

JavaFX 班活動報告書

大山航平* 稲垣和真† 玄元奏‡
中野海人† 野崎弘晃† 山口流星§
吉田享平‡ 立川泰暉¶ 坪倉奏太||
中山凌一** 堀越俊行**

2019 年 2 月 11 日

*情報理工学部 情報システム学科 三回生

†情報理工学部 情報理工学科 知能情報コース 二回生

‡情報理工学部 情報理工学科 実世界情報コース 二回生

§理工学部 数理科学科 二回生

¶情報理工学部 情報理工学科 セキュリティネットワークコース 一回生

||情報理工学部 情報理工学科 一回生

**情報理工学部 情報理工学科 システムアーキテクトコース 一回生

目次

第 I 部 全体報告	4
1 プロジェクト概要	5
2 Java 勉強会	5
3 JavaFX ハンズオン	6
第 II 部 作品紹介	8
1 RGB クイズ	9
1.1 作品の概要	9
1.2 作品に対する想い	9
2 FizzBuzzCounter	11
2.1 作品の概要	11
2.2 作品に対する想い	12
2.3 今後の展望	12
3 WorkManager	13
3.1 作品の概要	13
3.2 作品に対する想い	14
4 ログインが出来るアプリ	15
4.1 作品の概要	15
4.2 作品に対する想い	15

5	SPRoid	18
5.1	作品の概要	18
5.2	作品に対する想い	20
6	素数当てゲーム	21
6.1	作品の概要	21
6.2	作品に対する想い	23
7	ピングオアプリ	25
7.1	作品の概要	25
7.2	作品に対する想い	25
8	スロットマシーン	27
8.1	作品の概要	27
8.2	作品に対する想い	28
9	円グラフ生成器	29
9.1	作品の概要	29
9.2	作品に対する想い	29

はじめに

文責：吉田享平

世の中には様々なアプリケーション（以下、アプリ）が存在する。ネイティブアプリケーション（以下、ネイティブアプリ）もその一つである。今日、Web アプリケーションが普及し、ネイティブアプリが活躍する機会が少なくなってきた。しかしながら、ネイティブアプリの需要は無くなったわけではなく、これからもネイティブアプリは小さい市場の中で活躍し続けるだろう。

本プロジェクトでは、モダンなネイティブアプリを開発できる人材を輩出するといった目標のもと活動した。

第I部

全体報告

ここでは全体で活動したものを報告する。

1 プロジェクト概要

文責：吉田享平

本プロジェクトは Java の GUI ライブラリである「JavaFX」を用いたアプリケーションを制作した。活動はハンズオン形式でサンプルプログラムを確認し、各自オリジナルアプリを制作した。

しかしながら、プロジェクトのメンバーのほとんどが Java を扱ったことがなかったため、活動の前半は Java の基礎を勉強会方式で行った。活動の後半は先述した形式に進め、最後には個人が制作したアプリを発表し、共有した。

開発環境は IDE を採用し、IntelliJ IDEA を推奨とした。今回、コンソールでの開発は非推奨としている。Java のバージョンは 8 を採用している。JavaFX の開発にはこのバージョンが適していると判断したためである。

2 Java 勉強会

文責：堀越俊行

本プロジェクトの本題である JavaFX を勉強するにあたって、Java への理解を深め、独力で書き上げる能力を身に着けるために、Java の勉強会を前半 4 回に分けて行った。

第一回目の勉強会では、Java の環境構築及び簡単な概要をメンバーに周知した。

今回のプロジェクトでは、JetBreins 社の IDEA を使用エディタ、Java のバージョンは 8 を推奨し、班員の多くは環境構築を推奨環境で行った。

環境構築を終わらせたのち、Java の基本的な書き方を勉強し、全員が HalloWorld を済ませた。

第二回では、前回勉強したことの続きから行った。

具体的には、変数の概念を勉強し、その後に if 文や for 文、while 文などの条件分岐について勉強した。

変数はプロジェクトメンバーは C 言語を勉強していることもあり、あまりつまづくことはなかったように感じた。

第三回では、メソッド、スコープ、クラスについて勉強した。

Java におけるこれらの概念は一回生が勉強している C 言語とは違った点が多く、苦戦していたメンバーも多かった。

実際にプロジェクトリーダーが作成したコードを改変するなどし、よく起こるエラーの原因を勉強したりなどした。

第四回では、オブジェクト指向を勉強した。

今回はオブジェクト指向は本題ではないため、すべての班員が自信をもって理解できたと言えるまでは周知するまでには至らなかったが、今回の成果物においてその概念を使用しているメンバーも多くいることから、ある程度の理解は全員が得られたのではないかと考えられる。

