2022년도 졸업작품 중간발표

나무의 집

지도교수
이OO 교수님

2017180001 고선민
201718XXXX 우OO
201918XXXX 최OO

목차

1. 개요

01. 게임 소개

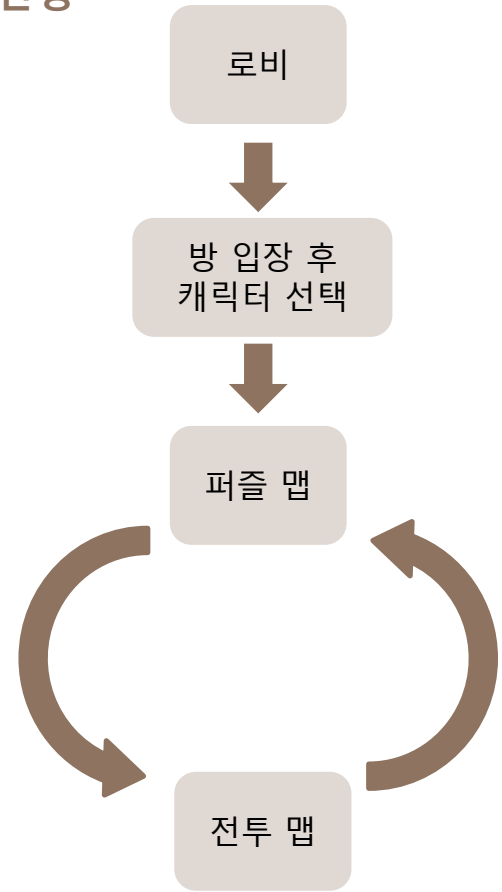
2인용 VR 퍼즐 어드벤쳐 게임

두 플레이어의 서로 다른 무기는
서로의 무기 혹은 맵 오브젝트와 상호작용

이를 이용해 퍼즐을 풀고 몬스터를
처치하며 나아가는 게임

