게임 UI

플레이어가 탐색하고 정보를 찾고 목표를 달성하는데 도움이 되는 구성 요소가 있는 인터페이스

게임 UI는 플레이어가 게임스토리와 상호작용하고 게임 공간으로 침입할 수 있도록 하는 시각적 구성 요소 시스템이다.

Diegetic interface :   
게임 내에 직접 랜더링되는 인터페이스  
HUD에 표시되는게 아니라 게임세계 안에 직접 오브젝트로 삽입되는 인터페이스



Non-Diegetic interface :   
게임세계 외부에 랜더링되는 인터페이스, 플레이어에게만 보임, HUD에 표시되는 정보

메타 UI :   
표현은 게임 세계에 존재할 수 있지만 반드시 플레이어를 위해 공간적, 정보적으로 시각화 되는 것이 아닌 몰입감과 대락적인 정보를 전달하기 위한 표현의 일부  
ex) 화면에 튀기는 피, 체력이 낮아지면 빨갛게 물드는 화면

UI 기본이론

UI란 사용자가 제품을 사용할 때, 마주하는 디자인, 레이아웃, 기술적인 부분 등을 말한다.

디자인의 구성 요소인 폰트, 컬러, 줄 간격 등 상세한 요소가 포함된다.

기술적인 부분은 반응성, 입출력 단계, 애니메이션 효과 등 여러 부분을 포함한다.

UI는 다양한 사용자가 사용할 수 있도록 보편성을 지녀야하며, 불편함이 없어야 한다.

좋은 UI는 대부분의 사용자가 높은 만족도를 지녀야하며, 어쩔 수 없이 발생하는 주관적인 불만족을 최소화시켜야 한다.

UX 란?

사용자가 어떤 제품, 시스템, 서비스 등을 직접적 혹은 간접적으로 이용하면서 느끼는 반응과 행동 등의 경험을 총체적으로 설계하는 것

정형화된 틀이 있는 UI와는 다르게 사용자의 경험을 중점적으로 돌아가기 때문에(이용자의 성향을 세대와 계층에 따라서 계속해서 변화함) 틀이 없고 객관적인 데이터와 통계를 바탕으로 접근하면 좋음

상차림을 예시로 든다면…

UI : 접시의 형태, 나이프와 포크의 디자인, 음식의 색감 등을 의미한다, 시각적인 부분  
UX : 식기와 접시의 배치, 음식에 곁들이는 소스와 양념의 추가, 접시에 담기는 음식의 크기 등, 체험적인 부분

SIMPLE IS BEST  
미니멀리즘, 핵심기능만을 남기기

VUI : 음성으로 컨트롤하는 인터페이스

Omnichannel UX : 다양한 기기와 채널에서도 하나의 서비스를 이용하는 것처럼 느낄 수 있도록 하는 것

UI 디자인 단계

1. 제품조사
2. 주요 자용자 그룹을 식별하고 대표 페르소나를 만들 것. 해당 페르소나의 일상을 그리고 어떻게 제품을 사용하게 되는지에 대한 시나리오를 짜보기
3. 정보 아키텍쳐 정의, 어떤 정보나 데이터에 접근하기 위해 필요한 depth를 구상
4. 와이어프레임 제작, 백지 ppt같은거 대략적인 배치와 구성에 필요한 구조
5. 프로토타입 제작
6. 제품 테스트 / UX디자이너는 테스트를 통해 사용자가 어떤 경험을 하는지

UX 심리 7요인

1. 시각 / 집중 : 가장 기본적이고 중요한 것, 주관의 영역이라 디자이너와 사용자의 선호가 다를 수 있으니 주의
2. ㅇ
3. ㅇ
4. 언어
5. 감정 : 사용자는 두려움, 및 좌절감을 느끼며 주저한다. 특정 구매or행동이 자신의 목표와 명성에 어떤 영향을 미칠지 궁금해한다.
6. 의사결정 : 의사결정을 내리고 구매를 하기까지 도움을 줄 수 있는 요소

HCI 이론

인터페이스 디자인이란 디지털 시스템의 입출력 장치의 모양과 내용을 디자인하는 것이다

드롭다운 리스트(Drop-down List)

* 객관식 문제에서 볼 수 있는 선택지와 유사한 개념

입력필드(Text Input Field)

* 사용자가 키보드로 직접 텍스트를 입력하는 텍스트 상자

콤보 박스(Combo Box)

* 드롭다운 리스트 + 입력 필드

라디오 버튼(radio Button)

* 여러 항목 중 하나의 항목만을 선택 할 수 있는 것

체크 박스(Check Box)

* 여러 항목 중 중복된 항목을 선택할 수 있음

토클(Toggle Button / Switch)

* 스위치를 좌우로 움직이거나 영역을 선택하여 On/Off 상태를 조절하는 것

버트콘(Butcon) / 툴바(Tool Bar)

