



장르 | 다크 판타지, APRG (모작)

개발 인원 | 1인 개인

개발 기간 | 2023.05.16 ~ 2023.10.20 (2개월 공부, 3개월 제작)

개발 도구 | Visual Studio 2017, C++, WinAPI, DirectX11
Adobe PhotoShop, Git

외부 라이브러리 | FMOD, FW1FontWrapper, DirectXTex

플레이 영상 | <https://www.youtube.com/watch?v=XqzxtB7Yp88>



영상 QR

A*

Tile 연산

그림자

다중 빛 연산

Body Parts

Tetris Inventory

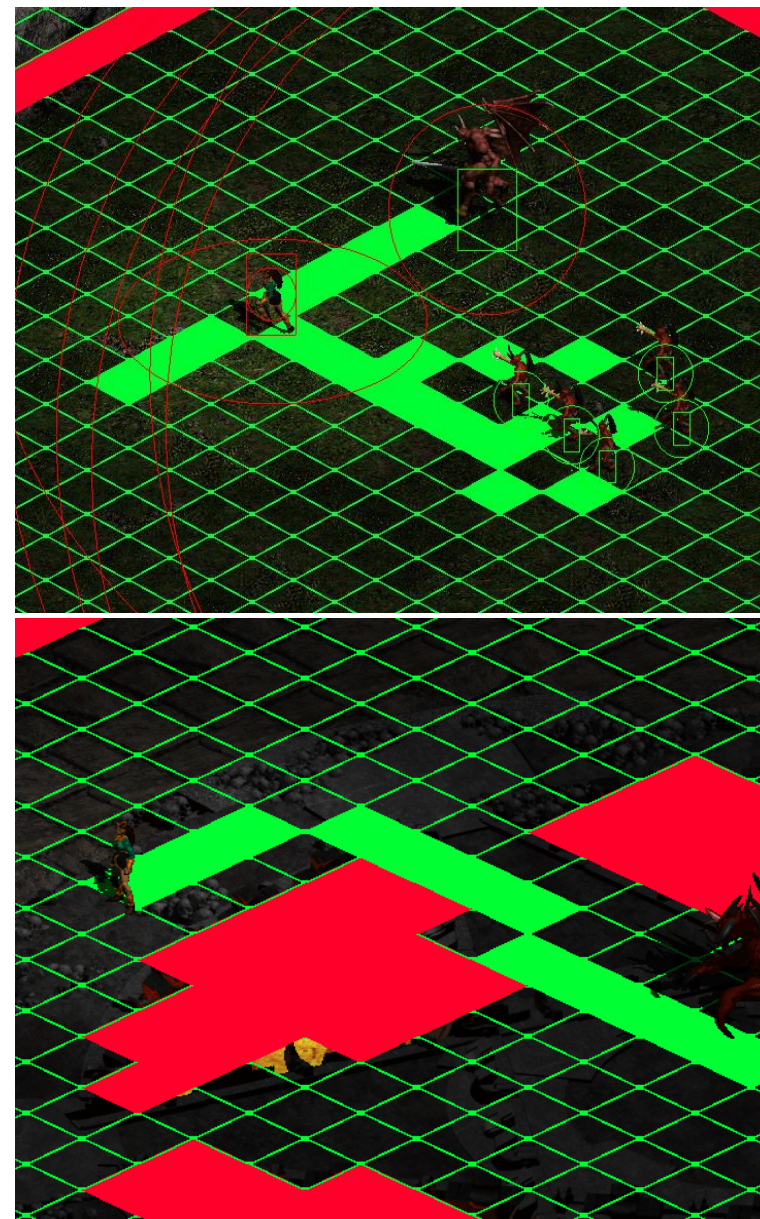
Skill 구현

Item

플레이어 캐릭터의 위치로부터 마우스가 Hover된 타일의 위치까지의 **최단거리를 A* 알고리즘으로 탐색**합니다.

몬스터는 현재 자신의 위치로부터 사거리에 들어온 플레이어의 위치까지의 최단거리를 A* 알고리즘으로 탐색합니다.

다른 몬스터들의 위치, 벽은 모든 위치탐색에서 제외됩니다.





