

# 目次

はじめに

## 1. マルチマーカ型 AR 制作

### 1-1. はじめに

<1>AR とは？

<2>AR の歴史

<3>AR の仕組み

### 1-2. 制作目的と目標

<1>目的

<2>目標

### 1-3. 制作準備

### 1-4. 制作過程

<1>3DCG モデル作成

<2>AR マーカー作成

<3>実行プログラム

### 1-5. 実行結果

### 1-6. 3D プリンタで印刷

### 1-7. HP 作成

### 1-8. まとめ

参考文献

## 2. 松山大学 PV 制作

### 2-1. 制作目的

### 2-2. 使用したビデオカメラ

### 2-3. 広報課へのアポイントメント

### 2-4. 教授と在学生へのアポイントメント

<1>授業風景撮影アポイントメント

<2>在学生インタビューアポイントメント

### 2-5. 撮影

<1>授業風景撮影

<2>インタビュー撮影

<3>施設撮影

<4>大変だったこと

## 2-6.編集作業

<1>編集手順

<2>大変だったこと

## 2-7.まとめ

参考文献

## 3.ゲームアプリ制作

### 3-1.はじめに

### 3-2.概要

### 3-3.動作説明およびプログラム説明

<1>データベース

<2>アラートダイアログ

<3>グラフィック描画

<4>効果音、BGM

<5>つづきから処理

### 3-4.プログラム紹介

<1>sample.sqlite1.java (メインアクティビティ)

<2>Plant.java (プラントクラス)

<3>TitleActivity.java (タイトル画面)

<4>music.java (BGM クラス)

<5>SePlayer.java (SE クラス)

<6>PersonOpenHelper.java (データベース)

<7>activity\_splite\_sample11.xml (メイン画面レイアウト)

<8>title1.xml (タイトル画面)

<9>string (オブジェクトの名称)

### 3-5.まとめ

参考文献

おわりに

# はじめに

2010年4月に松山大学に入学した経営学部生のうち、檀ゼミには17名のメンバーが6期生としてやって来た。そのうち7名はWindows Movie Makerによる映像編集などに取り組んだ「経営学部基礎演習」から継続してゼミに所属したメンバーである。

大きな地震を忘れることはできない 2011年度のゼミで最初に取り組んだのは、Adobe Flash Professional CS5 を用いたアニメーションによるコンピュータ・グラフィックス (CG) の制作である。みんなで選定したテキスト「速習デザイン Flash CS5」に沿って基本的な操作手順を習得し、いくつかのチームを編成してアニメーション作品の創作に取り組んだ。

学年の進行した 2012年度前期には、毎週月曜日の5時限と6時限にゼミの授業を行い、ActionScript 3.0 によるプログラミングを学んだ。その技術は、南海放送からの依頼を受けて実施する世論調査とデータ解析のために使われることを想定した。その中で、松山市内の電話番号を無作為抽出するシステムを実装したり、選挙結果を正しく予測するモンテカルロ法によるマルチエージェント・シミュレーションを実装したりした。これらは単なる授業の課題で終わることはなく、実際の世論調査で使用され、大きな成果を上げたものである。その後、2012年11月3日（文化の日）の経営学部ゼミナール大会に向けて4つのチームを編成し、テーマを確定させたあたりで、ゼミ担当教員はミュンヘン大学に旅立ってしまった！

その間、2つのチームが経営学部ゼミナール大会で優秀発表賞を受けたり、南海放送との共同世論調査を成功させたり、2名のメンバーが同時に経営学部長賞を受けたり、順調に就職活動ほか卒業後の進路を確定させたりと、ゼミ担当教員のいないときのほうが素晴らしい活躍をした頼もしいメンバーたちだった。

2013年9月からゼミの授業が再開し、半年後の卒業に向けて卒業制作に取り組むことが決まった。拡張現実 (AR) の技術を用いたアプリの開発、松山大学プロモーションビデオ (PV) の制作、Android 向けの植物育成系ゲームの開発といった主要テーマでチームを分け、プロジェクト活動に一生懸命取り組んだ。6号館2階の言語・情報研究センターを活動拠点として、ときには夜遅くまで頑張ったこともある。また、目の前に大きな壁が立ちはだかることもあっただろう。しかし、メンバー同士が励まし、協力し合いながらプロジェクトのゴールに向かって走っていった。この「卒業制作プロジェクト報告書」は、それらの成果をまとめた貴重な記録である。ぜひ、その様子をご覧いただきたい。

なお、檀ゼミとして報告書を刊行するのは今回が初めてのことである。

2014年3月1日 松山  
ゼミ担当教員 檀 裕也

3 回生までのゼミ活動では、個々で課題に取り組むことが多く、皆で何か一つのテーマを決め活動するといったことがなかった。そこで、最後に卒業制作として何か皆で作らばいいということになった。

最初は、皆で意見を出し合った中から「映像制作」と「アプリ制作」の 2 つのテーマに絞った。17 人全員で一つのテーマにしようとしていたが、役割分担などを考えると多すぎるということになり、映像 7 名、アプリ 10 名の 2 チームに分けることにした。

そしてアプリを作成するにあたり、どのようなアプリを作成するかという話し合いの中で 2 つの案が出たこと、プログラミングを複数人で同時進行することは困難という判断から「アプリ制作」を「AR 制作」と「ゲームアプリ制作」に分けることとなった。

2014 年 3 月 1 日 松山

AR 制作班

石村美樹 梶原利映 篠原健

政屋祐起 森田真由 渡部由依

PV 制作班

小川夏生 友石綾乃 田村香奈子 宮内隆成

矢葺瑛里加 山下真奈美 山脇明歩

ゲームアプリ制作班

岡本望希 檜垣旭宏 福森裕香 宮内英令奈



# 1. マルチマーカ型 AR 制作

## 1-1. はじめに

### <1> AR とは？

AR (Augmented Reality =拡張現実) とは、目の前に広がる現実の空間とそこから得られる様々な知覚情報に、デジタル情報を重ね、融合することで現実を拡張したり強化したりする技術のことである。これは現実世界とデジタル世界が重なった、実際には見えないものを可視化するこれまでにない空間を創り出すことができる。人間の視覚、聴覚、嗅覚、触覚、味覚などの 5 感を拡張するだけでなく、身体部位や身体の動きも拡張し、そこからコミュニケーションを増進させるツールである。

### <2> AR の歴史

- 1873 年 イギリスで明暗を電気の強弱に変えて遠方に伝えるテレビジョンの開発始まる。
- 1897 年 ドイツの **Karl Ferdinand Braun** によってブラウン管が発明される。
- 1901 年 拡張現実と同様のアイデアは、劇作家のライマン・フランク・ボームによって初めに述べられた。彼は、現実世界に創作されたデータを重ね合わせて表示する「キャラクター・マーカ」という電子デバイスを考案した。
- 1957-62 年 映像撮影技師の **Morton Heilig** はセンソラマというシミュレータを開発し特許を取得した。これは映像と音響と振動と香りを模擬するシステムであった。
- 1966 年 AR を実現するデバイスが実際に開発された事例として、アイバン・サザランド によるヘッドマウントディスプレイの発明などがなされてきた。
- 1975 年 **Myron Krueger** によって、ユーザーが仮想物体とやりとりできるビデオプレイスというシステムが初めて開発された。
- 1989 年 仮想現実という言葉が **Jaron Lanier** によって作られ、初めての仮想世界で行うビジネスを作り出した。  
「拡張現実」という名前は、1990 年にボーイングの技術者である **Tom Caudell** によって付けられた。
- 1992 年 アームストロング空軍研究所の **L.B. Rosenberg** は機能する AR システムとしては最初期のものである **Virtual Fixtures** を開発し、デモンストレーションを行った。

- 1992-93年 コロンビア大学で「KARMA」と呼ばれる AR を利用したシステムが開発された。これは、超音波センサを使ってヘッドマウントディスプレイにレーザープリンタの内部機構を表示し、プリンタの保守作業をサポートするシステムだ。
- 2000年 Bruce H. Thomas は初の屋外携帯 AR ゲームである「ARQuake」をウェアラブル・コンピュータ国際シンポジウムで発表しデモンストレーションを行った。  
このように軍事技術や自動車・航空機製造技術で主に利用されてきた AR だが、2000年代に入り携帯端末の普及に伴い一般消費者向けサービスへと利用されるようになった。
- 2007年以降 この頃から日本でも一般にも知られるようになった。同年に AR 技術を応用した初めてのゲーム「THE EYE OF JUDGMENT」が発売された。
- 2010年  
8月27日 日本の携帯電話のフィーチャーフォンで実装を実現した AR3DPlayer は、2次元バーコードの QR コードをそのまま、AR マーカーとして認識する物であった。またセカイカメラの様に GPS の位置情報を利用する物もある。テレビや映画や雑誌などといったエンターテイメント利用の事例も、AR 三兄弟などを中心に進んでいる。
- 2011年 詩人・アーティストの ni\_ka が AR 技術を用いた AR 詩を発表している。

### 〈3〉 AR の仕組み

AR は Web カメラや携帯端末に映し出された映像（リアル）に、コンピュータで作成されたアニメーション、3D モデル、画像、動画、文字情報などのデジタルを重ね合わせて表示するリアルを認識し、デジタル情報の効果を付加するという仕組みである。付加するものは視覚情報だけではなく、音などの聴覚情報も付加することができる。とくに最近ではグーグルグラスのように、簡単に装着して日常生活のなかに溶け込むような拡張体験を可能とするソリューションも提案されている。

また一口に AR といっても様々な制作手段があり

- ・ GPS と加速度センサーなどから位置情報を取得する、ロケーションベース AR
- ・ マーカー(QR コードのようなもの)から情報を取得する、マーカー型 AR
- ・ マーカーを使用せず物体や空間を認識から情報を取得する、マーカーレス型 AR

に分類され本制作ではすぐに利用できるライブラリが公開されていて、比較的容易に取り組みやすくとされるマーカー型 AR を用いて制作した。

## 1-2. 制作目的と目標

### <1>目的

- ①画面上でしか会えなかったキャラクターに会う
- ②AR を自分たちなりに咀嚼して楽しむ
- ③HP に公開することで檀ゼミを志望する後輩たちにゼミ活動ならびに AR を知ってもらう

### <2>目標

- ①オリジナルマーカ、3DCG モデルの作成
- ②プログラムを実行し web カメラでマーカを読み取りキャラクターの具現化

## 1-3. 制作準備

本制作で使用したソフトウェア及びハードウェアを表 1 に記す。Processing、Metasequoia ともにフリーソフトとしてダウンロード可能である。また、Processing には 3D グラフィックス処理を行うためのライブラリがデフォルトで入っているが、外部ライブラリを導入することでより機能的な AR を制作することができる。本制作で導入した外部ライブラリは表 2 に記す。

表 1

名称	用途
① Processing-1.5.1	プログラミング
② Metasequoia ver4.3	3DCG モデルの作成
③ Windows ペイント	オリジナルマーカの作成
④ Web カメラ	マーカの読み取り

表 2

名称(ライブラリ)	用途
① NyAR4psg	マーカベース AR 制作
② MQOloader	Processing 上で Metasequoia 形式データを表示

## 1-4. 制作過程

### <1> 3DCG モデル作成

キャラクターモデルは松山大学に関係のあるキャラクターがよいという意見があり、松山大学生協のキャラクターである「ばいんこ」(図1)が候補に挙げられた。松山大学生協専務理事の神田斉宜さんとの交渉の結果、「ばいんこ」をモデルキャラクターとしてARを制作する使用許可を頂いた。その際、「ばいんこ」の原案者である松山大学OGの矢野裕子さんから3DCGモデル作成の監修をやらせてほしいという申し出があり、メールを通じて監修をして頂くこととなった。こちらが作成したモデルのスクリーンショットを送り、それに修正ポイントを返信してもらう形で監修を受けた。また矢野さんとのメールのやり取りは神田さんに仲介してもらい、神田さんからも助言を頂いた。

初期案



図1



### 1回目の返信

お疲れ様です。さっそく制作者からのアドバイスが来ました。

せっくなので、ゆるーい感じそのまま掲載しますね。

3) だけ神田からも意見を追加させてください。

とさかの奥行きが顔と同じなので狭めること、若干後ろに倒すこと、の2つ指摘があります。

私からは、もう少し柔らかさが出るといいな〜と。

例えば上に行くほど先がとがるとか、後ろへの倒し方も少し弧を描くとか・・・。

あまりに角ばったまっすぐモヒカンなので(笑)

\*\*\*\*\*以下、制作者からのメールです\*\*\*\*\*

お世話になります〜〜〜

画像見ました!!

おお〜〜〜! 立体になってる!!



楽しそうやなあ・・・

気になったところを何点か～！

- 1) 足ですが、前に3本、うしろがわに小さい支えが一本、  
って感じです。(添付画像参照)

略してるのでありませんが基本的にはこういう立ち方  
だと思うので立体化でいろいろ追求するにはあったほ  
うがいいかも・・・？

とはいえ、あくまで架空のかんじなので前3本だけで  
もいいような気がしています。

いま、まんなか折れてぐいっとうしろにいつてるような感じで  
イタそうなので修正できればと思います！



- 2) しっぽですが上から長い～中くらい～短いつて感じでおねがいます。

(いまは全部同じ長さなので)

- 3) とさかですが太さ、もうちょい細くできたらと思います。

顔の奥行きの3/4くらいかな・・・

で、できれば若干、ほんとに若干(2度くらい)

奥に倒しておいてもらえるといいかと思います。

- 4) あと、これ制作途中で

下面の干渉がないから、だと思っのですが

シャドウが黒いのを、若干ライトを弱めるか

シャドウ自体の色を少しオレンジ系にするほうがいいかもです。

このソフトは使ったことがないので、できるかどうかはわからないですけど・・・

(あまりギャツと陰が入ると、ちょっと古い印象を受けるので・・・)

- 5) これもソフトの具合がなんともなんです

表面が現在つるりんとしてるので

若干毛並みといますか

ほわっとした感じといますか

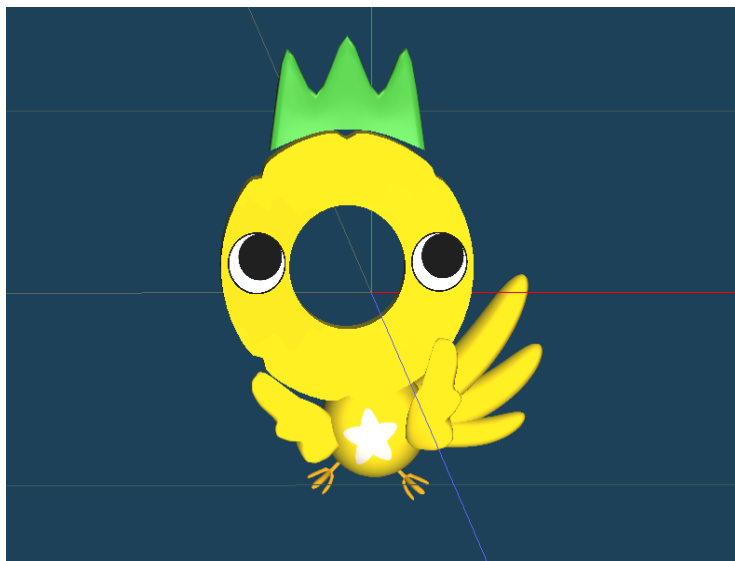
なんかできたらええなあと思いました。

身体だけ・・・

(さすがにこういうのは無理かな？ハードルあげてますねすみません)

以上です～～～！！

## 修正1回目



### ☆修正ポイント

- ・足の中指の角度修正
- ・尾の長さ、角度修正
- ・とさかの大きさ、角度修正
- ・複数からライトをあてシャドウが黒いのを改善



## 2回目の返信

制作者からの連絡がありました。王冠にしよう！との提案でした。  
確かに横から見てもパイナップルのヘタ風にみえるのでいいですね。

・・・キャラクター設定の謎の一部が決まった瞬間です（笑）

修正は大変かと思いますが、よろしくお願いします。

\*\*\*\*\*以下、制作者からのメールです\*\*\*\*\*

どもです！！

足とかそのあたりはこれで大丈夫かとおもいます！

とさかですが、ちょっとでかいかな???

平面のデータのみをみていただくとわかるかと思いますが

実際は現状見せていただいている3Dデータの半分くらいのサイズかとおもいます。

なかなかのつけるのは難しいので

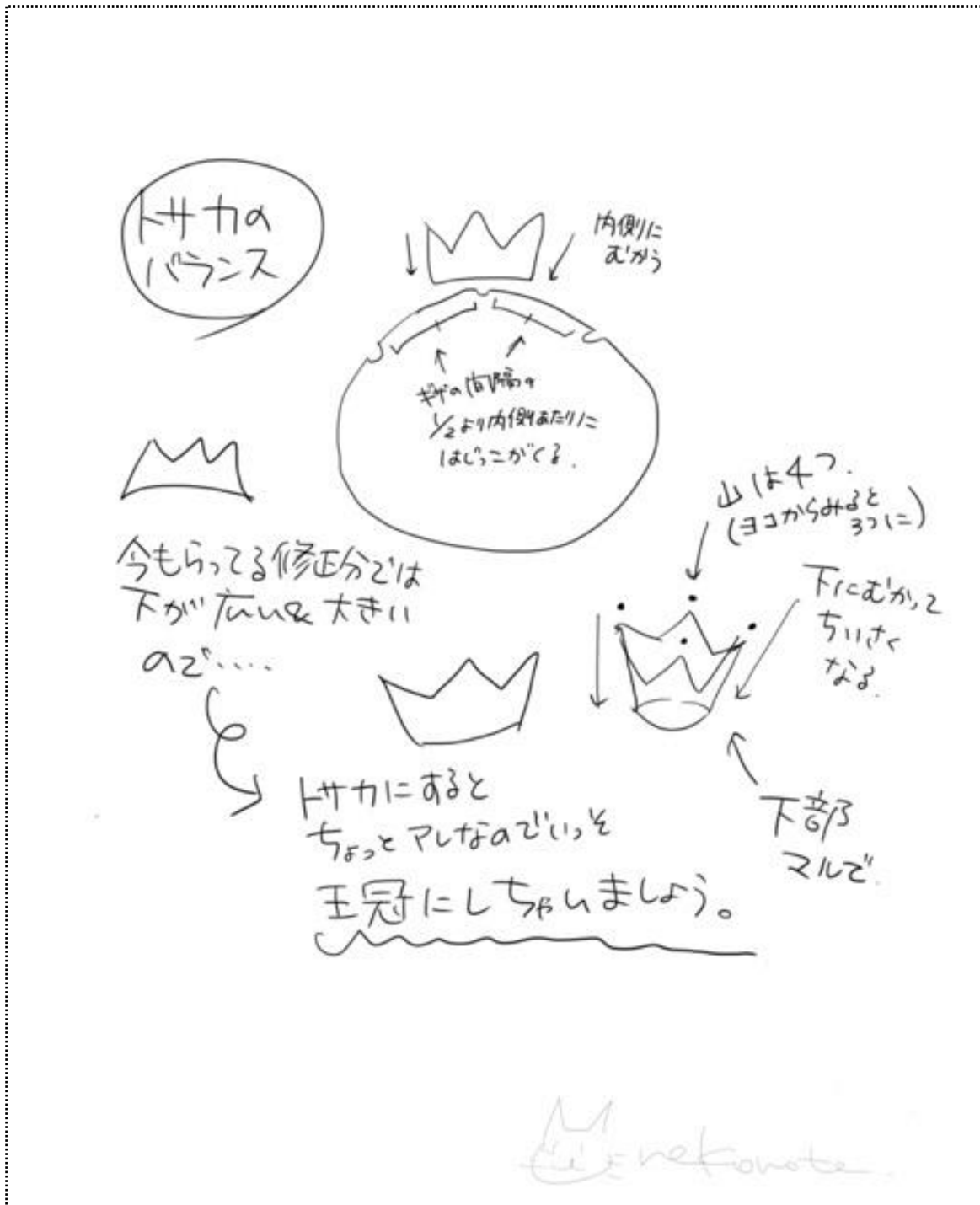
いっそのとさか、王冠にしちゃいましょう。

ちょこっと王冠がのってるような感じ・・・（添付画像参照）※1

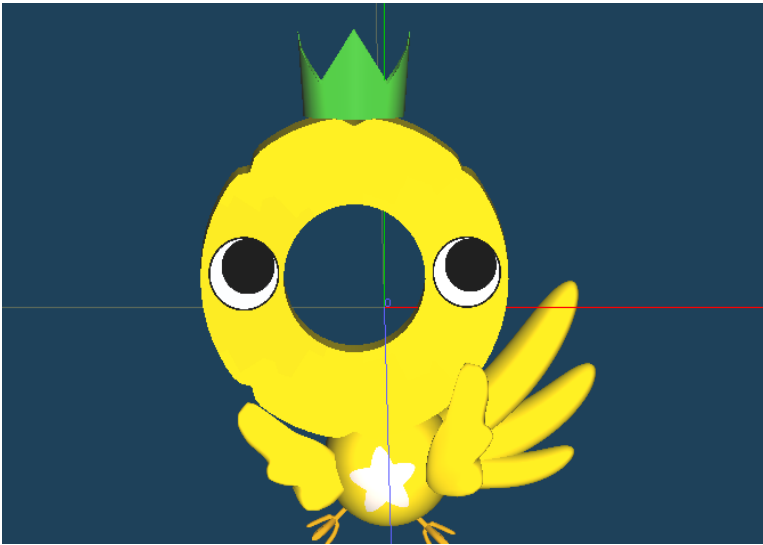
でどうでしょう???

