법 및 장비 관리, 교수님 수업조교		

별첨

2016 가상현실을 이용한 훈련·학습·재능 콘텐츠 아이디어 공모전 참가신청서 [개인]

신청자	성명	유준석	생년월일	1992.10.26
	주소	경기도 부천시 소사구 괴안동 범박휴먼시아 106동506호		
	연락처	010 7206 0506	e-Mail	topojs8@naver.com
	소속	한국산업기술대학교 게임공학과		
공작품	작품명	VR게임으로 재활치료		
	기획의도	뇌기능 손상환자의 기능개선 고소공포증 극복 신체의 특정부위를 YAG레이저 운동시킴으로써 게임을 결합하여 보다 재미있게 재활치료하고 치료효과를 향상시킬것이다.		
	작품설명	마이크로소프트의 키넥트(Kinect)를 이용한 '모션 게임'은 깊이(depth) 이미지를 형성하고 이렇게 구축된 이미지를 브래드 일렉트 인공지능을 만들어 게임 속 아바타를 통해 자신이 의도한 움직임을 화면 속에서 실현하는 원리		
기대효과	손에 감각이 없는 사람에게 가상의 물체를 보듯이 감각을 돌아오게 트레이닝. 거동이 불편한사람에게 가상현실에서 움직여 움직임의 물건을 가지고 움직일수있게하여 . 정확한노력까지 해줄수있다. 평소에는 할 수 없는 핵의 역할지나 심지어 단, 화성 등의 장소에서도 치료를 할 수 있는 제품입니다.			

※ 개인정보는 절대 외부로 유출되지 않으며, 상업적 용도로 사용되지 않음

Diorama :배경을 그림 막 앞에 여러 가지 물건을 배치하고 그것을 잘 조명하여 실물감이 나
게 하는 장치, 투시화, 배경 위에 모형물 설치해 하나의 장면을 만든 것.

단어 "디오라마"는 단어는 문자 그대로에서, "그 통제 볼 수있다"라는 뜻 1822 년 프랑스에
서 시진 감상 장치의 유형으로 1823 년 시작될 그리스어 디 + orama ""을 통해 "볼 수있
는 것을 , 시력 ", 디오라마가 발명 한 다케르 와 찰스 마리 Boutou은 첫 번째 전시, 런던
9 월 29 일 1823 "등 장면의 소규모 복제,"이미 1902 내지 [1]

다케르의 디오라마 양쪽에 그려진 재료의 조각으로 구성되었다. 장면에서 조명 할 때, 장면
은 하나의 상태와 볼 것이다 또 다른 단계 또는 측면 뒤에서 조명을 위해 전환으로 표시되
니다. 열광 장면 트랙 여행 기차가 충돌 것. 달빛으로 변경, 또는 지진은 이전과 사진 뒤에
표시됩니다.

visor head up display (HUD) :사용자가 필요없이 데이터를 제공하는 투명 디스플레이입니
다. 이름의 유래는 유래 파일럿 "최대"위치와 기대 헤드와 정보를 보려면, 대신에 낮은 약기
를 찾고 이래로 각도 할 수있는, HUD는 조종사의 눈이 필요없는 장점이었다 집중할
필학 가까운 약기보고 후 외부로 볼 수 있습니다.

Head Up Display의 약자로 차량의 현재 속도, 연료 잔량, 내비게이션 길 안내 정보 등을
운전자 바로 앞 유리창 부분에 그래픽 이미지로 투영해 주어 운전자가 불필요하게 시선을
다른 곳으로 옮기는 것을 최소화 시켜주는 디스플레이 장치입니다.

주로 전투기나 항공기에 쓰이던 디스플레이 방식으로 앞유리에 반사되는 정보를 볼 수 있어
운전하는 전방의 시선을 끊지 않고도 계기를 읽을 수 있기 때문에 빠른 반응을 할 수 있는
것이 특징입니다. LCD패널, 백라이트유닛(BLU), 반시멘즈 등으로 구성되며, 주로 운전석
계기판 뒤쪽의 대시보드에 설치됩니다.

reticle cursors : 십자선 커서

windshield hud : 폭우, 안개 등으로 운전 중 앞이 안보일 때, 증강현실 네비를 통해 내가
가는 길을 보여주는 기술.

