2016180037 임건호
2016182019 성기홍
2015182003 권호민

지도교수 : 정내훈

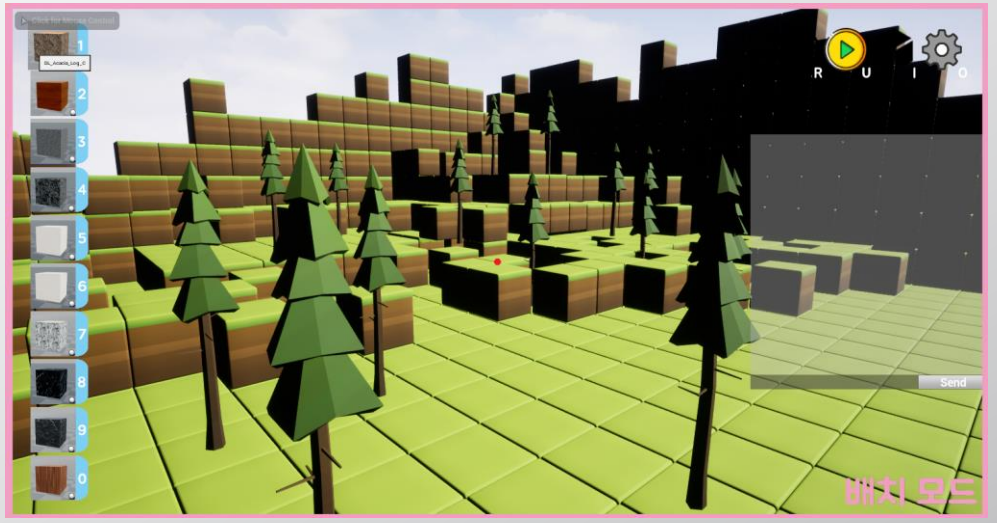
시간의 마녀
〈 Witch of Time 〉

중간 발표 자료

2021년도 졸업작품 중간 발표 자료

INDEX

1. 게임 개요



장르 : 샌드박스 기능을 가진 3D 플랫포머
(멀티플레이어 어드벤처 + 마인크래프트)

특징 : 언리얼 4, 클라이언트/서버

게임플레이 :

* 싱글 플레이의 경우
  - 멀티 플레이의 가이드 라인 느낌.
  - 목적 달성을 방해하는 함정과 몬스터.
  - 시간 마법을 활용한 함정 해결.
  - 3인칭 시점의 게임 플레이.

* 멀티 플레이의 경우
  - 여러 플레이어가 동시에 맵 제작.
  - 원하는 블록을 배치.
  - 원하는 블록의 커맨드 구현.
  - 만든 맵에서 게임 플레이.

2. 게임 방법

플레이 모드

- WASD : 캐릭터 이동
- 마우스 드래그 : 캐릭터 시선 이동
- Spacebar : 점프
- F : 상호작용
- 마우스 우 클릭 : 타겟 지정
- Q : 시간 감속
- E : 시간 가속
- M : 타이틀 화면

메이커 모드

- WASD : 카메라 이동
- Ctrl : 카메라 하강
- Spacebar : 카메라 상승
- R : 커맨드 모드와 배치 모드 전환
- U : 테스트 모드와 배치 모드 전환
- I : 블록 인벤토리
- O : 옵션 창
- M : 타이틀 화면

3. 기술적 요소 및 중점 연구 분야

시간 마법

시간 마법을 사용할 때 변화되는 모습을
부드럽게 연결되도록 구현

실시간 데이터 공유

같은 공간에서 여러 플레이어가 동시에 맵을
제작할 수 있도록 실시간 맵 동기화.

