

PART

01

소셜 게임 개발 기술



순식간에 세상을 석권하며 웹 애플리케이션 개발자에게 새로 익혀야 할 영역 중 하나로 떠오른 소셜 게임!

처음에는 피쳐폰 전용 게임이 큰 성장세를 보였고, 2010년 말부터는 스마트폰에서 인기를 얻고 있다.

DeNA는 특히 모바일 소셜 게임 개발에 주력하여 Mobage 플랫폼에서 게임을 전개하고 있다. 1부에서는 소셜 게임이란 무엇인가와 왜 성장하고 있는가에 대한 개론을 시작으로 피쳐폰과 스마트폰, 브라우저와 애플리케이션 각각에서 소셜 게임 개발의 테크닉에 대해 깊이 있는 설명을 할 것이다.

CHAPTER 1 소셜 게임 개론

CHAPTER 2 브라우저 기반의 소셜 게임: 피쳐폰

CHAPTER 3 브라우저 기반의 소셜 게임: 스마트폰

CHAPTER 4 애플리케이션 기반의 소셜 게임 개발

CHAPTER

01

소셜 게임 개론

이 장에서는 소셜 게임에 대해 간략히 설명하겠다. 또한 소셜 게임의 운용을 시스템과 서비스적인 관점에서 설명해 나가겠다.

1-1

소셜 게임이란?

2010년 ~ 2011년에 걸쳐 일본에서 가장 많이 팔린 콘솔 게임이라고 하면 (주)캡콤의 '몬스터 헌터 3판'이 아닐까? 2011년 7월 시점에 500만 카피가 넘어섰으니 말이다. PSP(플레이스테이션 포터블)의 일본 내 누적 판매 대수가 2011년 4월 시점에서 약 1,700만 대(엔터 브레인 2011년 4월 발표)로 추정되므로, PSP 사용자 3명 중 1명이 '몬스터 헌터 3판'을 즐기고 있는 것이다.

한편, (주)디엔에이(이하 DeNA)의 소셜 게임 '괴도 로얄'은 2009년 10월에 시작하여 2011년 6월에는 사용자 수 1,000만 명 이상의 규모가 되었다.

소셜 게임 '괴도 로얄'은 DeNA가 운영하는 소셜 게임 플랫폼 모바게(이하 Mobage)에서 즐길 수 있다. Mobage는 2011년 3월말 현재 총 사용자 수가 2,700만 명이므로 '몬스터 헌터 3판'과 동일하게 3명 중 1명이 즐기고 있다는 사실을 알 수 있다.

일본 최대의 소셜 네트워크 서비스(이하 SNS)인 mixi(이하 믹시)는 2009년 8월에 일본에서는 처음으로 소셜 애플리케이션이 제공할 수 있는 플랫폼 「믹시 애플리」를 시작으로 현재에는 외부 게임 회사로부터 다양한 소셜 게임을 배포하고 있다. 마찬가지로

가지로 DeNA는 2010년 1월에 「Mobage API」를, GREE는 2010년 6월에 「GREE 애플리」를 시작하고 있다. 표 1.1에 일본을 중심으로 소셜 게임의 역사를 연대별로 정리하였다.

판매 대수가 많고 총 사용자 수가 많기 때문에 히트작 게임이 나오기 쉽지 않을까 라고 생각하기 쉽지만 꼭 그렇지만은 않다. 실제로 현재의 시점에서는 수천 개 이상의 게임 중 히트작은 상위 몇 개뿐이다.

● 표 1.1 소셜 게임의 역사

시기	주요 내역	국가
2007년 5월	GREE가 '낙시 ★ 스타'를 시작	일본
2008년 1월	페이스북이 「Facebook API」를 공개	미국
2009년 6월	징가(Zynga)가 'Farm Ville'을 시작	미국
2009년 8월	믹시가 「믹시 애플리」를 시작	일본
	일본 Reko가 '선사인 목장'을 시작	일본
2009년 10월	DeNA가 Mobage 타운*1에서 '괴도 로얄'을 시작	일본
2010년 1월	DeNA가 Mobage 타운*1에서 「Mobage API」를 시작	일본
2010년 6월	GREE가 「GREE 애플리」를 시작	일본

*1 현재는 Mobage

소셜 게임이란 무엇인가?

소셜 게임은 SNS에서 사용자끼리 서로 경쟁하거나 협력하여 정보 교환을 하면서 함께 즐길 수 있는 상호 교류 기능을 지닌 게임이다.

또한 기본적으로 무료로 즐길 수 있는 게임이긴 하나, 보다 유리하게 게임을 진행 하거나 시간 단축 등 더 업그레이드하기 위해서 유료 아이템을 사용자가 구입할 수 있는 「아이템 과금」이라는 방식이 있다. 그리고 이를 통해 게임 제공 회사는 수익을 창출하고 있다.

2012년 1월 현재, Facebook(이하 페이스북)에서 가장 즐겨 사용되는 소셜 게임은 Zynga(이하 징가)의 CityVille 및 FarmVille로, 무려 한 달에 5,000만 명 이상이 이

용하고 있다. 징가의 Ville 시리즈는 도시와 농장, 폐허 등을 개척하여 돈을 모아 빌딩(건물)을 짓거나 농작물을 재배하거나 해서 돈을 모은다. 혼자서는 그것을 반복하기만 하는 단순한 게임이지만, 페이스북에서 친구를 초대하여 친구의 농장을 경작하거나 작업함으로써 게임에 필요한 유리한 경험을 하거나 친구를 끌어들이으로써 보다 높은 경험치와 점수를 얻을 수 있다. 사용자가 사용자를 초대함으로써 단기간에 사용자를 대폭 늘릴 수도 있다.

또한 게임의 경험치를 겨루거나 친구들과끼리 협력 플레이를 통해 보다 나은 건물이 지어지고 그룹 간 경쟁을 함으로써 게임의 재미를 확장시켜 나간다.

기존의 온라인 게임과 같이 장시간 플레이가 아닌, 가볍게 5분이나 10분 정도 플레이할 수도 있는 것도 소셜 게임의 특징이다. 결코 높은 수준의 그래픽은 아니지만, 사람과 사람 간의 부담 없는 인간 관계와 부담 없는 플레이가 결과적으로 편안하고 오랫동안 질리지 않게 플레이할 수 있는 연결점이 된다.

소셜 게임과 온라인 게임

위키피디아 영어판에서는 「Social Game」이라는 단어는 존재하지 않고, 「Social Network Game」이라는 단어가 게재되어 있다. 거기에서의 「Social Network Game」의 정의는 「소셜 네트워크 상에서 주로 배포하는 온라인 게임의 일종」이라고 되어 있다.

