

平成27年度

卒業論文

題目	Android アプリ 『memory up』
	-企画・開発及び評価-

担当教員 (自署)		印

学籍番号 201214069

氏名 鐘文聡

広島経済大学

要旨

私はゼミナールでは、現代で著しい成長を遂げているスマートフォンに着目し、android アプリ開発を行った。今流行っているアプリの現状、特徴、問題点を調べた上で、それに基づいて、自分は高齢者が楽しめて、人間の記憶力をアップできる「memory up」というアプリを開発することにしました。また、完成したアプリを周りの友達、同僚に使って評価してもらって、今後アプリを改善するために、その評価を踏まえて、今後このアプリ開発をもとにどう進化する根拠になります。

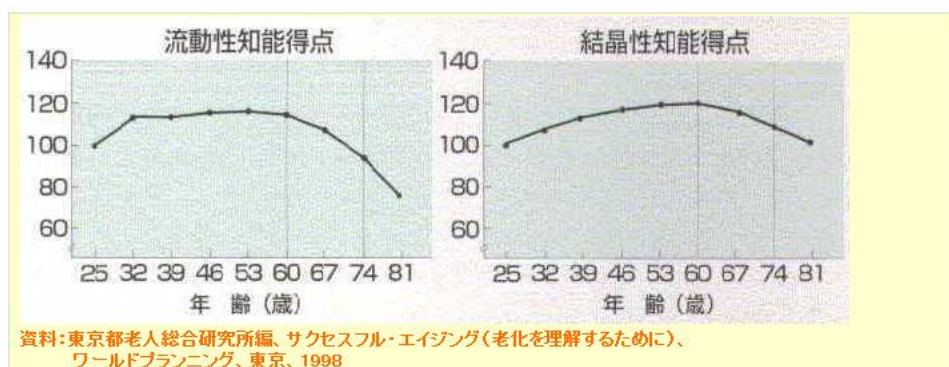
目次

第1章 はじめに.....	1
第2章 スマートフォンアプリの現状.....	4
第1節 脳トレアプリの動向.....	4
第2節 脳トレアプリの特徴.....	4
第1項 医学上に効果もある.....	5
第3節 脳トレアプリの課題.....	5
第3章 脳トレアプリの計画.....	7
第1節 脳トレアプリの企画.....	7
第1項 開発工程.....	8
第4章 脳トレアプリ開発の設計と仕様.....	10
第5章 実装.....	10
第1節 開発環境.....	10
第2節 開発手法.....	12
第3節 開発の流れ.....	14
第4節 プログラムの説明.....	17
第5節 実行問題の対策.....	28
第6節 実行結果.....	29
第6章 完成アプリの評価.....	32
第1節 アンケート内容.....	32
第2節 集計図.....	33
第7章 終わり.....	34
第1節 アプリ改善の提案.....	34
第2節 Android アプリ開発の心構え.....	35
参考文献.....	36

第1章 はじめに

脳の機能は青年期を過ぎると加齢とともに低下します。これはごく普通の生活をしていても体力や筋力が年々低下するのと同じです。しかし、体力や筋力は毎日の運動習慣で低下を防ぐ、もしくは向上させることができます。つまり体力は鍛えることができるのです。そして私たちの脳についてもこれらと同じだということがわかってきています。毎日、積極的に脳を使う習慣をつけることによって、脳の機能の低下を防ぐことができるのです。最近物忘れが多くなったと感じて、言いたい言葉がなかなか出てこないなど、思い当たることはありませんか？ 脳を鍛えることによって、こういった症状が改善される可能性があります。また、お子様や若い方も、脳を鍛えることによって創造力や記憶力を高め、我慢強くなるという効果が期待できます。

図6 知能の生涯発達曲線



[1]

流動性知能：新しいものを学習したり覚えたりするような、経験の影響を受けることが少ない、むしろ生まれながらもっている能力に左右される知能をいいます。この能力は30歳代にピークに達したあと60歳ごろまでは維持されます。そしてそれ以降は急速に低下していきます。このように流動性知能が老年期以降低下することは、加齢に伴う脳機能変化と関連したもので、いわば正常な老化性変化と思われます。

結晶性知能：一般的知識や判断力、理解力などで過去に習得した知識や経験をもとにして日常生活の状況に対処する能力です。この能力は、60歳ごろまで徐々に上昇し、その後は緩やかに低下していきます。しかし、結晶性知能は、70歳、80歳になればな

だらかに低下するもののそのレベルは20歳代に近い能力が維持されています。このことは高齢になっても何かを学び習得することが十分可能であることを示しています。

現代社会にスマートフォンの普及率が高まっていることが著しいです、スマートフォンの急速に普及することによって、多様なアプリケーションに開発されて、利用されています、そして多様な世代に利用されています。ただし、全てのユーザーがスマートフォンを持つことが前提であるため、高齢者でも遊べる、楽しめると感じる魅力が重要である。

今ネット上、様々な脳トレアプリケーションを開発されて、公開されています。例えば、アメリカ発の「脳トレ」ゲーム「Lumosity」とは、世界で7,000万人の利用者を超えるというモンスター級の「脳トレ」ゲームがこの度日本進出を果たした^[2]。2007年よりアメリカのWebサービスとしてスタートしたもので、その後スマートフォンアプリへと展開した。『Lumosity』とは、あなたの脳力を「記憶力」「注意力」「スピード」「柔軟性」「問題解決能力」の5項目に沿って測り、そして、脳力分析する機能付き、かんたんなゲームでそれらを鍛えていくトレーニングアプリケーションです。若者に対しては、確かに人気があって、それぞれに分野においてゲームが用意されて、楽しむことができます、そして、その成績を分析した結果を踏まえて、自分脳の苦手な分野を他の部分に合わせて、得意な部分を伸ばしていくのが特徴です。要求仕様は『Lumosity』対応 OS:iOS 7.0 以降/Android 1.0.213 以降、バージョン:iOS 3.6/Android 4.0、開発:Lumos Labs, Inc. 価格:無料(バージョンと価格は記事掲載時のものです)。

しかし、高齢者に対しては、利用する時、まずその最初の複雑さに悩まされる人が多いだと思っております、次は、ゲーム内容は英語を初めていろんな国の言語を使っています、母国語しか分からない人達はゲームの楽しさを楽しむことができなくなります。

これを解決するためには、私たち自身は、ある程度脳トレについて理解すること、高齢者や子供にとって負担にならないゲームアプリを考えること、これらの二点が重要だと考えられます。従って解決策を立証するため、第1章は、脳トレアプリの現状、特徴、問題点について書き、理解してもらいます。第2章は、脳トレアプリを開発する考案、現状アプ

り問題点の対策を説明します。第 3 章は、アプリを開発するために、どのような設計と仕様を説明します。第 4 章は、アプリを実装して、実行中に起きた問題点をどのように改善したのを説明します。第 5 章は、完成したアプリをタブレットで実行し、それについての評価を説明します、最後は、アプリについて、どうすればもっと使いやすく、脳を活性化できるのを改善します。

それで、ゼミの中で、注目した分野のアプリケーションを取り上げて、それを研究し、そして自分の考案を出して、Eclipse（「イクリプス」または「エクリプス」）は、IBM によって開発された統合開発環境（IDE）の一つ。高機能ながらオープンソースであり、Java をはじめとするいくつかの言語に対応する。Eclipse 自体は Java で記述されている^[3]。それを利用して高齢者でも遊べる、楽しめるアプリケーションを開発するのが一番重要だと考えます、ただ単純的に使うのでは無く、人間の脳力をアップすることができ、シンプルなアプリケーションを開発するのが目的です。

第2章 スマートフォンアプリの現状

第1節 脳トレアプリの動向

スマホアプリ開発者にとって、今後の市場規模の動向は気になるところです。スマートフォンは急激に普及しつつあり、あっという間に誰でも持っているようになりました。アプリの市場も急激に拡大しているように見えるけれど、実際の所はどうでしょうか。このまま同じように開発を続けていていいのでしょうかこの市場動向の予測から言うと、まだまだアプリ開発は大きくなります。

これから新しいアプリ開発をしてもいいし、今あるアプリをもっとバージョンアップして、ずっと使ってくれるユーザーの満足度を上げることが課題と考えられます。ただ、ユーザーの嗜好の変化に合わせて開発アプリの型を変えていくのは、動きの激しい業界で残っていくためにはどうしても必要なことなので、簡単な手続きでユーザーが手に入れるアプリを開発するのも重要と考えられます。そして、スマートフォンの急激な普及に伴うアプリケーション市場の急拡大ですが、現在 iPhone 系ではアプリ内課金型が、Android 系では無料アプリが主流になっています。しかし機種や OS に依存しない Web 型のアプリも増えていくと考えられるので、まだまだ参入の余地も拡大の余地もありそうです