언리얼과 서버와의 연동

커스텀 서버를 통한 멀티플레이
서버에서 모든 게임 컨텐츠 실행

맵 에디터 제작

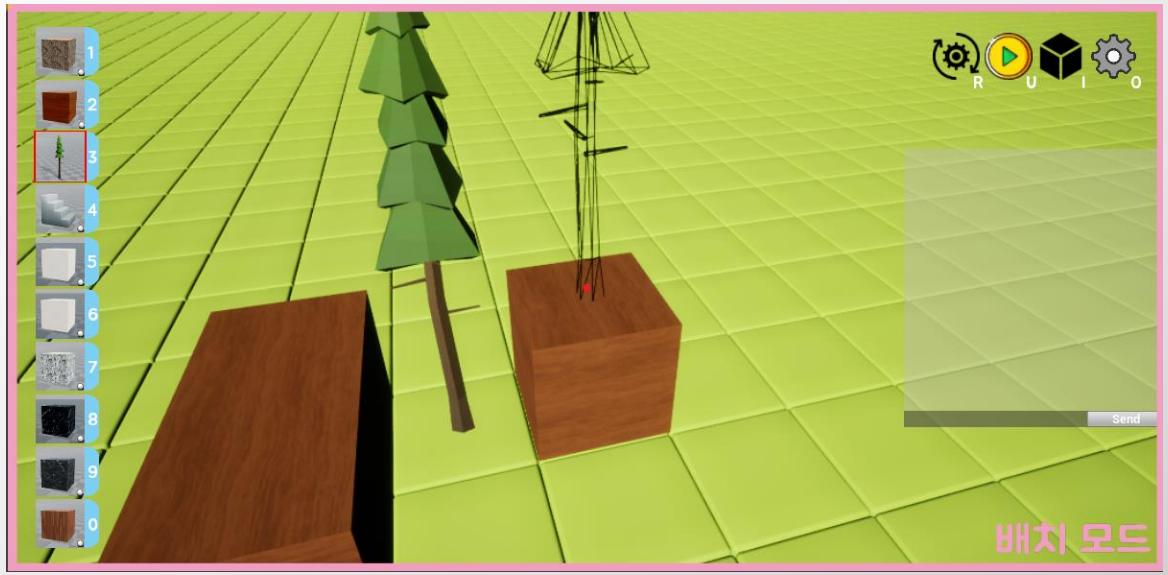
플레이어 친화적인 UI 제공으로 쉽고 간단하게
게임상에서 맵 제작이 가능하도록 구현

4. 구성원의 역할 분담

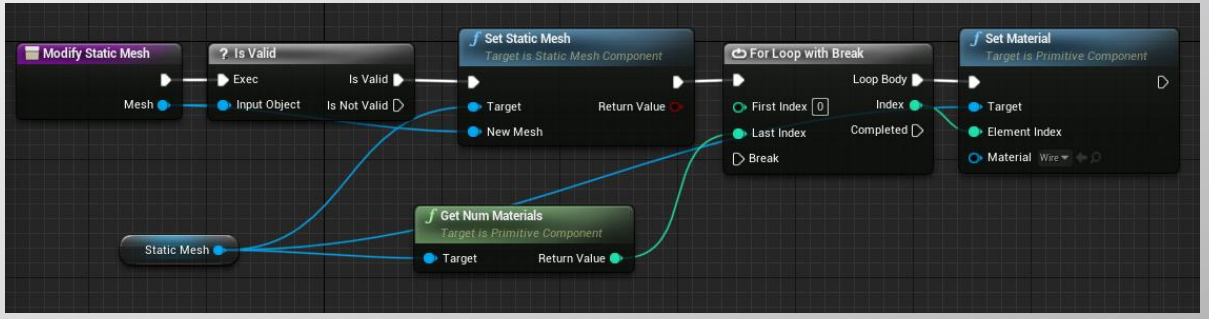
임건호	성기홍	권호민
메인 기획 SUB 클라이언트 프로그래밍 그래픽 리소스 확보 및 제작 - 전반적인 일정 조율 - 리소스 확보 및 제작 - 캐릭터 제작 및 이동 구현 - 블록 기획 및 구현 - UI 기획 및 제작 - 타이틀 화면 제작 - 게임 맵 제작 - 기획 문서 및 발표 문서 작성	메인 클라이언트 프로그래밍 - 핵심 게임 로직 제작 - 맵 에디터 핵심 기능 제작 - 블록 및 캐릭터 클래스 기본 설계 - 효율적인 작업을 위한 시스템 설계 - 특수 블록 기능 구현 - 블록 배치 및 삭제 구현 - 커맨드 블록 기능 및 적용 구현 - 인벤토리와 커맨드 블록 UI 등의 기능 구현 - 맵 저장 및 불러오기 구현	서버 프로그래밍 - 기본 서버 구조 작성 - 채팅 기능 및 UI 제작 - 서버를 통해 블록 배치 및 삭제 공유 - 플레이어 정보 공유 - 서버를 통해 각 클라이언트 간 블록 배치 기능을 이용한 맵 로드