ご確認くださいまし・・・！

※1 添付画像

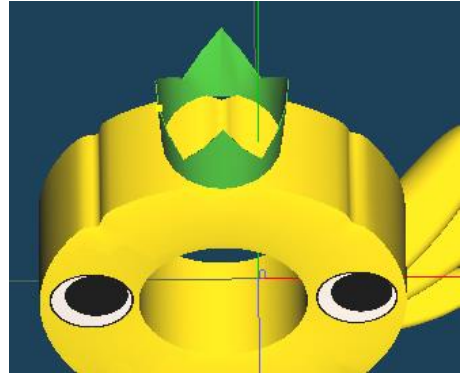


## 修正 2 回目



### ☆修正ポイント

- ・とさかを王冠に修正



## 3 回目の返信

お疲れ様でした。

おおむね OK のようです。

ただもし可能なら、王冠のてっぺんの「とんがり」がきつすぎていたそうなので、きもちだけカドを丸めにしてほしいのと、王冠と頭の間、これもほんの気持ち隙間をつくることをしてもらえますか。

「ただし王冠を浮かせずすぎると、影がごりっと出そうなので、微妙な感じに・・・」と難しい注文かもしれません。

これで完成ですね。

何とか年内におさまるかな？

神田

### 修正 3 回目



#### ☆修正ポイント

- ・王冠の頂点を削り微修正
- ・王冠を上部に移動



### 4 回目の返信

こんにちは。

先日の最終校正を作者に見せました。

以下のメッセージが届きました。

本当にお疲れ様でした。

よい作品が出来上がることを期待しています。

どんな動きをするんだろう？

ワクワクして待っています。

\*\*\*\*\*以下、制作者からのメールです\*\*\*\*\*

お世話になります！

最終みました～！

つついプロ的な目でみてしまうと

いろいろ気になるところもでてきますが

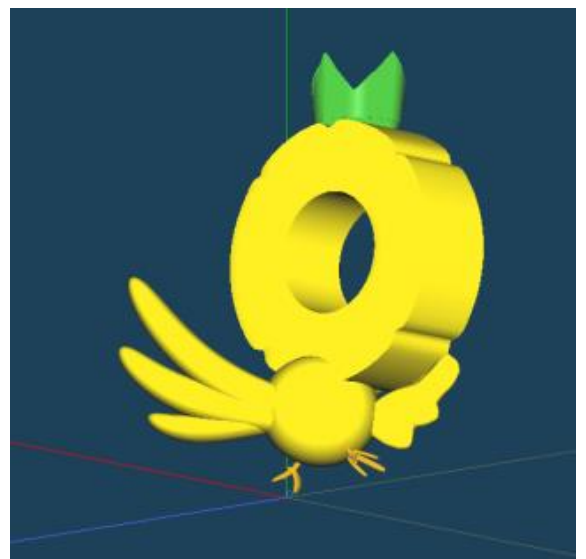
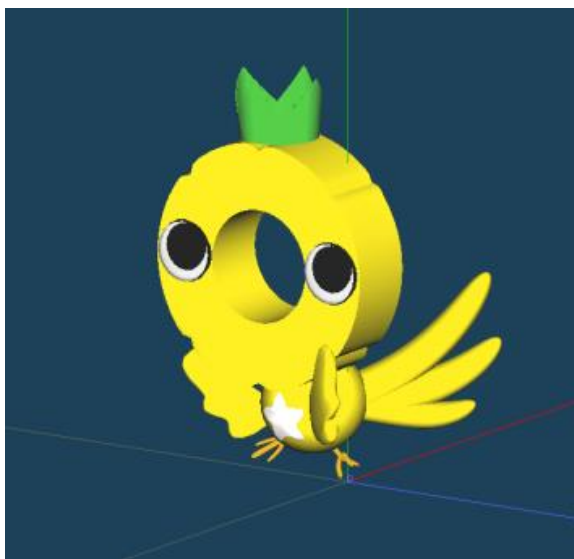
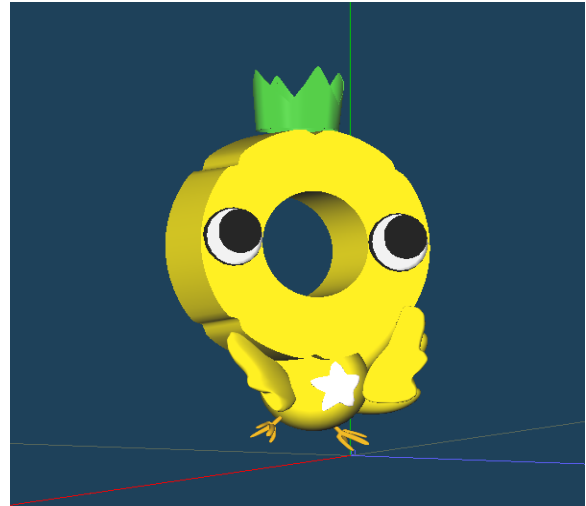
使用ソフトの限界もあるなか

とてもいい感じにできているのではと思います！

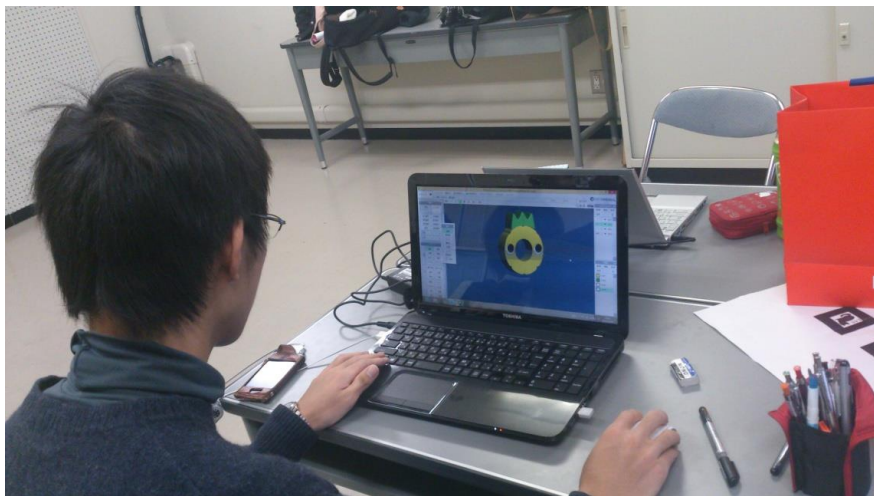
お疲れさまですとお伝えください～～～！

(^)

## 完成形



## 制作の様子



## <2> AR マーカー作成

AR マーカーとは画像認識型 AR (Augmented Reality : 拡張現実) システムにおいて、付加情報を表示する位置を指定するための標識となる決まったパターン of 画像のことである。なおマーカーはペイントソフト Gimp を用いて作成した(図 2, 3, 4)

図 2

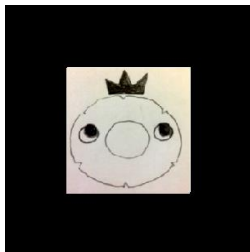


図 3



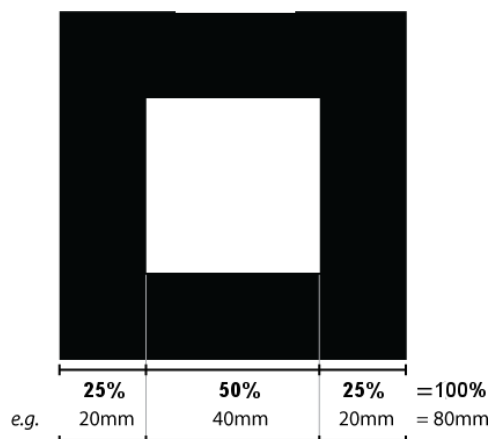
図 4



作成時の注意点として以下の点があげられる

- ・正方形であること
- ・基本は黒枠、内部の白領域の割合が、1:2:1 (図 5 参照)
- ・黒枠でマーカーの検出を行い、白領域内のパターンに応じてマーカーを判別
- ・白領域内のパターンは、点対称・線対称を避け、デザイン部分に細い線を用いないこと

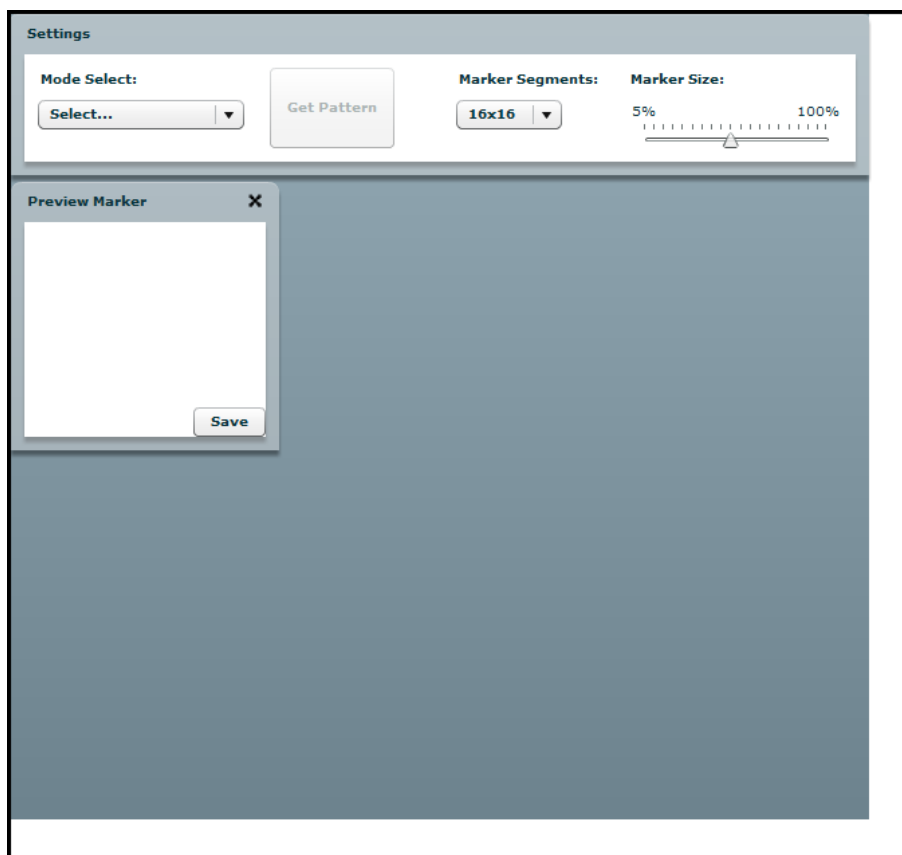
図 5



実際には使用状況に応じて適切に作成することが必要である。

比較的自由度が高いように思えるが、仕様環境に応じて制約がついてまわる。マーカーを決める場合は、3~5 パターン程度作成した上で、サンプルを用いて仕様想定環境下で試していくのを推奨する。

なお、作成した AR マーカーをプログラム上で認識させるには AR Toolkit Marker Generator Online【<http://flash.tarotaro.org/ar/MarkerGeneratorOnline.html>】を使用し登録する必要がある。



## ARToolKit Marker Generator Online 使い方

Flash Player10 が必要です。

1. マーカをデザインし、印刷してください。
2. Marker Segments コンボボックスから、マーカの分割数を選択してください。  
デフォルトは 16x16 です。
3. マーカサイズ設定スライダーを操作し、マーカの大きさを選択してください。  
マーカを大きくすると、枠が小さくなります。  
デフォルトは 50%で、これ以外だとプログラムに工夫が必要です。
4. Web カメラにマーカを映してください。
5. Get Pattern ボタンが利用可能になったら、クリックしてパターンのプレビューを見ることが出来ます。
6. プレビューウィンドウの右下にある Save ボタンを押すと、パターンを保存できます。
7. Mode Select コンボボックスから「Load marker Image」を選択すると、  
画像を読み込んでマーカを探索できます。  
マーカを画像として読み込み、パターン化することも可能です。
8. 作成したパターンは、他の ARToolKit 系列ライブラリで使用できます。  
ただし、マーカサイズを変更した場合、ライブラリを変更する必要があります。  
詳細は[こちら](#)をご覧ください。



認識が上手くいけばこのように模様が Preview Marker に表示される。この模様を web カメラが読み込み認識するのである。



### <3>実行プログラム

AR を実行するプログラムの作成にあたっては、Processing を使用した。

Processing の特徴として

- ・オープンソースソフトウェアなので無償で使用できる。
- ・短いソースコードで動いてくれる
- ・外部ライブラリが豊富にある

などが挙げられ、初心者でも扱いやすいソフトとなっている。

#### ▪Processing のインストール

Processing の公式サイト【<http://processing.org/>】からダウンロードする。



The screenshot shows the Processing 2 website. At the top, there is a navigation menu with links for Cover, Download, Exhibition, Reference, Libraries, Tools, Environment, Tutorials, Examples, Books, Overview, People, and Foundation. The main content area features a large 'P' logo and the text 'Download Processing. Processing is available for Linux, Mac OS X, and Windows. Select your choice to download the software below.' Below this, there are three columns of download links: 'Windows 64-bit' (highlighted with a red box), 'Linux 64-bit', and 'Mac OS X'. Underneath these, there are links for 'Windows 32-bit' and 'Linux 32-bit'. The version number '2.1 (27 October 2013)' is displayed. At the bottom, there are links for 'Github', 'Report Bugs', 'Wiki', and 'Supported Platforms', along with a note about the 'list of revisions'.

当初は OS にあった最新版の Processing-2.1-Windows 64-bit 版をダウンロードした。

しかし、最新版だと WEB カメラとの互換性の問題でうまく作動しなかったため、バージョンを落とした processing-1.51-Windows64-bit 版をダウンロードした。このように互換性の問題で最新版のソフトウェアが起動しないことがたびたび起こる。この問題を解決するためには自分の作業環境に見合ったバージョンを使っていくことが望ましい。

## ・ソースコード

キャラクターモデルを AR 化してマーカー上に出現させるためのプログラムを以下に記す。

```
import javax.media.opengl.*;
// Java の OpenGL (JOGL) の機能を使うために必要
import processing.opengl.*;
// Processing の OpenGL の機能を使うために必要
import jp.nyatla.nyar4psg.*;
// NyAR4psg を使うために必要
import processing.video.*;
// ビデオ入力を扱うので必要
import jp.nyatla.kGLModel.contentprovider.*;
// MQO ローダーを使うために必要

Capture cam;
// キャプチャ
MultiMarker ar;
// AR マーカーに関する処理をするオブジェクト
int id1,id2,id3;
// マーカーに割り当てられる ID 番号

KGLModelData model_data;
ContentProvider content_provider,content_provider1,content_provider2;

void setup() {

    size(640,480,OPENGL);
    cam = new Capture(this,width,height);
    // キャプチャの準備
    ar = new MultiMarker(this, width, height, "camera_para.dat",NyAR4PsgConfig.CONFIG_PSG);
    id1 = ar.addARMarker("cat.pat", 90);
    // パターンファイルとマーカーサイズの登録
    id2 = ar.addARMarker("danzemi.pat", 90);
    // パターンファイルとマーカーサイズの登録
    id3 = ar.addARMarker("pine.pat", 90);
    // パターンファイルとマーカーサイズの登録
```

```

// MQO モデルの準備
content_provider = new LocalContentProvider(this,dataPath("pineko.mqo"));
// ソースコードと同じ場所にある「data」フォルダの中にあるファイルの読み込み
PGraphicsOpenGL pgl = (PGraphicsOpenGL) g;
GL gl = pgl.beginGL();
model_data = KGLModelData.createGLModelPs(this, gl, null,this.content_provider,0.1,
      KGLExtensionCheck.IsExtensionSupported( gl,"GL_ARB_vertex_buffer_object"),
      true );
pgl.endGL();
}

```

```

void draw() {
  if(cam.available()==false) return;
  cam.read();
  // カメラ画像の読み込み
  background(0);
  // 画像の初期化
  ar.drawBackground(cam);
  // 背景画像の描画
  ar.detect(cam);
  // マーカー認識

```

```

// 猫のマーカーが検出されたら

```

```

  if ( ar.isExistMarker(id1) ){
    ar.beginTransform(id1);
    PGraphicsOpenGL pgl = (PGraphicsOpenGL) g;
    GL gl = pgl.beginGL();

```

```

// テクスチャとポリゴンの描画に関する設定

```

```

  gl.glTexEnvf(GL.GL_TEXTURE_ENV, GL.GL_TEXTURE_ENV_MODE, GL.GL_MODULATE);
  gl.glEnable(GL.GL_CULL_FACE);
  gl.glCullFace(GL.GL_FRONT);

```

```

// 光源の設定
mySetLight(gl, 0,-100,100);
gl.glRotatef(90,1,180,0);
// 立たせるために回転
gl.glRotatef(90,0,1,0);
// 正面を向かせるために回転
model_data.draw();
pgl.endGL();
ar.endTransform();
}

// パインコのマーカーが検出されたら
if ( ar.isExistMarker(id2) ){
ar.beginTransform(id2);
PGraphicsOpenGL pgl = (PGraphicsOpenGL) g;
GL gl = pgl.beginGL();

// テクスチャとポリゴンの描画に関する設定
gl.glTexEnvf(GL.GL_TEXTURE_ENV, GL.GL_TEXTURE_ENV_MODE, GL.GL_MODULATE);
gl.glEnable(GL.GL_CULL_FACE);
gl.glCullFace(GL.GL_FRONT);

// 光源の設定
mySetLight(gl, 0,-100,100);
gl.glRotatef(0,1,180,0);
// 立たせるために回転
gl.glRotatef(90,0,1,0);
// 正面を向かせるために回転
model_data.draw();
pgl.endGL();
ar.endTransform();
}