Tile 연산

A*

Tile 연산

그림자

다중 빛 연산

Body Parts

Tetris Inventory

Skill 구현

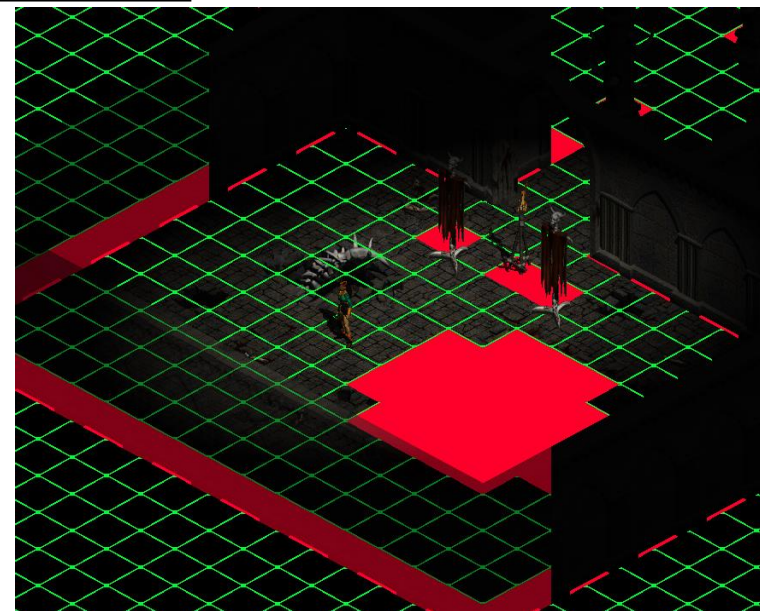
Item



영상 QR

프레임향상을 위해 Tile연산은 GPU에서 진행됩니다. 배열의 형태로 저장된 맵을 기반으로 **Geometry Shader로 Tile을 그려줍니다.** 그려진 타일들은 **Compute Shader에서 연산**되는데 마우스에 Hover되는 타일과 몬스터의 위치에 있는 타일, 마우스에 Hover되는 몬스터를 찾아냅니다.

스킬들의 위치도 Compute Shader로 넘어가 벽 충돌판정을 연산에 쓰입니다.





그림자

A*

Tile 연산

그림자

다중 빛 연산

Body Parts

Tetris Inventory

Skill 구현

Item

그림자를 그려야하는 플레이어, 몬스터, 장식물들은 ShadowObject를 가집니다. ShadowObject들은 자신을 가지는 Object의 Sprite를 받아와 대각선으로 기운 Mesh에 그려지게 됩니다.



영상 QR



다중 빛 연산

A*

Tile 연산

그림자

다중 빛 연산

Body Parts

Tetris Inventory

Skill 구현

Item



영상 QR

```
float lengthX = (position.x - lightsAttribute[idx].position.x) / lightsAttribute[idx].radiusX;  
float lengthY = (position.y - lightsAttribute[idx].position.y) / lightsAttribute[idx].radiusY;  
  
float distanceSquared = lengthX * lengthX + lengthY * lengthY;  
  
if (distanceSquared < 1.0f)  
{  
    float ratio = 1.0f - distanceSquared;  
  
    if (lightColor.a + lightsAttribute[idx].color.a * ratio < 1.f)  
        lightColor.a += lightsAttribute[idx].color.a * ratio;  
    else  
        lightColor.a = 1.f;  
  
    if (lightColor.g + lightsAttribute[idx].color.g * ratio < 1.f)  
        lightColor.g += lightsAttribute[idx].color.g * ratio;  
    else  
        lightColor.g = 1.f;  
  
    if (lightColor.b + lightsAttribute[idx].color.b * ratio < 1.f)  
        lightColor.b += lightsAttribute[idx].color.b * ratio;  
    else  
        lightColor.b = 1.f;  
  
    if (lightColor.r + lightsAttribute[idx].color.r * ratio < 1.f)  
        lightColor.r += lightsAttribute[idx].color.r * ratio;  
    else  
        lightColor.r = 1.f;  
}  
  
if (lightColor.r > 1.f)  
    lightColor.r = 1.f;  
  
if (lightColor.g > 1.f)  
    lightColor.g = 1.f;  
  
if (lightColor.b > 1.f)  
    lightColor.b = 1.f;  
  
if (lightColor.a > 1.f)  
    lightColor.a = 1.f;
```