第2節 脳トレアプリの特徴

市場に流行っているアプリはほとんど認知科学に基づく 20 種類以上のゲーム、それぞれのゲームで、観察力、記憶力、判断力、直感力、計算力、反応力を 6 つの力が鍛えられる項目が設置しています。まず、「スピード」「記憶力」「注意力」「柔軟性」「問題解決能力」の項目で、特にどの部分の能力を鍛えたいか選びます。そしてトレーニングスタートです。どれもシンプルな問題だからと気楽にはじめたつもりが、ハイスコアを取りたくて気が付くと夢中になって、普段使っていない脳が刺激されているのを実感しているうちに脳力がアップしていきます、それだけではなくて、様々な認知能力を鍛えるゲームをそろえてい

ます。新しいゲームが次々登場、飽きずにトレーニングいただけます。次は、いつでもどこでも気軽にトレーニングパソコン、スマホ、タブレットで空いた時間で気軽に頭脳ゲームを楽しんでいただけます。そして、トレーニングは続ける事がとても大事と考えられるので、ご希望のスケジュールに合わせてメールでお知らせします。最後は、他ユーザーと比較、認知能力分野ごとにあなたのスコアを同年代のユーザーと比較いただけます。脳トレアプリであなたの脳の可能性を見つけます。

第1項 医学上に効果もある

それだけではなくて、医学上にも効果があります、世界中に、70歳以上人口において2番目に多数を占める障害疾患である認知症、認知症とは後天的な脳の器質的障害により、いったん正常に発達した知能が不可逆的に低下した状態をいう^[4]、認知症の予防法は、原因によってさまざまですが、主な原因は加齢関連認知低下（Aging-associated Cognitive Decline : AACD）である。認知症の行動はもっともよくないのは「何もしないこと」と考えられます。認知症予防に効果があると考えられている食生活・運動・脳トレなどの方法認知症の仕組みはまだわかっていません。ただ、脳は使えば使うほど、良い刺激を与えれば与えるほど活性化し成長します。脳は何歳になっても成長することがわかっています。認知症の予防の為に脳は積極的に使っていければ、脳の活性化をできると考えられます。ほかに認知機能の低下が見られるような軽度の病気に患っている人でも、健常者においても、認知的課題やそれに伴う感情の制御、学業課題でのパフォーマンスの向上が見られ、その効果は、他の単純なゲーム課題などよりも効果的である。

第3節 脳トレアプリの課題

PC時代にユーザー登録といえばメールアドレスの登録が必須です。例えば、FacebookもGoogleもほぼすべてのPCサービスではサインアップする時にメールアドレスが求められています。今回はスマホ時代にもアプリを利用する時にこの仕組みがどう変わっていくの

か課題があると思います。まず、メールアドレスを登録するとき、メールアドレスは個人のものという前提が必要であることが知っていると思います。このユーザー登録は単純にアプリを使いたい人にとって、めんどくさい気持ちを与えてしまう、それで、このアプリを利用する人数もだいぶ減ると思います。そして一般的にユーザー登録する手順は仮登録→メールで認証→本登録完了の流れはセキュリティ的にも安全だろうという心配があって、チャレンジしてみたい人はやめてしまう可能性が高いと思います。次はスマホ時代にPC時代とちがって、みなさんも経験済みだと思うのですがスマホでメールアドレス入力には本当にめんどくさいでしょう。それだけでアプリを削除したくなる気持ちがあるのでしょう。そういえば、最近ではソーシャルアプリやゲームアプリはユーザーログイン機能付きのアプリが多々見ますが、しかも結構な数のアプリがこのサインアップ時にエラーが出て、アプリを使うところまでたどり着きません。せっかくダウンロードしたのに使えずに始末になる課題、それらはまだ進化する必要があると考えられます。

一方でログインなしのアプリもたくさんあるのでそれらと比べた時に圧倒的に有利です。なぜなのか、ログインしないと使えるアプリはもはやなくなっていくのだと思います。スマホアプリではデバイス ID というものを内部で作ってそれを 1 ユーザーのユニークな ID として利用することが多いです。このデバイス ID がこれまでのメールアドレスの代わりになるわけです。ユーザーが何も入力しなくてもモバイル端末自動認識という特性がこれを可能にしています。そうすると、アプリを利用したい人がすぐ使えて、楽しむことができます。ただし一つ問題があります。機種変更をして、複数デバイスを使いこなしている人を一人のユーザーとして引き継げないという問題が発生します。そうすると基本はユーザー登録なしでアプリは使えます、ただ複数デバイスを引き継ぎたいとき、個人メールアドレスが登録という形式に変わると思います。ちなみに、スマホ時代にアプリを利用する時に面倒な手続きがなくて、一人のユーザーとして複数のデバイスで引き継げられるという問題を解決するのが今後の課題になると考えられます。

第3章 脳トレアプリの計画

第1節 脳トレアプリの企画

どんなアプリでも、開発・維持管理を行っていくと思うと高いモチベーションが必要になります。そのためにはアプリ開発をする目的・動機・目標を確認しておく必要があります。明確な動機や目的・目標を持って開発を進めると、開発途中で難解な問題があっても頑張っ て立ち向かえると思います。自分は高齢者でも遊べる、楽しめるアプリケーションを開発するのが一番重要だと考えます、ただ単純的に使うのではなくて、人間の脳力をアップすることができ、シンプルなアプリケーションを開発するのが目的です。

アプリ開発の目的が明確になったら、どんなアプリにするか紙に書いてアイデアを可視化することを先生に勧められました。トップ画面からメニューに移動したらこんなデザインで、各コンテンツのデザインや構成はこんな感じで、サイトマップはこんな感じで、とアプリに搭載するコンテンツを一度可視化してみると頭の中が整理され、無駄なページがないか等もチェックでき、無駄な作業を未然に防ぐことができるとアドバイスされました。企画する時に情報収集とアプリに搭載する機能を整理しました。

スマートフォンは急激に普及するとともに、アプリの開発と市場も急激に拡大していることが著しくて、あっという間に誰でも持っているようになった時代に、アプリの開発は氾濫性から指向性になると思います、指向性と簡単に言ったら、どんな人を対象にして、その対象のためにアプリを開発すること。

先述の通り、市場に流行っている脳トレアプリはそれぞれのゲームで、観察力、記憶力、判断力、直感力、計算力、反応力を6つの力が鍛えられる項目が設置しています。確かに種類が多くて、たくさん楽しむことができるのですが、年寄りや子どもにとって、複雑な感覚を与えてしまって、トレーニングの効果が良くなります。ですから、シンプルで項目が少ないアプリを開発するのを考えました。もう一つ理由は現在ほとんどのアプリはユーザー登録が必要です。それで、このアプリの利用者が限られてしまい、インストール数は伸びないのではないかといった予想もありました、そして、ただチャレンジしてみた

い人にとって、面倒でインストールしない結果になります。つまり、高齢者でも遊べる、楽しめるアプリケーションを開発するのが一番重要な点は、人間の脳力をアップする効果があり、ユーザー登録なくて、シンプルなアプリケーションを開発することと思います。

第1項 開発工程

基本計画はシステム開発の最初の工程になります。基本計画の工程では主に以下のようなことを行います。

<ul style="list-style-type: none">ユーザーの現状の業務について調査分析を行い、新システムを策定してシステム化計画書にまとめます。
<ul style="list-style-type: none">プロジェクト実行計画をまとめ、システムに必要なコンピュータの種類・規模、データベースの大きさ、スケジュール（開発期間）などの見積もりを行い、開発計画書をまとめます。
<ul style="list-style-type: none">システムに対する要求をお客さんから聞いて、要求仕様書にまとめます。

要求分析はアプリ製品を作るにあたっての最初のタスクは要求を引き出す・集めることである。顧客はアプリに何をさせたいのかを知っているものである。しかし、その要求は不完全だったり、曖昧だったり、互いに矛盾していたりするかもしれません。ゼミの中でただ要求の担い手の立場から思ったことでこのアプリを開発することに決めましたわけ。

その要求は以下にまとめました：

① シンプルであること。	② ユーザー登録なしであること。
③ ゲーム機能が単一であること。	④文字が大きいであること。
⑤音が大きいであること。	⑥画面が明るいであること。