```

```

// danzemi のマーカーが検出されたら
if ( ar.isExistMarker(id3) ){
    ar.beginTransform(id3);
    PGraphicsOpenGL pgl = (PGraphicsOpenGL) g;
    GL gl = pgl.beginGL();

// テクスチャとポリゴンの描画に関する設定
    gl.glTexEnvf(GL.GL_TEXTURE_ENV, GL.GL_TEXTURE_ENV_MODE, GL.GL_MODULATE);
    gl.glEnable(GL.GL_CULL_FACE);
    gl.glCullFace(GL.GL_FRONT);

// 光源の設定
    mySetLight(gl, 0,-100,100);
    gl.glRotatef(90,1,180,0);
// 立たせるために回転
    gl.glRotatef(90,0,1,0);
// 正面を向かせるために回転
    model_data.draw();
    pgl.endGL();
ar.endTransform();
}
}

// 光源の設定を行う関数
void mySetLight(GL gl, float x, float y, float z){
    float[] light_diffuse = { 1.7, 1.5, 1.5, 1.5 };
// 拡散反射光
    float[] light_specular = { 1.5, 1.2, 1.2, 0.0 };
// 鏡面反射光
    float[] light_ambient = { 1.2, 1.0, 1.0, 1.0 };
// 環境光
    float[] light_position = { x, y, z, 0.0 };

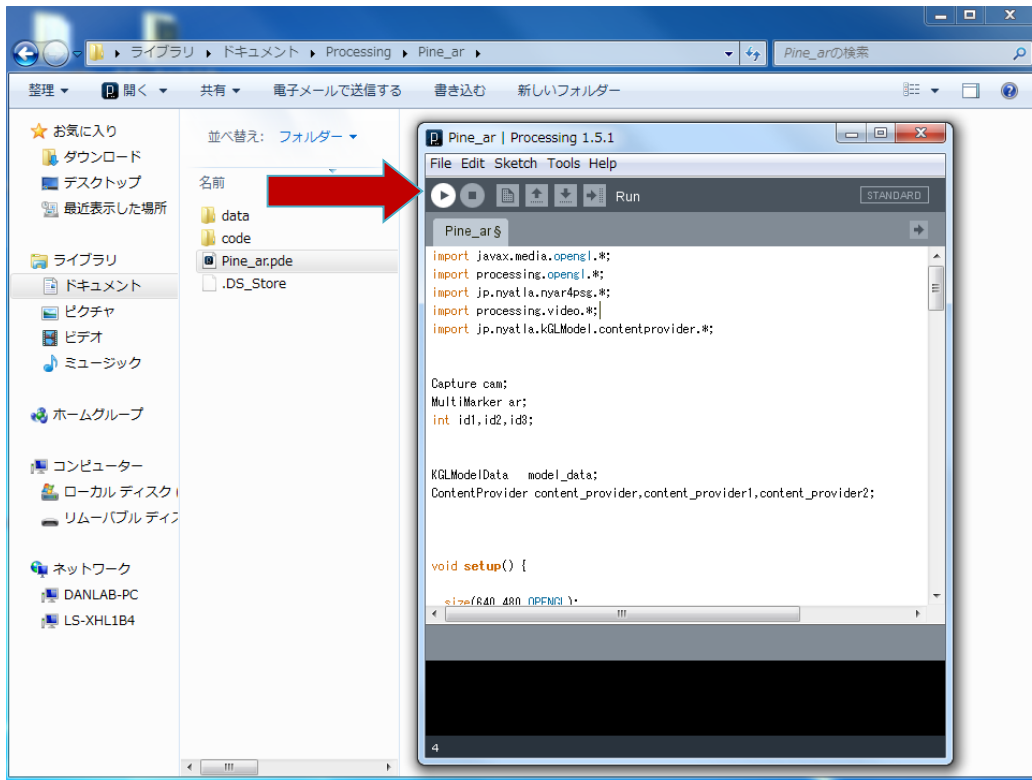
```

```
// 位置と種類
// 光源のパラメータの設定
gl.glLightfv( gl.GL_LIGHT0, gl.GL_DIFFUSE, light_diffuse, 0);
// 拡散反射光
gl.glLightfv( gl.GL_LIGHT0, gl.GL_SPECULAR, light_specular, 0);
// 鏡面反射光
gl.glLightfv( gl.GL_LIGHT0, gl.GL_AMBIENT, light_ambient, 0);
// 環境光
gl.glLightfv( gl.GL_LIGHT0, gl.GL_POSITION, light_position, 0);
// 位置と書類
gl.glShadeModel( gl.GL_SMOOTH );
// シェーディングの種類の設定
gl.glEnable( gl.GL_LIGHT0 );
// 光源 0 の有効化
gl.glEnable( gl.GL_LIGHTING );
// ライティングの有効化
}
```

## 1-5. 実行結果

### ・実行手順

- ① Processing を起動
- ② プログラムを実行

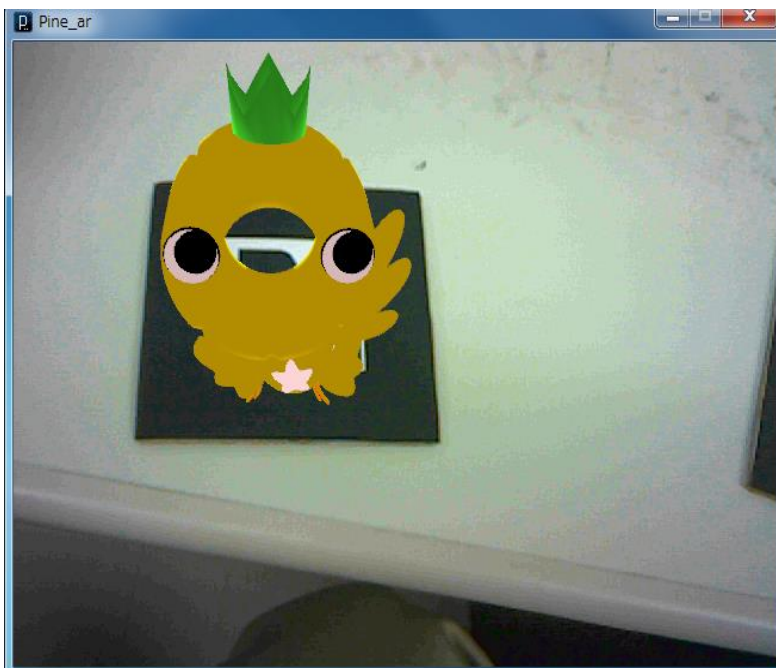




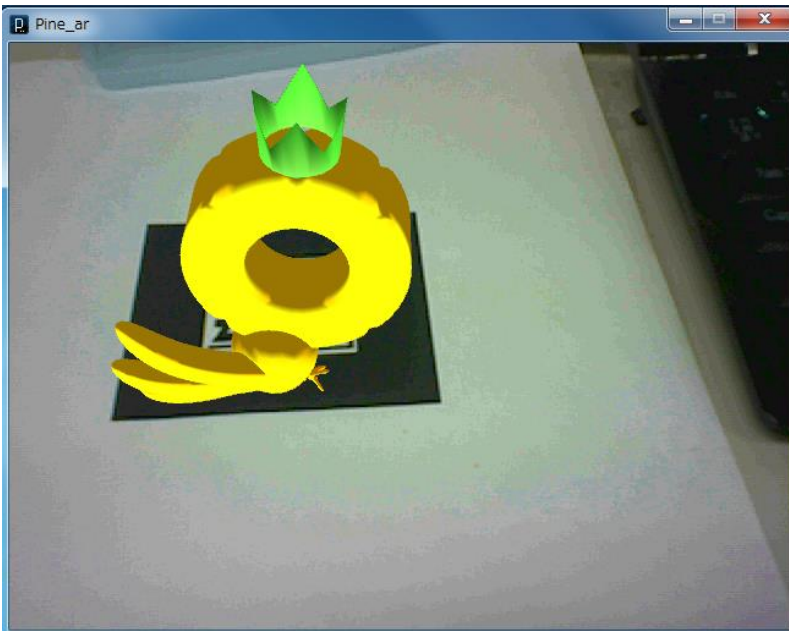
③ Web カメラでマーカーを撮影



モデル前面の様子



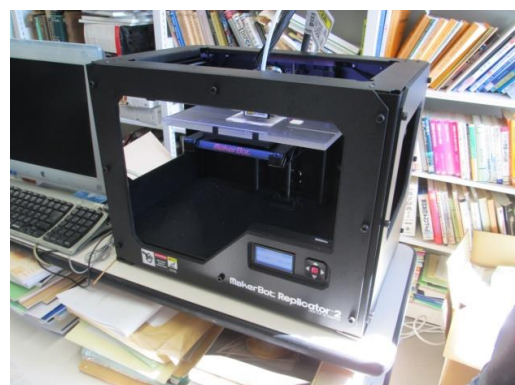
## モデル後面の様子



3DCG モデルに AR のモデルがどこまで近づけるか注目が集まったが、細部まで忠実に表現されていた。Web カメラはしっかりと正面から撮影しないとマーカーを認識することができず、前述の ARToolKit Marker Generator Online の Preview Marker に表示された形状を忠実に認識することが分かった。また撮影位置によって若干モデルの大きさも変化することも分かった。

## 1-6. 3D プリンタで印刷

3DCG モデルを AR 以外にも活用できないかということで、3D プリンタでモデルを印刷したらどうかという意見があがった。3D プリンタとは、アディティブ・マニュファクチャリング（Additive Manufacturing : AM）技術の略称で、3次元データを入力するとプラスチック、樹脂、金属粉などの材料を一層ごとに連続的に積層して立体物を造形する技術のことを指し、これまでのものづくりに革命をもたらすものとして昨今、世界で注目されている。そこで檀先生に 3D プリンタを所持されている愛媛大学教育学部の平田浩一先生を紹介してもらい、平田先生のご厚意で「ばいんこ」を印刷してもらうこととなった。

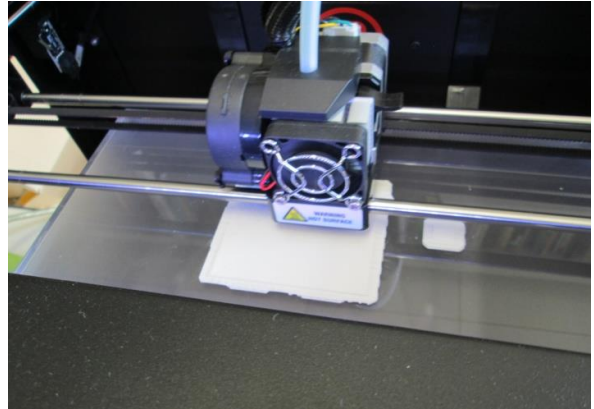


3D プリンタで印刷するには、現状の保存データの mqo 形式のファイルは対応しておらず、stl 形式のファイルにする必要があった。しかし Metasequoia は様々な形式で保存できるので今回求められた保存形式にも対応することができた。

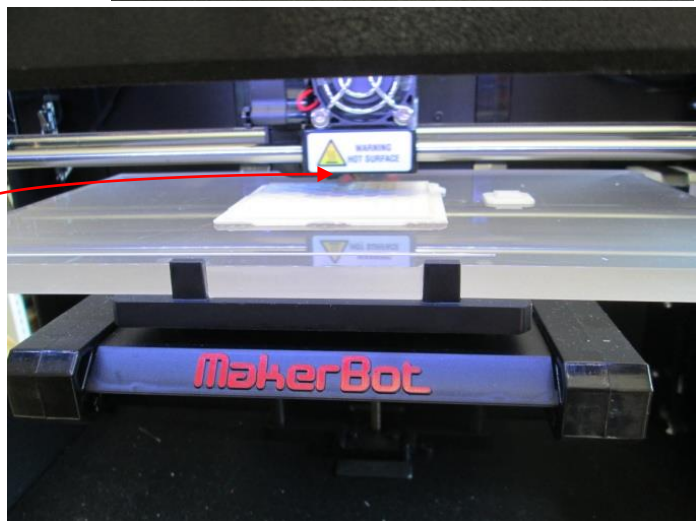
## 印刷物の材料となる ABS 樹脂



## 印刷の様子



このノズルより ABS 樹脂  
が 1mm 程度の糸状で出る



印刷は約 6 時間。ばいんこの各部につららのような柱が付いているが、これは下の土台から上に向けて積層を重ねるように印刷する都合上、本体を支えるために発生したものである。そして stl ファイルを読み込む際に、コンピュータがモデルの形、大きさを判断し自動的に生成されるようになっている。今回のモデルであれば仰向けに角度を変えて印刷することで柱を少なくすることができただろう。



3D プリンタが現在主に活用されているのは食品模形、学校の図形教材などで、まだまだその技術を生かした活用はあるように思われる。今回、実際に 3D プリンタの仕組み、印刷の様子を目の当たりにして、これまでの画一的で大量生産ありきのもづくりから、自分でオーダーメイドできる新しいものづくりに変わっていくための画期的な製品になると感じた。今後は人口骨、人工臓器など医療の分野でもその活用が期待されるのではないだろうか。これからの 3D プリンタの動向に要注目だ。



印刷直後のぱいんこ



柱を取り除き着色後のぱいんこ



## 1-7. HP作成

まず、全体的にどのようなデザインのホームページにするかを考える。誰が見ても分かりやすいホームページにするために、デザインはシンプルにして、松山大学のイメージカラーである青に統一した。また、トップページには、何のホームページか分かるように、ゼミの紹介文とゼミ活動の様子写真を掲載した。なお、デザインのテンプレートはこちらのサイトを利用した。<http://wdt.pekori.jp>



### 檀ゼミ6期生のホームページ

**TOP**

檀ゼミ生のホームページへようこそ！

#### 檀ゼミとは？

檀ゼミは、2006年度に松山大学経営学部においてスタートしました。現在、17名の学生が第5期生として檀ゼミに所属し、Webサイトの構築と運用にかかわる研究を進めています。2回生のゼミ・グループは、「演習第一」を履修します。この授業では、コンピュータを利用した専門的な情報処理の知識とスキルについて学びます。私たちは、FlashCS5というソフトを用いてアニメーション作製や簡単なゲーム作成を学びました。3回生のゼミ・グループでは、「演習第二」の授業でプログラミング言語として Java を学ぶとともに、選挙予測のプログラミングやゲーム開発を学びました。そして、4回生になると、各自の研究テーマに沿って、大学4年間における研究の成果を卒業論文としてまとめます。また、希望者には、基本情報技術者試験および応用情報技術者などの情報処理技術者試験、CGクリエイター検定、CGエンジニア検定およびWebデザイナー検定などのCG系検定の取得に向けて支援してくれます。ゼミ生同士の仲も良く、意欲があればなんでもやりたいことに挑戦できるゼミです。



## ・ソースコード

トップページのソースコードを以下に記す。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
<html lang="ja">
<head>
<meta http-equiv="content-type" content="text/html; charset=Shift_JIS">
<meta http-equiv="content-style-type" content="text/css">
<title>檀ゼミ</title>
<link rel="stylesheet" type="text/css" href="csstemplate/style.css">
</head>
<body id="INDEX">
<div id="PAGETOP">

<!--=====-->
<!-- ヘッダー -->
<!--=====-->
<div id="HEADER">
<h1>檀ゼミ 6 期生のホームページ</h1>
<ul id="PAN">
  <li>index</li>
</ul>
</div>
<hr>

<!--=====-->
<!-- メニュー -->
<!--=====-->
<div id="MENU">
<h2>menu</h2>
<ul>
  <li id="MENU01" class="menu-on"><a href="index.html">TOP</a></li>
  <li id="MENU02"><a href="AR.html">AR 班</a></li>
  <li id="MENU03"><a href="game.html">ゲームアプリ班</a></li>
  <li id="MENU04"><a href="pv.html">PV 班</a></li>
</ul>
</div>
```

```
<hr>

<!--=====-->
<!-- 記事部分 -->
<!--=====-->
<div id="KIZI">

<h2>TOP</h2>
<div class="text">
<p>檀ゼミ生のホームページへようこそ！</p>
<h3>檀ゼミとは？</h3>
<p>檀ゼミは、2006 年度に松山大学経営学部においてスタートしました。現在、17 名の学生が第 5 期生として檀ゼミに所属し、Webサイトの構築と運用にかかわる研究を進めています。

2 回生のゼミ・グループは、「演習第一」を履修します。この授業では、コンピュータを利用した専門的な情報処理の知識とスキルについて学びます。私たちは、FlashCS5 というソフトを用いてアニメーション作製や簡単なゲーム作成を学びました。
3 回生のゼミ・グループでは、「演習第二」の授業でプログラミング言語として Java を学ぶとともに、選挙予測のプログラミングやゲーム開発を学びました。
そして、4 回生になると、各自の研究テーマに沿って、大学 4 年間における研究の成果を卒業論文としてまとめます。
また、希望者には、基本情報技術者試験および応用情報技術者などの情報処理技術者試験、CGクリエイター検定、CGエンジニア検定およびWebデザイナー検定などのCG系検定の取得に向けて支援してくれます。
ゼミ生同士の仲も良く、意欲があればなんでもやりたいことに挑戦できるゼミです。

<p>

</p>
<ul class="modori">
  <li><a href="#PAGETOP">TOP</a></li>
</ul>
</div>
<hr>
```

```

<h2>履歴</h2>
<div class="text">
  <dl>
    <dt>2014年01月06日</dt>
    <dd>ホームページ公開開始</dd>
  </dl>
  <ul class="modori">
    <li><a href="#PAGETOP">TOP</a></li>
  </ul>
</div>
<hr>

</div>

<!--=====-->
<!-- フッター -->
<!--=====-->
<div id="FOOTER">
  <ul>
    <li id="FOOTER01"><a href="#PAGETOP">TOP</a></li>
    <li id="FOOTER02"><a href="http://www.cc.matsuyama-u.ac.jp/~dan/" TARGET="_blank">檀研
研究室</a></li>
    <li id="FOOTER03"><a href="http://www.matsuyama-u.ac.jp/" TARGET="_blank">松山大学
HP</a></li>
  </ul>
</div>