이와 동일하게 「Online Game」의 정의는 「컴퓨터 네트워크(지금은 대부분 인터넷을 가리킴)를 통해 플레이하는 게임」을 말한다. 넓은 의미에서는 소셜 게임도 온라인 게임의 일종(인터넷을 통해 하고 있기 때문)이라고 할 수 있지만, 큰 차이점은 SNS 플랫폼 상에서 전개되는 게임인가 아닌가에 따라 다르다고 정의할 수 있다. 즉, 소셜 게임은 SNS 안에서 만들어진 서비스의 하나로, 사람과 사람과의 연결을 기반으로 한 게임인데 반해, 기존의 온라인 게임은 비디오 게임에 네트워크 기능이 더해진 것이라고 말할 수 있다.

소셜 게임의 기본 사이클

소셜 게임뿐만 아니라 게임의 기본 동작에는 「액션」→「반응이 온다」→「판단한다」→「다음 액션」이라는 과정이 있다. 예를 들어, 「괴도 로얄」에서는 「미션을 수행한다」→「보물을 얻는다」→「앞으로 하나만 더 얻으면 미션 완료」→「배틀을 통해 얻어낸다」라는 식의 순서를 갖고 있다.

1-2

소셜 게임 운용

지난 몇 년 동안 소셜 게임의 이용자 수는 급증하여 DeNA가 운영하는 Mobage에서의 사용자 수는 약 4,000만 명(2012년 3월말 시점)에 도달하였다. 요즘은 인기 있는 소셜 게임이라면 이용자 수가 순식간에 수백만 명이 넘는다.

이러한 상황에서 안정적인 시스템 운용을 실현하는 데 있어 가장 중요한 것은 급증하는 트래픽에 대해 신속하게 시스템을 확장시키는 것이다. 그러기 위해서는 시스템 운용의 각 계층에서 다양한 기술적 과제를 극복해 나갈 필요가 있는데, 이는 엔지니어에게 매우 도전적인 일이자 자신의 실력을 발휘할 수 있는 기회라고 생각한다.

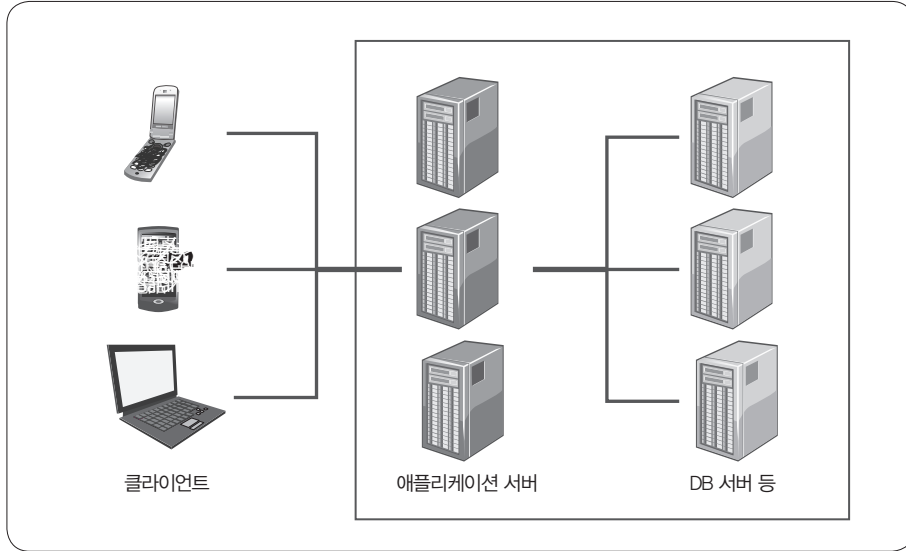
이 절에서는 소셜 게임 운용을 시스템적인 측면에서 어떻게 지원하는가에 대한 개요 수준의 설명을 할 것이다. 각 파트에서 구체적으로 어떠한 노력이 이루어지고 있는지에 대해서는 뒷장에서 설명할 것이다.

시스템 측면의 개선뿐만 아니라 서비스 측면에서도 지속적으로 개선해 나가는 것이 성공의 비결이라고 생각한다. 순간적으로 떠오른 기획(혹은 아이디어)으로는 좀처럼 성공하지 못한다. 「감성」에만 의존하지 않고 사용자의 이용 상황을 분석하여 최적의 시행책을 실시함으로써 게임을 진화시킬 수 있다.

시스템 구성

그림 1.1에 시스템 구성도를 나타내었다. 자세한 것은 이후를 참조하길 바란다.

● 그림 1.1 시스템 구성



클라이언트 사이드

소셜 게임을 시작할 초기에는 주로 PC와 피쳐폰을 위한 웹 애플리케이션으로 배포되었기에 매우 간단한 것이었다. 스마트폰 이용자가 증가함에 따라 보다 고급스런 조작성을 실현하기 위해 클라이언트 애플리케이션으로도 소셜 게임이 제공되기 시작했다. 이렇듯 클라이언트 사이드 기술 영역도 점차 확대해 나가고 있는 추세다.

여기서 시스템 측면에서의 과제라면 사용자가 가지고 있는 디바이스의 실행 환경의 차이를 고려해서 게임을 제공하는 것인데, 이 문제에 대해서는 몇 가지 기술적인 접근 방법이 존재한다.

웹 애플리케이션에서의 게임 배포에 대해서는 2장과 3장을, 클라이언트 애플리케이션에서의 게임 배포에 대해서는 4장을 참조하길 바란다.

애플리케이션 서버

사용자로부터의 요청을 수신하여 데이터베이스 액세스를 실행하고 사용자에게 응답을 전송하는 레이어를 애플리케이션 서버라고 부른다.

이 애플리케이션 서버 측에서는 이용자가 증가함에 따라 사용자 액세스 수의 집중과 처리할 데이터가 매우 커짐에 따라 성능이 저하되는 문제 등 운용 단계에서의 여러 가지 기술적인 문제가 있다. 이러한 과제를 해결하는 애플리케이션의 설계 및 개발이 요구되는데, 자세한 내용은 8, 9, 10장을 참조하길 바란다.

데이터베이스, 네트워크 인프라, 시스템 아키텍처

게임 데이터와 사용자 데이터, 게임 분석을 목적으로 이력 데이터 등 소셜 게임의 운용은 다양한 데이터를 처리한다. 그런데 이러한 데이터 양은 이용자 수의 증가에 따라 확대되어 간다. 이 대용량 데이터를 곧바로 처리해 나가기 위한 데이터베이스의 설계 및 운용이 요구되어지는 것이다.

마찬가지로 이용자 수의 증가에 따라 사용자와 서버 사이 그리고 서버들 사이에서 상호 작용하는 트래픽도 확대되어 가는데, 이 대용량 트래픽을 안정적으로 처리하도록 하는 네트워크 인프라를 구축할 필요가 있다.