레티클(reticle):가로 세로 1줄씩은 십자선, 대물경 혹은 렌즈의 초점에 레티클을 두고, 이를
별과 함께 루페로 보게 되는데, 보통은 점안렌즈 쪽에 선을 쳐서 점안렌즈의 시선조절을 해
준다. 선의 근처의 거미줄 등이 흔히 이용되는데, 유리에 선이 그려진 것을 사용하기도 한
다.



작품설명



- ▶ 사람의 손에 가장 많은 감각이 집중된다고하는데 이 감각으로 컨트롤할수있는 가상현실 체육관을 만드는 것입니다.
- ▶ 장갑을통해 실제 체육관에서 아령을 들면 실제 아령의 무게와 같이 무게감을 주어 근력을 강화시키는 것입니다.
- ▶ 또한 신발을 신음으로써 실제 걷지않지만 체육관에서 걷는듯한 느낌을 줄수있습니다.
- ▶ 게임뿐 아니라 각종 운동을 하는데도 매우 유용하게 활용할 수 있습니다.

II-iii. 인턴생활 - 4

기업명	KOG(엘소드 클라이언트)	소속파트	클라이언트
인턴기간	2018~01.02 ~ 2018-03.23	담당국가	브라질, 태국
주요내용	목찌바 미니게임 제작, 서버관리툴을이용한 브라질 태국 패치, 각종 버그이슈 및 라이브 서버 이슈대응, 루아스크립트를 이용한 UI개발및 해외통합, 텍스처 배치툴 개발, 가열기 튜토리얼(데이터베이스 공용테이블사용), 월드맵 기획서를 통한 개선점 개발.		

목차

1. [튜토리얼]목찌바 매니저.
2. [QA] 온천에서 마리포사 레퀴엠인장 이펙트가 물에 안잠기는 문제.
3. [QA] 2인 앉기 상태를 해제 시 같이 앉은 유저의 반응속도가 느린 문제.
4. [QA] 댄스 모션 중 앉기 사용 시 다른사람의 화면과 사용자의 화면이 다르게 노출되는 현상.
5. [개선사항] pvp대전시 নিজ사의캐릭터로 나오게하기 및 분할.
6. [개선사항] 퀘스트 원하는 장 UI에 드래그앤드롭.
7. [개선사항] TGA파일배치mfc툴프로토타입.

KOG'

[튜토리얼]목찌바 매니저



상태변수하나를 추가하여 가위바위보 상태에서 이기고 비겼는지 목찌바 상태에서 졌는지 판단하는 변수를 추가하여 만들었습니다.

[QA]온천에서 마리포사 레퀴엠인장 이펙트가 물에 안잠기는 문제

KOG'



플레이어가 물이 위치한 라인에 위에 있을때 인장 이펙트가 쉼터 안뒤 게 조건을 주어 해결하였습니다.

[QA] 2인 앉기 상태를 해제 시 같이 앉은 유저의 반응속도가 느린 문제

KOG'



2인 앉기중 승스트가 일어났을때 2번째 클라이언트의 상태를 ET_STANDUP으로 x2GUUser.cpp 에서 조건을 했습니다.

[QA] 댄스 모션 중 앉기 사용 시 다른사람의 화면과 사용자의 화면이 다르게 노출되는 현상

KOG'



2번째 클라이언트 앉기 상태값인 m_ePlayedEmotion을 첫번째 클라이언트가 ET_NONE인 상태일때 STANDUP상태가 아니면 다시 SITEADY상태로 바꿨습니다.

[개선사항] pvp대전시 নিজ사의 캐릭터로 나오게하기 및 분할

KOG'



미리 선언해둔 ui변수를 통해 2개의 uiDialog를 만들고 m_iPlayerSlotClass에 담아둔 각 슬롯의 직업정보들을 문자열에 담고 플레이어가 인원수에 따라 ui의 개수와 크기 위치를 일맞게 셋팅했습니다.

