

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  
学科長： 渡邊 義明 教授

February 1, 2012

Graduation Thesis

# Development of e-learning content search system 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 2013 年度（平成 25 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特徴空間におけるコンテンツの特徴ベクトルに基づく検索システムを構築した。

キーワード： mashup、Android、e-learning

## **Graduation Thesis Overview Year 2012**

# **Development of e-learning content search system by Mashup**

Tablet and Smartphone (e.g. iOS and Android ) devices today has a fairly, those app's value is appreciated in educational field(e.g. e-learning). However, those app's GUI is inexperienced and not considered carefully. Therefore, it is meaningful to build educational support service. This research is development and verification of e-learning search app with mashup in Android.

**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 . . . . .	7
4.2.3 Star-Helix モデル . . . . .	7
4.2.4 Star-Slide モデル . . . . .	11
4.3 採用手法 . . . . .	11
<b>第 5 章 開発手法</b>	<b>12</b>
5.1 概観 . . . . .	12
5.2 Away3D . . . . .	12
5.2.1 Away3D とは . . . . .	12
5.2.2 Android 上への移植 . . . . .	12
<b>第 6 章 Android アプリ 「LEDOXEA」</b>	<b>13</b>
6.1 使用方法 . . . . .	13
6.2 特徴 . . . . .	19
6.2.1 移植性 . . . . .	19
6.2.2 スペック性能への非依存性 . . . . .	19
6.2.3 フリック操作による直感的操作性 . . . . .	19
6.2.4 検索エンジンの同時検索、WWW 視覚化 . . . . .	19

<b>第 7 章 アンケートによる評価と考察</b>	<b>20</b>
7.1 評価結果 . . . . .	20
7.2 考察 . . . . .	20
<b>第 8 章 結論</b>	<b>21</b>
謝辞	22
参考文献	23
<b>付 錄 A プログラム</b>	<b>24</b>
A.1 main.as(ActionScript・2013 年 1 月 16 日現在) . . . . .	24
A.2 app.xml(XML) . . . . .	38

# 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ものではないものの、タッチパネル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、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、プラットフォームが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努力をする必要があった。しかし、Mashup では、既存の Web サービスを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、どちらか片方がヒットした場合の結果を引き出すようにする。