1. 개요

02. 게임 진행



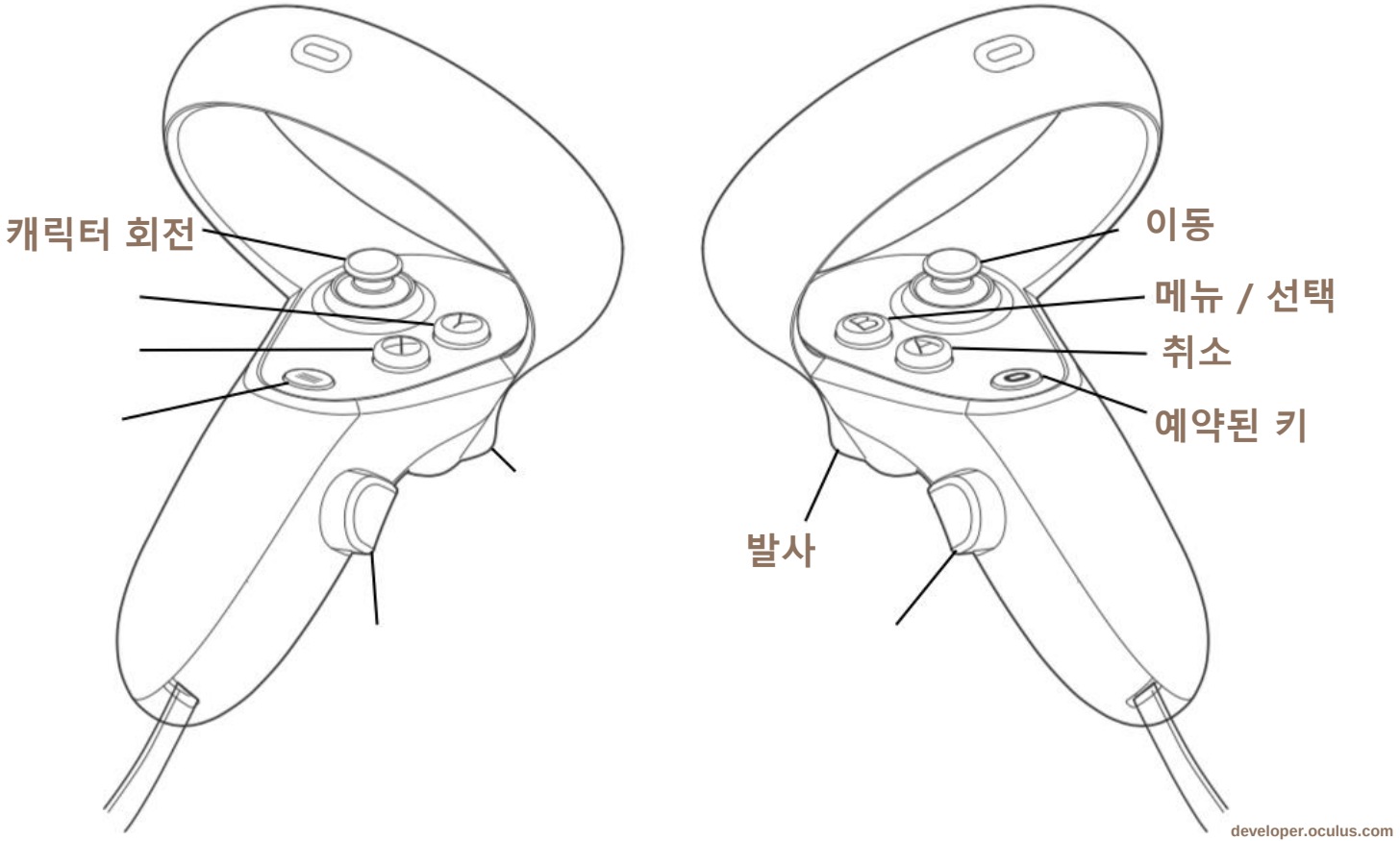
퍼즐 맵

두 무기를 활용하여
길을 막고있는 퍼즐을 차례대로
해결해가며 전진

전투 맵

두 무기를 활용하여
몬스터를 처치

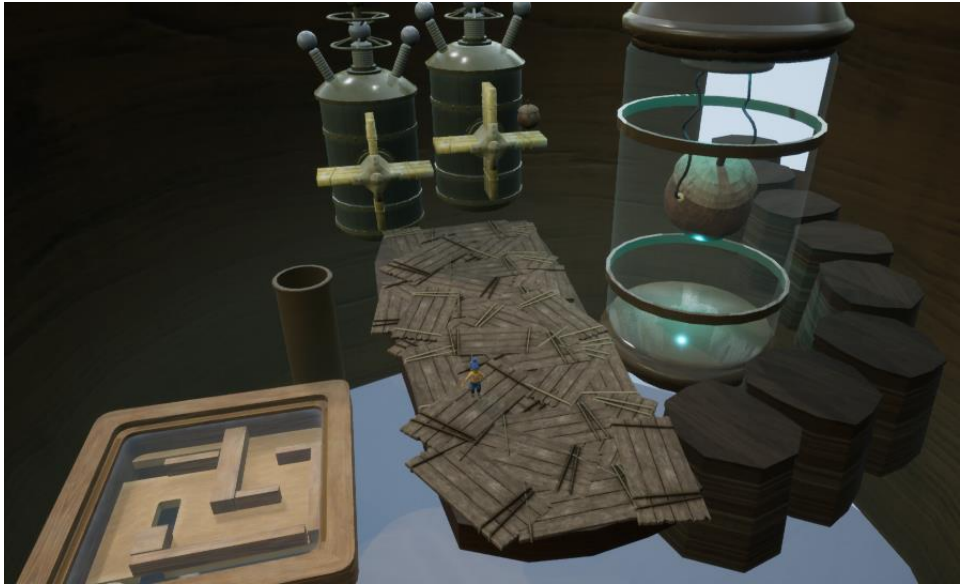
2. 게임 조작



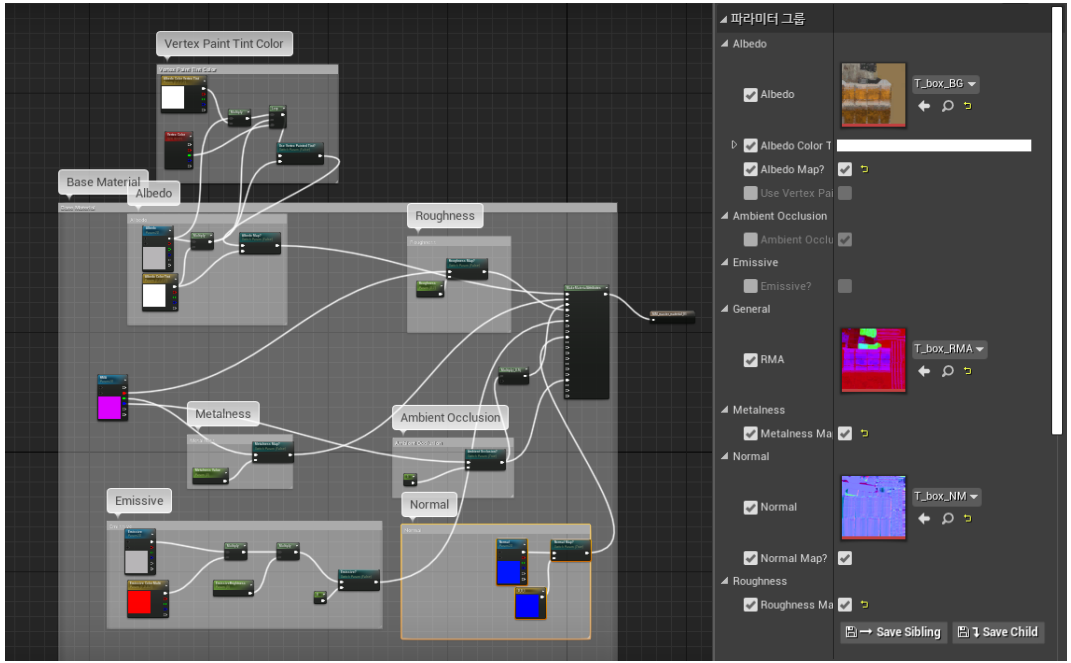
3.