이러한 각종 미들웨어 및 네트워크 인프라를 통합적으로 관리하여 어떤 상황이 발생해도 안정적인 서비스를 제공할 수 있도록 시스템 아키텍처를 설계하고 운영해야 한다.

데이터베이스의 운용 방법은 6장을, 네트워크 인프라의 운용 방법은 7장을, 시스템 전체의 아키텍처에 대해서는 5장과 6장을 참조하길 바란다.



소셜 게임 운용에 있어 시스템 구성의 개요를 설명하였다. DeNA가 운영하는 Mobage는 일본에서도 최대의 사용자 수와 트래픽을 보유하고 있으며, 각 기술 분야에서 다양한 도전을 하고 있다.

CHAPTER

02

브라우저 기반의 소셜 게임: 피쳐폰

이 장에서는 브라우저 상에서 동작하는 웹 애플리케이션의 소셜 게임 개발에 대해서 설명할 것이다.

2-1

피쳐폰이란?

피쳐폰은 통신 기능 이외에 고급 기능을 많이 탑재한 휴대 전화를 말한다. 스마트폰이 등장할 때까지는 휴대폰이라고 하면 이 피쳐폰(기능폰)이 주류였다.

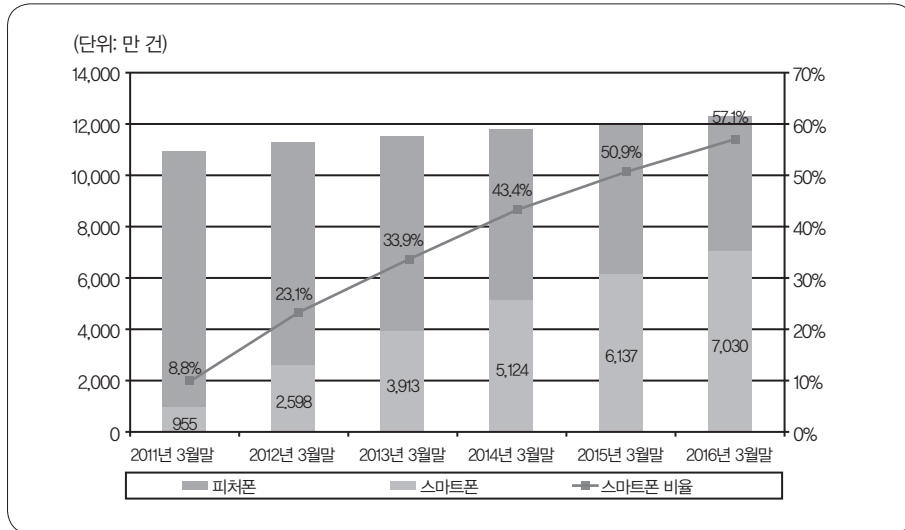
DeNA에서는 ‘괴도 로얄’을 비롯해 2009년부터 사내에서 소셜 게임을 개발하였다. 그 대다수가 피쳐폰 전용으로 개발된 것인데, 이것이 일본 소셜 게임의 붐을 일으킨 계기가 되었다. 피쳐폰은 docomo/au/SoftBank 등의 통신사업자에 따라 사양이 다르며, 단말기 제조사에 따라 동작도 다르다. 모든 단말기에서 정상적으로 작동하는 웹 애플리케이션을 개발하는 것은 참으로 힘든 작업이다. 하지만 피쳐폰에서 웹 애플리케이션이 개발되게 된 이후로도 몇 년이 경과하였기 때문에, 조사해보면 효율적으로 개발할 수 있는 환경을 구축할 수 있을 거라 생각한다. DeNA가 브라우저 기반의 소셜 게임을 개발하는 데 있어서 주의하고 있는 점이나 연구하고 있는 점에 대해 소개하겠다.

시장 규모와 장래성

현재 피쳐폰은 스마트폰의 등장으로 출하량이 감소 추세에 있지만, 이용자는 여전히 많아 소셜 게임 시장에서 아직 적지 않은 비중을 차지하고 있다(그림 2.1).

하지만 앞으로는 스마트폰의 보급 확대에 따라 보다 스마트폰 전용의 소셜 게임이 중심이 될 것으로 예측된다. 완전히 스마트폰으로 전환되면, 피쳐폰 대응을 생각할 필요가 없어 개발자가 기뻐할 수 있게 될지도 모르겠다. 그러나 그림을 보면 알겠지만, 완전히 스마트폰으로 이행하는 데는 몇 년은 더 걸릴 것이라는 것이 지배적이다. 따라서 소셜 게임은 사용자를 극대화하기 위해 아직은 피쳐폰이나 스마트폰 모두를 지원해야 한다.

● 그림 2.1 스마트폰과 피쳐폰의 계약 수와 스마트폰 비율(MM 종합연구소 예상)



2-2

피쳐폰 전용 웹 애플리케이션 개발

DeNA가 개발하는 소셜 게임은 소셜 게임 플랫폼인 Mobage와 동일 개발 프레임워크에서 개발하고 있다. 이 개발 프레임워크는 MobaSiF라는 이름으로 2008년에 오픈 소스화되었다.

MobaSiF는 모바일 전용으로 개발된 프레임워크로, 통신사업자에 의한 사양의 차이를 흡수하고 있기 때문에 간단히 피쳐폰 전용의 웹 애플리케이션을 개발할 수 있다.

피쳐폰 전용의 브라우저 기반 소셜 게임을 개발하기 위해서는 피쳐폰 특유의 기술 요소를 파악하고 있어야 한다. 특유의 기술 요소로서는 이모티콘 처리, 단말기 인증, 통신사업자 및 단말기 기종의 판별을 들 수 있다. 이들에 대해 간략하게 소개해 나가겠지만, 최신 정보는 각 이동 통신사의 웹 페이지에서 확인하길 바란다(이후 설명은 일본의 예라 우리나라에서의 개발 환경과는 다르다. 그러나 참고할 만한 요소들도 많으니 읽어보기를 추천한다_옮긴이).

이모티콘과 문자 인코딩

이모티콘 표현 방법은 휴대폰 통신사업자에 따라 다르다. Mobage에서는 통신사업자마다의 문자 인코딩을 표 2.1과 같이 취급하고 있다.

● 표 2.1 Mobage 내의 통신사업자별 문자 인코딩

통신사업자	문자 코드	비고
docomo	Shift-JIS	구 단말에서는 UTF-8 비대응
au	Shift-JIS	구 단말에서는 UTF-8 비대응
SoftBank	UTF-8	일부 3G 단말에서 Shift-JIS가 비대응

HTML에 이모티콘을 표시하려면 다음의 방법이 있다.

