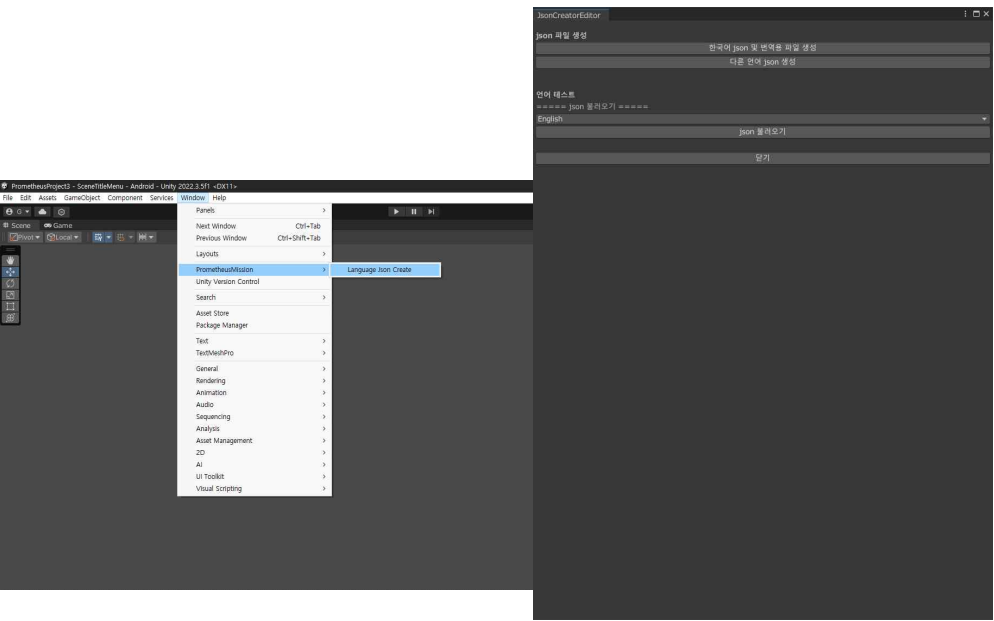


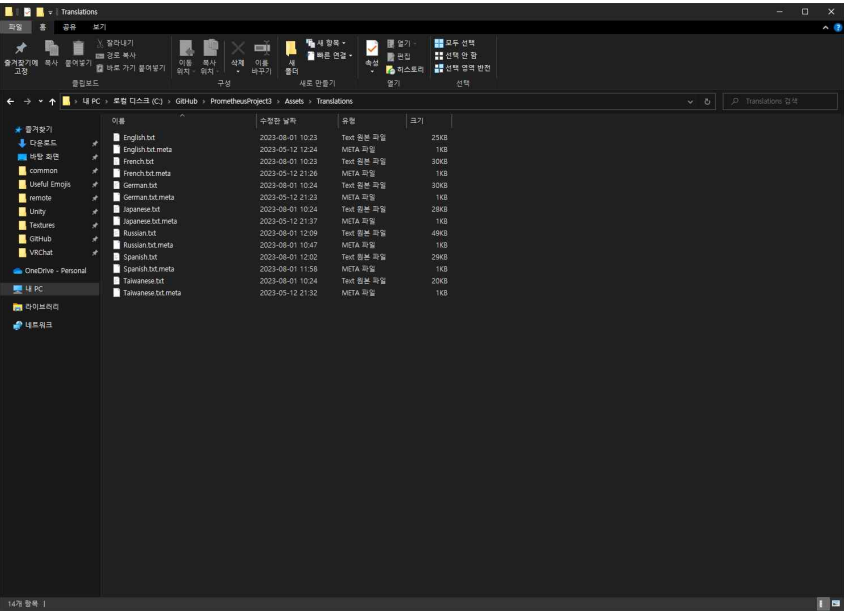
포트폴리오 설명문서  
분류: 시뮬레이션  
제목: 프로메테우스 미션



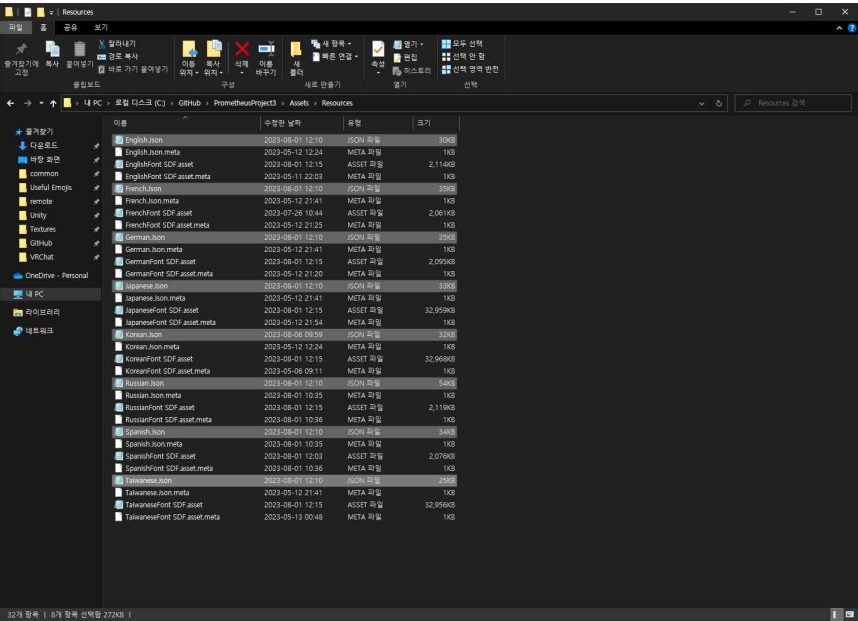
# 유니티 에디터 확장 기능



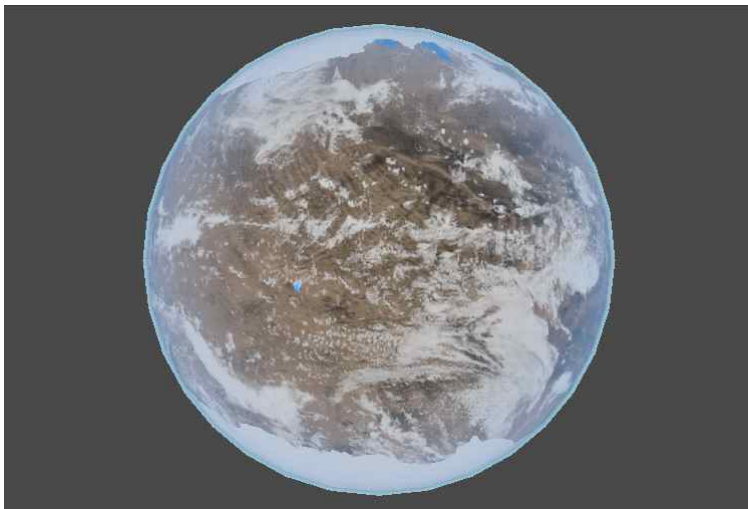
Json 파일 생성용 창을 소환하고 버튼을 눌러 파일 생성.



Translations 폴더의 텍스트 파일을 전부 읽어와서 Resources 폴더에 파일 이름은 그대로 json 형식으로 파일 생성.