その要求を解決するためにいちいち分析すると、まずは①番に対して、まずインターフェー

スの画面が簡単で読みやすくすること、それで利用者に嫌な感じをさせなくなる。②番はユーザー登録なしであることはアプリを起動するとすぐにプレイできる仕組みにすれば、利用者の好奇心がすぐに落ちません。③番は脳トレゲームの種類が少なくして、一つのアプリはただ一種類のゲームを楽しむことができる機能にすると、複雑な感じがなくなり、集中して楽しみながら脳をトレーニングすると考えました。④⑥は文字が大きくて、明るくすると、読みやすくなります。⑤はゲームも音が大きくすると、より楽しむ効果を上げられると考えられます。

第4章 脳トレアプリ開発の設計と仕様

実際のアプリを開発する時、以上の要求要件に基づいて、アプリの仕様が要求されるのです。要求仕様（ようきゅうしよう、Requirements Specification）とは、工学分野において特定の製品やサービスがどうあるべきかを記述する文書を指す。主にシステム工学とソフトウェア工学で使われる用語である。英語の requirement からリクワイアメント（リクワイアメント）ともいう^[5]。

上述の要求要件に対して、私はゼミの中でアプリを開発した際に、以下の要求仕様を考えました。まず、アプリをどのような OS を運用するのか決めなければなりません。ゼミは主に高機能ながらオープンソースであり、Java をはじめとするいくつかの言語に対応する Eclipse を利用して勉強、開発されているので、もちろん Android の OS を使うこととなります。シンプルにするためには、アプリを起動すると、ゲームを始められる TOP 画面に決めました。この画面は複雑な設定はしなくて、GO というボタンをタッチすれば、楽しめることにします、このボタンは TOP 画面の一番上に配置しました。ボタンの名前は「GO」に変えました。スタートという意味です。このようにすると、手間がかかるユーザー登録とクライアントがこのゲームに対する好奇心の低下を防ぐことも解決できると考えます。高齢者に対して、ゲームの機能が多すぎると、嫌な感覚を与えてしまうことに対して、機能は一つの目的にしました。記憶力をアップする脳トレゲームに決めました。人間は年を取ればとるほど、視力、聴力が弱くなってしまふことを考えらえるはずです。そのため開発する時文字の大きさ、音量の大きさ普通より一倍、二倍の大きさに設定すると決定しました。各イメージボタンにかわいくて、面白いアイコンを設定しました。そして、ゲームの楽しさを増やすために、音のほうは三種類の効果音も使うとか、タイム制限も加えました。

第5章 実装

第1節 開発環境

最初に、Android アプリを作るために必要な開発環境を整える必要があります。アンド

ロイドアプリは JAVA で書かれているので、開発には JAVA アプリケーションの開発環境が必要となってきます。Google 社の開発者情報によると、Eclipse (エクリプス) という開発環境に Android 用の SDK (ソフトウェア開発キット) を入れることによる Android アプリの開発環境を構築することが推奨されています。

Android アプリの開発環境を整えるためには、以下のものを PC にインストールしなければならないです。

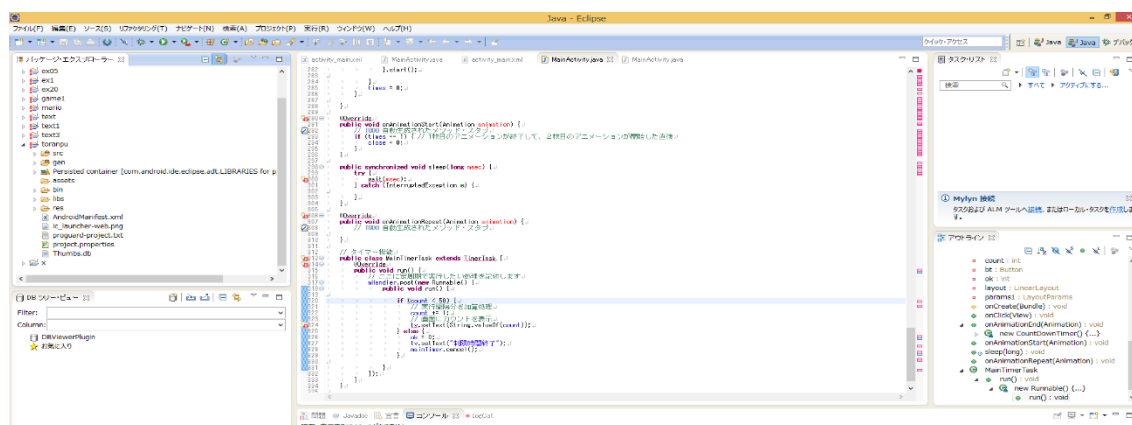
1. JDK (Java Development Kit) をインストールする
2. Android SDK をインストールする
3. Eclipse を導入する
4. Eclipse に Android 用のプラグイン (ADT) をインストール
5. Eclipse に AVD を作成する

ゼミの中で Java アプリケーションを開発するのは統合開発環境 (IDE) (図のように) を使って作業しました。統合開発環境とはアプリケーションの開発をサポートする環境で、Eclipse はオープンソースの統合開発環境 (IDE) (図のように) で、このソフトを使ってプログラムを開発していきます。この連載では多くの現場で使われている Eclipse (エクリプス) を選んでいます。今回は Java の開発環境を整え、サンプルのアプリケーションを作成し、それを実行することを目標としています。

Java で開発する前に、JDK (Java Development Kit) と呼ばれる Java のアプリケーションの開発ツールの集まりをインストールする必要があります。JDK は Java 言語で書かれたソースファイル (処理の記述がかかれたテキストファイル) をコンパイルするなど Java アプリケーションを作成する際に使うものです。JDK は、インストーラを使って簡単にインストールできます。また、この時に JDK のインストーラに含まれている JR (Java Runtime Environment) と呼ばれる環境も同時にインストールします。この JRE は Java アプリケーションを作成する機能は含まれておらず、実行する機能のみを持っているもの

になります。JREはJavaアプリケーションを実行するために使われます。そして、もし自分がタブレットのデバイスを持っているなら、Eclipseから自分が作ったアプリをダウンロードすることができます。よりリアルにテストしたり、体験したりすることができます。

次はAndroid SDKについて、アンドロイドアプリを開発するために必要なツールセット。アプリ開発、Androidのスクリーンショットなどが可能になります。プログラミング言語やAPIなどを利用してソフトウェアを開発する際に必要なツールのセットのことで、アンドロイドアプリを作る際に必要になってきます。注意点はAndroid SDKの画面が新しくなり、以前SDKをダウンロードしたのに、突然キャプチャできなくなったという人もいます。最新のバージョンをダウンロードしたら解決できると思っています。後はEclipseにADTを追加とAVDを作成も必要です。EclipseにADTを追加することによって、開発するに当たって、新しい機能を追加することができます。AVDはAndroidアプリをパソコン上で開発やデバックの際に必要なAndroidエミュレーターで、アプリを実行するのに必要な仮想デバイス、つまり、Androidアプリをパソコン上で擬似動作させるもの。これはインストールするのが強制ではないがもしAndroidのデバイスを持っていれば大丈夫だと思います。

