2D 포트폴리오 계획서

제목 프로메테우스 프로젝트
분류 시뮬레이션
작성자 김호곤

목차

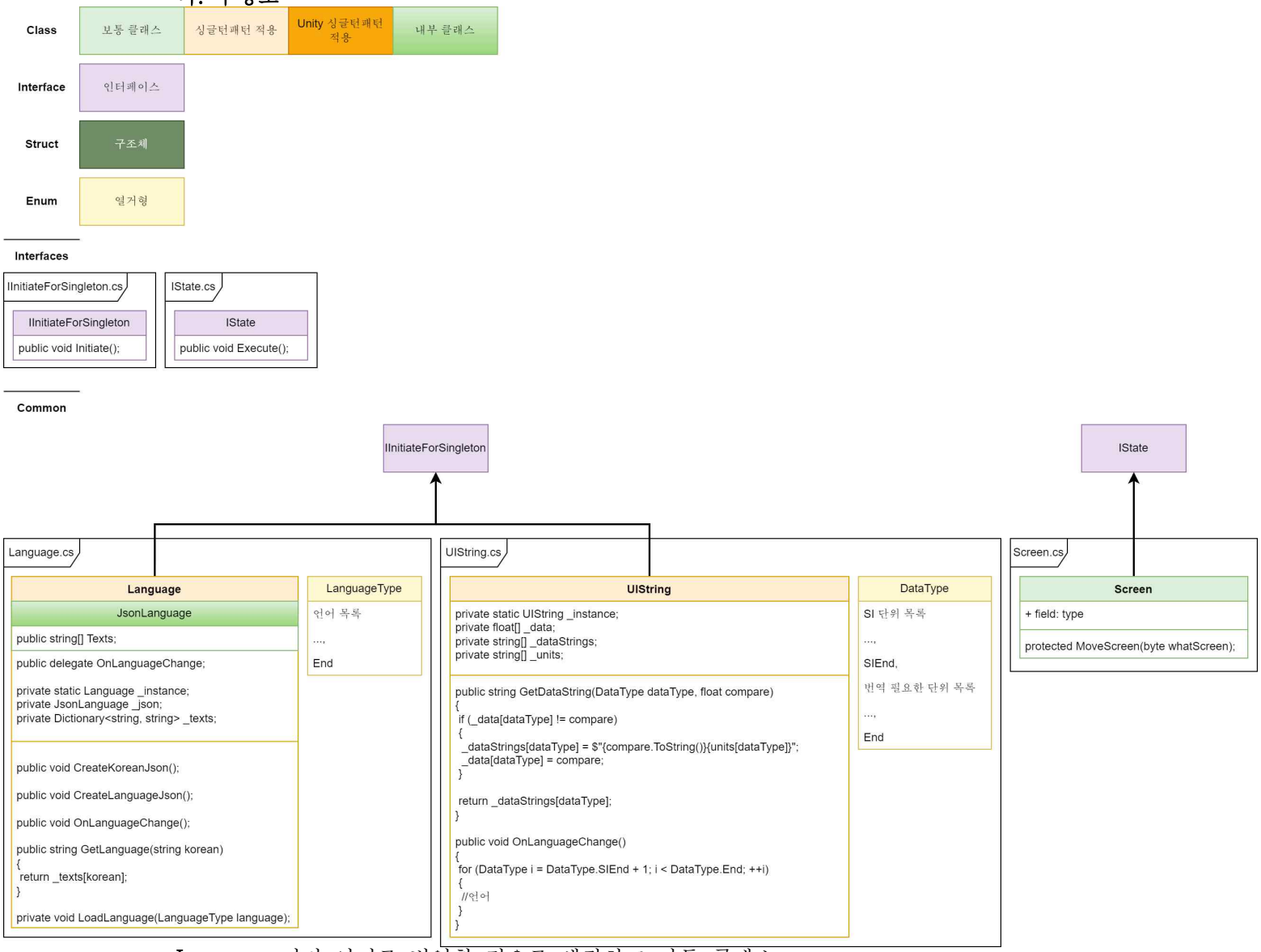
가. 개요

Unity를 기반으로 제작하는 테라포밍 시뮬레이션이다. 플레이어는 테라포밍 프로젝트를 총괄하는 한 우주 항공 기업의 AI 시점으로 플레이하는데, 승리 조건은 신 행성의 주권을 차지하는 것으로, 방법은 테라포밍을 완료하여 최다 공로를 인정받거나 다른 모든 경쟁 기업을 잠식하는 것이다.

