UX와 UI 개론

UI: 컴퓨터나 장치, 단말기에서 정보를 제공하고 유저에게 명령을 입력 받는 체계(사용자와 장치가 소통하는 모든 수단)

UX: 사람들이 어떤 프로그램을 쓰더라도 좌측 상단 버튼을 누르면 뒤로 간다 🡪 유저들에게 뒤로 가기라는 기능은 좌측 상단의 버튼을 누르면 된다는 학습

UI를 만들려면 이런 UI들을 만들려면 어떤 접근을 해야 되는가, 무엇을 보여줘야 하는가에 대한 고민이 필요

UI학습의 목표

단순한 UI 구성의 방법만을 학습하는 것이 아닌 UI를 통해 게임을 개선하고 특정 목적을 달성하는 것

좋은 UI는 유저 이탈률을 줄이거나, 결제율을 높이거나, 게임의 재미를 높일 수 있다.

사용자들이 **쉽게**, **원하는 기능**을, **편리하게** 사용할 수 있게 (디자인, 학습, 사용 측면)

UI의 범주

UI = HUD라고 혼동하는 경우가 많으나, 입출력에 관련된 모든 형태의 사용자 인터페이스는 UI에 해당

HUD, 조작방식, 타격 마크 위에 뜨는 데미지 수치나 크리티컬 텍스트, 정보를 알려주는 사운드(잔액이 부족합니다) 등

UX(User Experience)란?

타겟 사용자가 실제로 겪게 되는 체험

제품, 시스템, 서비스를 사용하거나 예상하는 사용법에 대한 사람들의 인식과 반응

UX 🡪 UI 설정 단계

리서치: 타겟 유저에 관한 각종 정보 수집 🡪 비슷한 유형의 게임 UI 분석

모델링: 페르소나, 유저 스토리를 통한 타겟 유저 명확화

전략 디자인: 타겟 유저에 대한 사용자 체험 전달 전략 수립

UI 프로토타이핑: 각종 프로토타이핑 방식을 통한 UI 제작

UI 완료

UX의 고려 요소

유용성(Useful)

사용하기 쉽고(Usable)

매력(Desirable)

찾기 쉽고(Findable)

접근성(Accessible)

신뢰성(Credible)

가치(Valuable)

페르소나

해당 프로젝트에 가장 적합한 가상의 유저

페르소나로부터 사용자의 **스토리**를 구성

프로젝트의 필요에 따라 페르소나의 속성 설정

페르소나는 유저 분석, 목표 수립, 기대 효과 예상 등 마케팅 분야에서도 광범위하게 유용

좋은 디자인

좋은 디자인은 UI, UX의 핵심 요소

좋은 디자인은 6가지로 정의

* 가시성
* 피드백
* 행위 유도
* 매핑
* 제약
* 일관성

가시성 = 눈에 잘 띄는가

* 디바이스의 상태와 가능한 액션을 인지할 수 있어야
* 한 정보와 그렇지 않은 정보를 구분 가능해야
* 정보는 필요한 상황이 아니면 숨기는 게 좋다
* 실한 정보 🡪 여러 번의 시도 유도

피드백 = 지금 어떤 상태인가? 어떤 액션을 수행 중인가?

* 유저의 행위에 따른 즉각적인 변화가 있어야
* 유저의 행위가 즉각적으로 표시되어야

행위 유도 = 어디에 쓰는 물건인가

해당 디자인의 기능, 사용법을 외형을 통해 예측 가능해야

모양, 텍스트, 툴 팁 등의 형태로 제공

매핑 = 콘트롤과 실제 효과가 대응되는 가

* 위치, 크기, 배치 순서 등에 따라 좋은 디자인 결정

제약 = 할 수 있는 행동과 할 수 없는 행동의 지정

* 에러 확률을 줄인다
* 필요한 업무나 행위에 대한 주의 집중

일관성 = 친숙하고 예측 가능한가

* 비슷한 기능을 수행할 경우 비슷한 디자인으로
* 유저들이 상식 또는 선행 학습된 정보를 통해 쉽게 적응 가능하게

UI 설계

사용성(Usability)

