

平成 25 年 2 月 1 日

卒業論文

**Mashup による e-learning コンテンツ検索システムの開発**

佐賀大学 理工学部知能情報システム学科

08233014  
甲斐 遼馬

指導教官： 新井 康平 教授  
学科主任： 渡邊 義明 教授

February 1, 2013

Graduation Thesis

# Development of e-learning content searcher by Mashup

Department of Information Science, Faculty of Science and Engineering, //  
Saga University

08233014  
Ryoma KAI

Supervisor : Professor Kohei ARAI

Chief of Department : Professor Yoshiaki WATANABE

## 卒業論文概要 2012 年度（平成 24 年度）

### Mashup による e-learning コンテンツ検索システムの開発

Mashup はポータルと同様にコンテンツ検索を可能にするツールであるが、Web2.0 を用いてコンテンツを分類する、異なるサイトの API を利用する、クライアント・サーバー双方で検索ができる、検索は任意の構造化ハイブリッドコンテンツとして個々のコンテンツを混合して行える、XML 変換の REST,RSS,Atom 等が利用等が異なる。この特徴を利用し、既提案モデル納豆ビューによるコンテンツを三次元空間にグラフとして展開し操作可能とする Android 上での検索システムを構築した。

キーワード： Mashup, Android, e-learning

## **Graduation Thesis Overview Year 2013**

### **Development of e-learning content searcher by Mashup**

Mashup enables us to search contents like a portal site. However, there is difference to classify contents with Web2.0, use webAPI in external sites, search in the client-side and the server-side, merge individual contents as a any hybrid content, and use REST, RSS, and Atom with converted XML. This time I developed the search system that make it possible to control the graph in the 3D space by existing proposed model(NATTO View), with these peculiarity on Android devices.

**Keywords:** Mashup, Android, e-learning

# 目次

<b>第 1 章 序論</b>	<b>1</b>
1.1 背景 . . . . .	1
1.2 本文書の構成 . . . . .	1
<b>第 2 章 Mashup と WebAPI</b>	<b>2</b>
2.1 Mashup とは . . . . .	2
2.2 WebAPI とは . . . . .	2
2.2.1 Yahoo!検索 Web API-ウェブ検索 API . . . . .	2
2.2.2 Yahoo!検索 Web API-画像検索 API . . . . .	2
2.2.3 Youtube Data API . . . . .	3
2.2.4 Product Advertising API . . . . .	3
<b>第 3 章 検索エンジンの精度向上</b>	<b>4</b>
3.1 先行研究 . . . . .	4
3.1.1 e-learning コンテンツにおけるドキュメントサーチの最適化 [5] . . . . .	4
3.2 サブキーワードの選出 . . . . .	4
3.2.1 選出過程 . . . . .	4
3.2.2 選出結果 . . . . .	4
<b>第 4 章 WWW 視覚化</b>	<b>5</b>
4.1 開発事例 . . . . .	5
4.1.1 納豆ビュー . . . . .	5
4.1.2 Flowser . . . . .	5
4.2 検索エンジンにおける WWW 視覚化 . . . . .	7
4.2.1 Helix モデル . . . . .	7
4.2.2 Star モデル . . . . .	8
4.2.3 Star-Helix モデル . . . . .	8
4.2.4 Star-Slide モデル . . . . .	8
4.3 採用手法 . . . . .	9
<b>第 5 章 開発手法</b>	<b>10</b>
5.1 概観 . . . . .	10
5.2 Away3D . . . . .	10
5.2.1 Away3D とは . . . . .	10
5.2.2 Android 上への移植 . . . . .	10
<b>第 6 章 Android アプリ 「LEDOXEA」</b>	<b>11</b>
6.1 使用方法 . . . . .	11
6.2 特徴 . . . . .	13
6.2.1 移植性 . . . . .	13
6.2.2 スペック性能への非依存性 . . . . .	13
6.2.3 フリック操作、画像ノードによる直感的操作性 . . . . .	13
6.2.4 検索エンジンの同時検索、WWW 視覚化 . . . . .	13

<b>第 7 章 アンケートによる評価と考察</b>	<b>14</b>
7.1 評価結果 . . . . .	14
7.2 考察 . . . . .	15
7.2.1 e-learning コンテンツ検索エンジンとしての性能 . . . . .	15
7.2.2 Android 端末における WWW 視覚化とユーザビリティ . . . . .	15
<b>第 8 章 結論</b>	<b>16</b>
8.1 成果 . . . . .	16
8.2 今後の課題と方針 . . . . .	16
<b>謝辞</b>	<b>17</b>
<b>参考文献</b>	<b>18</b>
<b>付 錄 A プログラム</b>	<b>19</b>
A.1 Main.as(ActionScript3.0) . . . . .	19
A.2 ledoxea-app.xml(XML) . . . . .	34

## 図目次

4.1	納豆ビュー	5
4.2	Flowser	6
4.3	Helix モデル	7
4.4	Star モデル	8
4.5	Star-Helix モデル	9
4.6	Star-Slide モデル	9
6.1	起動画面	11
6.2	検索キーワードを入力	11
6.3	検索ボタンを押下	12
6.4	コンテンツ表示	12
6.5	上にスワイプで上下移動	12
6.6	横にスワイプで左右回転	12

# 第1章 序論

本章では、本研究の背景、それを踏まえた上で研究の目標・目的、そして文書の構成について述べる。

## 1.1 背景

2007年、iOS、Android OS の両オペレーティング・システムを搭載したタブレット端末やスマートフォン端末が発表された。これら端末はスペック的にそれほど高くないものの、タッチパネルの搭載による直感的な操作、携帯性の高さ、Wi-fi 接続によるインターネット接続が可能といった多数のメリットを兼ね備えており、欧米を中心に今日まで爆発的に普及してきている。<sup>1</sup>

一方、e-learning とは、パーソナルコンピュータなどの情報機器を用いて行う学習のことである。1990年代後半からのPCの普及と共に様々な分野で用いられるようになり、現在では e-learning のコンテンツ共有を目的とした規格 [2] や大学設置基準に基づく文部科学省告示の中に e-learning に関する項目が記述される [4] など、制度や規格も整備されたものとなっている。

だが e-learning コンテンツを提供するサイトは、その多くがタブレット端末、スマートフォン端末が発表されるより前に製作されたものである。現状、スマートフォンやタブレットから e-learning コンテンツに対してダイレクトにアクセスするためには、まずはうまくコンテンツだけがヒットするような副次的な検索キーワードを考えて、PC用のサイトから小さなボタンをタップし、コンテンツをダウンロードし、更にテキストや動画で別々の検索エンジンを使わなければならない、といったようにかなりの労力を要する。<sup>2</sup>。

当研究では、こういった問題点を改良するため、スマートフォン端末・タブレット端末上で e-learning コンテンツを簡単に検索し、自在かつ直感的に閲覧、ダウンロードできるアプリの開発を行う。

## 1.2 本文書の構成

この第1章では、本論文を書くに至った背景とその構成を説明している。

第2章では、検索エンジンの構成に使用した WebAPI と、それらを統合した手法について説明する。第3章でキーワードを用いた検索結果の精度向上手法について説明する。第4章では、WWW 視覚化という目線での先行研究や開発事例、そして解決方法についての案を提起する。第5章では、アプリ開発のための言語や手法についての詳細を概説する。第6章では、開発したアプリの利用方法と、その特徴について説明する。第7章では、そのアプリについての評価を行い、その結果から考察を述べる。第8章では、本研究のまとめを行い、今後の課題を列挙する。

<sup>1</sup> 株式会社シード・プランニングの行った 2012 年 7 月の市場調査 [1] によると、日本でのスマートフォン普及率は 40% 前後と先進国の中ではやや低調である。元々高品質な携帯電話が普及しており、プラットフォームとしても盤石であったことが要因であると考えられている。

<sup>2</sup> iOS については、e-learning 用にユーザーインターフェースが最適化された iTunesU[3] が存在するが、これは iTunesStore 内にあるコンテンツのみを対象としており、WWW 上に存在するコンテンツをすべて検索対象とすることができない。

# 第2章 Mashup と WebAPI

本章では、表題となっている Mashup と呼ばれる開発手法に加え、WebAPI と呼ばれるタイプの API について解説する。

## 2.1 Mashup とは

Mashup とは、2つ以上の WebAPI を組み合わせて1つの Web サービスやアプリケーションを構成する手法のことである。元来、利用価値の高い Web サービスを作るためには、独自に検索エンジンや結果応答用のサーバを構築する必要があり、目的の Web サービスを作るために多大な労力を払う必要があった。しかし、Mashup では、既存のデータベースを有する Web サービスから得られた応答結果を自由に組み合わせることにより、短期間で価値の高い Web サービスを製作することができる。

## 2.2 WebAPI とは

WebAPI とは、インターネットを介して利用することのできるアプリケーション・プログラミング・インターフェイス (API) のことである。殆どの WebAPI が一般的な URL の形式を取っており、HTTP による POST メソッドを用いて、パラメータを付加した URL を使用してアクセスしてデータを取得する。返ってくるデータは XML、JSON のどちらかが一般的である。今回用いる WebAPI は、以下の4つである。

### 2.2.1 Yahoo!検索 Web API-ウェブ検索 API<sup>1</sup>

開発	ヤフー株式会社
URL	<a href="http://search.yahooapis.jp/PremiumWebSearchService/V1/webSearch">http://search.yahooapis.jp/PremiumWebSearchService/V1/webSearch</a>
機能	Web 上に公開されているページを検索する

### 2.2.2 Yahoo!検索 Web API-画像検索 API<sup>2</sup>

開発	ヤフー株式会社
URL	<a href="http://search.yahooapis.jp/PremiumImageSearchService/V1/imageSearch">http://search.yahooapis.jp/PremiumImageSearchService/V1/imageSearch</a>
機能	Web 上に公開されている画像を検索する

<sup>1</sup>ウェブ検索 API は、2013年3月頃を目処に API のリクエスト URL が変更される予定であり、これはそれまで公開されていたアップグレード版ウェブ検索 API を使用している。

