김경엽
Portfolio

H.P : 010-7226-1848
E.Mail : kkcom23@naver.com

INDEX.

1

유니티 게임
(개인), (팀)

(링크)

2

로그라이크 게임
창작
(개인)

(링크)

3

로그라이크 게임
(Crypt of the NecroDancer)
모작
(팀)

(링크)

4

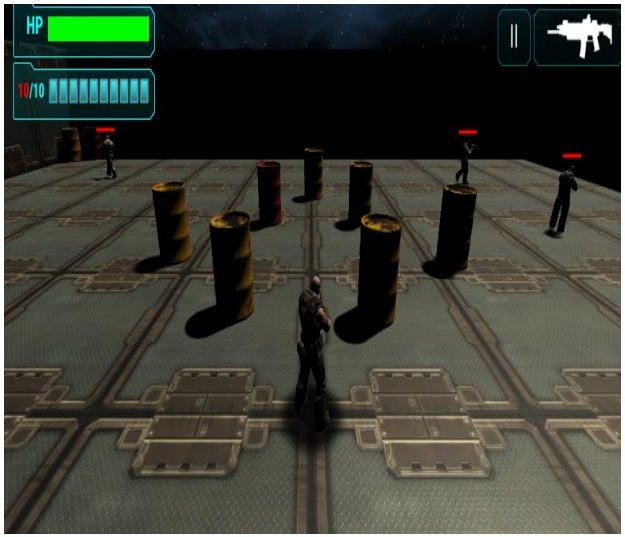
슈팅게임
창작
(개인)

링크

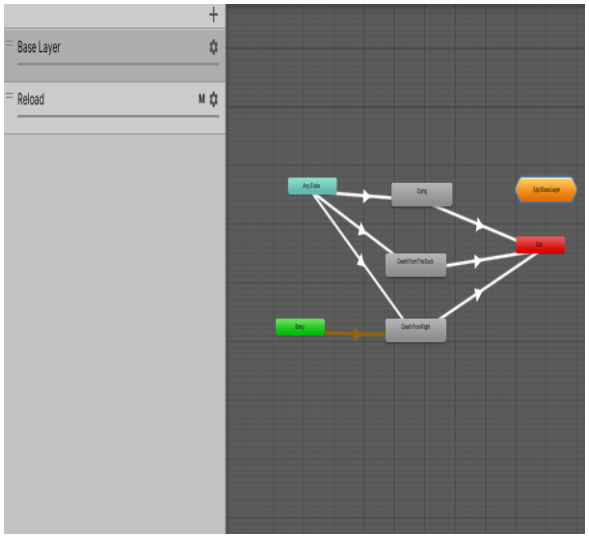
1 유니티 게임

절대강좌! 유니티2018책을 구현해보았습니다.

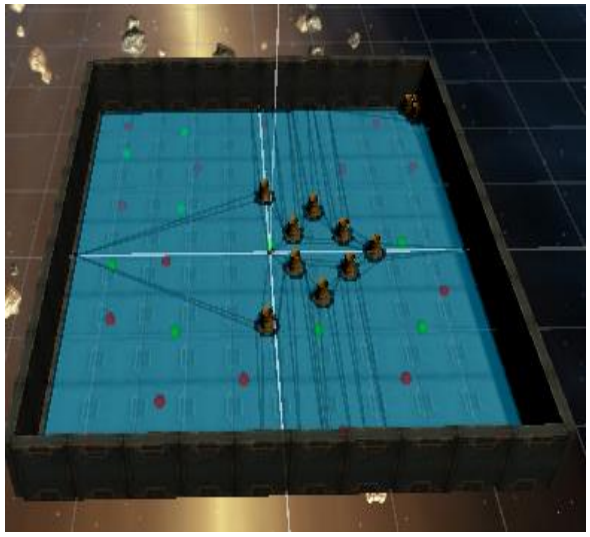
인원 : 1명
기간 : 3일(4개월차)
툴 : unity
언어 : C#
영상링크:



게임화면



메카님 애니메이션



NavMeshAgent컴포넌트,순찰지역

●절대강좌 유니티2018

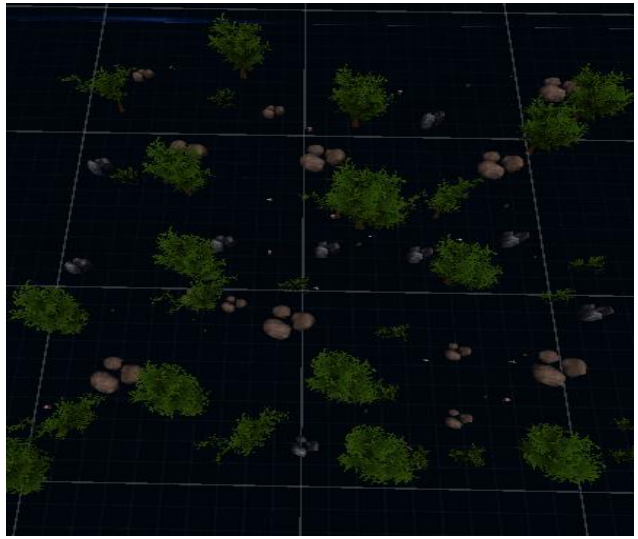
- 스카이 박스, 스테이지 제작
- RigidBody 컴포넌트를 이용한 중력
- Collider 를 통한 충돌 처리.
- AudioListener, AudioSource컴포넌트로 사운드구현
- 플레이어 움직임 레거시 애니메이션, 몬스터의 움직임을 메카님 애니메이션로 구현,
- 오브젝트 풀과 코루틴 함수로 최적화
- 카메라의 Shake효과 Fllow Cam 구현
- 애니메이션으로 구현. NavMeshAgent 컴포넌트로 AI 순찰 지점,플레이어 추적 구현
- UGUI를 이용한 체력바 등 UI를 구현.