UI의 핵심 가치, 사용성으로 번역

보다 사용하기 편리한 시스템을 만들기 위한 설계, 평가 과정

핵심 항목

1. 학습의 용이성: 초보자가 시스템에 대한 지식을 습득하기 쉬운지(아이콘)
2. 사용 편의성: 얼마나 쉽고 편리하게 해당 기능을 사용할 수 있는지(단축키)

* 위 두 개념은 상충되기 때문에 다중적 시스템으로 커버(초보자는 아이콘으로 상급자는 단축키로)

1. 기억 용이성
2. 적은 오류
3. 주관적 즐거움: 아름답다, 깔끔하다, 재미있다 등

UI에서 고려되어야 할 요소

1. 인지, 지각 및 일반적인 인간의 운동 능력
2. 사용자 계층의 특성
3. 사용자가 처한 환경에 따른 특성: HP물약 단축키 vs 강화 스크롤 단축키
4. 선행 조건 및 필요 조건: 이 갑옷은 힘이 부족해서 입을 수 없군 (불필요한 조작 실패 최소화)
5. 디바이스의 특징: 키보드, 조이스틱, 터치스크린, 진동 등

UI의 기대효과 측정 방법

1. 과제를 마치는데 걸리는 시간
2. 과제를 마칠 때까지 화면 수, 클릭 수 🡪 많을수록 복잡
3. 에러에서 복귀하는 데 걸리는 시간 🡪 홈쇼핑에서 모자 1개를 사려다가 7개를 다 사려고 취소하고 돌아갈 때
4. 잘 못 선택한 횟수
5. 반복된 에러의 횟수 🡪 메시지가 제대로 전달되지 않거나/유저들이 기존에 이용하던 것과 다른 모델인 경우
6. 도움을 받기 위해 UI 외부(다른 사용자, 헬프 데스크, 매뉴얼)을 이용한 빈도
7. 도움을 받는 절차와 편의성

사용자 측면의 분석

**콘텐츠 가치** 분석: 꼭 필요한 건가

* 콘텐츠에 대해 느끼는 가치는 사람마다 다르다

**사용자** 분석: 사용자가 중요하게 느끼는 가치는? (기능성, 유희성, 개성 등)

* 사용자의 숙련도는?

**과업** 분석: 실제로 어떤 방식으로 시스템을 사용하는가

* 10명의 고스트가 캐릭터 한 대에 동시에 락 다운을 걸려고 한 건 아는데 워3의 오토 캐스트, 스타2의 스킬 캐스팅

**상황** 분석: 어떤 상황에서 사용하는가

* 땅에 떨어진 물약, 강화 스크롤을 먹으면 인벤토리로 가야 되는지, 단축키 창으로 들어가야 하는지? 땅에 떨어진 무기를 주우면 바로 착용, 교체되어야 하는지

사업적 측면 분석

* 회사가 요구하는 목적
* 프로젝트의 목적: 기대하는 결과/결과물의 활용 방안
* 사용자 특징: 타겟 사용층의 다양한 세부 요소 파악

기술적 측면의 분석

* 구현 가능한 수준
* 디스플레이의 특성(사이즈, 해상도 등)
* 입력 장치
* 속도

UI와 다른 파트의 개발과의 차이

분석 🡪 구현 🡪 평가 과정의 반복

* 수시로 기획 수정 및 추가 필요

분석: 사업 목적과 사용자 목적 달성을 위한 Usability Goal설정 분석 필요

구현: 디자인, 테스트, 개발

평가: 테스트, 인터뷰, 설문지 등등 다양한 방법의 사용자 피드백

중요도, 예상 완료 시점 등에 따라 유동적으로 스케줄링해야 한다

현실은?