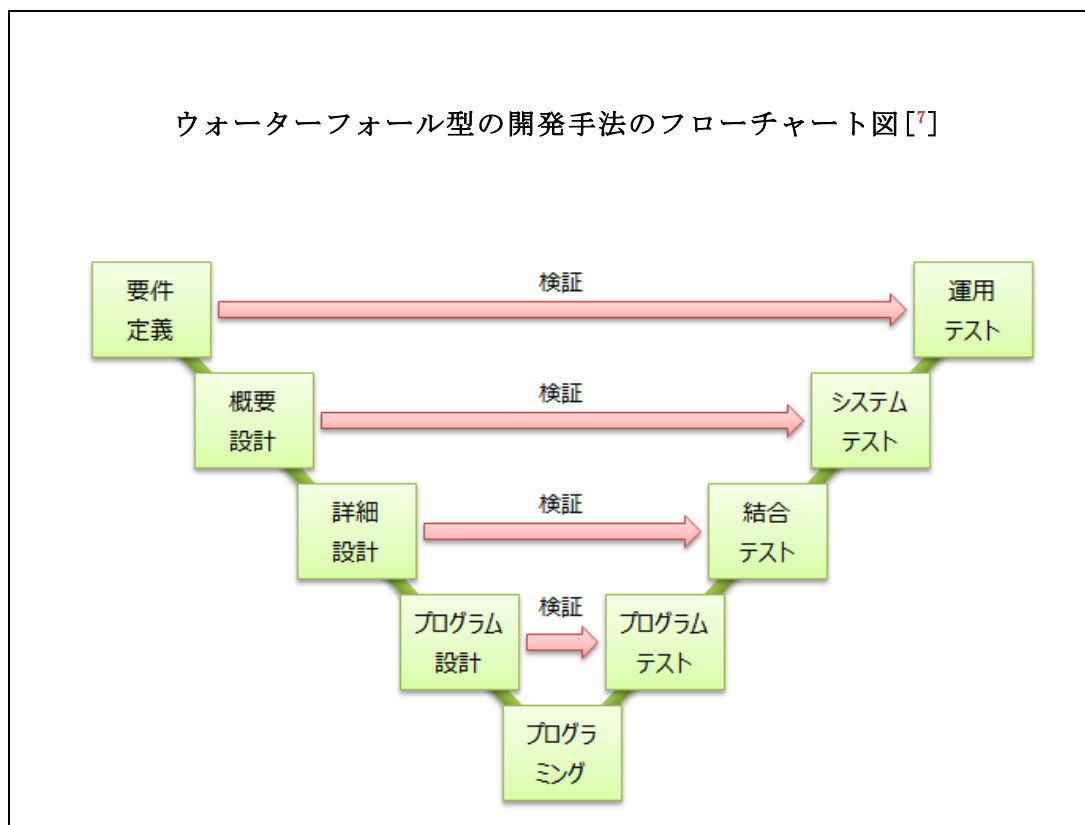


Eclipseで操作する画面

第2節 開発手法

開発手法と言ったら、よく使われるのは主にアジャイルソフトウェア開発型とウォーターフォール型です。アジャイルソフトウェア開発型は反復型開発からの派生手法である。

アジャイルでは、従来よりも軽量で人間中心の視点を導入した。アジャイルでは計画よりもフィードバックを重視する。フィードバックは主にテスト（評価）と開発途中のソフトウェアを外部にリリースすることで得られる。アジャイルソフトウェア開発は従来の方法論よりも効率的と思われる。少ない工数でより多くの機能を開発でき、品質の高いソフトウェアを開発できる。この開発手法はリリースを実現するのに時間と金銭をかかりますので、ゼミはやめました。より私たちが学期内で開発実現するのが易いのはウォーターフォール型です。ウォーターフォール型は最もよく知られた従来型の開発工程モデルはウォーターフォール・モデルである^[6]。このモデルでは、工程が上流から下流に順番に流れていくため、管理がしやすいです。開発者は上述の工程（局面、フェーズ）を順番に行う。要求仕様を作成し、それを分析し、解決法を設計し、そのためのソフトウェアフレームワークのアーキテクチャを作り、コードを書き、評価し（単体テスト→システムテストの順）、配備し、保守する。各工程が完了すると、次の工程に進むことができる。ゼミはこの開発手法を使って、毎週の授業で開発モデルに書いてある順番に行って、ソースコードを書いて、実行して、フィードバックして開発していました。利点はひとつの工程が終わるごとにその工程での成果が文書化されるので文書の管理がしやすいのが特徴です。つまり、システム開発を管理する立場の人たちにはすぐれた手法とすることができます。欠点は開発の最初の段階で仕様などの重要な決定を行うのがウォーターフォール型の特徴です。したがって後になってから仕様に変更が生じた場合など、前に決めたことを変更することが難しいということです。また、開発の最終段階にならないと実際に動作するシステムなど目に見えるものが出てこないため、ユーザーの意見を反映しにくいという欠点があります。



第3節 開発の流れ

Eclipse でアプリケーションを開発する手順を説明します。Eclipse でアプリケーション開発を行う際には、次の手順で実施します。

- a. プロジェクトを作成する
- b. プログラミングの作成
- c. Android エミュレーターでアプリケーションの動作確認

a. まず、プロジェクトの作成について

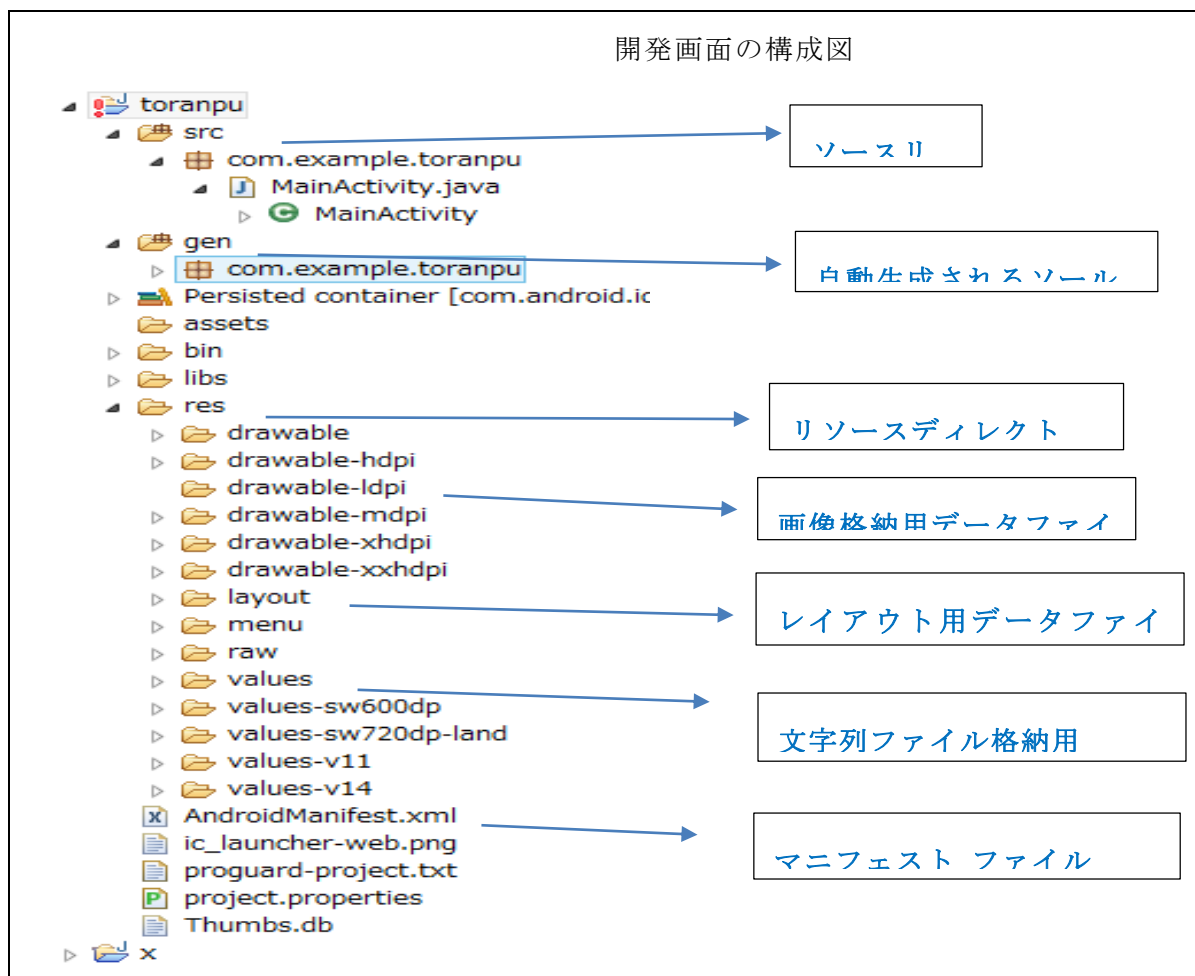
プロジェクトとは、1つのアプリケーションに使用される複数のプログラムをまとめて管理する単位のことです。まず、最初にプロジェクトを作成します。プロジェクトを作成する際には、プロジェクトの種類を選び、Eclipse の開発画面の構成やライブラリなどの開発環境を適切なものに設定する必要があります。ここで気を付けてほしいのは Android アプリケーションを開発するのが目的なので、プロジェクトの種類は「Android プロジェクト」を選びます。作成手順は「Eclipse を起動」→「ファイル」→「新規」→「プロジェクト」です。

b. 次はプログラミングの作成です

Android プロジェクトを作成すると、開発に必要なライブラリが設定されます。画面に表示するためのサンプルアプリケーションのプログラムや設定ファイル、ディレクトリ等も自動で生成されます。以下図のように)ここに上述したようなデータが「Eclipse のワークスペース」に追加している必要があります。主要なディレクトリを紹介します。