3 JavaFX ハンズオン

文責：吉田享平

JavaFX は従来の形式である、Java コードだけで GUI を構成するのではなく、FXML という XML 系のコードと Java を組み合わせさせて GUI を構成する。

JavaFX ハンズオンでは、カウンターアプリをサンプルプログラムとして JavaFX の基礎を習得した。カウンターアプリは、ボタンをクリックすると画面に表示されている数字がカウントアップするという簡単なものである。

今回、GUIを構成するFXMLをコーディングするにあたって、SceneBuilderというツールを使用した。これはGUI操作でGUIを構成することができ、FXMLを書き出すこともできる。FXMLは学習コストが高いため、このようなツールを使用した、

第II部

作品紹介

ここでは、JavaFX を学んだ集大成として各メンバーが制作したアプリケーションを紹介する。

1 RGB クイズ

文責：大山航平

1.1 作品の概要

表示される色を見て RGB 値を当てるクイズゲームである。回答フェーズ (図 1) では、画面左上に表示される四角形の色の RGB 値を予想して、画面左下のスライダーで入力する。その後画面左下の「決定ボタン」を押すことで結果発表フェーズ (図 2) に移行する。

結果発表フェーズでは、問題の四角形の横に自分の入力した色の四角形が表示される。また、その右には正解の RGB 値がそれぞれ表示される。スライダーの右には、自分の入力した値が正解とどれだけ近かったかが 100 点満点で評価され、表示される。画面右下の「次へボタン」を押すと次の問題の回答フェーズに移行する。

1.2 作品に対する思い

今回は JavaFX の練習も兼ねてのアプリケーション作成だったので、ある程度複雑な入れ子構造のレイアウト作成、スライダーによる数値入力、ボタン押下による制御など、JavaFX 開発における基本的な要素を複数一度に学べるようなアプリケーションを考えた。

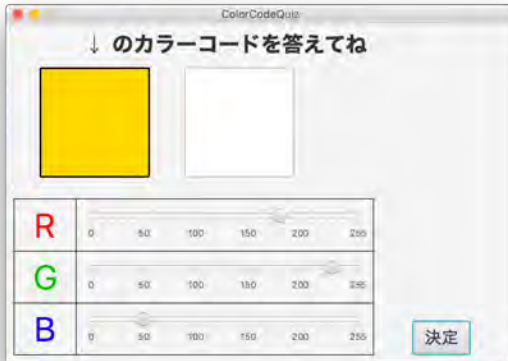


図 1: 回答フェーズ



図 2: 結果発表フェーズ

2 FizzBuzzCounter

文責：稲垣和真



図 1: 世界のナベアツ出力画面

2.1 作品の概要

ボタンを押すと、カウンターにカウントが1つずつ追加されるアプリケーションです。

また、カウンターの数字が3の倍数になった時に'世界のナベアツ'が登場します。

また、5の倍数になるとよくわからない方の画像が表示されます。

2.2 作品に対する想い

このプロジェクトに入って初めて javaFX でプログラミングをしたので、わからないことが多く苦勞した。

FizzBuzz を GUI 上で実行する単純なプログラムであるが、GUI 上でナビアツが出てくるように制御した。

2.3 今後の展望

このプロジェクトでは、Java の基本文法から学び、最終的に JavaFX を使って GUI アプリの作り方を学んだ。

しかし、自分の演習不足により、十分に JavaFX を使ったプログラミングに馴染めず、改善の余地が大いにあるアプリケーションしか作成することができなかった。そのため、自主的に javaFX の復習や演習を積んでいきたい所存である。



図 1: 作成したアプリのスクショ

3 WorkManager

文責：玄元奏

3.1 作品の概要

私が今回 JavaFX を学んで制作した作品は、給与計算アプリケーションである。本アプリケーションの使い方は作業時間と時給を入力し、下部にある更新ボタンを押すと、給与が算出されるというものである。オプションで作業日をドロップダウンリストから選択し、業務内容を記述することもできる。