## 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

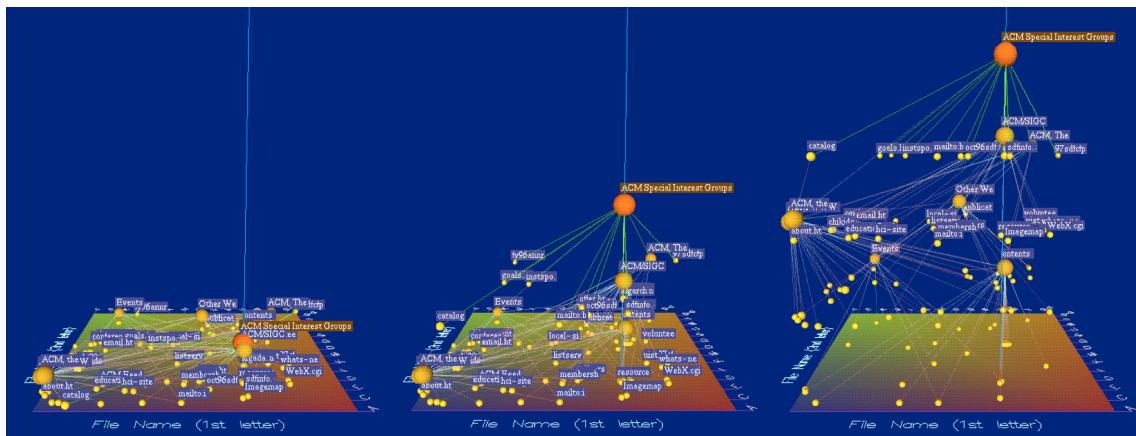


図 4.1: 納豆ビュー

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。

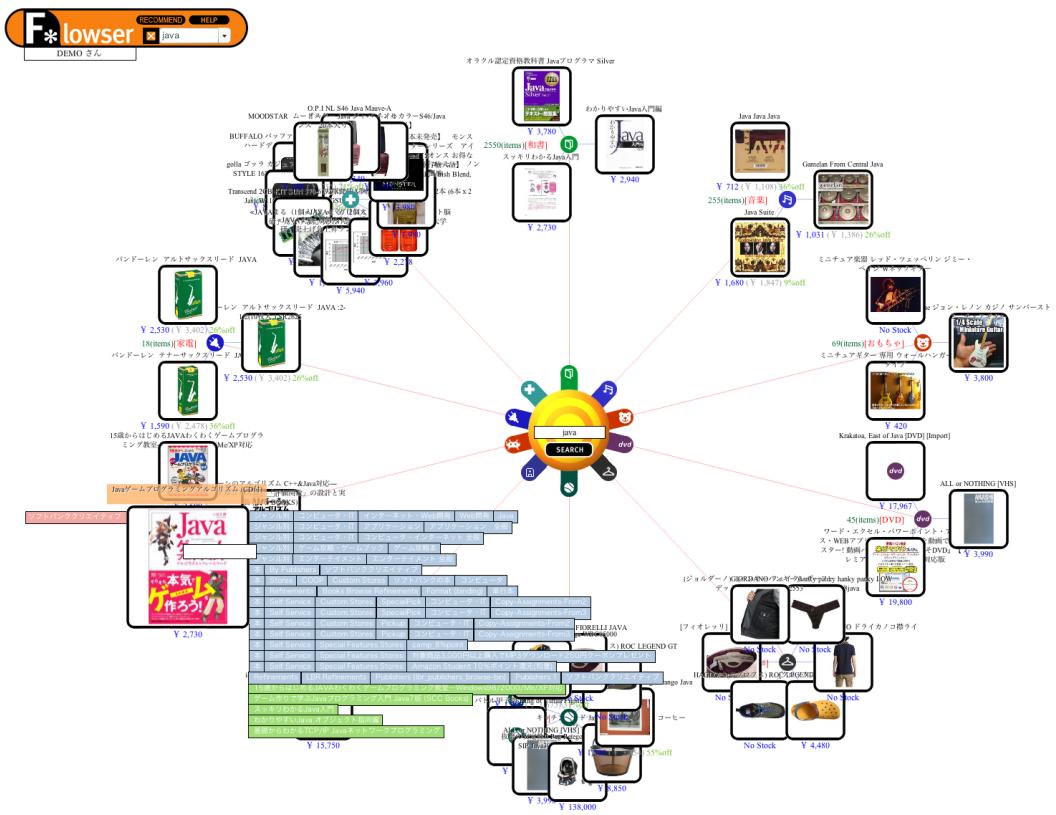


図 4.2: Flowsheet

## 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.com 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

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

複数コンテンツへの配置を最優先に考え、Flowser.com 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 model と Star model、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。