- docomo

- ① Shift-JIS 이진 코드 표현(1 이모티콘 2바이트)
- ② Unicode의 텍스트 표현(예: 「晴れ」 = )
- ③ UTF-8 표현

- au

- ① 태그를 이용한다
- ② i 모드와의 호환성을 이용한다(일부 이모티콘은 비슷하게 표시된다)
- ③ KDDI 이모티콘용 Shift-JIS

- SoftBank

- ① 숫자 문자 참조(예: 「太陽」 = &xE04A;)
- ② UTF-8 표현(예: 「太陽」 = 16진수 표기로 EE 81 8A)

또한 사용자로부터 FORM 태그를 통해 이모티콘을 전송하는 경우에는 표 2.2를 파악해둘 필요가 있다.

- 표 2.2 FORM으로부터의 이모티콘 송신

통신사업자	비고
docomo	Shift-JIS 페이지로부터는 Shift-JIS의 이진 코드 표현 UTF-8 페이지로부터는 UTF-8 표현
au	Shift-JIS 페이지로부터는 KDDI 이모티콘용 Shift-JIS 코드
SoftBank	UTF-8 페이지로부터는 SoftBank의 UTF-8 표현

피쳐폰 전용의 브라우저 기반 소셜 게임을 개발할 때 이러한 것에 주의하여 이모티콘을 취급해야 한다.

DeNA의 소셜 게임 개발에 있어서는 프레임워크로서 사용하고 있는 MobaSiF가 취급하기 쉽게 대응하고 있다. MobaSiF에서의 이모티콘 변환에 대해 설명하겠다.

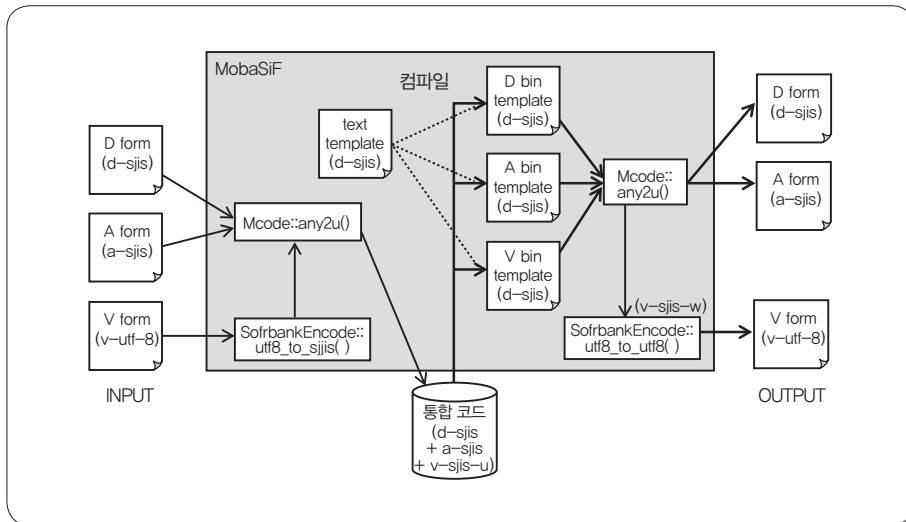
MobaSiF에서는 이모티콘 변환 Mcode와 SofbankEncode 모듈을 사용한다. 이러한 모듈은 XS(XS는 Perl과 (혹은 Perl과 함께 사용하고 싶은) C 코드(혹은 C 라이브러

리) 사이의 확장 인터페이스를 만드는 인터페이스 기술 파일 포맷이다_옮긴이)로 구현되어 고속으로 처리할 수 있는 특징이 있다. MobaSiF에서는 다음과 같은 것을 쉽게 구현할 수 있다.

- 사용자가 FORM에서 이모티콘 입력 가능
- 통신사업자마다 이모티콘의 사용 구분이 불필요
- 같은 통신사업자의 사용자 간에는 입력된 이모티콘이 그대로 표시 가능
- 다른 통신사업자를 사용하는 사용자 간에는 독자적인 매핑을 이용하여 비슷한 이모티콘 이 표시 가능
- 위의 것들을 실현하기 위한 DB 보관용 독자 통합 코드를 보유함

실제로는 그림 2.2와 같은 흐름으로 처리를 하고 있다. d-sjis는 docomo 이모티콘의 Shift-JIS 이진 코드 표현을, a-sjis는 au용 Shift-JIS 코드를, v-utf8은 SoftBank의 UTF-8 코드(&#x?????)와 같은 숫자 참조 형식이 아닌 3바이트의 것)를 각각 나타낸다.

● 그림 2.2 MobaSiF의 이모티콘 처리



v-sjis-u는 SoftBank용 이모티콘을 MobaSiF 내부에서 다루기 위한 독자적 형식이다. Web 코드라는 16진수 표기로 1B 24 {C1} {C2} 0F(총 5바이트)의 코드 형식으로부터 {C1} {C2}를 취하여, 16진수 표기로 0B {C1} {C2}라는 코드로 변환한 것을 v-sjis-u로서 취급하고 있다.

MobaSiF에서는 이와 같이 독자적으로 문자 인코딩 및 이모티콘을 처리하고 있다. 참고로, Perl의 경우는 CPAN 모듈에 Encode::JP::Mobile이라는 것이 있다. 이러한 것들을 사용해서도 이모티콘 변환을 할 수 있다.

또한 MobaSiF 템플릿 엔진에서는 한 장의 Shift-JIS로 작성된 텍스트 템플릿 파일을 미리 컴파일하여 통신사업자별 바이너리 템플릿을 출력한다. 텍스트 템플릿 파일에 이모티콘을 입력하려면 d-sjis로 기술한다. 이것에 의해 한 장의 텍스트 템플릿 파일을 제작하는 것만으로 3개의 통신사업자에 대응한 HTML을 출력할 수 있게 된다.

단말기 인증

사용자를 식별하는 데 필요한 단말기 인증 방법에 대해 설명하겠다. 일반적인 경우라면 사용자 식별 ID와 비밀번호 조합을 사용하지만, 피쳐폰의 경우는 개별 단말기를 식별하는 방법이 준비되어 있기 때문에 ID와 비밀번호를 생략할 수 있다.

각 통신사업자의 단말 인증 방법은 표 2.3과 같다.

이들 ID는 HTTP 요청에 포함된 정보이므로 접속하고자 하는 어드레스를 휴대폰 통신사업자에 한정하지 않는 경우(또는 접속하고자 하는 곳의 IP 어드레스를 허위로 해서 연결이 가능한 경우는 쉽게 스푸핑(spoofing)이 가능하므로 주의가 필요하다.