5. 개발 내용

블록 배치 / 삭제



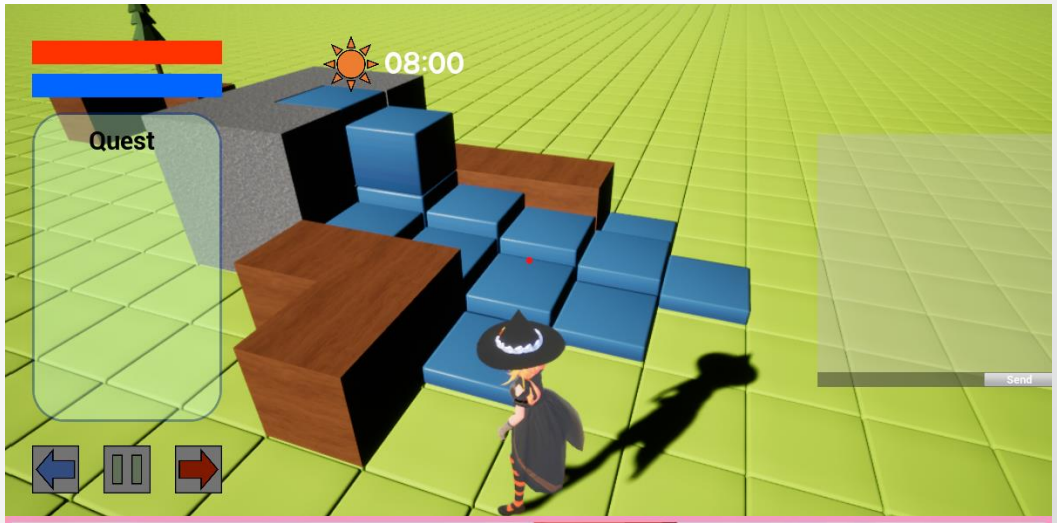

```
if (GetWorld()->LineTraceSingleByChannel(hitResult, CLocation, CLocation + CForwardVector * 2000, ECC_Camera, collisionParams))
{
    FActorSpawnParameters SpawnParams;
    SpawnParams.SpawnCollisionHandlingOverride = ESpawnActorCollisionHandlingMethod::AlwaysSpawn;
    FVector Location = hitResult.GetComponent()->GetComponentLocation();

    Location -= (FVector)hitResult.Location;

    SpawnParams.NameMode = FActorSpawnParameters::ESpawnActorNameMode::Requested;
    // 충돌 위치에 따른 블럭 배치 위치 조정
    { ... }
    auto spawned = GetWorld()->SpawnActor<AActor>(PlaceActor, (FVector)hitResult.Location, Rotator, SpawnParams);
```