「src フォルダ」	作成するプログラムファイルを格納するフォルダです
「gen フォルダ」	自動生成されるソースファイルを格納するディレクトリです
「res フォルダ」	画像データ・各レイアウトとビュー設定ファイルなどの「リソース」が、サブディレクトリに種類ごとに入っています。
「anim 」	アニメーションを定義した XML ファイル
「color 」	カラーリストを定義した XML ファイル
「drawable 」	画像ファイルや形状などを定義した XML ファイル
「layout 」	レイアウトとビューを定義した XML ファイル
「menu 」	アプリケーションのメニューを定義した XML ファイル
「values 」	「文字列 (の配列)」・「ビュースタイル」・「値」を定義した各種の XML

Android アプリケーションを開発する際に中心となるのが、「src 」ディレクトリ、「res 」ディレクトリ、「Android Manifest.xml 」ファイルです。ここで注意してほしい点はファイルを作成するとき、いくつかのアプリの名前は同じにしてはいけないことです。



c. プリケーションの動作確認

作成したアプリケーションを正常に動作するかどうかがエミュレーターで確認します。

実機での確認でもよいのですが、エミュレーターでデバッグしてからの方が安全に行うことができます。ゼミの中で私はエミュレーターでプログラムの動作を確認しました。

Android エミュレーターが起動するには少し時間がかかります。無事に起動すると以下のロック画面が表示されます。ロックを解除すると、すぐにアプリケーションが起動されるかと思えます。



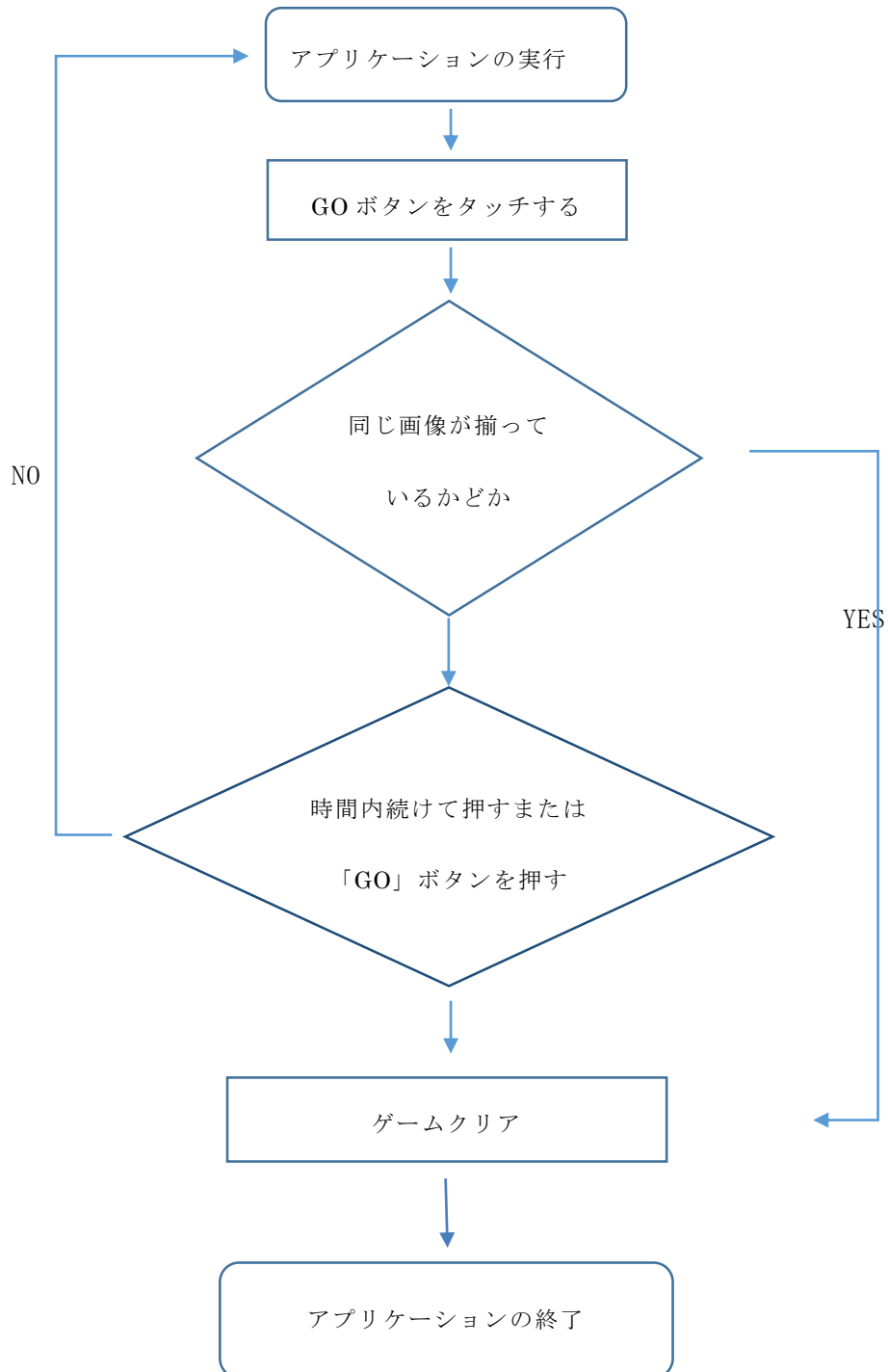
第4節 プログラムの説明

上述はプログラミングの大まかな流れと開発に必要な最低限の知識について説明しました。

実際に開発する時に、基礎のアプリ開発用語を理解しなければならないです。

アプリを実装する流れをフローチャート化にして、プログラミングする時によく使った用語を簡潔に説明しながら理解しやすくなると思っています。

アプリ実行のフローチャート図



「Memory UP」アプリケーションの処理の流れ

アクティビティ (Activity) は、ユーザーインターフェイス (画面) を提供し、イベント処理などユーザーの操作に応じた処理を持つオブジェクトです。また、アプリケーションは複数のアクティビティを持つことができます。プログラミングをする前に、以下ソースコードのように定義を宣言します。

```
public class MainActivity extends Activity implements OnClickListener,
    Animation Listener {
    private SoundPool sp;           //音を再生する機能の変数
    private ImageButton imbt[];    //画像を設定できるボタンの保存変数
    private LinearLayout ll[];     //画像全体のファイルを保存するための変数
    private int w, h;              //画像の幅と高さを入れる変数
    private int sound;             //効果音保存する変数
    private TranslateAnimation ta; //アニメーション
    private int selected, selected_1;
    private int[] cards_kind = new int[16];
```

```
private int[] cards_status = new int[16];
    private Random rnd;            //乱数を保存する変数
    private int r, times, kind, kind_1;
    private int close;
    // タイマー機能
    private Timer mainTimer;
    private MainTimerTask mainTimerTask;
```



```
private Handler mHandler;

private TextView tv;           //メッセージ表示用変数

private int count;           //数字を保存する変数

private Button bt;           //ボタン変数の宣言

private int ok;

private LinearLayout layout;  //画面全体の入れ物

private LinearLayout.LayoutParams params1;
```

以上は各データ変数の宣言です、それぞれの属性があります。上のソースコードによく使い、理解する必要がある用語の属性を説明します。まず、アクセス修飾子の `private` は同じクラス内のみアクセス可能ですが注意する点は Java クラスを一つの Java ファイルにまとめて定義することができますが、一つの Java ファイルには `public` 修飾子を記述したクラスは一つしか定義できません。ほかのアクセス修飾子は `protected` と `public` です、`protected` は同じクラス内、同じパッケージ、サブクラスからアクセス可能です。`public` は全てのクラスからアクセス可能です。今回はアクセス修飾子の `private` を使う理由はアプリケーションをアップロード・公開する時に情報をほかの人に見られないように保護する仕組みです。

画面全体レイアウトの作成ために、以下のソースコードように書きました。

```
@Override//メソッドに負荷する

//画面上で部品を配列するレイアウトオブジェクトを生成

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity_main);
```

```

//画面全体の入れ物レイアウトの作成

layout = new LinearLayout(this);

layout.setOrientation(LinearLayout.VERTICAL);

setContentView(layout);

//画面の幅と高さを取り出し、それぞれ W と H に入れる

WindowManager wm = (WindowManager) getSystemService(WINDOW_SERVICE);

Display disp = wm.getDefaultDisplay();

w = disp.getWidth();

h = disp.getHeight();

```

上のソースコードは自分がどのような画面を作成して、配列するのに利用します。MainActivity を配置した画面を表示するアクティビティは Activity implements を継承して作成する。そして、アプリケーション起動時に呼び出される onCreate メソッド内で MainActivity のレイアウトを設定した xml ファイルからレイアウトを作成す setOrientation メソッドでレイアウトの配置を設定されている。よく使われる用語は Override です。Override は Java SE5 から追加されたアノテーション型です、前に @ をつけてメソッドの前行に宣言することで、そのメソッドがオーバーライドメソッドであることをコンパイラに通知してくれます。Android アプリケーション開発の場合、onCreate メソッドのように Activity クラスなどの既存のクラスから継承してオーバーライドしなければいけないメソッドがいくつもあるため、記述ミスを防ぐためにも「@Override」アノテーションが必要です。