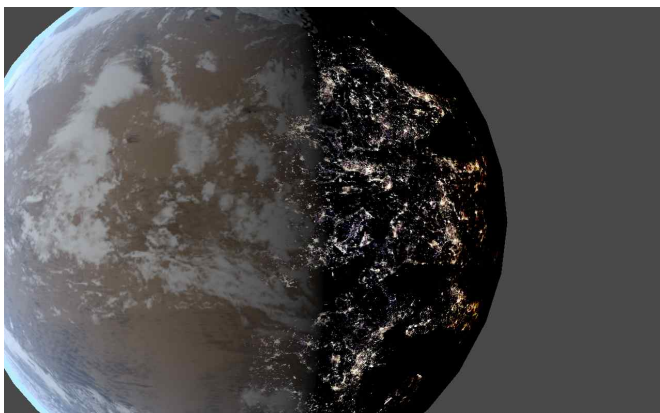


# Custom Shader 제작 - Built in



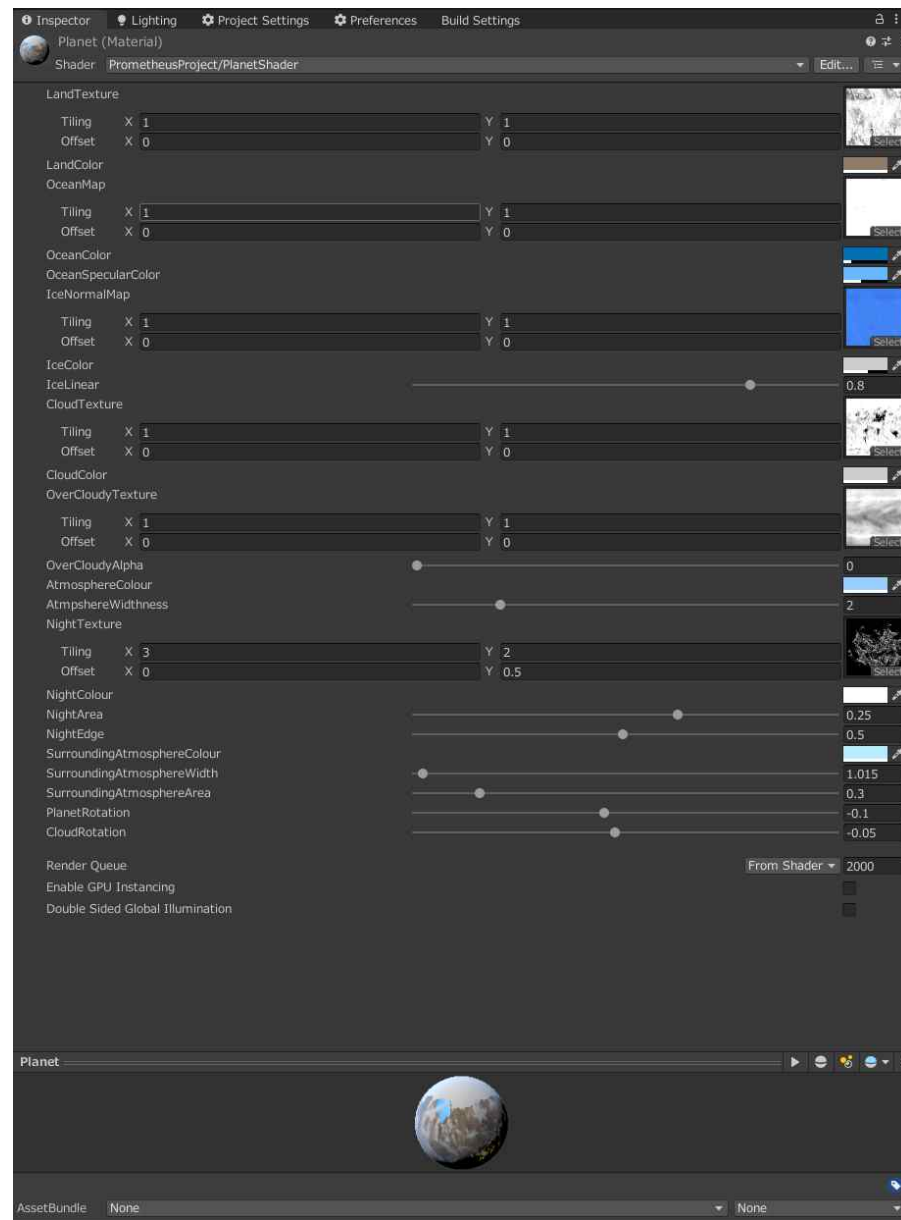
한 Sphere를 가지고 행성의 지표면, 해양, 빙하, 구름, 대기를 표현하기 위해 자체 셰이더를 제작.

행성 자전과 구름의 움직임은 UV 애니메이션으로 구현.



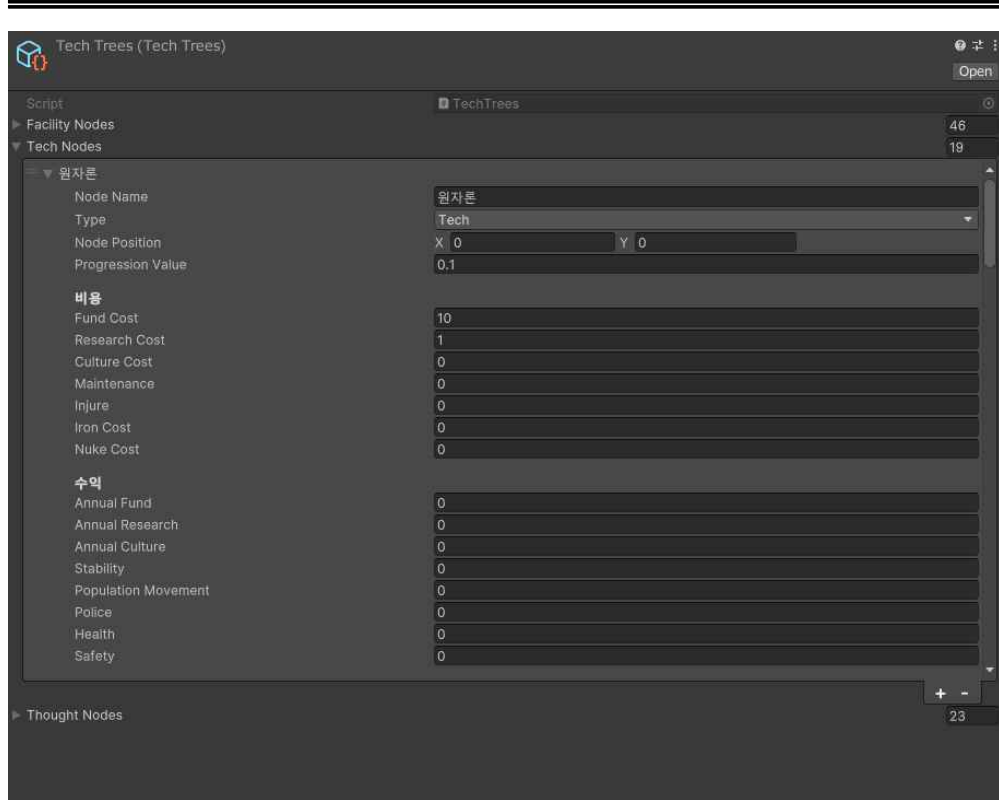
빛이 안 닿을 때의 야간 조명 표현.

가능한 현실적으로 보이기 위해 저녁 시간대에서부터 서서히 야간 조명 활성화.

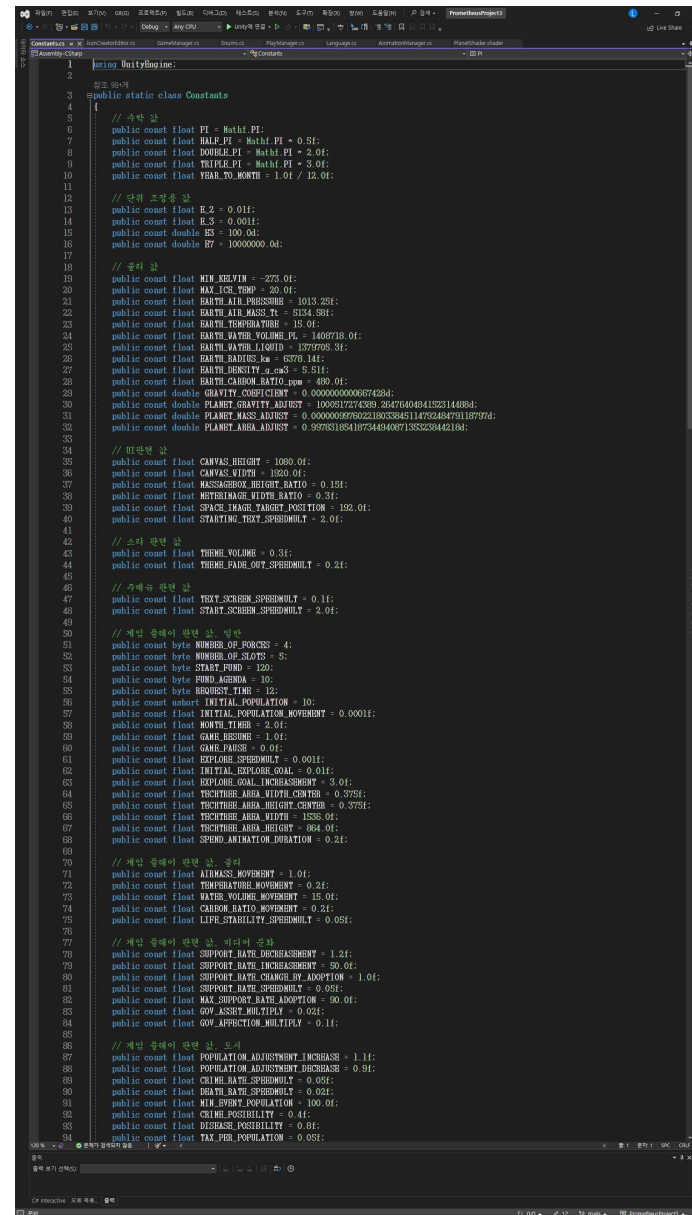


유니티 에디서에서 상세한 설정 가능.

# 편리한 정보 수정

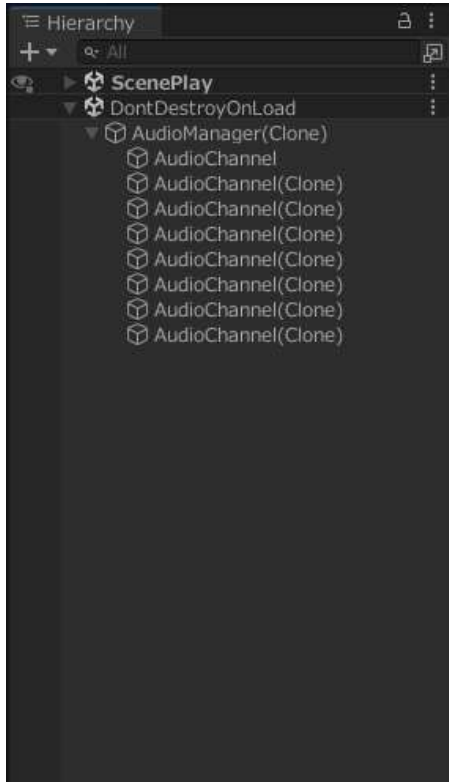


테크트리 정보처럼 자주 수정할 가능성이 높은 정보는 Scriptable Object로 만들어 유니티 에디터에서 누구나 편리하게 수정 가능.