</div>

</body>
</html>

```



## 1-8. まとめ

今回、グループで卒業制作を行うにあたってなかなかテーマが決まらなかったが、檀先生の「ARをやってみたら？面白いと思うのだけど」という一言がきっかけとなり AR の制作を決意した。AR とは何かそしてどのような技術なのかという基本的な部分すらままならない状況から制作が始まったが、檀先生の丁寧な指導、さまざまな方の協力のおかげで無事制作を終えることができた。今回はマーカー型 AR でモデルを表示したのみに終わったが、地図と連携した観光案内マップや Miku Miku Dance を使用したオリジナル PV など AR は様々な場面で活用が期待できる。AR は広告やゲームで使用され身近な技術になりつつある。そしてその技術の特殊性、話題性から利用、研究が更に進んでいく分野と思われる。今回実際に AR に触れ、制作をすることで関心を持つことができたので今後も注目していきたい。

## 参考文献

「AR プログラミング Processing でつくる拡張現実感のレシピ」 オーム社:橋本直

「3D キャラクターが現実世界に誕生! ARToolKit 拡張現実感プログラミング入門」

アスキー・メディアワークス:橋本直

<http://processing.org/> 【最終閲覧日 (12月22日)】

<http://flash.tarotaro.org/ar/MarkerGeneratorOnline.html> (12月18日)

<http://kougaku-navi.net/ARToolKit/> (12月18日)

<http://hp.knowledge-works.co.jp/services/ar/fusion/> (1月17日)

<http://homepage1.nifty.com/samito/TVhistory.htm> (1月23日)

<http://www.tdk.co.jp/techmag/knowledge/200909u/index.htm> (1月23日)

<http://wdt.pekori.jp> (1月27日)

## 謝辞

制作を進めるにあたって、キャラクターモデルの修正、監修に協力して下さった「ばいんこ」原案者の矢野裕子さん、大学生協の神田斉宜さん、そして 3D プリンタを貸与して下さった愛媛大学の平田浩一先生に感謝します。

## 2. 松山大学PV制作

### 2-1. 制作目的

檀ゼミでは、PC関連のスキルを身に付ける活動に取り組んでおり、その活動の一環としてPCでの動画編集作業を行った。その活動で得たスキルを何かに活かしたいと考え、卒業制作として松山大学プロモーションビデオ（以下PV）を作成することとなった。このPVを作成し、インターネット上に流すことにより、これを見た高校生が少しでも松山大学に関心を持ち、私たちの後輩が増えてくれることを期待している。

### 2-2. 使用したビデオカメラ

SONY CX430V

選んだ理由

- ・手ぶれ補正機能の搭載
- ・使用目的に十分な光学ズーム機能
- ・高感度マイク装備
- ・同価格帯で使いやすい



	CX430V	GZ-MG40-S
画素数	543 万画素	133 万画素
内臓メモリ	32GB	20GB
質量	約 320g	約 320g
ズーム	光学 30 倍 (デジタル 350 倍)	光学 15 倍 (デジタル 200 倍)
撮影可能時間	約 2 時間 15 分	約 4 時間 50 分

GZ-MG40-S と比較して、上記の点で優れていた為、この機種に決定した。

### 2-3. 広報課へのアポイントメント

- ① 広報課へアポイントのメール
- ② 学内の撮影許可を取るため、大学広報課へ行く
- ③ 詳細を説明し、撮影許可取得
- ④ 年間行事とサークル活動等の写真をお借りする（表 1）
- ⑤ ロゴマークを追加でお借りする（図 1～3 参照）

写真合計 94 枚、61.7 メガバイト

オープンキャンパス	入学式
新歓祭	卒業式
新入生サポート	ゼミナール表彰式

熱田津祭	経営学部長賞
四国インカレ	授業風景
文化サークル	就活関連

表 1



図 1



図 2



図 3

・広報課へのアポイントのメール内容

松山大学広報課 ご担当者様

松山大学 経営学部 4年次生  
檀ゼミPV班班長の宮内隆成と申します。  
いつもお世話になっております。

先日はPV撮影の許可をいただきましてありがとうございました。

さて、PV制作の件なのですが、年間行事の写真や動画の素材が少ないため  
年間行事などの写真や動画などのデータをお借りしたいと思っています。  
肖像権や持ち出しに関する許可があると思いますので、  
貸出について確認をたくメール致しました。

お忙しいところ恐れ入りますが、  
許可についてご検討のほど  
どうぞよろしくお願いたします。  
松山大学 経営学部 経営学科  
宮内隆成

図 4

松山大学広報課 ご担当者様

松山大学 経営学部 4年次生  
樽ゼミ宮内隆成です。

本日は写真貸出しの許可を頂きましてありがとうございました。

早速ですが、写真の要望リストは以下になります。  
貸し出して頂ける項目だけでも十分なので、ぜひ宜しくお願い致します。

- ・入学式
- ・新入生全員集合（集合写真、グループ毎）
- ・新歓祭、スポ祭
- ・オープンキャンパス（高校生と在校生が写っている風景）
- ・インターカレッジ
- ・学祭
- ・卒業式
- ・部活動、サークル（運動部、文化部10枚ずつ程）、出来れば活動中のもの
- ・授業風景（1学部2～3枚、出来れば薬学部は実験中のもの）
- ・新入生サポートの写真
- ・就活関連（支援バス、宿泊施設、松山大学合同説明会、就活講座、キャリアセンター）
- ・学部長の写真、ゼミナール表彰式

以上となります。

私達でも撮影に臨んでおりますが、とにかく沢山の素材が欲しい為  
行事等の写真に加えて項目が多くなってしまい申し訳ありません。  
可能ならば各項目5～10枚程お貸しして頂けるとありがたいです。

用意して下さったならば  
メール等でご都合の良い日時を指定していただければ、伺わせていただきます。

お忙しいところお手数をおかけしますが、どうぞ宜しくお願い致します。

松山大学 経営学部 経営学科  
宮内隆成

図 5

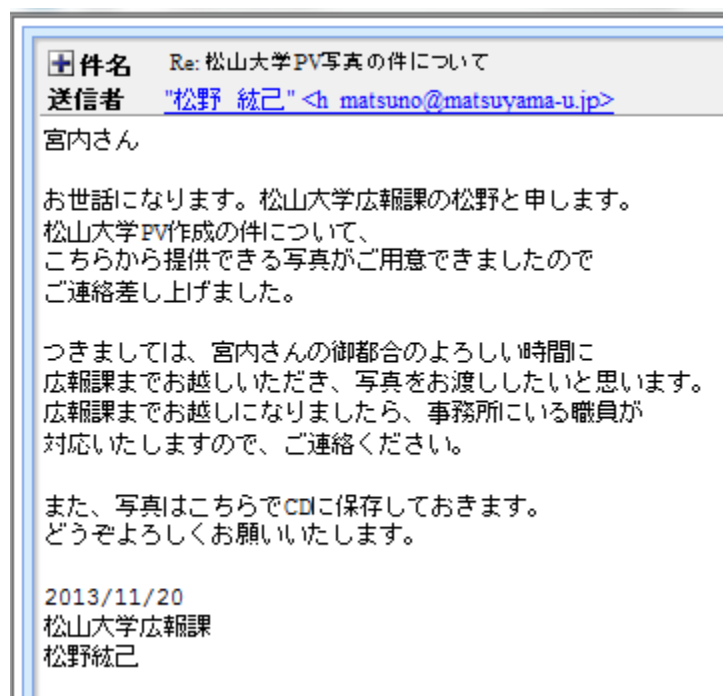


図 6

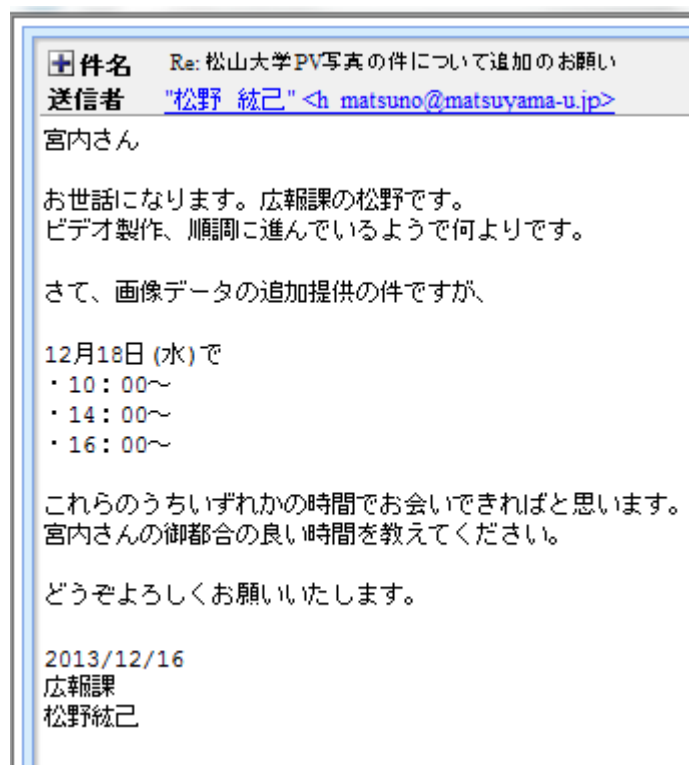


図 7

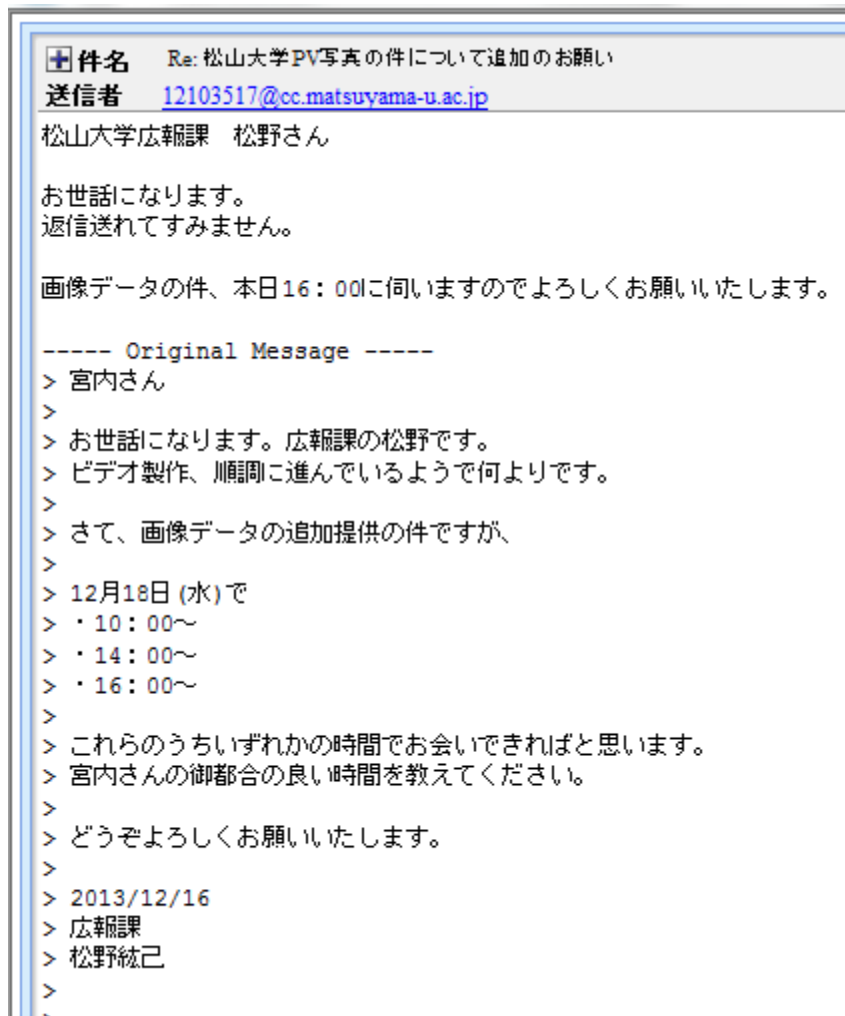


図 8

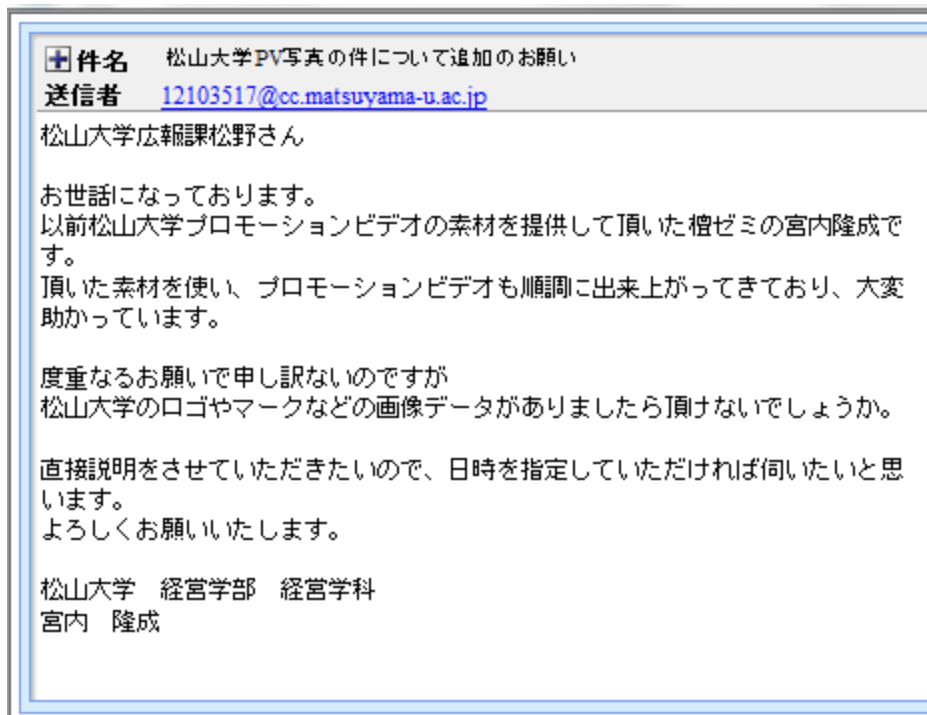


図 9

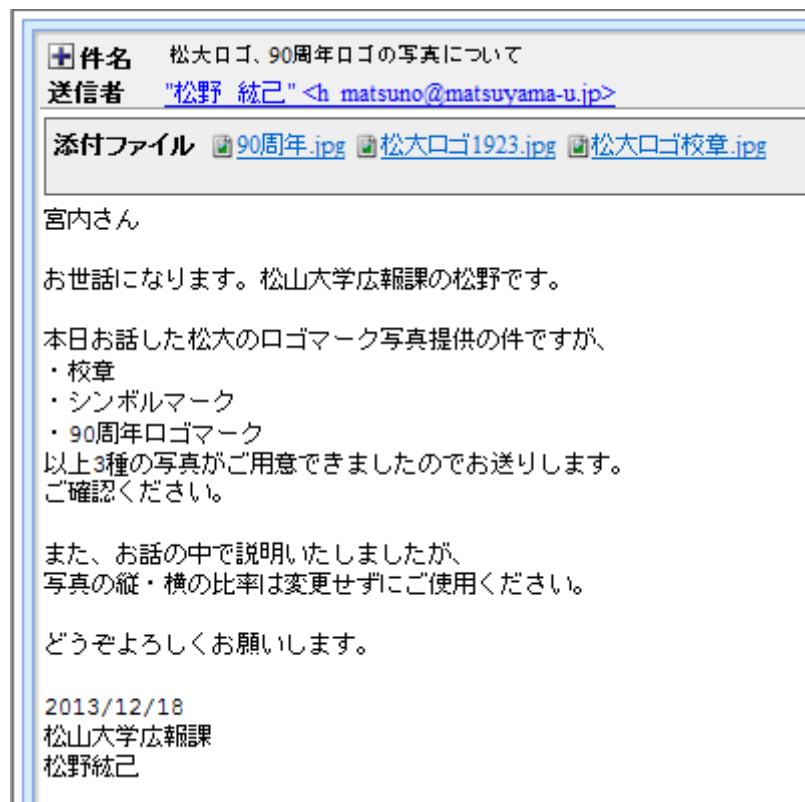


図 10



## 2-4. 教授と在学生へのアポイントメント

### <1>授業風景撮影アポイントメント

学部毎に授業風景を撮影させてもらう為に、在学生の声を基に授業を選択。授業を担当する教授に撮影協力依頼のメールを送り、アポイントを取る。(図 4 参照) 研究室へ詳細を説明しに伺い、撮影許可をいただき、日時を決め撮影。

学部	科目	教授名
経済	社会経済学入門	西尾圭一郎 先生
経営	経営科学	檀裕也 先生
人文 英語英米	英語コミュニケーションⅡ	Gregory Gray 先生
人文 社会	家族社会学	石川良子 先生
法	行政法Ⅱ	倉澤生雄 先生
薬	ゼミ	松岡一郎 先生

表 2. 撮影授業と担当教授一覧

```
>  
> -----Original Message-----  
> From: u.ac.jp  
> Sent: Monday, November 11, 2013 4:13 PM  
> To:  
> Subject: 松山大学経営学部の田村香奈子です。  
>  
> 松岡先生  
>  
> 初めまして。  
> 私は、松山大学経営学部檀ゼミの田村香奈子です。  
>  
> 今回、先生にお願いがあってご連絡させていただきました。  
> 私たち檀ゼミは、卒業制作として松山大学のプロモーションビデオを  
> 作っております。  
> そこで、学部紹介の際に、松岡先生のゼミの授業を  
> 撮影させていただきたいと思っております。  
>  
> 詳しい内容は、研究室にお伺いしてご説明したいと思っておりますので、  
> ご都合の良い日をご連絡いただけますでしょうか。  
>  
> よろしくお願ひ致します。
```

図 11

外国人の教授への依頼は、英語と日本語両方で送る。(図 12 参照) しかし、撮影の詳細を説明していなかったため、学生のプライバシーの問題などを考慮し一度断られる (図 13)。メールで研究室へ伺う日を決め (図 14)、詳細を説明しに研究室へ。プロジェクトの説明するレジュメを英語で作成。やはり、メールでは上手く伝わっておらず、撮影時間の長さの考えの違いが分かり、教授の

理解を得る。撮影させていただけることになる。

```
On Nov 11, 2013, at 4:47 PM, <12102339@cc.matsuyama-u.ac.jp>
<12102339@cc.matsuyama-u.ac.jp> wrote:

> Dear Prof. Dr. Gregory D.Gray,
>
> I am Kanako Tamura, a student of matsuyama university.
> I am writing to request you to permit our new project .
> We are making promotion video of matsuyama university as
graduation
> project.
> We would like to record your lesson(ex.English communication II)
in the
> video.
> If you accept our proposal,
> would you tell me when is the best day with your convenience for
video
> recording in your class?
>
> yours sincerely,
> Kanako Tamura
`
```

図 12

件名 Re: promotion video ▼ 操作を選択

送信者 "[Gregory Gray](mailto:Gregory Gray)" <[graygregory@hotmail.com](mailto:graygregory@hotmail.com)>

宛先 "[12102339@cc.matsuyama-u.ac.jp](mailto:12102339@cc.matsuyama-u.ac.jp)" <[12102339@cc.matsuyama-u.ac.jp](mailto:12102339@cc.matsuyama-u.ac.jp)> <[12102339@cc.matsuyama-u.ac.jp](mailto:12102339@cc.matsuyama-u.ac.jp)>

送信日時 2013年11月12日(火) 16:15:46

Dear Kanako,

Thank you for the invitation, but I will decline.  
I don't really want outside people coming into my classroom  
and filming. My students are kind of shy and I worry about privacy  
issues.

However, I would like to hear about your project and know what is  
the purpose.  
Please visit my office tomorrow at 2:00 p.m. if you can.

Sincerely,

G. Gray

図 13

