몬스터 디자인

몬스터 디자인 시 고려 사항

1. 게임의 장르 및 플레이 방식
2. 게임의 콘셉 기획
3. 게임의 세계관
4. 몬스터가 주는 게임 플레이의 재미, 난이도, 보상
5. 몬스터의 게임 내 성능 및 그에 따른 조합과 상성

전체 몬스터 구상

1. 몬스터 타입 별 분류
2. 몬스터 등급 별 분류
3. 몬스터 Variation 정책 및 방식 결정(텍스처, 애니메이션, 사이즈 등)
4. 맵 – 몬스터 연결

몬스터 별 시나리오 결정

기본적으로 캐릭터 시나리오 설정과 비슷한 방식으로 작업

몬스터의 등장 빈도나 등급에 따라 설정의 디테일을 조정

유니크, 보스 몬스터의 경우 전체 게임 내에서의 역할 및 비중을 상세히 정리

몬스터 별 비주얼 요소 설정

기본적으로 캐릭터 설정과 거의 동일하며 몬스터의 생리적 특성 부분을 예시 이미지와 함께 자세히 설명

대표적인 외적 특징

* **신체적 특징, 복장, 대표 아이템**: 부위 별로 예시 이미지와 함께 정리
* 외형 **Variation**: 기본형에서 바리에이션이 파생되는 경우 기본형을 중심으로 각 바리에이션을 설명
* 주요 **모션** 정리: 몬스터의 주요 모션을 예시 이미지나 동영상 링크, 연속 이미지와 함께 정리 (동영상의 경우 문서 공유가 쉽지 않아 가급적 이미지 사용)
* 위의 각 요소를 Visual적 참고 요소를 최대한 동원해서 정리

몬스터 별 게임 플레이 설정

기획적 기능: 재미, 동기 부여, 난이도 등

게임 플레이시 역할

주요 유저 체험 요소

상대하기 적합한 유저 레벨

함께 나오는 다른 몬스터 들과의 조합

Class 속성 별 수치 및 데이터 값

스킬, 부가능력, 특수능력

이동 형태: 이동 없음, 지상, 공중, 수중 등

AI 패턴

몬스터가 주는 각종 보상: 경험치, 돈, 드랍 아이템

몬스터 시스템 디자인

몬스터 시스템 설정의 고려 사항

아래의 질문에서부터 시작한다. 몬스터에 관련된 각종 시스템 기획 요소들(구조, 규칙, 시퀀스, UI)을 정리

* 몬스터를 사용하는 시스템, 콘텐츠는?
* 몬스터 시스템 기획을 위해 함께 결정되어야 할 요소는? (스킬 기획, 장비 아이템 기획 등)
* 몬스터 AI 및 행동 패턴

몬스터 구조 설정

아래의 항목들을 체크하면서 구조 설정

* 기본적으로 많은 부분이 캐릭터와 겹침: 밸런스 적 측면에서 관리하기 쉽게 하기 위해 + 게임 내에서 하는 행동이 상당히 비슷하기 때문에
* 몬스터의 기본 정보 및 몬스터와 다른 클래스의 연관 방식 정리(몬스터와 스킬, 몬스터와 AI 등)
* 몬스터의 파라미터 설정
* 몬스터의 상태 구분

몬스터 기본 정보

몬스터 기본 정보 정리

* ID, 몬스터 명(언어별), 아이콘, 메시, 텍스처, 이펙트, 배율(몸에 붙는 이펙트와 사이즈의 조정으로 새 몬스터를 만들 때), 뼈대 타입(2족 보행, 4족 보행 등), 이동 방식 등