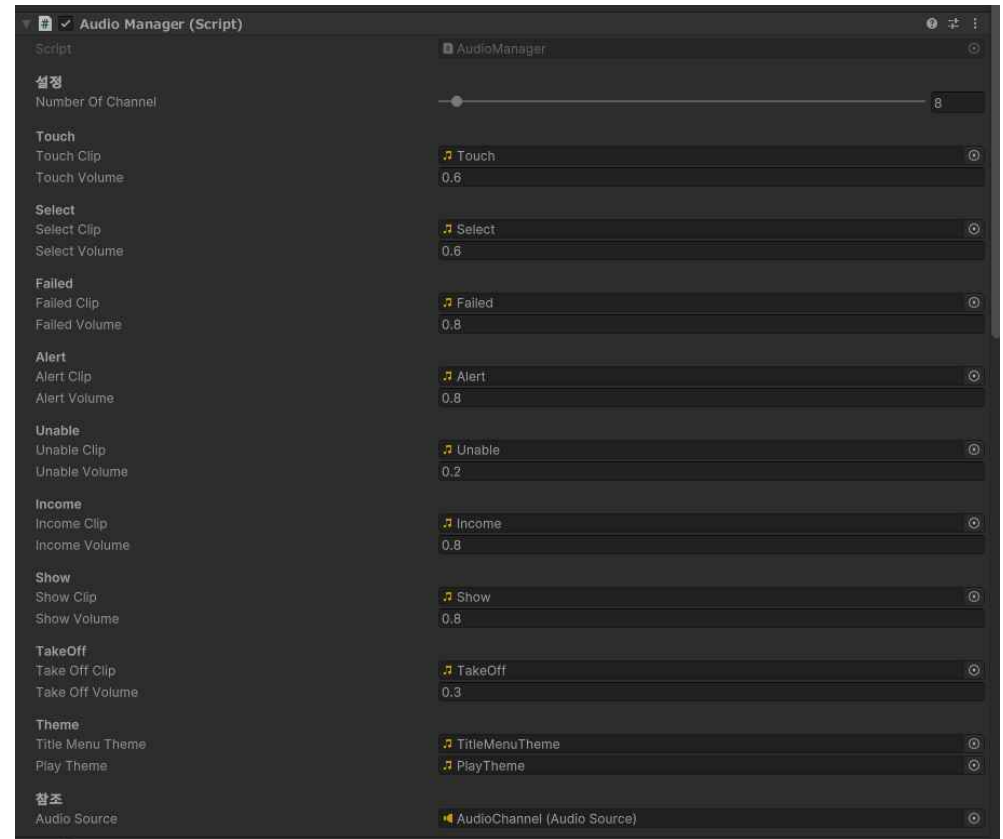
자주 쓰거나 중요한 상수를 모아둔 스크립트가 따로 존재함으로 관리가 용이.

# 다수의 오디오 채널



같은 소리를 연속으로 빠르게 재생하거나 소리 재생 중에 썸을 이동하는 경우 재생 중이던 소리가 끊어지는 현상이 발생하는데, 이를 방지하기 위해 여러 개의 소리 채널을 생성.

소리 재생을 위해 AudioManager에 요청하면 AudioManager는 각 채널에 순서대로 소리 클립을 할당하고 재생.

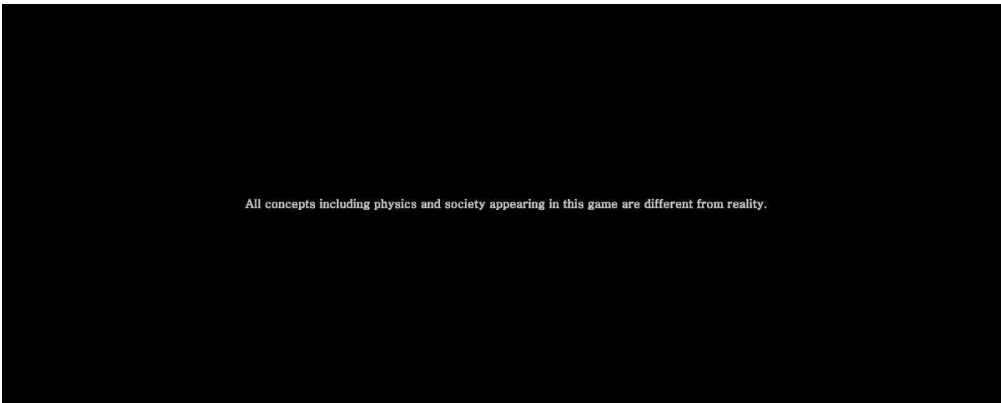


AudioManager는 Prefab으로 만들어져있고 인스펙터 창에서 소리 채널 개수 설정 가능.

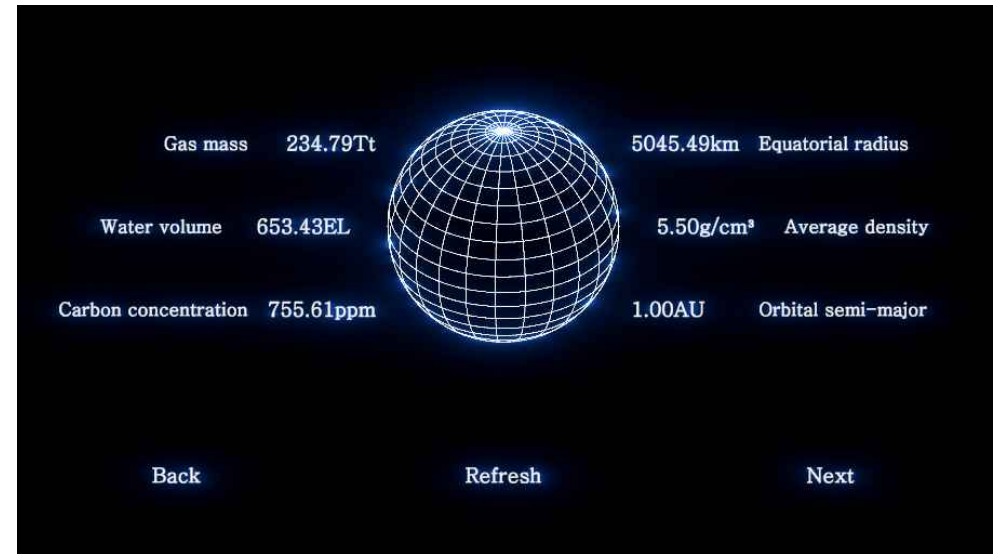
# 삼각함수를 이용한 애니메이션



반복적으로 밝아졌다 어두워지는 애니메이션에 삼각함수 이용.



안내문구가 밝아질 때 삼각함수로 부드럽게 밝아졌다가 일정 시간 그 밝기를 유지한 후 다시 삼각함수로 어두워짐.

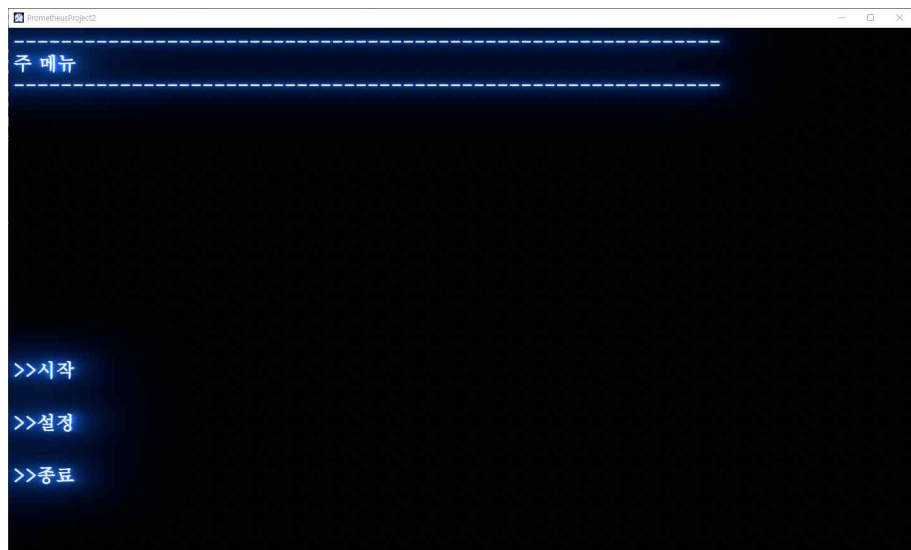


한 화면에 표시할 것이 많은 경우 보기 좋은 표시를 위해 순서에 맞춰 애니메이션 실행.

위 장면의 경우 가운데 구가 먼저 나타나고, 그다음 구의 양옆 텍스트, 하단의 버튼 순서로 나타남.

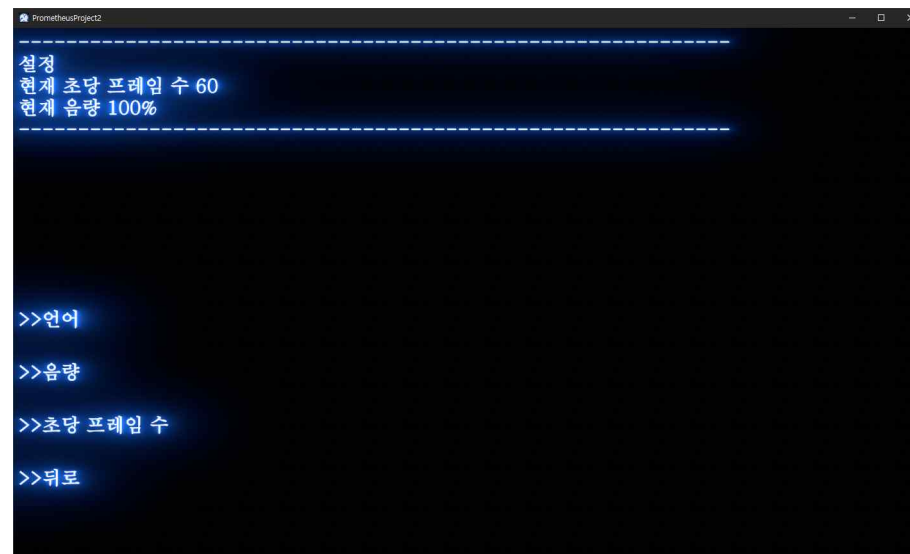
새로 고침한 경우 반대 순서로 사라졌다가 다시 원래 순서로 나타남.