3.2 作品に対する想い

このアプリケーションを製作した理由は、私はアルバイトをしており、自分の給与を計算するときには電卓で毎回手打ちして概算していたので、その処理を効率よくしたかったからである。今後の展望としては、保存ボタンをつけたものの、保存する機能はまだ実装できていないので、各項目を入力した後に保存ボタンを押すと、テキストファイルなどに書き出して保存できるように改善したい。

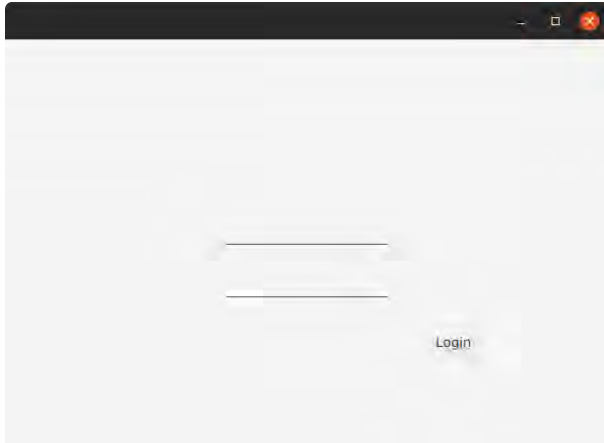


図 1: ログイン画面

4 ログインが出来るアプリ

文責：中野海人

4.1 作品の概要

名前とパスワードを入力するとログインが出来るアプリである。ログイン成功後は別の画面に遷移する。ログアウトボタンでログイン画面に戻り、またログイン可能な状態になる。

4.2 作品に対する思い

Scene builder での画面作成時に「JFoenix」というライブラリをつかった。これを使うことで、パスワード入力用のテキストボツ

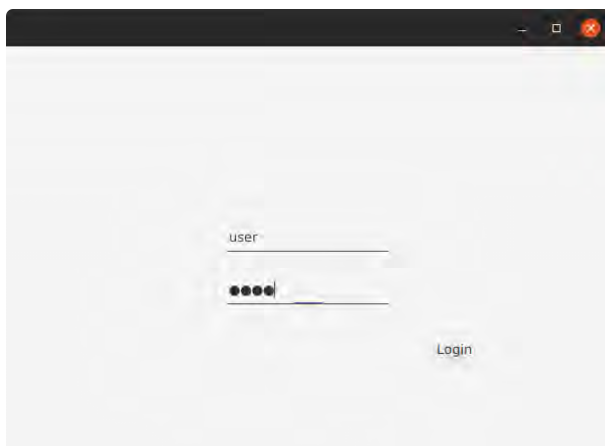


図 2: ユーザー名とパスワード

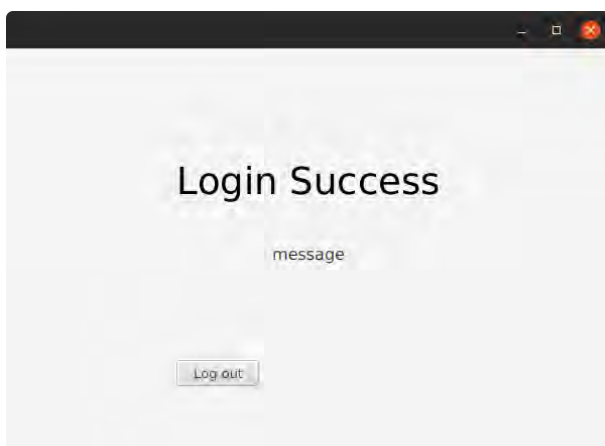


図 3: ログイン後の画面

クスでは入力した内容を表示させないようにしている。今後の改善点としてはユーザー名とパスワードをファイルで管理し、ユーザーごとにログイン後の表示を変えることが挙げられる。

5 SPROID

文責：野崎弘晃

5.1 作品の概要

今回作成したアプリケーションは「SPROID」というものである。これは、Youtubeやニコニコ動画の生放送で、配信者が発話した音声をデスクトップに字幕で表示させるというものである。

