
포트폴리오 기술 문서

| 유수빈 | subeenyoo1002@gmail.com |

1

프로필

2

프로젝트 목록

3

일정 및 프로젝트 관리 방법

4

프로젝트 및 기술 상세 설명



유수빈



1998 . 10 . 02



subeenyoo1002@gmail.com



E-MAIL



GITHUB



BLOG

SKILL

C++

DirectX11

HLSL

Unity

PROJECT

2022. 12 **Shattered** 모작

2022. 8 **Celeste** 모작

2022. 2 **Helltaker** 모작

2021. 10 **School Nocturnble** 제작

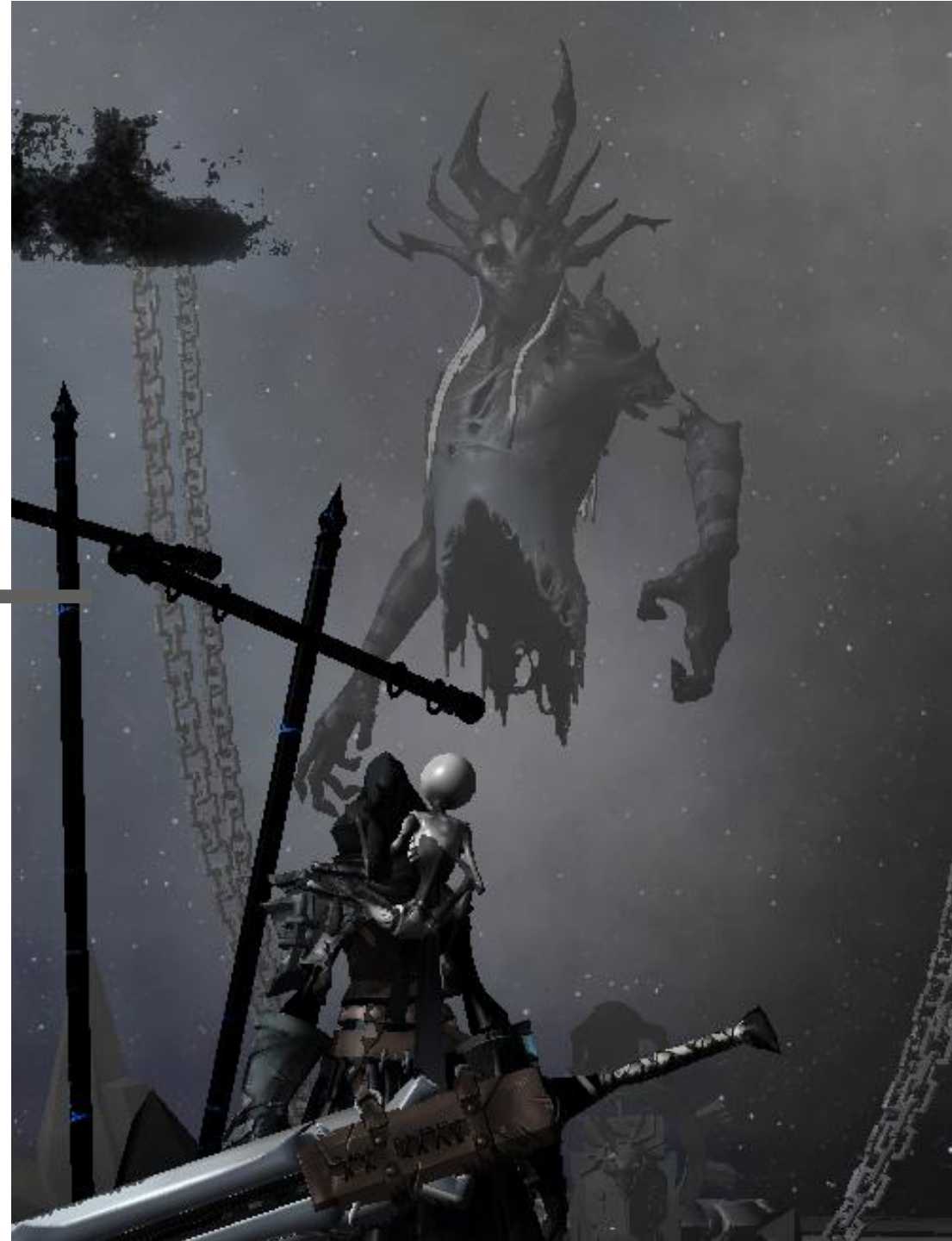
프로젝트 목록

2022. 10 ~ 2022.12

Shattered: Tale of the Forgotten King 모작 team

| DirectX11 | HLSL | FMOD | FBXSDK | ImGui | C++ |

[상세 페이지 바로가기](#)



2022. 7 ~ 2022. 8

Celeste 모작

| DirectX11 | HLSL | FMOD | ImGui | C++ |

[상세 페이지 바로가기](#)



2022. 2 ~ 2022. 3

Helltaker 모작

| Win32 API | FMOD | C++ |

[상세 페이지 바로가기](#)



2021. 5 ~ 2021.10

School Nocturnble team

| Unity | Tobii Eyetracker Unity SDK | C# |

[상세 페이지 바로가기](#)



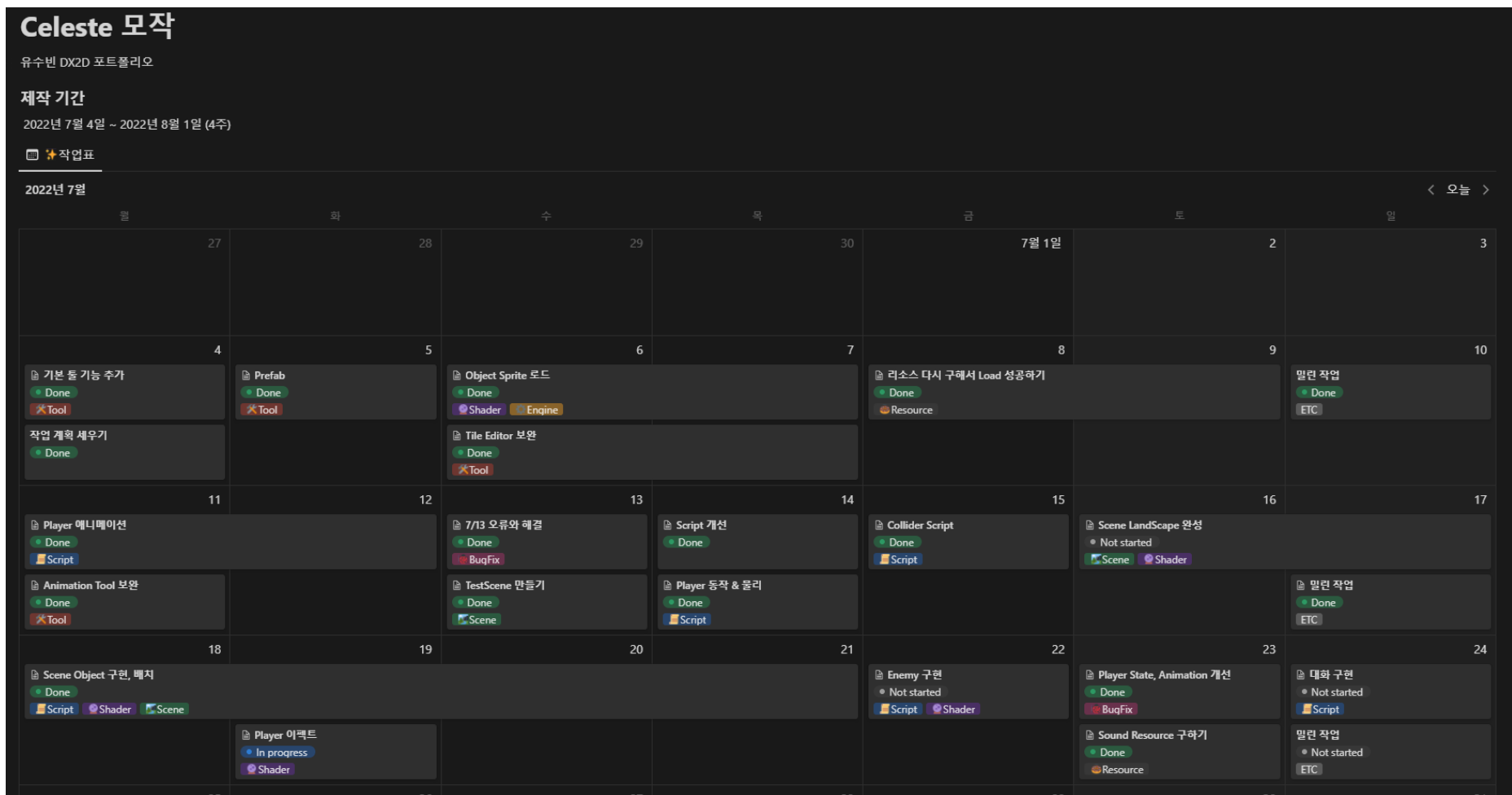
A black and white photograph of a person's hands typing on a laptop keyboard. The person is wearing a dark, textured sweater. The background is dark and out of focus, showing another person's hand near a laptop. The overall mood is professional and focused.