# 주 화면



주 화면에서 메뉴를 이동할 때마다 텍스트가 위에서 한 줄씩 차례대로 나타나는 애니메이션 적용.

문자열을 자주 편집하기 때문에 StringBuilder를 사용.



설정 메뉴에서 언어, 전체 음량, 초당 프레임 수 설정 가능.

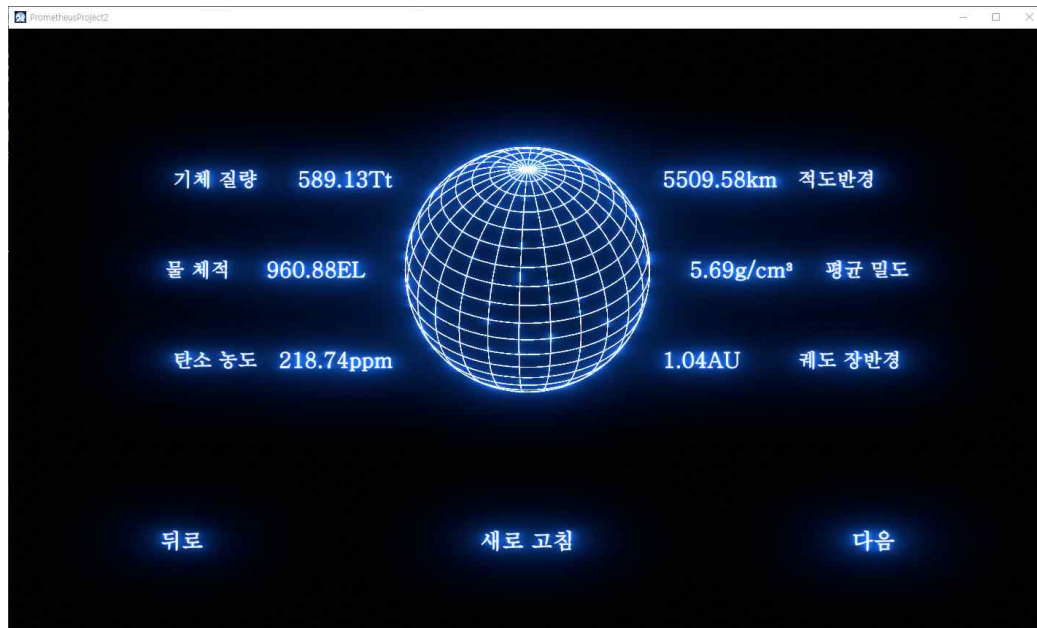
메뉴 화면은 상태 패턴을 적용.

MainMenuManager 클래스가 상태 기계 역할을 하고, 각 상태에 따라 MainMenuManager에서 다르게 동작하여 '다형성' 적용.

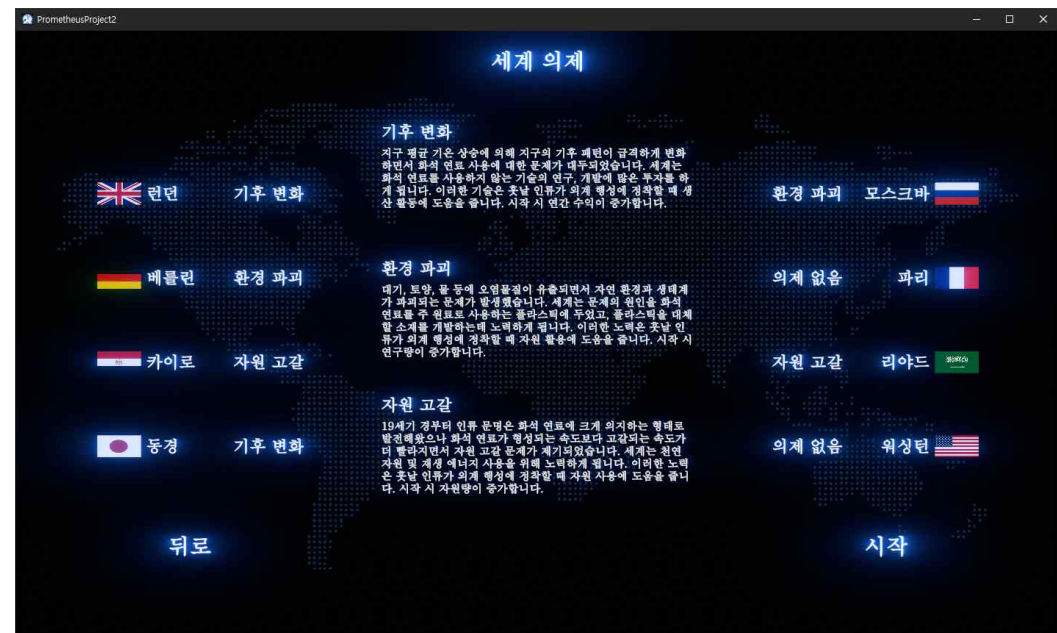


# 게임 시작 전

게임 시작 전에 행성 물리량이나 시작 부스트 값을 줄 수 있고, 이는 게임 난이도에 영향을 줍니다.



행성 물리량을 변경하는 경우 사용자가 직접 원하는 값을 입력하는 것이 아니라, 거주 가능한 외계행성을 찾겠다는 설정으로 전부 무작위 값으로 갱신.

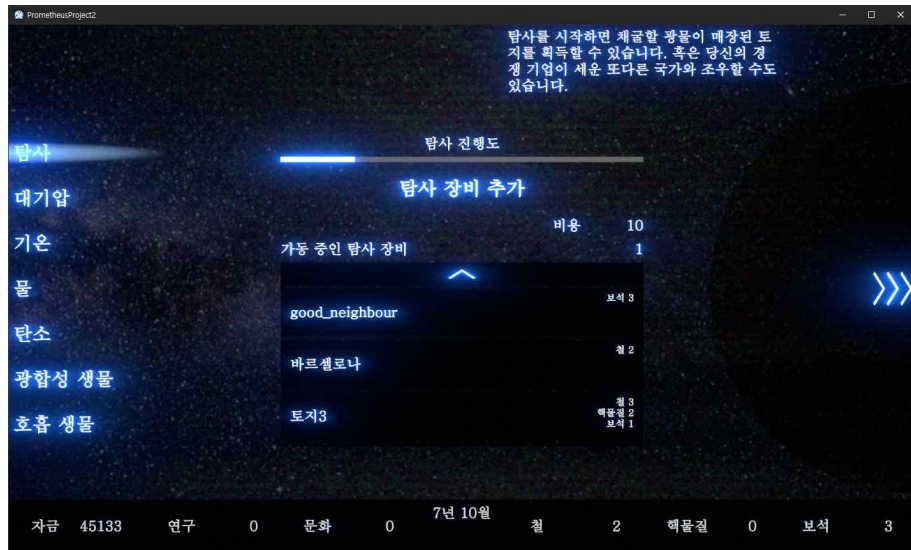


세계 의제는 플레이어가 처음에 가지는 자원량을 결정하는 것으로, 각 의제의 기여도에 따라 각기 다른 자원을 줍니다.

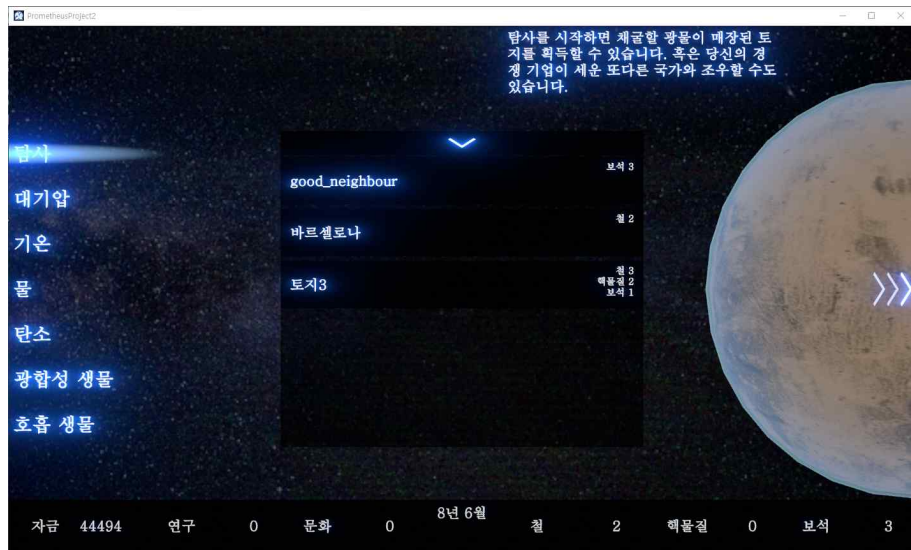
이는 플레이어가 지구 역사를 마음대로 바꿀 수 있다는 느낌이고, 그 어떤 의제에도 기여도가 전혀 없으면 난이도 대폭 상승.



# 목록 UI



플레이어의 행동으로 얻은 아이템은 목록에 표시.