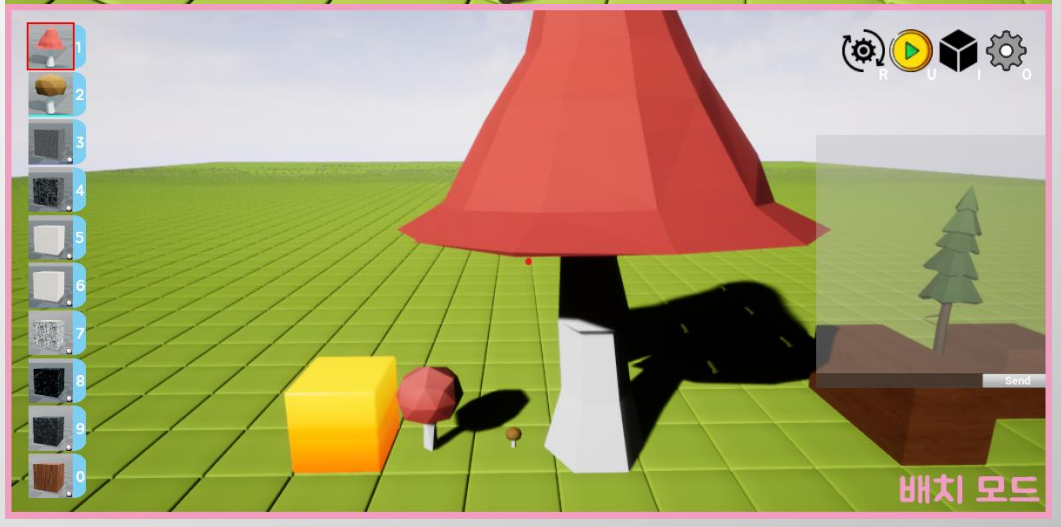


- LineTrace를 사용하여 블록이 배치될 위치 파악
- 파악 후 실시간으로 배치될 블록의 와이어 메테리얼 표시
- 배치 명령을 통하여 와이어 메테리얼을 일반 블록으로 변환
- 삭제 명령을 통하여 일반 블록 제거

5. 개발 내용

특수 블록 구현






```
FVector Location = this->GetActorLocation();
auto flow = PlaceDummy(Location - FVector(0, 0, 199), true);

if (!flow)
{
    if (m_height > 0.3f)
    {
        PlaceDummy(Location - FVector(200, 0, -1), false);
        PlaceDummy(Location + FVector(200, 0, 1), false);
        PlaceDummy(Location - FVector(0, 200, -1), false);
        PlaceDummy(Location + FVector(0, 200, 1), false);
    }
}
```

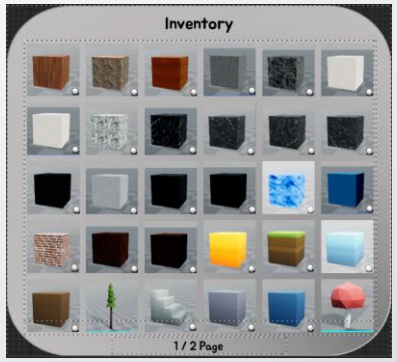
```
auto PlacedActor = GetWorld()->SpawnActor<AActor>(Place, location, Rotator, SpawnParams);
auto casted = Cast<ALiquidBlockBase>(PlacedActor);
if (IsFlow == false)
{
    if (casted != NULL)
    {
        casted->SetHeight(m_height - 0.2f,false);
        casted->SetParant(this);
        casted->SetGrandParant(GrandParentBlock);
        ChildBlocks.Add(casted);
        spawncount++;
    }
    else
    {
        UKismetSystemLibrary::SphereOverlapActors(GetWorld(), location, 100.f, Types, NULL, Ignores, Actors);
        for (AActor* Actor : Actors)
        {
            casted = Cast<ALiquidBlockBase>(Actor);
            if (casted != NULL)
            {
                if (casted->m_height < m_height - 0.2f)
                {
                    casted->SetHeight(m_height - 0.2f, false);
                    casted->SetParant(this);
                    casted->SetGrandParant(GrandParentBlock);
                    ChildBlocks.Add(casted);
                    spawncount++;
                }
```

- 물 블록 사방에 부모 블록보다 낮은 자식 블록을 복사하여 배치
- 이를 통하여 물이 흘러내리는 유체를 표현
- 부모 블록을 제거하면 자식 블록은 일정 시간이 지난 뒤 삭제

- 블록은 온도 정보를 가지고 있음
- 이를 통하여 특정 블록들의 상 변환을 구현
- 또한, 시간 마법을 통하여 메쉬의 자연스러운 변환 블록 구현