1 유니티 게임

대학교 팀과제 MY Ground FPS 땅따먹기 게임

인원 : 4명
기간 : 2주
툴 : unity
언어 : C#
영상링크:

게임화면


맵배치

Over Scene

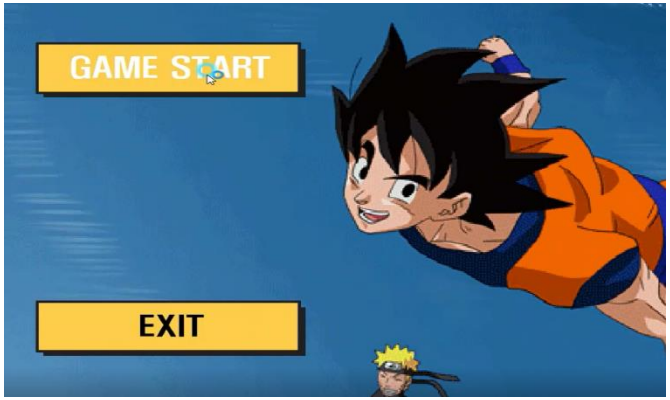
●게임 기획, 플레이어, 맵 배치, 승리종료Scene 파트 담당

- 전체적인 게임 기획, 레벨 디자인
- 맵에 오브젝트들을 Prefab으로 생성, 배치하여 플레이어와 충돌 처리
- 플레이어를 레거시 애니메이션으로 구현
- 마우스 상하좌우 회전각도 제한, 에임UI, 달리기, 걷기, 레이 캐스트를 이용하여 한번 점프 할 수 있게 예외처리
- 사운드(발소리, 총소리) 구현
- 종료 Scene에서는 버튼을 통해 scene을 전환 시켜주는 기능 구현

2 로그라이크 게임 창작

만화 캐릭터를 모으며 스테이지를 하나 씩 클리어하며
보스 스테이지까지 가는 게임

인원 : 1명
기간 : 4주(3개월차)
툴 : Windows API
언어 : C++
영상링크 : https://youtu.be/5vBXQYqe4LY



메뉴scene

```
void image::frameRender(HDC hdc, int destX, int destY, int currentFrameX, int currentFrameY)
{
    _imageInfo->currentFrameX = currentFrameX;
    _imageInfo->currentFrameY = currentFrameY;

    //트랜스 컬러 처리를 해야하나
    if (_trans)
    {
        //화면에 뿌려줄때 특정 픽셀값을 빼고 출력해준다
        GdiTransparentBlt(
            hdc,                                        //복사할 영역의 DC
            destX,                                      //복사할 좌표 X
            destY,                                      //복사할 좌표 Y
            _imageInfo->frameWidth,                     //복사할 크기
            _imageInfo->frameHeight,                    //복사할 크기
            _imageInfo->hMemDC,                         //복사할 DC
            _imageInfo->currentFrameX * _imageInfo->frameWidth,
            _imageInfo->currentFrameY * _imageInfo->frameHeight,                    //복사해올 좌표
            _imageInfo->frameWidth,                     //복사해올 크기
            _imageInfo->frameHeight,
            _transColor);                               //복사해올때 제외할 칼라

    }
    //아니나
    else
    {
        BitBlt(hdc, destX, destY,
            _imageInfo->frameWidth,
            _imageInfo->frameHeight,
            _imageInfo->hMemDC,
            _imageInfo->currentFrameX * _imageInfo->frameWidth,
            _imageInfo->currentFrameY * _imageInfo->frameHeight, SRCCOPY);
    }
}
```

이미지 FrameRender()

```
HRESULT button::init(const char * imageName, int x, int y, POINT btnDownFramePoint, POINT btnUpFramePoint, CALLBACK_FUNCTION cbFunction)
{
    _callbackFunction = static_cast<CALLBACK_FUNCTION>(cbFunction);

    _direction = BUTTONDIRECTION_NULL;

    _x = x;
    _y = y;

    _btnUpFramePoint = btnUpFramePoint;
    _btnDownFramePoint = btnDownFramePoint;

    _imageName = imageName;
    _image = IMAGEMANAGER->findImage(imageName);

    _rc = RectMake(x, y, _image->getFrameWidth(), _image->getFrameHeight());

    return S_OK;
}
```

버튼 클래스 초기화

```
void button::update()
{
    if (PtInRect(&_rc, _ptMouse))
    {
        if (KEYMANAGER->isOnceKeyDown(VK_LBUTTON))
        {
            _direction = BUTTONDIRECTION_DOWN;
        }
        else if (KEYMANAGER->isOnceKeyUp(VK_LBUTTON) && _direction == BUTTONDIRECTION_DOWN)
        {
            _direction = BUTTONDIRECTION_UP;
            _callbackFunction();
        }
    }
    else _direction = BUTTONDIRECTION_NULL;
}
```

버튼 클래스 업데이트

● 배경

Singleton으로 되어있는Image클래스에서 FrameRender()함수로 이미지를 계속 돌려주어서 배경 이미지를 영상처럼 보이게 해주었습니다.