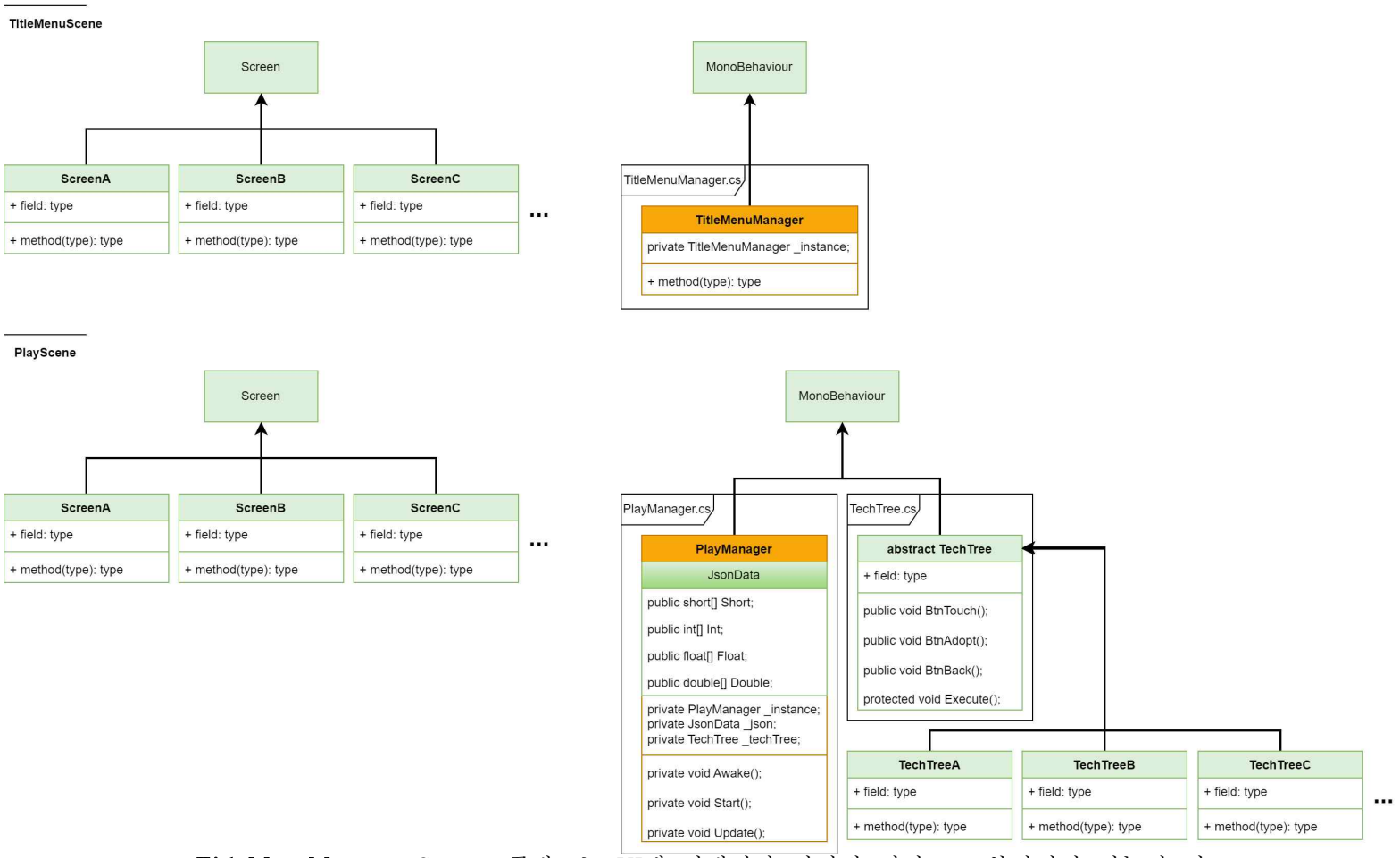
나. 개발환경

1. Unity 2021.3.24.f1
2. 비쥬얼스튜디오

다. 구상도



Language 여러 언어로 번역할 것으로 생각하고 만든 클래스.
UIString 행성 물리량 같은 정보를 UI에 표시할 때 동일한 정보를 여러 메뉴 화면에 보여줄 수 있는데, 그때마다 새로운 문자열을 생성하는 것보다 같은 문자열을 참조하는 것이 효율적일 것으로 생각하여 만든 클래스.
Screen 모든 메뉴 화면의 부모 클래스.



TitleMenuManager Screen 클래스는 UI에 관해서만 작성될 것이므로 부가적인 기능이 필요하면 TitleMenuManager에서 수행한다.
PlayManager TitleMenuManager와 역할은 비슷하다.
TechTree 각 TechTree는 추상클래스를 상속받는다.

라. 계획

1. 프로그래밍 계획

가) UI에 표시할 언어는 Language 클래스에서 참조해온다.

여러 언어로 번역할 것을 상정하기 때문에, 단어나 문장을 표시할 때마다 분기분을 만들 수 없다. 단어 및 문장은 모두 json 파일에서 읽어오고 Dictionary라는 해쉬테이블에 저장될 것이다. 한국어로 된 '키'로 접근하면 번역된 '값'이 반환되는 형태다.

나) json 파일은 Editor 스크립트를 통해 생성한다.