* 버튼 기능이 있는 아이콘
* 버트콘을 Bar 형태로 모아놓은 것이 툴바

툴팁(Tooltip)

* 마우스 롤 오버 시 해당 메뉴에 대한 설명이 말 풍선 형태로 제공되는 것

스피너(Spinner)

* 숫자 입력 컨트롤, 필드에 입력된 숫자 값을 화살표로 조절 가능

다이얼(Dial)

* 놉(Knob)이라고도 함
* 실제 아날로그 기기의 조작잠을 재현하고자 하는 목표에서 개발됨

슬라이더(Slider)

* 입력의 최소와 최대치를 직관적으로 알 수 있음
* 드래그로 조절

드롭다운 메뉴(Drop-down Menu)

* 메뉴의 제목을 선택하면 메뉴가 아래로 펼쳐짐

리본 메뉴(Ribbon Menu)

* 아이콘과 메뉴명을 함께 표기
* Office 최신버전의 메뉴표시 방법

트리 메뉴(Tree Menu)

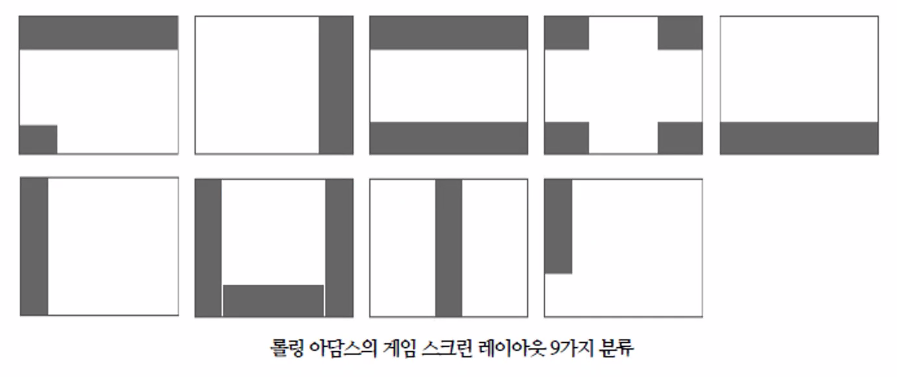
* 목록을 계층 구조를 이용해 표시
* Like 윈도우 탐색기

대화상자(Dialog Box)

* 지시 사항이나, 결정을 묻기위해 사용자가 하던 일을 잠시 멈추게 하는 창
* 팝업(Pop-up)이라고도 함

가상 키보드 입력상태(모달 상태)

롤링 아담스의 게임 스트린 레이아웃 9가지 분류



가장 중요한 정보는 스크린 좌측 하단에 위치하는 경우가 많음

복잡한 인터페이스는 넓은 층의 고객층을 유지하는데 도움되지 X

유동형 vs 고정형

전투화면과 정보화면의 분리

* 모바일 환경으로 전환되면서 스크린 사이즈가 현격히 줄어들었음 → 콘텐츠별로 화면의 모드 전환이 이루어지는 특징을 가지게 됨

스마트 디바이스의 특성

* N-Screen환경(다중 디바이스, 다중 플랫폼 지원)에 대응할 수 있는 설계

UI의 재활용

* 차기 콘텐츠 개발에 있어서 인터페이스의 재활용과 확장성을 고려

Jesse James Garrett : 사용자의 경험을 5가지 계층으로 분류  
각 계층을 하위 계층에 종속됨(하위 계층이 어그러지면 상위 계층도 의미가 없어진다)

* 1계층 전략(Strategy) : 사용자와 제공자가 추구하는 목표
* 2계층 범위(Scope) : 어떠한 기능을 포함하고 어떠한 기능은 포함하지 않아야 하는가, 세부적인 요구사항
* 3계층 구조(Structure) : 사용자가 제품과 상호작용하는 방식, 사용자의 행동에 따른 제품의 응답과 반응
* 4계층 윤곽(Skeleton) : 시각적인 윤곽, 와이어프레임
* 5계층 표면(Surface) : 디자인적인 부분들, 시각적인 감각

사용자의 전문성을 고려할 것

* 초보자는 다층의 레이어(펼쳐진 느낌), 전문가는 레이어를 적게(알아서 찾음) 설계할 것
* 기능의 익숙함만이 아닌, 이용자의 사회적 배경을 고려할 것

네비게이션 설계

* 메뉴 그룹핑과 계층구조는 명확하고, 가시적으로
* Depth를 적절하게(3단계가 적절)
* 사용자 수준에 맞는 용어를 사용하기
* 상, 하위 메뉴간 관계가 적절한가?
* 내가 어디에 있나?, 내가 어디에 간적이 있나?, 어디로 갈 수 있나?
* 앱의 네비게이션 시스템이 운영체제의 기본 네비게이션 시스템과 혼동되지 않은가?