<sup>2</sup>画像検索 API は、2013年3月頃を目処に API のリクエスト URL が変更される予定であり、これはそれまで公開されていたアップグレード版画像検索 API を使用している。

### **2.2.3 Youtube Data API**

開発	Google Inc.
URL	<a href="http://gdata.youtube.com/feeds/api/videos">http://gdata.youtube.com/feeds/api/videos</a>
機能	Youtube の機能（動画の検索、アップロード、再生リストの作成など）を利用する

### **2.2.4 Product Advertising API**

開発	Amazon.com, Inc.
URL	<a href="http://ecs.amazonaws.jp/onca/xml">http://ecs.amazonaws.jp/onca/xml</a>
機能	Amazon の商品情報や関連コンテンツを検索する

# 第3章 検索エンジンの精度向上

この章では、キーワードを用いた検索結果の精度向上手法について説明する。

## 3.1 先行研究

当研究室にて行われた研究として、以下の研究がある。

### 3.1.1 e-learning コンテンツにおけるドキュメントサーチの最適化 [5]

Yahoo!Search BOSS API を用いて、有用な Web ページへのヒット率を向上させる実験が行われた。その結果、検索を行う主なキーワードとは別に、HTML ページ中に埋め込まれた `<meta>` タグの `keyword` 属性のパラメータとして特に多いものをサブキーワードとして検索を行う方法にて、有用な Web サイトへのヒット率が向上することが確認された。

## 3.2 サブキーワードの選出

上記研究結果から、e-learning コンテンツへのアクセス精度を高めるため、サブキーワードを選出する方法を考える。

### 3.2.1 選出過程

サブキーワードを選出する方法に上記の先行研究結果を用いることを試みたのだが、HTML をキャッシュせず、検索を行うたびに毎回数十ページへアクセスを行なっていたため、タブレット端末やスマートフォン端末など、通信が不安定になる可能性が高い端末でこれを用いることは難しいと判断した。よって今回は、予めヒット率が向上すると考えられるキーワードを適当に予測、検証し、1～2 個程度のサブキーワードを選出した。

### 3.2.2 選出結果

- 基礎
- 講座

以上の 2 個をサブキーワードとする。サブキーワードは OR に設定して検索を行い、どちらか片方が、メインキーワードと共にヒットした場合の結果のみを引き出すよう設定する。この手法により、e-learning コンテンツのみに絞った検索結果を webAPI より得られるよう構成した。

# 第4章 WWW 視覚化

この章では、WWW 視覚化という目線での先行研究や開発事例、そして解決方法についての案を提起する。

## 4.1 開発事例

この分野における開発事例や先行研究は多数存在しているが、中でも 3 次元 CG による視覚化を実現している UNIX ソフトウェア「納豆ビュー[6]」と、mashup による検索エンジンの WWW 視覚化を実現している Web サービス「Flowser[7]」について解説する。

### 4.1.1 納豆ビュー

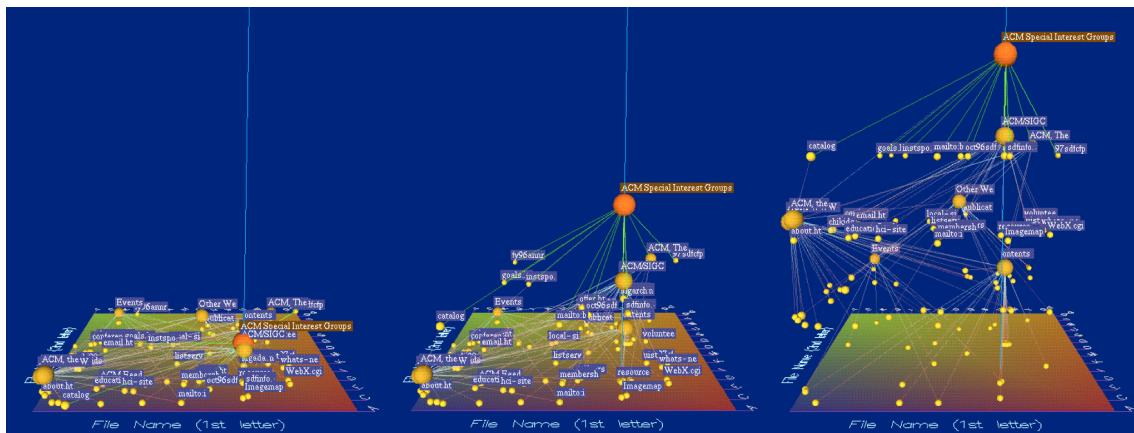


図 4.1: 納豆ビュー

UNIX の X-Window System、Mesa、GLUT を用いており、3 次元 CG グラフ上に展開したノードをリンクに見立て、リンク・被リンクにある関係がエッジで表示される。xy 平面には一意的な座標が与えられ、z 軸方向にはノードを摘んで持ち上げる、つまりユーザによる操作が可能となっている。これにより、複雑なネットワークをユーザーの意志によってわかりやすく可視化できるようになっている。

### 4.1.2 Flowser

Product Advertising API を用いた Web サービスであり、Mashup の作例でもある。検索ボックスに入力したキーワードから Amazon の複数ジャンルの商品を一気に検索・閲覧することができ、通常のサイトを通じた検索では得られなかつた情報を届けることを可能にした。

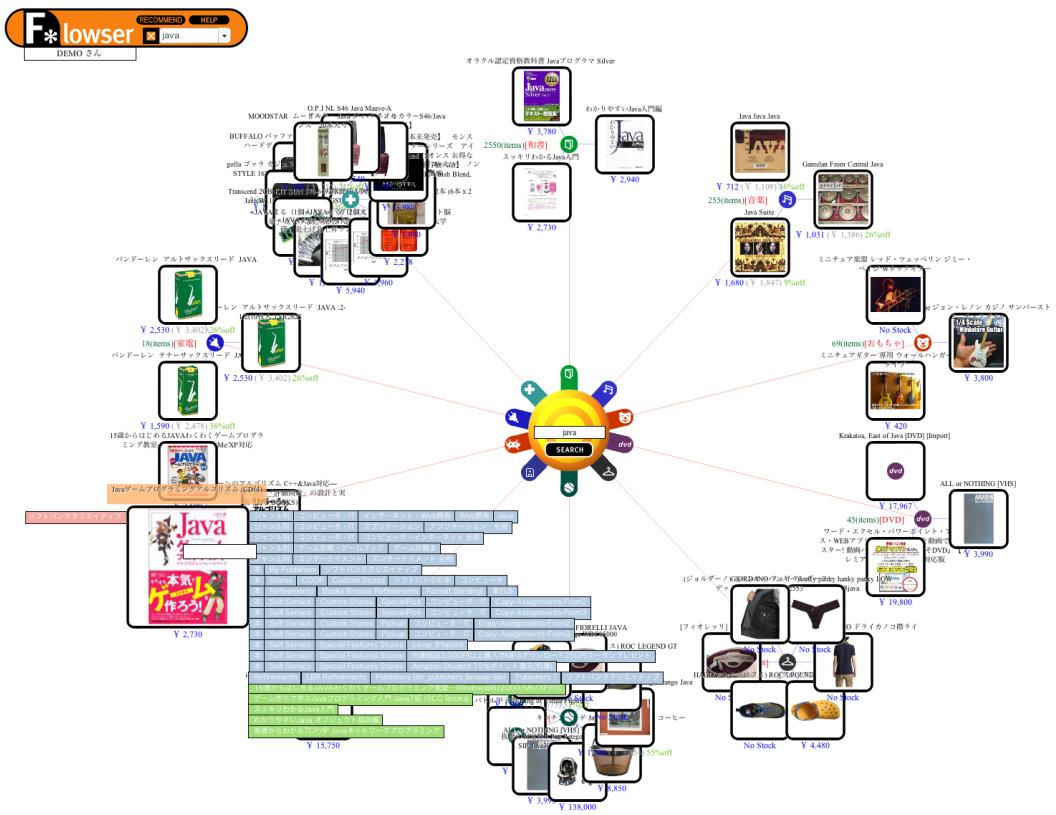


図 4.2: Flowsfer

## 4.2 検索エンジンにおける WWW 視覚化

以上2つの開発事例を見たところで、改めて、今回の研究で解決したい問題を整理する。

- タブレットやスマートフォン上から検索結果への直感的な操作  
直感的に結果を閲覧するためには、納豆ビューのようにユーザー自身によって結果をわかりやすく可視化できるようにする工夫が必要になる。また、タブレットやスマートフォンでの閲覧を前提とするため、画面サイズやボタンサイズなど、操作性に対する配慮も必要となる。
- 複数のコンテンツ(検索エンジン)を同時に検索  
e-learning コンテンツは、Web ページだけでなく、動画、画像、PDF などの文書ファイルのように、多数の形式に分かれていることが考えられる。であれば、Flowser のように、複数結果をそれぞれ分離して見やすく表示する必要がある。
- e-learning コンテンツへの導線  
コンテンツを検索して終了、ではなく、検索したコンテンツへはダイレクトにアクセス可能にする。ページであればブラウザでの表示を行い、動画であればタップと同時に動画サイトやアプリへ遷移し、再生を開始する必要がある。

以上の条件から、4つの表示モデルを考えた。

### 4.2.1 Helix モデル

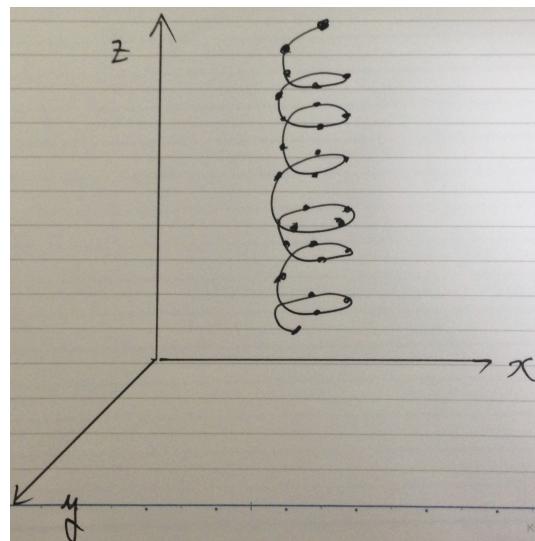


図 4.3: Helix モデル

3次元螺旋の円周上に検索結果のコンテンツを配置し、螺旋階段を降りるように検索結果の下位コンテンツへと閲覧していくモデルである。この方法の優れた点として、螺旋階段の中心線からコンテンツを見た時、コンテンツ同士の下位と上位が判断しやすく、ユーザがどのような方向にスワイプしても検索結果を辿れる、つまり上下左右方向への持ち替えが容易であるといった利