DeNA 오픈 플랫폼을 이용하여 소셜 게임을 제공하는 경우는 Gadget 서버를 경유하여 요청이 발생한다. 사용자의 개인정보 보호를 위해 Gadget 서버로부터 요청이 발생하는 경우는 개인 정보에 관한 정보(단말기 ID 등)가 삭제된다. 즉, 오픈 플랫폼을 이용하여 소셜 게임을 개발할 때 단말 ID의 취득이나 취급을 의식하지 않아도 된다는 뜻이다.

● 표 2.3 단말 인증 방법

통신사업자	인증 ID	방법	비고
docomo	i모드 ID(URL에 guid=ON이라는 쿼리 파라미터를 부여하고 있을 때 웹 서버에 통지되는 사용자 고유의 ID)	HTTP 요청 헤더에 X-DCMGUID 라는 필드가 포함되어 그 값이 i 모드 ID가 된다(7자리 숫자, 대소 문자 구분)	사용자에 의해 비통지 설정이 가능. SSL 통신 시에는 부여되지 않는다.
au	EZ 번호(가입자ID)(EZweb 계약에 따라 할당되는 번호)	HTTP 요청 헤더에 X-Up-Subno 필드가 포함되어 이 값이 EZ 번호가 된다.	사용자에 의해 비통지 설정이 가능
SoftBank	UID(사용자 ID)(송출원의 단말기 식별자)	HTTP 요청 헤더에 jphone-uid 라는 필드가 포함되어 있으며, 이 값이 UID가 된다(16행의 영 숫자).	사용자에 의해 비통지 설정이 가능

통신사업자와 단말 기종의 판별

피쳐폰용의 웹 애플리케이션에서는 접속이 어떤 통신사업자인가를 판별하는 것이 중요하다. 통신사업자의 판별 방법으로는 두 가지를 생각할 수 있다.

- IP 어드레스에 의한 통신사업자의 확인
- User-Agent에 의한 통신사업자(그리고 단말기 기종)의 판별

IP 어드레스에 의한 통신사업자의 확인

각 통신사업자는 사용자가 사이트에 접속할 때 접속원의 IP 어드레스를 공개하고 있다. 이 정보로 IP 어드레스에 의한 통신사업자 판별이 가능하다. 또한 앞서 언급한 단말에 의한 인증의 안전성을 높일 수 있다.

User-Agent에 의한 통신사업자와 단말 기종의 판별

HTTP 요청 헤더에 포함된 User-Agent 필드를 사용하여 통신사업자와 기종을 식별하는 데 사용할 수 있다. 통신사업자별 형식을 표 2.4에 정리하였다.

● 표 2.4 통신사업자별 판별 형식

통신사업자	판별 형식	예
docomo	FOMA 「DoCoMo/2.0 기종명」으로부터 시작한다.	DoCoMo/2.0 N905i (c100;TB;W24H16)
	mova 「DoCoMo/1.0/기종명」으로부터 시작한다.	DoCoMo/1.0/N505i/c20/TB/ W20H10
au	KDDI 또는 UP.Browser로 시작한다.	KDDI-SA31 UP.Brower/6,2,0,7,3,129(GUI)
	MMP/2.0	UP.Browser/3,04-ST13 UP.Link/3,4,5,9
	SA31이나 ST13이 기종 판별에 이용되는 디바이스 ID	KDDI 공식 페이지에서 디바이스 ID로부터 기종명을 찾아볼 수 있다.
SoftBank	J-PHONE, Vodafone, Soft-Bank, MOT- 중 하나로 시작하는 문자열로 시작한다.	Vodafone/1.0/V904SH/SHJ001/ SN*****Browser/VF-NetFront/ 3,3 Prole/MIDP-2.0
	Conguration/CLDC-1.1	SoftBank/1.0/910T/TJ001/ SN*****Browser/ NetFront/3,3 Prole/MIDP-2.0 Conguration/CLDC-1.1

MobaSiF는 위와 같은 방법으로 통신사업자와 단말기의 판별을 실시하고 있다. DeNA의 개방형 플랫폼을 이용하여 소셜 게임을 제공하는 경우에도 HTTP 요청 헤더의 User-Agent 필드와 다른 필드의 정보를 취득하여 단말에 따른 처리의 배분 등을 할 수 있다.

2-3

피쳐폰 전용의 HTML

앞의 절까지는 피쳐폰 전용의 웹 애플리케이션 개발에 필요한 기술 요소에 대하여 설명하였다. 그리고 이 절에서는 브라우저 기반의 소셜 게임 개발을 실시하는 데 있어 빠뜨릴 수 없는 HTML 제작에 대한 주의점을 소개하겠다.

HTML의 주의점

한 페이지의 콘텐츠 용량은 100KB 이내로 한다

HTML 코드, 이미지를 포함하여 한 페이지의 콘텐츠 용량은 100KB 이내로 유지한다(새로운 단말은 100KB 이상도 표시 가능하지만 분류 처리가 귀찮기 때문).

이미지의 주변 표시는 style = "float:left"와 align = "center"를 모두 쓴다

가끔 au에서 이미지의 주변 표시가 제대로 되지 않은 소셜 게임을 보곤 한다. 이것은 style="float:left;"만 적용하는 경우다. 반대로 docomo에서 이미지의 주변 표시가 제대로 되지 않은 경우는 align="center"만 반영된 경우다. 3개의 통신사업자에서 확실히 주변 표시를 실행하려면 둘 다 지정해야 한다.

단말과 캐시를 고려하여 필요에 따라 URL을 동적으로 변경한다

특히 au는 단말기에서 콘텐츠를 캐시하기 때문에 이미지 파일을 교체했을 때 새로운 파일로 변환되지 않는 경우가 있다. 이 경우에는 이미지 파일의 URL을 변경하여 대응할 필요가 있다.

Mobage에서는 단말에 캐시하길 원하지 않는 페이지나 이미지 파일의 URL에 UTC 시간을 넣어 URL이 매초마다 변경되도록 하고 있다.

화면 크기는 기본 240px

피쳐폰 전용의 HTML 제작은 가로 240px로 통일하고 있다. 단, au의 휴대 전화에서는 세로로 긴 페이지인 경우에 세로 스크롤바가 표시되기 때문에 세로 230px로 표시된다.

또한 SoftBank의 새로운 단말에서는 화면 사이즈가 VGA에 대응하고 있어 일반적으로 표시하면 이미지가 작게 표시된다. 이에 대응하기 위해 Mobage는 가로 240px로 만든 파일을 가로 480px로 확대 표시하도록 하고 있다. 그러나 이 방법은 이미지가 좀 거칠게 나타나기 때문에 가능하면 가로 480px 이미지를 준비하여 가로 240px의 단말에서는 리사이징하여 표시하는 것이 좋다.