● 버튼

재사용을 위해 클래스로 만들어서 enum으로 버튼상태를 나누었고, PtInRect함수로 누르게 되면 다른 이미지가 나오게 하였습니다. 누르고 나서 씬 전환은 **callback**함수로 구현하였고,
종료는 exit()로 구현했습니다.

2 로그라이크 게임 창작

만화 캐릭터를 모으며 스테이지를 하나 씩 클리어하며
보스 스테이지까지 가는 게임

인원 : 1명
기간 : 4주(3개월차)
툴 : Windows API
언어 : C++
영상링크 : https://youtu.be/5vBXQYqe4LY

로딩화면

스테이지

```
void loadingScene::update()
{
    _loadingBar->update();
    _loadingBar->setGauge(_currentCount, LOADINGMAX);

    if (_currentCount == LOADINGMAX)
    {
        SCENEMANAGER->changeScene("인스테이지");
    }

}
```

loading

```
//y축 루프할때 처리
for (int y = 0; y < drawAreaH; y += sourHeight)
{
    //화면에서 나간 영역만큼을 잡는다
    rcSour.top = (y + offSetY) % _imageInfo->height;
    rcSour.bottom = _imageInfo->height;

    //잘라내서 다시 화면 안으로 넣을 크기를 잡는다
    sourHeight = rcSour.bottom - rcSour.top;

    //화면밖으로 나간 영역을 다시 화면안으로 넣는 작업을 반복한다.
    if (y + sourHeight > drawAreaH)
    {
        rcSour.bottom -= (y + sourHeight) - drawAreaH;
        sourHeight = rcSour.bottom - rcSour.top;
    }

    //뿌려줄 영역에다 뿌려준다.
    rcDest.top = y + drawAreaY;
    rcDest.bottom = rcDest.top + sourHeight;
```

Looprender()

● 로딩화면

SingleThread를 사용한 singleton패턴을 기반으로 하여 이미지 및 사운드 리소스를 실제로 메모리 공간에 미리 올려 두어 실제 게임에서의 호출 속도를 였습니다.

●스테이지

배경은 image클래스의 looprender함수로 3개의 이미지를 다른 속도로 움직이게 하여 원근감을 주었습니다.

2 로그라이크 게임 창작

만화 캐릭터를 모으며 스테이지를 하나 씩 클리어하며
보스 스테이지까지 가는 게임

인원 : 1명
기간 : 4주(3개월차)
툴 : Windows API
언어 : C++
영상링크 : https://youtu.be/5vBXQYqe4LY

```
//맵에 좌표 알아오는 함수
virtual void CheakLocation();
//플레이어랑 맵 충돌함수
virtual void PlayerAndMapCollion(int groundRECT, int leftRECT, int rightRECT);
//몬스터랑 맵 충돌함수
virtual void MonsterAndMapCollion(int groundRECT);
//몬스터랑 플레이어 충돌
virtual void MonsterAndPlayerCollion();
//플레이어랑 아이템 충돌
virtual void PlayerAndItemCollion();
//땅 렉트 그려보는함수
virtual void GroundRectRender(int groundRECT);
//오른쪽 벽 렉트그려보는함수
virtual void LeftRectRender(int leftRECT);
//왼쪽 벽 렉트그려보는 함수
virtual void RightRectRender(int rightRECT);
```

스테이지 가상함수

포탈

책 상점 스테이지

상점 스테이지

●모든 스테이지

AllMap클래스의 자식 클래스이며 부모 클래스에서 만든 virtual가상 함수로 매개변수만 넣어주면 맵에 배치된 Rect들과 플레이어가 충돌하여 편하게 맵을 찍어 낼 수 있습니다.

●포탈, 책 상점

문이 닫힌 이미지에서 몬스터 Vector에 Size()가 0이 되면 문이 열리는 이미지로 바꿔주었고 스테이지 문에 플레이어가 다가가면 'F'가 나오게 해주었습니다.
책 상점은 3가지 캐릭터 책 중에 하나가 랜덤으로 하나가 나옵니다.

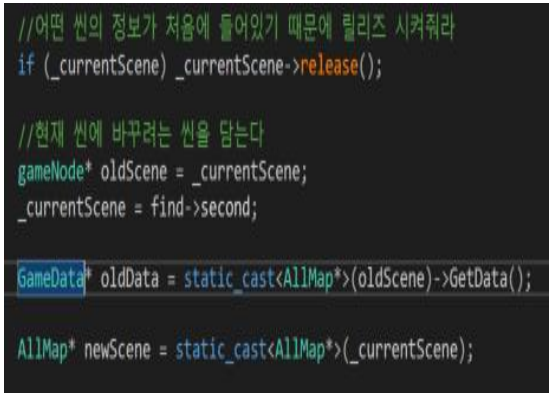
●일반 상점

8가지 아이템 중 랜덤함수로 매대에 배치되고 마우스로 갖다 대면 아이템 인덱스별로 적어 놓은 설명들을 TextOut합니다.
현재 가진 돈과 아이템가격을 비교해서 못산다면 가격이 빨간색 바탕으로 출력되게 SetBkColor함수를 사용하였습니다
플레이어가 아이템 앞에서 F를 눌렀을 때 가지고 있는 금액 조건이 만족한다면 충돌을 시켜주는 bool값이 true가 되어서 Intersectrect로 충돌시켜 아이템 인덱스를 받아온 뒤 vector에서 erase시켜주고 플레이어 DATA에 상태를 바꿔줍니다.