作成したレイアウトに対して細かく設計するために、以下のソースコードように考えました。

```
//ボタンの作成
bt = new Button(this);
bt.setText("GO"); //ボタン名前の変更
bt.setOnClickListener(this); //ボタンに対してクリック事件をチェックする
layout.addView(bt);
```

```
//メッセージ表示窓の作成
tv = new TextView(this);
layout.addView(tv);
```

```
//音声源を入れる
sp = new SoundPool(1, AudioManager.STREAM_MUSIC, 0);
sound = sp.load(this, R.raw.bp, 1);
//幅と高さをそれぞれの面の幅の 1/4 に設定するパラメータを作成
params1 = new LinearLayout.LayoutParams(w / 4, w / 4);
imbt = new ImageButton[16];
ll = new LinearLayout[4];
```

上述は画面全体に対して、具体的に設計するプログラミングです。例えば、ボタン `bt` の位置・高さ・デザインとかの細かく作成します、また、メッセージが出るタイミングと音声の流すタイミング、種類を作成します。ここでアプリを開発する経緯の中で重要な部分だと思っているのは全体画面を 16 等分にすることです。すると、アプリの最初の形ができていると見えると思っています。

割り当てる入れ物にイメージボタンを挿入するために、以下のようなソースコードに書きました。

```
// レイアウトの中に LL という入れ物を縦に個入れるための FOR 文
```

```
for (int i = 0; i < 4; i++) {  
    ll[i] = new LinearLayout(this);
```

```
//LL の中にボタンを横に個入れるための FOR 文
```

```
for (int j = 0; j < 4; j++) {  
    imbt[i * 4 + j] = new ImageButton(this);    //イメージボタン IMBT 「番号」  
    の作成  
    imbt[i * 4 + j].setLayoutParams(params1);    // ボタンのサイズを params  
    imbt[i * 4 + j].setBackgroundResource(R.drawable.mo); //入れ物の背景  
    ll[i].addView(imbt[i * 4 + j]);                //入れ物 LL にボタンを入れる  
    imbt[i * 4 + j].setOnClickListener(this);    //クリックされたときのリスナ  
    ー  
    layout.addView(ll[i]);                          //レイアウトに LL 入れ物を入れ  
    る
```

どのボタン位置にあるのかをチェックして位置を変数 space に入れる

```
for (int i = 0; i < 16; i++) {  
    cards_kind[i] = -1;                            // カードの絵柄。-1 はまだ絵柄な  
    し  
    cards_status[i] = 0;                          // 0:まだそろってないカード、1:そ  
    ろったカード }  
    rnd = new Random();                            // 乱数発生機能を作成 }  
    public void onClick(View v) {  
        if (v == bt) {  
            // タイマー機能
```

```

mainTimer = new Timer();

mainTimerTask = new MainTimerTask();

mainTimer.schedule(mainTimerTask, 0, 1000); // 0 秒後にスタートして、1000
ミリ秒ごとに時間をチェック

mHandler = new Handler();

count = 0;

```

上述は作成した layout 入れ物に乱数変数で画像を割り当て、それぞれの欄に配列されます。そして、それぞれのイメージボタンがタッチされることに対して、タイマーのチェック機能を入れたことです。ここで使った FOR 文は指定した回数だけ処理を繰り返す時に使われる構文です。for のよる繰り返しが始まるにあたり乱数で配列された絵柄が揃ったら 1 にします、揃ってないなら 0 にします。自分はタイマー機能の 1000 ミリ秒ごとにチェックすることにしましたが、自由に変更してもいいです。

ここはイメージボタンに対して、アニメーションの機能を追加するために、以下のようにプログラミングしました。

```

@Override

public void onAnimationEnd(Animation animation) {

Log.d("VarDebug", "data=" + cards_kind[selected]);

// アニメーション終了時に、もともと割り当てられた絵柄を表示する

imbt[selected].setBackgroundResource(R.drawable.animation);

Drawable drawableFrame = ((AnimationDrawable) imbt[selected]

.setBackground()).getFrame(cards_kind[selected]);

```

```

imbt[selected].setBackgroundDrawable(drawableFrame);

imbt[selected].postInvalidate();

// 2枚目のカードのアニメーション終了時に、そろってなかったらもとに戻す
処理

times++;

if (times == 1) {

kind_1 = kind;

selected_1 = selected;

} else {

close = 1;

if (kind_1 == kind) {

cards_status[selected_1] = 1;

cards_status[selected] = 1;

} else {

new CountdownTimer(1000, 1000) ;

@Override

public void onTick(long millisUntilFinished) {

}

@Override

public void onFinish() {

Log.d("VarDebug", "selected_1=" + selected_1);

imbt[selected_1]

.setBackgroundResource(R.drawable.animation);

```

```

Drawable drawableFrame1 = ((AnimationDrawable) imbt[selected_1]
    .getBackground()).getFrame(8);
imbt[selected_1].setBackgroundDrawable(drawableFrame1);
imbt[selected_1].postInvalidate();

imbt[selected]
    .setBackgroundResource(R.drawable.animation);

Drawable drawableFrame2 = ((AnimationDrawable) imbt[selected]
    .getBackground()).getFrame(8);
imbt[selected].setBackgroundDrawable(drawableFrame2);
imbt[selected].postInvalidate();
}.start();

times = 0;
}
    
```

上のソースコードは一枚目のカードはタッチすると、そのままのように表示します。二枚目のカードをタッチしたら、二つの処理が行われます。まず、もし同じ絵柄のカードだったらそのままずっと表示します。次は、二枚目の絵柄カードは揃ってなかったら、もとに戻す処理します。

最後はエラーをチェックする機能とタイマー機能は循環行えるために、以下のソースコードに書きました。

```
@Override

public void onAnimationStart(Animation animation) {

    if (times == 1) { // 1枚目のアニメーションが終了して、2枚目のアニメーションが開始した直後

        close = 0;

    }

    public synchronized void sleep(long msec) {

        try {

            wait(msec);

        } catch (InterruptedException e) {

        }

    }

    @Override

    public void onAnimationRepeat(Animation animation) {

    }

    // タイマー機能

    public class MainTimerTask extends TimerTask {

        @Override

        public void run() {

            // ここに定周期で実行したい処理を記述します

            mHandler.post(new Runnable() {

                public void run() {

                    if (count < 10) {

                        // 実行間隔分を加算処理

                        count += 1;

                    }

                }

            });

        }

    }

}
```



```
// 画面にカウントを表示  
tv.setText(String.valueOf(count));  
} else {  
    ok = 0;  
    tv.setText("制限時間終了");  
    mainTimer.cancel();  
}
```

上述によく使う try~catch~finally 文は、例外 (Exception) が発生することを事前に予測して、発生した時の処理を記述しておくときに使用する文法です。まず、try~catch の間に例外が発生する可能性がある処理を記述し、catch に発生した例外ごとの処理を記述します。finally に書いた処理は、例外が発生してもしなくても必ず実行されます。そして、ゲームを制限するタイマー時間の設定処理も加えました。

第5節 実行問題の対策

最初、私がアプリを開発した際に、さまざまな問題が出てきました。よく出る問題を説明します。例えば、以下のように乱数、音声の定義のソースコードです。

```
private Random rnd;    乱数の定義  
private int sound;    音声の定義
```

自分がよく間違えたのは入力ミスです。上のソースコードの一文字を間違えると、実行することができなくなります。対策はそれを修正すれば、うまく行けるのです。ほかに、下のソースコードの最後に” } ”がないと、エラーにもなります。ですから、

