

### 3 Android・iOS 向け AR アプリケーションの作成

工藤 大輝, 坂本 恒一, 藤原 海斗

指導教員 小野 陽子

#### 1. 目的

AR について興味があり、スマートフォンを使った AR アプリを作成したいと考えた。

今回作成するアプリは、昨年度の卒業研究で作成された「楽園祭向け AR アプリケーション」を基に作成する。

詳細として、Android・iOS 用アプリの作成・拡張、3D モデルの作成を行う。

#### 2. 研究概要

##### 2.1 開発環境

開発環境は以下のとおりである。

表 1 Android 用アプリ開発環境

OS	Windows 7 Ultimate
IDE	Android Studio 1.2
使用言語	Java
使用機材	実機(HTC J butterfly HTL21)
SDK	Android SDK, VuforiaSDK

表 2 iOS 用アプリ開発環境

OS	Mac OS X 10.11.1
IDE	Xcode 7.1
使用言語	Objective-C
使用機材	iPhone4s
SDK	VuforiaSDK

表 3 3D モデル用開発環境

OS	Windows7 Ultimate
ツール	Metasequoia 4.5.2

#### 2.2 AR について

AR(Augmented Reality, 拡張現実)とは、バーチャルリアリティ(VR)の変種であり、その周囲を取り巻く現実環境に情報を付加・削除・強調・減衰させ、文字通り人間から見た現実世界を拡張するものを指す。

バーチャルリアリティが人工的に構築された現実感と現実を差し替えるのに対し、拡張現実は、現実の一部を改変する技術である。拡張現実感、強化現実、増強現実とも言う。

##### 2.2.1 Vuforia の利用

利用予定だった Metaio SDK が利用できなかつたため、Qualcomm 社による AR プラットホーム Vuforia を利用する。Vuforia は、Android・iOS・Unity に対応し、無料で利用することができる。

また、AR で用いるマーカーについても、サイトへ登録さえすれば、自分のデータベースを作成し、そこにマーカーとして利用したい画像をアップロードすることで、アプリに組み込めるような形に変換されたものをダウンロード出来るようになる。



図 1 今回マーカーとして使用した画像

### 2.3 Android Studioについて

Android Studio(アンドロイド スタジオ)とは、Googleが提供するAndroidプラットフォームに対応する統合開発環境である。

従来 Eclipse と ADT(Android Developer Tools)により実現されていたAndroidアプリの開発環境が、Android Studioを用いることで実現できる様になった。

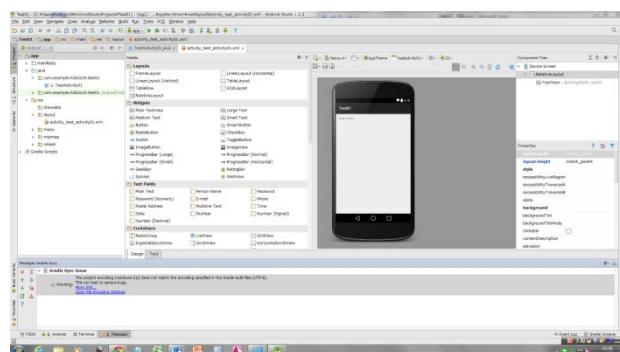


図2 Android Studio の画面

### 2.4 Xcode とは

ソフトウェアを開発するためのアップルの統合開発環境(IDE)であり、Mac OS Xに付属する形で配布されている。

Xcode6までは、実機デバッブをするために開発者登録(有料)をする必要があった。

しかし、Xcode7からは開発者登録をしなくても実機デバッブができるようになった。

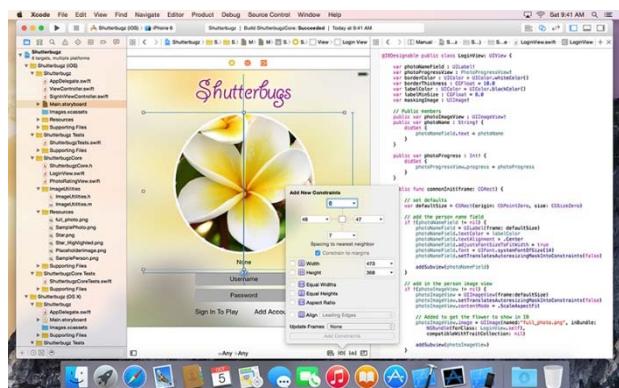


図3 Xcode の画面

(公式 HP より)

