게임 디자인 기초: 시스템 디자인 입문

시스템 기획 = 프로그램 설계

어떤 기능 요소로 만들어야 되고, 그 기능에 필요한 값들은 무엇이고, 어떤 규칙이 필요하고, 어떤 흐름의 시퀀스, 팔로우를 갖고 있는지, 정보(UI)를 사용자가 어떤 식으로 받아들이게 할 지

플레이를 위한 메카닉스를 설계하는 일

Ex) 점프 게임에서 점프를 하려면 어떤 요소가 필요하고, 어떤 규칙, 어떤 구조, 어떤 UI, 어떤 시퀀스를 가지고 있는 지

* 점프를 하고 나서 0.5초 안에 점프 키를 누르면 2단 점프가 되게 하자(이걸 구현했을 때 유저들이 이것을 받아들이고 충분히 조작할 수 있는 지, 일정 구간부터 2단 점프를 하게 했을 때 이게 타당한지)

콘텐츠 생산

몬스터를 만들 때 어떤 규칙을 갖고 있는 지, 어떤 UI를 갖고 있는 지

개발의 효율성을 위한 어플리케이션 설계

시스템 디자인 방식

마케팅, 게임 분석에서의 3가지 키워드는 게임 시스템 기획에서도 유효: **의도, 구현 방식, 실제 효과**

구현 방식, 실제 효과 검증: 아래 그림에서 왼쪽으로 갈수록 확실한 검증 가능, 오른쪽으로 갈수록 소모 자원의 최소화

게임 장르와 상황에 따라 왼쪽과 오른쪽 극단의 중간 방식을 잘 선택하는 것이 중요

예시)필드에서 몬스터를 잡다가 죽으면 경험치가 깎인다 🡪 플레이의 긴장감을 주기 위해서, 콘텐츠 소모 속도 조절, 수익 모델을 형성하기 위해서

UI: 메뉴 버튼은 어떻게 작동해야 되는데? 크기는? 어디에 있어야 되는데? 입력하는 방식은?

UX: 사용자가 체험할 수 있는 모든 체험들(사용자가 어떻게 플레이하게 만들 것이냐)

예시) 달빛 조각사, V4, 리니지M, 검은 사막 🡪 UI 구조가 비슷(UX가 비슷) – 디자인 전달 방식도 매우 유사 🡪 새로운 것을 만들어보려고 해도 유저들의 학습 거부감이 심해서 못 만들어 봄

조금이라도 바꿔보려고 시도하는 게 UI 디자인 형태, 색깔 변화 시도

시스템 디자인의 항목

“모든 시스템 설계는 다음의 항목을 포함한다”

1. 구조: 해당 시스템은 어떤 구조, 어떤 조건에 따라 분기가 이루어 지는 지
2. 규칙
3. 시퀀스
4. UI

시스템 구현: 구조

시스템에 연관된 각 요소들이 어떤 속성과 행동으로 구성되어 있으며 요소 간은 서로 어떤 연관을 가지는 지

객체 지향 설계 방식에 맞추어 게임 내 각종 요소들을 설계

자료 형, 입력 값 범위 등 **제약 조건**에 대한 명확한 정의가 중요(예를 들면 만렙, 화폐 소지 제한 금액 등)

Cf) 객체: 구체적인 대상을 의미하며 다른 객체와 구분되어야 한다(객체에는 속성, 행동이 필요하다)

캐릭터 속성 예시: ID, 이름, 캐릭터 모델링, 레벨, HP, 스탯, 스킬, 이동 속도, 사용 가능 무기 타입 등

예시) 런 게임 시스템의 캐릭터 시스템 요소의 속성: ID, 이름, 사이즈(이미지 크기), 이동 속도, 점프력, 캐릭터 모델링, 체력, 고유 능력, 점프 가능 거리, 점프 높이 등

ID = 숫자 형/이름 = 문자 형/캐릭터 모델링 = 파일 이름(문자 형)/레벨 = 숫자 형/HP = 숫자형/고유 스킬 = 문자 형

행동: 4 가지 범주로 정의

1. 스스로 하는 행동
2. 행동을 할 것을 제시했을 때 행동(특정 값이 주어졌을 때 할 수 있는 것)
3. 어떤 값도 필요 없고 어떤 값을 전달할 필요 없지만 어떤 값을 전달받아야 할 수 있는 거
4. 특정 값을 지정할 필요는 없지만 내가 결과를 제시해야 되는 행동/어떤 값을 받으면 그 값을 가지고 특정 행동을 수행해서 결과를 제시