일정 및 프로젝트 관리법

일정 및 프로젝트 관리법

| 01. 일정 관리 방법

Notion을 이용해 프로젝트의 일정 관리를 했습니다.



프로젝트 'Celeste 모작'과 관련된 개발 일정을 Notion Calendar로 정리

| 02. 프로젝트 관리 방법

TortoiseSVN을 이용해서 프로젝트 버전 관리를 하고 팀원들과 협업했습니다.

The screenshot shows the TortoiseSVN Log Messages window for a project named 'Shattered'. The window title is 'C:\Users\bueon\source\repos\MyShattered_ - Log Messages - TortoiseSVN'. The interface includes a search filter, date range selection (From: 12/13/2022, To: 12/19/2022), and a table of revisions. Revision 221 is selected, showing its message and affected paths.

Revision	Actions	Author	Date	Message
228	[Info]	babo_ysb	Monday, December 19, 2022 4:20:35 PM	magam
227	[Info]	babo_jjm	Monday, December 19, 2022 1:30:43 PM	state
226	[Info]	babo_yik	Monday, December 19, 2022 1:13:26 PM	1. sound
225	[Info]	babo_ysb	Monday, December 19, 2022 12:22:18 PM	아마도 최종본
224	[Info]	babo_ssw	Monday, December 19, 2022 11:02:38 AM	CSound FMOD에 제대로 동작하도록 수정
223	[Info]	babo_yik	Monday, December 19, 2022 10:23:49 AM	1. sound BGM
222	[Info]	babo_jjm	Monday, December 19, 2022 10:06:06 AM	player
221	[Info]	babo_yik	Monday, December 19, 2022 9:02:02 AM	sound
220	[Info]	babo_yik	Monday, December 19, 2022 8:52:54 AM	1. monster die mtrl
219	[Info]	babo_yik	Monday, December 19, 2022 8:24:56 AM	1. fog color 2. add sky box dds 3.
218	[Info]	babo_jjm	Monday, December 19, 2022 8:15:12 AM	Cut Scene Spline CinemaCamScript Route Create

1. Boss Stage Navi 적용
2. Boss Stage prefab 수정
3. Navimap UI 수정

Path	Action	Copy from path	Revision
/OutputFile/bin/content/scene/TestScene.scene	Modified		
/OutputFile/bin/imgui.ini	Modified		
/OutputFile/bin_debug/ScriptList.txt	Modified		
*/Project/Client/CTestScene.cpp	Modified		
*/Project/Script/CScriptMgr.cpp	Modified		
/Project/Script/CScriptMgr.h	Modified		

Showing 100 revision(s), from revision 129 to revision 228 - 1 revision(s) selected, showing 1 / changed paths

Show only affected paths
 Stop on copy/rename
 Include merged revisions

Show All | Next 100 | Refresh

Statistics | Help | OK

프로젝트 'Shattered'의 TortoiseSVN 로그



프로젝트 및 기술 상세 설명

The background is a soft grey gradient. A thin, white, curved line arches across the middle of the frame. A clear, glass-like sphere is positioned at the peak of this arc, appearing to rest on it. The overall aesthetic is clean and modern.

**Project.1 Shattered: Tale of the
Forgotten King 모작**

Shattered: Tale of the Forgotten King 모작 team

개발 인원

4인 개발

프로젝트 설명

본 프로젝트는 DirectX 11을 이용해 4명의 팀원과 함께 3개월간 만든 3D 프로젝트입니다. 원작은 3인칭 소울라이크 게임으로 HLSL을 이용하여 다양한 그래픽 효과를 직접 구현하기 좋다는 점에서 모작 게임으로 선정하였습니다.

구현 내용

Nav Mesh 제작, Boss 전투(2페이지)와 이펙트

구현 초점

구현한 기능을 팀원들이 쉽게 접근하여 사용할 수 있게 하는 것에 초점을 두고 개발하였습니다.

01. Navi Map (Nav Mesh)

CNaviMapData

Save/Load

CNaviCell

CNaviMap

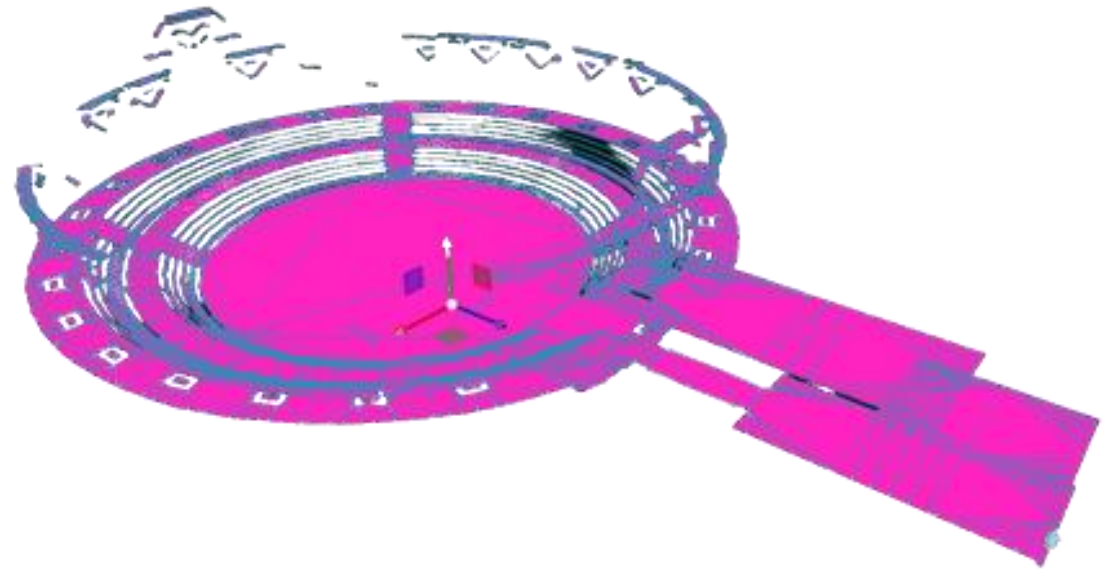
CNaviAgent

충돌 테스트

CNaviMapData

: 맵 지형 정보 클래스

```
class CNaviMapData : public CRes
{
private:
    CMesh*          m_pNavMesh;
    vector<CNaviCell*> m_vecNaviCell;
```



01. Navi Map (Nav Mesh)

CNaviMapData

Save/Load

CNaviCell

CNaviMap

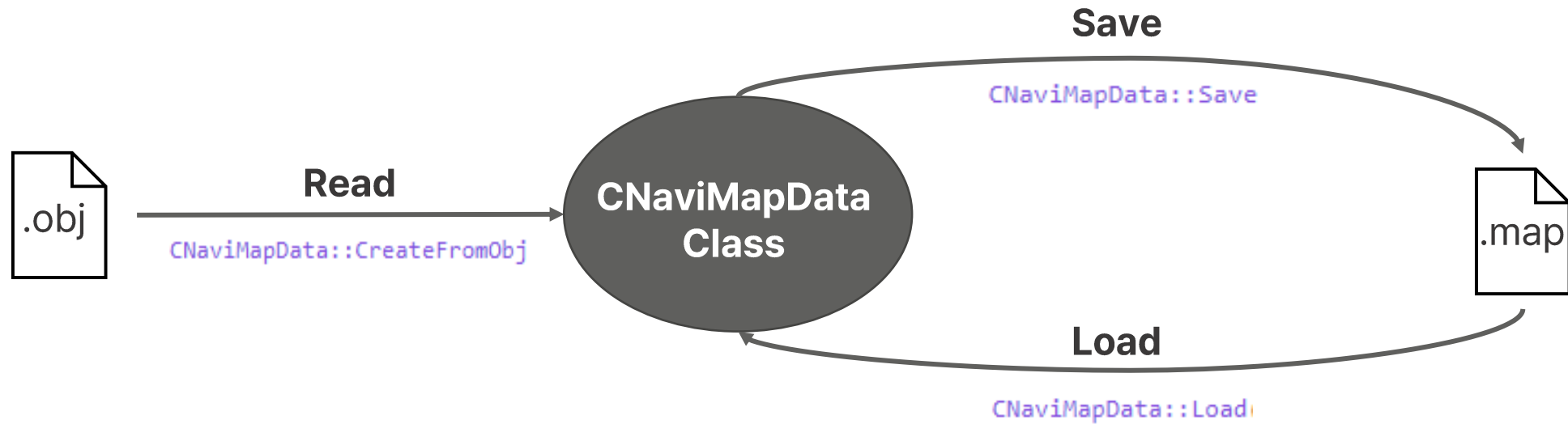
CNaviAgent

충돌 테스트

NaviMapData Save / Load

지형 표면 메쉬가 담긴 .obj 파일로부터

CNaviMapData를 생성하여 이를 파일로 저장했습니다.