### 2.5 3D モデル作成ツール

3D モデルを作成するツールとして Metasequoia を使用している。

Metasequoia(メタセコイア)とは、Windows用3DCGソフトウェアである。ポリゴンメッシュ(3次元コンピュータグラフィックスとソリッドモデリングの多面体オブジェクトの形状を定義する頂点、辺、面の集合のこと)による3Dモデルの作成(モデリング)に特化している。データ可搬性を重視しており、様々なファイル形式で出力することができる。

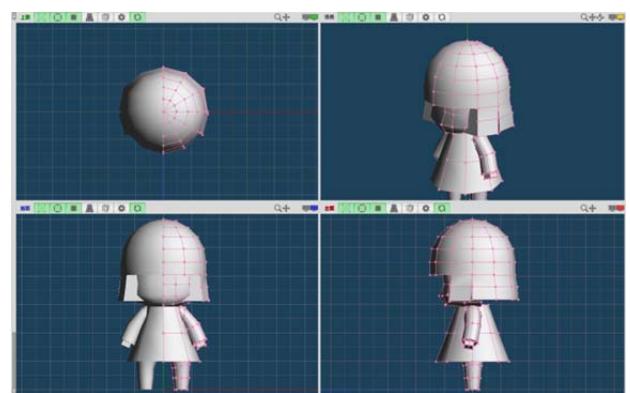


図4 Metasequoia の画面

### 3. 進捗状況

#### 3.1 3D モデルについて

##### 3.1.1 3D モデルの下絵

オリジナルの 3D モデル作成のため、過去の先輩が作成した「さんタン」を 3D モデル化できるよう、画像を元に作成した。



図 5 さんタン(正面図)

しかし、さんタンの画像は正面図のみで、1 枚だけでは作成が困難なため、横から見た図を想像して書いてもらい、正面図とあわせて下絵として利用した。



図 6 さんタン(左図)

#### 3.1.2 作り方・作成箇所

今回モデルを作るためにとった方法として、2 枚の下絵を配置し、作りたい部位・パーツにあわせて円柱、立方体など大まかな形の図形を配置。その後各頂点を動かす、面を押し出すなど、図形を動かしていき、作りたいパーツの形に近づけていく。