5. 개발 내용

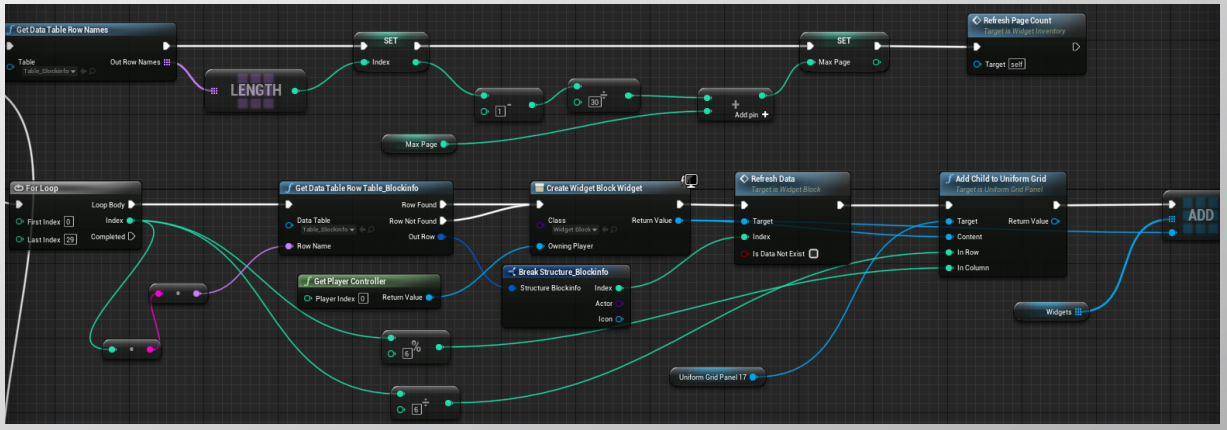
작업 효율화



Data Table | Data Table Details

Search

	Row Name	Index	Actor	Icon
1	0	0	BlueprintGeneratedClass'/Game/BL_Blocks/BL_Acacia.BL_Acacia_C'	Texture2D'/Game/Icon/IC_Acacia.IC_Acacia'
2	1	1	BlueprintGeneratedClass'/Game/BL_Blocks/BL_Acacia_Log.BL_Acacia_Log_C'	Texture2D'/Game/Icon/IC_Acacia_Log.IC_Acacia_Log'
3	2	2	BlueprintGeneratedClass'/Game/BL_Blocks/BL_acacia_planks.BL_acacia_planks_C'	Texture2D'/Game/Icon/IC_acacia_planks.IC_acacia_planks'
4	3	3	BlueprintGeneratedClass'/Game/BL_Blocks/BL_andesite.BL_andesite_C'	Texture2D'/Game/Icon/IC_andesite.IC_andesite'
5	4	4	BlueprintGeneratedClass'/Game/BL_Blocks/BL_bedrock.BL_bedrock_C'	Texture2D'/Game/Icon/IC_bedrock.IC_bedrock'
6	5	5	BlueprintGeneratedClass'/Game/BL_Blocks/BL_birch1.BL_birch1_C'	Texture2D'/Game/Icon/IC_birch1.IC_birch1'
7	6	6	BlueprintGeneratedClass'/Game/BL_Blocks/BL_birch2.BL_birch2_C'	Texture2D'/Game/Icon/IC_birch2.IC_birch2'
8	7	7	BlueprintGeneratedClass'/Game/BL_Blocks/BL_birch_log.BL_birch_log_C'	Texture2D'/Game/Icon/IC_birch_log.IC_birch_log'
9	8	8	BlueprintGeneratedClass'/Game/BL_Blocks/BL_black_glazed_terracotta.BL_black_glazed_terracotta_C'	Texture2D'/Game/Icon/IC_black_glazed_terracotta.IC_black_glazed_terra
10	9	9	BlueprintGeneratedClass'/Game/BL_Blocks/BL_black_glazed_terracotta1.BL_black_glazed_terracotta1_C'	Texture2D'/Game/Icon/IC_black_glazed_terracotta1.IC_black_glazed_terr
11	10	10	BlueprintGeneratedClass'/Game/BL_Blocks/BL_black_glazed_terracotta2.BL_black_glazed_terracotta2_C'	Texture2D'/Game/Icon/IC_black_glazed_terracotta2.IC_black_glazed_terr
12	11	11	BlueprintGeneratedClass'/Game/BL_Blocks/BL_black_glazed_terracotta3.BL_black_glazed_terracotta3_C'	Texture2D'/Game/Icon/IC_black_glazed_terracotta3.IC_black_glazed_terr
13	12	12	BlueprintGeneratedClass'/Game/BL_Blocks/BL_black_terracotta.BL_black_terracotta_C'	Texture2D'/Game/Icon/IC_black_terracotta.IC_black_terracotta'
14	13	13	BlueprintGeneratedClass'/Game/BL_Blocks/BL_black_wool.BL_black_wool_C'	Texture2D'/Game/Icon/IC_black_wool.IC_black_wool'
15	14	14	BlueprintGeneratedClass'/Game/BL_Blocks/BL_blackmat.BL_blackmat_C'	Texture2D'/Game/Icon/IC_blackmat.IC_blackmat'
16	15	15	BlueprintGeneratedClass'/Game/BL_Blocks/BL_blackmetal.BL_blackmetal_C'	Texture2D'/Game/Icon/IC_blackmetal.IC_blackmetal'
17	16	16	BlueprintGeneratedClass'/Game/BL_Blocks/BL_blue_ice.BL_blue_ice_C'	Texture2D'/Game/Icon/IC_blue_ice.IC_blue_ice'



