



ALL ABOUT BUTTON DESIGN

목차

- 버튼의 개요
- 정적인 버튼 상태
- 선택한 버튼 상태
- 눌린 버튼 상태
- 동적인 버튼 상태
- 선택 불가 버튼 상태
- 애니메이션 버튼
- 소리
- 프로그램과 버튼



버튼의 개요

○ 버튼의 정의

- 누르거나 하여 선택할 수 있도록 한 인터페이스.

○ 버튼의 장점

- 우리 주변에 이미 많이 사용되고 있다.
 - 리모트 컨트롤러, TV, 비상벨 등등...
일반 사람들에게 이미 친숙하다.
- 표현의 용이함
 - 메뉴의 보여짐과 선택문을 선택했을 시 보여지는 비주얼한 부분.

○ 내용에 앞서서.

- 버튼이 유용하기는 하지만 게임마다 반드시 필요한 것은 아니다.
- 버튼은 용도에 따라서 인터페이스 자체에 도움을 주기도 하지만 게임을 플레이 하는데 도움을 줄 수 있다.



정적인 버튼 상태

○ 정의

- 아무 것도 눌리지 않은 상태를 나타냄
- 아무런 움직임도 없다.

○ 표현 방법

- 누를 수 있는 것이라고 플레이어가 느낄 수 있어야 한다.
 - 버튼에 음영을 집어 넣어 튀어나온 것처럼 표현

안중은 버튼

중은 버튼

- 눈에 띄이게 함.

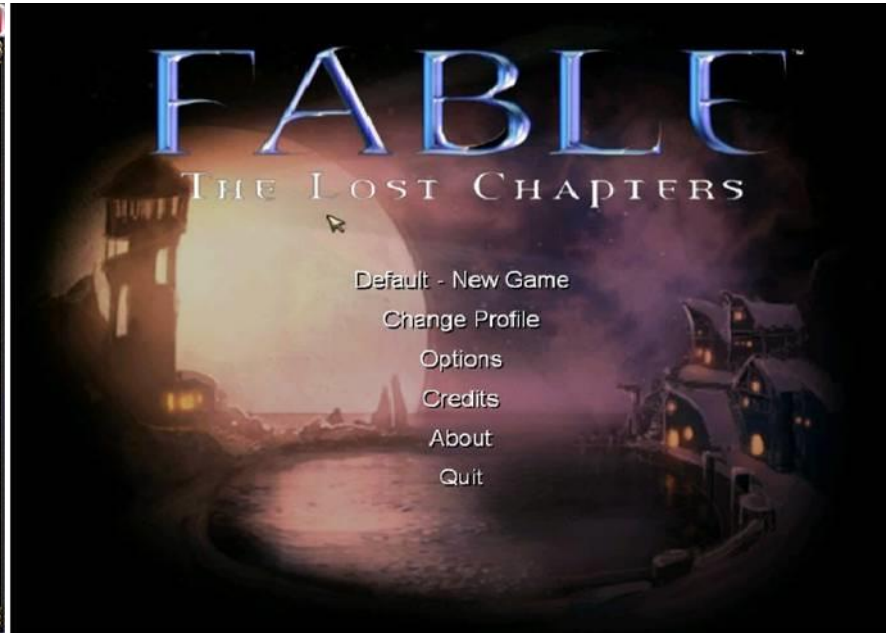
○ 설계 시 기억해야 할 사항

- 메뉴를 선택했을 시와 눌렸을 경우를 모두 대비해야 한다.



정적인 버튼 상태

○ 게임에서의 예



좋은 버튼 사용

눌러야 한다는 것을 바로 알 수 가 있음

나쁜 버튼 사용

색이 명확하지 않고 누를 수 있는 것인지
아닌지 건드려 보지 않는 이상 알 수 없음



선택한 버튼 상태

○ 정의

- 사용자가 메뉴를 선택한 상태(아직 누른 상태가 아님)

○ 표현 방법

- 정적인 버튼 상태와 달라야 함
 - 대비색이나 눈에 띄는 색을 통한 변화
 - 오브젝트의 추가
 - 애니메이션 사용

○ 설계 시 요구 사항

- 배경과 선택되지 않은 메뉴의 관계를 잘 보아야 함
- 가장 눈에 잘 보일 수 있도록 할 것.



선택한 버튼 상태

○ 게임에서의 예



옅도록 밝은 색을
사용하여 눈에 잘
띄이게 함



눌린 버튼 상태

정의

- 사용자가 메뉴의 기능을 사용하기 위해 누른 상태

표현 방법

- 눌렀다는 것을 확실하게 보여줌
 - 버튼의 높임을 음영의 역변화를 통해 표현



- 애니메이션 사용하여 눌렀다는 것을 알려 줌

설계 시 중요한 것

- 기능이 돌아간다는 것을 기억해라. 즉, 선택 후 기능이 사용자에게 보여주기 까지 시간이 걸릴지도 모른다. 만약 수초가 지난 뒤에 보여지게 될 것이라면 로딩 바나 최소한 어떠한 변화를 주어야 한다.
Because) 즉각적인 반응이 아니면 사용자는 되지 않는 버튼으로 인지



동적인 버튼 상태

○ 정의

- 메뉴를 선택 했을 시 효과가 있는 상태

○ 표현방법

- 버튼 주변에 테두리를 밝게 함



- 버튼 옆에 추가적인 이미지를 넣어 줌.



