

6 楽園祭向け AR アプリケーションの作成

1 番 石川 智信, 10 番 熊谷 司, 20 番 福田 彩乃

指導教員 小野 陽子

1. はじめに

私たちは、モバイルアプリと AR について興味があった。そのためスマートフォンを使った AR アプリを作りたいと考えた。

今回、楽園祭の道案内をするモバイルアプリを作成する過程で AR の基礎研究をおこなう。

また、オリジナルの 3D モデルを作り、アニメーションをつけることで、親しみやすく視覚的に分かりやすくしていきたい。

2. 研究概要

2.1 AR とは

AR(Augmented Reality, 拡張現実)とは、カメラを通して映し出された「現実」の映像に、画像や動画、音声などの電子情報を反映(拡張)し、現実空間をコンピューターにより改変する技術である。ウェアラブル端末にも応用され、今再び注目されている技術と言えるだろう。



図 1 現実世界(右)と拡張現実(左)

AR は代表的に、

- ・GPS タイプ
- ・マーカータイプ
- ・マーカレスタイプ

の 3 種類がある。

GPS タイプは GPS(Global Positioning System, 位置情報)を利用して AR を表示するタイプである。

マーカータイプは QR コードのような AR マーカー(図 2)を利用して AR を表示するタイプである。



図 2 AR マーカー

AR マーカーは点対称な図形でなければ個人で自由に作成できる。

マーカレスタイプは特定のマーカーなどを利用することなく、現実の環境に存在している物体や、その環境自体を空間的に認識する、自由度の高いタイプの新しい技術である。



図 3 空間を認識して 3D モデルが表示される

今回はマーカータイプとマーカレスタイプについて研究していく。

2.2 開発環境

開発環境は以下のとおりである。

表 1 Windows アプリ開発環境

OS	Windows 7 64bit
IDE	Visual C++ 2008
SDK	ARToolKit

表 2 iOS アプリ開発環境

OS	Mac OS X 10.10
IDE	Xcode 6.1
SDK	Metaio SDK
使用機材	iPhone5s, iPad Air

3. 仕様 SDK, IDE について

3.1 ARToolKit について

ARToolKit は AR の研究のために開発されたソフトウェアライブラリであり、奈良先端科学技術大学院大学の加藤博一教授によって開発され、ワシントン大学の Human Interface Technology Laboratory (HIT Lab) によってサポートされている。

今回使用する ARToolKit はマーカータイプ AR に対応している。

3.2 Metaio SDK について

Metaio SDK は、iPhone/iPad(iOS), Android, Windows 向けの AR アプリを作成するための開発キットである。高レベルの API と最新の画像認識技術により、開発者は高品質なアプリを生成することが出来る。

Metaio SDK は 3 種類すべての AR に対応しているが、今回はマーカレス AR を使用する。

Metaio は画像や地形情報と、モデルの関係をトラッキングファイルというもので定義している。

トラッキングファイルは xml で記述されているため、手動でも作成することができる。しかし手動で作成するのは困難なため、トラッキングファイル作成支援ソフトの Metaio Creator というソフトを利用した。

このソフトはシェアウェアで、画像とモデルの関係を 2 つまでしか作成できない。そのため本研究では作成されたものの一部を利用して半手動でトラッキングファイルを作成していく。