2 로그라이크 게임 창작

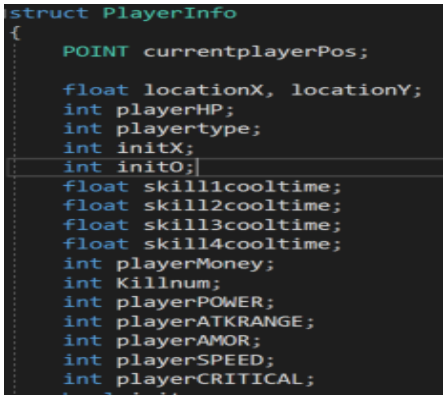
만화 캐릭터를 모으며 스테이지를 하나 씩 클리어하며
보스 스테이지까지 가는 게임

인원 : 1명
기간 : 4주(3개월차)
툴 : Windows API
언어 : C++
영상링크 : https://youtu.be/5vBXQYqe4LY



플레이어 데이터 전송

플레이어 정보UI



플레이어 정보변수

●플레이어 데이터

GameData클래스에 공격력, 방어력, 킬 수, 사거리, 크리티컬 확률,
위치, 각 스킬 쿨타임, 캐릭터 슬롯창vector 등을 가지고있습니다.
스테이지들의 부모 객체에서 선언 되어 있는 플레이어DATA를
Singleton으로 되어있는 SceneManager에서 scene이 전환 될 때 AllMap클래스에 있는 GameData를 static_cast로 캐스팅하여
데이터를 전송합니다. 'I'를 누르면 플레이어 정보창이 나오고 아이템과 상호작용하여 플레이어 정보가 변합니다.

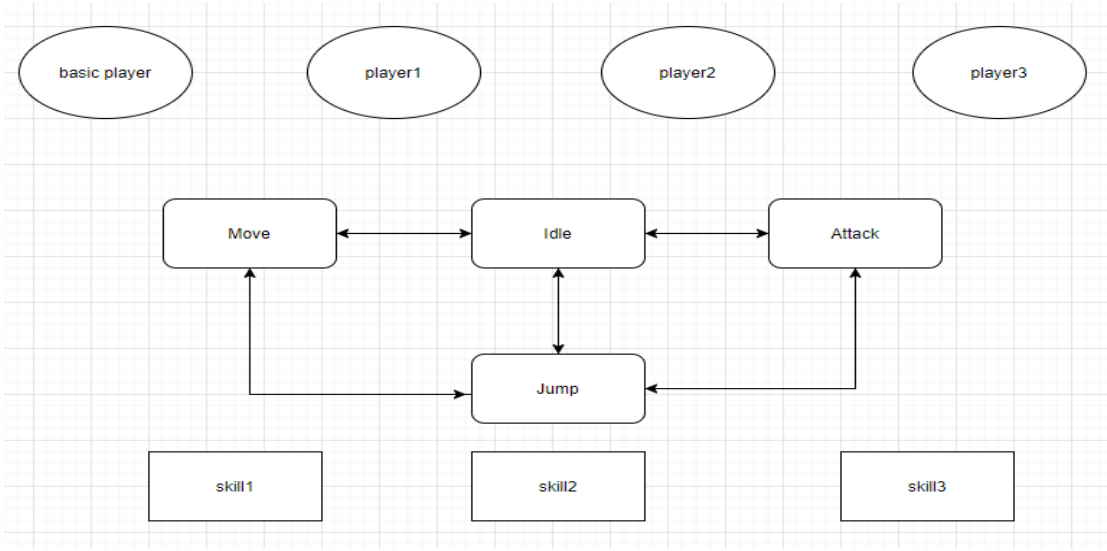
2 로그라이크 게임 창작

만화 캐릭터를 모으며 스테이지를 하나 씩 클리어하며
보스 스테이지까지 가는 게임

인원 : 1명
기간 : 4주(3개월차)
툴 : Windows API
언어 : C++
영상링크 : https://youtu.be/5vBXQYqe4LY

```
enum PlayerCharacters
{
    BASIC,
    NARUTO,
    LUFFY,
    SONOGONG

};


enum PlayerState
{
    PLAYER_LEFT_IDLE,
    PLAYER_RIGHT_IDLE,
    PLAYER_LEFT_MOVE,
    PLAYER_RIGHT_MOVE,
    PLAYER_LEFT_JUMP,
    PLAYER_RIGHT_JUMP,
    PLAYER_LEFT_ATK,
    PLAYER_RIGHT_ATK,
};
```