Case by case

UI는 게임 기획 후반 단계에 ‘급하게’ 들어오는 경우가 대부분

아티스트는 재작업하다가 멘탈 나감 (재미도 없고 해도 표시 안 나고 등등)

UI 위주의 플랫폼이나 장르가 아니면 UI의 중요도에 비해 개발 비중이 낮다

제이콥 닐슨의 체크리스트

가시성: 현재 어떤 작업을 진행하고 있는지 알아볼 수 있는가

피드백 존재 유무

* 현재 어떤 버튼이 눌려 있는지에 대한 피드백이 있는가
* 어떤 대상이 선택되어 있고 움직일 수 있는지에 대한 정보가 있는가

피드백의 명확성

* 상태와 작동 가능 여부를 보기만으로 알 수 있는가?
* 어떤 메뉴 항목이 선택되었는지 명확하게 표시되어 있는가

피드백의 즉시성

* 작업에 비해 적당한 반응 속도를 제공하고 있는가
* 지연이 발생하면 사용자에게 진행 속도를 알려주고 있는가

현실 세계와의 부합: 사용자가 상식적으로 납득할 수 있는 시스템, 단어, 글을 사용하고 있는가

실 세상과의 부합 정도

* 실생활에서 늘 사용하는 용어를 사용하는가
* 추상화한 아이콘이 실제 세상과 비슷하게 표현되어 있는가

상식적인 논리와의 부합

* 계층과 관계 별로 구분되어 있는가

사용자 과업과의 부합

* 실제 사용자가 해야 하는 작업과 동일한 이름을 사용하는가

예상되는 행위와의 부합

* 어디를 선택해야 하고 어떤 작업을 하는지 명확한 가

콘트롤: 사용자가 원하는 콘트롤을 자유롭게 할 수 있는가

전반적인 사용자 주도권

* ‘사용자가 왜 이리 못하게 하는 게 많아?’ 라고 느끼지 않는지

취소 가능성

* 사용자가 선택한 행위를 쉽게 취소할 수 있는가
* 취소할 내용을 일부 선택할 수 있는가

사용자의 자유도

* 사용자가 다양한 방식으로 과업을 수행할 수 있는가
* 자유로운 UI 이용이 가능한가
* 부분 수정, 부분 입력된 데이터도 저장 가능한가

일관성: 일관되고 표준화된 체계가 있는가

일관성 있는 이름

* 메뉴나 명령어, 약자가 일관성 있게 제시되는가

일관성 있는 정보

* 메뉴, 제목, 페이지, 에러 메시지 등이 일관성이 있는가

일관성 있는 구조

* 입, 출력 창의 구조가 화면마다 비슷한 구조를 갖고 제공되는가
* 확인, 취소, 도움말 등 중요 버튼의 위치가 일관성이 있는가

일관성 있는 표현

* 아이콘, 스타일, 색감, 콘셉의 일관성이 있는지

에러 방지: 사용자가 실수를 하지 않도록 유도하는 장치를 마련했는가

* 오류가 발생할 가능성은?
* 오류 발생 시 치명적인 영향을 미치는지?