게임 실행 중에 json 파일을 생성할 일은 없으므로 이 기능은 Unity 에디터에서만 수행한다. Editor 스크립트를 만들어 인스펙터 창에 버튼을 만들면 간편하게 json 파일을 생성할 수 있을 것이다. Unity가 아니었으면 콘솔 프로그램을 따로 만들어서 수행했을 것이었다.

다) UI에 정보를 표시할 때 UIString 클래스에서 참조해온다.

행성 물리량이나 인구수 등과 같은 정보는 여러 메뉴 화면에 반복해서 표시할 수 있다. 그때마다 각각의 문자열을 생성하기보다 한 문자열을 참조하는 것이 좀 더 효율적일 것이다. 그러므로 UIString 클래스는 공용으로 사용할 정보 문자열을 관리한다.

라) 열거형에는 끝을 표시한다.

배열을 인덱스 번호로 접근할 때 정수형 숫자로 접근하는 것보다 열거형 숫자로 접근하는 것이 의미를 알기 좋을 것이다. 열거형은 편집할수록 가장 처음 요소나 가장 마지막 요소가 변경됐을 가능성이 높은데, 이런 경우 열거형을 사용한 for문을 지속적으로 수정해야 될 수도 있다. 그래서 수정이 불필요하도록 End같은 이름의 요소를 마지막에 추가하고, 열거형을 수정하더라도 End 요소를 항상 마지막으로 고정한다.

마) 상태패턴을 위한 인터페이스를 만든다.

현재 계획 상으로는 상태패턴을 메뉴 화면에만 사용하고 있는데, 개발 중에 상태패턴을 또 사용할 곳이 생기는 것을 대비해 인터페이스로 만들었다. C#에서 클래스는 다중 상속이 불가능하므로 인터페이스로 만드는 것이 더 보기 좋을 것이다.

바) 자료형은 가능하면 메모리를 아끼도록 설정한다.