게임(갤로그) 내부 화면 예시

타이틀

속성: 최고 점수, 동전 투입 메시지, 로고, 별이 흐르는 이펙트 등

행동: 동전을 넣으면 동전 투입 메시지 없어진다. 스타트 누르면 게임 화면으로 넘어간다

게임 화면

속성: 전투기 – 연사 속도(초 당 몇 발 발사할 수 있는 지), 이동 속도(초 당 몇 픽셀 이동하는지), HP = 1, 사이즈, 피격 범위, 이미지, 라이프/적 – 대형 전함, 붉은 전투기, 푸른 전투기/점수/칸 수

행동: 적을 배치/적 전투기가 피격되면 점수가 올라간다/플레이어 전투기가 파괴되면 새로운 전투기가 나타난다/적 전투기를 모두 파괴하면 다음 스테이지로 넘어간다 등/전투기 – 이동, 발사, 합체, 폭파, 납치

게임 오버

속성: 순위, 스코어, 게임 오버 메시지

시스템 구현: 규칙

게임 세계를 구성하는 규칙

게임 플레이의 각종 룰

게임의 각 요소, 요소 간의 관계, 시퀀스에 적용되는 세부적인 규칙 등이 포함

**각종 예외 상황**에 대한 꼼꼼한 정의가 매우 중요

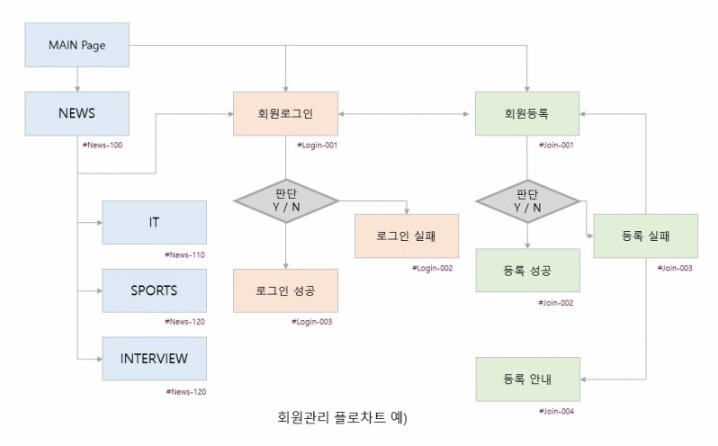
시스템 구현: 시퀀스

시스템이 어떤 플로우와 시퀀스로 구성되는 지

페이지 및 신 전환, 메모리 관리, 서버 – 클라이언트 간 데이터 처리 방식 등이 시퀀스 설계를 통해 구체화되어야 한다

* 시퀀스: 해당 시스템의 흐름에 대한 순서를 의미

예시)



시스템 구현: UI

해당 시스템이 어떤 UI를 통해 정보를 입력, 출력하는 가

예시)

이동 – 레버를 좌우로 흔들면(또는 좌우 방향키를 입력하면) 이동을 한다

발사 – 버튼을 누르면 총알 한 발 나간다(혹은 누르고 있는 동안 총알이 나가게 한다)

전투기의 수나 HP를 어떻게 나타낼 것인 가

HUD: 떠 있는 창, 팝업 만이 아니라 전투 시 나타나는 데미지 숫자, 생명력이 아슬아슬 할 때 화면 전체가 빨갛게 변한다던 가 등의 효과를 주는 UI들

효과 검증

1. 플레이 가능 버전 제작
2. 프로토타이핑
3. 시뮬레이션 문서
4. 타 게임 사례 분석

역 기획

역 기획: 기존에 출시된 게임의 전체 또는 일부를 다시 개발 가능한 상태의 기획서로 정리한 것

일반 업체에서 기획자의 포트폴리오로 가장 선호하는 형태의 문서 중 하나

(왜 이렇게 만들었을까: 의도 파악, 왜 이런 속성들이 필요할까 까지 고려된 역 기획서 선호)

시스템 디자인 프레임워크

프레임워크 = 일을 편하게 할 수 있는 구조

MDA Framework

게임의 구조와 시스템을 분석하기 위한 기법 중 하나

구조(mechanics)