01. Navi Map (Nav Mesh)

CNaviMapData

Save/Load

CNaviCell

CNaviMap

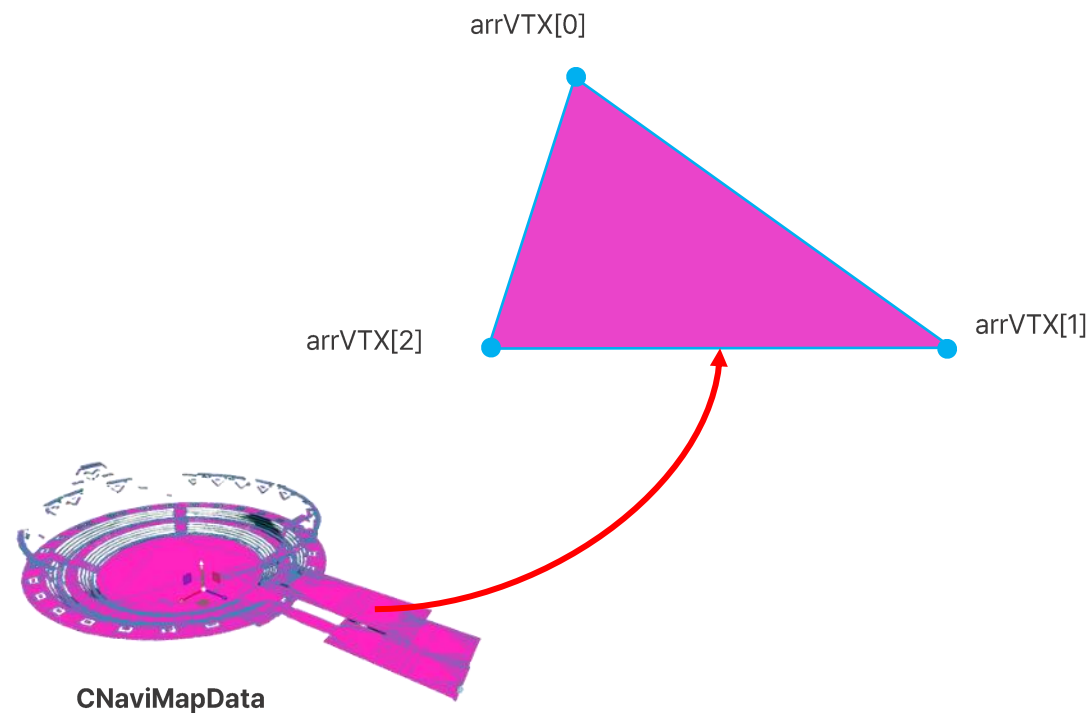
CNaviAgent

충돌 테스트

CNaviCell

: NaviMap을 구성하는 삼각형 클래스

```
class CNaviCell
{
private:
    int         m_iIDX;
    Vtx         m_arrVTX[3];
    vector<int> m_vecLinkedCell;
```



01. Navi Map (Nav Mesh)

CNavimapData

Save/Load

CNavicell

CNavimap

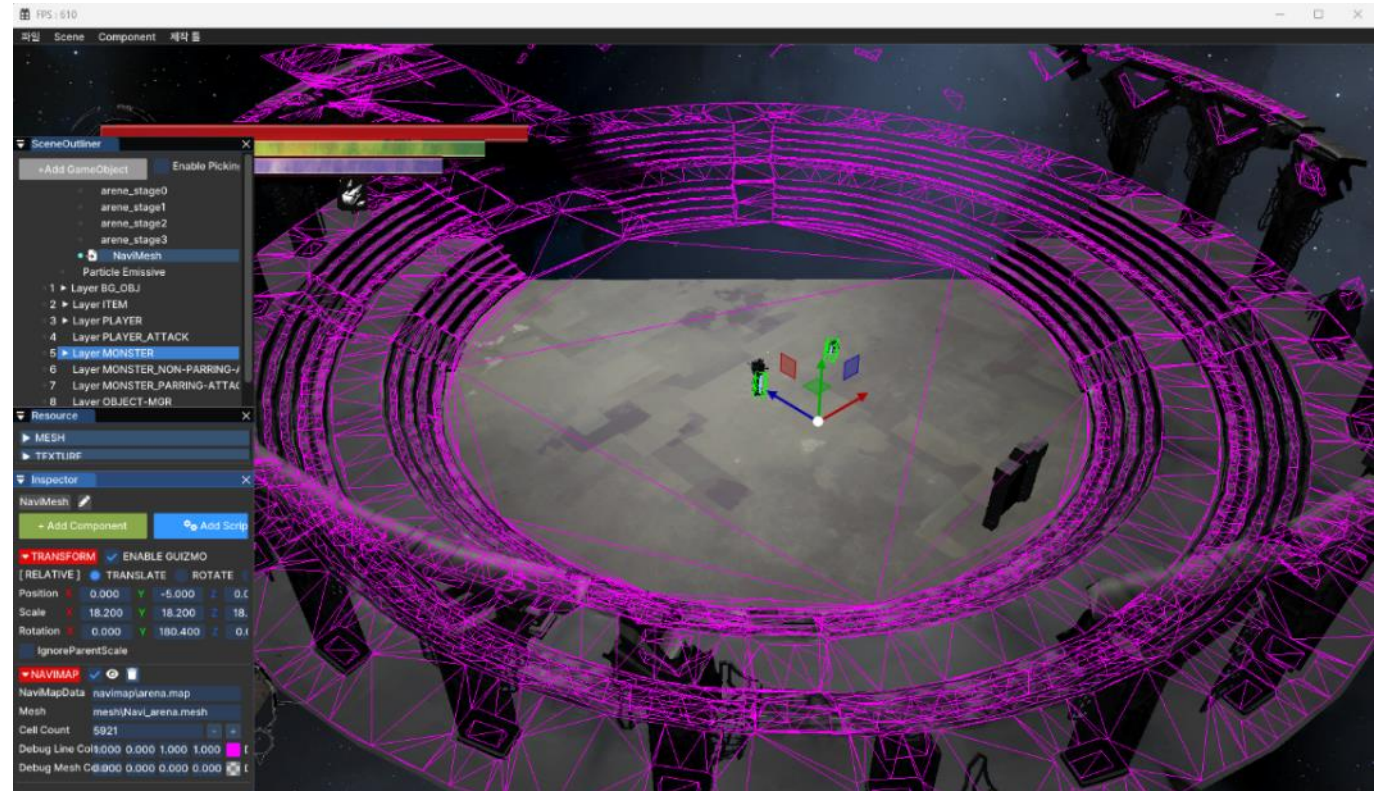
CNaviaAgent

충돌 테스트

CNavimap

: CNavimapData를 컴포넌트화 해주는 클래스

```
class CNavimap : public CComponent
{
private:
    Ptr<CNavimapData> m_pNavimapData;
    Vec4                m_vDebugMeshColor;
    Vec4                m_vDebugLineColor;
```