今回、このような方法で胴体、頭、腕、足、スカート、髪、腕、手、帽子、杖を作成した。

#### 3.1.3 進捗

各パーツを作成し、ある程度の形まで作成できた。

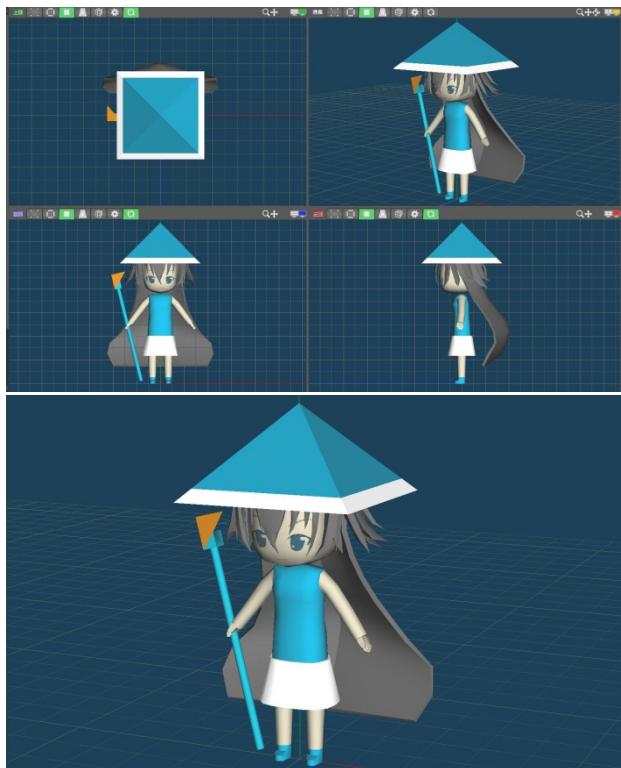


図 7 現在までのモデル作成状況

### 3.1.4 MMD の作成

ある程度形になってきたため、作成したモデルを動かすことに挑戦した。今回は MMD で動かすために、pmd(pmx)ファイルの作成に取り組んだ。

MMD とは、キャラクターの 3D モデルを操作しコンピュータアニメーションを作成する 3DCG ソフトウェアである。

#### ① ボーンの設定

モデルを動かすための土台となるボーンは、既存の 3D モデルを Metasequoia で読み込み、使われているボーンの設定を xml ファイルとして出力した。そのファイルを配置し、さんタンの形に合うよう変更を加えた。

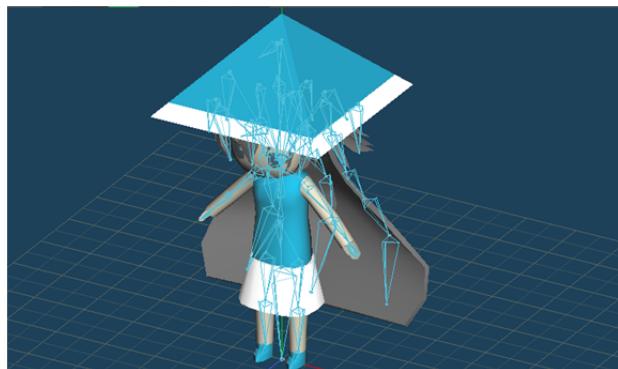


図 8 ボーン設定の様子

#### ② ウェイトの設定

ボーンの配置後、どのボーンでどのパーツが動くのか指定するためにウェイトの設定を行った。パーツによっては、他のパーツで隠れている部分があるため、一時的に見えないようにしながら設定していく。

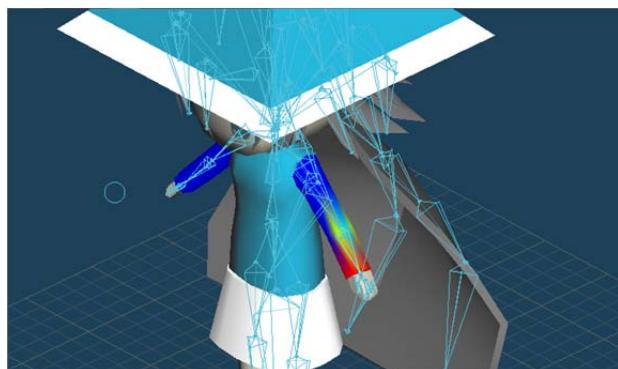


図 9 ウェイト設定の様子

ウェイトをつけ終わり、Metasequoia 内でボーンとモデルがうまく連動していることを確認したら、pmd ファイルで出力し、MMD で表示した。

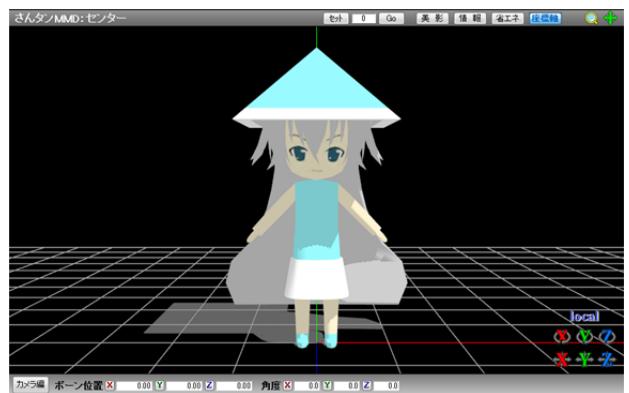


図 10 MMD で表示したモデルの様子

### 3.1.5 今後の作業計画

ある程度形もでき、モーションをつけて動かすこともできるようになった。しかし、動かした際にロングヘアのパーツ大きいために他のパーツを貫通するといったことが起きた。pmx エディタ等、他のアプリケーションを使用し剛体とジョイントを設定することによって、物理演算によって髪などを自然に動かすことができるが、修正時間がかかるため、他で修正したい点が見つかれば直していきたい。