2-4

Flash Lite

피쳐폰의 브라우저는 표현력에 한계가 있다. 표현력을 높이기 위해 소셜 게임에서는 피쳐폰용으로 개발된 「Flash Lite」라는 기술을 사용한다. Flash Lite는 Adobe가 개발한 RIA 기술인 「Flash」를 모바일 전용으로 튜닝한 것으로, 피쳐폰에서만 사용할 수 있다.

그러나 Flash Lite도 단말의 모델이나 통신사업자에 따라 버전이나 사양이 달라 동작이 변하기에, 다양한 단말에 대응하기 위해서는 이 사양을 이해해야 한다.

Flash Lite를 도입하는 장점으로는 첫째, 휴대폰의 표현력을 높이는 효과가 있다. 이용 상황으로서는 괴도 로얄의 레벨업 연출, 보물의 완성 연출 등이 있다. 연출로만 보일 뿐 아니라, 서버 사이드에서 동적으로 처리하여 사용자에게 따라 표시를 변경하는 등의 처리를 하고 있는 부분도 있다.

Flash Lite의 주의점

Flash Lite는 통신사업자 또는 단말에 따라 버전과 동작이 다르다. DeNA가 소셜 게임을 개발하는 데 주의하고 있는 것을 소개하고자 한다.

버전은 Flash Lite 1.1을 사용한다

새로운 단말이면 Flash Lite 3.x와 같은 버전을 사용할 수 있지만, Flash Lite는 단말 측에서 버전업할 수 없기 때문에 보다 다양한 단말에서 동작하기 위해서는 Flash Lite 1.1로 제작하는 것이 가장 좋다.

파일 크기를 100KB 이내로 한다

HTML의 경우와 마찬가지로 통신사업자마다 파일 용량의 제한이 설정되어 있다. 대부분의 단말의 경우 docomo와 au는 최대 100KB, SoftBank는 최대 150KB로 되어 있기 때문에 3개 통신사 모두에서 실행되는 Flash를 제작하기 위해서는 100KB 이내로 개발할 필요가 있다.

무비 클립의 중첩 수는 6계층까지로 한다

복잡한 애니메이션과 인스턴스의 중복 사용에 의한 효율화를 이루기 위해 무비 클립을 중첩하는데, Flash Lite 1.1은 7계층 이상이 되면 볼 수 없다.

단말의 키 할당은 「상하」「결정」「숫자(0-9), #, *」만으로 한다

Flash를 조작하기 위해서 키 이벤트를 할당해야 하는데, 「상하」「결정」「숫자 0 ~ 9, #, *」만을 사용하도록 하고 있다. SoftBank는 「좌우」키를 사용하는 것이 가능하지만, docomo와 au는 브라우저의 「Back」키로 할당되어 있어 사용하지 않는다.

외부 통신은 버튼 이벤트로 실시한다

getURL 등으로 Flash 외부 페이지로 전환하는 경우에는 버튼 이벤트로부터 호출해야 한다. 프레임 내의 액션에서 호출할 수 없기 때문에 주의가 필요하다.

디바이스 폰트는 12px이나 24px로 설정한다

단말에서는 글꼴 크기가 제한되어 있어서 10px로 설정해도 12px 크기로 변한다. 따라서 기본 폰트의 사이즈는 12px로 해두면 레이아웃 붕괴를 방지할 수 있다.

POST 요청으로 Flash를 기동시키지 않는다

au 단말기는 POST 요청으로 Flash를 시작하려고 하면 표시되지 않는 현상이 있다. GET 요청을 사용하는 편이 무난하다.

가급적 사용 메모리의 크기를 1MB 이내로 억제한다

파일 사이즈에는 주의하기 쉽지만, 사용 메모리 사이즈에 주의하지 않으면 단말에 따라서는 표시되지 않는 경우도 있다. 사용 메모리를 억제하기 위한 방안으로 다음과 같은 것들이 있다.

- 색상 수가 적으면 jpeg보다도 png를 사용한다
- 복잡한 벡터는 이미지로 한다
- 변수 명, 프레임 명을 짧게 한다
- 간단한 도형은 심볼화하여 재활용한다
- 이미지의 알파 채널은 사용하지 않고 Flash 측에서 마스크한다

동적 Flash를 생성하는 방법

정적 Flash 콘텐츠는 이미지와 같이 작성한 SWF 파일을 호출할 뿐이지만, 소셜 게임에서 Flash를 사용하는 경우는 사용자의 정보(캐릭터, 진행 상황 등)에 따라 동적으로 콘텐츠를 생성하고 싶어진다. 사전에 모든 패턴을 정적으로 만들어 놓을 수 있을 만큼의 패턴 수인 경우라면 문제가 없지만, 패턴 수가 많으면 동적으로 생성해야 한다.

DeNA의 소셜 게임은 Flash의 동적 생성을 위해 Ming이라는 라이브러리를 사용하여 Perl로부터 SWF를 동적으로 생성하도록 하고 있다. SWF를 동적으로 생성하고자 한다면 Ming의 다른 여러 가지 방법이 있다.

SWF 바이너리 구조에 직접 손대기

Flash 내부 사양에 대해 이해할 수 있으면 구현 가능하다. 그러나 SWF 바이너리 구조는 복잡해서 이해하는 데 시간이 걸린다. DeNA에서도 이 방식으로 동적으로 Flash를 생성하는 방식을 사용했지만, 도입의 용이성 문제로 소셜 게임 개발에서는 Ming을 사용한 방법을 사용하였다.

swfmill을 사용하여 SWF 새로 짜넣기

swfmill이라는 툴이 있어 SWF와 XML을 상호 변환하는 것이 가능하다. 작성해 낸 SWF 파일에 대해 변경할 수 있기 때문에 용이하게 도입할 수 있다. 그러나 검증 결과, 성능 면에서 기대했던 결과를 얻을 수 없었기 때문에 사용하고 있지는 않다.

flasm을 사용하여 actionscript 대체하기

flasm이라는 툴을 사용하면 actionscript의 추출 및 치환이 가능하게 된다.

동작을 변경하는 용도로 충분하지만, 디자인 부분이 변경되지 않기 때문에 활용도는 제한되어 있다. 요구사항을 만족시키지 못해 이용하지 않는다.

Ming을 사용하여 서버 측 Flash 생성하기

Ming은 Flash를 생성하기 위해 개발된 라이브러리로, 프로그램 작성으로 Flash를 생성할 수 있다. Python, Ruby, Perl, C와 같은 언어로 포팅되어 있다. DeNA에서는 이 Ming을 사용하여 Flash를 생성하는 방법을 채용했다.