01. Navi Map (Nav Mesh)

CNaviMapData

Save/Load

CNaviCell

CNaviMap

CNaviAgent

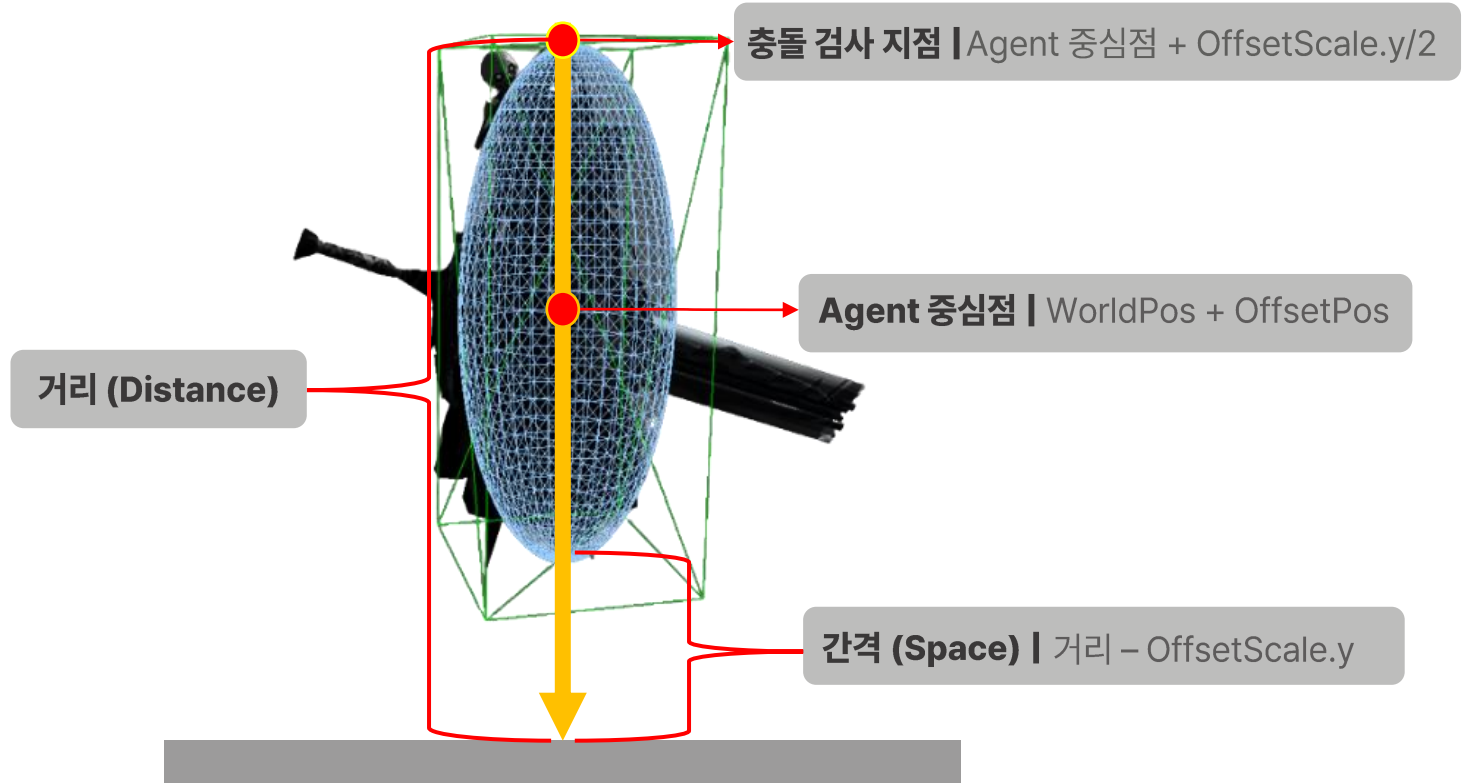
충돌 테스트

NaviAgent

: NaviMap위를 이동하는 캐릭터에 추가하는
컴포넌트 클래스

```
class CNaviAgent : public CComponent
{
private:
    CNaviMap* m_pTargetNaviMap;
    CRigidBody* m_pRigidBody;

    Vec3 m_vOffsetSize;
    Vec3 m_vOffsetPos;
    Vec3 m_vPrevPos;
    Vec4 m_vDebugAgentColor;
    int m_iCurCellIndex;
    bool m_bGravity;
```



01. Navi Map (Nav Mesh)

CNaviMapData

Save/Load

CNaviCell

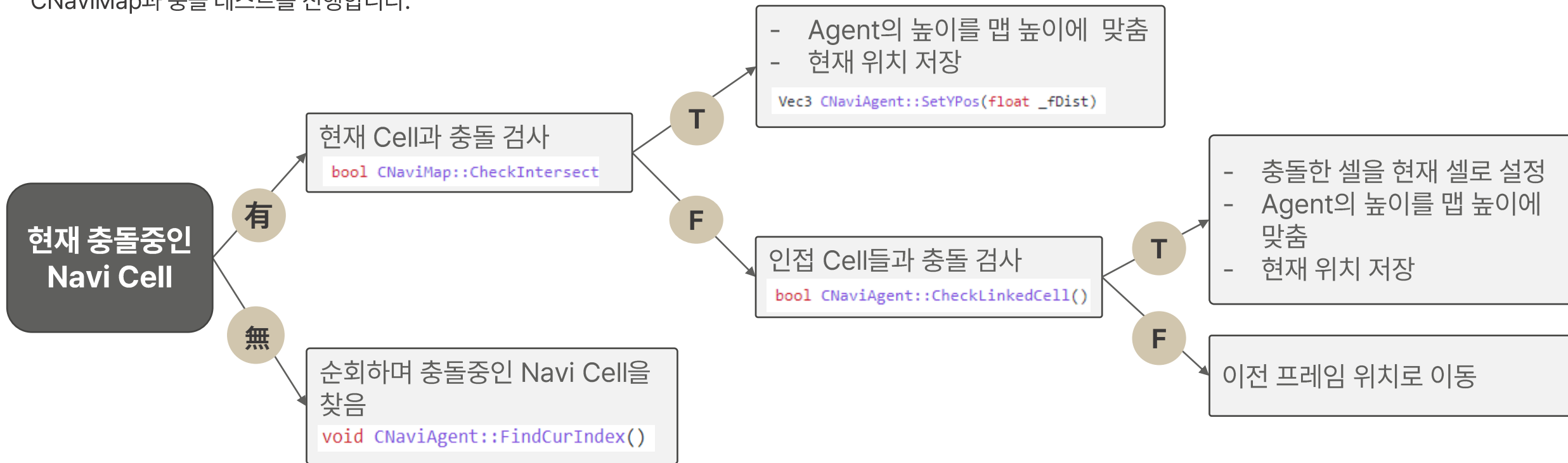
CNaviMap

CNaviAgent

충돌 테스트

충돌 테스트 과정 설계

CNaviAgent는 다음과 같은 과정으로
CNaviMap과 충돌 테스트를 진행합니다.



02. Boss

FSM

Object Pool

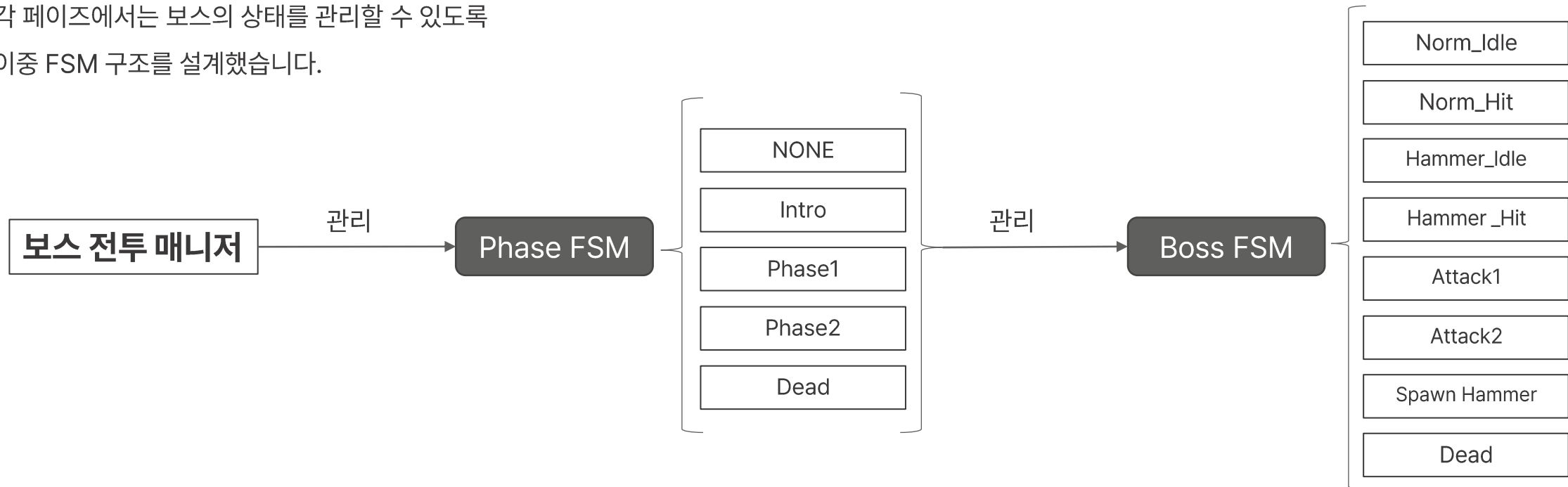
Effect

이중 FSM 구조 설계

보스 전투 매니저에서 페이즈를 관리하고

각 페이즈에서는 보스의 상태를 관리할 수 있도록

이중 FSM 구조를 설계했습니다.