Data Table | Data Table Details

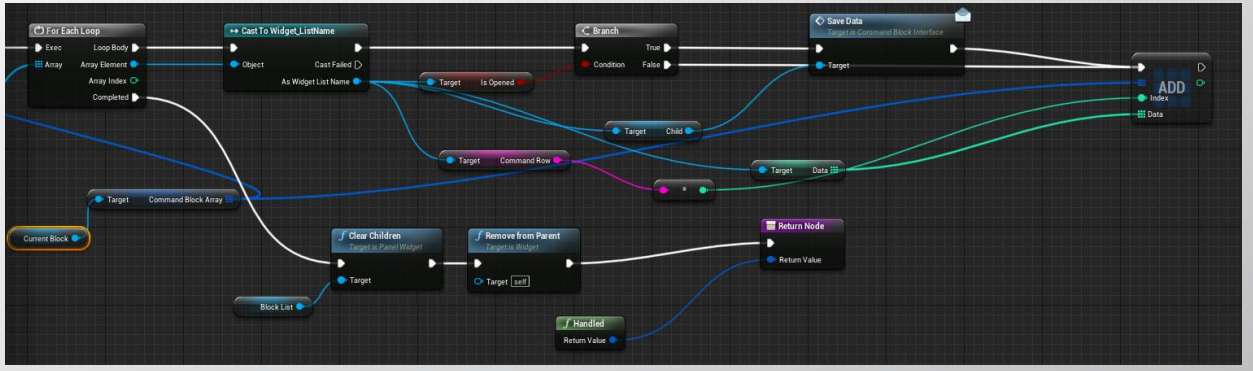
Search

	Row Name	Index	CommandName	WidgetClass
1	0	0	블록 이동	WidgetBlueprintGeneratedClass'/Game/Level/PlaceBlockTest/UI/Comma
2	1	1	반복	WidgetBlueprintGeneratedClass'/Game/Level/PlaceBlockTest/UI/Comma
3	2	2	블록 리셋	WidgetBlueprintGeneratedClass'/Game/Level/PlaceBlockTest/UI/Comma
4	3	3	대기	WidgetBlueprintGeneratedClass'/Game/Level/PlaceBlockTest/UI/Comma
5	4	4	크기 변환	WidgetBlueprintGeneratedClass'/Game/Level/PlaceBlockTest/UI/Comma
6	5	5	상호작용	WidgetBlueprintGeneratedClass'/Game/Level/PlaceBlockTest/UI/Comma
7	6	6	데미지 부여	WidgetBlueprintGeneratedClass'/Game/Level/PlaceBlockTest/UI/Comma
8	7	7	소리 재생	WidgetBlueprintGeneratedClass'/Game/Level/PlaceBlockTest/UI/Comma
9	8	8	회전	WidgetBlueprintGeneratedClass'/Game/Level/PlaceBlockTest/UI/Comma