플레이어 상태코드



플레이어 상태표

●플레이어 상태

기본 캐릭터 1개 변신 가능 캐릭터 3개입니다. enum으로 4가지 캐릭터를 나누고 행동패턴을 enum으로 나눠서 관리합니다.
키보드 이벤트 처리로 각 상태로 변하며 상태가 끝나게 되면 Idle로 돌아갑니다.
상태는 크게 기본상태, 기본움직임, 점프, 공격, 스킬 4가지로 나눠져 있습니다.
Isground라는 bool값을 주어서 공중에 있는지 땅에 있는지를 판별하여 중력을 주었고, 점프는 2번까지 할 수 있도록 예외 처리하였습니다.
Attack상태 안에서는 위,아래 커맨드 공격 상태를 나눠주었고, 스킬 3가지는 실행 중에 다른 상태가 작동이 안되게 하기 위하여
상태를 따로 만들었습니다.

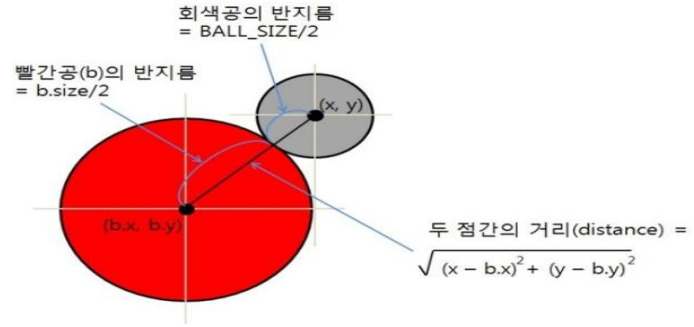
2 로그라이크 게임 창작

만화 캐릭터를 모으며 스테이지를 하나 씩 클리어하며
보스 스테이지까지 가는 게임

인원 : 1명
기간 : 4주(3개월차)
툴 : Windows API
언어 : C++
영상링크 : https://youtu.be/5vBXQYqe4LY

```
float getDistance(float startX, float startY, float endX, float endY)
{
    float x = endX - startX;
    float y = endY - startY;

    return sqrtf(x * x + y * y);
}

//두 점의 사이각을 알아오는 함수
float getAngle(float x1, float y1, float x2, float y2)
{
    float x = x2 - x1;
    float y = y2 - y1;

    float angle = -atan2f(y, x);

    return angle;
}
```

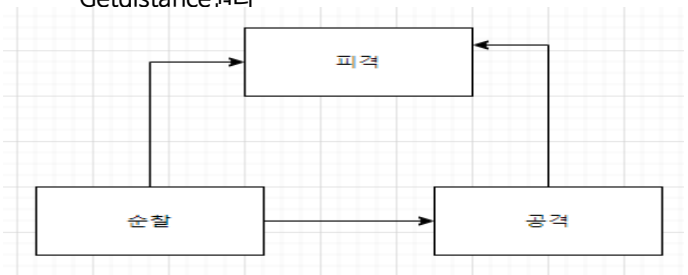
Getdistance, GetAngle 함수



Getdistance원리

```
void cameraManager::Camera_WorldDC_Shake()
{


    set_CameraXY( _cameraX + WINSIZEX / 2 + RND->getFromIntTo(-3, 3), _cameraY + WINSIZEY / 2 + RND->getFromIntTo(-3, 3));
    set_CameraXY( _cameraX + WINSIZEX / 2 + RND->getFromIntTo(-3, 3), _cameraY + WINSIZEY / 2 + RND->getFromIntTo(-3, 3));
    set_CameraXY( _cameraX + WINSIZEX / 2 + RND->getFromIntTo(-3, 3), _cameraY + WINSIZEY / 2 + RND->getFromIntTo(-3, 3));


}
```

cameraShake함수



몬스터 상태

●몬스터 패턴 설명

Getdistance함수(플레이어 추적)

거리안에 들어왔을 때 추적 기능은 두 점에 각 원을 그렸을 때 반지름으로
빗변 = 루트((x거리)제곱 + (y거리)제곱) 공식으로 거리를 구합니다.

GetAngle함수 (플레이어 추적 투사체)

플레이어를 따라가는 투사체는 아크 탄젠트 공식으로 두 점 사이의 각도를 return하여
각도로 추적하는 투사체를 만들어주었습니다.

●몬스터 상태 설명

몬스터 들은 좌우로 순찰하다가 플레이어가 범위안으로 들어오게 되면
몇 초간 공격 할 준비로 대기하게 되고 공격합니다. 플레이어에게 공격
당하면 피격 상태로 돌아가게 됩니다.