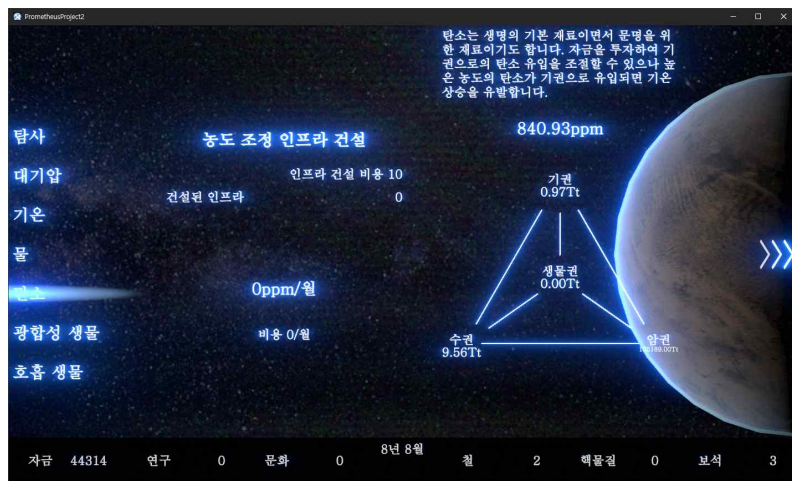
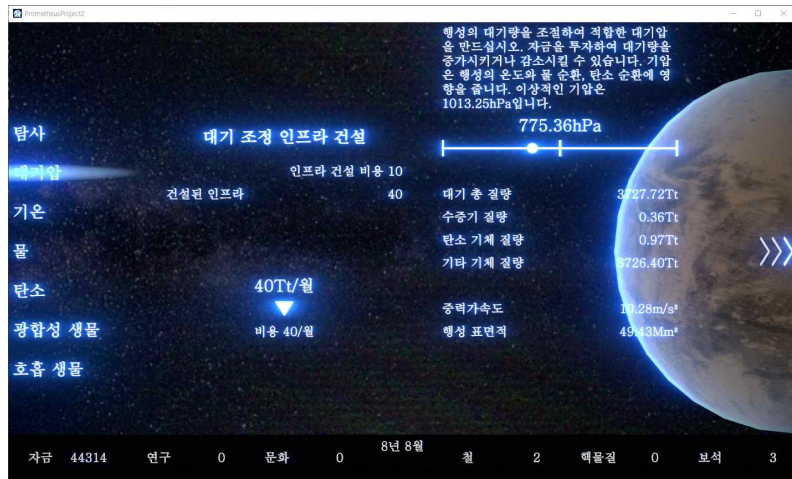
더 많은 아이템을 한꺼번에 보기 위해 목록을 확장할 수 있음.



아이템을 클릭하면 각각의 아이템 정보를 표시.

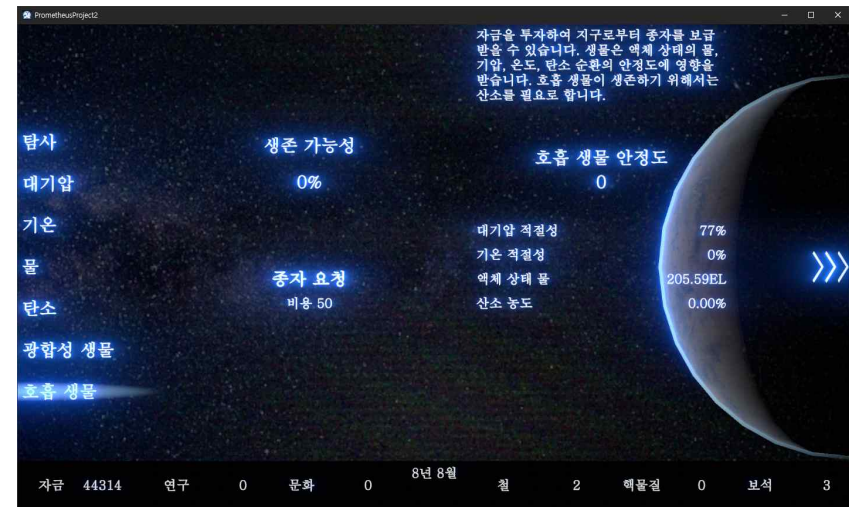
위처럼 배경이 어두워지고 특정 정보를 표시하는 화면이 나올 때 게임은 일시 정지 상태가 됨.

# 테라포밍



플레이어의 목표는 호흡 생물의 안정도를 50 이상으로 끌어올리는 것이고, 플레이어가 조절하는 행성의 물리량은 서로 영향을 줍니다.

플레이어는 물리량 수치를 실시간으로 확인할 수 있습니다.



물리량의 변화에 따라 배경에 있는 행성의 모습도 변화합니다.

# 문자열 중복 생성 방지

사용자에게 물리량 같은 정보를 표시할 때, 그 정보는 여러 화면에 반복해서 표시할 수 있으므로 표시할 문자열을 각 화면마다 따로 생성하는 것보다 한 문자열을 생성해서 그 문자열을 참조하는 것이 더 효율적일 것으로 생각해서 만든 클래스가 UString이라는 클래스.

UString은 공통으로 사용할 문자열을 관리.

```
public string this[VariableFloat variable]
{
    get
    {
        // 값이 바뀌었거나 문자열을 생성한 적 없을 때
        if (_currentFloatValues[(int)variable] != PlayManager.Instance[variable] || null == _floatStrings[(int)variable])
        {
            // 현재 값 저장
            _currentFloatValues[(int)variable] = PlayManager.Instance[variable];

            // 표시할 단위가 없을 때
            if (null == _floatUnits[(int)variable])
            {
                _floatStrings[(int)variable] = _currentFloatValues[(int)variable].ToString("F2");
            }
            // 표시할 단위가 있을 때
            else
            {
                _floatStrings[(int)variable] = $"{_currentFloatValues[(int)variable].ToString("F2")} {_floatUnits[(int)variable]}";
            }
        }

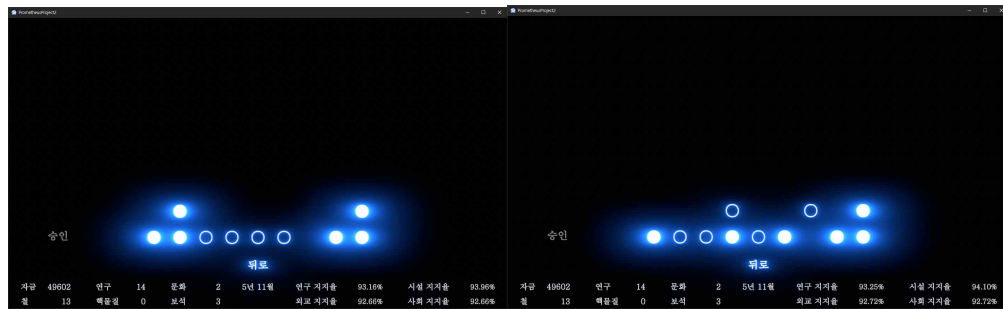
        // 반환
        return _floatStrings[(int)variable];
    }
}
```

UString 클래스에 접근하는 경우는 대부분 이 클래스가 관리하는 문자열을 참조하기 위함이기 때문에 해당 기능을 인덱서로 만들어서 접근이 편리하도록 함.

누군가가 어떤 문자열을 가져오고 싶을 때, UString 클래스는 해당 정보를 읽어온 다음 그 정보를 문자열로 만들어 반환하고, 읽어온 정보를 저장하고 있다가 다음 누군가가 그 문자열을 가져오고 싶을 때, UString은 그 정보 값이 변했는지 확인하고 변하지 않았으면 이전에 생성한 문자열을 반환함.



# 테크트리



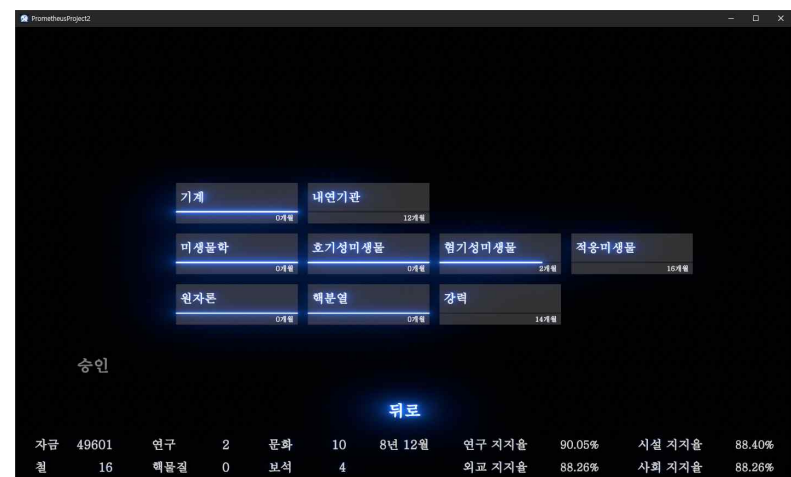
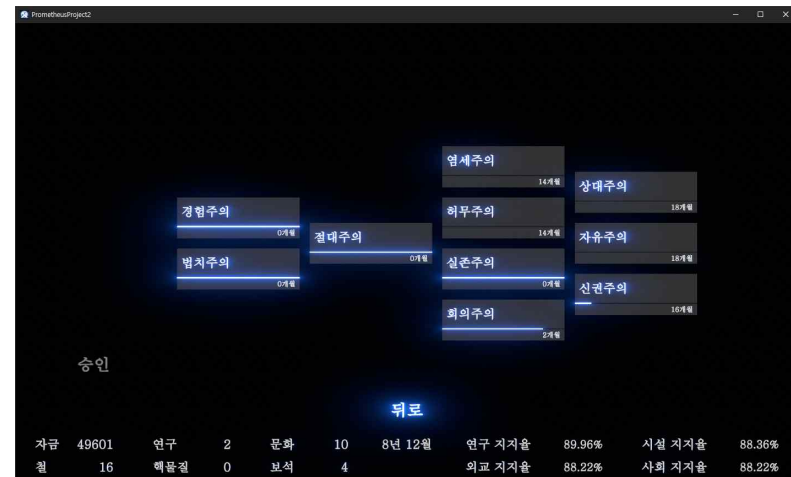
시설 테크트리는 도시마다 따로 적용.