点がある。しかし、複数コンテンツへの対応を考えた際、螺旋一つでは画面上に対しての情報量に無駄が多く、複数コンテンツへ対応しきれない可能性が高いと考えられる。

#### 4.2.2 Star モデル

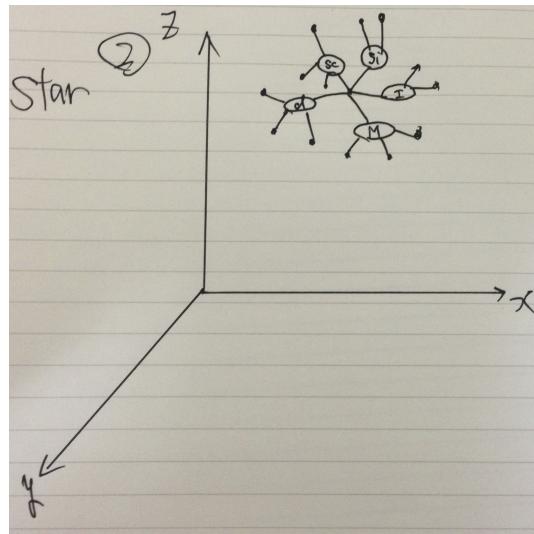


図 4.4: Star モデル

複数コンテンツへの配置を最優先に考え、Flowser のような円形状のコンテンツ展開を考えたモデルである。コンテンツ毎にそれぞれのノードに分類され、そこから 3 次元上にランダムにノードが生えており、このノードまでのエッジの長さが検索結果の上位と下位を表している。しかし、3 次元上に無作為にノードが存在しているため、コンテンツの一覧性には著しく欠けており、視点を定め辛いことが考えられる。

#### 4.2.3 Star-Helix モデル

Helix モデルと Star モデル、両者のメリットをうまく併せて設計したモデルである。螺旋はコンテンツジャンルの数だけ存在し、円形状に展開した始点から同方向に対して一様に伸びている。しかし、複数の螺旋を同時に見れる視点の位置が一意に決まらないため、ユーザーが混乱する可能性が高い。

#### 4.2.4 Star-Slide モデル

Star model のコンテンツ分離性を活かし、そこから垂直、同方向にコンテンツを配置したモデルである。このモデルを外周から見ると、ちょうど画面上にすべてのコンテンツが入るようになっており、非常に視認性に優れている。加えて、スワイプによって移動する方向も一意であることから、ユーザーが混乱しにくい。また、最前方に配置するコンテンツは丸型ハンガーラックのように回転するようになっており、別のコンテンツをメインに見たい、という時には横方向へのスワイプで自由に切り替えられるようになっている。

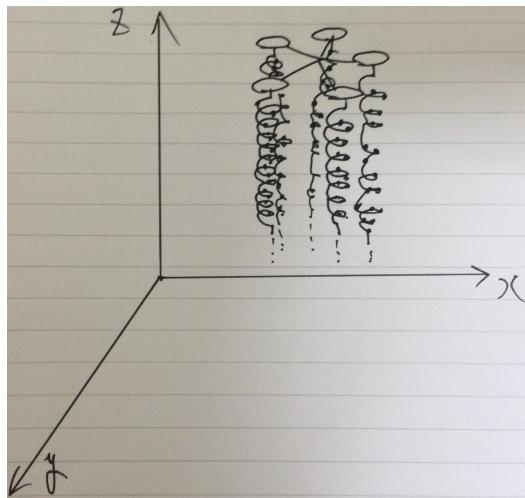


図 4.5: Star-Helix モデル

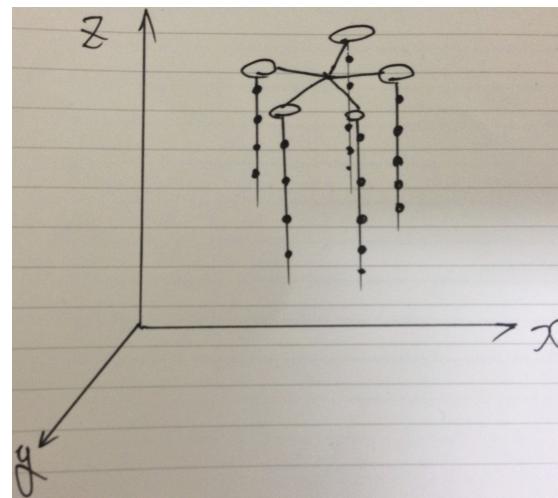


図 4.6: Star-Slide モデル

### 4.3 採用手法

以上より、最も多くの問題を改善できた方法として、Star-Slide モデルの採用を決定した。

# 第5章 開発手法

本章では、本研究の背景、それを踏まえた上で研究の目標・目的、そして文書の構成について述べる。

## 5.1 概観

タブレットやスマートフォンアプリで使える言語としては、Java や Objective-C が挙げられる。しかし、これらの言語で作る3次元CGはOpenGLを用いるものが多いため、全体的に難易度が高く、一度コードを書いてしまうと移植性も低い。その点を考慮した結果、今回はADOBE AIR(ActionScript)上で、Away3Dをライブラリとして採用する方針を取った。

## 5.2 Away3D

Away3Dについて、基本的な構造を解説する。

### 5.2.1 Away3D とは

Away3D[8]とは、Stage3Dと呼ばれるADOBE AIRからGPUを利用するAPI上で機能する、3D描画ライブラリである。基本的な3Dの描画機能に加え、プリミティブ数やエフェクトの種類が豊富に揃っており、3次元CGの一通りの機能を利用することができる。Apache 2.0 ライセンスにて無償で配布されている。

### 5.2.2 Android 上への移植

また、Away3Dの特徴として、ADOBE AIRがマルチプラットフォームに対応しているため、端末を選ばずに移植が容易であることが挙げられる。今回は開発の都合上、Androidスマートフォンでの設計を行なっているが、コードを一部手直しするだけで、iOSやPC(ブラウザ)上でも利用可能になるよう設計を行った。

# 第6章 Android アプリ 「LEDOXEA」

本章では、開発した Android アプリ 「LEDOXEA」 の利用方法と、その特徴について説明する。

## 6.1 使用方法

起動後、図 6.1 のような画面が表示される。上部にある検索ボックスをタップすると図 6.2 のようにソフトウェアキーボードが表示されるので、ここに検索したいキーワードを入力する。日本語入力も可能である。入力が完了したら、図 6.3 のように右側の検索ボタンを押下する。すると検索が行われ、図 6.4 のように、合計 5 本のラインと星形のコンテンツ一覧が画面上に出現する。これらを上下にスワイプすると、図 6.5 のように、検索結果を上下に移動できる。これにより、ひとつひとつ検索結果を示すアイコンを参照していくことが可能である。また、別の種類のコンテンツを参照したい時は左右にスワイプする。すると、図 6.6 のように、回転ハンガーのようにコンテンツが隣のコンテンツまで回転し、別のコンテンツが中央に移動し、上下のスワイプで同様に閲覧が可能となる。詳しく見たいコンテンツがある場合、画面下部に表示されたタイトル・サマリの表示部分をタップすると外部アプリやブラウザが起動し、コンテンツの閲覧に即座に移動する。見終わった後、デバイス上のバックボタンを押すとグラフの画面に戻り、別の結果の確認へと移ることができる。



Title  
Description

図 6.1: 起動画面



図 6.2: 検索キーワードを入力



図 6.3: 検索ボタンを押下



図 6.4: コンテンツ表示



図 6.5: 上にスワイプで上下移動

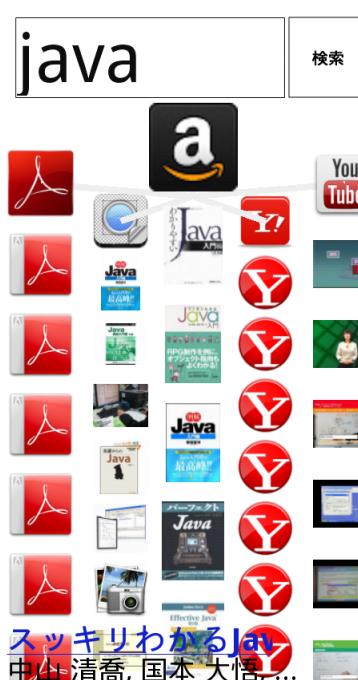


図 6.6: 横にスワイプで左右回転

## 6.2 特徴

それぞれ実現することができた、特徴について記述する。

### 6.2.1 移植性

ADOBE AIR の移植性の高さにより、Android に留まらず、iOS、PC の Web ブラウザ、PC アプリケーションとして、将来的に様々な形態、画面サイズに対応できる。

### 6.2.2 スペック性能への非依存性

非常に簡易な 3 次元 CG により製作されているため、30fps 程度の安定した高速描画を実現している。コンテンツ間の移動の際は、カメラが線形に移動するよう設計されており、滑らかに移動過程がレンダリングされるようになっている。また、すべてのアイコンにビルボード処理を施しており、どのように回転しても、すべてのコンテンツがカメラの方に正面を向いており、三次元空間上にありながらも二次元的な視界の良さを保つことが可能となった。

### 6.2.3 フリック操作、画像ノードによる直感的操作性

検索結果を上下や左右に移動することは、指先を使った移動方法として直感的に理解しやすく、項目を確認する際に発生する苦痛を軽減することが可能になる。また、グラフ表示上のノードに画像を採用したことにより、より内容が分かりやすく理解可能となった。