Ming은 프로그래밍에 의해 처음부터 Flash를 생성할 수 있기 때문에 프로그램 측에서 Flash를 작성하는 용도로 매우 유용하지만, 원래 만들어져 있는 SWF 파일을 변경하거나 대체하거나 하는 용도로는 불편하다. DeNA에서는 대체 기능을 독자적으로 준비하여 새로운 Flash를 만들 수도, 기존의 SWF 파일을 대체하여 사용할 수도 있게 하고 있다.

Ming을 사용한 Flash Lite를 생성하는 방법에 대해 설명하겠다.

Ming의 사용법

Ming을 사용하기 위해서는 서버에 Ming를 설치할 필요가 있다. 이 책에서는 설치 순서에 대해서는 생략하고 있으므로 인터넷 등에서 확인하길 바란다. Ming을 사용하여 Flash Lite의 SWF를 생성하는 프로그램에 대해서만 설명한다. 리스트 2.1 프로그램은 Flash를 생성하는 데 필요한 기본 프로그램이다. 단독으로 실행하면 시커먼 Flash가 표시된다. 프로그램을 보는 것만으로는 무엇을 하고 있는지 잘 모를 거라 생각되므로 순서대로 설명해 나가겠다.

리스트 2.1 Ming을 사용한 SWF 생성

```
use SWF qw(:ALL);  
  
SWF::setVersion(4); ← ❶  
SWF::setScale(1.0);
```

```

my $movie = new SWF::Movie();
$movie->setBackground(0,0,0); ← ❹
$movie->setRate(12); ← ❷
$movie->setDimension(4000, 4000); ← ❸

# 처리를 여기에 작성

# 출력 ← ❶
my $body = $movie->data();
print "Connection: close\n";
print "Content-type: application/x-shockwave-flash\n";
print "Access-Control-Allow-Origin: *\n";
print "\n";
print $body;

```

버전의 지정

아래에 발췌한 리스트 2.1❶은 생성할 Flash 버전을 지정할 수 있다. 4로 하고 있는 이유는 Flash Lite 1.1이 Flash Player 4에 해당하기 때문이다.

```
SWF::setVersion(4);
```

프레임 속도 지정

리스트 2.1❷은 생성된 Flash 콘텐츠의 재생 프레임 속도를 지정하는 항목이다.

```
$movie->setRate(12);
```

Flash 콘텐츠가 재생되는 속도는 단말의 성능에 따라 다르다. 단말에 따라 최적화를 실시한다면 단말에 따라 값을 변경시키는 것이 좋다. 운용 실적상 12fps로 설정하고 있다면 대부분의 경우에는 문제가 없다. 그러나 Mobage의 일부 소셜 게임이나 액션성이나 타이밍이 요구되는 Flash 콘텐츠에서는 단말마다 최적의 프레임 속도 값을 설정하고 있다.

출력 크기 설정

생성하는 Flash 콘텐츠의 크기를 지정한다. 아래에 발췌한 리스트 2.1③의 예에서는 세로 200px, 가로 200px로 출력된다.

```
$movie->setDimension(4000, 4000);
```

4000이라는 숫자를 지정하고 있는 것은 SWF 포맷에서 twips라는 단위로 지정해야 하기 때문이다. twips는 1/20 논리 픽셀 크기를 나타내므로 px의 20배 값을 지정한다.

SWF::setScale에 20을 지정하면 Ming 내부에서 행해지는 비율이 20배가 된다. 이 경우는 setDimension 값을 px 값으로 지정하길 바란다.

배경색의 지정

이 항목은 배경색을 지정할 수 있다(리스트 2.1④ 발췌). 설정하는 값은 RGB의 0 ~ 255의 값이다.

```
$movie->setBackground(0,0,0);
```

출력 방법

다음과 같이 응답을 생성하여 출력할 필요가 있다(리스트 2.1⑤ 발췌).

```
my $body = $movie->data();  
print "Connection: close\n";  
print "Content-type: application/x-shockwave-flash\n";  
print "\n";  
print $body;
```

이상이 기본적인 Ming의 사용법이다.

그럼 이어서 소셜 게임 개발에 필요한 처리에 대해 설명해 나가겠다.

이미지를 읽어들이어 표시하기

Flash 내에 이미지를 읽어들이고 싶으면 다음과 같이 한다.

```
my $movie = new SWF::Movie();

my $scale = 20;
my $bitmap = new SWF::Bitmap('image.gif');
my $width = $bitmap->getWidth() * $scale;
my $shape = new SWF::Shape();
my $fill = $shape->addFill($bitmap, SWFFILL_CLIPPED_BITMAP);
$fill->scaleTo($scale);
$shape->setLeftFill($fill);
$shape->drawLineTo($width, 0);
$shape->drawLineTo($width, $height);
$shape->drawLineTo(0, $height);
$shape->drawLineTo(0, 0);

$movie->add($shape);
```

어렵게 보일지도 모르겠으나, 위에서 하고 있는 작업은 이미지 파일을 비트맵 객체의 형식으로 읽어들이어 Shape라는 그리기 객체에 채우기 설정을 하고 있을 뿐이다.

이미지의 일부만 표시하려면 Shape의 그리기 범위와 이미지의 그리기 위치를 변경한다.

또한 반복 패턴으로 설정하려면 SWFFILL_CLIPPED_BITMAP을 SWFFILL_TILED_BITMAP으로 변경하면 된다.

기존 SWF 파일을 읽어들이어 표시하기

Ming에서 동적으로 Flash를 생성한다고 해도 모든 것을 처음부터 조립하는 것은 힘들다.

이미지와 마찬가지로 이미 생성된 SWF 파일을 읽어들이려는 경우가 많을 거라 생각한다. Ming에서 SWF 파일을 읽어들이려면 다음과 같이 처리한다.

```
my $mc = new SWF::PrebuiltClip('sample.swf');
$movie->add($mc);
```

이것만으로 Flash 상에서 sample.swf의 내용이 표시되게 된다. 단, 추가되는 부분이 add된 축의 원점(0,0)이 되기 때문에 화면에서 보면 왼쪽 상단에 표시된다. 표시 위치를 이동할 경우에는 add를 실행할 때 반환값으로 돌아오는 객체의 핸들을 사용하여 다음과 같이 처리한다.

```
my $mc = new SWF::PrebuiltClip('sample.swf');
my $item = $movie->add($mc);
$item->moveTo(2000, 2000);
```

moveTo 함수를 사용함으로써 add한 객체를 조작할 수 있다. 위의 소스는 x와 y 축에 대해 각각 2000twips(=100px)를 이동시키게 된다. 이외에도 객체를 조작하는 메소드는 존재한다. 중요한 메소드를 표 2.5에 나타내었다. 표 2.5에 있는 상대 좌표 계란 현재 위치이며, 글로벌 좌표계는 원점(0,0)을 가르킨다.