### 3.1.6 研究を通して

今回、プログラミングではなく 3D モデルの作成に主に取り組んだ。どのようにパーツを作成すれば画像のように見えるのか想像し作成することが大変だが、していく過程も目に見えているのでその分達成感も得ることができた。

### 3.2 Android アプリについて

#### 3.2.1 Android Studio へ Vuforia を導入

Android Studio に Vuforia SDK を導入した。その際に、Vuforia への登録、ライセンスキーの取得をした。また、Android 端末用のサンプルプログラムの実行をさせることができた。

#### 3.2.2 アプリの開発

サンプルプログラムを基に Android アプリケーションの作成を開始した。内容としては、昨年の研究であった AR アプリの移植、追加機能の実装を目標としていた。今の時点でのアプリの流れとしては、以下のようになる。

##### ① アプリの起動

アプリケーションを起動して最初の画面では「産技短」の文字が表示される。



図 11 アプリ起動時の画面

##### ② メニュー画面の表示

行きたい科を選択させる画面。縦スクロールで、項目を選択できるようになっている

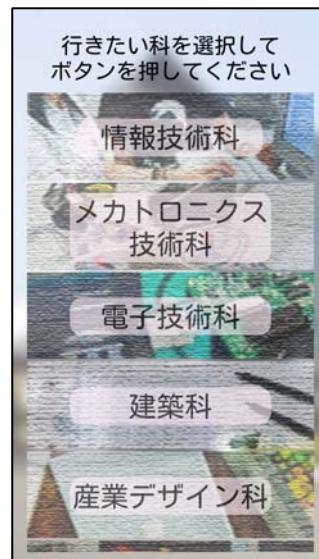


図 12 行きたい科を選択させる画面

##### ③ カメラの起動

ボタン押下により、カメラが起動する。

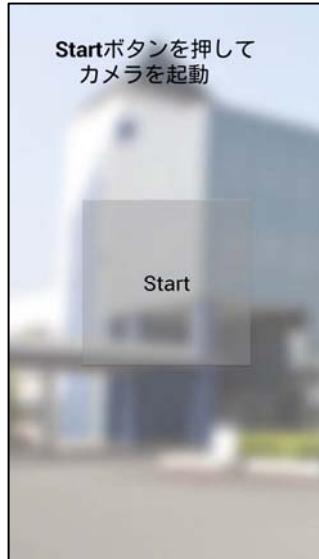


図 13 カメラを起動する画面

#### ④ AR の表示

カメラでマーカーを認識させ、3D モデルを表示させる。マーカーは任意のものに変更する事が可能である。3D モデルは自作のものとなっている。モデルに本来貼り付けるテクスチャの設定は試行中のため、現在は仮のテクスチャを設定している。

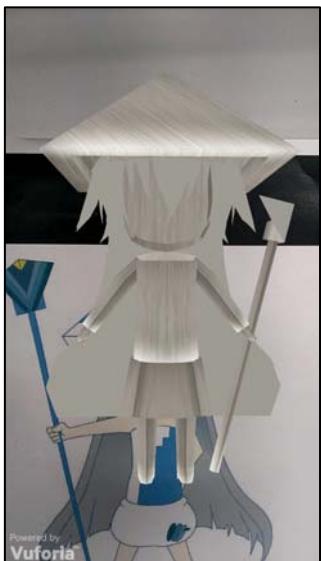


図 14 マーカーを認識している様子

#### 3.2.3 3D モデルの読み込み

3D モデルのファイル形式としては、x ファイルを使用している。ファイルから、頂点座標、インデックスバッファ(各ポリゴンを構成する頂点)の情報、法線の情報、UV 座標の合わせて 4 つを用いてモデルを表示している。ファイルから上記の情報を読み込むためのプログラムは作成した。