오류를 범할 확률 낮추기

* 헷갈리는 문자, 혼동되는 데이터 등

오류를 범하기 쉬운 것은 보여주지 않기

* 현재 상태에서 가능한 행동만 보여줌

심각한 오류를 범하기 힘들게 하기

* 결과 경고, 기능키나 버튼의 위치 차별화, 재확인 등

예상되는 결과 보여주기

즉각적 인지: 보는 즉시 이것이 무엇인지 알 수 있어야

기억하기 쉽게 하기

* 작업 순서, 메뉴 항목의 배치

명확한 명칭

* 아이콘이나 명령어의 이름을 명확히 제시, 툴팁 제공

적당한 그룹

* 비슷한 항목끼리 논리적으로 구성

시각적 계층 구조

* 사용자의 시선 이동, 중요한 요소의 시각적 인지

명확한 시각적 구분

* 메뉴나 아이콘 간의 구분, 선택 및 활성화 요소의 구분

융통성: 사용방법 1이 불편할 경우 대안이 있는가

전문성에 따른 유연성

* 숙달될수록 복잡한 명령 사용, 많은 정보 제공

옵션의 제공

* 같은 작업이라도 여러 가지 방법으로 수행할 수 있는가

개인화

* 사용자가 시스템을 자신에게 편리하게 재구성할 수 있는가

신속한 수행 방법

* 계층 구조와 관계없이 즉시 사용해야 하는 기능이 있는가

자동적인 수행 제공

* 기계적인 반복 작업을 대신 처리해 주는지

비주얼: 디자인적으로 완성도가 있는지

심미성: 색상, 도형, 서체, 가독성, 배치와 정렬

최소한의 표현: 서로 다른 색의 강도, 색상, 폰트, 도형, 크기 등이 너무 많지 않은가

최소한의 입력 요구: 사용자에게 필요 이상의 정보나 동작을 요구하지 않는지

에러 해결: 오류 발생 시 쉽고 빠르게 해결할 수 있는지

에러 발생에 대한 감지

* 유저가 에러 상황을 정확히 판단할 수 있도록 전달되는가

감지된 에러의 원인 진단

* 유저가 에러의 원인을 정확하게 판단할 수 있는가

에러 복구

* 유저가 에러 복구 방법을 판단할 수 있는가
* 자주 발생되는 에러일 경우 자동 보정 해주는지

보충 설명: 도움말과 보충 설명이 있는가

도움말 제공: 상황 파악, 따라하기 쉬움, 대안 제공

도움말의 표현

* 눈에 잘 띄는가
* 이해하기 쉬운 표현인가
* 글과 그림을 적절히 사용했는가

작업 전환의 용이함

* 원래 작업과 도움말 간의 전환이 쉬운가

게임 UI 기획

UI설계 용어

와이어프레임(Wireframe): 화면 레이아웃을 표시하기 위한 UI 배치

목업(Mockup): 실제와 유사한 정적인 형태의 모형

스토리보드(Storyboard): 정책, 절차, 와이어프레임, 설명 등이 모두 포함된 문서

프로토타입(Prototype): 다양한 인터랙션이 결합되어 실제 서비스처럼 작동하는 모형

스토리보드

UI 스토리보드 🡪 UI가 필요한 신이 어떤 신이 있고 필요한 작동, 필요한 데이터 설명

컷 신 연출 스토리보드

스케일 폼

플래시와 액션스크립트를 사용하는 UI 개발용 미들 웨어

언리얼, 크라이엔진에 내장

이외 타 3D 게임 엔진과 함께 사용 가능

3D를 기반으로 다양한 형태의 동적인 형태의 UI 개발 가능

UI 기획 순서

UI가 필요한 상황 리스트 정리

상황 별 기획

* UI 페이지 리스트 정리
* 페이지 연결도 작성

페이지 별 기획

* UI 콘셉 기획: 기능과 정보를 UI 체크리스트를 참고해 정리
* 세부 스토리보드 기획
* 세부 UI요소에 대한 설명: 텍스트 크기는 얼마, 폰트는 어떤 거, 최대 글자 수는 몇 자까지 등/버튼: 어떤 이름으로 부를 건지, 구성은 눌리기 전과 눌렸을 때로 구성한다 등