- 데이터 테이블을 통하여 블록을 관리하여 작업 효율성 증가
- 인벤토리와 커맨드 블록은 데이터 테이블을 읽어와서 구현

5. 개발 내용

커맨드 블록

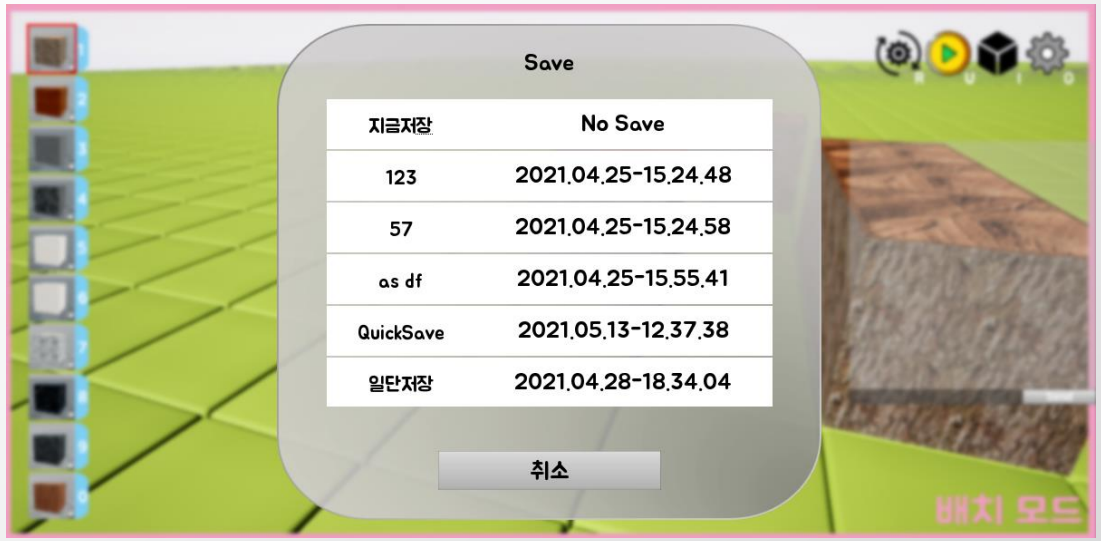



```
if (CommandBlockArray.Num() != 0)
{
    ExecuteCommandBlock(CommandBlockArray[CurrentCommandBlock], DeltaTime);
}
```

- 원하는 커맨드 블록을 등록하고 창을 닫으면 Array로 저장
- 세부 설정 값을 통하여 세부적인 컨트롤이 가능
- 플레이 모드 시작 시 Array에서 커맨드 정보를 가져와서 실행

5. 개발 내용

세이브 로드




```
void UWidget_SaveData::SaveGame()
{
    USaveEditorLevel* Instance = Cast<USaveEditorLevel>(UGameplayStatics::CreateSaveGameObject(USaveEditorLevel::StaticClass()));

    if (SaveName->GetText().ToString() != "New File")
    {
        TArray<AActor*> actors;

        UGameplayStatics::GetAllActorsWithTag(GetWorld(), "Saveable", actors);

        for (auto target : actors)
        {
            Fblockinfo temp;
            auto casted = Cast<ABlockBase>(target);
            if (casted)
            {
                temp.location = casted->GetOrigin();
                temp.CommandArray = casted->CommandBlockArray;
            }
            else
                temp.location = target->GetActorLocation();

            temp.blockclass = target->GetClass();


            Instance->blockarray.Add(temp);
        }

        Instance->Savetime = FDateTime::Now();
        UGameplayStatics::SaveGameToSlot(Instance, SaveName->GetText().ToString(), Instance->UserIndex);
    }
}
```

```
TArray<FString> Saves;
const FString SavesFolder = FPaths::ProjectSavedDir() + TEXT("SaveGames");

if (!SavesFolder.IsEmpty())
{
    FFindSavesVisitor Visitor;
    FPlatformFileManager::Get().GetPlatformFile().IterateDirectory(*SavesFolder, Visitor);
    Saves = Visitor.SavesFound;
}

return Saves;
```

- 게임 시작 시 이전에 만든 맵을 선택해서 불러올 수 있음
- 게임 종료 시 자동 저장 구현
- 원하는 명칭으로 저장 가능
- 저장한 시간도 같이 저장
- 멀티 플레이를 통하여 접속 시 만들고 있는 맵 정보를 불러옴

6. 문제점 및 보완책

시간 마법

문제점
- 다른 작업에 비하여 우선순위에서 밀림.
- 진척 상황이 매우 낮음.