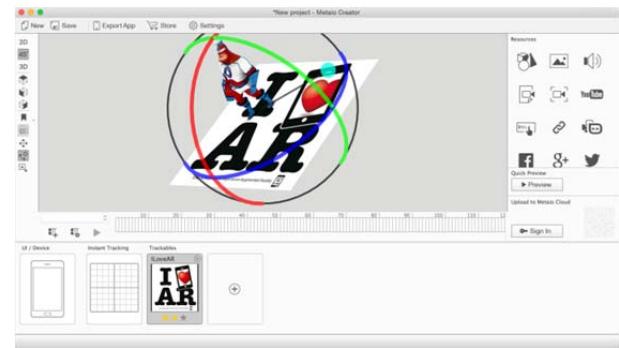


図 4 Metaio Creator の画面

3.3 Xcode について

Xcode とは Apple 製の Mac 専用統合開発環境(IDE)である。

Macintosh にて Mac OS X あるいは iOS 用アプリケーションを開発する場合に Xcode が必要になる。

エディタ画面は eclipse と似た画面である。

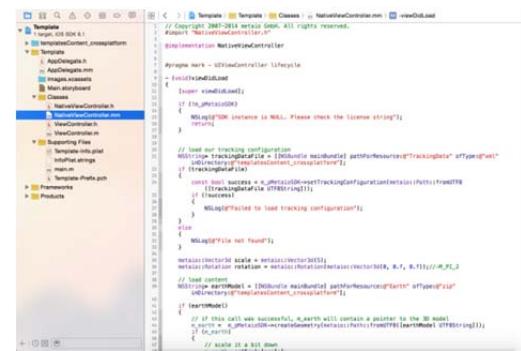


図 5 Xcode のエディタ画面

ストーリーボードと呼ばれるアプリの画面デザインを作成するものがある。これは Visual Studio でアプリケーションを作成する際のデザインを編集する感覚と似ているため、視覚的な部分を思ったように作成することができる。さらに、ある程度の画面遷移などをプログラミング不要で作成できる。

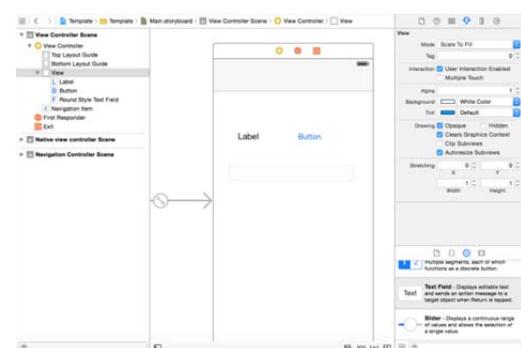


図 6 ストーリーボードの編集画面

4. 3D モデルリングソフトについて

4.1 Blender について

Blender とは 3 次元コンピュータグラフィックスソフトウェアの一つで 3D モデルの作成、レンダリングのほかアニメーション、コンポジット機能も備える。



図 7 Blender の画面

特徴として、第一にオープンソースである事が挙げられる。世界中の有志が開発をおこなっており、開発に参加するのも、商用、非商用問わず Blender を使用するのも自由である。

4.2 Blender を選定した理由について

Blender のほかに有名な 3D モデリングソフトで、Metasequoia というものがある。

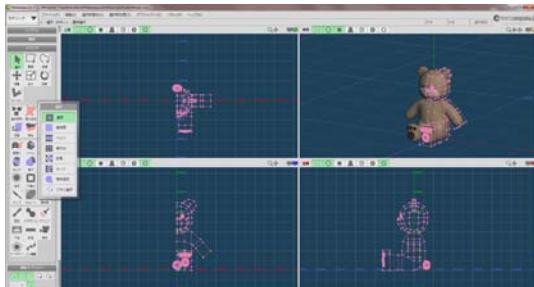


図 8 Metasequoia の画面

Metasequoia は MQO ファイルを標準で利用できる。しかし今回の研究ではこちらは採用しなかった。理由は、Metasequoia がシェアウェアであるため、ライセンスを購入しないと機能が制限されているからだ。また、Blender にはアニメーションを作成する機能がある。



図 9 Blender のアニメーション作成画面

そういった理由で、Metasequoia でのモデル作成を断念し、Blender を使用することとした。

ARToolKit で使用するモデルは MQO ファイルであるため、ネット上で配布されている”メタセコイア(.mqa)インポーター／エクスポート”を利用して書き出す。

Metaio で使用するモデルは Blender で利用できる OBJ ファイルで作成する。

5. 進捗状況

5.1 楽園祭での展示について

平成 26 年度の楽園祭にて、途中まで完成したアプリケーションを説明パネルとマーカー 6 つ付加した校内案内図と一緒に展示し、アプリのデモを行った。.