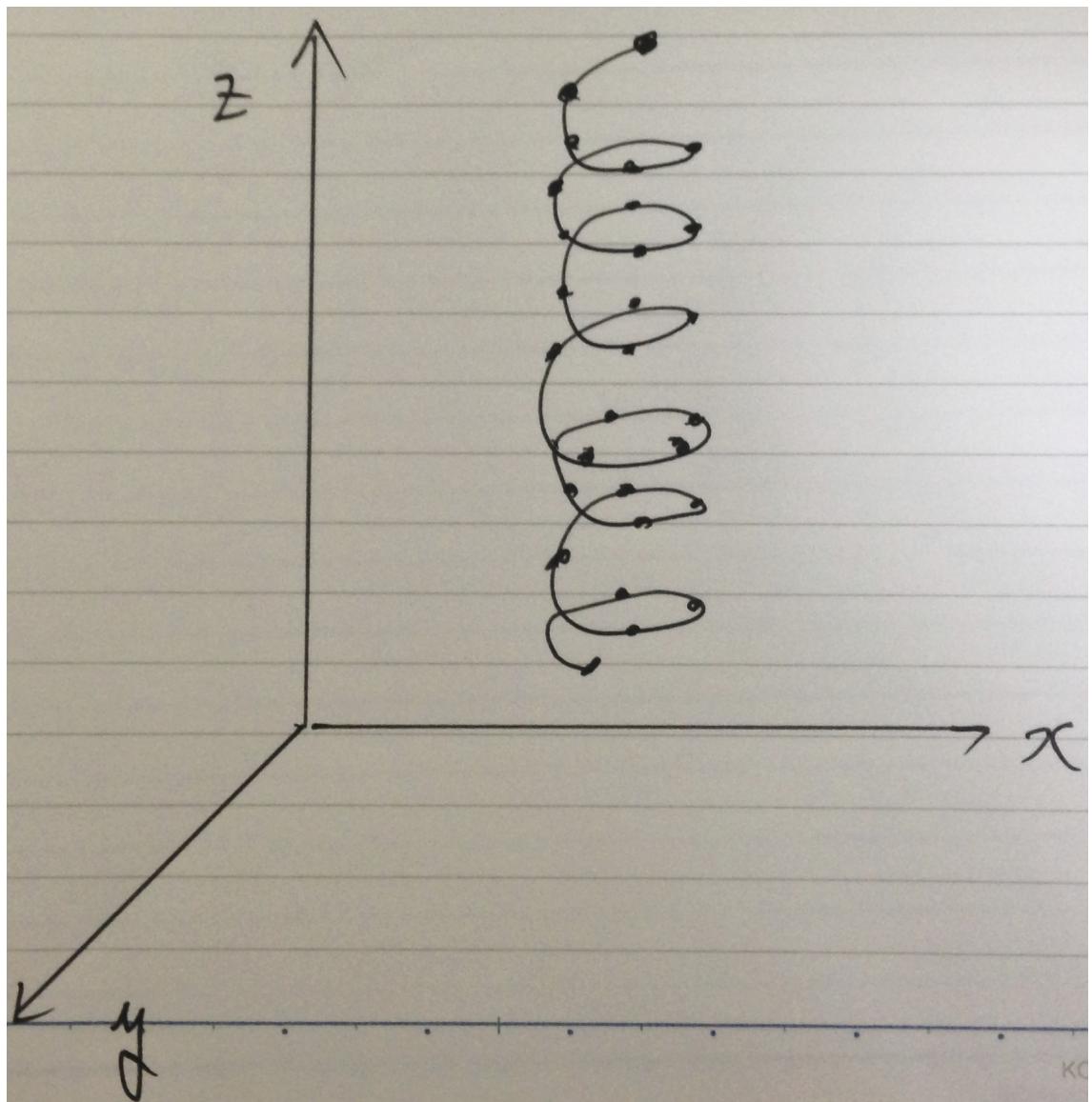


図 4.3: Helix モデル

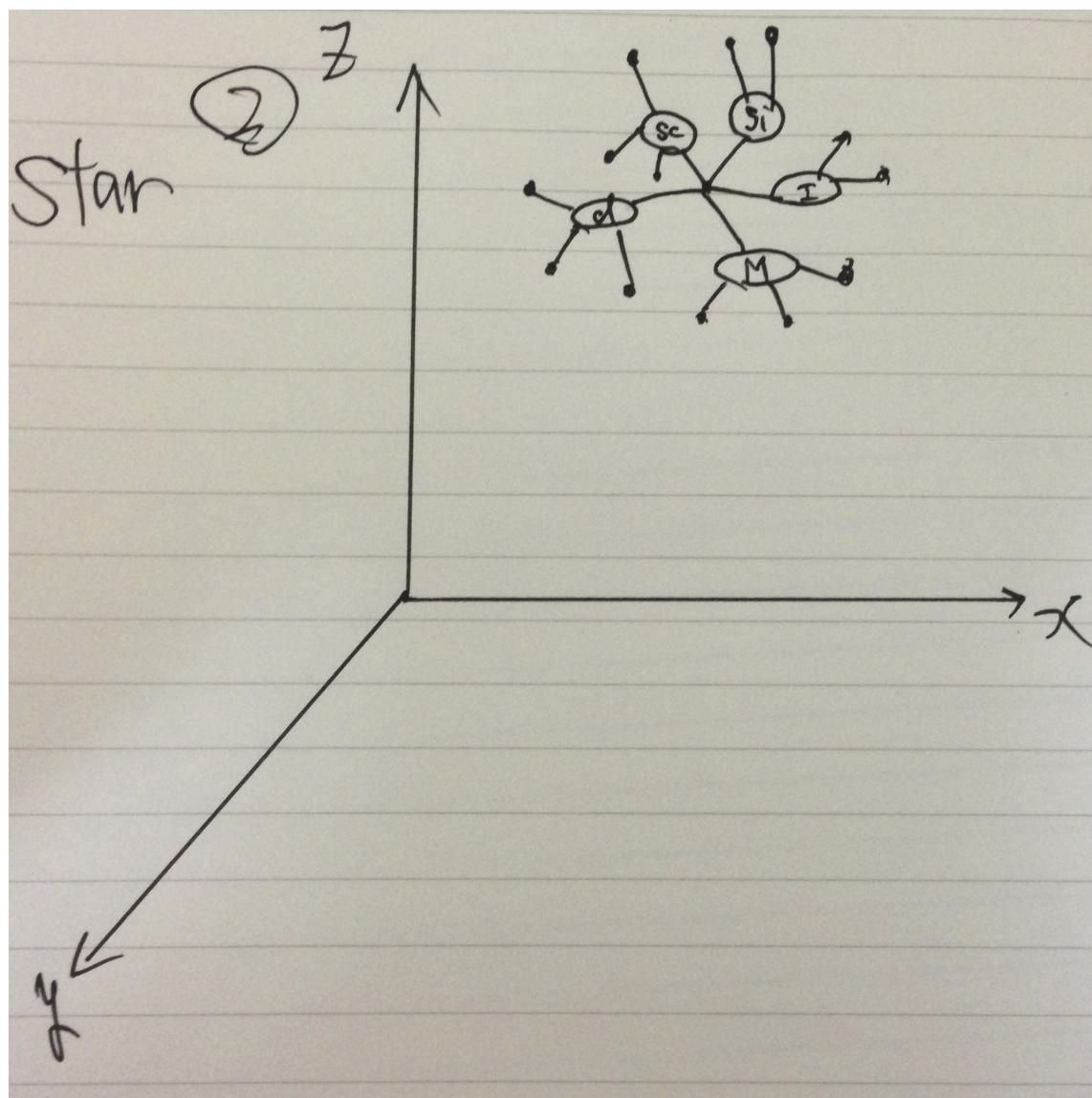


図 4.4: Star モデル

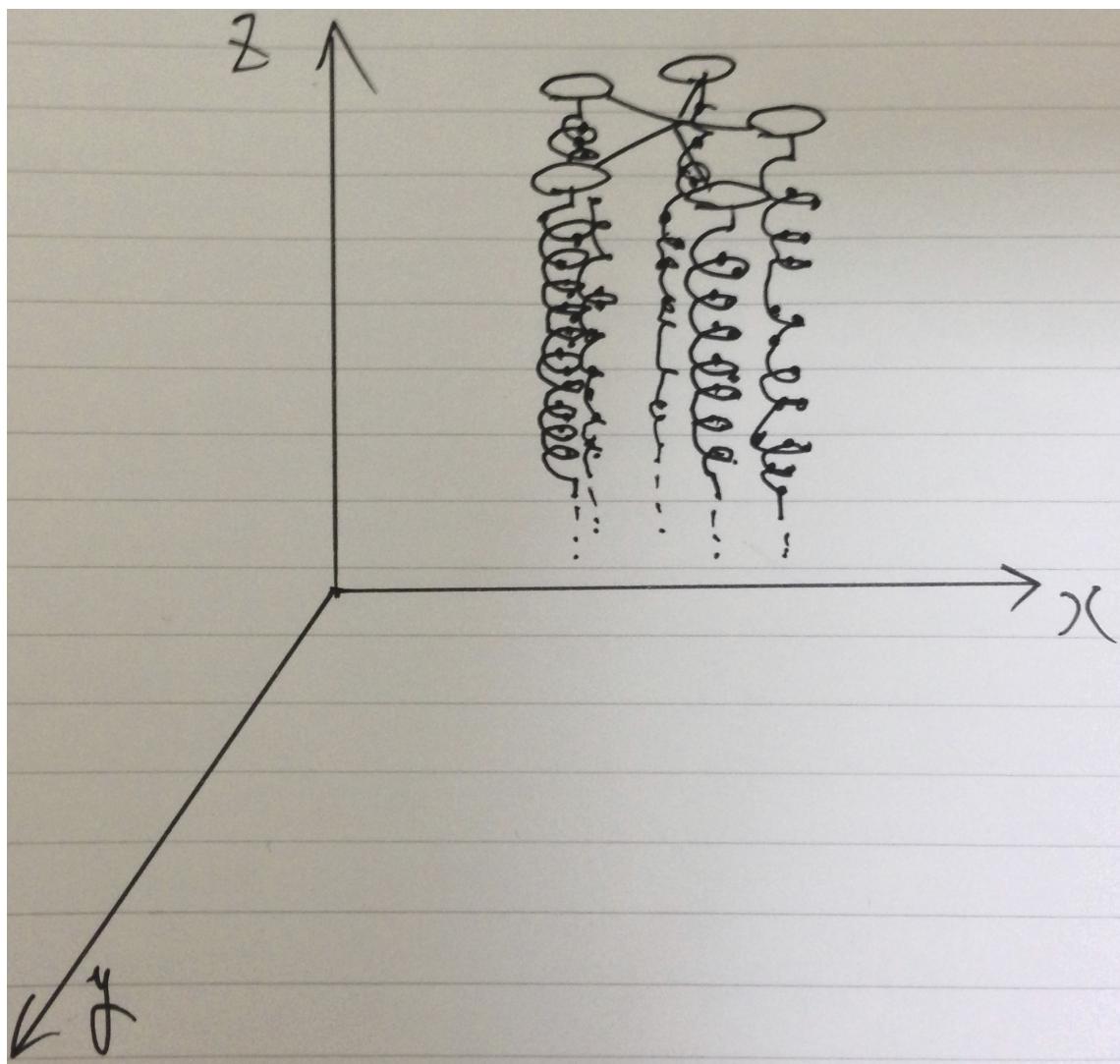


図 4.5: Star-Helix モデル

#### 4.2.4 Star-Slide モデル

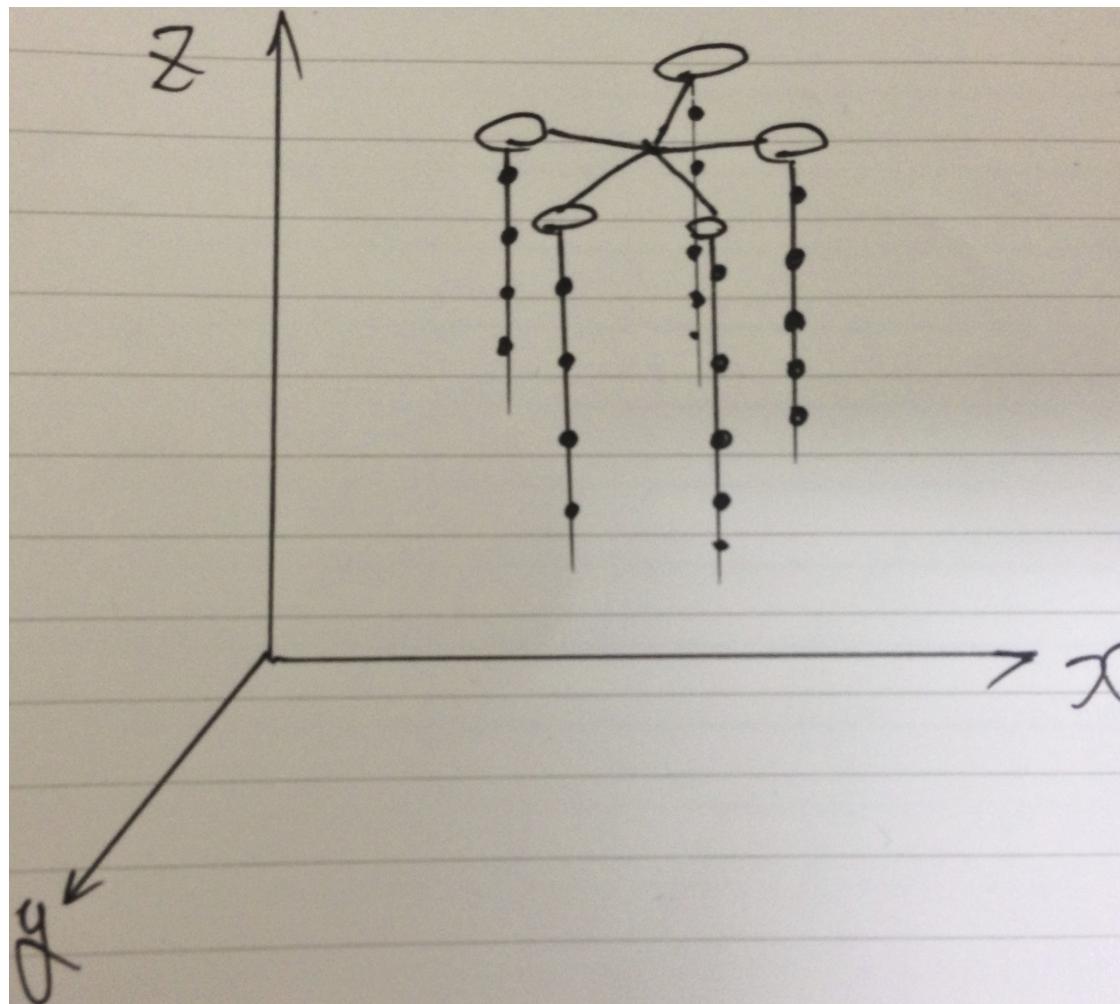


図 4.6: Star-Slide モデル

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。

### 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 model 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ます。

### 5.2.2 Android 上への移植

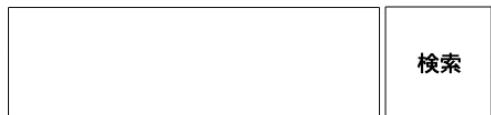
また、Away3Dの特徴として、ADOBE AIRがマルチプラットフォームに対応しているため、端末を選ばずに移植が容易であることが挙げられる。今回は開発の都合上、Androidスマートフォンでの設計を行なっているが、コードを一部手直しするだけで、iOSやPC(ブラウザ)上でも利用可能になるよう設計を行った。

# 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、上部にある検索ボックスから、検索したいキーワードを入力する。入力が完了したら、右側の検索ボタンを押下する。すると検索が行われ、合計 5 本のラインが画面上に出現する。これらを上下にスワイプすると、検索結果の上下移動が可能であり、ひとつひとつ項目を参照することが可能である。また、別のコンテンツを参照したい時は、左右にスワイプすると回転ハンガーのようにコンテンツが回転し、別のコンテンツが中央に移動する。