02. Boss

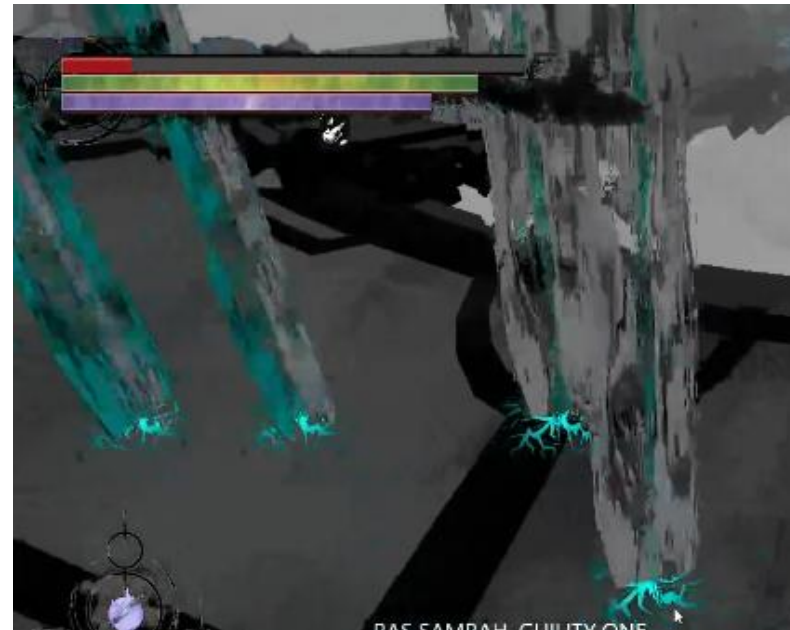
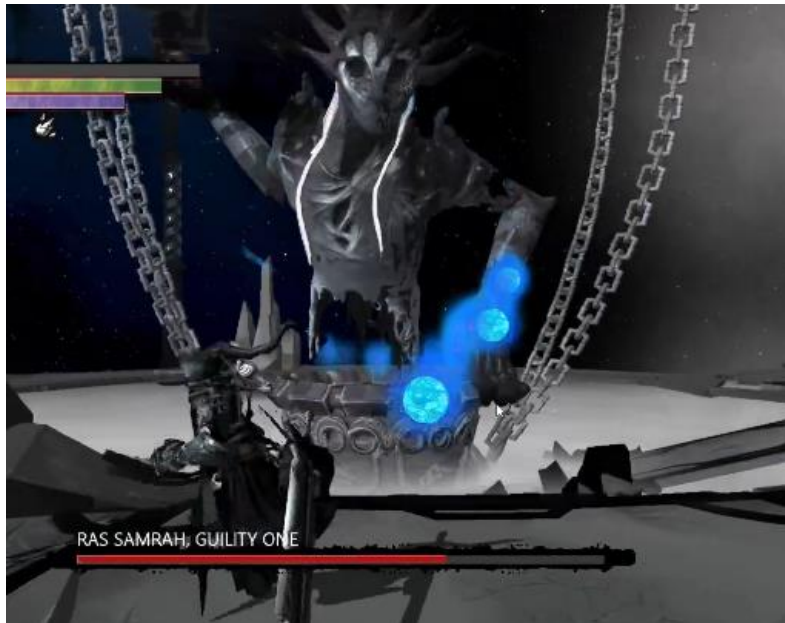
FSM

Object Pool

Effect

Object Pool

반복적인 오브젝트 생성/삭제로 인한 프레임 드랍을 해결하기 위해
보스 공격 오브젝트에서 오브젝트 풀을 사용했습니다.



02. Boss

FSM

Object Pool

Effect



Distortion(post processing)



Decal

The background is a soft grey gradient. A thin, white, curved line arcs across the middle of the frame. A small, clear glass sphere is positioned on this line, slightly to the right of the center. The sphere is reflective, showing highlights and shadows. In the top-left and bottom-right corners, there are large, semi-transparent, light grey circular shapes that appear to be parts of larger spheres or planets.

Project.2 Celeste 모작

Celeste 모작

개발 인원

1인 개발

프로젝트 설명

본 프로젝트는 DirectX 11을 이용해 1개월간 제작한 2D 프로젝트입니다. 원작은 2D 플랫폼어 게임으로 분위기와 플레이 시스템이 매력적이어서 모작 게임으로 선정하였습니다.

구현 내용

Tile Tool, Animation Tool, Game Play

구현 초점

게임 제작에 필요한 툴을 직접 만들고 이를 이용하여 게임을 제작하는 것에 초점을 두고 개발하였습니다.

01. Tool

Animation Tool

Tile Tool

Animation Tool로 애니메이션 만들기



1

아틀라스 텍스처 변경

2

드래그하여 현재 프레임 선택



01. Tool

Animation Tool

Tile Tool

Animation Tool로 애니메이션 만들기



3

옵션 조절

4

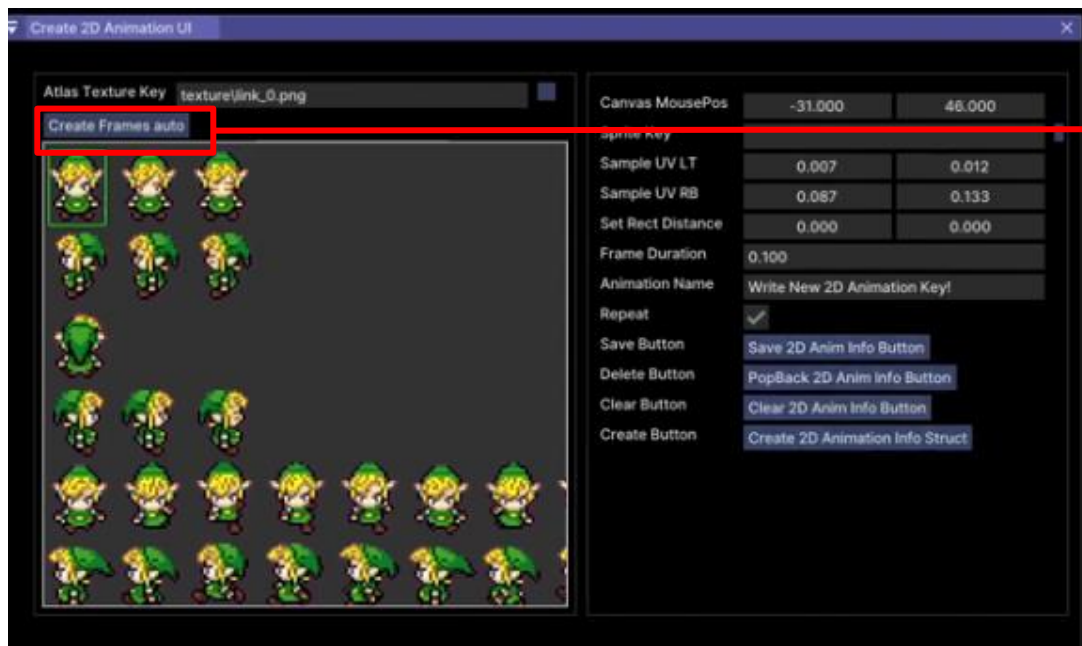
애니메이션 생성

01. Tool

Animation Tool

Tile Tool

Animation Tool로 애니메이션 만들기



아틀라스에 대한 프레임 정보(.xml)가 있다면
자동으로 애니메이션을 생성해줍니다.

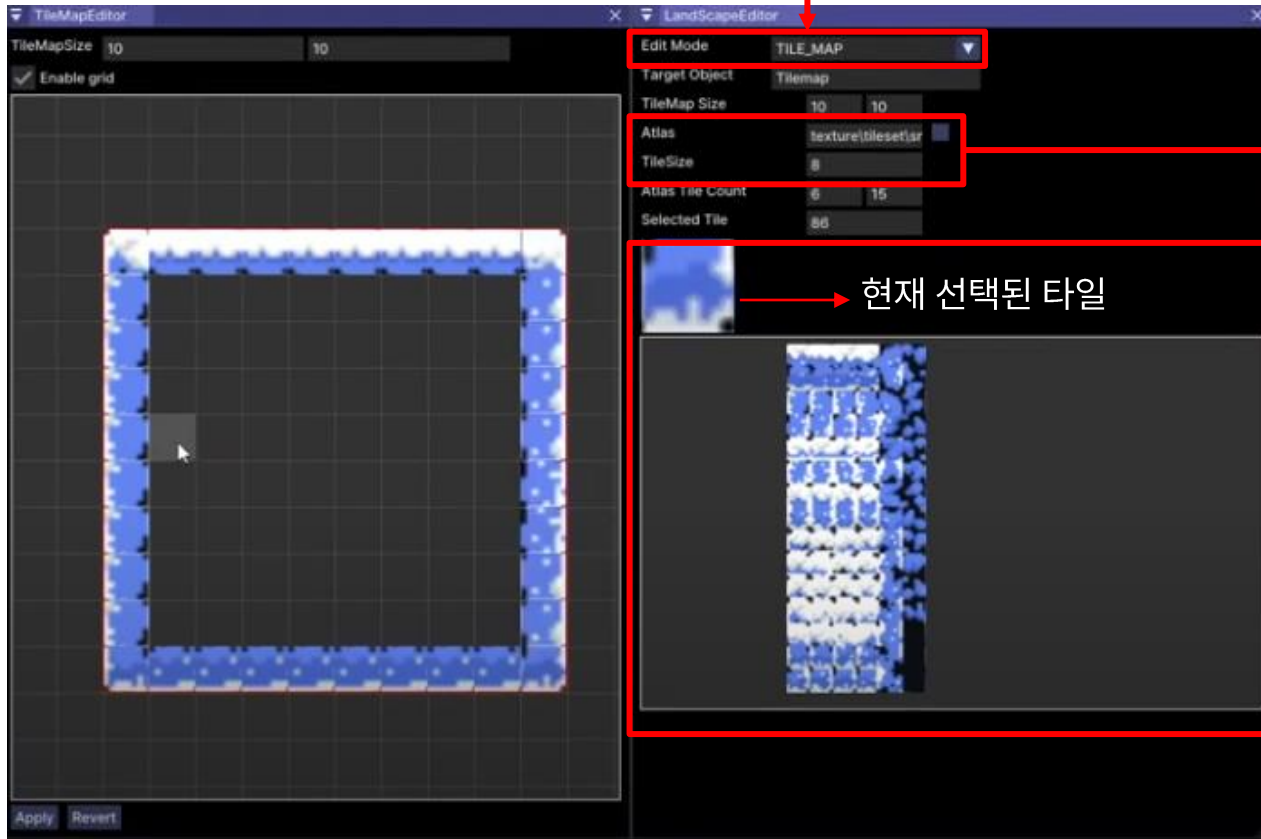
```
<TextureAtlas imagePath="Player_clim.png" width="480" height="32">
  <sprite n="climb00.png" x="0" y="0" w="32" h="32"/>
  <sprite n="climb01.png" x="32" y="0" w="32" h="32"/>
  <sprite n="climb02.png" x="64" y="0" w="32" h="32"/>
  <sprite n="climb03.png" x="96" y="0" w="32" h="32"/>
  <sprite n="climb04.png" x="128" y="0" w="32" h="32"/>
  <sprite n="climb05.png" x="160" y="0" w="32" h="32"/>
  <sprite n="climb06.png" x="192" y="0" w="32" h="32"/>
  <sprite n="climb_lookBack00.png" x="224" y="0" w="32" h="32"/>
  <sprite n="climb_lookBack01.png" x="256" y="0" w="32" h="32"/>
  <sprite n="climb_lookBack02.png" x="288" y="0" w="32" h="32"/>
  <sprite n="climb_lookBack03.png" x="320" y="0" w="32" h="32"/>
  <sprite n="climb_lookBack04.png" x="352" y="0" w="32" h="32"/>
  <sprite n="climb_lookBack05.png" x="384" y="0" w="32" h="32"/>
  <sprite n="climb_lookBack06.png" x="416" y="0" w="32" h="32"/>
  <sprite n="climb_lookBack07.png" x="448" y="0" w="32" h="32"/>
  <sprite n="climb_idle00.png" x="0" y="0" w="32" h="32"/>
  <sprite n="climb_idle01.png" x="0" y="0" w="32" h="32"/>
</TextureAtlas>
```