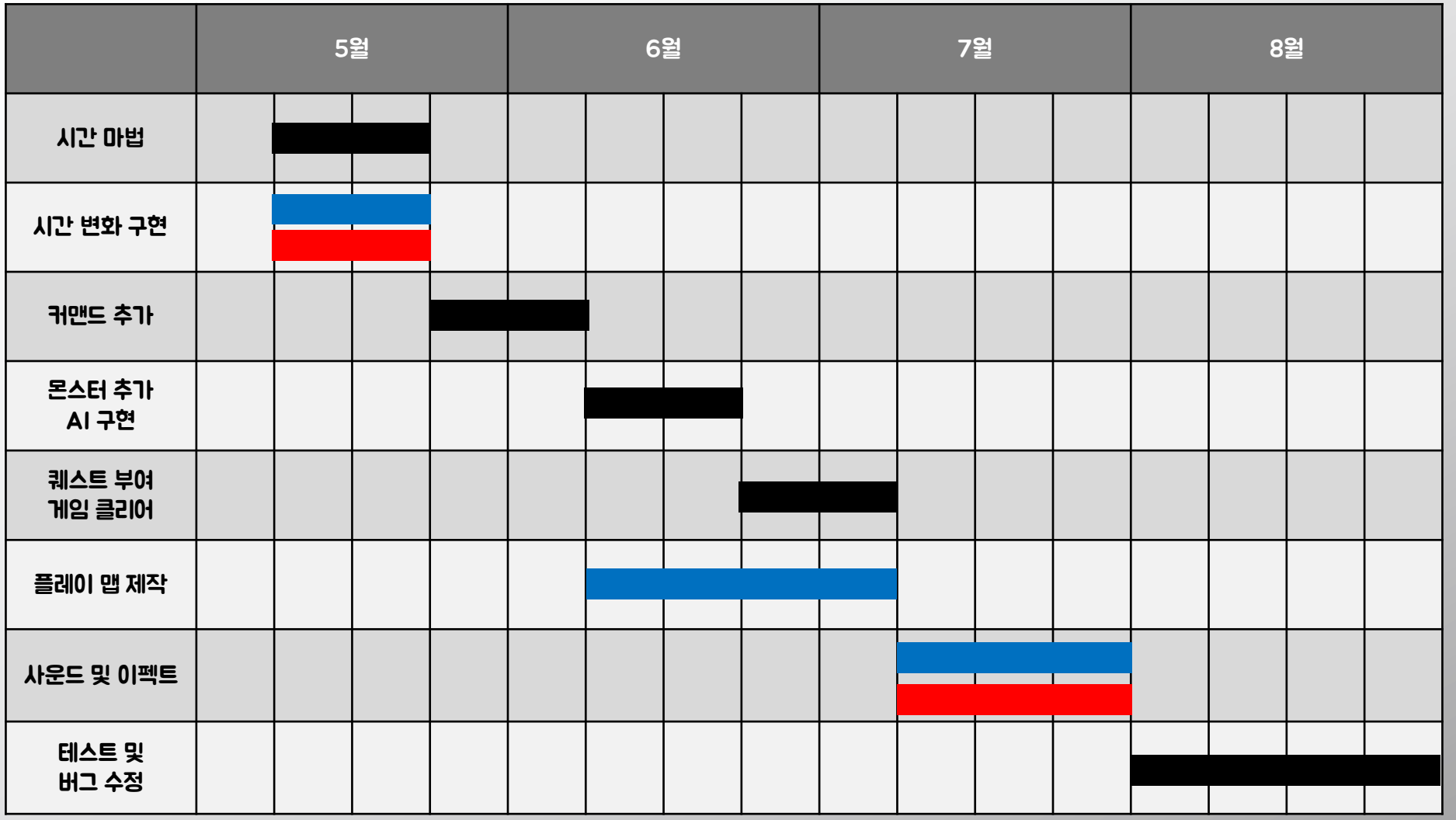
보완책
- 다른 작업들이 거의 끝난 만큼 진행하면 됨.

서버 관련

문제점
- 전체적인 서버 관련 구현 속도가 매우 느림
- 진척 상황이 매우 낮고 각종 버그 존재

보완책
- 꾸준한 시간 투자
- 지도교수님 및 팀원들과 상담

7. 향후 개발 일정



임건호	
성기홍	
권호민	
모두	

8. 데모 시연