게임 주제 상 많은 변수를 사용하게 될 것이고 또한 천문학적인 숫자를 다루게 되므로 long이나 double같은 큰 변수를 사용할 수 있기 때문에 메모리를 아낄 필요가 있다. 물론 이 프로젝트는 PC로 빌드될 것이나, 모바일 환경에서 빌드될 것으로 상정하면 메모리 절약이 필요할 수도 있다. 따라서 255 미만은 byte, 3만 미만은 short, 21억 미만은 int를 사용하고, 부호가 필요하지 않으면 unsigned 자료형을 이용해서 더 작은 자료형을 사용할 여지를 늘릴 수 있을 것이다.

사) JsonData에서는 자료형 별로 배열을 선언한다.

게임 주제 상 변수를 많이 사용할 것이다. JsonData 클래스는 josn 파일을 위해 데이터를 저장만 할 것인데, 변수 이름을 일일이 구분해주는 것은 불필요하다. 따라서 JsonData는 데이터를 배열의 형태로 일괄 저장하고, 대신 JsonData의 데이터에 접근할 때는 PlayManager의 프로퍼티로 접근한다. 각 프로퍼티는 모두 고유의 이름을 가지고 JsonData가 가지고 있는 배열의 인덱스 중 하나에 접근한다. 따라서 JsonData는 모든 변수를 구분할 필요 없고, 그 외 다른 클래스는 PlayManager의 프로퍼티 구분 덕분에 원하는 인덱스를 헷갈리지 않아도 된다.

아) TechTree는 부모클래스만 참조함으로 다형성을 시도한다.

모든 TechTree는 추상클래스를 상속받는다. 한 번에 두 개 이상의 TechTree와 상호작용할 경우는 없을 것이므로 PlayManager는 부모클래스 형식의 참조변수만 가지고 그 변수가 참조하는 자식클래스만 달리 할 것이다. 따라서 PlayManager는 같은 동작을 할 것이나 결과는 참조하는 클래스에 따라서 달라질 것이다. 참조할 클래스를 변경하는 것은 Screen에 해당하는 클래스가 담당할 것이다.

라) 같은 타이밍에 수행할 동작은 대리자에 등록한다.

언어 설정을 변경하는 것처럼 한 번의 동작에 여러 가지가 변화해야 되는 경우, 그 동작은 모두 한 대리자에 등록한다. 그러면 언어 설정을 변경하는 동작 하나에서 대리자가 등록된 모든 동작을 수행할 것이다.

2. 컨셉 계획

가) 물리

아래 물리량은 모두 지구 기준이다.

1) 대기압

-공식

대기압 = 기체질량 * 가속도 / 면적

기체질량 = 수증기 + 탄소 + 기타
가속도 = (중력상수 * 행성질량) / 적도반경^2 * 가속도보정
행성질량 = 4 / 3 * pi * 적도반경^3 * 밀도 * 질량보정
면적 = 4 * pi * 적도반경^2 * 면적보정
(적도반경은 가장 긴 반경)

-실제 물리량

중력상수 = 6.67428 * 10^-11
지구 중력가속도 = 9.80665m/s^2
지구 적도반경 = 6378.14km
지구 질량 = 5974.2Zt
지구 밀도 = 5.51g/cm^3
지구 면적 = 510100000km^2
지구 대기 전체 = 5148.0Tt
지구 대기 수증기 = 12.7Tt
지구 대기 탄소 = 0.72Tt
지구 대기 기타 = 5134.58Tt

-계산

지구 기준 가속도보정 = 1.000517365620376063897459786836 4
지구 기준 질량보정 = 0.997602218033845114792484791187 97
지구 기준 면적보정 = 0.997831854187344940871353238442 18

2) 온도

-지어낸 공식

온도 = -273
+ 입사에너지 * 240
+ 흡수에너지 * 15.797788309636650868878357030016
+ 수증기 온실 + 탄소 온실 + 기타 온실
(입사에너지, 흡수에너지는 0에서 1 사이라고 가정)