### 6.2.4 検索エンジンの同時検索、WWW 視覚化

e-learning コンテンツを複数の検索エンジンで同時に高精度で検索できるため、一度の検索で非常に多くの有用な検索結果を得ることができる。

# 第7章 アンケートによる評価と考察

本章では、そのアプリについての評価を行い、考察を述べる。

## 7.1 評価結果

今回は、当研究室のメンバーに対してアプリを実際に利用してもらい、アンケートを行った。その結果、以下の様な意見を得た。

- 非常に動きが独特で心地良い。
- 横や縦にスライドして動く描写が楽しく、e-learning コンテンツを見ていく面白さがあつて良い。
- パブリッシュされた一つのアプリとしてリリースするより、HTML ページとしてリリースする方が良い。より一般的に利用して貰えるようになるし、iOS やタブレット端末への移植の手間も省ける。
- PDF ファイルまでアクセスできるのは素晴らしいが、一度ダウンロードが行われてからビューワーを起動するため手間がかかるのではないか。PDF ファイルを HTML で直接閲覧できるよう構築できるようにすべき。
- タブレット等の横幅のあるデバイスならば、レイアウトを工夫して、画面遷移なしでブラウザを横に起動させたりできるのではないか。そうすればより良い使いやすさを追求できると考える。
- 検索ボックスやタイトルの表示部が透過されているため、下のアイコンと被っており見にくい。
- 検索ボックスが非常にシンプルなため、一見しただけではどのように使うのか分からぬ。説明を加えてほしい。
- 現在どの結果に注目しているのかが分かりにくいため、アイコンを拡大したり、印をつけるなどしてほしい。
- アイコンをタップすれば良いと思ってしまうが、実際には下のタイトル部分をタップしなければならないため、紛らわしい。アイコンから直接リンクできるようにすれば良いのではないか。
- アイコンが大量に並ぶと、どれが重要なコンテンツなのか判断がつかない。表示数を制限して、より重要な結果に絞ってから表示するべきである。
- 検索結果で出てきたアイコンは、単なる検索結果だけなので、その中における序列がなく最も有効なコンテンツがわかりづらい。検索結果を深く融合して、まとめてランク付けを行なってほしい。
- URL からのアイコン画像取得に失敗するエラーがある。アプリ自体の価値が極度に下がつてしまうため、なるべく改善すべき。
- 画像ファイルの読み込みが遅い。画像処理専用のサーバ等を使い、ファイルをより早く処理できるようにすれば良いのではないか。

- 全体的にどれが何を表しているのか分からづらい。ある程度慣れているユーザならば、PDFのマークには見覚えがあるが、初めて見たユーザや子供には、それが何を表しているのか、今ひとつ分からない。随所に説明を入れたり、アドバイス用のメッセージを出すなどしてほしい。
- カスタマイズ性を重視して、拡張によって検索エンジンを導入できるような機能がほしい。
- 表示形式を柔軟に変更できるようにしてほしい。画面を回転・ズームし、自分の好きな方向に簡単に移動できる機能があれば、より操作性を高めることができると考える。

## 7.2 考察

今回の解決すべき命題は、

- Mashup の特性を活かした、e-learning コンテンツ検索エンジン開発
- Android 端末上であることを意識した、ユーザビリティの高い WWW 視覚化

の 2 点であった。

### 7.2.1 e-learning コンテンツ検索エンジンとしての性能

検索結果の一覧性が高いことや、PDF データや YouTube アプリ等にダイレクトにアクセスできることを評価する声が一定数あった。この点より、当初の目的としての検索エンジン開発は概ね達成したと言えるだろう。その一方、API を通じて得た検索結果が、アイコンとしてただ広がるだけの GUI に不満があったことは否めず、よりシンプルな、ひと目で分かる検索結果を返すための仕組みを、Mashup を通じて再度構成するべきだろう。

### 7.2.2 Android 端末における WWW 視覚化とユーザビリティ

スワイプ動作の独特的な操作性や、レンダリングがスムーズで心地良く快適だとの意見を多数得られた。これらより、Android 端末としての特徴を活かし、ユーザエクスペリエンスにおいてある程度の評価を得られたと考える。しかし、ユーザビリティの高さ、という部分が多く無視されている面に関しては、意見の数々より自明のことであろう。また、納豆ビュー [6] に実装されていた、z 軸方向における回転は実現できたものの、x-y 方向への一意的な意味付け、回転操作等は実現できず、三次元空間における操作性の点に関しても、幾つかの課題が残った。より自由な操作性を実現するため、ローテートやダブルタップなど、より複雑な操作によって実現できることも検討するべきだろう。

# 第8章 結論

ここまで的研究により、e-learning コンテンツをスマートフォン端末で検索するための三次元視覚化に対して、ある程度の成果と、今後の課題や方針を得られたと考える。それらを以下に示す。

## 8.1 成果

- Mashup によって、e-learning コンテンツを素早く、便利に検索するための手法を提案できた。
- 納豆ビューを参考として、よりスマートフォン端末に最適化した形でのグラフ型 WWW 視覚化を発案できた。
- Android 端末上であることを意識し、ユーザにとって容易な操作方法を実装した。

## 8.2 今後の課題と方針

- 別のプラットフォームに対しての移植が可能であるかどうかを検討し、デバイス毎にチューニングを施していくことを考える。
- それぞれの検索結果をより深く融合して、特徴空間におけるコンテンツの特徴ベクトルに基づくシステムへ再構成し、よりシンプルに結果まで辿り着くようにする。
- ユーザが、より思ったとおりの操作を実現できるよう、メソッドや操作方法を追加・編集し、アプリとしてのユーザビリティを向上させる。
- 画像ファイルの扱いをより極小化し、応答速度の向上を目指す。
- 現在残っている数種のバグを取り除き、アプリとしての完成度をより高める。

## 謝辞

本研究を卒業論文として完成させることができたのは、担当して頂いた新井康平教授、Herman Tolle 博士研究員の熱心なご指導や、第 4 研究グループの皆様方に協力して頂いたおかげです。皆様へ心より感謝の気持ちと御礼を申し上げたく、謝辞に代えさせていただきます。

## 参考文献

- [1] 株式会社シートプランニング: 世界のスマートフォン普及予測  
<http://www.seedplanning.co.jp/press/2012/2012072601.html>, 2012.7.26.
- [2] Advenced Distributed Learning(ADL): SCORM, <http://www.adlnet.gov/capabilities/scorm>, 2004.
- [3] Apple,Inc.: iTunes U, <http://www.apple.com/jp/education/itunes-u/>, 2004.
- [4] 文部科学省: 平成十三年文部科学省告示第五十一号（大学設置基準第二十五条第二項の規定に基づく大学が履修させることができる授業等）,  
[http://www.mext.go.jp/b\\_menu/hakusho/nc/k20010330001/k20010330001.html](http://www.mext.go.jp/b_menu/hakusho/nc/k20010330001/k20010330001.html), 2001.
- [5] 原真琴: e-learning コンテンツにおけるドキュメントサーチの最適化, 2012.3.
- [6] 塩澤秀和, 西山晴彦, 松下温: 「納豆ビュー」の対話的な情報視覚化における位置づけ, 情報処理学会論文誌 Vol.38 No.11, pp.2331-2342, 1997.11.
- [7] 有限会社ジャックポット: Flowser, <http://flowser.com/>, 2005.4.
- [8] Away3D Team: Away3D, <http://away3d.com/>, 2007.
- [9] Arthur Debert: Bulk Loader, <http://github.com/arthur-debert/BulkLoader>, 2010.
- [10] Henri Torgemane: As3 Crypto, <http://code.google.com/p/as3crypto/>, 2008.6.

# 付録A プログラム

実装した ActionScript のソースコード、並びに Android アプリを定義する xml ファイルを掲載する。実装は FlashProfessionalCS6 を用いて、Android2.2 端末 (IS04) での動作を確認している。また、このプログラムのパブリッシュ、デバッグのためには以下のライブラリ、フレームワーク、画像ファイルが必要となるため、導入・設定の後実行する必要がある。

- Away3D 4.1.0 Alpha (away3d-core-fp11\_4\_1\_0\_Alpha.swc)[8]
- Bulk Loader (bulk\_loader.swc)[9]
- As3 Crypto (as3crypto.swc)[10]
- norm-top.png (128x128px の png ファイル)
- norm.png (128x128px の png ファイル)
- img-top.png (128x128px の png ファイル)
- img.png (128x128px の png ファイル)
- doc-top.png (128x128px の png ファイル)
- doc.png (128x128px の png ファイル)
- mv-top.png (128x128px の png ファイル)
- ama-top.png (128x128px の png ファイル)
- ama.png (128x128px の png ファイル)