```
> Date: Tue, 12 Nov 2013 18:32:51 +0900
> From: 12102339@cc.matsuyama-u.ac.jp
> To: graygregory@hotmail.com
> Subject: Re: promotion video
>
> Dear Prof. Dr. Gregory D.Gray,
>
> Thank you very much four your email.
> I understand your opinion.But I will have a schedule tomorrow on
that
> time.
> That's why I can't go your office.
>
> Would you tell me when is the another day with your convenience
for
> explanation ?
>
> I'm looking forward to seeing you.
>
> yours sincerely,
> Kanako Tamura
>
```

図 14

件名 Re: promotion video ▼ 操作を選択

送信者 [12102339@cc.matsuyama-u.ac.jp](mailto:12102339@cc.matsuyama-u.ac.jp)

Dear Prof. Dr. Gregory D.Gray,

Thank you for cooperate our graduation project today.

I'll send your e-mail when our PV is completed.

Kanako Tamura  
松山大学  
経営学部 経営学科  
田村香奈子  
[12102339@cc.matsuyama-u.ac.jp](mailto:12102339@cc.matsuyama-u.ac.jp)

図 15

## 〈2〉在学生インタビューアポイントメント

学部別にインタビューをお願いする学生を選び、インタビューを行う。

〈インタビュー内容〉

### 1. 経済学部

- ・経済学部を選んだ理由
- ・松大に入学してよかったこと、大変だったこと
- ・おすすめの授業
- ・なぜ部活をしようと思ったか
- ・部活をしていて身についたこと
- ・部活と授業の両立はできたか
- ・アルバイトをする時間はあったか
- ・部活内の上下関係
- ・松大に入ってくる人に一言

### 2. 経営学部

- ・松大を選んだ理由
- ・経営学部を選んだ理由
- ・松大に入学してよかったこと、大変だったこと
- ・ミューズになぜ入ったか
- ・ミューズはどんな活動をしているか
- ・ミューズの代表をしていてよかったこと
- ・松大に入ってくる人に一言

### 3. 人文学部英語英米学科

- ・松大を選んだ理由
- ・人英を選んだ理由
- ・松大に入学してよかったこと、大変だったこと
- ・おすすめの授業
- ・なぜ留学しようと思ったか
- ・留学までの流れ
- ・留学してよかったこと、大変だったこと
- ・英語力はどのくらいありましたか
- ・松大に入ってくる人に一言

### 4. 人文学部社会学科

- ・人社を選んだ理由
- ・松大に入学してよかったこと、大変だったこと

- ・おすすめの授業
- ・何の資格を持っているか
- ・なぜその資格を取ろうと思ったか
- ・その資格をとってよかったこと
- ・どんな講座を受けたか、その内容
- ・松大に入ってくる人に一言

## 5. 法学部

- ・松大を選んだ理由
- ・法学部を選んだ理由
- ・松大に入学してよかったこと、大変だったこと
- ・就活では何社受けたか
- ・松大の就活サポートはどうだったか
- ・就活のための講座やセミナーはどうだったか
- ・インターンシップを受けた感想
- ・就活に関するアドバイス
- ・松大に入ってくる人に一言

## 6. 薬学部

- ・薬学部を選んだ理由
- ・松大に入学してよかったこと、大変だったこと
- ・おすすめの授業
- ・松山は住みやすいか
- ・家はどうやって決めたか
- ・一人暮らしで大変だったこと
- ・食事はどのようにしていますか
- ・松大に入ってくる人に一言



図 16



図 17

## 2-5.撮影

### <1> 授業風景撮影

各授業、最初にこの卒業制作の詳細を説明し、学生にも協力を得る。それと同時に、学生のプライバシーの問題もあるため、写りたくない学生には挙手してもらい、写さないように配慮。授業開始から 20～30 分程度、教室の四隅から授業風景を撮影。

### <2> インタビュー撮影

一人ひとり撮影場所を変え、3～5 分程度で 7～9 個の質問に対して答えてもらう。

### <3> 施設撮影

正門と温山会館の撮影は、車などが映りこまないように早朝に撮影。

### <4> 大変だったこと

授業風景の撮影では、同じ画にならないように四隅から撮ったり、三脚を伸ばし高い所から撮影したりした。ズームをする時に、ぶれないようにゆっくりズームするように気を付けた。インタビューの撮影では、ビデオカメラが周りの音をよく拾うので、静かな所を撮影場所を選択した。

## 2-6.編集作業

Adobe Premiere Pro CC というソフトを使い、動画と写真を編集。

<流れ>

- 1、松大ロゴ
- 2、正門
- 3、温山会館を背景に、大学の歴史を説明
- 4、授業風景を背景に、学部紹介・在学生インタビュー
- 5、部活動・サークル紹介
- 6、年間行事紹介

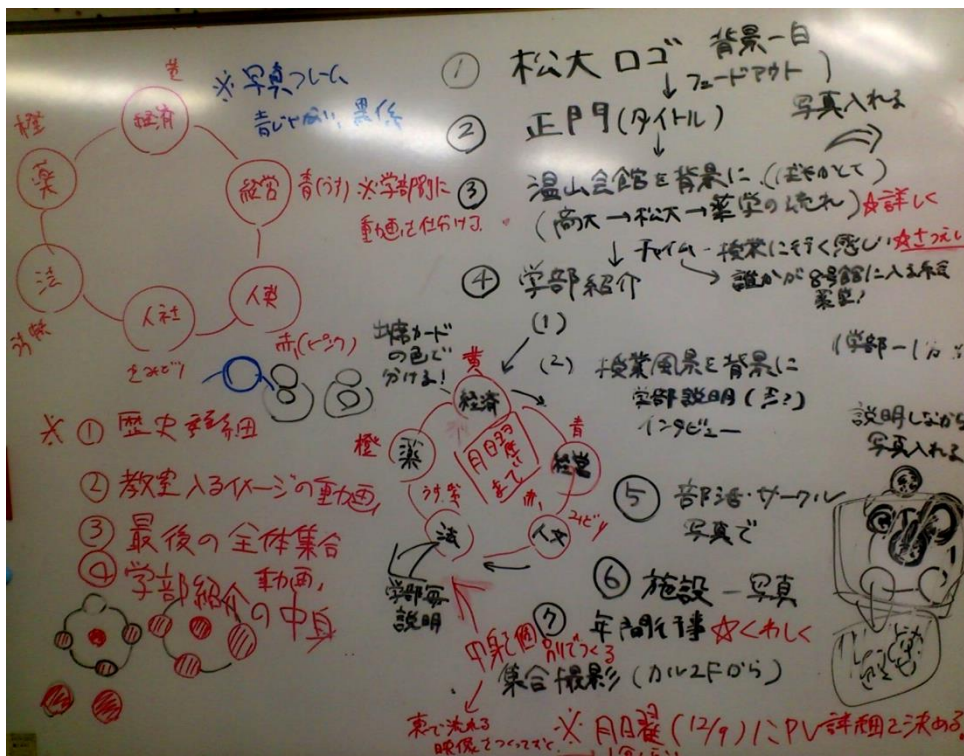


図 18

### <1> 編集手順

撮影した動画を編集するため、プロモーションビデオに使用する部分をだまかに選択。そして、インタビューも2〜3個の質問に絞る。

### <2> 大変だったこと

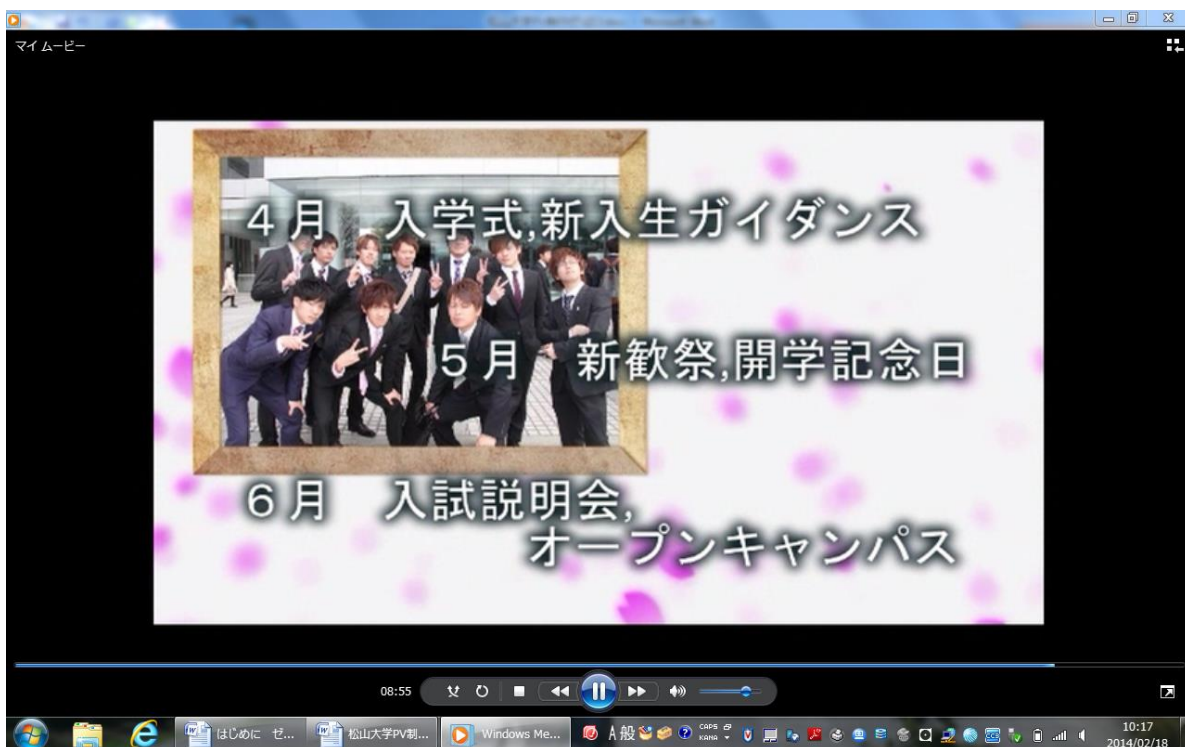
一からこのソフトを使った為、操作に慣れることから始まった。特殊効果など、どこに何の機能があるかが分からなかったり、最終的にはナレーションも入れることになっていたため、そのナレーションの時間を考えて動画を編集したりすることが大変だった。さらに、外部で写真のフレームなどを作成するのも大変だった。











## 2-7.まとめ

最初想像していた以上に許可を取ることや動画構成を考えることに手間取り、撮影を始めるまでに時間がかかりました。撮影になかなか移れないことで、きちんと完成できるのかとても不安でした。しかし、チーム内で役割分担をすることによって効率良く進めることができ、さらに、檀先生にアドバイスをいただいたことで無事完成させることができました。

このPVを見た受験生が、少しでも松山大学に興味を持ってくれば嬉しいです。

### 謝辞

制作を進めるにあたって、撮影許可・写真とロゴマークを提供して下さった広報課の皆さん、授業風景の撮影協力をして下さった西尾圭一郎先生、Gregory Gray先生、石川良子先生、倉澤生雄先生、松岡一郎先生、インタビューを引き受けて下さった学生の皆さん、撮影協力をして頂いた学生の皆さん、撮影協力と制作指導をして下さった担当教授の檀裕也先生に感謝致します。

## 参考文献

- ・ Premiere Pro CC スーパーリファレンス (阿部信行 2013年 ソーテック社)
- ・ はじめての動画処理プログラミング (土井滋貴 2007年 CQ 出版株式会社)
- ・ 動画素材 123 (ES...(Molojun) 2009年 株式会社ビー・エヌ・エヌ新社)
- ・ 松山大学 2014年度 入学案内パンフレット
- ・ 2013年度 松山大学入学案内パンフレット
- ・ Club & Circle 2013 パンフレット
- ・ 新入生ガイドブック 2013 経済学部パンフレット
- ・ 経営学部ガイドブック
- ・ 経営学部新入生ガイドブック
- ・ 松山大学人文学部社会学科 2013年度パンフレット
- ・ 松山大学人文学部英語英米文学科 2013年度パンフレット
- ・ 法学部パンフレット
- ・ 松山大学薬学部医療薬学科パンフレット

本章に記載されている会社名、商品名、製品名などは、各社の商標または登録商標です。

# 3. ゲームアプリ制作

## 3-1. はじめに

携帯ゲーム機の進化、携帯電話やスマートフォンの普及によってゲーム産業が注目を浴びている。中でも「パズル&ドラゴン」は2300万ダウンロードを突破し、グッズ販売や3DSでのゲームソフト販売をするなど、その人気は衰えない。そこでAndroidのゲームアプリケーションというものに興味を持ち、卒業研究として作成することにした。

アプリケーションを作成するにあたって「Eclipse」という統合開発環境の元、「JAVA言語」を使用している。

## 3-2. 概要

今回作ったゲームは、農業シミュレーションゲームである。植物の種を植え、収穫をし、販売することによってお金を得る。そのお金で新しい種を購入し、また植える。この動作を繰り返し行うゲームである。

またこのアプリケーションはいくつかのプログラムを大元にその応用で動いている。そこで主要プログラムの説明をしていく。

## 3-3. 動作説明およびプログラム説明

以下のプログラムは下記の順番で説明していく。

- どのようなプログラムなのか
- ゲームにどのように使われているのか
- プログラムの説明

### <1> データベース

特定のテーマに沿ったデータを集めて管理し、容易に検索・抽出などの再利用をできるようにしたものである。「SQLite」と呼ばれるアプリケーションに組み込んで利用される軽量のデータベース管理システムを使用している。

タネ、収穫物などの個数や金額を保存に使用している。

図1より、20行目の「create table」でテーブルを作成している。「person\_table」はテーブル名、

```
18 // 種のテーブル
19 db.execSQL(
20     "create table person_table(i text, name text not null, age text);"
21 );
22 //表の行の挿入
23 db.execSQL("insert into person_table(i,name,age) values (0, 'トマト', 2);");
24 db.execSQL("insert into person_table(i,name,age) values (1, 'きゅうり', 1);");
25 db.execSQL("insert into person_table(i,name,age) values (2, 'ピーマン', 1);");
26 db.execSQL("insert into person_table(i,name,age) values (3, 'かぼちゃ', 1);");
27 db.execSQL("insert into person_table(i,name,age) values (4, 'ニンジン', 0);");
28 db.execSQL("insert into person_table(i,name,age) values (5, 'ハクサイ', 0);");
```

「i」は番号、「age」は個数を表している。23行目から28行目は

表1 データベースイメージ

i	name	age
0	トマト	2
1	きゅうり	1
2	ピーマン	1
3	かぼちゃ	0
4	ニンジン	0
5	ハクサイ	0

図1 PersonOpenHelper.java データベース作成

「insert」によってテーブルの行を追加している。テーブルは表 1 のような形で保存されている。またデータを読み出す際は「rawQuery」を使用する。「where」で読み出す条件を付けることができ、図 2 では「age が 1 以上の」という条件を付けている。

```
呼び出したデータ //数量が1以上のタネを表示させる
データベースを Cursor cursor = db.rawQuery("select i, name, age from person_table where age >= ?;", new String[]{"1"});
```

図 2 samplesqlite1.java データベース呼び出し

用する際にデータを配列に格納している。配列とは同じ型のデータを複数もつことができるオブジェクトである。図 3 では、文字型の「items[]」、数値型の「items2[]」の配列を準備している。「getCount」によってデータベースのデータの数だけデータを持つことができる。どのようなデータをいくつ格納しているかという、ループ処理である「for 文」を使用して行っている。数値型「l」の初期値を 0 とし、1 ループごとに「l + 1」を行う。図 3 の 1 行目の「moveToFirst」で最初のデータを読み込みが終わって「l」は 0 なので、「items[l]」の 0 番目に文字型に変換した図 3 から「name」の列と、「age」の列に「個」を付けた 1 行目のデータを格納し、「items2[l]」の 0 番目に数値型の「i」の列の 1 行目のデータの格納を行っている。この動作を 2 行目、3 行目と繰り返し行い、最終的に「is End of file」

```
boolean isEOF = cursor.moveToFirst();
final CharSequence[] items = new CharSequence[cursor.getCount()];
final int[] items2 = new int[cursor.getCount()];

for (int l = 0; isEOF; l++) {
    items[l] = String.format("%s : %d個", cursor.getString(1), cursor.getInt(2));
    items2[l] = cursor.getInt(0);
    isEOF = cursor.moveToNext();
}

cursor.close();
```

まで、つまり格納できるデータが無くなるまでループを繰り返す。図 2 より「age が 1 以上」なので表 2 のように配列に格納されている。

図 3 samplesqlite1.java 配列に格納

表 2 配列のイメージ

items			items2	
l	name	age	l	i
0	トマト	2	0	0
1	きゅうり	1	1	1
2	ピーマン	1	2	2

また、データベースに保存されているデータを変更する動作として「values.put」がある。図 4 では、「person\_table」の「i」が「shopid」と同じ値と対応した「age」に数値の 1 を入力して更新している。

```
//タネを増やす
String a1 = String.format("select i, name, age from person_table where i = %d", shopid);
Cursor c = db.rawQuery(a1, null);
c.moveToFirst();
ContentValues values4 = new ContentValues();
values4.put("age", 1);
db.update("person_table", values4, "i = " + shopid, null);
c.close();
```

図 4 samplesqlite1.java データベースアップデート

最終的にデータベースを閉じる処理として「close」を行うことでアプリケーションのメモリを軽くすることができる。

## ＜2＞ アラートダイアログ

通常、アラートダイアログとはポップアップやメッセージを表示させる為に用いる。しかし今回のゲームアプリでは、ダイアログに選択肢を付け加えたり、ダイアログからダイアログに推移させることで、アイテムボタンや shop ボタンに使用している。

```

607 //アイテムボタン
608 public void doAction(View view){
609     koukaon.playSe();
610
611     //数量が1以上のタネを表示させる
612     Cursor cursor = db.rawQuery("select i, name, age from person_table where age >= ?;", new String[]{"1"});
613     boolean isEof = cursor.moveToFirst();
614
615     AlertDialog.Builder builder = new AlertDialog.Builder(this);
616     builder.setTitle("SEEDS");

```

図 5 samplesqlite1.java アラートダイアログ準備

アイテムボタンをタップした時の動作指定をする場所に 615 行目のようにダイアログを作り、次の行でタイトルをつける。

```

618     final CharSequence[] items = new CharSequence[cursor.getCount()];
619     final int[] items2 = new int[cursor.getCount()];
620
621     for (int i = 0; isEof; i++) {
622         items[i] = String.format("%s : %d個", cursor.getString(1), cursor.getInt(2));
623         items2[i] = cursor.getInt(0);
624         isEof = cursor.moveToNext();
625     }

```

図 6 samplesqlite1.java ダイアログの項目準備

618～625 行目で図 8 のフローチャートのループを行っている。621 行目でデータベースを呼び出し、i を 0 から始めてデータベースが終わるまでループを 1 回回るとに i に 1 ずつ足していく。ループの中では 622 行目でデータベースにあるタネの名前と数量を items で取得している。623 行目では後から使用するタネの番号を items2 で取得している。

```

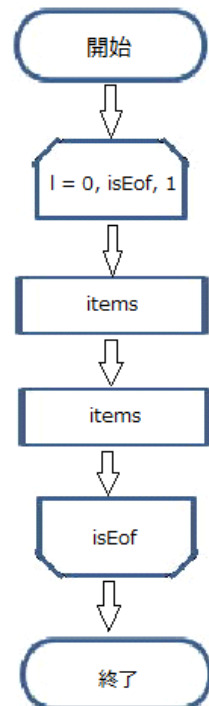
629 //アイテムを選択したとき時
630 builder.setSingleChoiceItems(items, -1, new DialogInterface.OnClickListener(){
631     public void onClick(DialogInterface dialog, int which){
632         taneid = items2[which];
633         pref = PreferenceManager.getDefaultSharedPreferences(activity);
634         Editor edit = pref.edit();
635         edit.putInt("GAZ04", taneid);
636         edit.commit();
637     }
638 });

```

図 7 samplesqlite1.java タネ番号を保存



図 9 ダイアログ動作画面



取得した items は 630 行目でダイアログに表示させている。それと同時に「SingleChoiceItems」でラジオボタン (図 9 ダイアログ右側) をつける。631～636 行目はラジオボタンで選択したタネの番号を保存している。



### <3> グラフィック描画

「View」というクラスの「onDraw」メソッドを使用してグラフィック描画、あるいは表示の更新を行うことができる。

植物の描画および成長によつてのグラフィックの更新に使用している。

アイテムダイアログから植えたい植物を選択することで（図 7）、対応する「taneid」を取得することができる。

図 9 からアイテム選択後に「OK」ボタンを押すことでメイン画面に戻り、「invisible」（図 10）の「チェックボックス」が 3 つと「植える」ボタンが「VISIBLE」（図 11）に変更されメイン画面に表示される。（図 12）

```
<Button
    android:id="@+id/uerubutton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="81dp"
    android:onClick="doAction4"
    android:text="@string/uerubutton"
    android:visibility="invisible" />

<CheckBox
    android:id="@+id/checkBox1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignRight="@+id/button"
    android:layout_below="@+id/uerubutton"
    android:layout_marginRight="28dp"
    android:layout_marginTop="49dp"
    android:text="@string/CheckBox1"
    android:visibility="invisible" />
```

図 10 activity\_splite\_sample11.xml  
invisible 設定