KOG 인턴 결과물

[개선사항] 퀘스트 원하는 창 UI에 드래그앤드롭

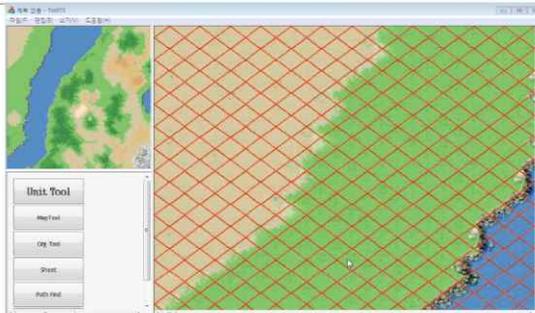
KOG'



CX2UIQuestNew::OnFrameMove() 함수에서 m_bMouseDrag 변수가 true 일때 현재 마우스의 위치에서 버튼 가로 세로 사이즈차이를 빼고 퀘스트 UI를 이동하게 하였다.

[개선사항] TGA파일배치 MFC 툴을 이용한 프로토타입

KOG'



MFC 툴을 이용해 기존 Lua 스크립트의 정보 형태로 파일입출력되는 텍스처 툴을 만들었습니다.

느낀점

KOG'

실제 Top20위 안에 들고 라이브 서비스하는 엘소드 게임의 코드를 내가 직접 생각하는 로직을 적용시키고 버그, 개선할 점을 직접 개발할 수 있어서 보람찼습니다. 그리고 프로그래머 로써가 아닌 게임회사에 게임 개발자로 일할 수 있어서 좋았고 다른 분야의 사람들과 소통할 수 있어서 좋았습니다.

II-iii. 동아리 및 기타활동 - 5

동아리명	BB(Bowling & BasketBall)	소속대학	한국산업기술대
활동기간	2015.03.01.~ 현재	지도교수	임창주 교수님
주요내용	본 동아리는 볼링과 농구를 접목시킨 게임공학부 소모임입니다. 저는 2015년도 부회장을 하면서 학교 농구체전 준우승까지하였고 농구만이 아닌 각종 레저활동 및 친목단체로서 게임공학부 학생들끼리의 끈끈한 정을 느낄 수 있는 동아리입니다. 농구뿐만 아니라 다른 학교와의 연합동아리인 EXIT 3기출신입니다. 이 동아리의 장점은 다양한 분야의 사람들과 교류하였다는 점인데 공대뿐만 아니라 우리 학교의 단점을 보다 다양한 사람들과 취미활동을 하며 사회생활을 익혔습니다.		



동아리 및 기타활동



포트폴리오 동영상 링크

1) 드래곤볼 (C++,API)

<https://www.youtube.com/watch?v=QSDqeV8cXok>

2) 물고기키우기 (C++,API)

<https://www.youtube.com/watch?v=vPwxexhaUn8>

3) 포켓몬스터 (C++,API)

<https://www.youtube.com/watch?v=7gbaN1whk6c>

4) 3차원 총알피하기 (C++,OpenGL)

<https://www.youtube.com/watch?v=7GFXQjwuWe0>

5) 런닝 큐브 (C++,OpenGL)

<https://www.youtube.com/watch?v=Aqlr995VafM>

6) 식물 vs 좀비 (Python, Pico2d)

<https://www.youtube.com/watch?v=pN-WZPFfpwE>

7) 오버워치 (Unity, pc, mobile, C#)

<https://www.youtube.com/watch?v=upW0U58b8VY>

8) 메이플스토리 (C++,API)

https://www.youtube.com/watch?v=XU_GEw_vmxs&feature=youtu.be

9) 스톤에이지 (C++, Direct2D, MFC)

https://www.youtube.com/watch?v=hdm_kycRsY

10) 블레이드 앤 소울 (C++, Direct3D, MFC)

<https://www.youtube.com/watch?v=NFvZv3yfaig&feature=youtu.be>

11) 마법사의하루 (C#, Unity, VR)

<https://www.youtube.com/watch?v=e5Kng51ufy8&t=1s>

12) 드래곤볼서버 (C++, API, TCP/IP)

<https://www.youtube.com/watch?v=Tenbp6ckip4&t=75s>

13) 오버워치 (TCP/IP, DirectX9, MFC, C++)

https://www.youtube.com/watch?v=Abtu_jMtIJI&feature=youtu.be

14) 베트맨 아캄시티 (DirectX11, HLSL, MFC, C++)

<https://www.youtube.com/watch?v=BASgtN9sVHk&feature=youtu.be>

15) 배틀 건 (DirectX11, HLSL, IOCP, MFC, C++)