2 로그라이크 게임 창작

만화 캐릭터를 모으며 스테이지를 하나 씩 클리어하며
보스 스테이지까지 가는 게임

인원 : 1명
기간 : 4주(3개월차)
툴 : Windows API
언어 : C++
영상링크 : https://youtu.be/5vBXQYqe4LY

```
class StateBase
{
public:
    StateBase(Boss* p)
        :me(p) {}
    virtual void Enter() = 0;
    virtual void Update() = 0;
protected:
    Boss* me; // 상속한다 Monster클래스 me 값을 상속
};

class Idle : public StateBase
{
public:
    Idle(Boss* p) : StateBase(p) {}
    virtual void Enter();
    virtual void Update();
private:

};

class Skill1 : public StateBase
{
public:
    Skill1(Boss* p) : StateBase(p) {}
    virtual void Enter();
    virtual void Update();
private:

};

class Skill2 : public StateBase
{
public:
    Skill2(Boss* p) : StateBase(p) {}
    virtual void Enter();
    virtual void Update();
```

상태 클래스

```
states.insert(make_pair("기본", new Idle(this)));
states.insert(make_pair("미호크", new Skill1(this)));
states.insert(make_pair("에넬", new Skill2(this)));
states.insert(make_pair("하얀수염", new Skill3(this)));
states.insert(make_pair("우솝", new Skill4(this)));
states.insert(make_pair("스모크", new Skill5(this)));
```

Map

●보스 몬스터

friend class로 상태를 나누어 5가지 상태패턴 관리합니다. 랜덤함수로 1부터 5까지 난수를 받아서 switch문으로 랜덤으로 스킬이 나오게 해줍니다. Map을 사용하여 Key값을 string자료형으로 주어서 문자로 편하게 상태를 찾을 수 있게 하였고, 삼각함수를 이용하여 총알을 발사하는 몬스터와 플레이어 위치를 받아와서 공격을 하는 몬스터 들을 잠시 소환해줍니다.

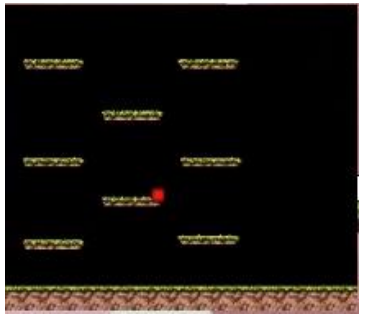
2 로그라이크 게임 창작

만화 캐릭터를 모으며 스테이지를 하나 씩 클리어하며
보스 스테이지까지 가는 게임

인원 : 1명
기간 : 4주(3개월차)
툴 : Windows API
언어 : C++
영상링크 : https://youtu.be/5vBXQYqe4LY

캐릭터, 스킬슬롯

패배scene



미니맵

돈UI

```
if (don >= 1000)
{
    int a;
    int b;
    int c;
    int d;
    a = don / 1000;
    b = (don % 1000) / 100;
    c = (don % 1000) % 100 /10;
    d = (don % 1000) % 100 % 10;
    Number[a]->render(getMemDC(), WINSIZEX - 120, heightrange);
    Number[b]->render(getMemDC(), WINSIZEX - 100, heightrange);
    Number[c]->render(getMemDC(), WINSIZEX - 80, heightrange);
    Number[d]->render(getMemDC(), WINSIZEX - 60, heightrange);
}
else if (don >= 100) {
    int a;
    int b;
    int c;
    a = don / 100;
    b = (don % 100) / 10;
    c = (don % 100)  % 10;
    Number[a]->render(getMemDC(), WINSIZEX - 100, heightrange);
    Number[b]->render(getMemDC(), WINSIZEX - 80, heightrange);
    Number[c]->render(getMemDC(), WINSIZEX - 60, heightrange);
}
```

돈UI숫자

●UI

캐릭터 슬롯

2개의 vector공간에 캐릭터 인덱스가 들어오게 되면 이미지를 나타내주고 스페이스를 누르면 바뀌게 해주었습니다.

스킬 슬롯

캐릭터의 인덱스에 따라 스킬이 바뀌며 밑에 쿨타임 게이지가 Max가 되면 스킬을 사용할 수 있게 예외처리 하였습니다.

패배scene

플레이시간은 TIMEMANAGER에서 worldtime을 가져왔고, 게임이 시작되고나서 부터 저장되는 Player의 DATA에 정보들을 가져와서 TextOut하였습니다.

미니맵

현재맵을 8/1크기로 오른쪽 하단에 잘라왔으며, 캐릭터 위치를 가져와서 미니맵과 같은 비율의 Rect로 그려주었습니다.

2 로그라이크 게임 창작

만화 캐릭터를 모으며 스테이지를 하나 씩 클리어하며
보스 스테이지까지 가는 게임

인원 : 1명
기간 : 4주(3개월차)
툴 : Windows API
언어 : C++
영상링크 : https://youtu.be/5vBXQYqe4LY

```
if (!choiceAngle)
{
    firstPosY = itemposX;
    angle = RND->getFromIntTo(85, 105);
    choiceAngle = true;
}

if (ismove)
{

    itemposX -= cosf(DO*angle) * Xspeed;
    itemposY -= sinf(DO*angle) * Yspeed;
    Yspeed -= gravity;


}
```

돈뿌려지는 코드

아이템 종류

```
//크리티컬

int rand = RND->getFromIntTo(1, 101);
cout << rand << endl;
for (int q = 0; q < DATA->playerInfo.playerCRITICAL; q++)
{
    if (q == rand)
    {
        MonsterMG->getVmonster()[i]->hitDamage(50 + DATA->playerInfo.playerPOWER);
        EFFECTMANAGER->play("치명타", RND->getFromIntTo(monsterRC.left, monsterRC.right)
            , RND->getFromIntTo(monsterRC.top, monsterRC.bottom));
    }

}
```