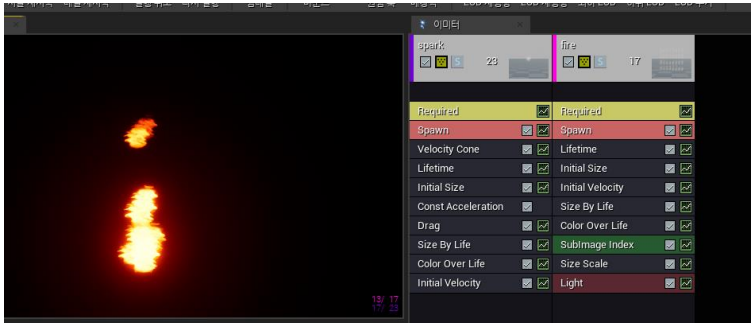
개발 내용

그래픽 리소스 – 맵



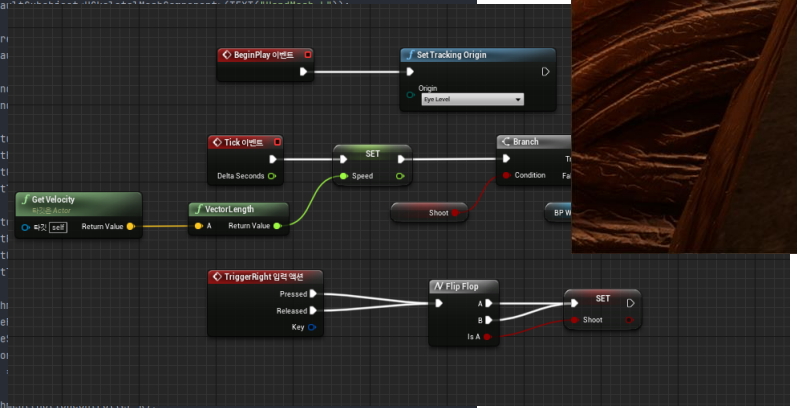
3. 개발 내용
그래픽 리소스 – 머테리얼





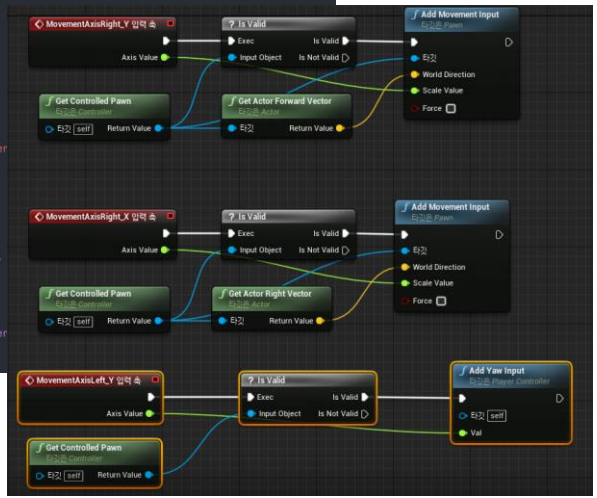
3. 개발 내용
클라이언트 – VR 캐릭터

```
// Set this character to call Tick() every frame.  You can turn this off to improve performance if yo
PrimaryActorTick.bCanEverTick = true;

static ConstructorHelpers::FObjectFinder<USkeletalMesh> SK_Hand(
    TEXT("SkeletalMesh'/Game/Resource/Actor/Hand/MannequinHand_Right.MannequinHand_Right'"));

GetMesh()->SetOwnerNoSee(true);
GetMesh()->SetRelativeLocationAndRotation(FVector(-15, 0, -88), FRotator(0, -90 ,0));
GetMesh()->SetRelativeScale3D(FVector(1.7, 1.7, 1.7));

VR_Camera = CreateDefaultSubobject<UCameraComponent>(TEXT("Camera"));
VR_Camera->SetupAttachment(RootComponent);
VR_Camera->SetRelativeLocation(FVector(0, 0, 80));

MotionController_L = CreateDefaultSubobject<UMotionControllerComponent>(TEXT("MotionContoller_L"));
HandMesh_L = CreateDefa

MotionController_R = Cr
HandMesh_R = CreateDefa

Hand_L = EControllerHan
Hand_R = EControllerHan

MotionController_L->Set
MotionController_L->Set
MotionController_L->Set
MotionController_L->Set

MotionController_R->Set
MotionController_R->Set
MotionController_R->Set
MotionController_R->Set

HandMesh_L->SetupAttach
HandMesh_L->SetRelativeL
HandMesh_L->SetRelative
HandMesh_L->SetCollisio
HandMesh_L->CastShadow

HandMesh_R->SetupAttach
HandMesh_R->SetRelativeRotation(FRotator(-30, 0, 90));
HandMesh_R->SetCollisionProfileName(TEXT("NoCollision"));
HandMesh_R->CastShadow = false;
```