図 10 展示の様子

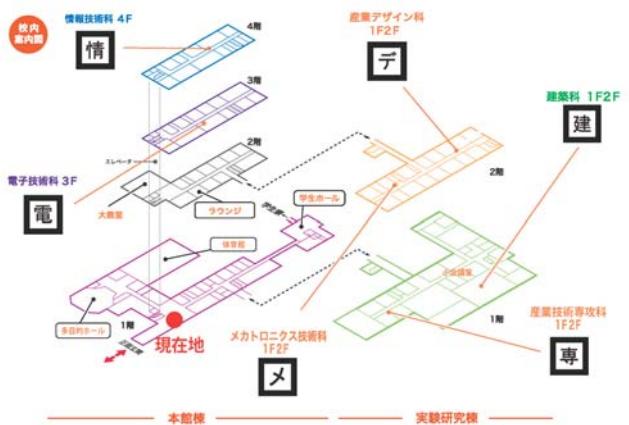


図 11 展示した案内図

このアプリケーションは、複数のマーカーをカメラで読み取り、それぞれに対応した 3D モデルを表示するプログラムである。



図 12 展示したアプリの実行イメージ

また、表示してある 3D モデルも作成した。

デモでは、アプリケーションを起動し実際に来場者の方にカメラでマーカーを読み取ってもらい、PC の画面に 3D モデルを表示させた。カメラを持ちマーカーにかざすだけなので子供でも簡単に使用してもらうことが出来た。



図 13 表示させた 3D モデルの例

5.2 Windows 版アプリの完成イメージ

楽園祭で展示した、6 つのマーカーを付加した校内案内図に、案内マーカーを追加する。

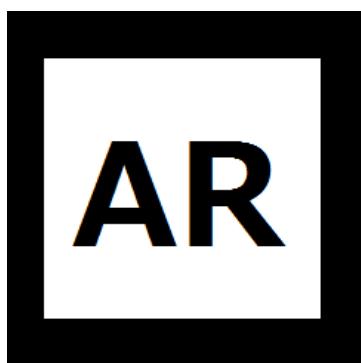


図 14 追加する案内マーカー

案内モデルは目的の科がクリックされると、その科のマーカーがあるところまで歩く。目的の科の場所まで到着すると、その科のモデルとバトン

タッチし、案内を交代する。



図 15 案内モデルのイメージ

今後は、目的の科をクリックした時にクリックした位置を判定し、それぞれのアクションを実行させるマウス入力処理の追加と、3D モデルの作成と歩くアニメーションの作成をしていきたいと考えている。

現在はマウス入力処理の理解と研究に取り掛かっている。

5.3 iOS 版アプリの完成イメージ

楽園祭後、私たちは iOS 版のアプリケーションの作成を検討した。

iOS は現在大きく分けると iPhone と iPad の 2 種類に分かれる。今回作成するアプリケーションではそれぞれレイアウトの異なるものを作成する。

5.3.1 iPhone での動作

メイン画面ではどこの科へ行きたいかを選択する。選択するとカメラが起動する。ポスターにある(マーカー)をカメラで読み取ると 3D モデルが表示される。モデルには看板を持たせて、そこからの案内を文字で表示する。



図 16 iOS 版のメイン画面とカメラ画面

iOS 版ではさらにかくれんぼのような小さな子どもたちでも楽しめるような機能を追加したいと考えている。



図 17 空間を認識している様子

図 17 のように空間を認識して 3D モデルを表示することができるため、校内の地形を利用して 3D モデルを表示させる。3D モデルを探して校内を歩き回るので校内探索、といった意味ではいいのではないかだろうか。また、かくれんぼに似た要素があるため、小さな子どもでも楽しめると考えている。

現状は、案内の面で力不足である。案内方法を再度検討し、実際に楽園祭に来た人が利用した時に目的の科までたどり着けるように改良したいと考えている。

5.3.2 iPad での動作

iPad は iPhone とは異なり複数の View を表示することができる。そのため iPad 版では広い画面を最大限利用し、極力画面遷移を少なく、サブビューを用いてカメラ画面と地図画面を同時に表示できるようにした。