```
//植えるボタンとチェックボックスを表示させる
@Override
public void onClick(DialogInterface dialog, int which){
    findViewById(R.id.uerubutton).setVisibility(View.VISIBLE);
    findViewById(R.id.checkBox1).setVisibility(View.VISIBLE);
    findViewById(R.id.checkBox2).setVisibility(View.VISIBLE);
    findViewById(R.id.checkBox3).setVisibility(View.VISIBLE);
}
```

図 11 samplesqlite1.java VISIBLE 設定



図 12 チェックボックス

「チェックボックス」にチェックを入れると状態が「true」に変更される。「植える」ボタンを押すことで「plant」クラスが生成される。図 7 で取得した「taneid」を元に「gazou\_table」を参照して対応するグラフィックデータを変数「x」に取得する。変数「x」は「plant」クラスに継承されるため、変数「x」と変数「n」（図 13）は同じデータを持っていることになり、最終的に「onDraw」メソッドで「drawable」によって描画される。（図 14 および図 15）

```

if(chkbox1.isChecked() == true){
    check = check + 1;
    uetaswitch1 = 1;
    plant1 = new Plant(activity, db, taneid);
    paramschk1.setMargins(chkbox1.getLeft(), chkbox1.getTop(),0,0);
    String a = String.format("select i, w, name, syo from gazou_table where i = %d", taneid);
    Cursor cursor = db.rawQuery(a, null);
    cursor.moveToFirst();
    int x = cursor.getInt(3);
    plant1.setImage(x);
    cursor.close();}

```

図 13 samplesqlite1.java グラフィック準備

```

public void setImage(int n){
    Resources resources = context.getResources();
    drawable = resources.getDrawable(n);
}

```

図 14 Plant.java グラフィック準備

```

@Override
protected void onDraw(Canvas c){
    super.onDraw(c);
    if (drawable != null){
        drawable.setBounds(10,10,100,100);
        drawable.draw(c);
    }
}

```

図 15 Plant.java グラフィック描画

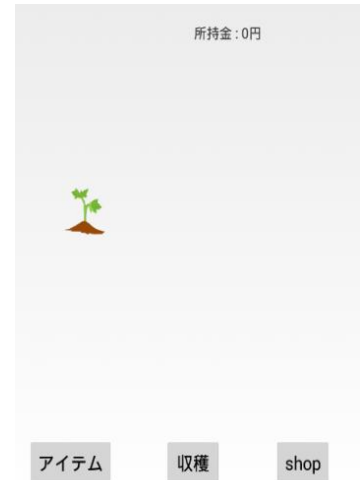


図 16 グラフィックの描画

成長には三段階のステップがあり、時間経過で成長していく。時間は「timer」と「run」メソッドで動かしており、図 17 のプログラムは「1000 ミリ秒」毎に動くようになっている。経過した時間はわかるように「age1」に保存している。「growsparm[r]」、「growsparm2[r]」は何秒経過したら

```

timer = new Timer(true);
timer.schedule( new TimerTask(){
    @Override
    public void run() {

        mHandler.post( new Runnable() {
            public void run() {
                Log.v("age1", "" + age1);
                if (age1 > growsparm[r]) {
                    Resources resources = context.getResources();
                    drawable = resources.getDrawable(items3[1]);
                    status = 2;
                    invalidate();
                }

                if (age1 > growsparm1[r]) {
                    Resources resources = context.getResources();
                    drawable = resources.getDrawable(items3[2]);
                    status = 3;
                    invalidate();
                }

                age1 ++;
            }
        });
    }
}, 1000, 1000);

```

図 17 Plant.java 成長処理

ら画像が変わるか、という時間が配列で保存されていて、「r」は図 7 の「taneid」と対応しているため、各 0 番目がトマト、1 番目がきゅうり...となっている。「items3」には成長の画像が保存されていて、「age1」が「growsparm[r]」、「growsparm2[r]」の数値を超えると「invalidate」によって描画するようになっている。「status」に「3」の数値が入ると収穫できるようになる。

```

int[] growsparm = {2, 1, 2, 2};
int[] growsparm1 = {4, 2, 4, 4};

```

図 18 Plant.java 各植物の成長速度



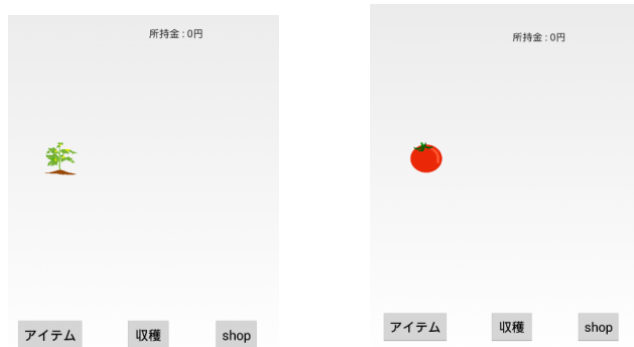


図 19 成長画面

#### <4> 効果音、BGM

アプリ起動時に音楽をかける。ボタンをタップした時に効果音を鳴らす。

```

1 package gudon.sample.sqlite1;
2
3 import android.media.MediaPlayer;
4 import android.view.View;
5 import android.content.Context;
6
7 //BGM
8 public class music extends View{
9     MediaPlayer bgm = MediaPlayer.create(getContext(), R.raw.yasashi);
10
11     public music(Context context) {
12         super(context);
13         bgm.setLooping(true);
14         bgm.start();
15     }
16
17     public void pause() {
18         bgm.pause();
19     }
20
21     public void reStartsound(){
22         bgm.start();
23     }
24 }
25
26
27

```

図 20 music.java BGM 準備

```

1 package gudon.sample.sqlite1;
2
3 import android.content.Context;
4 import android.media.AudioManager;
5 import android.media.SoundPool;
6
7 //効果音
8 public class SePlayer
9 {
10     private SoundPool soundPool;
11
12     private int koukaon;
13
14
15     public SePlayer(Context context)
16     {
17         // new SoundPool(読み込むファイル数,読み込む種類,読み込む質)
18         this.soundPool = new SoundPool(1, AudioManager.STREAM_MUSIC, 0);
19
20         // load(コンテキスト,読み込むリソースID,音の優先度)
21         this.koukaon = soundPool.load(context, R.raw.kettei, 1);
22     }
23
24     public void playSe()
25     {
26         // play(再生するサウンドID,左のボリューム,右のボリューム,優先度,ループ回数(0はしない、-1は無限),再生レート)
27         soundPool.play(koukaon, 1.0f, 1.0f, 1, 0, 1.0f);
28     }
29 }
30

```

図 21 SePlayer 効果音準備

BGM、効果音の素材を raw ファイルに入れておき、BGM 用(music クラス)図 20 と効果音用(SePlayer クラス)図 21 を新しく用意する。

```

60 //効果音・BGM準備
61 koukaon = new SePlayer(this);
62 bgm = new music(this);
63 layoutbgm = new LinearLayout(this);
64 setContentView(layoutbgm);
65 layoutbgm.addView(bgm);

```

図 22 samplesqlite1.java BGM、効果音準備

SQLiteSample1 クラスでは次の 61、62 行目で用意したものを呼び出す。63～65 行目でアプリに BGM をつけることができる。効果音をつけるには、

```

607 //アイテムボタン
608 public void doAction(View view){
609     koukaon.playSe();
610
611     //数量が1以上のタネを表示させる
612     Cursor cursor = db.rawQuery("select i, name, age from person_table where age >= ?;",new String[]{"1"});
613     boolean isEof = cursor.moveToFirst();
614
615     AlertDialog.Builder builder = new AlertDialog.Builder(this);
616     builder.setTitle("SEEDS");
617
618     final CharSequence[] items = new CharSequence[cursor.getCount()];
619     final int[] items2 = new int[cursor.getCount()];
620

```

図 24 samplesqlite1.java 効果音の動作プログラム

そのボタンのタップした時の動作設定の部分に 609 行目の 1 文を書き込むと、ボタンタップ時に果音を鳴らすことができる。

## <5> つづきから処理

「preferenceManager」を使うことでデータの書き込み、書き出しを行うことができる。「put」で書き込み、「get」で書き出しを行うことができる。

成長段階、経過時間の保存を行っている。

「pref」から保存の領域を作っておく。「datap」が「false」の場合、植物が植えられている状態なので、その場合に「put」によって保存が行われている。「pa」などは呼び出しを行う key でコンマの後ろの「plant1.age」が保存したいデータである。最終的に「commit」で確定することができる。(図)

```

pref = PreferenceManager.getDefaultSharedPreferences(this);
Editor edit = pref.edit();
if(TitleActivity.datap == false){
edit.putInt("pa", plant1.age1);//植えてからの経過時間
edit.putInt("pr", plant1.r);//種のID
}
//書き込みと呼び出しprefとセット
edit.putLong("currentstart", currentstart);
edit.putBoolean("dp", TitleActivity.datap);
edit.commit();
db.close();

```

図 25 samplesqlite1.java 書き込み処理

ゲームにはタイトル画面があり、「つづきから」からを押した後「datap」が「false」の場合書き出しが行われる。「seichoutime」にはバックグラウンドで経過した時間を取得している。その後「getInt」

```
if(TitleActivity.datap == false){
    //現在の時刻(エポックからのミリ秒取得)
    final long currentend = System.currentTimeMillis();
    int d = pref.getInt("pa", 0);
    long currentstart;
    long keikatime;
    currentstart = pref.getLong("currentstart", -1);

    keikatime = currentend - currentstart;
    keikatime = (long)Math.floor(keikatime / 1000);
    seichoutime = d + (int)keikatime;

    taneid = pref.getInt("GAZ04", 0);
    plant1 = new Plant(activity, db, taneid);

    //画面を切り替えたときに、成長していたら1段階目を表示させない
    int a = seichoutime - plant1.growsparam[taneid];

    if (a <= 0) {
        Resources resources = activity.getResources();
        plant1.drawable = resources.getDrawable(plant1.items3[0]);
        plant1.status = 1;
        plant1.setImage(plant1.items3[0]);
    }
}
```

図 26 samplesqlite1.java 呼び出し処理

によって植えている「taneid」を取得。その後上記のグラフィック描画と同じ動作を行うことで、植えている植物、さらにバックグラウンドでの時間経過を呼び出し復元することができる。(図 26)



図 27 タイトル画面

## 3-4. プログラム紹介

ここにプログラムの全文を記述しておく。

### <1> sample.sqlite1.java(メインアクティビティ)

```
1 package gudon.sample.sqlite1;
2
3 import android.os.Bundle;
4 import android.preference.PreferenceManager;
5 import android.content.DialogInterface;
6 import android.content.SharedPreferences;
7 import android.content.SharedPreferences.Editor;
8 import android.content.res.Resources;
9 import android.database.sqlite.SQLiteDatabase;
10 import android.app.Activity;
11 import android.app.AlertDialog;
12 import android.database.Cursor;
13 import android.view.View;
14 import android.content.ContentValues;
15 import android.widget.LinearLayout;
16 import android.widget.TextView;
17 import android.view.ViewGroup;
18 import android.widget.CheckBox;
19
20 public class SQLiteSample1 extends Activity {
21
22
23     music bgm;
24     PersonOpenHelper helper;
25     SQLiteDatabase db;
26
27
28     private LinearLayout.LayoutParams paramschk1 = new LinearLayout.LayoutParams
29         (ViewGroup.LayoutParams.WRAP_CONTENT, ViewGroup.LayoutParams.WRAP_CONTENT);
30     private LinearLayout.LayoutParams paramschk2 = new LinearLayout.LayoutParams
31         (ViewGroup.LayoutParams.WRAP_CONTENT, ViewGroup.LayoutParams.WRAP_CONTENT);
32     private LinearLayout.LayoutParams paramschk3 = new LinearLayout.LayoutParams
33         (ViewGroup.LayoutParams.WRAP_CONTENT, ViewGroup.LayoutParams.WRAP_CONTENT);
34     private int uetaswitch1 = 0;
35     private int uetaswitch2 = 0;
36     private int uetaswitch3 = 0;
37     private final int WC = ViewGroup.LayoutParams.WRAP_CONTENT;
38     private LinearLayout layoutmoney;
39     private LinearLayout layoutyasai1;
40     private LinearLayout layoutyasai2;
41     private LinearLayout layoutyasai3;
42     public static Plant plant1;
43     public static Plant plant2;
44     public static Plant plant3;
45     private int taneid = -1;
46     private int shopid = -1;
47     private int shoppedan = 0;
48     private int shozikin = 0;
49     private int syukakugoukei = 0;
50     private TextView tvmoney;
51     private MediaPlayer koukaon;
52     private LinearLayout layoutbgm;
53     private SharedPreferences pref;
54     final Activity activity = this;
55     public static int seichoutime;
56
57     @Override
58     protected void onCreate(Bundle savedInstanceState) {
59
60         //効果音・BGM準備
61         koukaon = new MediaPlayer(this);
62         bgm = new music(this);
63         layoutbgm = new LinearLayout(this);
64         setContentView(layoutbgm);
65         layoutbgm.addView(bgm);
66
67         super.onCreate(savedInstanceState);
```

```

88
89 //レイアウト表示
90 setContentView(R.layout.activity_sqlite_sample7);
91
92
93 //データベース準備
94 helper = new PersonOpenHelper(this);
95 db = helper.getReadableDatabase();
96
97 //所持金の表示
98 layoutmoney = new LinearLayout(this);
99 this.addView(layoutmoney, new ViewGroup.LayoutParams(WC, WC));
100 LinearLayout.LayoutParams params1 = new LinearLayout.LayoutParams
101     (ViewGroup.LayoutParams.WRAP_CONTENT, ViewGroup.LayoutParams.WRAP_CONTENT);
102 params1.setMargins(400, 80, 100, 100);
103 Cursor cursor = db.rawQuery("select name, kane from money_table where kane >= ?;", new String[]{"0"});
104 cursor.moveToFirst();
105 final CharSequence[] items7 = new CharSequence[cursor.getCount()];
106 items7[0] = String.format("%s : %d円", cursor.getString(0), cursor.getInt(1));
107 cursor.close();
108 tvmoney = new TextView(this);
109 tvmoney.setText(items7[0]);
110 layoutmoney.addView(tvmoney, params1);
111
112 //野菜表示
113 layoutyasai1 = new LinearLayout(this);
114 this.addView(layoutyasai1, new ViewGroup.LayoutParams(WC, WC));
115
116 layoutyasai2 = new LinearLayout(this);
117 this.addView(layoutyasai2, new ViewGroup.LayoutParams(WC, WC));
118
119 layoutyasai3 = new LinearLayout(this);
120 this.addView(layoutyasai3, new ViewGroup.LayoutParams(WC, WC));
121
122 //保存した野菜を呼び出し
123 pref = PreferenceManager.getDefaultSharedPreferences(this);
124
125 if (TitleActivity.datap == false) {
126     //現在の時刻 (エポックからのミリ秒取得)
127     final long currentend = System.currentTimeMillis();
128     int d = pref.getInt("pa", 0);
129     long currentstart;
130     long keikatime;
131     currentstart = pref.getLong("currentstart", -1);
132
133     keikatime = currentend - currentstart;
134     keikatime = (long) Math.floor(keikatime / 1000);
135     seichoutime = d + (int) keikatime;
136
137     taneid = pref.getInt("GAZ04", 0);
138     plant7 = new Plant(activity, db, taneid);
139
140     //画面を切り替えたときに、成長していたら1段階目を表示させない
141     int a = seichoutime - plant7.growtparam[taneid];
142
143     if (a <= 0) {
144         Resources resources = activity.getResources();
145         plant7.drawable = resources.getDrawable(plant7.items3[0]);
146         plant7.status = 1;
147         plant7.setImage(plant7.items3[0]);
148     }
149
150     int b = seichoutime - plant7.growtparam[taneid];

```

```

135         if (b >= plant1.growtparam[taneid]) {
136             Resources resources1 = activity.getResources();
137             plant1.drawable = resources1.getDrawable(plant1.items3[1]);
138             plant1.status = 2;
139             plant1.setImage(plant1.items3[1]);
140         }
141     }
142
143     int c = seichoutime - plant1.growtparam1[taneid];
144
145     if (c >= plant1.growtparam1[taneid]){
146         Resources resources2 = activity.getResources();
147         plant1.drawable = resources2.getDrawable(plant1.items3[2]);
148         plant1.status = 3;
149         plant1.setImage(plant1.items3[2]);
150     }
151
152     LinearLayout.LayoutParams paramschk1 = new LinearLayout.LayoutParams
153         (ViewGroup.LayoutParams.WRAP_CONTENT, ViewGroup.LayoutParams.WRAP_CONTENT);
154     paramschk1.setMargins(100, 800, 100, 100);
155
156     layoutyasail.addView(plant1, paramschk1);
157
158     plant1.invalidate();
159
160     }
161
162     }
163
164 }
165
166 //バックグラウンド時の処理
167 @Override
168 public void onPause(){
169     super.onPause();
170     //bgm.pause();
171     //現在の時刻(エポックからのミリ秒取得)
172     final long currentstart = System.currentTimeMillis();
173
174     pref = PreferenceManager.getDefaultSharedPreferences(this);
175     Editor edit = pref.edit();
176     if(TitleActivity.datap == false){
177         edit.putInt("pa", plant1.age1); //種えてからの経過時間
178         edit.putInt("pr", plant1.r); //種のID
179     }
180     //書き込みと呼び出しprefとセット
181     edit.putLong("currentstart", currentstart);
182     edit.putBoolean("dp", TitleActivity.datap);
183     edit.commit();
184     db.close();
185 }
186
187
188 //バックグラウンドから復帰した時の処理
189 @Override
190 public void onResume(){
191     super.onResume();
192     //bgm.reStartsound();
193 }
194
195
196 //アプリを終了させた時の処理
197 public void onDestroy(){
198     super.onDestroy();
199     //bgm = null;
200 }
201

```

```

202
203 //植えるボタン
204 public void doAction4(View view){
205     //koukaon.playSe();
206     final CheckBox chkbox1 = (CheckBox)findViewById(R.id.checkBox1);
207     final CheckBox chkbox2 = (CheckBox)findViewById(R.id.checkBox2);
208     final CheckBox chkbox3 = (CheckBox)findViewById(R.id.checkBox3);
209
210     //植えるボタンとチェックボックスを消す
211     findViewById(R.id.uerubutton).setVisibility(View.INVISIBLE);
212     findViewById(R.id.checkBox1).setVisibility(View.INVISIBLE);
213     findViewById(R.id.checkBox2).setVisibility(View.INVISIBLE);
214     findViewById(R.id.checkBox3).setVisibility(View.INVISIBLE);
215
216     if(taneid >= 0){
217
218         TitleActivity.datap = false;
219
220         pref = PreferenceManager.getDefaultSharedPreferences(activity);
221         Editor edit = pref.edit();
222         edit.putBoolean("dp", TitleActivity.datap);
223         edit.commit();
224
225         int check = 0;
226         if(chkbox1.isChecked() == true){
227             check = check + 1;
228             uetaswitch1 = 1;
229             plant1 = new Plant(activity, db, taneid);
230             paramschk1.setMargins(chkbox1.getLeft(), chkbox1.getTop(),0,0);
231             String a = String.format("select i, w, name, syo from gazou_table where i = %d", taneid);
232             Cursor cursor = db.rawQuery(a, null);
233             cursor.moveToFirst();
234             int x = cursor.getInt(3);
235             plant1.setImage(x);
236             cursor.close();
237
238             if(chkbox2.isChecked() == true){
239                 check = check + 1;
240                 uetaswitch2 = 1;
241                 plant2 = new Plant(activity, db, taneid);
242                 paramschk2.setMargins(chkbox2.getLeft(), chkbox2.getTop(),0,0);
243                 String a2 = String.format("select i, w, name, syo from gazou_table where i = %d", taneid);
244                 Cursor cursor = db.rawQuery(a2, null);
245                 cursor.moveToFirst();
246                 int a = cursor.getInt(3);
247                 plant2.setImage(a);
248                 cursor.close();
249
250                 if(chkbox3.isChecked() == true){
251                     check = check + 1;
252                     uetaswitch3 = 1;
253                     plant3 = new Plant(activity, db, taneid);
254                     paramschk3.setMargins(chkbox3.getLeft(), chkbox3.getTop(),0,0);
255                     String a3 = String.format("select i, w, name, syo from gazou_table where i = %d", taneid);
256                     Cursor cursor = db.rawQuery(a3, null);
257                     cursor.moveToFirst();
258                     int a = cursor.getInt(3);
259                     plant3.setImage(a);
260                     cursor.close();
261
262                     String ko1 = String.format("select i, name, age from person_table where i = %d", taneid);
263                     Cursor cursor = db.rawQuery(ko1, null);
264                     cursor.moveToFirst();
265                     int ko2 = cursor.getInt(2);
266                     cursor.close();
267
268                     if(check <= ko2){

```