크리티컬

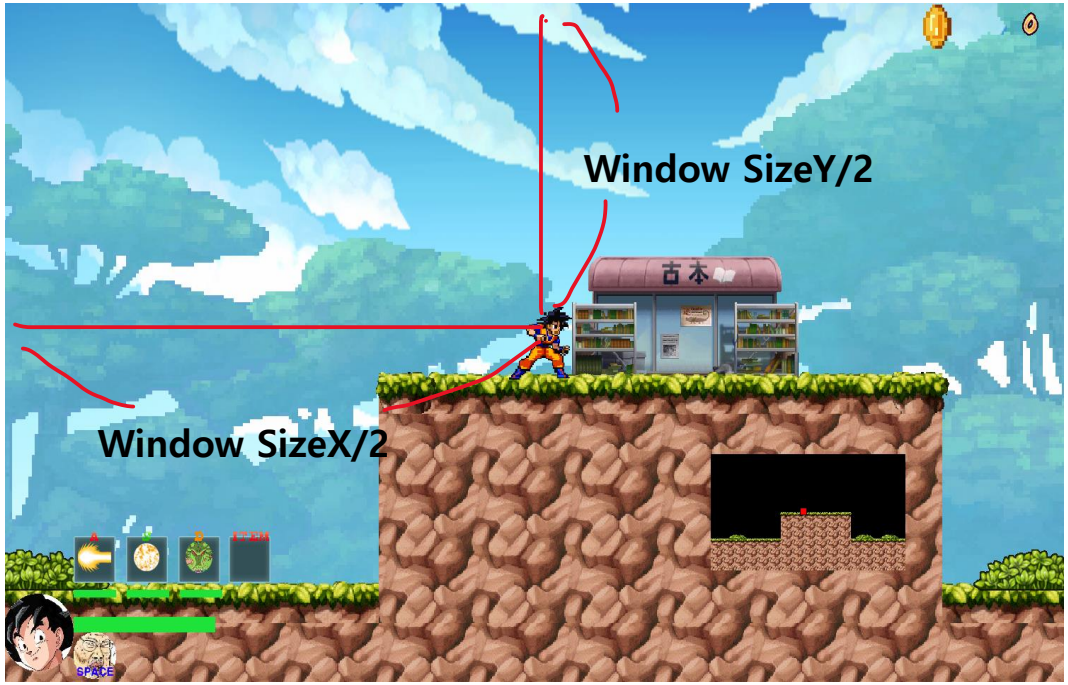
●ITEM

각 장비 아이템은 상점에서 랜덤으로 나와서 살수 있고, 공격력, 사거리, 방어력, 이동속도, 치명타를 올려줍니다.
몬스터가 드랍 하는 아이템은 포션과 동전이 있는데 포션은 랜덤하게 나오게 하였고,
동전은 삼각함수를 이용하여 각도를 랜덤으로 주어서 사방으로 뿌려지는 효과를 주었습니다.
사거리는 공격 할 때의 충돌체 RECT의 크기를 좀 더 늘려 주었습니다
크리티컬은 1에서 100까지 난수를 주어서 for문으로 확률을 만들고 다른 이펙트가 출력 되게 하였습니다.

2 로그라이크 게임 창작

만화 캐릭터를 모으며 스테이지를 하나 씩 클리어하며
보스 스테이지까지 가는 게임

인원 : 1명
기간 : 4주(3개월차)
툴 : Windows API
언어 : C++
영상링크 : https://youtu.be/5vBXQYqe4LY



카메라 움직임

```
void cameraManager::Camera_Correction()
{
    if (_cameraX < 0) _cameraX = 0;
    if (_cameraY < 0) _cameraY = 0;
    if (_cameraX + _cameraSizeX > _cameraWorldSizeX)
    {
        _cameraX = _cameraWorldSizeX - _cameraSizeX;
    }
    if (_cameraY + _cameraSizeY > _cameraWorldSizeY)
    {
        _cameraY = _cameraWorldSizeY - _cameraSizeY;
    }
}
```

카메라 고정

●카메라

WorldDC에 그려지는 이미지를 플레이어 좌표에서 WINDOW SIZE X축과 Y축의 2/1만큼 씩 뺀 값을 카메라의 기준점으로 잡고 가져왔습니다.
플레이어가 맵에 끝 부분에 가게 되면 카메라가 움직이지 않게 고정 해주는 예외 처리를 하였습니다.
땅을 흔드는 몬스터가 있을 때 카메라를 랜덤함수로 흔들어 주었고, 최적화를 위해 화면에 보이는 부분만 이미지를 그려주었습니다. 화면상 보이지 않을 때 움직일 필요가 없는 몬스터 들은 프레임 드랍을 방지하기위해 Update를 막아주는 클리핑 예외 처리를 하였습니다.