UI가 필요한 상황 정리

비슷한 장르, 플랫폼의 타 게임을 참고하는 것이 편리

머리 속으로만 생각하면 상황이나 페이지가 누락될 가능성이 많다

게임 흐름에 따른 분류 – 시작, 로그인, 로비, 인 게임, 마을, 배틀, 종료 등

흐름 별 기능 분류 – 마을: 상점, 대장장이, 경매장 등

페이지 정리

상황 리스트를 바탕으로 UI 페이지 리스트 정리

페이지 간의 연결 및 연결 조건을 정리

UI 콘셉 기획

입/출력 되어야 하는 정보는 무엇인가

해당 정보는 어떤 방식으로 표현되어야 하는가

기타 이전에 배운 UI 개론 내용 바탕으로 해당 UI의 콘셉 정리

스토리보드/세부 기획

와이어 프레임 기반으로 제작

UI간 연결

각 컴포넌트의 **위치와 크기, 기능** 정리

각 컴포넌트의 **액션 및 상태** 정리

게임 내에서 기획해야 하는 UI는 화면 최 상단의 HUD뿐만 아니라 게임 오브젝트 게임 월드 안에서 표시되는 캐릭터 명 표시, 채팅 말풍선, 이동 지역 가이드 등도 포함된다

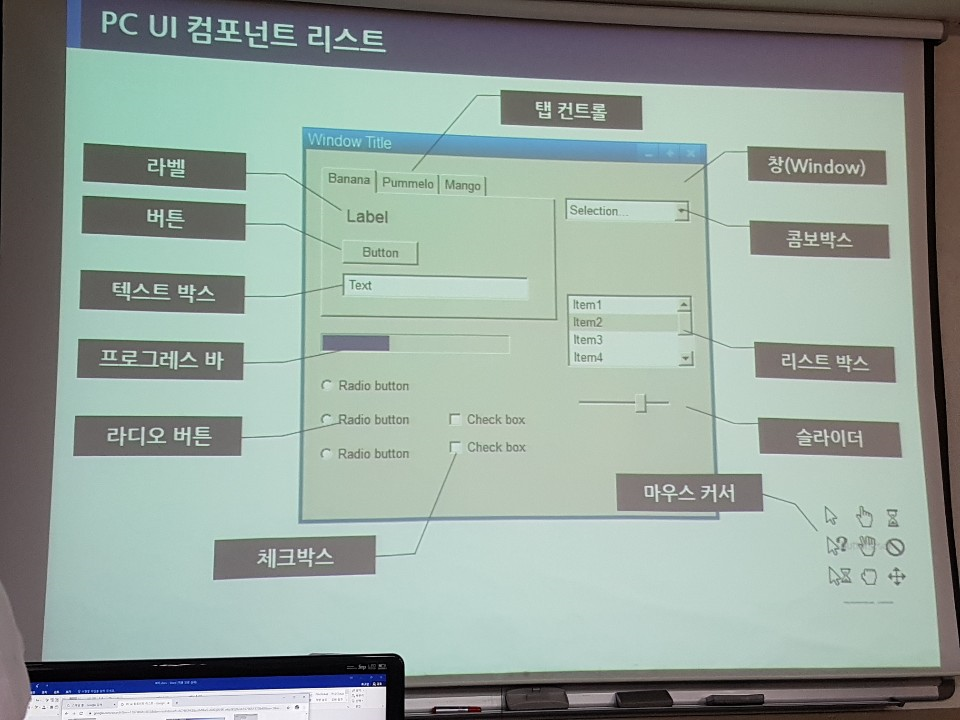
UI기획, 역 기획이 시스템 기획 훈련에 가장 좋은 이유

구조와 규칙 자체가 타 시스템에 비해 명료하고 예측이 힘든 예외 상황 발생이 제한적이기 때문

또한 위와 같은 이유로 대부분의 초보 시스템 기획자에게 UI기획을 맡긴다

UI기획 참고 요소

Pc UI 컴포넌트 리스트



컴포넌트 별 정리 정보

텍스트가 표시되는 컴포넌트

* 사이즈, 폰트, 색상, 최대 표시 글자 수, 줄 바꿈 여부, 글자가 많아졌을 때 스크롤 바 표시 등의 액션

버튼 등 인터랙션이 있는 컴포넌트

* 상태 리스트(기본적으로 Normal, Focused, Pressed, Disabled의 4가지 상태 정리) 정리

기타

* 이펙트, 툴팁 등의 정리

모든 컴포넌트의 사이즈와 위치는 가능한 8의 배수에 맞추고 최소한 짝수 단위로 지정되어야 한다 🡪 최소한 가급적 짝수로 구성해야 한다는 규칙은 지켜야 한다