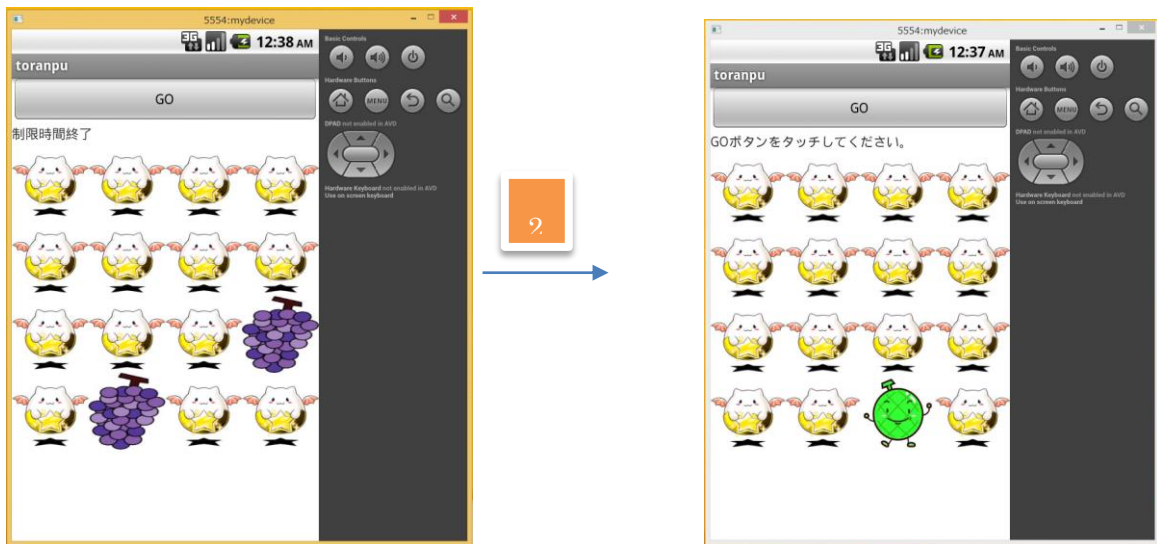
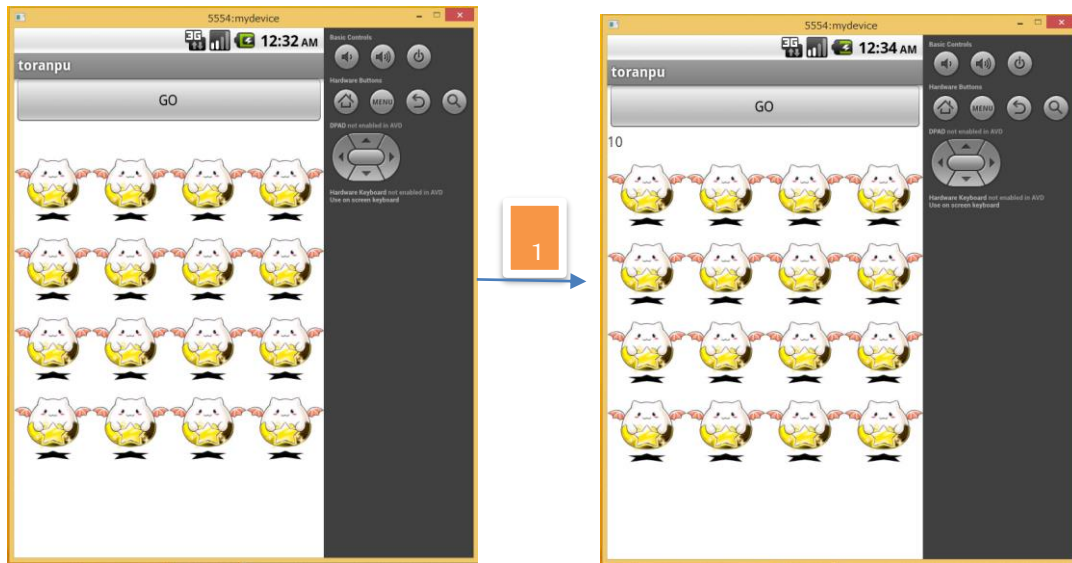
最初に打つ時に先に {} を入力する習慣をつけると大切だと思っています。

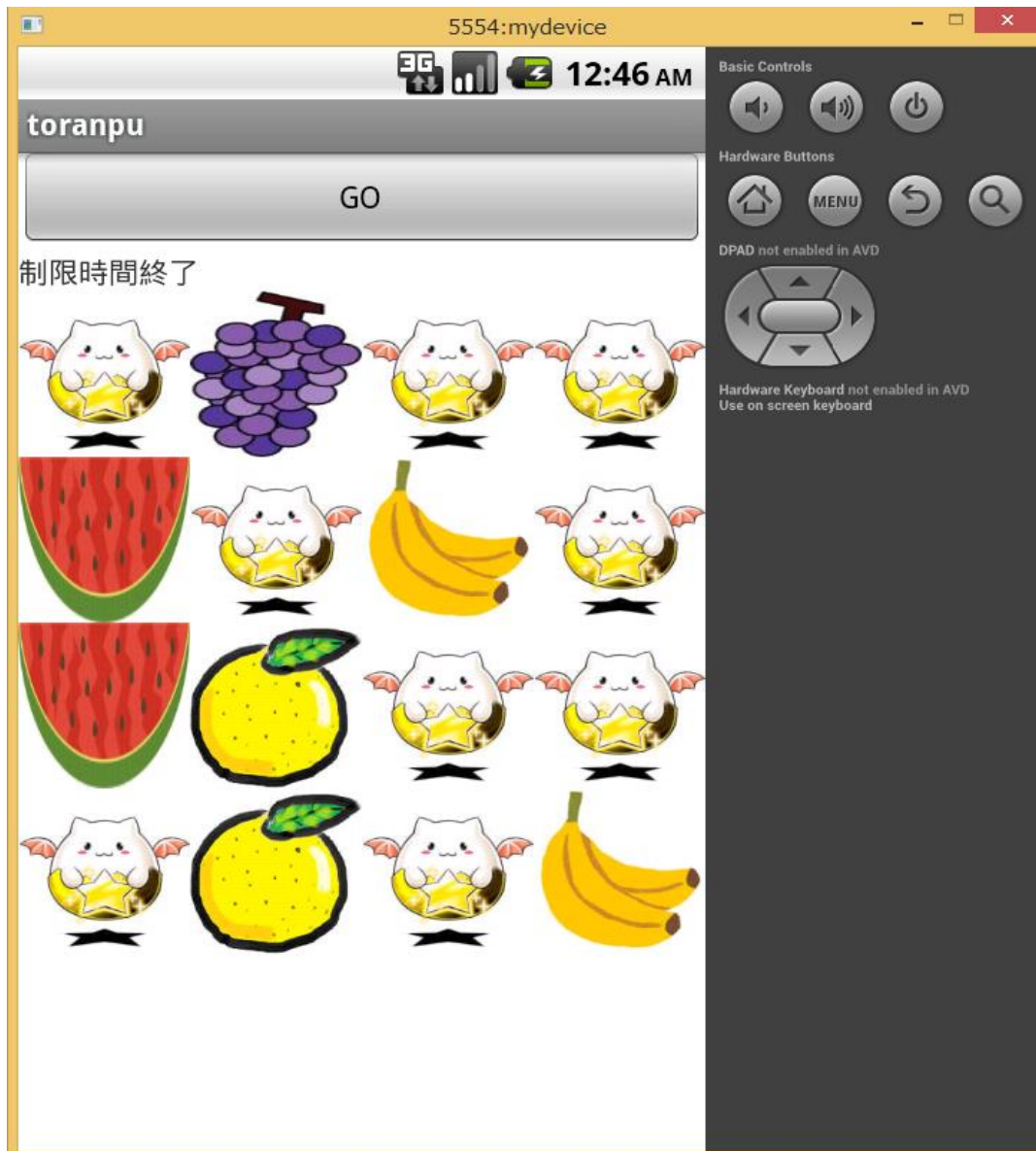
```
for (int i = 0; i < 16; i++) {  
    cards_kind[i] = -1; // カードの絵柄。-1 はまだ絵柄なし  
    cards_status[i] = 0; // 0:まだそろってないカード、1:そ  
    ろったカード} ;
```

以上のエラーは一般的です。それだけではなくて、アプリを開発する際に、自分がなかなか気付くことができない問題もありました。アプリを開発して以来、ソースコードが積み重ねて、その量が非常に多くなりました。そして、実行したり、修正したりすることによって、知らず知らずのうちに大量ソースコードの中に必要がないコードが含まれています。必要がないソースコードが大量になると、エラーが出るところを見つけるのが大変し、総容量が多くなるし、実行する速度が遅くなりました。対策は実行しながら、ソースコードを一行ずつ確認して、要らない部分を削除しました。結果として、総容量が減りまして、処理速度が以前より良くなりました。