흡수에너지 = 입사에너지 * (1 - 반사율)
입사에너지 = 거리비율^2
거리비율 = 1(AU) / 거리(AU)
반사율 = 지각 반사율 * 0.10 + 물 반사율 * 0.035
+ 빙하 반사율 * 0.90 + 구름 반사율 * 0.35
구름 반사율 = 기체 상태의 물의 체적 / 12.7 * 0.25 / 0.35
빙하 반사율 = (1 - 구름) * 고체 상태의 물의 체적
* 1.340996168582375478927203065134 1e-5
물 반사율 = (1 - 구름 - 빙하) * 액체 상태의 물의 체적
* 2.939717674451244004104314200 9695e-8
지각 반사율 = 1 - 구름 - 빙하 - 물

수증기 온실 = 24.06923987 * log(기체 상태의 물의 질량 + 1)
탄소 온실 = 20.97411189 * log(기체 상태의 탄소의 질량 + 1)
기타 온실 = 1.53162266 * log(대기 기타 질량 + 1)
(평형 유지를 위해 log 함수 선택)

-실제 물리량

구름 반사율 35%
빙하 반사율 90%
물 반사율 3.5%
지각 반사율 10%
지구 전체 중 구름에 의한 반사 11.7%
지구 전체 중 빙하에 의한 반사 10%
지구 전체 중 바다, 강, 호수에 의한 반사 0.7845%
지구 전체 중 지각에 의한 반사 0.9155%

3) 물의 체적

-지어낸 공식

기체 비율 = 8.326994938358479249807994927615 1e-7
* 1.063078158938699235618459045898 4^(온도
+ 23.94160508)

고체 비율 = (1 - 기체) * 0.026455369380107815338212253336 51
* log(21 - 온도)

액체 비율 = 100% - 고체 - 기체

-실제 물리량

지구 물의 체적 1408.718EL

지구 액체 물 체적 1379.7053EL

지구 고체 물 체적 29EL

지구 기체 물 체적 0.0127EL

포화수증기량

10℃ - 7.6g/kg

20℃ - 14.7g/kg

30℃ - 27.1g/kg

4) 탄소 순환

탄소 순환은 제작 중에 구현 대상에서 제외될 수 있다. 플레이어가 제어할 수 있는 요소가 아니라면 구현하지 않는 것이 차라리 났다.

-지어낸 공식

기권 = 기압 / 1013.25 * 0.72 + 대기오염

수권 = 물 액체 / 1408.718EL * 37.4

생물권 = 생물 * (적도반경 / 6378.14)^2 * 2

-실제 물리량

지구 기권 탄소 질량 = 0.72Tt

지구 수권 탄소 질량 = 37.4Tt

지구 생물권 탄소 질량 = 2Tt

지구 지권 탄소 질량 = 60041.3Tt

5) 광합성 생물

-지어낸 공식

안정도 = 1 - |대기압 / 1013.25 - 1| * |온도 / 15 - 1| * (액체 / 1379.7053)

(안정도는 0에서 1까지로 가정)