これらソースコードパッケージの最新版は Github の以下の URL にて公開している。

<https://github.com/legnoh/ledoxea>

## A.1 Main.as(ActionScript3.0)

```
package {
import flash.display.BitmapData;
import flash.display.Sprite;
import flash.display.SimpleButton;
import flash.events.*;
import flash.filesystem.StorageVolumeInfo;
import flash.geom.Matrix;
import flash.geom.Vector3D;
import flash.text.*;
import flash.utils.ByteArray;
import flash.utils.escapeMultiByte;
import flash.net.URLLoader;
import flash.net.URLRequest;
import flash.ui.Keyboard;
import flash.desktop.NativeApplication;
import away3d.containers.*;
import away3d.entities.*;
import away3d.materials.TextureMaterial;
import away3d.primitives.*;
import away3d.textures.BitmapTexture;
import br.com.stimuli.loading.BulkLoader;
import br.com.stimuli.loading.BulkProgressEvent;
import com.hurlant.crypto.hash.HMAC;
```

```

import com.hurlant.crypto.hash.SHA256;
import com.hurlant.util.Base64;

/*
-----*
Away3D による WWW 可視化、Android による縦横方向への操作可能
@author Ryoma Kai
-----*/
public class Main extends Sprite {

    private var view:View3D;
    private var text_field:TextField;
    private var title_field:TextField;
    private var disc_field:TextField;
    private var format:TextFormat;
    private var titleFmt:TextFormat;
    private var discFmt:TextFormat;
    private var up:State;
    private var over:State;
    private var starPoints:Array;
    private var keyword:String;
    private var normXML:XML;
    private var imgXML:XML;
    private var docXML:XML;
    private var mvXML:XML;
    private var amaxXML:Array;
    private var normNS:Namespace;
    private var imgNS:Namespace;
    private var docNS:Namespace;
    private var mvNS:Namespace;
    private var amaNS:Namespace;
    private var media:Namespace;
    private var imgs:BulkLoader;
    private var yThumbLoader:BulkLoader;
    private var num:int;
    private var col:String;
    private var startZ:int;

    public function Main():void {

        text_field = new TextField();
        title_field = new TextField();
        disc_field = new TextField();
        format = new TextFormat();
        titleFmt = new TextFormat();
        discFmt = new TextFormat();
        up = new State(0x0, stage.stageWidth*0.2);
        over = new State(0xFF4500, stage.stageWidth*0.2);
        starPoints = new Array();
        keyword = new String();
        num = 0;
        startZ = -200;
        col = "norm";

        normXML = new XML();
        imgXML = new XML();
        docXML = new XML();
        mvXML = new XML();
        amaxXML = new Array();
        normNS = new Namespace("urn:yahoo:jp:srch");
        imgNS = new Namespace("urn:yahoo:jp:srchmi");
        docNS = new Namespace("urn:yahoo:jp:srch");
        mvNS = mvXML.namespace("");
        media = mvXML.namespace("media");
        amaNS = new Namespace("http://webservices.amazon.com/AWSECommerceService/2011-08-01");
        imgs =new BulkLoader();
        yThumbLoader = new BulkLoader();

        ****ここから検索ボックス、検索結果の文章表示部まわりの描画****

        //検索ボックス、ボタン、結果表示部の2行をステージに乗せて、フォーマット・スタイル定義
        setSearchBox();
    }
}

```

```

stage.addChild(title_field);
setTitleFormat(titleFmt,"");
title_field.defaultTextFormat = titleFmt;
setTitleStyle(title_field);
stage.addChild(disc_field);
setDescriptionFormat(discFmt,"");
disc_field.defaultTextFormat = discFmt;
setDescriptionStyle(disc_field);
makeButton();

/*ここから Away3D 系の描画処理*/
view = new View3D();
addChild(view);
view.backgroundColor=0xFFFFFFFF; // 背景を白色に
view.camera.lens.far=10000; // 遠くも見えるように設定

addEventListener(Event.ENTER_FRAME, function(e:Event){ view.render(); });
addEventListener(TransformGestureEvent.GESTURE_SWIPE, moveCamera );

// XML をロードより早く書き換える関係上、こっそり先にローカルの XML を読み込んでおく
var imgLoader:URLLoader = new URLLoader(new URLRequest("defImgXML.xml"));
imgLoader.addEventListener(
"complete", function(e:Event):void { imgXML = XML(imgLoader.data); });
}

/**検索結果表示後、スワイプに合わせてカメラの移動*/
private function moveCamera(e:TransformGestureEvent):void{

//どの方向にスワイプしたかを判定
if (e.offsetY == -1) {

//下から上へスワイプした場合、19 個目の検索結果でない場合、下の項目へ移動
if (num < 19) {

//下の要素へ参照を移動
num++;

//カメラ位置の変更・1 フレーム毎のレンダリング処理
for (var i:int=0; i<10; i++) {
view.camera.z-=20;
startZ-=20;
setRotation();
}
} else if (e.offsetY == 1) {

//上から下へスワイプした場合、num が 0 でないならば、上の項目へ移動
if (num > 0) {

//上の要素へ参照を移動
num--;

//カメラ位置の変更・1 フレーム毎のレンダリング処理
for (var j:int=0; i<10; i++) {
view.camera.z += 20;
startZ += 20;
setRotation();
}
} else if(e.offsetX==1){

//左から右へスワイプした場合、左列の要素に参照を移す
if(col=="norm") {

//Yahoo!検索結果の列なら Youtube 検索結果の列へ遷移
col="mv";
cameraSwipeRotate(0, 72);

} else if(col=="img") {

```

```

//画像検索結果の列なら Yahoo!検索結果の列へ遷移
col="norm";
cameraSwipeRotate(288, 360);

}else if(col=="doc"){

//ドキュメント検索結果の列なら画像検索結果の列へ遷移
col="img";
cameraSwipeRotate(216,288);

}else if(col=="ama"){

//Amazon 検索結果の列ならドキュメント検索結果の列へ遷移
col="doc";
cameraSwipeRotate(144, 216);

}else if(col=="mv"){

//Youtube 検索結果の列なら Amazon の検索結果の列へ遷移
col="ama";
cameraSwipeRotate(72, 144);
}

} else if(e.offsetX== -1){

///右から左へスワイプした場合、右列の要素に参照を移す
if(col=="mv") {

//Youtube の検索結果の列なら Yahoo!検索結果の列へ遷移
col="norm";
cameraSwipeRotate(72, 0);

}else if(col=="norm"){

//Yahoo!検索結果の列なら画像検索結果の列へ遷移
col="img";
cameraSwipeRotate(360, 288);

}else if(col=="img"){

//画像検索結果の列ならドキュメント検索結果の列へ遷移
col="doc";
cameraSwipeRotate(288, 216);

}else if(col=="doc"){

//ドキュメント検索結果の列なら Amazon の検索結果の列へ遷移
col="ama";
cameraSwipeRotate(216, 144);

}else if(col=="ama"){

//Amazon なら Youtube の検索結果列へ遷移
col="mv";
cameraSwipeRotate(144, 72);
}

}

//タイトル・Disc 部分変更・フォーマット上書き
rewriteSummary();
}

/**カメラを左右に滑らかに移動する*/
public function cameraSwipeRotate(before:int, after:int){

if( before < after ){
for(before; before <= after+1; before=before+4){
view.camera.x = -1 * Math.sin(before * (Math.PI/180) ) * 1600;
view.camera.y = Math.cos(before * (Math.PI/180) ) * 1600;
setRotation();
}
}

```

```

}else{
for(before; before >= after-1; before=before-4){
view.camera.x = -1 * Math.sin(before * (Math.PI/180) ) * 1600;
view.camera.y = Math.cos(before * (Math.PI/180) ) * 1600;
setRotation();
}
}
}

/**カメラの位置から傾きを指定する*/
public function setRotation(){
if(view.camera.y > 0){
view.camera.rotationX = 90;
view.camera.rotationY = 0;
view.camera.rotationZ = -1 * Math.atan( view.camera.x / view.camera.y ) /( Math.PI / 180);
}else if(view.camera.y < 0){
view.camera.rotationX = -90;
view.camera.rotationY = 180;
view.camera.rotationZ = -1 * Math.atan( view.camera.x / view.camera.y ) /( Math.PI / 180);
}else{
view.camera.rotationX = 90;
view.camera.rotationY = 0;
view.camera.rotationZ = 90;
if(view.camera.x > 0){
view.camera.rotationZ *= -1;
}
}
}
view.render();
}

/**検索ボックスのフォーマットを定義する*/
public function setSearchBox() {
stage.addChild(text_field);
format.align = TextFormatAlign.LEFT;// 整列
format.font = "MotoyaIMaru";
format.size = 85;// 文字のポイントサイズ
format.color = 0x0E0E0E;// 文字の色
format.kerning = true;// カーニングが有効か？（埋め込みフォント時のみ動作）
text_field.defaultTextFormat = format;
text_field.x=18;// x 座標
text_field.y=18;// y 座標
text_field.width = stage.stageWidth * 0.7;// 幅
text_field.height=100;// 高さ
text_field.type=TextFieldType.INPUT;// テキストフィールドのタイプ
text_field.antiAliasType=AntiAliasType.ADVANCED;// アンチエイリアスの種類
text_field.autoSize=TextFieldAutoSize.NONE;// サイズ整形の種類
text_field.background=false;
text_field.border=true;// 境界線があるか？
text_field.borderColor=0x0F0F0F;// 境界線の色
text_field.condenseWhite=false;// HTML 表示時にスペース改行などを削除するか？
text_field.gridFitType=GridFitType.NONE;// グリッドフィッティングの種類
text_field.multiline=false;// 複数行か？
text_field.selectable=true;// 選択可能か？
text_field.sharpness=0;// 文字エッジのシャープネス
text_field.thickness=1;// 文字エッジの太さ
text_field.useRichTextClipboard=false;// コピペ時に書式もコピーするか？
text_field.wordWrap=false;// 折り返すか？
text_field.text="";
}

/**検索結果のタイトル表示スタイルを定義する*/
public function setTitleStyle(txt:TextField) {
txt.x=5;// x 座標
txt.y=772;// y 座標
txt.width = stage.stageWidth * 0.7;// 幅
txt.height=100;// 高さ
txt.type=TextFieldType.DYNAMIC;// テキストフィールドのタイプ
txt.antiAliasType=AntiAliasType.ADVANCED;// アンチエイリアスの種類
txt.autoSize=TextFieldAutoSize.NONE;// サイズ整形の種類
txt.condenseWhite=false;// HTML 表示時にスペース改行などを削除するか？
txt.gridFitType=GridFitType.NONE;// グリッドフィッティングの種類
}