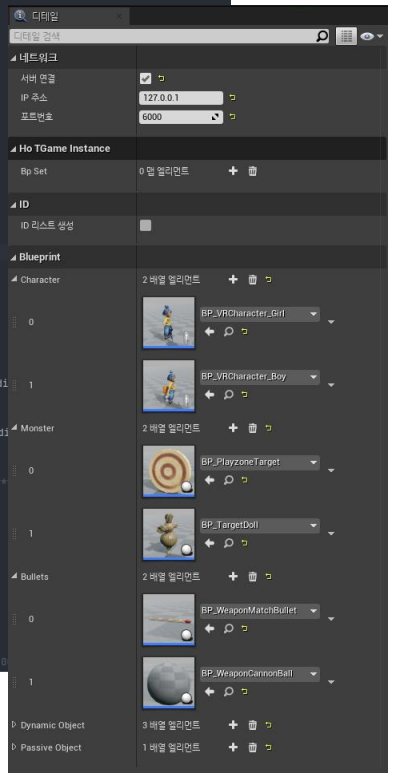


3. 개발 내용
클라이언트 – 블루프린트

```
UCLASS()
1 derived blueprint class
class HOUSE_OF_TREE_API AVRPlayerController_Base : public APlayerController
{
    GENERATED_BODY()
public:
    AVRPlayerController_Base();

protected:
    UPROPERTY()
    UHoTGameInstance* gameInst;

    UPROPERTY(EditDefaultsOnly, Category = "플레이어 캐릭터", DisplayName="캐릭터")
    TArray<TSubclassOf<APawn>> Characters;

    UPROPERTY(BlueprintReadOnly)
    int playerID;

    UPROPERTY()
    AVRCharacter_Base* vrPlayer;

    UPROPERTY(BlueprintReadOnly)
    TMap<int, AActor*> actorList;

    void SetPlayerCharacter();

    // Network system
public:
    Concurrency::concurrent_queue<char> buffer
    std::atomic<int> bufferSize {_Value 0};

protected:
    void ProcessPacket();
    void SendPlayerData();

    void PutObject(int actorID, int objectID,

public:
    virtual void BeginPlay() override;

    virtual void Tick(float DeltaSeconds) over
};
```




```
class AVRPlayerController_Base;

UCLASS()
1 derived blueprint class
class HOUSE_OF_TREE_API UHoTGameInstance : public UGameInstance
{
    GENERATED_BODY()

public:
    UHoTGameInstance();
    virtual ~UHoTGameInstance() override;

    virtual void Init() override;

    void InitSocket();

    void SetInfo();
    UClass* GetActor(int ObjectID);

    ClientSocket* SocketInstance;

private:
    UPROPERTY(EditDefaultsOnly, Category=Network, DisplayName="서버 연결")
    bool ConnectNetwork;  "True" (Default_BP_GameInstance_C)

public:
    UPROPERTY(EditDefaultsOnly, Category=Network, DisplayName="IP 주소", meta = (Edi
    FString ipAddr;  "127.0.0.1 " (Default_BP_GameInstance_C)

    UPROPERTY(EditDefaultsOnly, Category=Network, DisplayName="포트번호", meta = (Edi
    int Port;  "6000" (Default_BP_GameInstance_C)

    //*****************************************************************************

private:
    UPROPERTY(EditAnywhere)
    TMap<int, UClass*> bpSet;  Unchanged in assets

    // ID 리스트 파일을 새로 생성하라고 표시하는 명령어