3 로그라이크 게임 모작(팀)

내크로 댄서 모작게임(몬스터 담당)

인원 : 5명
기간 : 2주(3개월차)
툴 : DirectX2D
언어 : C++
영상링크 : https://youtu.be/5vBXQYqe4LY

내크로댄서

```
states.insert(make_pair("Idle", new MonsterStateIdle(this)));
states.insert(make_pair("Move", new MonsterStateOneStep(this)));
states.insert(make_pair("Atk", new MonsterStateAtk(this)));
```
map

```
friend class MonsterStateBase;
friend class MonsterStateOneStep;
friend class MonsterStateAtk;
friend class MonsterStateIdle;
```
상태클래스

```
float Math::Lerp(float val1, float val2, float amount)
{
    return val1 + (val2 - val1)* amount;
}
```
Lerp함수

●몬스터 파트 담당

몬스터의 key값을 받아서 상태패턴으로 움직임을 주었습니다. 다음에 갈 타일의 인덱스를 가져와서 속성을 확인하고
못 가는 곳이면 못 가도록 예외 처리하였고 몬스터의 움직임을 부드럽게 하기위해 선형 보간 법을 써서 정해진 시간안에 거리를 가게 만들었습니다.
박자 노드에 맞춰서 움직이므로 박자순간마다 한번 돌아가는 Upeate에서 Map에 등록된 상태클래스에 key값으로 찾아갑니다.
플레이어와의 상호 작용은 메시지 패턴을 이용하였습니다.

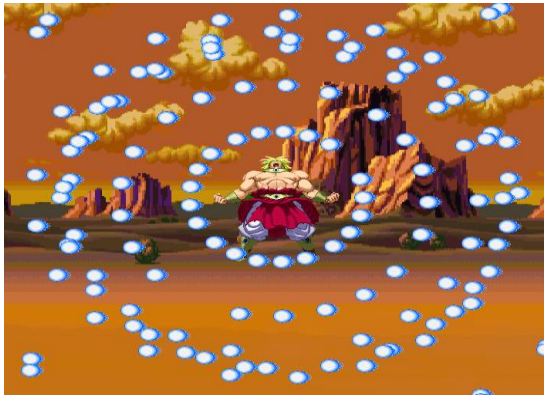
4 슈팅 게임 창작

몬스터를 잡으며 보스 스테이지로 가는 슈팅게임

인원 : 1명
기간 : 1주(2개월차)
툴 : Windows API
언어 : C++
영상링크 : https://youtu.be/aXzjcTjWz_E


alphaFrameRender()


몬스터움직임


보스 총알

●알파프레임

플레이가 몬스터 총알에 맞으면 alphaFrameRender() 에 값을 조절하여 FadeIn FadeOut을 반복하며 몇 초간 무적 상태가 됩니다.

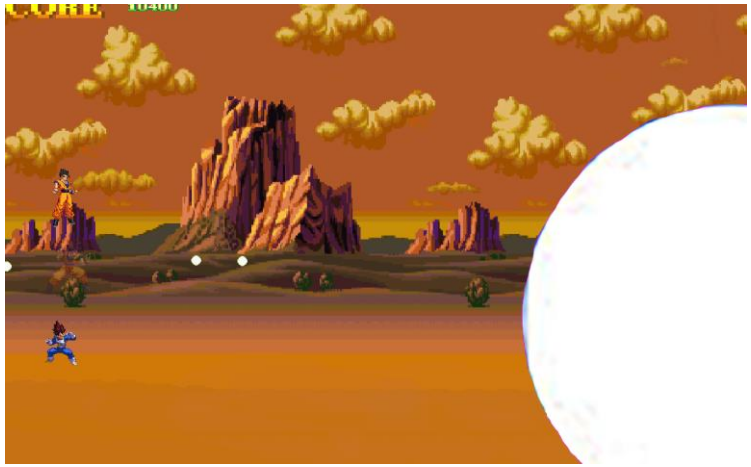
●몬스터 움직임, 총알

몬스터에 자연스러운 곡선 움직임을 주었고, 몬스터 총알도 360도로 쏘는 기능을 삼각함수로 구현하였습니다. 보스 패턴은 hp에 따라서 3가지 패턴으로 분리하였습니다.

4 슈팅 게임 창작

몬스터를 잡으며 보스 스테이지로 가는 슈팅게임

인원 : 1명
기간 : 1주(2개월차)
툴 : Windows API
언어 : C++
영상링크 : https://youtu.be/aXzjcTjWz_E



```
for (int i = 0; i < _bullet3->getVBullet().size(); i++)
{
    RECT temp;
    RECT monsterbullet = _bullet3->getVBullet()[i]->rc;
    if (IntersectRect(&temp, &monsterbullet, &_player->getRect()))
    {
        if (_player->getState() == 5) {
            player_guardCount++;
        }
        if (player_guardCount > 2) {
            _player->set_IsDead(true);
        }
```

●플레이어

원기옥은 원의 깔끔한 충돌을 위해 두 객체의 원점사이의 거리를 빗변을 구해서 충돌 시켜주었고 유체화는 alphaFrameRender()를 사용하여 투명화 시켜주며 총알과 충돌을 잠시 막아 주었습니다. 막기기능으로 count변수를 주어서 적 총알을 2회 이상 막을 수 있게 하였고, bool값을 주어서 아이템을 먹으면 친구1, 친구2 객체를 생성 시켜주었습니다

●충돌처리

두개의 rect를 if문을 여러 개 만들어서 처리 할 수 있지만, Intersectrect함수를 사용하여는 것이 더 간결하다고 생각하여 사용하였습니다. 충돌하게 되면 각 클래스 간에 Getter Setter로 HP를 감소 시켜줍니다.

감사합니다.