6) 호흡 생물

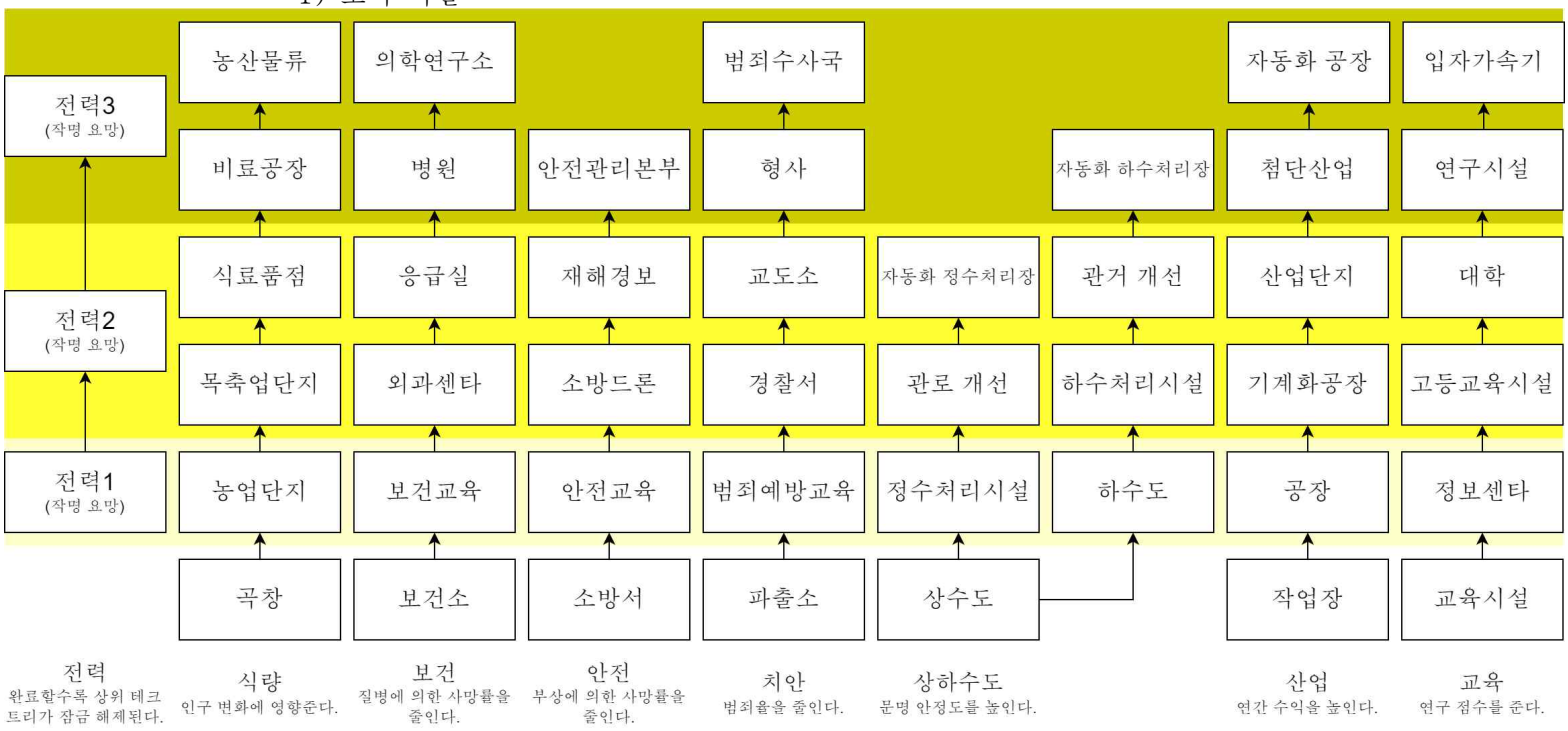
-지어낸 공식

안정도 = 1 - |대기압 / 1013.25 - 1| * |온도 / 15 - 1| * |산소 / 21 - 1| * (액체 / 1379.7053)

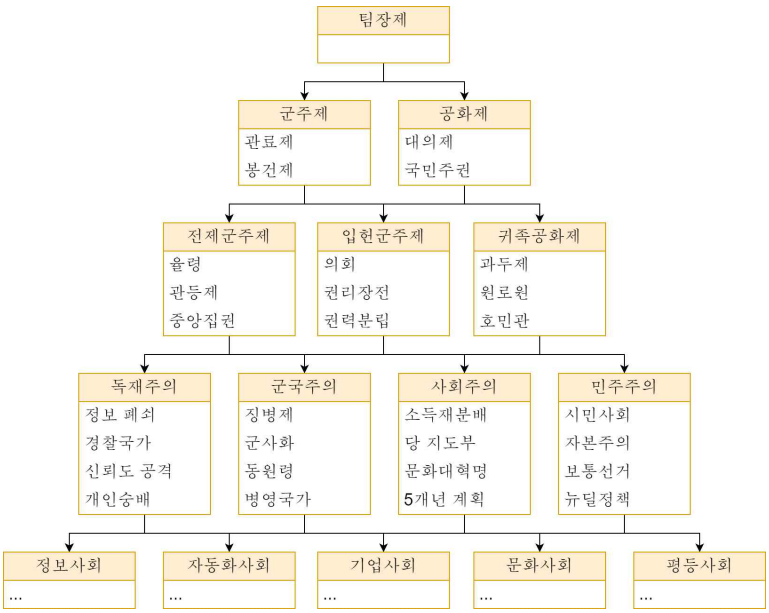
(안정도는 0에서 1까지로 가정)