    UPROPERTY(EditDefaultsOnly, Category="ID", DisplayName="ID 리스트 생성")
    bool makeIDList;  Unchanged in assets

    // 사용하는 블루프린트 액터를 기록하는 컨테이너
    UPROPERTY(EditDefaultsOnly, Category="Blueprint", DisplayName="Character") // 0
```



3. 개발 내용
클라이언트 – 네트워크 통신

```
bool ClientSocket::ConnectServer()
{
    WSADATA wsaData;
    int nRet = WSAStartup(MAKEWORD(2, 2), &wsaData);
    if (nRet != 0)
    {
        return false;
    }

    Socket = WSASocket( af: AF_INET,  type: SOCK_STREAM,  protocol: 0,  lpProtocolInfo
    if (Socket == INVALID_SOCKET)
    {
        return false;
    }

    SOCKADDR_IN stServerAddr;

    stServerAddr.sin_family = AF_INET;
    stServerAddr.sin_port = htons(gameInst->Port);

    stServerAddr.sin_addr.s_addr = inet_addr(TCHAR_TO_ANSI(ToCStr(gameI

    nRet = connect(Socket, (sockaddr*)&stServerAddr,  namelen: sizeof(sock

    GEngine->AddOnScreenDebugMessage( Key: -1,  TimeToDisplay: 5.f, FColor::Re

    if (nRet == SOCKET_ERROR)
    {
        GEngine->AddOnScreenDebugMessage( Key: -1,  TimeToDisplay: 5.f, FColor
        return false;
    }
```

```
LogTemp: Warning: 10
LogTemp: Warning: putobject BP_VRCharacter_Girl_C
LogTemp: Error: Actor ID: 0, ObjectID: 100001
LogTemp: Warning: 10
LogTemp: Warning: putobject BP_VRCharacter_Boy_C
LogTemp: Error: Actor ID: 1, ObjectID: 100002
LogTemp: Warning: 10
LogTemp: Warning: putobject BP_Door_C
LogTemp: Error: Actor ID: 10, ObjectID: 500001
LogTemp: Error: Send Packet
LogTemp: Warning: Packet SIZE 78
LogTemp: Warning: 13
LogTemp: Warning: SC_PLAYER_DATA 1
LogTemp: Error: Send Packet
LogTemp: Warning: Packet SIZE 78
LogTemp: Warning: 13
LogTemp: Warning: SC_PLAYER_DATA 1
LogTemp: Error: Send Packet
LogTemp: Warning: Packet SIZE 78
LogTemp: Warning: 13
LogTemp: Warning: SC_PLAYER_DATA 1
LogTemp: Error: Send Packet
LogTemp: Warning: Packet SIZE 78
LogTemp: Warning: 13
LogTemp: Warning: SC_PLAYER_DATA 1
LogTemp: Error: Send Packet
LogTemp: Warning: Packet SIZE 78
LogTemp: Warning: 13
LogTemp: Warning: SC_PLAYER_DATA 1
LogTemp: Error: Send Packet
```