Title  
Description

図 6.1: 起動画面

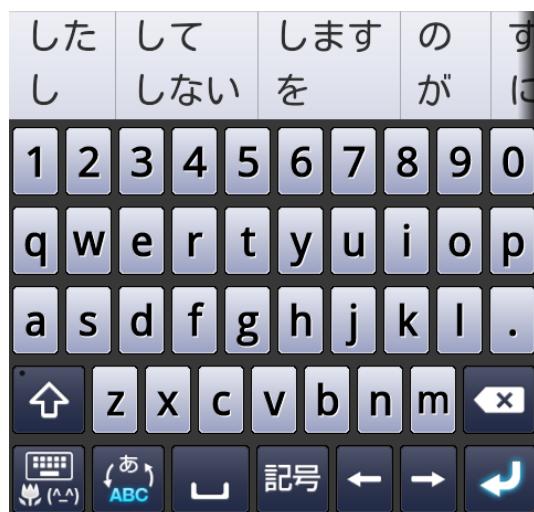
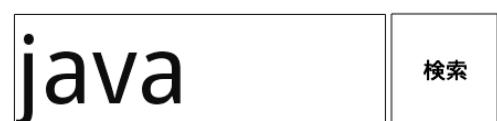
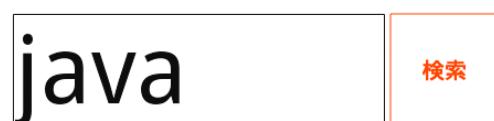


図 6.2: 検索キーワードを入力



**Title**  
Description

図 6.3: 検索ボタンを押下

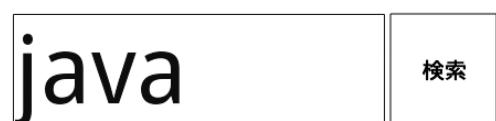


図 6.4: コンテンツ表示



図 6.5: 上にスワイプすることでコンテンツを下に見ていくことが可能



図 6.6: 横にスワイプすることでコンテンツが回転

## **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様々な形態に対応できる。

### **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、60fps 程度の安定した高速描画を実現している。

### **6.2.3 フリック操作による直感的操作性**

検索結果を上下や左右に移動するのは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に役立つ。

### **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量の収穫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。

- まだやってません…。

## 7.2 考察

考察らしきことを書きたいですがまだやってません。

## 第8章 結論

今回は Android 上での開発に終始したが、比較的移植が容易であるため、別のプラットフォームに対して積極的に移植を検討したいと考えている。また、独自に検索エンジンを追加、または削除する機能を追加することで、カスタマイズを容易にできるようにしたい。

加えて反省点として些事ではあるが、開発期間の見通しの甘さから、本来の開発期間内に開発が完了できず、いくつかの機能が未実装になったことが挙げられる。今後はスコープをしっかりと決め、プロジェクトが破綻しないよう、マネジメントを行えるようにしたい。

## 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.

## 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)
  - BulkLoader (bulk\_loader.swc)
  - as3crypto (as3crypto.swc)
  - star.png (256x256px の png ファイル)
  - pdf.png (128x128px の png ファイル)

## A.1 main.as(ActionScript · 2013年1月16日現在)

```
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 View3D {  
  
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 amaXML:XML;
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 num:int;
private var col:String;
private var startZ:int;

public function Main():void {

    /****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****/
    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 = new int();
    num = 0;
    startZ = new int();
    startZ = -200;
    col = new String;
    col = "norm";

    normXML = new XML();
    imgXML = new XML();
    docXML = new XML();
    mvXML = new XML();
    amaXML = new XML();
    imgs =new BulkLoader();

    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");

    //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****/
    backgroundColor=0xFFFFFF;// 背景を白色に
    camera.lens.far=10000;// 遠くも見えるように設定
}

```

```

// 線を作成
var lines:SegmentSet = new SegmentSet();
scene.addChild(lines);

addEventListener(Event.ENTER_FRAME, enterFrameHandler);
addEventListener(TransformGestureEvent.GESTURE_SWIPE, moveCamera);
}

/**view3D 空間のレンダリングを行う*/
private function enterFrameHandler(e:Event):void {

camera.x = 0;
camera.z = 2000;
camera.y = 0;
camera.lookAt(new Vector3D(0, 0, 0));
render();
}

/**検索ボタン押下直後の初期位置までのレンダリング (カメラの初期位置移動)*/
private function enterSearch(e:Event) {
camera.x = 0;
camera.y = 1600;
camera.z = -200;
camera.lookAt(new Vector3D(0,0,-200));
camera.rotationY = 0;

render();

removeEventListener(Event.ENTER_FRAME, enterSearch);
addEventListener(Event.ENTER_FRAME, enterMoveCamera);
}

private function enterMoveCamera(e:Event){
render();
}

/**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{

var cameraRot:Array = new Array(); // カメラの回転角を保持する行列

//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++) {
cameraRot = [camera.rotationX,camera.rotationY,camera.rotationZ];
camera.z-=20;
startZ-=20;
camera.lookAt(new Vector3D(0,0,startZ));
camera.rotationX = cameraRot[0];
camera.rotationY = cameraRot[1];
camera.rotationZ = cameraRot[2];
render();
}
}

//上から下にスワイプした場合、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++) {
cameraRot = [camera.rotationX,camera.rotationY,camera.rotationZ];
camera.z += 20;
startZ += 20;
camera.lookAt(new Vector3D(0,0,startZ));
camera.rotationX = cameraRot[0];
camera.rotationY = cameraRot[1];
camera.rotationZ = cameraRot[2];
render();
}
}

} 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";
camera.x = -1521;
camera.y = 494;
camera.lookAt(new Vector3D(0,0,camera.z));
camera.rotationX = 90;
camera.rotationY = 0;
camera.rotationZ = 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";
camera.x = 0;
camera.y = 1600;
camera.lookAt(new Vector3D(0,0,camera.z));
camera.rotationX = 90;
camera.rotationY = 0;
camera.rotationZ = 0;

} 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";
camera.x = 1521;
camera.y = 494;
camera.lookAt(new Vector3D(0,0,camera.z));
camera.rotationX = 90;
camera.rotationY = 0;
camera.rotationZ = -72;

} 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";
camera.x = 940;
camera.y = -1294;
camera.lookAt(new Vector3D(0,0,camera.z));
camera.rotationX = -90;
camera.rotationY = 180;
camera.rotationZ = 36;

} 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";
camera.x = -940;
camera.y = -1294;
camera.lookAt(new Vector3D(0,0,camera.z));
camera.rotationX = -90;
camera.rotationY = 180;
camera.rotationZ = -36;
}
}