```

```

txt.multiline=false; // 複数行か?
txt.selectable=true; // 選択可能か?
txt.sharpness=0; // 文字エッジのシャープネス
txt.thickness=1; // 文字エッジの太さ
txt.useRichTextClipboard=false; // コピペ時に書式もコピーするか?
txt.wordWrap=false; // 折り返すか?
txt.text="Title";
}

/**検索結果のタイトル表示フォーマットを定義する*/
public function setTitleFormat(fmt:TextFormat,uri:String) {
fmt.align = TextFormatAlign.LEFT; // 整列
fmt.font = "MotoyaLMaru";
fmt.bold = true;
fmt.underline = true; // アンダーラインを表示するか?
fmt.size = 40; // 文字のポイントサイズ
fmt.color = 0x1122CC; // 文字の色
fmt.kerning = true; // カーニングが有効か? (埋め込みフォント時のみ動作)
fmt.url = uri; // ハイパーリンク先を文字列で指定
fmt.target = null; // ハイパーリンク先のターゲットウィンドウ
}

/**検索結果のディスクリプション表示スタイルを定義する*/
public function setDescriptionStyle(txt:TextField) {
txt.x=5; // x 座標
txt.y=814; // y 座標
txt.width = stage.stageWidth;
txt.height=100;
txt.type = TextFieldType.DYNAMIC;
txt.antiAliasType=AntiAliasType.ADVANCED; // アンチエイリアスの種類
txt.alwaysShowSelection = true; // フォーカスが無くなても選択状態を維持するか?
txt.condenseWhite = false; // HTML 表示時にスペース改行などを削除するか?
txt.multiline = false; // 複数行か?
txt.selectable = true; // 選択可能か?
txt.textColor = 0x000000; // テキストの色
txt.autoSize = TextFieldAutoSize.LEFT;
txt.useRichTextClipboard = false; // コピペ時に書式もコピーするか?
txt.wordWrap = false; // 折り返すか?
txt.text="Description";
}

/**検索結果のディスクリプション表示フォーマットを定義する*/
public function setDescriptionFormat(fmt:TextFormat,uri:String) {
fmt.align = TextFormatAlign.LEFT; // 整列
fmt.font = "MotoyaLMaru";
fmt.size = 35; // 文字サイズ
fmt.color = 0x000000; // 文字の色
fmt.underline = false; // アンダーラインを表示するか?
fmt.kerning = true; // カーニング有効
fmt.url = uri; // ハイパーリンク先を文字列で指定
fmt.target = null; // ハイパーリンク先のターゲットウィンドウ
}

/**検索ボタンを作る*/
public function makeButton() {
var btn:SimpleButton = new SimpleButton();

btn.upState = up;
btn.downState = over;
btn.overState = up;
btn.hitTestState = up;
btn.x = 360;
btn.y = 18;
btn.addEventListener(MouseEvent.CLICK, onButtonPush);
stage.addChild(btn);
}

/**星の 5 点の座標を計算する*/
public function makeStar(rad:int, grad:Number) {

var starPoint:Array = []; //xy 座標を交互に入れた配列

```

```

//1 つ目の座標
starPoint.push(rad * Math.cos(grad));
starPoint.push(rad * Math.sin(grad));

//2 つ目の座標
starPoint.push(rad * Math.cos(0.4*Math.PI + grad));
starPoint.push(rad * Math.sin(0.4*Math.PI + grad));

//3 つ目の座標
starPoint.push(rad * Math.cos(0.8*Math.PI + grad));
starPoint.push(rad * Math.sin(0.8*Math.PI + grad));

//4 つ目の座標
starPoint.push(rad * Math.cos(1.2*Math.PI + grad));
starPoint.push(rad * Math.sin(1.2*Math.PI + grad));

//5 つ目の座標
starPoint.push(rad * Math.cos(1.6*Math.PI + grad));
starPoint.push(rad * Math.sin(1.6*Math.PI + grad));

return starPoint;
}

/**5 件の検索をネットワーク越しにぶん投げて結果を XML で受け取る*/
private function onButtonPush(e:MouseEvent):void {

//現在のビューを廃棄し、新しいビューを作る
removeChild(view);
view.dispose();
view = new View3D();
addChild(view);
view.backgroundColor=0xFFFFFF; // 背景を白色に
view.camera.lens.far=10000; // 遠くも見えるように設定

//星を描く 5 点の座標を計算する
starPoints = makeStar(400,0.5*Math.PI);

//それぞれの頂点を描く
drawEdge(starPoints[0],starPoints[1],0,"norm",true);
drawEdge(starPoints[2],starPoints[3],0,"mv",true);
drawEdge(starPoints[4],starPoints[5],0,"ama",true);
drawEdge(starPoints[6],starPoints[7],0,"doc",true);
drawEdge(starPoints[8],starPoints[9],0,"img",true);

//加えてノードを描く
drawNode(0,0,0,starPoints[0],starPoints[1],0);
drawNode(0,0,0,starPoints[2],starPoints[3],0);
drawNode(0,0,0,starPoints[4],starPoints[5],0);
drawNode(0,0,0,starPoints[6],starPoints[7],0);
drawNode(0,0,0,starPoints[8],starPoints[9],0);

//5つの検索用リクエスト URL にアクセスし、結果を描画する
normSearch(1);
imgSearch(1);
docSecrch(1);
mvSearch(1);
amaSearch(1);
amaSearch(2);

//カメラの位置を Yahoo! 検索で持ってきた値のトップに据える
num = 0;
startZ = -200;
col = "norm";
view.camera.x = 0;
view.camera.y = 1600;
view.camera.z = -200;
setRotation();
}

```

```

/***Yahoo!検索(通常の検索)を行う*/
public function normSearch(page:int){

//リクエストURL
var url = "http://search.yahooapis.jp/PremiumWebSearchService/V1/webSearch?"
+"appid=LFTNFXWxg66eYMmIvFGGF1qZLo8V8HMPgqMou_HBPPhe4p5LSNCf.lkFBRjKNtE-&"
+"results=20&"
+"query=" + text_field.text + " 基礎 OR 講座";

var loader:URLLoader = new URLLoader();
var request:URLRequest=new URLRequest(url);

loader.addEventListener(Event.COMPLETE,function(event:Event) {

//XMLから名前空間を削除
normXML = new XML(event.target.data)
.removeNamespace("http://www.w3.org/2001/XMLSchema-instance");

//タイトル部分を検索1番目のものに変更・フォーマット上書き
title_field.text = normXML.normNS::Result[0].normNS::Title;
setTitleFormat(titleFmt,normXML.normNS::Result[0].normNS::Url);

//Summary部分も同様に変更・フォーマット上書き
disc_field.text = normXML.normNS::Result[0].normNS::Summary;
setDescriptionFormat(discFmt,normXML.normNS::Result[0].normNS::Url);

//テキストフォーマットに設定し直す
title_field.setTextFormat(titleFmt);
disc_field.setTextFormat(discFmt);

//検索結果に対応するノードを20個描画する
for(var i:int=0; i<20;i++){
drawEdge(starPoints[0],starPoints[1],(-2000*(page-1))+(-200*i)-200,"norm",false);
}
});

loader.load(request);
}

/***Yahoo!画像検索を行う*/
public function imgSearch(page:int){

//リクエストURL
var url = "http://search.yahooapis.jp/PremiumImageSearchService/V1/imageSearch?"
+"appid=LFTNFXWxg66eYMmIvFGGF1qZLo8V8HMPgqMou_HBPPhe4p5LSNCf.lkFBRjKNtE-&"
+"results=20&"
+"query=" + text_field.text + " 基礎 OR 講座";

var loader:URLLoader = new URLLoader();
var request:URLRequest=new URLRequest(url);

loader.addEventListener(Event.COMPLETE,function(event:Event) {

//XMLから名前空間を削除
imgXML = new XML(event.target.data)
.removeNamespace("http://www.w3.org/2001/XMLSchema-instance");

//サムネで画像を20個描画する
yThumbLoader.add("asset/img.png");

for(var i:int=0; i<20;i++){
yThumbLoader.add(imgXML.imgNS::Result[i].imgNS::Url.toString());
if(imgXML.imgNS::Result[i+1]==null){break;}
}

yThumbLoader.addEventListener(BulkLoader.COMPLETE,thumbComplete);

//失敗しやすいので、エラーの時に次に飛ぶようにしておく
yThumbLoader.addEventListener(BulkLoader.ERROR, function (e:ErrorEvent):void{

```

```

var pattern:RegExp = /https?:\/\/[-_.!~*'()]\w;\/?:@&=+$,%#]+/gi;
var failed:Array = e.text.match(pattern);
var failedUrl:String = failed[0];

//失敗したファイルを remove する
yThumbLoader.removeFailedItems();

//とりあえず今の箇所は空白にして処理し、サイズは基準のサイズに調整しておく
imgXML.imgNS::Result.(imgNS::Url==failedUrl).imgNS::Width = 128;
imgXML.imgNS::Result.(imgNS::Url==failedUrl).imgNS::Height = 128;
imgXML.imgNS::Result.(imgNS::Url==failedUrl).imgNS::Url = "";
});

yThumbLoader.start();
});
loader.load(request);
}

/**Yahoo!画像検索の BulkLoader が終わった後の処理*/
private function thumbComplete(event:BulkProgressEvent):void{

// 画像を表示
for(var i:int=0; i<20;i++){
var img = imgXML.imgNS::Result[i].imgNS::Url.toString();
var bmdWidth:int = imgXML.imgNS::Result[i].imgNS::Width.toString();
var bmdHeight:int = imgXML.imgNS::Result[i].imgNS::Height.toString();
var thumbSize:int = 128;

//.bmp .png .jpg .jpeg の拡張子を持つファイル名の正規表現
var regPattern:RegExp = /.+(bmp|png|jpg|jpeg)$/i;

// イメージがきちんと拡張子をもっている場合
// 2 の乗数倍でそれを両辺上回るまで引き伸ばしてリサイズ
if(regPattern.test(img)){
var bmd:BitmapData = yThumbLoader.getBitmapData(img);

while (thumbSize < bmdHeight || thumbSize < bmdWidth){
thumbSize = thumbSize * 2;
}

bmd = resize(bmd,thumbSize / bmdWidth, thumbSize / bmdHeight);

}else{
//持っていない場合、処理不可につきデフォルトの画像で代用
bmd = yThumbLoader.getBitmapData("asset/img.png");
bmdWidth = 128;
bmdHeight = 128;
}

//1 ピクセルズレる時があるので、それを防止するため
//両辺のサイズが違う場合、更に実寸サイズを測ってリサイズをかける
if(bmd.height!=bmd.width){
if(bmd.height > bmd.width){
bmd = resize(bmd, bmd.height/bmd.width , 1 );
}else{
bmd = resize(bmd, 1 , bmd.width/bmd.height );
}

bmd = resize(bmd, thumbSize * 2 / bmd.width , thumbSize * 2 / bmd.height );

}

var texture:BitmapTexture = new BitmapTexture(bmd);
var material:TextureMaterial = new TextureMaterial(texture);
material.alphaBlending=true;

//ビルボード処理
var sprite3D:Sprite3D=new Sprite3D(material,180,180);

if(bmdWidth!=bmdHeight){