01. Tool

Animation Tool

Tile Tool

Tile Tool로 타일맵 제작하기

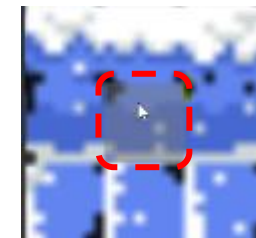


① 타일 아틀라스 선택 후 타일 사이즈 조절

② 팔레트에서 타일 선택
타일 사이즈에 맞게 팔레트에서 선택되는 영역의 크기가 조절됩니다.

8px

16px

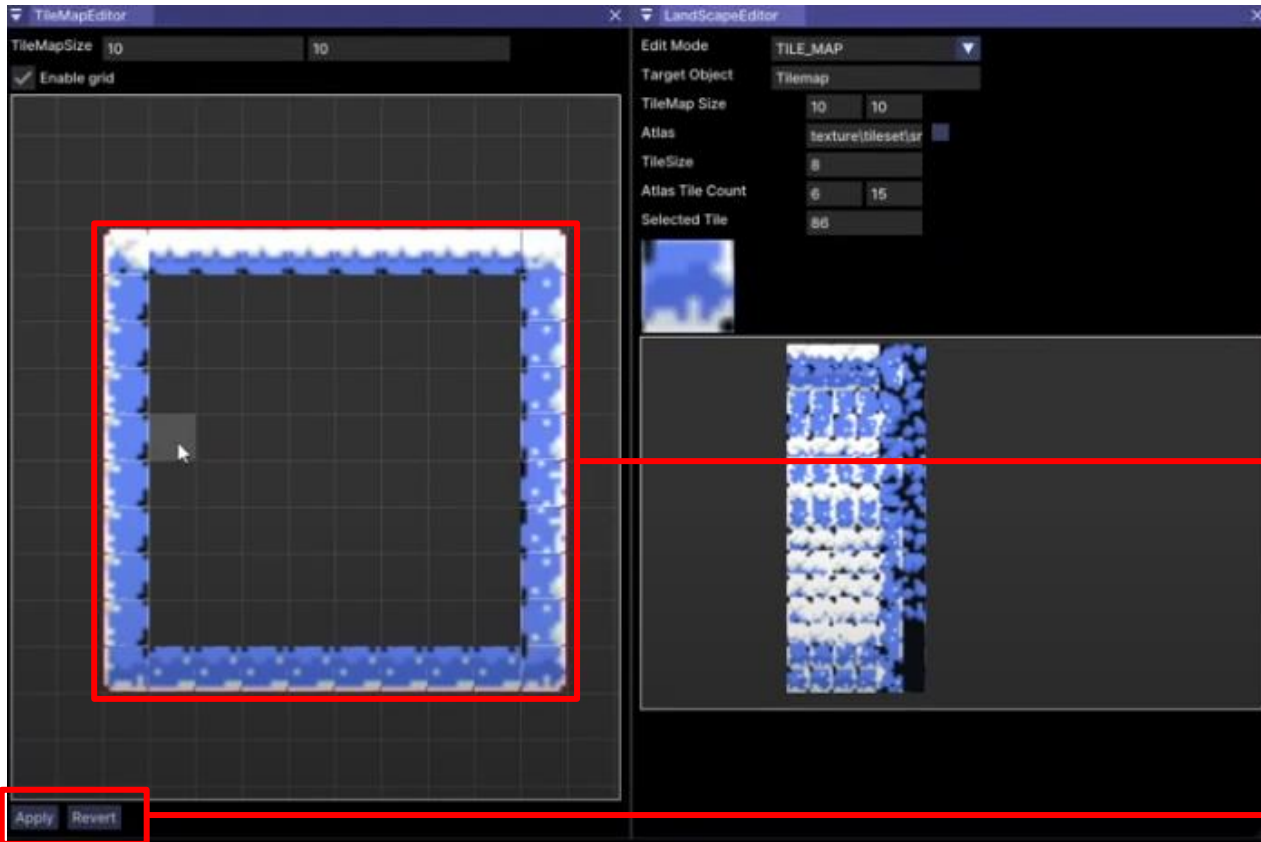


01. Tool

Animation Tool

Tile Tool

Tile Tool로 타일맵 제작하기



3

타일맵 에디터에서 타일 찍기

4

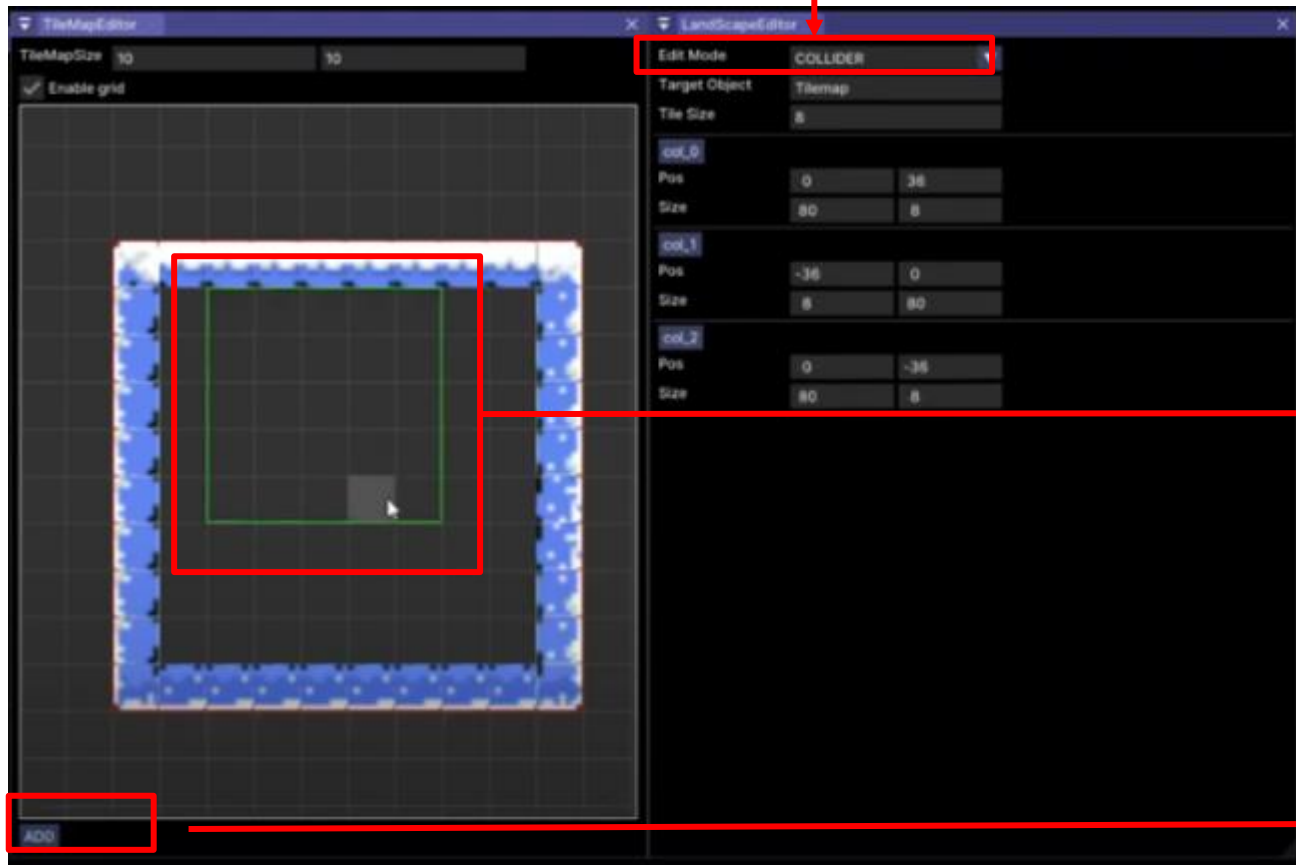
Apply를 눌러 타일 적용

01. Tool

Animation Tool

Tile Tool

Tile Tool로 충돌체 추가하기



①

충돌체 영역을 드래그하여 표시

④

Apply를 눌러 충돌체 적용

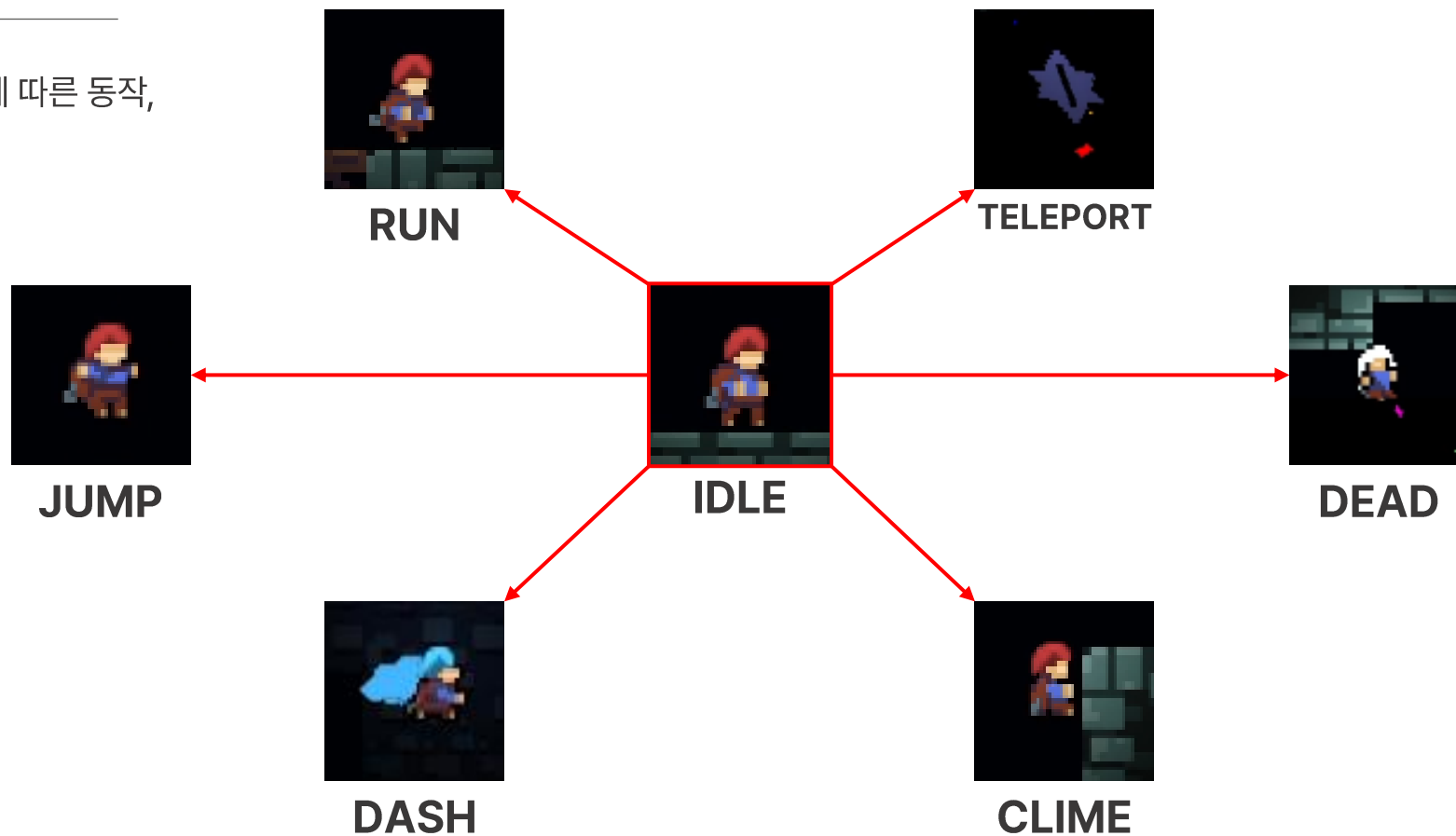
02. Game Play

Player

최적화

Player States

다양한 플레이어 상태와 그에 따른 동작,
기능을 구현했습니다.

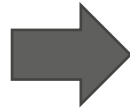
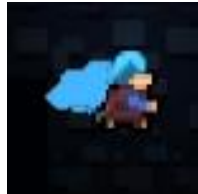


02. Game Play

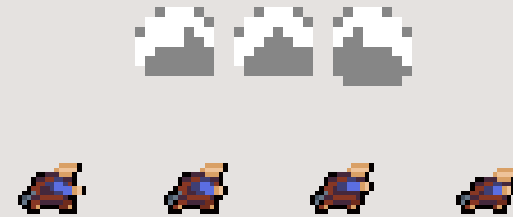
Player

최적화

Player Hair



Player가 Dash 시에 머리카락을 파란색으로 표시



- 상황에 따른 hair 컬러 셰이더 연산
- 애니메이션에 따라서 hair 위치 조정

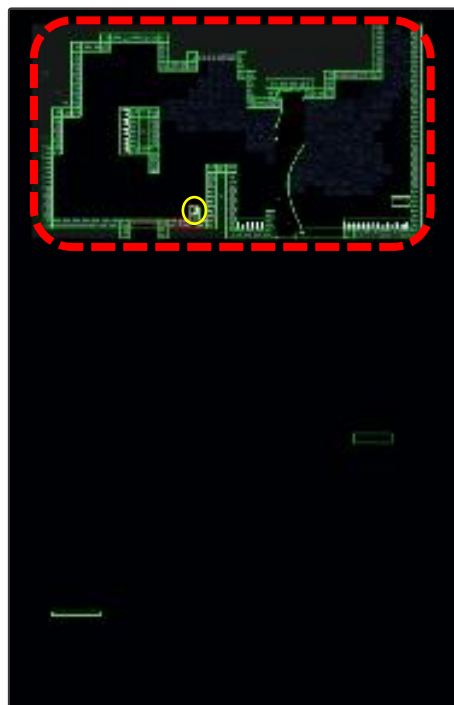
02. Game Play

Player

최적화

Stage 최적화