몬스터 Class와 타Class와의 관계도

* 종족, 스킬, AI, 드랍 아이템, 몬스터 그룹(한 묶음 단위로 생성되는 몬스터) 등

캐릭터와 마찬가지로 밸런싱 이슈가 자주 발생하는 항목은 별도의 Class로 분리

위의 관계도를 바탕으로 각 관계에 대한 시스템 기획 정리

몬스터 능력치와 파라미터

몬스터의 레벨, 등급

* 일반, 강화 보스 등

기본 능력치와 파라미터 리스트, 능력치와 파라미터 연결은 캐릭터 Class의 항목을 상속받아 사용

몬스터 상태 구분

기본적으로 캐릭터의 상태, 버프, 디버프 Class의 항목을 참고

몬스터 타입, 뼈대 타입 등에 따라 상태 설정

* 예: 2족 보행, 4족 보행, 식물 등 고정형, 비행형 등

몬스터와 관련된 각종 규칙 설정

기본적으로 캐릭터 규칙 설정과 업무 진행 과정이 동일

몬스터 클래스의 각 속성과 행동을 체크하면서 필요한 시스템 기획 리스트 정리

몬스터의 상태 변화에 관련된 세부 규칙 정리

몬스터와 다른 클래스와의 관계 별로 시스템 기획 정리

* 아이템 장착, 아이템 드랍, AI, 패턴 등

몬스터 시스템 기획의 각종 시퀀스, UI 기획도 캐릭터 기획과 진행 과정이 동일

AI와 패턴 설계

AI의 활용

실생활에서 실제로 일어나는 행동의 구현

유저의 감정 몰입

전형적이지 않은, 항상 새로운 상황의 구현

기본적인 AI/패턴 설계 요소

AI가 자신의 기준으로 상황을 **분석**해야 한다

분석에 따라 **반응**해야 한다

필요하다면, 분석에 따라 이후 상황을 **예측**해야 한다

AI와 패턴 설정의 고려 사항

1. 해당 몬스터의 주요 설계 목적
2. 공격 전환
3. 도망 설정
4. 공격 타겟 결정
5. 몬스터의 상태 및 행동 리스트
6. 몬스터의 조합

예시 IDLE 상태 구성

|  |  |
| --- | --- |
| 특정 구역 배회 | NPC의 최초 위치 설정, 배회하는 구역 좌표 값 설정, 이동 속도 설정, NPC는 자신의 현재 위치를 분석, 이동 방향은 설정한 구역 내에서 무작위로 설정, 배회할 시간 주기 설정, 오브젝트 충돌 시 방향 전환, 산정한 좌표가 설정한 좌표가 선정 좌표를 벗어날 경우, 좌표를 재설정 |
| 순찰 | 일정 구간 좌표 값 설정, 이동 속도 설정, 구간 끝에 도달하면 다시 반대 방향으로 이동, 순찰 구역 내에 적이 존재하지 않을 때의 모션, 적을 인지할 수 있는 거리, 시야 안에 또는 일정 반경 이내에 적이 들어오면(소리 크기가 일정 크기 이상이면), 적이 보였을 때 공격 모션으로 전환, 공격 사거리, 공격 타입, 도망가는 적을 추적하는 최대 거리, 일정 거리 이상 멀어질 경우 원래 순찰 위치로 복귀 |
| 호위 | 호위할 NPC 또는 보스 몬스터 지정, 지정된 NPC 또는 보스 몬스터에게서 벗어날 수 있는 최대 거리 설정, 거리를 넘어서면 원래 위치로 복귀, 플레이어가 해당 NPC 또는 보스 몬스터에 일정 거리 이상 접근할 경우 플레이어에게 먼저 접근한 후 공격, 전투가 가능한 최대 거리, 여러 플레이어가 같이 해당 NPC 또는 보스 몬스터를 공격할 경우 NPC 또는 보스 몬스터에게 가장 많은 데미지를 준 플레이어를 우선적으로 공격 |
| 인사, 잡담 | 자신의 일정 반경 내에 플레이어가 들어오면 플레이어의 방향을 바라보기, 바라본 다음 인사 메시지 출력, 캐릭터마다 설정된 대화 모션 출력, 일정 시간마다 인사 메시지 출력, 잡담은 해당 메시지 몇 개를 설정한 다음 일정 시간마다 설정한 메시지들 중에서 무작위 출력 |
| 두리번거리기, 꾸벅꾸벅 졸기 | 일정 시간마다 해당 모션을 발생, 일정 시간 내로 플레이어의 자극이 없으면 조는 모션을 출력, 일정 시간이 지나면 다시 원래 모습으로 복귀 |

공격 전환

몬스터 AI타입에 따라 다음 상황에서 IDLE 🡪 공격 상태로 전환

* 캐릭터의 공격을 받을 경우
* 시야 안에 캐릭터가 포착되었을 경우
* 이벤트 등에 의한 강제 전투

반대로 다음 상황에 공격 🡪 IDLE 상태로 전환

* 시야에서 캐릭터가 사라짐
* 공격 타겟의 사망
* 캐릭터가 몬스터의 순찰 범위 바깥으로 이동
* 이벤트 등에 의한 강제 전투 종료

도망 체크

공격 행동과 도망 행동을 결정

도망 체크 예시

* 기본 용기 수치 = 100 \* (HP/MaxHP)
* 몬스터 테이블(또는 AI 테이블)에 **도망 수치** 지정
* 시야 안에 동료가 있을 때 추가되는 보너스 용기 수치를 테이블에 지정

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 동료 1 ID | 동료 1 추가 용기 | 동료 2 ID | 동료 2추가 용기 | 동료 3 ID | 동료 3 추가 용기 |
| 10015 | 5 | 10016 | 10 | 10017 | 15 |

* 몬스터의 용기가 도망 수치보다 높을 경우 공격, 이하일 때는 도망을 선택

공격 타겟 설정

적대도를 체크하는 테이블로 가장 적대적인 상대를 체크

체크 타겟 ID

* 몬스터 주변의 모든 적대적인 객체(캐릭터, 소환 수, NPC, 공격 건물 등)를 체크하는 ID
* 해당 ID는 서버 프로그래머의 설계 규약에 맞춰 설정