```

```

if(bmdWidth > bmdHeight){
sprite3D = new Sprite3D(material, 180, 180 * bmdHeight / bmdWidth );
}else{
sprite3D = new Sprite3D(material, 180 * bmdWidth / bmdHeight, 180 );
}
}

view.scene.addChild(sprite3D);
sprite3D.x = starPoints[8];
sprite3D.y = starPoints[9];
sprite3D.z = /**(-2000*(page-1))+*/(-200*i)-200;

if(imgXML.imgNS::Result[i+1]==null){break;}
}

/**Yahoo!検索 (PDF のみ検索) を行う*/
public function docSecrch(page:int){

//リクエスト URL
var url = "http://search.yahooapis.jp/PremiumWebSearchService/V1/webSearch?"
+ "appid=LFTNFXWxg66eYMmIvFGGF1qZLo8V8HMPgqMou_HBPPhe4p5LSNCf.lkFBRjKNtE-&"
+ "results=20&"
+ "format=pdf&"
+ "query=" + text_field.text + " 基礎 OR 講座";

var loader:URLLoader = new URLLoader();
var request:URLRequest=new URLRequest(url);

loader.addEventListener(Event.COMPLETE,function(event:Event) {

//XML から名前空間を削除
docXML = new XML(event.target.data)
.removeNamespace("http://www.w3.org/2001/XMLSchema-instance");

//検索結果に対応するノードを 20 個描画する
for(var i:int=0; i<20;i++){
drawEdge(starPoints[6],starPoints[7],(-2000*(page-1))+(-200*i)-200,"doc",false);
}
});
loader.load(request);
}

/**Youtube 検索を行う*/
public function mvSearch(page:int){

//リクエスト URL
var url = "http://gdata.youtube.com/feeds/api/videos?"
+ "max-results=20&"
+ "q=" + text_field.text + " 基礎 OR 講座";

var loader:URLLoader = new URLLoader();
var request:URLRequest=new URLRequest(url);

loader.addEventListener(Event.COMPLETE,function(event:Event) {

//XML から必要のない 5 つの名前空間を削除
mvXML = new XML(event.target.data)
.removeNamespace("http://a9.com/-/spec/opensearchrss/1.0/")
.removeNamespace("http://schemas.google.com/g/2005")
.removeNamespace("http://gdata.youtube.com/schemas/2007")
.removeNamespace("http://www.opengis.net/gml")
.removeNamespace("http://www.georss.org/georss");

media = mvXML.namespace("media");
mvNS = mvXML.namespace("");

var thumbLoader = new BulkLoader();

// サムネイルの URL を BulkLoader に追加
for(var i:int=0; i<20;i++) {

```

```

thumbLoader.add(
mvXML.mvNS::entry[i].media::group.media::thumbnail[2].@url.toString()
);
if(mvXML.mvNS::entry[i+1]==null) {break;}
}

thumbLoader.addEventListener(BulkProgressEvent.COMPLETE,
function(event:BulkProgressEvent):void{

//画像を表示
for(var i:int=0; i<20;i++) {

//サムネイル画像の URL
var img = mvXML.mvNS::entry[i].media::group.media::thumbnail[2].@url.toString();
var bmd:BitmapData = thumbLoader.getBitmapData(img);

//256x256 にリサイズ
bmd = resize(bmd, 256/mvXML.mvNS::entry[i].media::group.media::thumbnail[2].@width,
256/mvXML.mvNS::entry[i].media::group.media::thumbnail[2].@height);

//アルファチャンネルを含むサムネイルが画像サイズをオーバーしている時のために、
//正方形になっていないファイルは更に画像サイズ確認の後リサイズし、256x256 でリサイズ
if(bmd.height!=bmd.width) {
if(bmd.height > bmd.width){bmd = resize(bmd, bmd.height/bmd.width , 1 ); }
else{bmd = resize(bmd, 1 , bmd.width/bmd.height ); }
bmd = resize(bmd, 256/bmd.width,256/bmd.height);
}

//256x256 を越えた場合、512x512 で再度リサイズ
if(bmd.height>256) {
bmd = resize(bmd,512/bmd.width,512/bmd.height);
}

var texture:BitmapTexture = new BitmapTexture(bmd);
var material:TextureMaterial = new TextureMaterial(texture);
material.alphaBlending=true;

//ビルボード処理をしつつ、20 個のノードをサムネイル画像で描画
var sprite3D:Sprite3D=new Sprite3D(material,160,120);
if(img=="") { sprite3D = new Sprite3D(material,200,200); }
view.scene.addChild(sprite3D);
sprite3D.x = starPoints[2];
sprite3D.y = starPoints[3];
sprite3D.z = (-2000*(page-1))+(-200*i)-200;

if(mvXML.mvNS::entry[i+1]==null) {break;}
}

});

thumbLoader.start();
});
loader.load(request);
}

/**Amazon 商品検索を行う*/
public function amaSearch(page:int){

var loader:URLLoader = new URLLoader();
var requestUri:String = "ecs.amazonaws.jp";
var requestPath:String = "/onca/xml";
var amazonSecretKey:String = "JwjNUQm36RDnv2T7yW8QUaVYN1CeHmogsD/rjhpe";
var timestamp:String = makeTimeStamp();

var query:String = "AWSAccessKeyId=AKIAJEEEDDCG5F452QBHA&
+ "AssociateTag=lyok-22&
+ "ItemPage=" + page + "&
+ "Keywords=" + escapeMultiByte(text_field.text)+"&
+ "Operation=ItemSearch&
+ "ResponseGroup=Medium&
+ "SearchIndex=Books&

```

```

+ "Service=AWSECommerceService&"  

+ "Timestamp=" + escape(timestamp) + "&"  

+ "Version=2011-08-01";  

    // 署名する文書  

var signatureText:String = ["GET", requestUri, requestPath, query].join("\n");  

    // 署名の生成  

var signature:String = makeSignature(signatureText, amazonSecretKey);  

    // URL エンコードした署名をクエリの最後に追加して、リクエスト URL を完成させる  

query += "&Signature=" + escapeMultiByte(signature);  

var url:String = "http://" + requestUri + requestPath + "?" + query;  

    // amazon へアクセス  

loader.addEventListener(Event.COMPLETE, function(event:Event) {  

    // XML をページに分けて配列化  

amaXML[page-1] = new XML(event.target.data);  

var thumbLoader = new BulkLoader();  

thumbLoader.add("asset/ama.png");  

for(var i:int=0; i<20;i++){  

var img =  

amaXML[page-1].amaNS::Items.amaNS::Item[i].amaNS::MediumImage.amaNS::URL.toString();  

if(img!="") {  

thumbLoader.add(img);  

}  

if(amaXML[page-1].amaNS::Items.amaNS::Item[i+1]==null){  

break;  

}  

}  

thumbLoader.addEventListener(  

BulkProgressEvent.COMPLETE, function(event:BulkProgressEvent):void{  

var bmd = thumbLoader.getBitmapData("asset/ama.png");  

// 商品画像がある場合は、256x256 でリサイズする  

// 無い場合は星画像を代理で使用する  

for(var i:int=0; i<20;i++){  

var img = amaXML[page-1].amaNS::Items.amaNS::Item[i].amaNS::MediumImage.amaNS::URL.toString();  

if(img!=""){  

bmd = thumbLoader.getBitmapData(img);  

bmd = resize(bmd,  

256/amaXML[page-1].amaNS::Items.amaNS::Item[i].amaNS::MediumImage.amaNS::Width.toString(),  

256/amaXML[page-1].amaNS::Items.amaNS::Item[i].amaNS::MediumImage.amaNS::Height.toString());  

}else{  

bmd = thumbLoader.getBitmapData("asset/ama.png");  

}  

var texture:BitmapTexture = new BitmapTexture(bmd);  

var material:TextureMaterial = new TextureMaterial(texture);  

material.alphaBlending=true;  

// ビルボード処理しつつ、商品画像のノードを描画する  

// 商品画像がある場合はそのサイズで、ない場合は星画像の大きさで調整する  

var sprite3D:Sprite3D= new Sprite3D(material,  

amaXML[page-1].amaNS::Items.amaNS::Item[i].amaNS::MediumImage.amaNS::Width.toString(),  

amaXML[page-1].amaNS::Items.amaNS::Item[i].amaNS::MediumImage.amaNS::Height.toString()  

);  

if(img==""){ sprite3D = new Sprite3D(material,200,200); }  

view.scene.addChild(sprite3D);  

sprite3D.x = starPoints[4];  

sprite3D.y = starPoints[5];  

sprite3D.z = (-200*(page-1))+(-200*i)-200;  

if(amaXML[page-1].amaNS::Items.amaNS::Item[i+1]==null){break; }