* 시스템으로 써의 게임을 설명하는 공식적인 **규칙**들과 **개념**들

역학(Dynamics)

* 게임을 플레이하는 과정에서 나타나는 플레이어의 **행동**

미학(Aesthetics)

* 게임 디자이너가 역학을 통해 의도한 플레이어들의 감성적 반응(즉, 바꾸어 말하면 **재미**)

예시) 팩맨



구조: 빨간 유령은 현재 팩맨의 위치로, 핑크 유령은 팩맨이 바라보는 타일 위치로 움직임

역학: 유저는 빨간 유령과 핑크 유령이 자신이 둘러싸는 것을 경험

미학: 게임의 긴장감 느끼게 됨

MDA 프레임워크 효과

게임 디자이너는 구조 🡪 역학 🡪 미학의 시각으로, 플레이어는 미학 🡪 역학 🡪 구조의 시각으로 게임을 바라보게 된다

MDA프레임 워크에 의한 분석은 게임 디자이너들이 **“사용자 체험 중심의 설계”**를 할 수 있게 한다

MDA 프레임워크: 미학

1. 감동(Sensation): 즐거움(sense – pleasure)으로서 게임
2. 공상(Fantasy): 공상(make – believe)으로써 게임
3. 서사(Narrative): 이야기(drama)로서 게임
4. 도전(Challenge): 장애물 경주(obstacle course)로서 게임
5. 연대감(Fellowship): 사회적 틀(social framework)로써 게임
6. 발견(Discovery): 미개척지(uncharted territory)로써 게임
7. 표현(Expression): 자기 발견(self - discovery)로서 게임
8. 몰입(Submission): 오락(pastime)으로서 게임

MDA 프레임워크: 역학

역학: 미학적 경험을 전달해주는 단계

예시)

* 도전: 제한된 시간, 적의 거센 공격
* 연대감: 정보의 공유, 혼자 달성하기 어려운 목표 설정

역학 모델: 확률

확률을 통해 유저들을 제어할 수 있다

* 3강부터는 아이템 강화 시 아이템 파괴될 수도 있다: 상위 단계의 유저 수를 제한, 유료 아이템 결제 유도
* 선수 카드를 하루에 한 장만 뽑을 수 있다. S급 아이템을 뽑을 확률은 1%입니다: 게임 플레이 타임 및 콘텐츠 소모 속도 조절

역학 모델: 게임 내 자원

게임 내 자원의 관리를 통해 유저의 행동을 제어, 예측할 수 있다(영어로는 Token이라고도 함)

대표적으로 돈, 시간, 경험치 등이 있다

* 애니팡 하트는 5분에 1개 씩 재생된다: 유저의 회당 플레이타임의 길이는 제한(게임의 권태감 감소), 대신 재 접속 횟수를 늘림(게임의 수명 연장)
* 재생 불가능한 자원(제한 시간, 전략 게임에서 자원 밭 등)으로 유저의 행동 조절

자원의 다양한 형태

자원의 목적, 해당 자원을 사용하는 게임플레이, 획득 조건, 소비 조건, 조절 가능 여부, 변동 방식, 변동 빈도, 변동폭 등에 따라 다양한 형태로 게임의 구성에 영향을 미친다

값이 유지되거나 증가하는 자원: 경험치 등

값이 유지되거나 감소하는 자원: 제한 시간 등

유저가 목적을 가지고 소비하는 자원: 돈, 재료 등

역학 모델: 피드백 시스템

피드백 시스템은 자신의 상태를 확인하고 조절한다

부정적인 피드백

* **시스템을 안정화**시킨다. 대전 게임의 경우 약팀에게 역전의 기회를 제공하고 의욕을 제공한다

긍정적인 피드백

* **시스템을 불안정**하게 한다. 대전 게임의 경우 강팀에게 더욱 큰 메리트를 부여하여 전세가 결정된 후 무의미한 버티기, 시간 낭비 등 방지

예시)

부정적 피드백

* 라인전에서 미니언을 많이 처치할수록 캐릭터는 점점 더 위험에 처하게 된다

긍정적 피드백

* 타워를 빨리 철거할수록 상대 정글과 다른 라인을 더욱 강력히 압박할 수 있다
* 코에이 삼국지 시리즈의 경우 후반 긍정적인 피드백에 대한 설계가 약해 일정 순간 이후부터 게임이 급격히 늘어진다