동적인 버튼 상태

○ 표현 시 주의 사항

- 실제로 눌렀을 시의 효과보다 비주얼이 강해서 눈 안된다.

○ 게임에서의 예



캐릭터 얼굴을 확대
시켜 줌으로써 적당한
효과를 나타냄

선택 불가 버튼 상태

정의

- 사용할 수 없는 버튼을 나타냄

표현 방법

- 가장 일반 적인 것이 어둡게 하여 선택할 수 없게 표현
 - 그래픽으로 하지 않고 프로그래밍으로써 표현 할 수 있다.
- 그 외에도 사용할 수 없다는 오브젝트를 붙일 수도 있음

주의 사항

- 사용하지 못한다 할지라도 어떠한 기능을 가지고 있는지 알 수 있도록 보일 수 있도록 하라.

Example) 어둡게 표현 할 때 회색을 50% 가미하면 버튼도 보이고 사용할 수 없다는 것을 표현할 수도 있다.



선택 불가 버튼 상태

○ 게임에서의 예



사용 가능

사용 불가



애니메이션의 사용

정의

- 버튼을 움직이게 하여 사용자에게 어필을 강하게 하고자 함

기술론

- 몇 개의 장면을 만들어 순환 식으로 계속 보여 주는 방법
 - 각각의 프레임 그리기
 - 몇 개의 프레임을 만들어 놓고 마지막 프레임까지 순차적으로 보여준 뒤 다시 첫 프레임으로 되돌아 가는 방식
 - 알고리즘 사용
 - 프로그램 언어를 이용하여 이미지나(3D의 경우 폴리곤)을 변형 하거나 효과를 부여 함으로써 그래픽 용량을 줄이고 효과를 줄 수 있다.
- 움직임
 - 버튼의 위치를 움직이게 하는 방식



○ 게임에서의 예



동영상임, 화면 클릭 시 재생

○ 애니메이션 표현 시 유념 사항

- 역동적인 것은 사용자에게 잘 보일 수 있다는 장점이 있지만 반대로 역으로 작용 할 수 있다.
 - 게임이 역동적으로 진행되는 상태에서 버튼 마저 움직인다면 혼란을 가져올 가능성이 있다.
 - 메뉴에서 백그라운드가 동적으로 움직이는 상황이라면 메뉴가 그에 묻히지 않도록 유의 해야 함
- 재사용 할 수 있는 부분은 최대한 활용 할 것.
 - 게임 안에서 인터페이스의 용량 부분이 많이 차지 해서 좋을 것은 없음
 - 다양한 인터페이스는 시간 대비 능률적이지 못하다.
 - 사용자는 일관된 인터페이스를 더 친숙하게 사용할 수 있음.
(학습을 덜 시키고 클릭을 덜 할 수록 좋은 인터페이스)



사운드

○ 소리의 중요성

- 사람은 보는 것보다 듣는 것이 인지가 더 빠름
- 사용자로 하여금 버튼을 눌렀다는 것을 확신 시키는 역할.
- 느낌을 더 잘 전달 할 수 있음

○ 소리의 제작

- 게임의 컨셉트와 맞추도록 노력하라.
 - 예를 들어 레이싱 게임이라면 버튼을 눌렀을 때 엔진음이나 크러션과 같은 음을 사용하면 효과적일 것이다.
- 많이 사용한다고 좋은 것은 아니다.
 - 과도한 사용은 오히려 혼란스러워 질 수 있음.
- 배경음을 염두하라.
 - 배경음과 어울리지 않는 효과음은 게임 전체의 소리를 망친다.



프로그램과 버튼

○ 프로그래머와의 상의

- 결과물은 프로그램에 의해서 창출
 - 용량과 속도, 그리고 기술적인 문제에 의해서 구현이 안 되는 경우가 있다. 특히 3D로 인터페이스를 구성할 때 더욱 그러하다.
 - 쓸데없는 그래픽의 양과 작업 시간의 감소
 - 알고리즘을 통해서 효과를 대신 할 수 있는 것들이 있다.
 - 대표적인 효과들
 - 확대와 축소
 - 불투명 및 투명 효과
 - 이동
 - 색깔 변화
- 그 외에도 더 많은 것이 있다.



추가적인 내용

- 3D로 구성된 버튼 인터페이스의 예



지구본을 돌려서 원하는 버튼을 찾아낼 수 있다.