타원의 빛이라 X, Y의 Radius가 다름

위치값과 빛의 사이거리 계산 후
반지름으로 정규화

길이 구하기

구해진 비율로 빛을 곱해 색에 더해주기

너무 밝은색이 나오지 않게
최종 나온 색이 1을 넘으면 1로





Body Parts

A*

Tile 연산

그림자

다중 빛 연산

Body Parts

Tetris Inventory

Skill 구현

Item

플레이어의 **파츠를 각각 불러와 아이템 착용**에 모습이 달라질 수 있도록 구현했습니다. 각각의 Body Parts는 각자의 Object로 만들어지고 소유하는 Object가 모두의 Animation Sync를 맞춥니다.



영상 QR



Tetris Inventory

A*

Tile 연산

그림자

다중 빛 연산

Body Parts

Tetris Inventory

Skill 구현

Item

디아블로의 아이템은 각자 인벤토리에서 크기가 다르고 크기에 따라 인벤토리를 차지하는 칸 수도 다릅니다. 각자의 사용하는 칸도 겹치면 안됩니다.

아이템의 위치, 크기와 인벤토리 각 칸의 위치, 크기를 비교해 **충돌체 없는 충돌처리**로 로직의 부하를 최대한 줄였습니다.



영상 QR



Skill 구현

A*

Tile 연산

그림자

다중 빛 연산

Body Parts

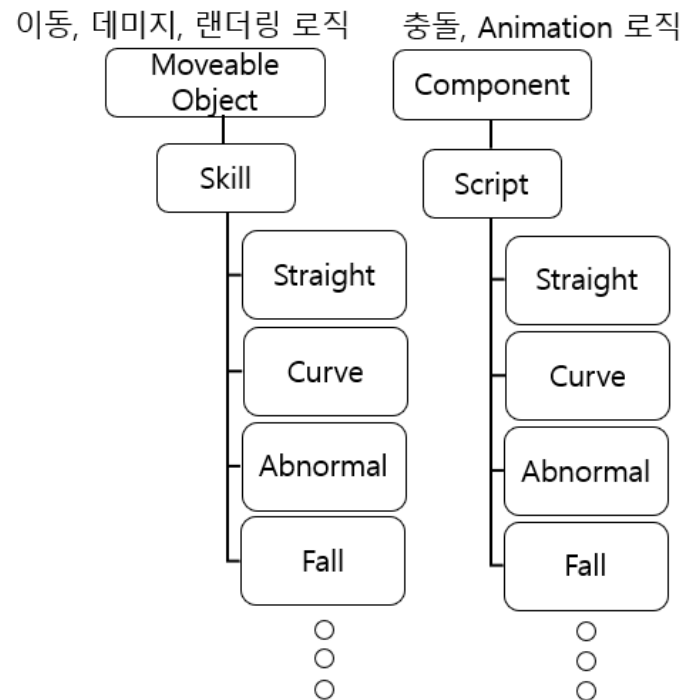
Tetris Inventory

Skill 구현

Item

소서리스의 모든 액티브 스킬만 22개이고 보스몬스터의 스킬 까지 합친다면 모두 27개입니다. 구현에 있어서 **모듈화와 상속**이 필수였습니다.

직선으로 나가다 터진 후 여러 곡선방향의 투사체가 필요하다면 따로 만들 필요없이 Straight를 상속받고 Straight Script 사용하여 Script에서 충돌처리할 때 Curve Skill 만드는데 가능한 구조를 만들었습니다.



영상 QR



Item

A*

Tile 연산

그림자

다중 빛 연산

Body Parts

Tetris Inventory

Skill 구현

Item



영상 QR

디아블로2에서 아이템이 이동할 수 있는 곳은 상점, 인벤토리, 장비칸, 물약칸, 필드로 총 5개가 있습니다. 5개를 자유롭게 왔다갔다 하면서 **그때마다 형식에 맞게 변해하는 구조**였습니다.

모든 **Item**은 **같은 부모를 상속받게** 하고 **static타입의 Manager**를 하나 두어 이동과 배치를 관리하게했습니다.