```

269 //植えたタネを描画
270
271 if (checkbox1.isChecked() == true){
272     layoutyasai1.addView(plant1, paramschk1);
273     checkbox1.setChecked(false);
274     Cursor tane = db.rawQuery("select i, name, age from person_table where i = ?;", new String[]{" " + plant1.r});
275     tane.moveToFirst();
276     int a = tane.getInt(2);
277     int b = a - 1;
278     ContentValues values = new ContentValues();
279     values.put("age", b);
280     db.update("person_table", values, "i = " + plant1.r, null);
281     tane.close();
282 }
283
284 if (checkbox2.isChecked() == true){
285     layoutyasai2.addView(plant2, paramschk2);
286     checkbox2.setChecked(false);
287     Cursor tane = db.rawQuery("select i, name, age from person_table where i = ?;", new String[]{" " + plant2.r});
288     tane.moveToFirst();
289     int a = tane.getInt(2);
290     int b = a - 1;
291     ContentValues values = new ContentValues();
292     values.put("age", b);
293     db.update("person_table", values, "i = " + plant2.r, null);
294     tane.close();
295 }
296
297 if (checkbox3.isChecked() == true){
298     layoutyasai3.addView(plant3, paramschk3);
299     checkbox3.setChecked(false);
300     Cursor tane = db.rawQuery("select i, name, age from person_table where i = ?;", new String[]{" " + plant3.r});
301     tane.moveToFirst();
302     int a = tane.getInt(2);
303     int b = a - 1;
304     ContentValues values = new ContentValues();
305     values.put("age", b);
306     db.update("person_table", values, "i = " + plant3.r, null);
307     tane.close();
308 }
309
310 check = 0;
311 taneid = -1;
312
313 }else{
314     final AlertDialog.Builder builder3 = new AlertDialog.Builder(activity);
315     builder3.setMessage("種が足りないよ！");
316     builder3.setPositiveButton("OK", new DialogInterface.OnClickListener(){
317         public void onClick(DialogInterface dialog, int which){
318             checkbox1.setChecked(false);
319             checkbox2.setChecked(false);
320             checkbox3.setChecked(false);
321             //koukaon.playSe();
322         }
323     });
324     builder3.create();
325     builder3.show();
326 }
327
328 //shopボタン
329 public void doAction3(View view){
330     //koukaon.playSe();
331     //shopデータベース呼び出し
332     Cursor cursor = db.rawQuery("select i, name, ne from shop_table where i >= ?;", new String[]{"3"});
333     boolean isEof2 = cursor.moveToFirst();
334     AlertDialog.Builder builder2 = new AlertDialog.Builder(this);
335     builder2.setTitle("shop");

```



```

336 final CharSequence[] items5 = new CharSequence[cursor.getCount()];
337 final int[] items9 = new int[cursor.getCount()];
338 final int[] items10 = new int[cursor.getCount()];
339 for (int n = 0; !isEof2; n++) {
340     items5[n] = String.format("%s : %d円", cursor.getString(1), cursor.getInt(2));
341     items9[n] = cursor.getInt(0);
342     items10[n] = cursor.getInt(2);
343     isEof2 = cursor.moveToNext();
344 }
345 cursor.close();
346
347 //購入処理
348 builder2.setSingleChoiceItems(items5, -1, new DialogInterface.OnClickListener(){
349
350     @Override
351     public void onClick(DialogInterface dialog, int which){
352         //koukaon.playSe();
353         //shopのID、金額準備
354         shopid = items9[which];
355         shopnedan = items10[which];
356         //所持金データベース呼び出し
357         Cursor c3 = db.rawQuery("select name, kane from money_table where kane >= ?", new String[]{"0"});
358         c3.moveToFirst();
359         shozikin = c3.getInt(1);
360         c3.close();
361     }
362 });
363
364 //shopでOKを押したとき
365 builder2.setNeutralButton("OK", new DialogInterface.OnClickListener(){
366     public void onClick(DialogInterface dialog, int which){
367         //koukaon.playSe();
368
369         //買えるとき
370         if (shozikin >= shopnedan){
371             layoutmoney.removeView(tvmoney);
372             LinearLayout.LayoutParams params1 = new LinearLayout.LayoutParams
373                 (ViewGroup.LayoutParams.WRAP_CONTENT, ViewGroup.LayoutParams.WRAP_CONTENT);
374             params1.setMargins(400, 80, 100, 100);
375
376             //タネを増やす
377             String al = String.format("select i, name, age from person_table where i = %d", shopid);
378             Cursor c = db.rawQuery(al, null);
379             c.moveToFirst();
380             ContentValues values4 = new ContentValues();
381             values4.put("age", 1);
382             db.update("person_table", values4, "i = " + shopid, null);
383             c.close();
384
385             //所持金を減らす
386             Cursor c3 = db.rawQuery("select name, kane from money_table where kane >= ?", new String[]{"0"});
387             c3.moveToFirst();
388             int k4 = shozikin - shopnedan;
389             ContentValues values5 = new ContentValues();
390             values5.put("kane", k4);
391             db.update("money_table", values5, "name = '所持金'", null);
392             c3.close();
393
394             //所持金再描画
395             Cursor c4 = db.rawQuery("select name, kane from money_table where kane >= ?", new String[]{"0"});
396             c4.moveToFirst();
397             final CharSequence[] items11 = new CharSequence[c4.getCount()];
398             items11[0] = String.format("%s : %d円", c4.getString(0), c4.getInt(1));
399             c4.close();
400
401             tvmoney.setText(items11[0]);
402

```

```

403 //shopから買われたアイテム削除
404 String a2 = String.format("select i, name, ne from shop_table where i = %d", shopid);
405 Cursor cursor = db.rawQuery(a2, null);
406 cursor.moveToFirst();
407 db.delete("shop_table", "i = " + shopid, null);
408 cursor.close();
409
410 layoutmoney.addView(tvmoney, params1);
411
412 shopid = -1;
413
414 //買えないとき
415 }else {
416     final AlertDialog.Builder builder3 = new AlertDialog.Builder(activity);
417     builder3.setMessage("お金が足りないよ!");
418     builder3.setPositiveButton("OK", new DialogInterface.OnClickListener(){
419         public void onClick(DialogInterface dialog, int which){
420             //koukaon.playSe();
421         }
422     });
423     builder3.create();
424     builder3.show();
425 }
426 });
427
428 //shopでもどるを押したとき
429 builder2.setNegativeButton("もどる", new DialogInterface.OnClickListener(){
430     public void onClick(DialogInterface dialog, int which){
431         //koukaon.playSe();
432     }
433 });
434 builder2.create();
435 builder2.show();
436 }
437
438 //収穫ボタン
439 public void doAction2(View view){
440     //koukaon.playSe();
441     //植物が収穫できる状態になった時
442     if(TitleActivity.datap == false ){
443         if(uetaswitch1 == 1){
444             if(plant1.status == 3){
445                 seichoutime = 0;
446                 plant1.age1 = 0;
447
448                 Cursor cursor1 = db.rawQuery("select i, name, age from person_table where i = ?;", new String[]{"" + plant1.r});
449                 cursor1.moveToFirst();
450                 Cursor cursor2 = db.rawQuery("select i, name, kosu, kakaku from kajitu_table where i = ?;", new String[]{"" + plant1.r});
451                 cursor2.moveToFirst();
452
453                 //タネの数を増やす
454                 int a = cursor1.getInt(2);
455                 int b = a + 1;
456                 ContentValues values = new ContentValues();
457                 values.put("age", b);
458                 db.update("person_table", values, "i = " + plant1.r, null);
459
460                 //果実の数を増やす
461                 int c = cursor2.getInt(2);
462                 int d = c + 1;
463                 ContentValues values1 = new ContentValues();
464                 values1.put("kosu", d);
465                 db.update("kajitu_table", values1, "i = " + plant1.r, null);
466
467
468

```

```

469     plant1.status = 0;
470
471     TitleActivity.datap = true;
472     pref = PreferenceManager.getDefaultSharedPreferences(this);
473     Editor edit = pref.edit();
474     edit.remove("pa");
475     edit.remove("pr");
476     edit.remove("GAZ04");
477
478     edit.putBoolean("dp", TitleActivity.datap);
479     edit.commit();
480
481     cursor1.close();
482     cursor2.close();
483
484     layoutyasai1.removeView(plant1);
485
486     plant1.timer.cancel();
487     plant1 = null;
488     plant2 = null;
489     plant3 = null;
490     uetaswitch1 = 0;
491 }
492
493
494 if(uetaswitch2 == 1){
495     if(plant2.status == 3){
496
497         seichoutime = 0;
498         plant2.age1 = 0;
499
500         Cursor cursor1 = db.rawQuery("select i, name, age from person_table where i = ?;", new String[]{" " + plant2.r});
501         cursor1.moveToFirst();
502         Cursor cursor2 = db.rawQuery("select i, name, kosu, kakaku from kajitu_table where i = ?;", new String[]{" " + plant2.r});
503         cursor2.moveToFirst();
504
505         //タネの数を増やす
506         int a = cursor1.getInt(2);
507         int b = a + 1;
508         ContentValues values = new ContentValues();
509         values.put("age", b);
510         db.update("person_table", values, "i = " + plant2.r, null);
511
512         //果実の数を増やす
513         int c = cursor2.getInt(2);
514         int d = c + 1;
515         ContentValues values1 = new ContentValues();
516         values1.put("kosu", d);
517         db.update("kajitu_table", values1, "i = " + plant2.r, null);
518
519         plant2.status = 0;
520
521         TitleActivity.datap = true;
522         pref = PreferenceManager.getDefaultSharedPreferences(this);
523         Editor edit = pref.edit();
524         edit.remove("pa");
525         edit.remove("pr");
526         edit.remove("GAZ04");
527
528         edit.putBoolean("dp", TitleActivity.datap);
529         edit.commit();
530
531         cursor1.close();
532         cursor2.close();
533
534         layoutyasai2.removeView(plant2);
535

```

```

536         plant2.timer.cancel();
537         plant2 = null;
538         uetaswitch2 = 0;
539     }
540 }
541
542 if(uetaswitch3 == 1){
543     if(plant3.status == 3){
544         seichoutime = 0;
545         plant3.age1 = 0;
546
547         Cursor cursor1 = db.rawQuery("select i, name, age from person_table where i = ?;", new String[]{"" + plant3.r});
548         cursor1.moveToFirst();
549         Cursor cursor2 = db.rawQuery("select i, name, kosu, kakaku from kajitu_table where i = ?;", new String[]{"" + plant3.r});
550         cursor2.moveToFirst();
551
552         //タネの数を増やす
553         int a = cursor1.getInt(2);
554         int b = a + 1;
555         ContentValues values = new ContentValues();
556         values.put("age", b);
557         db.update("person_table", values, "i = " + plant3.r, null);
558
559         //果実の数を増やす
560         int c = cursor2.getInt(2);
561         int d = c + 1;
562         ContentValues values1 = new ContentValues();
563         values1.put("kosu", d);
564         db.update("kajitu_table", values1, "i = " + plant3.r, null);
565
566         plant3.status = 0;
567
568         TitleActivity.datap = true;
569         pref = PreferenceManager.getDefaultSharedPreferences(this);
570         Editor edit = pref.edit();
571         edit.remove("pa");
572         edit.remove("pr");
573         edit.remove("GAZ04");
574
575         edit.putBoolean("dp", TitleActivity.datap);
576         edit.commit();
577
578         cursor1.close();
579         cursor2.close();
580
581         layoutyasai3.removeView(plant3);
582
583         plant3.timer.cancel();
584         plant3 = null;
585         uetaswitch3 = 0;
586     }
587 }
588
589 else{
590     AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this);
591     alertDialogBuilder.setTitle("エラー");
592     alertDialogBuilder.setMessage("収穫できないよ！");
593     // アラートダイアログの肯定ボタンがクリックされた時に呼び出されるコールバックリスナーを登録します
594     alertDialogBuilder.setPositiveButton("OK",
595         new DialogInterface.OnClickListener() {
596             @Override
597             public void onClick(DialogInterface dialog, int which) {
598             }
599         });
600     alertDialogBuilder.setCancelable(true);
601     AlertDialog alertDialog = alertDialogBuilder.create();
602

```

```

603         alertDialog.show();
604     }
605 }
606
607 //アイテムボタン
608 public void doAction(View view){
609     koukaon.playSe();
610
611     //数量が1以上のタネを表示させる
612     Cursor cursor = db.rawQuery("select i, name, age from person_table where age >= ?;", new String[]{"1"});
613     boolean isEof = cursor.moveToFirst();
614
615     AlertDialog.Builder builder = new AlertDialog.Builder(this);
616     builder.setTitle("SEEDS");
617
618     final CharSequence[] items = new CharSequence[cursor.getCount()];
619     final int[] items2 = new int[cursor.getCount()];
620
621     for (int i = 0; isEof; i++) {
622         items[i] = String.format("%s : %d個", cursor.getString(1), cursor.getInt(2));
623         items2[i] = cursor.getInt(0);
624         isEof = cursor.moveToNext();
625     }
626
627     cursor.close();
628
629     //アイテムを選択したとき時
630     builder.setSingleChoiceItems(items, -1, new DialogInterface.OnClickListener(){
631         public void onClick(DialogInterface dialog, int which){
632             taneid = items2[which];
633             pref = PreferenceManager.getDefaultSharedPreferences(activity);
634             Editor edit = pref.edit();
635             edit.putInt("GAZ04", taneid);
636             edit.commit();
637         }
638     });
639
640     //収穫物ダイアログ
641     builder.setPositiveButton("収穫物", new DialogInterface.OnClickListener(){
642
643         //数量が1以上の果実を表示させる
644         @Override
645         public void onClick(DialogInterface dialog, int which){
646             //koukaon.playSe();
647
648             Cursor cursor = db.rawQuery("select i, name, kosu, kakaku from kajitu_table where kosu >= ?;", new String[]{"1"});
649             boolean isEof1 = cursor.moveToFirst();
650
651             final AlertDialog.Builder builder1 = new AlertDialog.Builder(activity);
652             builder1.setTitle("収穫物");
653
654             final CharSequence[] items4 = new CharSequence[cursor.getCount()];
655
656             int a;
657             for (a = 0; isEof1; a++) {
658                 items4[a] = String.format("%s : %d個 : %d円", cursor.getString(1), cursor.getInt(2), cursor.getInt(3));
659                 isEof1 = cursor.moveToNext();
660             }
661
662             cursor.close();
663
664             String kajitu = "";
665
666             //果実が0の時の表示
667             if(a == 0){
668
669

```

```

670     kajitu = "何もないよ";
671 //果実が1以上の時の表示
672
673     }else{
674         for(int i = 0; a > i; i++){
675             kajitu = kajitu + items4[i] + "\n";
676         }
677     }
678 builder1.setMessage(kajitu);
679
680 //収穫物でOKを押したとき
681 builder1.setNeutralButton("OK", new DialogInterface.OnClickListener(){
682     @Override
683     public void onClick(DialogInterface dialog, int which){
684         //koukaon.playSe();
685         layoutmoney.removeView(tvmoney);
686
687         Cursor cursor1 = db.rawQuery("select i, name, kosu, kakaku from kajitu_table where kosu >= ?;",new String[]{"1"});
688         boolean isEof1 = cursor1.moveToFirst();
689
690         int[] items5 = new int[cursor1.getCount()];
691         int[] items6 = new int[cursor1.getCount()];
692
693         //売却金額の合計
694         for (int m = 0; isEof1; m++) {
695             items5[m] = cursor1.getInt(2);
696             items6[m] = cursor1.getInt(3);
697             int t;
698             t = items5[m] * items6[m];
699             syukakugoukei = syukakugoukei + t;
700             t = 0;
701             isEof1 = cursor1.moveToNext();
702         }
703
704         cursor1.close();
705
706         LinearLayout.LayoutParams params1 = new LinearLayout.LayoutParams
707             (ViewGroup.LayoutParams.WRAP_CONTENT, ViewGroup.LayoutParams.WRAP_CONTENT);
708         params1.setMargins(400, 80, 100, 100);
709
710         //所持金に売却金額を足す
711         Cursor cursor2 = db.rawQuery("select name, kane from money_table where kane >= ?;",new String[]{"0"});
712         cursor2.moveToFirst();
713         int a = 0;
714         int b;
715
716         a = cursor2.getInt(1);
717         b = a + syukakugoukei;
718
719         ContentValues values2 = new ContentValues();
720         values2.put("kane", b);
721         db.update("money_table", values2, "name = '所持金'", null);
722
723         cursor2.close();
724
725         //所持金描画
726         Cursor cursor3 = db.rawQuery("select name, kane from money_table where kane >= ?;",new String[]{"0"});
727         cursor3.moveToFirst();
728         final CharSequence[] items8 = new CharSequence[cursor3.getCount()];
729
730         items8[0] = String.format("%s : %d円", cursor3.getString(0), cursor3.getInt(1));
731
732         cursor3.close();
733
734         tvmoney.setText(items8[0]);
735
736

```

```

737
738 //果实データベースの数を0にする
739 Cursor cursor4 = db.rawQuery("select i, name, kosu, kakaku from kajitu_table where kosu >= ?;", new String[]{"1"});
740 cursor4.moveToFirst();
741
742 ContentValues values3 = new ContentValues();
743 values3.put("kosu", 0);
744 db.update("kajitu_table", values3, null, null);
745
746 cursor4.close();
747
748 layoutmoney.addView(tvmoney, params1);
749
750 syukakugoukei = 0;
751 }
752 });
753
754 //収穫物でもどるを押したとき
755 builder1.setNegativeButton("もどる", new DialogInterface.OnClickListener(){
756 //アイテムダイアログ表示
757 @Override
758 public void onClick(DialogInterface dialog, int which){
759 //koukaon.playSe();
760 doAction(null);
761 }
762 });
763 builder1.create();
764 builder1.show();
765 }
766 });
767 });
768
769 //アイテムダイアログOKボタン
770 builder.setNeutralButton("OK", new DialogInterface.OnClickListener(){
771 //植えるボタンとチェックボックスを表示させる
772 @Override
773 public void onClick(DialogInterface dialog, int which){
774 findViewById(R.id.uerubutton).setVisibility(View.VISIBLE);
775 findViewById(R.id.checkBox1).setVisibility(View.VISIBLE);
776 findViewById(R.id.checkBox2).setVisibility(View.VISIBLE);
777 findViewById(R.id.checkBox3).setVisibility(View.VISIBLE);
778 }
779 });
780 });
781
782 //アイテムダイアログでもどるを押したとき
783 builder.setNegativeButton("もどる", new DialogInterface.OnClickListener(){
784 public void onClick(DialogInterface dialog, int which){
785 //koukaon.playSe();
786 }
787 });
788 builder.create();
789 builder.show();
790 }
791 }
792 }
793 }

```

## <2> Plant.java(プラントクラス)