しかし、今のところ、法線の情報のみ、ファイルからの取得が可能となっており、他の 3 つのものについては、直接座標等の情報をプログラム内に組み込む形となっている。そのため、これらもファイルから取得できるよう調整していく。

#### 3.2.4 今後の作業計画

課題としていた 3D モデルの変更が出来るようになつたものの、テクスチャやファイルからの読み込み等、まだ不完全な点がある。

また、道案内の機能については完成しておらず、残りの期間は、モデルの表示と UI を完成に向けて取り組んでいきたい。

#### 3.2.5 研究を通して

Android で AR アプリの開発という事で、行き詰まる点も多く、予定通りに進めることができなかつた。要因として、Vuforia のサンプルプログラムを改良しながら作成していくという方法をとっていた事がある。これによりプログラムを作成していく中で、書かれている処理の内容を読み解きながら、自分のやりたいように変えるという事に、多くの時間が掛かってしまった。

また、追加機能の実装に至ることが出来ず、案で終わる結果となつてしまつた。

## 3.3 iOS アプリについて

## 3.3.1 Vuforia の導入

VuforiaSDK を導入し、サンプルプログラムが動作できるようにした。

Xcode 単体で動かしている資料が全くなく、Unity と組み合わせたものがほとんどだったのでサンプルを動かすのに大分手間取ってしまった。

## 3.3.2 アプリの概要

Android 同様に、昨年作成されたアプリに追加機能を実装する予定だったが、Metaio のサービス終了によってそれが不可能となってしまったので、VuforiaSDK のサンプルプログラムを基にアプリケーションの作成を行っていく。

また、本来は Swift でコーディングをする予定だったが、サンプルのプログラムが Objective-C で書かれていた為、Objective-C でコーディングをしていく。

画面の遷移は以下のようになる。

- ① 起動をするとロゴが表示される。

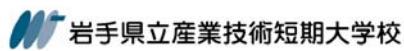


図 15 起動

- ② ここで行きたい科を選択する。



図 16 メニュー画面

- ③ 上にあるスタートボタンを押す。



図 17 スタートボタンを押す画面

- ④ カメラ画面に移る.  
 ⑤ マーカーを読み取ってモデルの表示.



図 18 モデルの表示

### 3.3.3 3D モデルの読み込み

本来は、「obj2opengl.pl」という obj 形式のファイルを h 形式のファイルに変換するスクリプトを実行して、頂点座標や法線座標を取得したものを利用する予定だったが、以下の画像のように何故か歯抜けで表示されてしまう。

そこで、Android 側で利用している座標データを使用したところうまくいったため、それを利用することにした。

### 3.3.4 今後の作業計画

テクスチャの作成・変更がまだできていないので、テクスチャの変更をすることが第一に挙げられる。

また、Metaio のサービスが終了していたことによる Vuforia への転換などにより作業に遅れがでたため、道案内機能の実装は今回は行わず、3D モデルの完全な表示に取り組む。

### 3.3.5 研究を通して

iOS のアプリ開発から学んだことよりも、3D モデルを表示する仕組みがどのようになっているかなど、3D モデルの知識について得たものが多かった。

Swift による iOS アプリの開発にも興味があるので、機会があればやってみたいと思う。

## 4. 参考文献

- 3D プリンタのためのデジタル造形術  
(株式会社ホビージャパン)
- Metasequoia – Wikipedia  
(<http://ja.wikipedia.org/wiki/Metasequoia>)
- Android Studio – Wikipedia  
([https://ja.wikipedia.org/wiki/Android\\_Studio](https://ja.wikipedia.org/wiki/Android_Studio))
- Xcode - 新機能 - Apple Developer  
(<https://developer.apple.com/jp/xcode/index.html>)



図 19 モデル表示の失敗