플레이 스타일에 따라 각 도시별로 기능을 다르게 만들 수 있음.



연구 진행 상황 실시간으로 확인 가능.

동시에 여러 연구를 진행하는 경우 남은 개월수가 가장 적은 것을 우선으로 표시.



연구 진행에 따라 다음 테크 잠금 해제.

위 테크트리는 모두 같은 부모클래스를 상속받아 '상속성'을 적용.

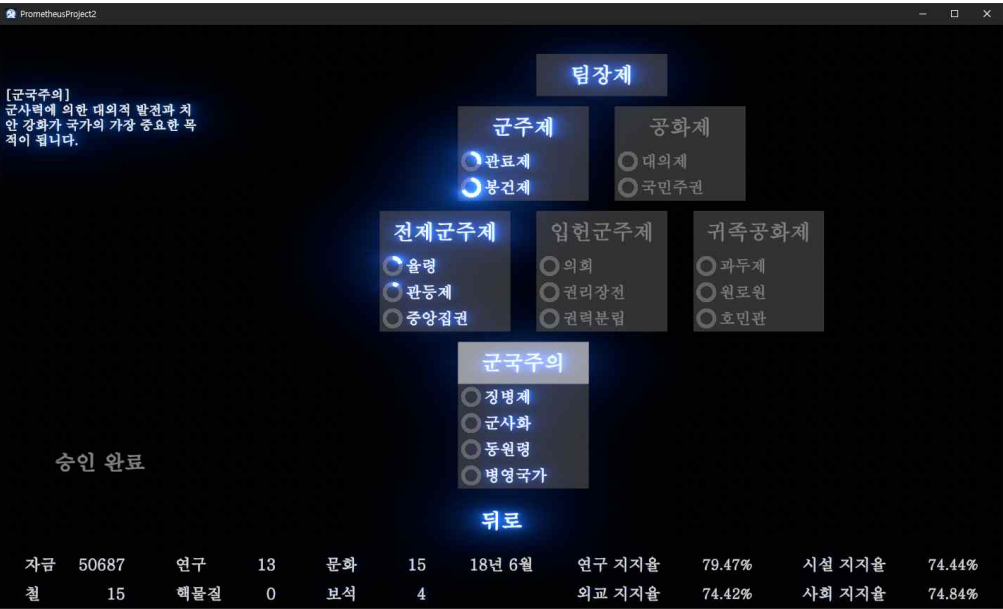
어떤 테크트리가 다른 테크트리의 잠금 해제 조건에 있기도 함.

테크트리 정보는 ScriptableObject로 만들어진 것을 참조.

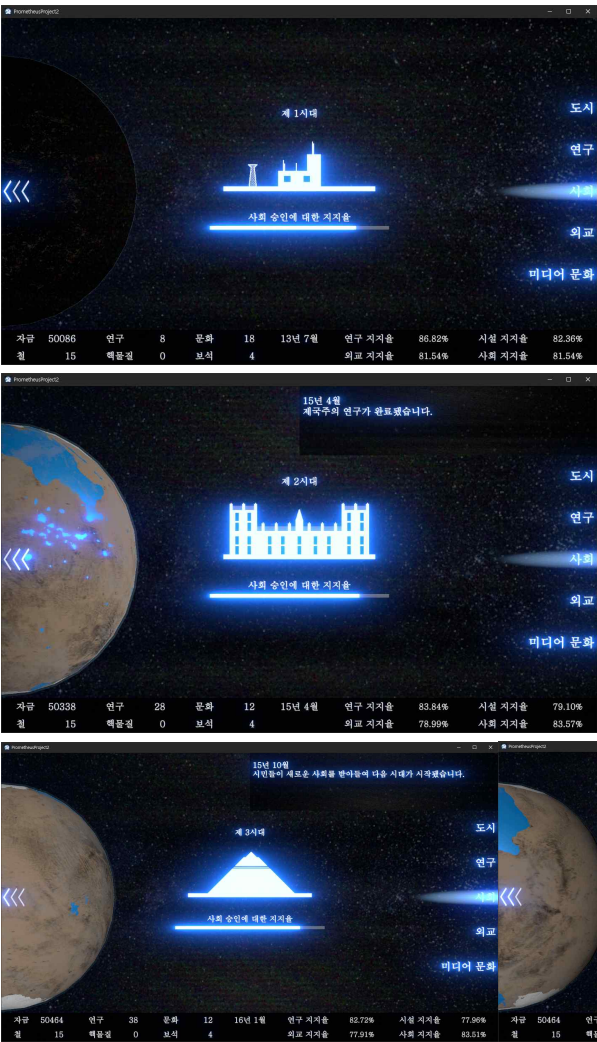
테크트리 노드 배치 또한 테크트리 정보에서 읽어온 대로 자동 배치.

# 사회 테크트리

사회 테크트리는 다른 테크트리와 상이해서 같은 부모클래스를 상속 받지 않음.

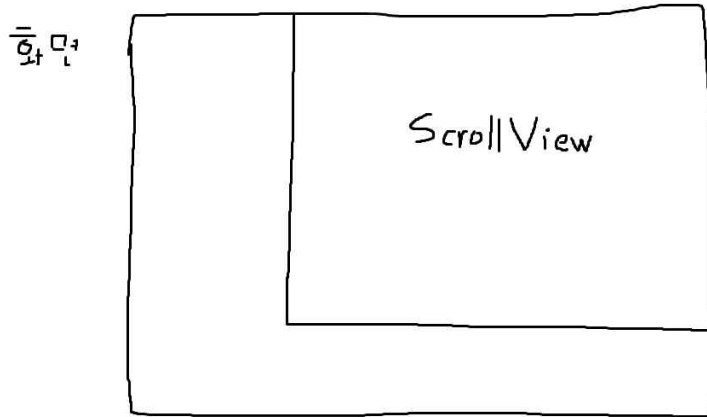


사상 연구에 따라 승인 가능한 사회가 잠금 해제됨.  
같은 높이에 있는 사회 중 하나를 승인했으면 다른 사회는 승인 불가.  
사회 승인 후 하위 정책을 승인할 수 있고, 하위 정책은 1회 승인 시 정책 진행도가 일부분만 상승.

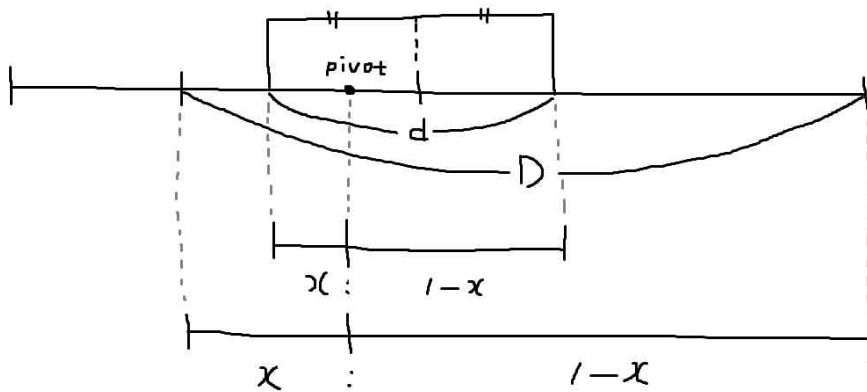


사회 승인할 때마다 정부 이미지 변경.

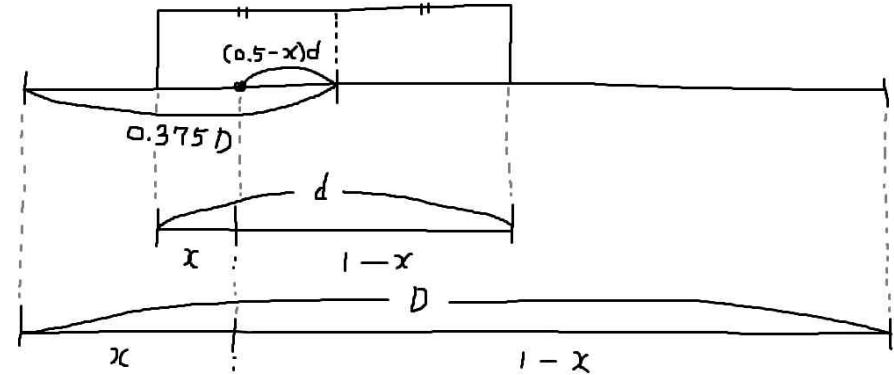
# 테크트리 노드 가운데 정렬



테크트리 노드는 ScrollView 안에 있고 ScrollView는 화면 한쪽으로 치우쳐져 있는데, 테크트리 노드가 화면상 정중앙에 위치하기 위해 Content의 pivot 조정.



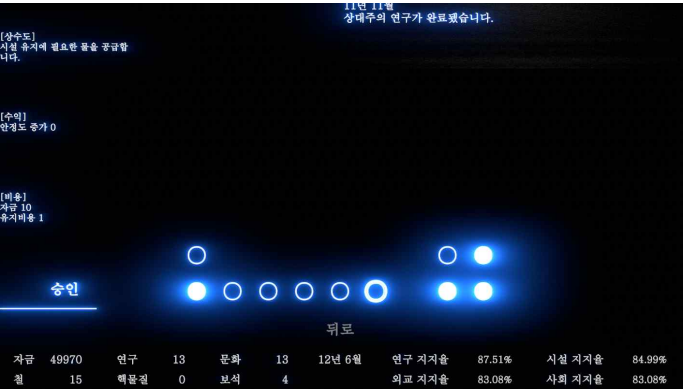
ScriptableObject로 만들어진 테크트리에 입력된 대로 테크트리 노드를 배치하면, 전체 노드의 위치, 길이에 따라 전체 Content의 길이를 계산하고, 전체 노드가 화면상 정중앙에 위치하기 위한 pivot 값을 계산함.