나) 문명

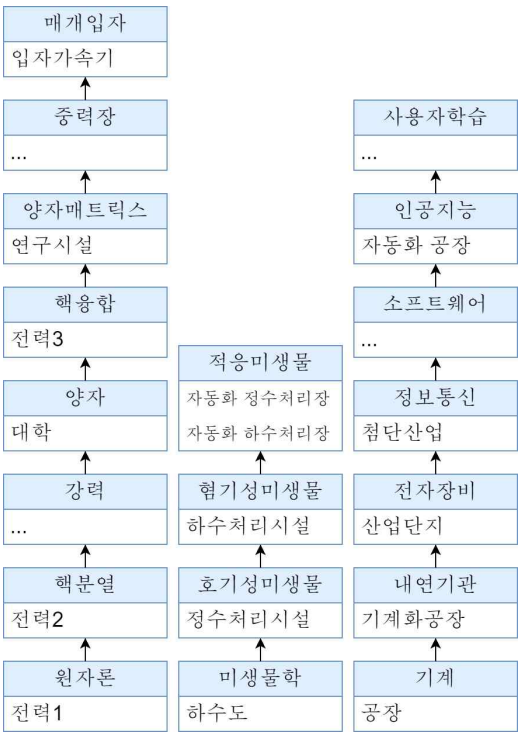
1) 도시 시설



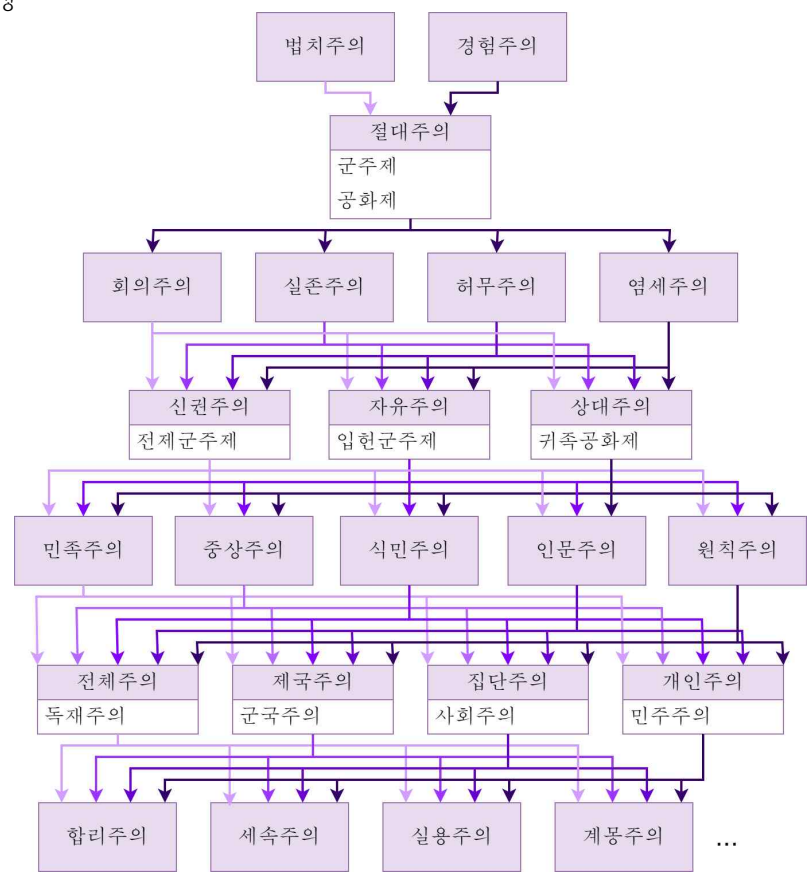
2) 사회



3) 기술


매개입자
입자가속기
중력장
...
양자매트릭스
연구시설
핵융합
전력3
양자
대학
강력
...
핵분열
전력2
원자론
전력1
적응미생물
자동화 정수처리장
자동화 하수처리장
혐기성미생물
하수처리시설
호기성미생물
정수처리시설
미생물학
하수도
사용자학습
...
인공지능
자동화 공장
소프트웨어
...
정보통신
첨단산업
전자장비
산업단지
내연기관
기계화공장
기계
공장

4) 사상


법치주의
경험주의
절대주의
군주제
공화제
회의주의
실존주의
허무주의
염세주의
신권주의
전제군주제
자유주의
입헌군주제
상대주의
귀족공화제
민족주의
중상주의
식민주의
인문주의
원칙주의
전체주의
독재주의
제국주의
군국주의
집단주의
사회주의
개인주의
민주주의
합리주의
세속주의
실용주의
계몽주의
...

5) 문화

미디어 문화 (대중의식 제어)

자산화 (영향력 최대치 결정)
작가협회, 시나리오 검수, 시상식
(각각 재사용 대기시간 존재)

영향력 (영향력이 높을수록 대중의식 제어 효과 상승)
예술 의식, 대중영화, 인터넷 밈 유머
(각각 재사용 대기시간 존재)

혼인률 제어
책임 중심 (인구 증가)(혹은 가족 중심이라고 이름 짓는다)
쾌락 중심 (인구 감소)

지지율 제어
시설 - 발전 가능성
사회 - 유토피아 희망
연구 - 편리성 추구
외교 - 자국 우월주의

고유문화 (소프트파워 증가)
의복
음식
건축

6) 외교

공공외교 (우호도 상승)

문화잠식 (고정 수치만큼 우호도 증가)