스마트 디바이스 UI 기획

디바이스 별로 해상도가 다르기 때문에 각 디바이스에 맞춘 UI 기획 필요

가로, 세로 혹은 둘 다 지원하는 지 확인

모든 디바이스에 맞출 수는 없다. 게임 서비스 타겟에 맞는 몇 개의 해상도를 선택해서 작업해야 한다

모든 컴포넌트의 사이즈, 위치 값은 짝수로 지정

버튼 등 인터렉션이 있는 컴포넌트의 경우 터치 영역을 고려해 배치

* iOS에서는 최소 44\*44 이상의 사이즈로 디자인할 것을 권고

스마트 디바이스 입력

PC 디바이스 입력 체계

* 마우스 & 키보드
* 제한적인 상황에서 조이스틱, 마이크 등이 사용되기도 한다

스마트 디바이스의 입력 체계

* 터치 제스처: 터치 스크린
* 기울기 및 움직임의 속도와 방향: 자이로 센서
* 위치, 방위 정보 체크: GPS 및 네트워크
* 조도 및 이미지 감지: 카메라

폰트

게임 내 텍스트 표시 방법 분류

클리어 타입 폰트

이미지 폰트(픽셀 폰트, 비트맵 폰트)

클리어 타입 폰트

벡터 방식의 트루 타입 폰트 기술 사용

안티 알리하싱, 해상도 조절 등이 가능

폰트 사용 시 라이선스 문제 해결 필요

벡터 프로그램 및 조합 처리로 인한 프로그램 부하가 상대적으로 큼

이미지 폰트

완성형 글자 세트들을 하나의 이미지 안에 늘어놓고 사용

해상도 등 서비스 환경에 유동적이지 못 함

폰트 디자인을 보다 유동적으로 사용 가능

픽셀 폰트는 매우 작은 크기로 텍스트를 표시해야 할 경우 유용

글자에 연출이나 이펙트들을 많이 표현해야 할 때 메모리의 부하를 줄여준다

클리어 타입의 폰트보다 프로그램 부하는 적지만 많은 글자를 표현하는 경우 해당 글자만큼의 메모리 부하 발생

세리프와 산세리프

세리프

글자의 기호를 이루는 획의 일부 끝이 돌출된 상태

가독성이 좋아(많은 양의 텍스트를 읽어야 할 때) 인쇄물용으로 주로 사용

산세리프

글자 획의 굵기가 변하지 않고 획의 끝에 장식이 없는 폰트

판독성이 좋아(글꼴의 형태를 식별) 컴퓨터 화면상에 주로 사용

폰트 사용 가이드라인

폰트 크기가 작아질수록 픽셀 폰트의 가독성이 더 좋다

디스플레이 상에서는 세리프체보다 산세리프체의 가독성이 좋다

상하 몸체를 벗어나는 글씨가 많은 서체의 경우 가독성이 떨어지고 UI설계가 힘들다

영문의 경우 대문자와 소문자를 섞어 쓰는 것이 가독성이 높다

한 화면에서 사용하는 폰트는 2개 이하가 적당

다국어 버전의 경우 UTF-8규격의 문자 셋을 지원하는 폰트를 구성 🡪 모든 문자 표현 가능

게임 화면 전반의 색감을 고려해 폰트 색상 및 텍스트 박스의 색상 결정

Csv를 익스포트하여 유니티에서 인포트를 했을 때 글자가 깨지는 경우 🡪 글자 계행 방식 차이 때문에 발생