3. 개발 내용
서버 – IOCP MO 서버 제작

```
void cIOCPBase::StartServer() { ... }
void cIOCPBase::CloseServer() { ... }
void cIOCPBase::BindAndListen(USHORT server_port) { ... }
void cIOCPBase::Disconnect() { ... }
void cGameServer::Init() { ... }
void cGameServer::Start() { ... }
void cGameServer::TimerThread() { ... }
void cGameServer::WorkerThread() { ... }
void cGameServer::Accept(cExpOver* exp_over) { ... }
void cGameServer::Disconnect(const unsigned int _user_id) { ... }
void cGameServer::Send(cExpOver* exp_over) { ... }
void cGameServer::Recv(cExpOver* exp_over, const unsigned int _user_id, const DWORD num_byte) { ... }
void cGameServer::ProcessPacket(const unsigned int _user_id, unsigned char* _p) { ... }
```

```
template<typename T>
class cObjectPool
{
public:
    cObjectPool(int _max_size) { ... }
    ~cObjectPool() { ... }

    T* PopObject() { ... }
    void PushObject(T* _object) { ... }

    void ExpandPool(int _size) { ... }

private:
    Concurrency::concurrent_queue<T*> m_objects;
    std::atomic<int>                  m_max_size;
};
```

4. 기술 요소와 중점 연구 분야

VR

Oculus VR 및 Oculus Touch를 이용한
VR 게임 플레이

IK를 활용한 VR 플레이어 캐릭터 구현

IOCP

IOCP 서버 구현

2인 세션 MO서버를
멀티스레드 IOCP로 구현

4. 기술 요소와 중점 연구 분야

Seamless Map

로딩 화면 없는 자연스러운 맵 이동

비동기 레벨 로딩을 구현
자연스러운 전환을 위해 배치 및 연구 중

Meta Ball

남자 캐릭터가 발사하는 수액을
Meta Ball로 구현

전체적인 개발 일정 지연으로 인해
진행 X

5. 문제점 및 보완책

어색한 캐릭터 애니메이션

IK 활용 능력 부족으로 인한
어색한 애니메이션.

더 나은 애니메이션을 위한 연구

맵 디테일

PC와는 다른 VR 플레이 환경

VR 플레이 환경에 맞춰
게임 디테일 향상

6. 개발 일정 및 역할분담

01. 기존 개발 일정

Index			
고선민	우OO	최OO	ALL

구분	항목	12	1	2	3	4	5	6	7	8
그래픽	캐릭터 모델링+애니									
	배경 모델링									
	이펙트									
	UI									
클라이언트	플레이어 시스템									
	몬스터 시스템									
	퍼즐과 맵 제작									
	서버 연결 및 테스트									
서버	기본 구조 설계									
	게임 로직 제작									
	동기화 확인									
	다중 세션 및 최적화									
QA	버그 테스트									

6. 개발 일정 및 역할분담

02. 변경 개발 일정

Index			
고선민	우OO	최OO	ALL

구분	항목	12	1	2	3	4	5	6	7	8
그래픽	캐릭터 모델링+애니									
	배경 모델링									
	이펙트									
	UI									
클라이언트	플레이어 시스템									
	몬스터 시스템									
	퍼즐과 맵 제작									
	서버 연결 및 테스트									
서버	기본 구조 설계									
	게임 로직 제작									
	동기화 확인									
	다중 세션 및 최적화									
QA	버그 테스트									

완료 / 진행중 / 지연

7. 향후 개발 일정

1. IK 개선
2. 레벨 추가 구현
3. 로비, UI 추가
4. 맵 라이팅, 머테리얼 개선
5. 통신 및 서버 연산 최적화를 통한 동접자 상승 도모

8. 데모 시연

Q & A