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 타겟1 ID | 타겟 1 적대도 | 타겟 2 ID | 타겟 2 적대도 | 타겟 3 ID | 타겟 3 적대도 |
| 103 | 10 | 104 | 20 | 105 | 30 |

* 공격 타이밍이 되면 적대도가 가장 높은 타겟을 공격

공격 타겟 설정

적대도는 다음과 같은 경우에 상승

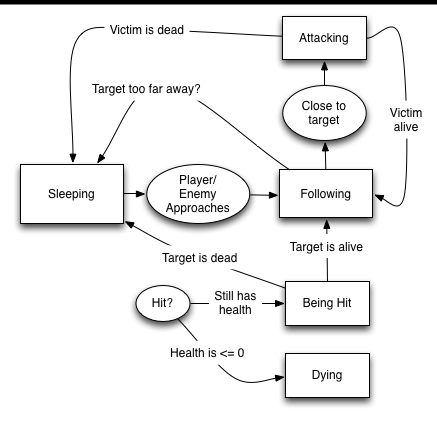
1. 일반 공격을 당함(데미지를 적대도로 환산)
2. 스킬 공격을 당함(데미지, 적대 보너스를 적대도로 환산)
3. 적대도 체크 테이블에 있는 대상을 치료, 버프
4. 디버프를 당함(적대 보너스를 적대도로 환산)
5. 무조건 어그로를 끄는 스킬을 당함
6. 적대도 테이블은 수시 체크 + 체크 대상이 많기 때문에 필요 없는 적대 체크 대상은 주기적으로 삭제해줘야 한다

FSM(Finite State Machine)과 BT(Behavior Tree)

FSM (Finite State Machine) – 상태는 어떤 게 존재하고 어떤 조건에 의해서 상태가 전이되는지 정의

일반적으로 많이 사용하는 모델로 각 상태들을 조건의 흐름에 따라 정의하여 AI를 구성하는 방식

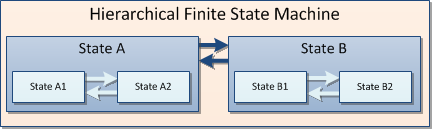
간단한 상태 변화를 가진 AI를 구현하기에는 좋으나 변화하는 상태가 많아질수록 관리하기가 힘들어 상태 변화의 확장이 제한됨



HFSM (Hierarchical Finite State Machine)

일반적인 FSM의 모델을 확장한 모델로 전체 흐름을 그룹화하고 그룹을 세부 FSM으로 보완하여 다양한 상태들을 처리

세부 FSM을 모듈화 하여 조합하면 다양한 AI 유형을 만들 수 있다.



BT (Behavior Tree)

상태를 계층적으로 나누어 정의하는 방식으로 전체 상태를 파악하고 모듈화 하는데 편리

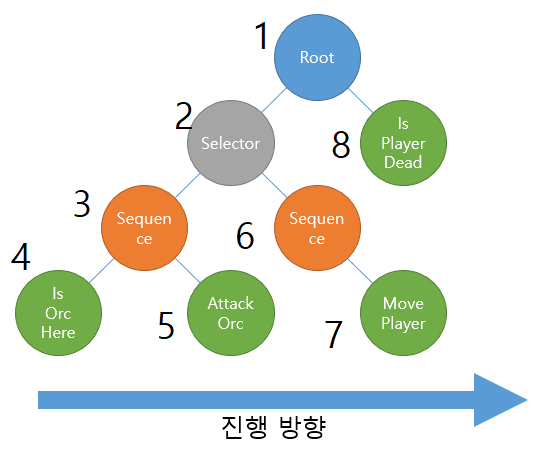
트리 탐색 구조를 이용해 행동의 우선 순위를 정함

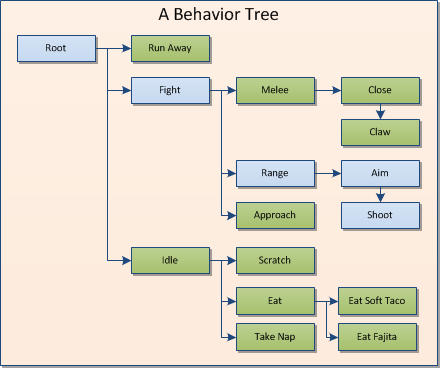
FSM: 상태를 정의하고 이벤트에 따라 상태가 바뀌는 것을 구조화

BT: AI가 달성할 목표를 정의하고 서브 태스크로 나누어 구조화

Selector: 1개라도 참이면 실행

Sequence: 모두 참이어야 실행





나이츠 오브 클랜: 영웅들의 AI를 유저가 직접 설정하여 전투 가능



GLADIABOTS: 유저들이 유닛들의 AI를 BT기반으로 편집하여 전투