図 18 iPad 版の画面

図 18 のように地図画面の View の中にカメラ画面の View を配置し、カメラ画面の View のみ非表示にできるようにした。非表示にする処理はスト

ーリーボードでは行うことができないため、直接プログラムに処理をした。

しかし、これらの View はそれぞれ異なるプログラムで動いているため、再度カメラ画面を表示すには地図画面からデータを送る必要がある。データの受け渡しには基本的に Delegate を用いる。

Delegate は、あるクラスだけでは処理できない命令を、そのクラスの変わりに行うクラスに引き渡すことができる。

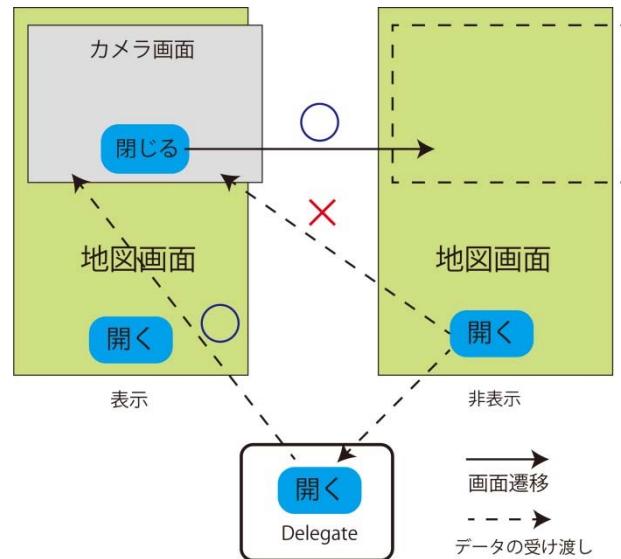


図 19 データの受け渡し

図 19 のように Delegate を経由することで再度カメラ画面を表示することができる。

現在、地図画面には地図を表示する処理を行っていないため、カメラ画面と同様に別 View を置き、画像を表示させようと考えている。



図 20 構成図

6. 分担について

Windows 版アプリケーションはある程度のプログラムができているため、1 人で作業を行うこと

とする。

iOS 版のアプリケーションについては、iPhone 版、iPad 版とあるため、二人で作業を行う。1人は画面遷移の簡単な iPhone 版と両方で共通なプログラム、トラッキングファイルを作成していく。もう1人は画面遷移の難しい iPad 版の主にレイアウトの部分を担当する。

3D モデルに関しては、Windows 版の担当者が行う。

7. 今後の作業計画

Windows 版、iOS 版ともに実際に楽園祭の時に使用できることを目指とする。

Windows 版は未実装の機能について研究を進める。また 3D モデルにアニメーションを付加させる。

iOS 版は App Store に公開して、実際ダウンロードできるようにしたい。また、地形を利用したものがあるため、地形データを採取することが必要である。アプリケーションの完成度を高めるとともに地形データを採集する。

8. 参考文献

ARToolKit 拡張現実プログラミング入門

(アスキーメディアワークス

<http://books.ascii.jp/9784048673617/>

3DCG ソフト&3D プリンタで作ろう

(株式会社ソーテック)

Blender – Wikipedia

<http://ja.wikipedia.org/wiki/Blender>

Metasequoia – Wikipedia

<http://ja.wikipedia.org/wiki/Metasequoia>

工学ナビ

<http://kougaku-navi.net/ARToolKit/>

とある PG の研究記録Ⅱ

<http://nullorempty.jimdo.com/>

iOS7 と metaio で AR アプリを作ろう

<http://dev.classmethod.jp/smartphone/iphone/ios-metaio-1/>

ドットインストール

<http://dotinstall.com/>

metaio Developer Poertal

<http://dev.metaio.com/sdk/getting-started/ios/creating-a-new-ar-application/>