第6節 実行結果

以下が作成したアプリケーションの実行結果です。作成したアプリケーションは、Android SDK に付属しているエミュレーターで実行しています。





エミュレーターで実行結果図

第6章 完成アプリの評価

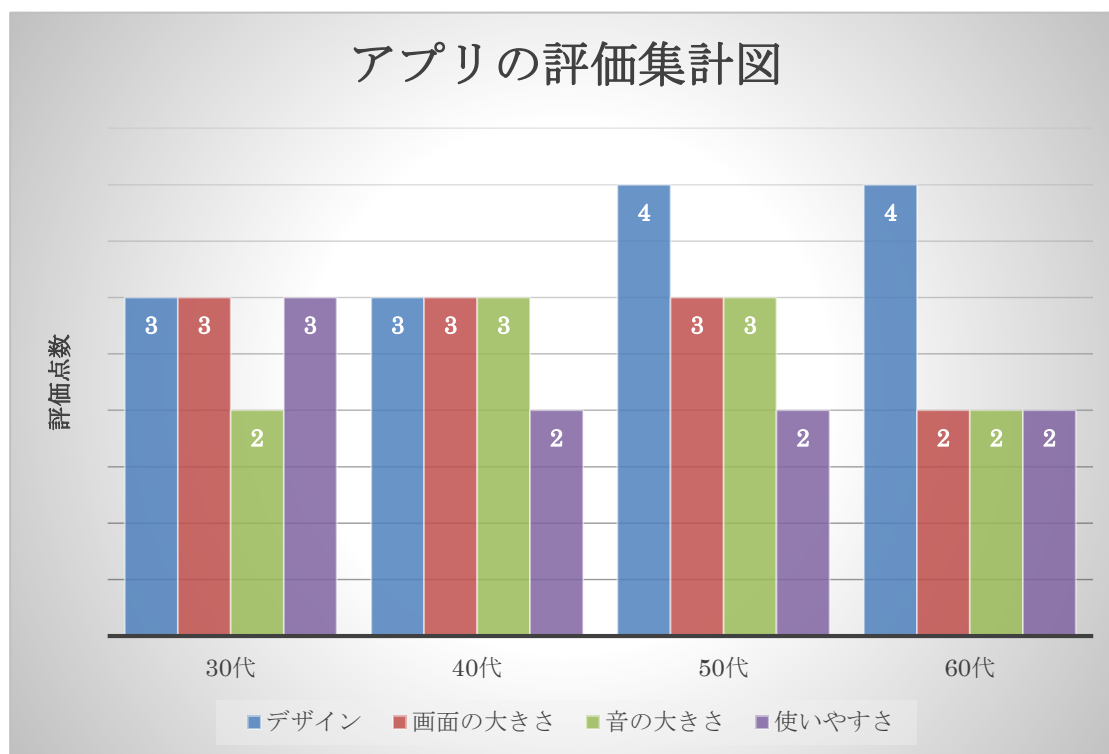
第1節 アンケート内容

アプリの評価について、ゼミはアンケートの形式で行うことになりました。私は年寄の人が遊べる、楽しめるアプリケーションを開発するのが一番重要だと考えます、ただ単純的に使うのでは無くて、人間の脳力をアップすることができ、シンプルなアプリケーションを開発するのが目的です。ですから、アンケートをする対象は主に30代から60代までの人にしました、学校で若者が多くて、アプリの結果を出せないと思って、学外に行いました。自分バイト先で行いました、対象人数は20人ぐらいです、アンケートを行った結果は以下の集計図のように、

年齢 \ 評価	デザイン	画面大きさ	音大きさ	使いやすさ
30代	2	3	2	2
40代	3	3	2	3
50代	4	2	3	3
60代	4	2	2	3

評価点数は 1点 2点 3点 4点 5点 (5点満点)

第2節 集計図



アプリ評価の結果からみると、やはりこのアプリは若い世代の人にとってあまり気に入らないと判断できます。まず、デザインからみると、各世代はいい点数を評価しました。画面の大きさは点数が50代まで大丈夫ですが、60代になるとだんだん下がっています。音のほうは年寄にとって、やや小さいと判断しますが若い者は同じ点数で評価した理由はたぶんもっと大きな音でより楽しむことができると思っています、最後、重視するところはアプリの使いやすさです。各世代の評点は低かったです、今後からの主に改善する方向着手するのが大切だと考えています。

第7章 終わり

第1節 アプリ改善の提案

今後、アプリの改善についていくつかの点からまとめて説明します。まず、利用する人の観点から見ればアンケートを行った結果からみると、一番足りない点は使いやすさでした。この点はやはりアプリを開発する人にとって、重視する必要があるところだと思っています。今後は、使いやすさ以外の点も少しずつ改善しながら、主にやって行こうと思っています。また、自分の考えからみると、まだまだ足りない部分があります。自分が改善したいところもいくつかの点から説明します。現代社会に android、iPhone などのスマートフォン端末の普及に伴って、アプリ利用者のマーケットもぐんぐん広がっています。そんな中で今後、android アプリの開発はますます注目を集めるであろうと予測されています。 android アプリは開発に使用するソフトが無償公開されていること、Windows 上でも開発可能であることなどの理由で、一般的に初心者でも開発しやすいと言われています。今回は、android アプリを開発するソフトを利用しましたが今後は iPhone アプリ開発ソフトを使って同じアプリ開発するつもりと考えています。そして、今回のアンケートを行ったのですが、自分が予想している結果と違いました。アンケートする対象がまだまだ増やしたほうが良いと思っています、これからは、アンドロイドマーケットでアプリケーションを登録する為には、開発者アカウントを取得して Google play に公開登録して、より客観性の評価を得られると考えています。後はアプリ自体に対して、今後も現在の機能の上に、アプリを使用していると、デバイスの傾けると画面表示も回転するように、タブレットには上下が固定されていないため、タブレット用で開発したアプリが回転することで表示がおかしくなるということもあります。そのため上下や横幅が自動調整できるように設定しておくことです。

第2節 Android アプリ開発の心構え

今回は、初心者からアプリ開発を始めた私が、これから Android 開発を始める方や、興味はあるけどどうしたらいいのかと思っている方に向けて、Android アプリ開発の心構えや勉強方法、つまづきやすいポイントなどをお伝えしたいと思います。まったくの初心者からアプリ開発を初め、個人でアプリをリリースして稼いでいる方や、個人で受託仕事をしている方、企業へ就職や転職される方も多くいます。アプリ開発をマスターする人のポイントは以下の5つになります。1、人にも頼るが、自分で解決しようとする。2、テキストや問題集(書籍)を反復で練習する。3、難しいことに焦点を当てず、一つ一つできることから行う。4、毎日開発環境を触る。5、自分の作りたいアプリがある。または、作りたいアプリを見つけようとする。やるならぜひ、Android アプリ開発をマスターすると心に決めて学んでほしいと思います。

謝辞

本研究を進めるにあたり、終始熱心に御指導頂きました田中章司郎教授には心より御礼申し上げます。また、同じ研究室の皆様には数々の御協力と御助言を頂きましたこと、深く感謝致します。なお、本論文、本研究で作成したプログラム及び、データ、並びに関連する発表資料などの全ての知的財産権を本研究の指導教員の田中章司郎教授にご譲渡致します。

参考文献

[1]http://www.health-net.or.jp/tairyoku_up/chishiki/ninchisyuu/t03_08_02_05.html

[2]http://game.watch.impress.co.jp/docs/interview/20150114_683538.html

[3]<https://ja.wikipedia.org/wiki/Eclipse>

[4]<https://ja.wikipedia.org/wiki/%E8%AA%8D%E7%9F%A5%E7%97%87>

[5]<https://ja.wikipedia.org/wiki/%E8%A6%81%E6%B1%82%E4%BB%95%E6%A7%98>

[6]<https://ja.wikipedia.org/wiki/%E3%82%BD%E3%83%95%E3%83%88%E3%82%A6%E3%82%A7%E3%82%A2%E9%96%8B%E7%99%BA%E5%B7%A5%E7%A8%8B>

[7]<http://techfun.cc/devbase/devbase2.html>

付録

アプリ評価シート

Android アプリ「memory up」のアンケート

201214069

鐘文聡

本アプリの主旨は人間の脳力をアップすることができ、年寄の人が遊べる、楽しめる、シンプルなアプリケーションである。

(点数は5点満点)

1. あなたの性別を選択してください。

- 男性 女性

2. あなたの年齢を記入してください。

- 30代 40代 50代 60代

3. 本アプリのデザインはどうでしたか？

- 1点 2点 3点 4点 5点

4. 本アプリは使いやすいですか？

- 1点 2点 3点 4点 5点

5. 本アプリ画面の大きさはどうでしたか？

1点 2点 3点 4点 5点

6. 本アプリの音はどうでしたか？

1点 2点 3点 4点 5点

7. このようにすればもっとよくなる点があれば記入してください。

ご協力ありがとうございます。