Android design guidelines

* 표면, 효과, 음향 등의 다양한 방법으로 사용자를 즐겁게 하기
* 실제 물건은 버튼이나 메뉴보다 흥미롭다(텍스트보다는 이미지 아이콘)
* 글은 간단히 작성할 것, 글보다는 사진을 사용할 것
* 조건은 제시하되 마지막 선택은 사용자가 하도록하기
* 필요한 경우에만 메뉴를 표시하기, 중요한 경우에만 사용자에게 인터럽트를 걸기
* 사용자의 기록을 저장하기(최근에 사용했던 메뉴를 다시 표시 하는 등)
* 초보자를 전문가처럼 느끼게 만들기(너무 자세한 설명은 ㄴㄴ)
* 중요한 것은 빠르게 실행할 수 있도록

한글 서체의 경우 ‘고딕’계열과 ‘명조’계열, 영문의 경우 돌기가있는 ‘셰리프’와 돌기가 없는 ‘산 셰리프’ ← 이 서체들의 스타일에서 크게 벗어나지 않는 것이 좋음

LinearLayout : 위젯을 orientation 옵션에따라서 왼쪽 위에서 아래로, 혹은 왼쪽위에서 오른쪽으로 배치함

FrameLayout : 하나의 자식 view 위젯만 표시할 때 사용하는 Layout 클래스(하나의 화면에 여려개의 콘텐츠가 겹쳐져서 들어감)

TableLayout : 자식 view 위젯들이 행과 열의 형태로 배치, 크기에 관계없이 정렬하려고 함  
완전히 표 같은 형태로 들어가면 이거임 / 실제로 표를 보이는 상황이 아니라면 지양되고있음

GridLayout : 표와는 좀 다름, Layout 배치에 알맞게 개발되었음(‘배치’에 조금 더 중점을 둔 형태임 타 콘텐츠 영역을 침범할 수 있음)

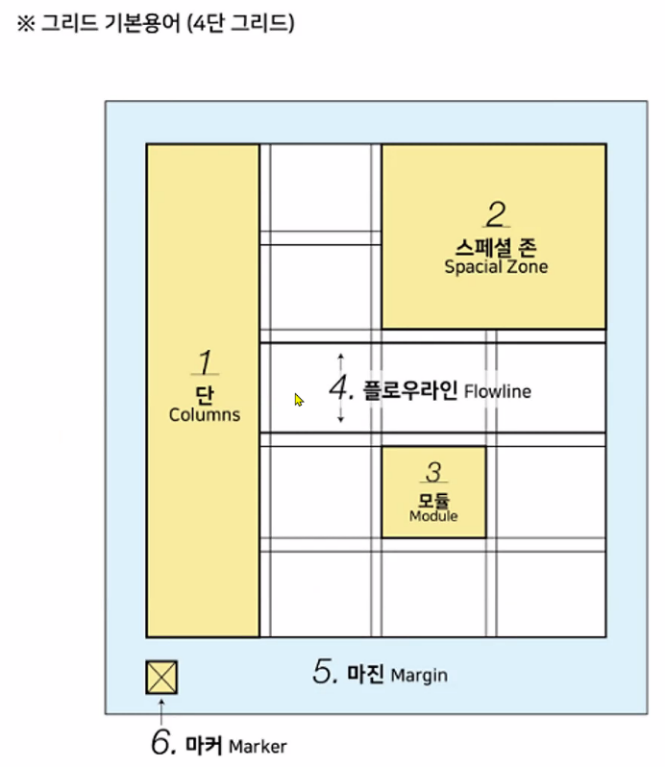
RelativeLayout : 위젯들을 상대적인 위치에 배치할 수 있음

Wrap Content : 담고있는 콘텐츠의 크기에 맞추어서 사이즈 조절

Match Parent : 부모의 크기에 맞춤

Grid 구조

1. 좌표에 의해 지정된 점을 수직선과 수평선으로 균일하게 배치한 그물망이다.
2. 수직선은 단을 나누고 수평선은 이미지의 크기를 결정한다



Shape : 각 컴포넌트의 주목도

Status : 각 컴포넌트의 현 상태를 알려준다.

* Nomal(일반)
* Focus(집중)
* Loading(로딩중)
* Hover(마우스오버)
* Active(눌렀을 때)
* Disabled(비활성화)

폰트 크기

* 구글 : 14~16px 권장
* 에플 : 14px 이상 권장

Em : 해당 폰트의 대문자 M의 너비를 기준으로

Ex : 해당 폰트의 소문자 x의 높이를 기준으로

Px : pixel, 출력장치에 따라서 상대적인 크기를 가진다

% : 기본 글꼴의 크기를 기준으로 상대적인 크기

Pt : point, 문서 프로그램에서 사용되는 단위