```

```

}
});

thumbLoader.start();
});

loader.load(new URLRequest(url));
}

/**col の要素と num の列で指定された Summary に書き換える*/
public function rewriteSummary(){

//col の値が norm ならば、 num の値で指定された値を Summary に置き換える
if(col=="norm") {
title_field.text = normXML.normNS::Result[num].normNS::Title;
setTitleFormat(titleFmt,normXML.normNS::Result[num].normNS::Url);
disc_field.text = normXML.normNS::Result[num].normNS::Summary;
setDescriptionFormat(discFmt,normXML.normNS::Result[num].normNS::Url);
}

//col の値が img ならば、 num の値で指定された値を Summary に置き換える
if(col=="img") {
title_field.text = imgXML.imgNS::Result[num].imgNS::Title;
setTitleFormat(titleFmt,imgXML.imgNS::Result[num].imgNS::RefererUrl);
disc_field.text = imgXML.imgNS::Result[num].imgNS::Summary;
setDescriptionFormat(discFmt,imgXML.imgNS::Result[num].imgNS::RefererUrl);
}

//col の値が doc ならば、 num の値で指定された値を Summary に置き換える
else if(col=="doc") {
title_field.text = docXML.docNS::Result[num].docNS::Title;
setTitleFormat(titleFmt,docXML.docNS::Result[num].docNS::Url);
disc_field.text = docXML.docNS::Result[num].docNS::Summary;
setDescriptionFormat(discFmt,docXML.docNS::Result[num].docNS::Url);
}

//col の値が mv ならば、 num の値で指定された値を Summary に置き換える
else if(col=="mv") {
title_field.text = mvXML.mvNS::entry[num].mvNS::title;
setTitleFormat(titleFmt,
mvXML.mvNS::entry[num].mvNS::link.(@rel=="alternate").@href.toString());
disc_field.text = mvXML.mvNS::entry[num].mvNS::content;
setDescriptionFormat(discFmt,
mvXML.mvNS::entry[num].mvNS::link.(@rel=="alternate").@href.toString());
}

//col の値が ama ならば、 num の値で指定された値を Summary に置き換える
else if(col=="ama") {

var i:int = 0;
var amaNum:int = num;
//ページ数判定
if(amaNum < 10) {
i = 0;
} else{
i = 1;
amaNum -= 10;
}

var auth:String = "";

if(amaXML[i].amaNS::Items.amaNS::Item[amaNum].amaNS::ItemAttributes.amaNS::Author[0]!=null){

//Author 項目を出す
auth =
amaXML[i].amaNS::Items.amaNS::Item[amaNum].amaNS::ItemAttributes.amaNS::Author[0].toString();

//Author が複数いる場合、2人より多い場合は3点リーダで丸める
if(amaXML[i].amaNS::Items.amaNS::Item[amaNum].amaNS::ItemAttributes.amaNS::Author[1]!=null){
auth += ", " +
}
}
}
}

```

```

amaXML[i].amaNS::Items.amaNS::Item[amaNum].amaNS::ItemAttributes.amaNS::Author[1].toString();
if(amaXML[i].amaNS::Items.amaNS::Item[amaNum].amaNS::ItemAttributes.amaNS::Author[1]!=null){
auth += ", ...";
}
}
}else{
//Author が無いときは Manufacture を使う
auth =
amaXML[i].amaNS::Items.amaNS::Item[amaNum]
    .amaNS::ItemAttributes.amaNS::Manufacturer[0].toString();
}

title_field.text =
amaXML[i].amaNS::Items.amaNS::Item[amaNum].amaNS::ItemAttributes.amaNS::Title.toString();
setTitleFormat(titleFmt,amaXML[i].amaNS::Items.amaNS::Item[amaNum].amaNS::DetailPageURL);
disc_field.text = auth;
setDescriptionFormat(discFmt,amaXML[i].amaNS::Items.amaNS::Item[amaNum].amaNS::DetailPageURL);
}
title_field.setTextFormat(titleFmt);
disc_field.setTextFormat (discFmt);
}

/**指定した 2つの Scene3D 地点間にノードを描画する*/
public function drawNode(x:int,y:int,z:int,x2:int,y2:int,z2:int) {

//引数で貰った 2 点間にラインを引く
var lines:SegmentSet = new SegmentSet();
view.scene.addChild(lines);
lines.addSegment(
new LineSegment(new Vector3D(x,y,z),new Vector3D(x2,y2,z2),0xEEEEEE,0xEEEEEE,4)
);
}

/**指定した Scene3D 地点上に指定された種類のエッジを描画する*/
public function drawEdge(x:int,y:int,z:int,src:String, top:Boolean) {

var bmd; //アイコン画像

if(src=="norm") {
[Embed(source="asset/norm.png")]
var normImg:Class;
bmd = new normImg().bitmapData;
if(top==true){
[Embed(source="asset/norm-top.png")]
var normImgT:Class;
bmd = new normImgT().bitmapData;
}
} else if(src=="img") {
[Embed(source="asset/img.png")]
var imgImg:Class;
bmd = new imgImg().bitmapData;
if(top==true){
[Embed(source="asset/img-top.png")]
var imgImgT:Class;
bmd = new imgImgT().bitmapData;
}
} else if(src=="doc") {
[Embed(source="asset/doc.png")]
var docImg:Class;
bmd = new docImg().bitmapData;
if(top==true){
[Embed(source="asset/doc-top.png")]
var docImgT:Class;
bmd = new docImgT().bitmapData;
}
} else if(src=="mv") {
[Embed(source="asset/mv-top.png")]
var mvImgT:Class;
bmd = new mvImgT().bitmapData;
} else{
}
}

```

```

[Embed(source="asset/ama.png")]
var amaImg:Class;
bmd = new amaImg().bitmapData;
if(top==true){
[Embed(source="asset/ama-top.png")]
var amaImgT:Class;
bmd = new amaImgT().bitmapData;
}
}

//画像を表示
var texture:BitmapTexture=new BitmapTexture(bmd);
var material:TextureMaterial=new TextureMaterial(texture);
material.alphaBlending=true;

//ビルボード処理
var sprite3D:Sprite3D=new Sprite3D(material,180,180);
view.scene.addChild(sprite3D);
sprite3D.x = x;
sprite3D.y = y;
sprite3D.z = z;
}

/**署名する文書と、ハッシュキーを引数として受取り、String型の HMAC-SHA256 署名データを返す*/
private function makeSignature(signatureText:String, key:String):String{

//HMAC-SHA256 署名を行うクラス
var hmac256:HMAC = new HMAC(new SHA256());

// 署名する文書のバイトデータ
var signatureTextBytes:ByteArray = new ByteArray();
signatureTextBytes.writeUTFBytes(signatureText);

// 署名用のハッシュキーのバイトデータ
var keyBytes:ByteArray = new ByteArray();
keyBytes.writeUTFBytes(key);

// HMAC-SHA256 署名を行い、ダイジェストを生成
var digest256:ByteArray = hmac256.compute(keyBytes, signatureTextBytes);

// ダイジェストを Base64 で String 型にエンコード
var signature:String = Base64.encodeByteArray(digest256);

return signature;
}

/**AmazonAPI 用タイムスタンプを生成する*/
private function makeTimeStamp():String{
var timeStamper:Date = new Date(); //現時刻
var timestamp:String = timeStamper.getUTCFullYear() + "-"+
+ to2size(timeStamper.getUTCMonth() + 1) + "-"
+ to2size(timeStamper.getUTCDate()) + "T"
+ to2size(timeStamper.getUTCHours()) + ":"
+ to2size(timeStamper.getUTCMinutes()) + ":"
+ to2size(timeStamper.getUTCSeconds());
return timestamp;
}

/**日付や時分秒が 10 未満の場合、2 桁となるよう左に 0 を挿入する*/
private function to2size( data:Number ):String{
var strData:String = data.toString();
if (data < 10) { strData = "0" + strData; }
return strData;
}

/**Bitmap データをリサイズする*/
public function resize(src:BitmapData, hRatio:Number, vRatio:Number):BitmapData{
var res:BitmapData =
new BitmapData( Math.ceil(src.width * hRatio), Math.ceil(src.height * vRatio) );
res.draw(src, new Matrix(hRatio, 0, 0, vRatio), null, null, null, true);
return res;
}

```

```

}

}

}

import flash.display.Sprite;
import flash.text.TextField;
import flash.text.TextFormat;

/**ボタンの色を与えてボタンのデザインを形作る State クラス*/
class State extends Sprite{

public function State(color:int,btnWeight:int){

graphics.lineStyle(1.0, color);
graphics.beginFill(0xFFFFFF);
graphics.drawRect(0, 0, btnWeight, 100);
graphics.endFill();

var tf:TextField = new TextField();
tf.defaultTextFormat = new TextFormat("_typeWriter", 20, color, true);
tf.text = "検索";
tf.autoSize = "left";
tf.x = (this.width - tf.width) / 2;
tf.y = (this.height - tf.height) / 2;
tf.selectable = false;
addChild(tf);
}
}

```

## A.2 ledoxea-app.xml(XML)

```

<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<application xmlns="http://ns.adobe.com/air/application/3.2">
  <id>ledoxea</id>
  <versionNumber>1.0.0</versionNumber>
  <filename>LEDOXEA</filename>
  <description>LEDOXEA is 3D view e-learning searcher.</description>
    <!-- To localize the description, use the following format for the description element.
<description>
  <text xml:lang="en">LEDOXEA is 3D view e-learning searcher.</text>
  <text xml:lang="ja">LEDOXEA は、3D 表示型の e ラーニングドキュメント検索エンジンです。</text>
</description>
-->
  <name>LEDOXEA</name>
    <!-- To localize the name, use the following format for the name element.
<name>
  <text xml:lang="en">LEDOXEA</text>
  <text xml:lang="ja">LEDOXEA</text>
</name>
-->
  <copyright></copyright>
  <initialWindow>
    <content>ledoxea.swf</content>
    <systemChrome>standard</systemChrome>
    <transparent>false</transparent>
    <visible>true</visible>
    <fullScreen>true</fullScreen>
    <autoOrients>false</autoOrients>
    <aspectRatio>portrait</aspectRatio>
    <renderMode>direct</renderMode>
    <depthAndStencil>true</depthAndStencil>
  </initialWindow>
  <customUpdateUI>false</customUpdateUI>
  <allowBrowserInvocation>false</allowBrowserInvocation>
  <icon>
    <image36x36>icon/icon_036.png</image36x36>
    <image48x48>icon/icon_048.png</image48x48>
    <image72x72>icon/icon_072.png</image72x72>
  </icon>
</application>

```

```
</icon>
<android>
    <manifestAdditions>
<! [CDATA[<manifest>
<uses-permission android:name="android.permission.INTERNET"/>
</manifest>]]>
</manifestAdditions>
    </android>
    <supportedLanguages>en ja</supportedLanguages>
    <versionLabel></versionLabel>
</application>
```