음악 공연, 동양화, 의복 유행
(각각 재사용 대기시간 존재)

문화강탈 (플레이어의 소프트파워가 해당 기업을 압도할수록 효과 증가)
영화 수출, 문화 편견, 자국 영웅화
(각각 재사용 대기시간 존재)

문화공정 (플레이어의 소프트파워가 해당 기업을 압도하지 않으면 역효과)
역사 기원, 역사 비하, 정치인 압박
(각각 재사용 대기시간 존재)

공작 활동 (혼란 유발)

선동

극단적 채식주의 선동, 약자 혐오 선동, 전자 페미니즘 선동
(각각 재사용 대기시간 존재)

음모론

창조과학론 투입, 설거지론 투입, 정치 음모론 투입
(각각 재사용 대기시간 존재)

(혹은 가능하면 현실에 없는 음모론으로 이름을 지어낼 것)

마. 목표

1. 포트폴리오 목적

가) 상태패턴을 이용한 유한상태기계를 사용한다.

나) 현실과 가까운 물리를 게임에 적용한다.

다) 복잡한 테크트리 구현 시 생산성 높은 코드를 만든다.

2. 게임 목적

가) 사람들한테 물리, 우주 및 인류 역사에 관심을 유도한다.

나) 학문에 대한 긍정적인 인식을 유도한다.

다) 게임이 교육적일 수 있음을 알린다.

3. 주차별 실행 목표

1주차: 프로토타이핑

플레이 중의 화면과 기본적인 동작을 구현하고 물리 공식으로 산출된 값이 평형을 이루는지 확인.

2주차: 알파버전

행성 테라포밍 기능을 완성하고 문명 운영 및 테크트리 기능을 구현 시작한다.

3주차: 베타 버전

계속해서 기능을 구현하거나 버그를 고치고 구현했던 모든 기능을 안정적인 상태로 만든다.

4주차: 최종 버전

구현했던 기능 중 일부를 더 구체화하거나 우선순위가 낮아서 미뤘던 아이디어를 시도해본다.