불필요한 충돌체 연산을 줄이기 위해 Trigger를 설치하여 플레이어가 지나온 Stage는 삭제하고, 플레이어가 진입할 Stage를 생성해주었습니다.



1.
현재 스테이지 생성



2.
다음 스테이지 생성



3.
전전 스테이지 삭제

The background is a smooth grey gradient. A thin, white, curved line arcs across the middle of the frame. A small, clear glass sphere is positioned on this line, slightly to the right of the center. The sphere is transparent and shows reflections. In the top-left and bottom-right corners, there are large, semi-transparent, light grey circular shapes that appear to be parts of larger spheres or planets.

Project.3 Helitaker 모작

Helltaker 모작

개발 인원

1인 개발

프로젝트 설명

본 프로젝트는 Windows API을 바탕으로 1개월간 제작한 2D 프로젝트입니다. 원작은 2D 퍼즐 게임으로 깔끔한 리소스가 WinAPI 환경에서 활용하기 좋아 모작 게임으로 선정하였습니다.

구현 내용

Tile Tool, Game Play

구현 초점

타일 단위 이동하고 오브젝트와 상호작용하며 퍼즐을 푸는 게임 플레이를 구현하는데 집중했습니다.

01. Game Play

Tile 기반 게임 플레이

플레이어는 타일을 기반으로 움직입니다. 다음 타일의 속성 혹은 오브젝트 유무에 따라 이동/상호작용하거나 다음 스테이지로 넘어갑니다.



02. Tile Edit

Tile 편집

Debug모드에서 타일을 클릭하여 편집할 수 있습니다. 편집한 타일은 파일로 저장됩니다.