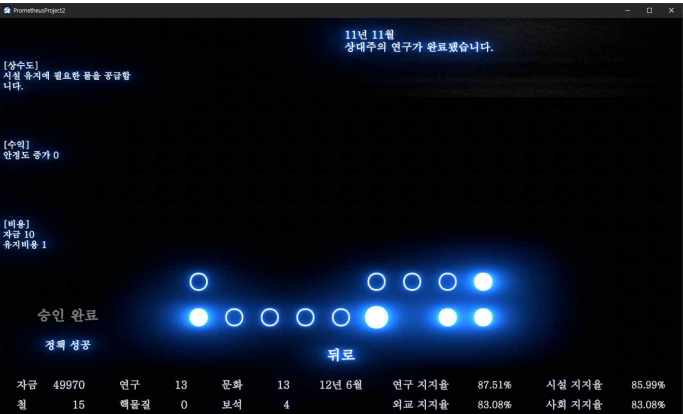
ScrollView의 위치, 크기 등 미리 정해진 값은 먼저 계산해서 상수로 만들어두고, 스크립트에서는 미지수만 계산함.



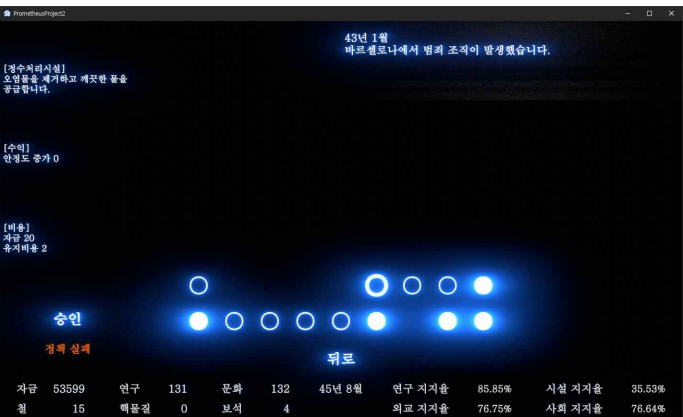
# 정책 성공과 실패



정책 승인 진행 중.



정책 성공.

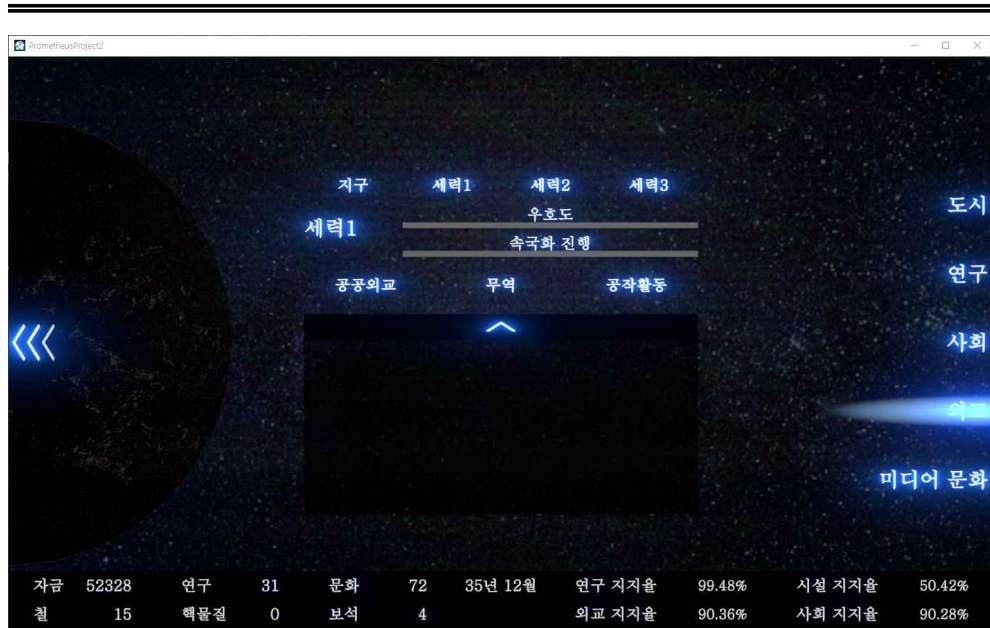


정책 실패.

시설, 연구, 사회, 외교 각 분야의 지지율에 따라 정책 성공 확률 결정.

정책 성공 시 일시적으로 지지율이 상승하고, 정책 실패 시 일시적으로 지지율 하락.

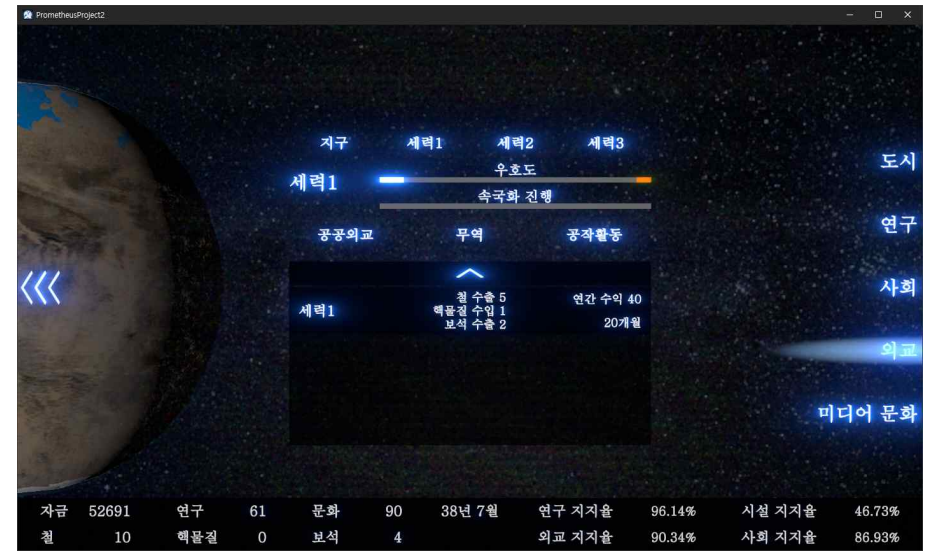
# 무역



외교 항목에서 외부 세력과 무역 가능.



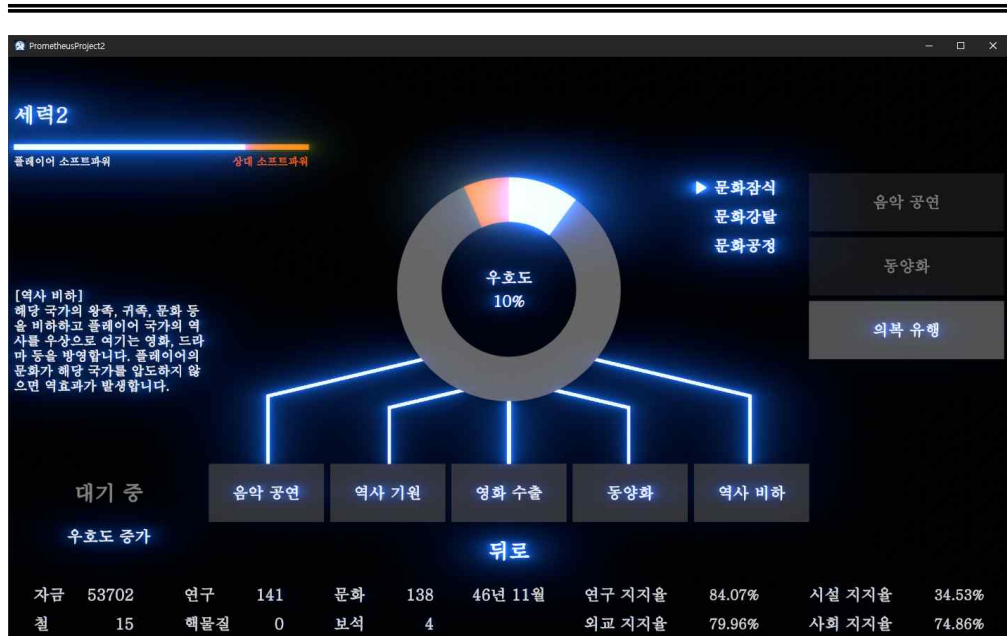
해당 세력의 우호도, 적대감 정도에 따라 수입, 수출 가격이 플레이 어에게 유리해지거나 불리해짐.



무역 시 무역 목록에 정보를 표시하고, 거래량만큼 서서히 우호도 증가.

일정 시간 후 거래 만료.

# 공공외교



공공외교는 소프트웨어의 영향을 받음.

소프트파워는 축적된 문화 양에 비례하고, 상대 세력보다 높을수록 유리.

공공외교를 위한 버튼은 재사용 대기시간이 있고, 대기 중일 때는 화면에 선으로 연결된 것으로 표시.