● 표 2.5 객체를 조작하는 메소드

메소드	의미
move	상대 좌표계에서 객체를 이동하기
moveTo	글로벌 좌표계에서 객체를 이동하기
scale	상대 좌표계에서 객체를 확대 또는 축소하기
scaleTo	글로벌 좌표계에서 객체를 확대 또는 축소하기
rotate	상대 좌표계에서 회전시키기
rotateTo	글로벌 좌표계에서 회전시키기

액션스크립트 제어하기

Ming에서 액션스크립트를 추가하려면 다음과 같이 한다.


```
my $movie = new SWF::Movie();
my $as = SWF::Action->new(<<"AS");
/:url = 'hohogoge';
AS
$movie->add($as);
```

위는 `/:url`이라는 전역 변수에 `hohogoge`라는 값을 설정하는 프로그램이다.

Flash를 동적으로 생성하는 장점은 사용자가 파라미터를 변경하거나 읽어들이 이미지를 바꾸거나 할 수 있다는 점에 있다.

간단한 부분이라면 화면 전환할 곳을 변경하거나 애니메이션을 재생할 횟수를 변경하려는 경우가 많을 거라 생각한다.

그럼 위의 예제를 `Ming`을 사용하여 구현하는 방법에 대해 설명하겠다.

우선 SWF 파일을 준비한다. 특히 SWF 파일(`sample.swf`) 안에 `/:num`의 값만큼 버튼을 누르면 `/:url`로 화면 전환하는 작업을 할 액션스크립트를 작성한다. 주의할 점으로, 기본값은 대입하지 않지 않는다. 실제 프로그램은 다음과 같다.

```
my $movie = new SWF::Movie();
my $mc = SWF::PrebuiltClip->new('sample.swf');
$movie->add($mc);

# 액션스크립트
my $as = SWF::Action->new(<<"AS");
/:url = 'http://hoge.jp/fuga';
/:num = 20;
AS
$movie->add($as);
```

`PrebuiltClip`을 사용하여 `sample.swf`를 읽어들이 `Movie`에 추가하고, 액션스크립트를 프로그램에서 추가한다.

단지 이것만으로도 20번 버튼을 누르면 `http://hoge.jp/fuga`로 이동하는 Flash가 된다. 어떻게 `sample.swf` 안에 작성된 액션스크립트의 제어가 가능한 것일까? 그것은 변수의 이름에 붙어 있는 `/:`이 루트 상의 전역 변수로서 어디서나 참조 및 업

데이트가 가능하기 때문이다.

/: 변수를 잘 사용함으로써 하나의 SWF 파일의 처리를 제어할 수 있을 뿐만 아니라, 다른 SWF 파일들과의 처리나 Ming에서 생성한 무비 클립 처리와도 연계가 가능하다.

간단히 Ming을 사용한 Flash 콘텐츠의 동적 생성을 소개하였다. 다른 유용한 기능과 사용법도 있으므로 자세한 내용은 Ming의 사이트 및 인터넷에서 검색해보길 바란다.

2-5

소셜 게임의 UI 설계

브라우저 기반의 소셜 게임은 게임성뿐만 아니라 UI에 따라 결과가 달라질 수 있다. 특히 퍼스트 링크와 퍼스트 뷰는 DeNA 소셜 게임 개발에 중요한 포인트로 생각하고 있다.

퍼스트 링크는 표시 화면에 처음으로 앵커가 있는 링크를 의미하며, 피쳐폰의 브라우저 인터페이스에서 가장 누르기 쉬운 링크가 된다. 표시되는 화면의 첫 링크에는 사용자가 우선적으로 해주었으면 하는 것을 배치하도록 한다. 퍼스트 링크가 전혀 누르지 않을 링크로 되어 있으면, 그 화면이 표시될 때마다 사용자는 아래 버튼을 눌러 원하는 링크로 이동해야만 한다. 작은 일이지만 웹 페이지에서의 작업이 하나 증가하는 것만으로도 사용자는 귀찮아 한다. 퍼스트 뷰뿐만 아니라 링크 순서도 사용자가 자주 누르는 것들 위주로 표시할 수 있도록 궁리해야 한다.

다음으로 중요한 것은 퍼스트 뷰라는 것이다. 퍼스트 뷰는 특정 페이지를 볼 때 가장 먼저 휴대 화면에 표시되는 범위를 말한다. 이 첫 번째 뷰에 사용자가 원하는 정보와 링크를 배치해둬으로써 조작성이 향상된다. 이것이 잘 설계되어 있지 않으면 사용자는 원하는 정보를 보기 위해 매번 화면을 스크롤해야 하므로 한 단계의 번거로운 추가 동작이 발생하게 된다. 단, 단말기에 따라 화면 크기가 다르기 때문에 퍼스트 뷰를 고려할 때는 표준적인 단말의 화면 크기로 검토하고 있다.

2-6

보안 대책

브라우저 기반의 소셜 게임에서 보안은 매우 중요하다. 웹 애플리케이션에서 주의할 XSS(크로스 사이트 스크립팅), CSRF(크로스 사이트 요청 위조), SQL 인젝션 대책은 물론, 파라미터 변조에 의한 액세스, 브라우저 백에 의한 불규칙적인 화면 전환 조작, 화면 메모나 북마크에 의한 직접 화면 전환 등도 보안 대책으로 의식하고 있어야 한다.

사내 소셜 게임에서는 파라미터의 체크에 유효성 검사기를 사용하고 있다. Perl에서는 FormValidator::Simple처럼 검증 모듈이 있으므로 적극적으로 이용하길 바란다.

Column

통신사업자의 리다이렉트 횟수 제한

피쳐폰의 사양으로 리디렉션 횟수에 주의할 필요가 있다. 통신사업자와 단말기에 따라 제한 횟수가 다르지만, 대다수의 단말에 대응하기 위해 다음의 횟수를 기준으로 한다.

● 표 A 통신사업자당 리디렉션 횟수

통신사업자	횟수	비고
Softbank	최대 3회(C형 최대 2회)	초과했을 경우: 무효한 데이터를 수신하였습니다.
Docomo	최대 4회	초과했을 경우: 이 페이지는 표시할 수 없습니다.
au	최대 6회	초과했을 경우: 이 페이지로의 액세스는 거부되었습니다.

※ 최신 단말에서는 10회 이상 가능한 것도 있다.

이 제한은 웹 애플리케이션 내에서 에러 핸들링을 수행하여 올바른 페이지로 리디렉션 처리 및 광고 등의 제휴로 리디렉션하는 경우에 주의할 필요가 있다. 또한 PC 브라우저나 에뮬레이터에서는 재현하지 않기 때문에 실제 단말로 확인할 필요가 있으므로 주의하길 바란다.