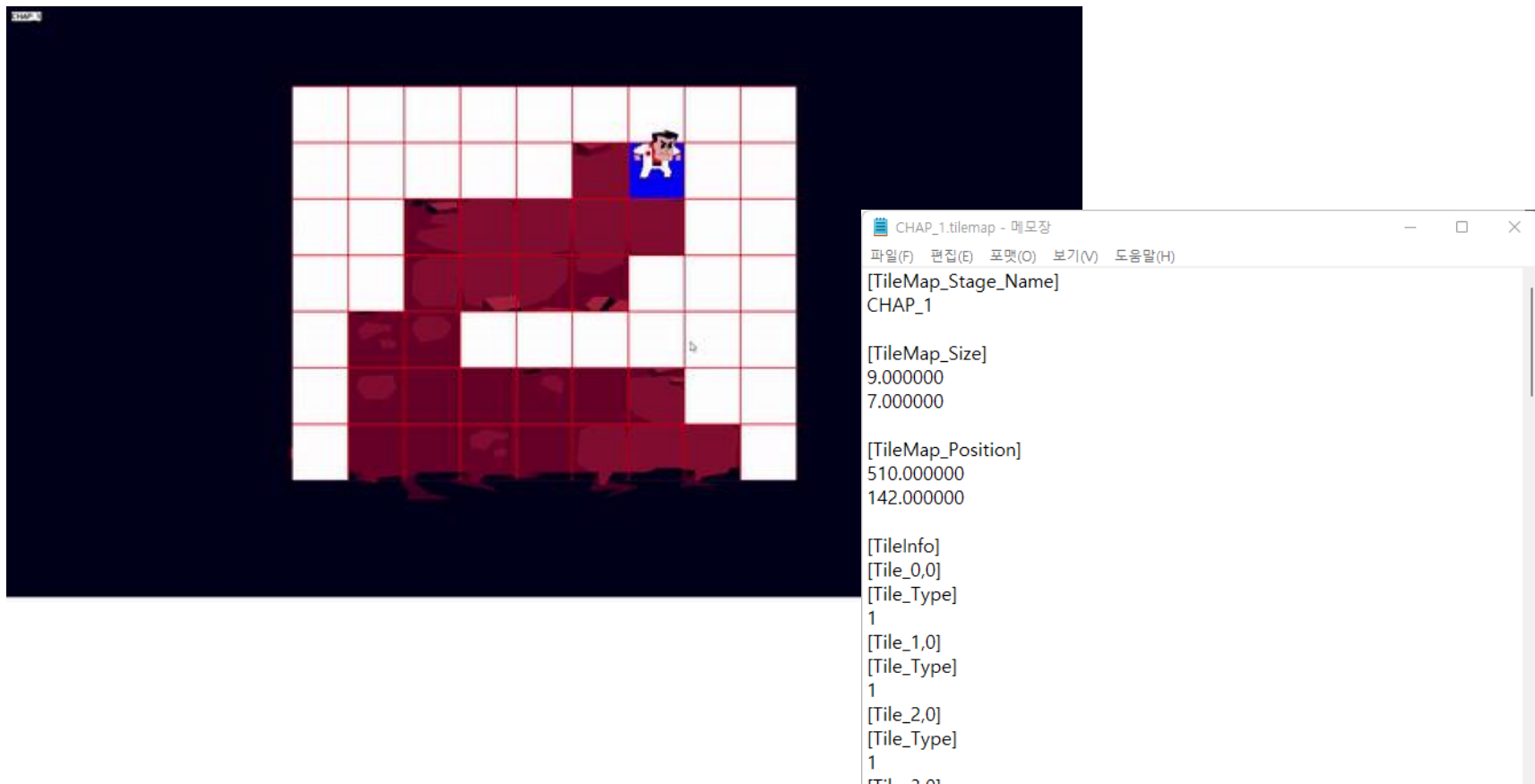
하양 - 벽

파랑 - 플레이어 출발 지점

빨강 - 목표 도착 지점

초록 - NPC 위치

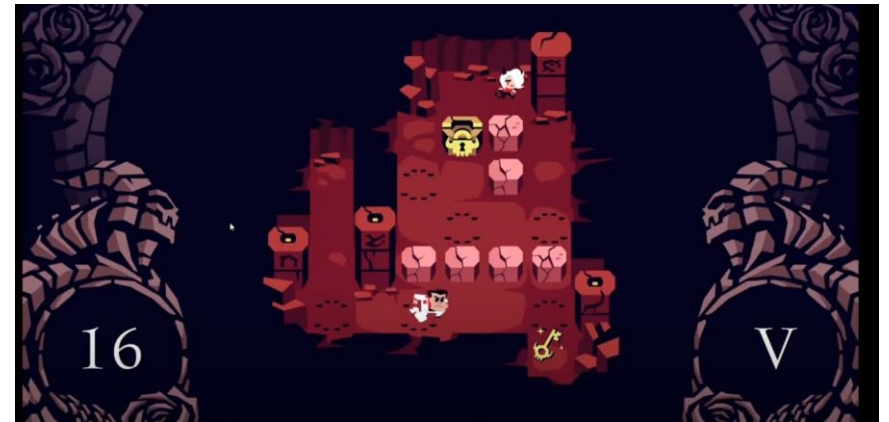
노랑 - 키 위치



03. UI

UI

선택지를 키보드로 선택할 수 있는 Dialog타입과 이동 횟수와 스테이지 정보를 표시해주는 인 게임 UI 두가지를 제작하였습니다.



The background is a smooth grey gradient. A thin, white, curved line arches across the middle of the frame. A clear, glass-like sphere is positioned at the peak of this arc. In the top-left and bottom-right corners, there are large, semi-transparent, light grey circular shapes that appear to be parts of larger spheres or planets.

Project.4 School Nocturnble

School Nocturnble

개발 인원

3인 개발 (아트 2인 + 프로그래밍 1인)

프로젝트 설명

본 프로젝트는 Unity를 이용해 10개월간 제작한 창작 2D 프로젝트입니다. 덕성여자대학교 졸업 작품으로 제작하였으며 Eyetracker를 컨트롤러로 사용했다는 점이 특징입니다.

구현 내용

퍼즐, 턴제 전투, SDK 연동, 인벤토리, 대화 시스템

구현 초점

Eyetracking을 이용한 게임 콘텐츠 제작에 초점을 맞춰 개발하였습니다.

01. EyeTracking

SDK 사용

Game Play

SDK를 이용하여 Eyetracker 이용하기

Tobii Eyetacker로 얻은 눈동자 위치 정보를 SDK를 이용해 오브젝트의 움직임을 구현했습니다.

```
public void MoveGOwithEyetracker(GameObject _go, float _speed)
{
    if (!TobiiAPI.IsConnected && !TobiiAPI.GetDisplayInfo().IsValid)
        return;

    GazePoint gazePoint = TobiiAPI.GetGazePoint();
    if (!float.IsNaN(gazePoint.GUI.x) || !float.IsNaN(gazePoint.GUI.y))
    {
        if (gazePoint.IsRecent())
        {
            Vector3 targetPos = Managers.Eyetracker.ProjectionToPlaneInWorld(gazePoint, Camera.main);
            _go.transform.position = Vector2.Lerp(_go.transform.position, new Vector2(targetPos.x, targetPos.y), Time.deltaTime * _speed);
        }
    }
}
```

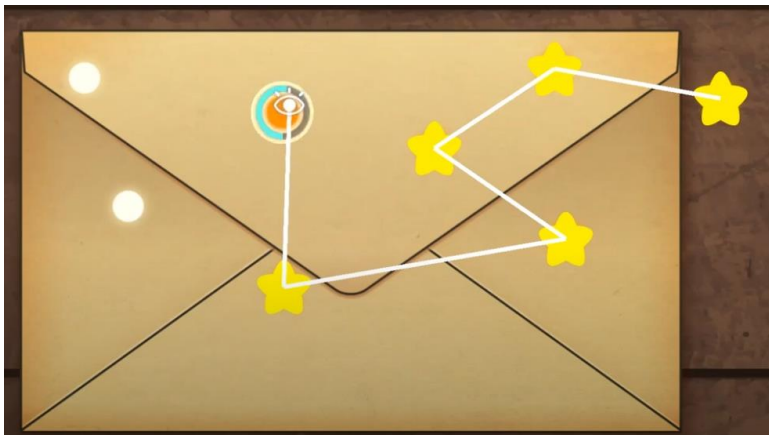
01. EyeTracking

SDK 사용

Game Play

Eyetracking을 활용한 게임 플레이

Eyetracking을 활용해 퍼즐, 오브젝트 상호작용, 전투 등을 구현했습니다.



퍼즐



오브젝트 상호작용



전투

02. Game Play

그 외 다양한 게임 콘텐츠 구현

퀘스트 시스템, 대화 시스템, 인벤토리, 턴제 전투 등 다양한 콘텐츠를 구현했습니다.



감사합니다.