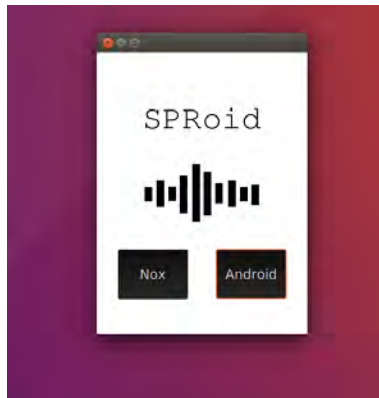


図 1: 最初のメニュー画面

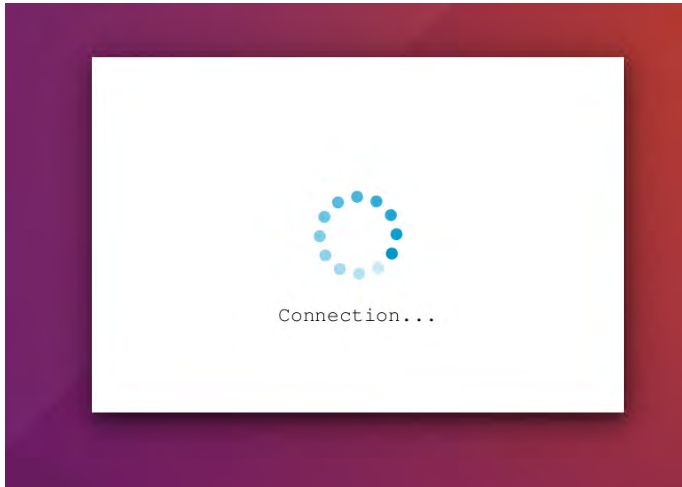


図 2: 接続待機画面

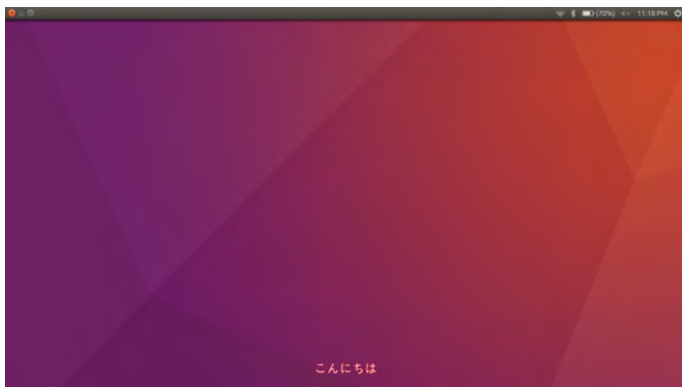


図 3: 字幕画面

5.2 作品に対する想い

動画の生配信をする際、生声で放送することに抵抗がある人は少なくないだろう。しかし、いちいちキーボードで文字を打ち込んで視聴者にメッセージを伝えることはなかなか難しい。そこで、Android の音声認識 API を利用して文字に変換し、Socket 通信で配信用 PC に送信した後、配信用 PC で字幕を表示させるアプリケーションを作成した。なお、iOS ユーザーは「Nox」と呼ばれる Android エミュレータを利用することで、同じように音声字幕を起こすことが可能である。