```
1 package gudon.sample.sqlitel;
2
3 import java.util.TimerTask;
4 import java.util.Timer;
5 import android.os.Handler;
6 import android.content.Context;
7 import android.content.res.Resources;
8 import android.database.Cursor;
9 import android.database.sqlite.SQLiteDatabase;
10 import android.graphics.*;
11 import android.graphics.drawable.Drawable;
12 import android.util.AttributeSet;
13 import android.view.View;
14 import android.util.Log;
15
16 //植物描画
17 public class Plant extends View {
18     private Context context;
19     public Drawable drawable;
20
21     public static Timer timer;
22     public int age1 = 0;
23     public int r;
24
25     int[] growsparam = {2, 1, 2, 2};
26     int[] growsparam1 = {4, 2, 4, 4};
27     public int status;
28
29     public int[] items3;
30     Handler mHandler = new Handler();
31
32     SQLiteDatabase db;
33
34     public Plant(final Context context, SQLiteDatabase db0, int r0) {
35         super(context);
36         this.context = context;
37         db = db0;
38         r = r0;
39         status = 1;
40
41         String a1 = String.format("select i, w, name, syo from gazou_table where i = %d", r);
42         Cursor c = db.rawQuery(a1, null);
43
44         boolean isEof = c.moveToFirst();
45         items3 = new int[3];
46
47         for (int l1 = 0; isEof; l1++) {
48             items3[l1] = c.getInt(3);
49             isEof = c.moveToNext();
50         }
51
52         Resources resources = context.getResources();
53         drawable = resources.getDrawable(items3[1]);
54
55         invalidate();
56         c.close();
57
58         timer = new Timer(true);
59
60         age1 = age1 + SQLiteSample1.seichoutime;
61
62         timer.schedule(new TimerTask(){
63             @Override
64             public void run() {
65
66                 mHandler.post(new Runnable() {
67                     public void run() {
```



```

68         Log.v("age1", "" + age1);
69         if (age1 > growtparam[r]) {
70             Resources resources = context.getResources();
71             drawable = resources.getDrawable(items3[1]);
72             status = 2;
73             invalidate();
74         }
75
76         if (age1 > growtparam1[r]) {
77             Resources resources = context.getResources();
78             drawable = resources.getDrawable(items3[2]);
79             status = 3;
80             invalidate();
81         }
82         age1 ++; //植えてからの経過時間
83     }
84 }
85 }
86 }
87 }
88 }, 1000, 1000);
89
90 }
91
92
93 public Plant(final Context context, AttributeSet attrs){
94     super(context, attrs);
95     this.context = context;
96 }
97
98 public void setImage(int n){
99     Resources resources = context.getResources();
100     drawable = resources.getDrawable(n);
101 }
102
103 public void setImage2(int a, int b){
104     r = a;
105     age1 = b;
106
107     String al = String.format("select i, w, name, syo from gazou_table where i = %d", r);
108     Cursor c = db.rawQuery(al, null);
109
110     boolean isEof = c.moveToFirst();
111
112     items3 = new int[3];
113
114     for (int i1 = 0; isEof; i1++) {
115         items3[i1] = c.getInt(3);
116         isEof = c.moveToNext();
117     }
118
119     Resources resources = context.getResources();
120     drawable = resources.getDrawable(items3[0]);
121
122     invalidate();
123
124     c.close();
125 }
126
127
128 @Override
129 protected void onDraw(Canvas c){
130     super.onDraw(c);
131     //c.drawColor(Color.BLACK);
132     if (drawable != null){
133         drawable.setBounds(10,10,100,100);
134         drawable.draw(c);
135     }
136
137 }
138

```

### <3> TitleActivity.java(タイトル画面)

```
1 package gudon.sample.sqlite1;
2
3 import android.app.Activity;
4 import android.content.Intent;
5 import android.content.SharedPreferences;
6 import android.database.sqlite.SQLiteDatabase;
7 import android.os.Bundle;
8 import android.preference.PreferenceManager;
9 import android.view.MotionEvent;
10 import android.view.View;
11 import android.view.ViewGroup;
12 import android.widget.LinearLayout;
13 import android.widget.TextView;
14
15 public class TitleActivity extends Activity{
16     private final int WC = ViewGroup.LayoutParams.WRAP_CONTENT;
17     private LinearLayout layout1;
18     private TextView tv1;
19     public static boolean datap = true;
20
21     PersonOpenHelper helper;
22     SQLiteDatabase db;
23
24     @Override
25     public void onCreate(Bundle savedInstanceState){
26         super.onCreate(savedInstanceState);
27         setContentView(R.layout.title7);
28
29         layout1 = new LinearLayout(this);
30         this.addContentView(layout1, new ViewGroup.LayoutParams(WC, WC));
31
32         LinearLayout.LayoutParams params1 = new LinearLayout.LayoutParams
33             (ViewGroup.LayoutParams.WRAP_CONTENT, ViewGroup.LayoutParams.WRAP_CONTENT);
34         params1.setMargins(270, 80, 100, 100);
35
36         tv1 = new TextView(this);
37         tv1.setText("タイトル画面");
38
39         layout1.addView(tv1, params1);
40
41         helper = new PersonOpenHelper(this);
42         db = helper.getReadableDatabase();
43     }
44
45     public void onPause(){
46         super.onPause();
47         this.finish();
48     }
49
50     //はじめからボタン
51     public void doAction4(View view){
52         Intent selectIntent = new Intent(this, SQLiteSample1.class);
53         startActivity(selectIntent);
54         db.execSQL("drop table if exists person_table");
55         db.execSQL("drop table if exists gazou_table");
56         db.execSQL("drop table if exists kajitu_table");
57         db.execSQL("drop table if exists shop_table");
58         db.execSQL("drop table if exists money_table");
59         helper.onCreate(db);
60     }
61
62     //つづきからボタン
63     public void doAction5(View view){
64         Intent selectIntent = new Intent(this, SQLiteSample1.class);
65         SharedPreferences pref = PreferenceManager.getDefaultSharedPreferences(this);
66         datap = pref.getBoolean("GAZ03", true);
67         startActivity(selectIntent);
68     }
69 }
```

### <4> music.java(BGM クラス)

```
1 package gudon.sample.sqlite1;
2
3 import android.media.MediaPlayer;
4 import android.view.View;
5 import android.content.Context;
6
7 //BGM
8 public class music extends View{
9     MediaPlayer bgm = MediaPlayer.create(getContext(), R.raw.yasashi);
10
11     public music(Context context) {
12         super(context);
13         bgm.setLooping(true);
14         bgm.start();
15     }
16
17     public void pause() {
18         bgm.pause();
19     }
20
21     public void reStartsound(){
22         bgm.start();
23     }
24
25 }
```

## <5> SePlayer.java(SE クラス)

```
1 package gudon.sample.sqlite1;
2
3 import android.content.Context;
4 import android.media.AudioManager;
5 import android.media.SoundPool;
6
7 //効果音
8 public class SePlayer
9 {
10     private SoundPool soundPool;
11
12     private int koukaon;
13
14
15     public SePlayer(Context context)
16     {
17         // new SoundPool(読み込むファイル数,読み込む種類,読み込む質)
18         this.soundPool = new SoundPool(1, AudioManager.STREAM_MUSIC, 0);
19
20         // load(コンテキスト,読み込むリソースID,音の優先度)
21         this.koukaon = soundPool.load(context, R.raw.kettei, 1);
22     }
23
24     public void playSe()
25     {
26         // play(再生するサウンドID,左のボリューム,右のボリューム,優先度,ループ回数(0はしない、-1は無限),再生レート)
27         soundPool.play(koukaon, 1.0f, 1.0f, 1, 0, 1.0f);
28     }
29 }
```

## <6> PersonOpenHelper.java(データベース)

```
1 package gudon.sample.sqlite1;
2
3 import android.content.Context;
4 import android.database.sqlite.SQLiteDatabase;
5 import android.database.sqlite.SQLiteOpenHelper;
6
7
8 public class PersonOpenHelper extends SQLiteOpenHelper {
9     final static private int DB_VERSION = 1;
10
11     public PersonOpenHelper(Context context) {
12         super(context, "mydata.db", null, DB_VERSION);
13     }
14
15     @Override
16     public void onCreate(SQLiteDatabase db) {
17
18         // 種のテーブル
19         db.execSQL(
20             "create table person_table(i text, name text not null, age text);");
21     };
22     //表の行の挿入
23     db.execSQL("insert into person_table(i,name,age) values (0, 'トマト', 2);");
24     db.execSQL("insert into person_table(i,name,age) values (1, 'きゅうり', 1);");
25     db.execSQL("insert into person_table(i,name,age) values (2, 'ピーマン', 1);");
26     db.execSQL("insert into person_table(i,name,age) values (3, 'かぼちゃ', 1);");
27     db.execSQL("insert into person_table(i,name,age) values (4, 'ニンジン', 0);");
28     db.execSQL("insert into person_table(i,name,age) values (5, 'ハクサイ', 0);");
29
30
31     //画像のテーブル
32     db.execSQL(
33         "create table gazou_table(i text, w text, name text not null, syo int);");
34     };
35     //表の行の挿入
36     db.execSQL(String.format("insert into gazou_table(i, w, name,syo) values (0, 1, 'トマト1', %d);", R.drawable.i1));
37     db.execSQL(String.format("insert into gazou_table(i, w, name,syo) values (0, 2, 'トマト2', %d);", R.drawable.i2));
38     db.execSQL(String.format("insert into gazou_table(i, w, name,syo) values (0, 3, 'トマト3', %d);", R.drawable.i3));
39     db.execSQL(String.format("insert into gazou_table(i, w, name,syo) values (1, 1, 'きゅうり1', %d);", R.drawable.i4));
40     db.execSQL(String.format("insert into gazou_table(i, w, name,syo) values (1, 2, 'きゅうり2', %d);", R.drawable.i5));
41     db.execSQL(String.format("insert into gazou_table(i, w, name,syo) values (1, 3, 'きゅうり3', %d);", R.drawable.i6));
42     db.execSQL(String.format("insert into gazou_table(i, w, name,syo) values (2, 1, 'ピーマン1', %d);", R.drawable.i7));
43     db.execSQL(String.format("insert into gazou_table(i, w, name,syo) values (2, 2, 'ピーマン2', %d);", R.drawable.i8));
44     db.execSQL(String.format("insert into gazou_table(i, w, name,syo) values (2, 3, 'ピーマン3', %d);", R.drawable.i9));
45     db.execSQL(String.format("insert into gazou_table(i, w, name,syo) values (3, 1, 'かぼちゃ1', %d);", R.drawable.i10));
46     db.execSQL(String.format("insert into gazou_table(i, w, name,syo) values (3, 2, 'かぼちゃ2', %d);", R.drawable.i11));
47     db.execSQL(String.format("insert into gazou_table(i, w, name,syo) values (3, 3, 'かぼちゃ3', %d);", R.drawable.i12));
48
49
50     //実のテーブル
51     db.execSQL(
52         "create table kajitu_table(i text, name text not null, kosu text, kakaku text);");
53     };
54     //表の行の挿入
55     db.execSQL("insert into kajitu_table(i,name,kosu,kakaku) values (0, 'トマト', 0, 1000);");
56     db.execSQL("insert into kajitu_table(i,name,kosu,kakaku) values (1, 'きゅうり', 0, 200);");
```

```

57 db.execSQL("insert into kajitu_table(i,name,kosu,kakaku) values (2, 'ヒーマン', 0, 300);");
58 db.execSQL("insert into kajitu_table(i,name,kosu,kakaku) values (3, 'ダイコン', 0, 400);");
59 db.execSQL("insert into kajitu_table(i,name,kosu,kakaku) values (4, 'ニンジン', 0, 600);");
60 db.execSQL("insert into kajitu_table(i,name,kosu,kakaku) values (5, 'ハクサイ', 0, 800);");
61
62 // 店のテーブル
63 db.execSQL(
64     "create table shop_table(i text, name text not null, ne text);"
65 );
66 //表の行の挿入
67 db.execSQL("insert into shop_table(i,name,ne) values (3, 'ダイコン', 1000);");
68 db.execSQL("insert into shop_table(i,name,ne) values (4, 'ニンジン', 1300);");
69 db.execSQL("insert into shop_table(i,name,ne) values (5, 'ハクサイ', 1500);");
70
71
72 // 所持金のテーブル
73 db.execSQL(
74     "create table money_table(name text not null, kane text);"
75 );
76 //表の行の挿入
77 db.execSQL("insert into money_table(name,kane) values ('所持金', 0);");
78
79 }
80
81
82 @Override
83 public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
84     // データベースの変更が生じた場合は、ここに処理を記述する。
85 }
86 }

```

## <7> activity\_splite\_sample11.xml(メイン画面レイアウト)

```

1 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2     xmlns:tools="http://schemas.android.com/tools"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:paddingBottom="@dimen/activity_vertical_margin"
6     android:paddingLeft="@dimen/activity_horizontal_margin"
7     android:paddingRight="@dimen/activity_horizontal_margin"
8     android:paddingTop="@dimen/activity_vertical_margin"
9     tools:context=".SQLiteSample1"
10    android:background="@drawable/haikai2" >
11
12    <CheckBox
13        android:id="@+id/checkbox3"
14        android:layout_width="wrap_content"
15        android:layout_height="wrap_content"
16        android:layout_above="@+id/button2"
17        android:layout_marginBottom="781dp"
18        android:layout_toRightOf="@+id/button"
19        android:visibility="invisible"
20        android:text="@string/checkbox3" />
21
22    <CheckBox
23        android:id="@+id/checkbox1"
24        android:layout_width="wrap_content"
25        android:layout_height="wrap_content"
26        android:layout_above="@+id/checkbox3"
27        android:layout_alignLeft="@+id/button"
28        android:layout_marginBottom="67dp"
29        android:layout_marginLeft="14dp"
30        android:visibility="invisible"
31        android:text="@string/checkbox1" />
32
33    <CheckBox
34        android:id="@+id/checkbox4"
35        android:layout_width="wrap_content"
36        android:layout_height="wrap_content"
37        android:layout_above="@+id/button"
38        android:layout_alignLeft="@+id/checkbox1"
39        android:layout_marginBottom="64dp"
40        android:visibility="invisible"
41        android:text="@string/checkbox4" />
42
43    <CheckBox
44        android:id="@+id/checkbox2"
45        android:layout_width="wrap_content"
46        android:layout_height="wrap_content"
47        android:layout_alignBaseline="@+id/checkbox1"
48        android:layout_alignBottom="@+id/checkbox1"
49        android:layout_alignLeft="@+id/checkbox5"
50        android:visibility="invisible"
51        android:text="@string/checkbox2" />
52
53    <CheckBox
54        android:id="@+id/checkbox5"
55        android:layout_width="wrap_content"
56        android:layout_height="wrap_content"
57        android:layout_alignBaseline="@+id/checkbox4"
58        android:layout_alignBottom="@+id/checkbox4"
59        android:layout_alignRight="@+id/button2"
60        android:layout_marginRight="14dp"
61        android:text="@string/checkbox5"
62        android:visibility="invisible" />
63
64    <Button
65        android:id="@+id/button3"

```

```

86     android:layout_width="wrap_content"
87     android:layout_height="wrap_content"
88     android:layout_alignParentRight="true"
89     android:layout_alignTop="@+id/button2"
90     android:layout_marginRight="19dp"
91     android:onClick="doAction3"
92     android:text="@string/button3" />
93
94 <Button
95     android:id="@+id/uerubutton"
96     android:layout_width="wrap_content"
97     android:layout_height="wrap_content"
98     android:layout_above="@+id/checkbox2"
99     android:layout_alignRight="@+id/button2"
100    android:onClick="doAction4"
101    android:visibility="invisible"
102    android:text="@string/uerubutton"
103    />
104
105 <Button
106     android:id="@+id/button"
107     android:layout_width="wrap_content"
108     android:layout_height="wrap_content"
109     android:layout_below="@+id/checkbox3"
110     android:layout_toLeftOf="@+id/button2"
111     android:onClick="doAction"
112     android:text="@string/button" />
113
114 <Button
115     android:id="@+id/button2"
116     android:layout_width="wrap_content"
117     android:layout_height="wrap_content"
118     android:layout_alignParentBottom="true"
119     android:layout_marginBottom="40dp"
120     android:layout_marginRight="14dp"
121     android:layout_toLeftOf="@+id/button3"
122     android:onClick="doAction2"
123     android:text="@string/button2" />
124
125 </RelativeLayout>

```

## <8> title1.xml(タイトル画面)

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:orientation="vertical" >
6
7     <Button
8         android:id="@+id/button2"
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:layout_alignLeft="@+id/button"
12        android:layout_alignTop="@+id/button"
13        android:layout_marginTop="72dp"
14        android:onClick="doAction5"
15        android:text="@string/button5" />
16
17     <Button
18         android:id="@+id/button"
19         android:layout_width="wrap_content"
20         android:layout_height="wrap_content"
21         android:layout_alignParentTop="true"
22         android:layout_centerHorizontal="true"
23         android:layout_marginTop="138dp"
24         android:onClick="doAction4"
25         android:text="@string/button4" />
26
27 </RelativeLayout>

```

## <9> string(オブジェクトの名称)

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3
4     <string name="app_name">samplesqlite1</string>
5     <string name="action_settings">Settings</string>
6     <string name="hello_world">Hello world!</string>
7     <string name="button">アイテム</string>
8     <string name="button2">収穫</string>
9     <string name="button3">shop</string>
10    <string name="button4">はじめから</string>
11    <string name="button5">つづきから</string>
12    <string name="checkbox1"></string>
13    <string name="checkbox2"></string>
14    <string name="checkbox3"></string>
15    <string name="checkbox4"></string>
16    <string name="checkbox5"></string>
17    <string name="uerubutton">植える</string>
18
19 </resources>

```

### 3-5. まとめ

このゲームは未完成である。そこで直すべきこと、追加すべきことを記載しておく。

- ・ アプリを動かしていくとメモリを圧迫してアプリケーションが止まってしまう
- ・ 複数の「timer」が動くと収穫の際にいくつかの「timer」が止まらない
- ・ デバイスによって植物のグラフィックの比率を変える
- ・ タイトルを考える
- ・ タイトル画面の背景グラフィックを挿入する
- ・ 最終目的を決める

以上の点の解決を目指して「市場公開」を目標に残りの学生生活を送っていきたい。

### 参考文献

- ・ 柴田望洋 (2007) 『明解 Java 入門編』 ソフトバンククリエイティブ株式会社
- ・ 掌田津耶乃 (2012) 『Android わくわくゲームプログラミング教室』 株式会社ラトルズ
- ・ 掌田津耶乃 (2013) 『Eclipse ではじめる Android プログラミング入門 SDK2.3/4.2 対応』 株式会社秀和システム
- ・ フランク・アブルソン、チャーリー・コリンズ、ロビ・セン (2010) 『コードからわかる Android プログラミングのしくみ』 日経 BP 社
- ・ Ed Burnette (2010) 『Hello,Android Introducing Google's Mobile Development Platform Third Edition』 Ptigmatic Bookshelf
- ・ (2013) 「これだけ知っていれば大丈夫! Android アプリ開発者のための Java 超入門」 『日経ソフトウェア』 2013 年 12 月号, 日経ソフトウェア
- ・ 「愚鈍人」 < <http://ichitcltk.hustle.ne.jp/gudon/> >(2013/11 アクセス)

## おわりに

PV制作・AR制作・ゲームアプリ制作を行うにあたり、映像編集やプログラミングなど、それぞれのチームで異なった作業ではあったものの、初挑戦のことばかりだった。2013年9月から卒業制作を始め、これまでの半年間は私たちにとって短い期間であったが、それぞれが懸命に取り組んだ結果を残すことができ嬉しく思う。

この報告書を見て、後輩たちには独創心溢れる制作に挑戦してもらいたい。