```

```

} 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";
camera.x = 0;
camera.y = 1600;
camera.lookAt(new Vector3D(0,0,camera.z));
camera.rotationX = 90;
camera.rotationY = 0;
camera.rotationZ = 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";
//Img への遷移 (確定)
camera.x = 1521;
camera.y = 494;
camera.lookAt(new Vector3D(0,0,camera.z));
camera.rotationX = 90;
camera.rotationY = 0;
camera.rotationZ = -72;

} 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";
camera.x = 940;
camera.y = -1294;
camera.lookAt(new Vector3D(0,0,camera.z));
camera.rotationX = -90;
camera.rotationY = 180;
camera.rotationZ = 36;

} 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";
camera.x = -940;
camera.y = -1294;
camera.lookAt(new Vector3D(0,0,camera.z));
camera.rotationX = -90;
camera.rotationY = 180;
camera.rotationZ = -36;

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

//Amazon なら Youtube の検索結果列へ遷移
col="mv";
camera.x = -1521;
camera.y = 494;
camera.lookAt(new Vector3D(0,0,camera.z));
camera.rotationX = 90;
camera.rotationY = 0;
camera.rotationZ = 72;
}

}

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

/**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 = "MotoyaLMaru";
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="java";
}

/**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 = 0xFF0000;// 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 setDiscriptionFormat(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 = true; // 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 件のリクエスト URL を投げて結果を受け取る*/
private function onButtonPush(e:MouseEvent):void {

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

//星のエッジを描く
drawEdge(starPoints[0],starPoints[1],0);
drawEdge(starPoints[2],starPoints[3],0);
drawEdge(starPoints[4],starPoints[5],0);
drawEdge(starPoints[6],starPoints[7],0);
drawEdge(starPoints[8],starPoints[9],0);
}