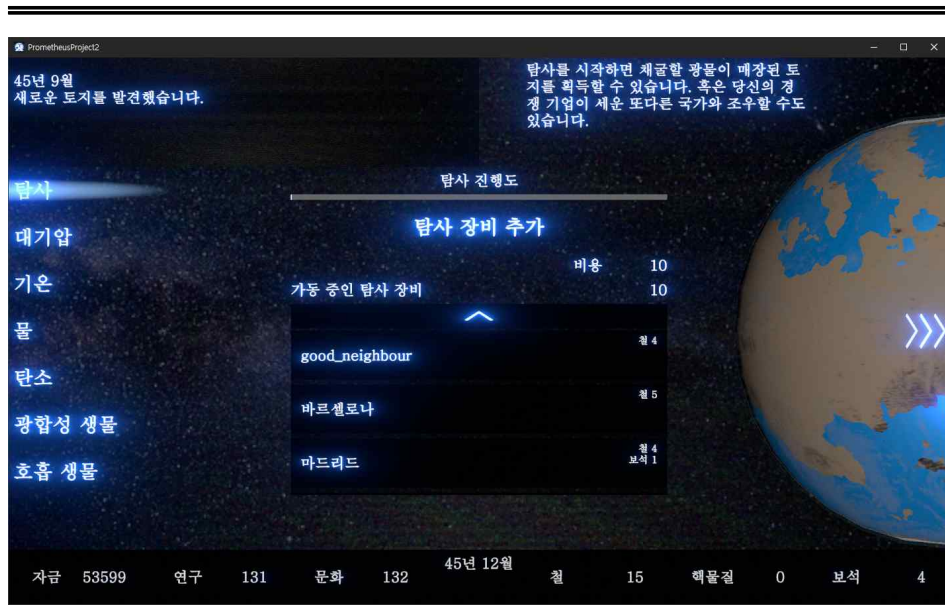
각 버튼은 대기시간이 다르고, 대기시간이 완료돼서 슬롯이 비었으면 좌측부터 빈 슬롯을 다시 채울 수 있음.

‘문화잠식’은 일정량만큼 우호도 증가.

‘문화강탈’은 플레이어의 소프트웨어가 상대를 압도할수록 우호도 증가.

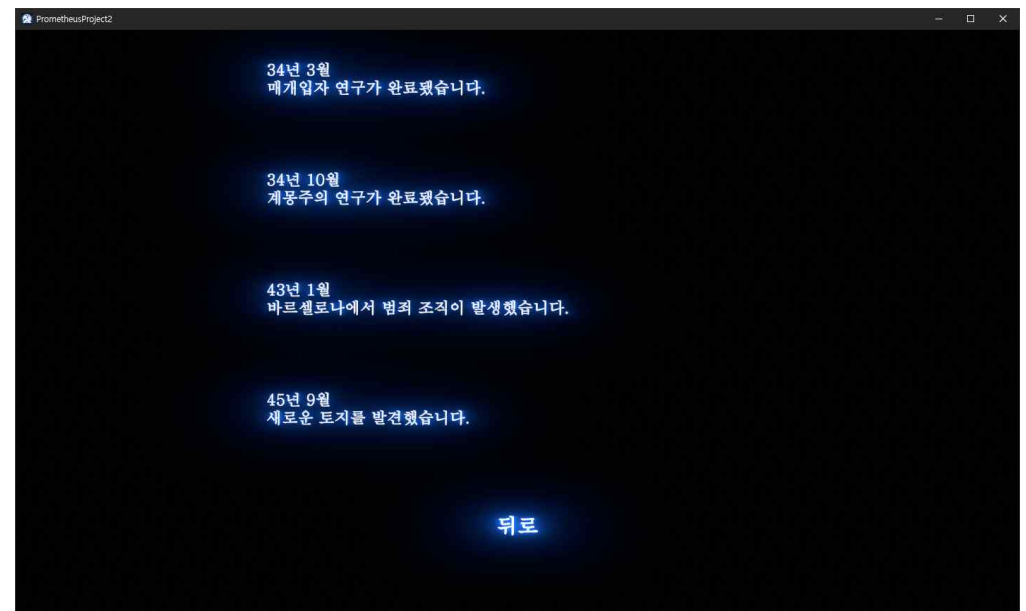
‘문화공정’은 플레이어의 소프트웨어가 상대를 압도하지 않으면 역효과.

# 알림 메시지



특정 사건이 발생하면 화면 상단에 메시지 창이 뜨면서 사건 날짜와 사건 내용 표시.

여러 사건이 동시에 발생할 경우, 플레이어가 메시지 창을 읽을 시간을 주기 위해 사건 메시지를 큐에 담아두고 일정 시간 간격으로 하나씩 표시한다.



메시지 기록은 언제든지 찾아볼 수 있음.

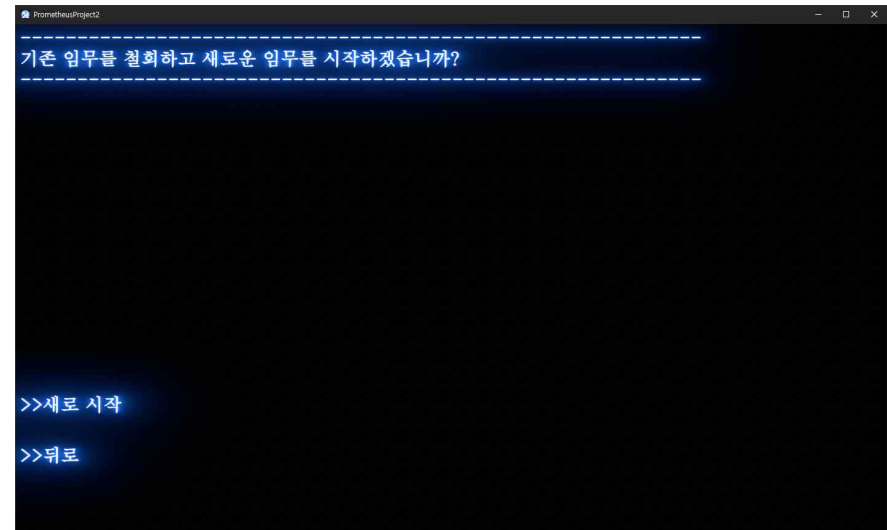
# 저장 및 불러오기

게임 진행 사항은 기본적으로 게임 시간으로 매년 저장되고, 주 화면으로 나가거나 게임을 종료하는 시점에 한 번 더 저장함.

따라서 어떤 식으로 게임을 종료하더라도 게임을 이어서할 수 있음.

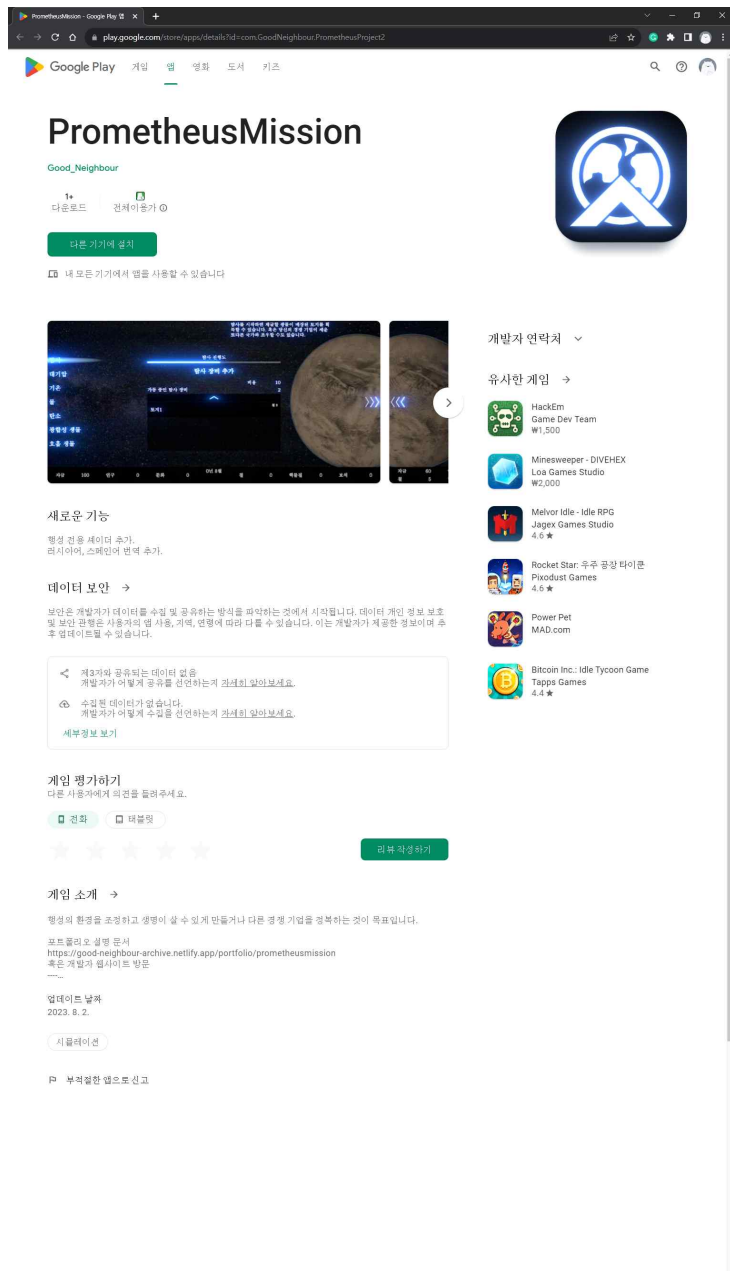


저장된 게임이 있을 경우 주 화면에서 불러오기 버튼을 볼 수 있음.



저장된 게임이 있는데 새 게임을 시작하려고 하면 저장된 게임을 포기할 것인지 묻는 화면을 볼 수 있음.

# 제품 출시 및 디버깅



다양한 사람들에게 게임 테스트를 요청할 목적으로 Google Play Store에 게임 출시.

이를 통해 테스트 요청 시 앱이 안전하다는 것을 증명할 수 있고, 무언가 업데이트가 있을 때마다 일일이 업데이트 버전을 보내줄 필요 없이 Google Play Store에서 업데이트 가능함.