```

```

//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);

//カメラの位置を Yahoo! 検索で持ってきた項目のトップ位置に据える
removeEventListener(Event.ENTER_FRAME, enterFrameHandler);
addEventListener(Event.ENTER_FRAME, enterSearch);
}

/**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=基礎" + text_field.text;

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=0; i<20;i++){
drawEdge(starPoints[0],starPoints[1],(-2000*(page-1))+(-200*i));
}
});

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=基礎" + text_field.text;

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
    var thumbLoader = new BulkLoader();
    thumbLoader.add("star.png");

    for(var i:int=0; i<20;i++){
        thumbLoader.add(imgXML.imgNS::Result[i].imgNS::Url.toString());
        if(imgXML.imgNS::Result[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++){
            var img = imgXML.imgNS::Result[i].imgNS::Url.toString();

            // .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、1024x1024 まで引き伸ばしてリサイズ
            if(regPattern.test(img)){
                var bmd:BitmapData = thumbLoader.getBitmapData(
                    imgXML.imgNS::Result[i].imgNS::Url.toString()
                );
                bmd = resize(bmd,1024/imgXML.imgNS::Result[i].imgNS::Width.toString(),
                1024/imgXML.imgNS::Result[i].imgNS::Height.toString());
            }else{
                bmd = thumbLoader.getBitmapData("star.png");
            }
            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,1024/bmd.width,1024/bmd.height);
            }
            if(bmd.height>256){
                bmd = resize(bmd,1024/bmd.width,1024/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,170,170);
            if(img=="") { sprite3D = new Sprite3D(material,200,200); }
            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;}
        }
    });
    thumbLoader.start();
});
loader.load(request);
}

/**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;  
  

var loader:URLLoader = new URLLoader();  

var request:URLRequest=new URLRequest(url);  
  

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

[Embed(source="pdf.png")]  

var ImageClpdf:Class; // PDF の画像  
  

//XML から名前空間を削除  

docXML = new XML(event.target.data)  

.removeNamespace("http://www.w3.org/2001/XMLSchema-instance");  
  

var bmd:BitmapData = new ImageClpdf().bitmapData;  

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 個のノードを描画する  

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

var sprite3D:Sprite3D=new Sprite3D(material,170,170);  

scene.addChild(sprite3D);  

sprite3D.x = starPoints[6];  

sprite3D.y = starPoints[7];  

sprite3D.z = (-2000*(page-1))+(-200*i)-200;  

}  

});  

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=基礎"+escape("+")+text_field.text;  
  

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); }
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=AKIAJEEDDCG5F452QBHA&
+ "AssociateTag=lyok-22&
+ "Keywords="+escapeMultiByte("test") + escapeMultiByte(text_field.text) + "&
+ "Operation=ItemSearch&
+ "ResponseGroup=Medium&
+ "SearchIndex=Books&
+ "Service=AWSCommerceService&
+ "Timestamp=" + escape(timestamp) + "&
+ "Version=2011-08-01";

// 署名の生成
var signatureText:String = ["GET", requestUri, requestPath, query].join("\n");
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 から名前空間を削除
amaXML = new XML(event.target.data);

var thumbLoader = new BulkLoader();
thumbLoader.add("star.png");

for(var i:int=0; i<20;i++){
var img = amaXML.amaNS::Items.amaNS::Item[i].amaNS::MediumImage.amaNS::URL.toString();
if(img!="") {thumbLoader.add(img);}
if(amaXML.amaNS::Items.amaNS::Item[i+1]==null) {break;}
}

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

var bmd = thumbLoader.getBitmapData("star.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.amaNS::Items.amaNS::Item[i].amaNS::MediumImage.amaNS::URL.toString();

if(img!=""){
bmd = thumbLoader.getBitmapData(img);
bmd = resize(bmd,
256/amaXML.amaNS::Items.amaNS::Item[i].amaNS::MediumImage.amaNS::Width.toString(),
256/amaXML.amaNS::Items.amaNS::Item[i].amaNS::MediumImage.amaNS::Height.toString()
);
} else{
bmd = thumbLoader.getBitmapData("star.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.amaNS::Items.amaNS::Item[i].amaNS::MediumImage.amaNS::Width.toString(),
amaXML.amaNS::Items.amaNS::Item[i].amaNS::MediumImage.amaNS::Height.toString());

if(img=="") { sprite3D = new Sprite3D(material,200,200); }
scene.addChild(sprite3D);
sprite3D.x = starPoints[4];
sprite3D.y = starPoints[5];
sprite3D.z = (-2000*(page-1))+(-200*i)-200;

if(amaXML.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::Url);
disc_field.text = imgXML.imgNS::Result[num].imgNS::Summary;
setDescriptionFormat(discFmt,imgXML.imgNS::Result[num].imgNS::Url);
}

//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;
var testin = mvXML.mvNS::entry[num].mvNS::link.(@rel=="alternate").@href.toString();
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") {
title_field.text =
amaxXML.amaNS::Items.amaNS::Item[num].amaNS::ItemAttributes.amaNS::Title.toString();
setTitleFormat(titleFmt,
amaxXML.amaNS::Items.amaNS::Item[num].amaNS::DetailPageURL);
disc_field.text =
amaxXML.amaNS::Items.amaNS::Item[num].amaNS::ItemAttributes.amaNS::Author.toString();
setDescriptionFormat(discFmt,
amaxXML.amaNS::Items.amaNS::Item[num].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();
scene.addChild(lines);
lines.addSegment(new LineSegment(
new Vector3D(x,y,z),new Vector3D(x2,y2,z2),0xFFFFFFFF,0xFFFFFFFF,4)
);
}

/**指定した Scene3D 地点上にエッジを描画する*/
public function drawEdge(x:int,y:int,z:int) {

[Embed(source="star.png")]
var ImageCls:Class; // 星の画像

//画像を表示
var bmd:BitmapData = new ImageCls().bitmapData;
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,200,200);
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(0xFFFFFFFF);
        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 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>Lodoxea is 3D view e-learning searcher.</description>
    <name>LEDOXEA</name>
    <copyright></copyright>
    <initialWindow>
        <content>Lodoxea.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_036.png</image36x36>
        <image48x48>icon_048.png</image48x48>
        <image72x72>icon_072.png</image72x72>
    </icon>
    <android>
        <colorDepth>16bit</colorDepth>
        <manifestAdditions><![CDATA[<manifest>
<uses-permission android:name="android.permission.INTERNET"/>
</manifest>]]></manifestAdditions>
    </android>
    <versionLabel></versionLabel>
    <supportedLanguages>en ja</supportedLanguages>
</application>

```