6 素数当てゲーム

文責：山口 流星

6.1 作品の概要

数字を一目見たとき，誰しも一度は「これは素数だろうか」と考えたことがあるだろう．このゲームはまさにそのような方に捧げるゲームである．

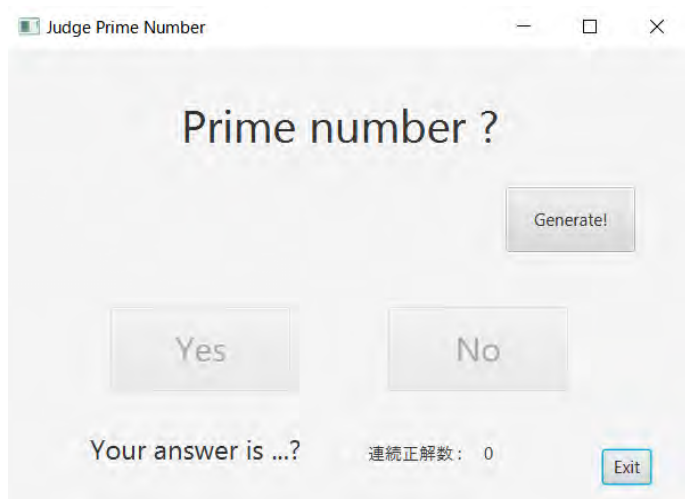


図 1: アプリ起動時の画面

アプリを立ち上げた画面はこのようになっている．Generate ボタンをクリックすると，素数候補の数字が表示される．

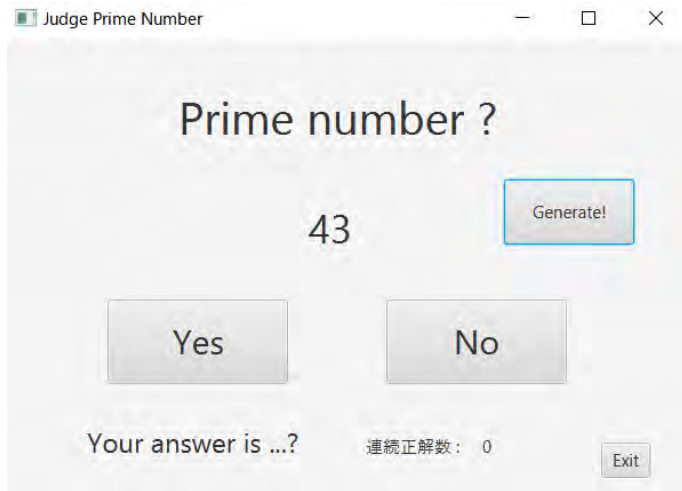


図 2: Generate 後の画面

ユーザは素数ならば、Yes ボタン、合成数ならば No ボタンをクリックする。正誤に関わらず次の素数候補の数字が表示されるため、続けて判定できるようになっている。



正解するごとに連続正解のカウントが増えていくので、己の限界を目指して当て続けてもらいたい。アプリを終了する際は、右下にある Exit ボタンを押す。

6.2 作品に対する想い

筆者はプログラミング経験がほとんど無かったため、シンプルに楽しめて、かつ設計が容易なゲームを作ることを目標とした。

プロダクトとしては、素数を判定するという非常に単純なゲームであるため、素数候補の数字がストレスなく表示されるように工夫した。アプリを立ち上げた後は、スタートボタンの役割である Generate ボタンを押すまでは判定ボタンを押せないようにした。

また、素数候補の数字は乱数から生成しているが、その際に2の倍数と5の倍数であるとき、再抽選する仕様にした。下一桁だけで判断できてしまうため、面白くないという理由である。

素数判定に関しては，エラトステネスの篩を使用した．プロダクトの設計時の書きやすさと，計算時間がそれほどかからなかったことから，こちらを採用した．

素数を愛してやまない人に向けてのプロダクトのため，有名なあの素数に対しては特別な結果が表示されるようになっている．このプロダクト唯一のアピールポイントと言ってもよいだろう．

7 ビンゴアプリ

文責：立川 泰暉

7.1 作品の概要

Next button を押すことで 0 から 75 までの数字がランダムに表示される。表示された数字の履歴はテキストエリアで確認することができる。一度表示された数字は重複しないようにした。



図 1: ビンゴアプリの画像

7.2 作品に対する想い

一度表示された数字が重複しないようにすることに苦労した。最初のプログラムではまず、1 から 75 の範囲に限定した乱数を生成

するプログラムを書いた。その後、生成した乱数を代入した配列全てと新たに生成された乱数を代入した変数を比較し、重複していれば新たに乱数を生成していた。そのため、新たに生成された乱数と過去に配列に代入された値の重複を避けることが出来なかった。そこでリストを使って生成した1から75の数を `Collection.shuffle()` を使い、リストの中身をシャッフルすることで表示する数字の重複を起らないようにすることができた。

8 スロットマシーン

文責：坪倉奏太

8.1 作品の概要

Timeline クラスを用いて、スロットの数字を更新しつつ、スロットの動作音を流し、スロットの数字が揃えば動作音を止め、新たなウィンドウをポップアップさせるアプリケーションを作成した。



図 1: スロットが揃ったとき



図 2: スロットが揃ったときの
結果出力



図 3: スロット画面

8.2 作品に対する想い

アプリ制作初心者ということもあり，JavaFX の仕様について調べながら制作するような形となってしまう，ほぼ全ての段階で苦労かつ紆余曲折した．例えば，音楽の実装をしようとした時，最初の方は，ボタンを押したときに動作するコードの部分で動作音を止めようとしていたり，音量ゼロにすれば実質音楽止めたことになるだろうと思って音量設定を行えるオブジェクトを取得したら，コンパイルした時に「そんなん知らんわ」って言われてたりしてました．(実装してから振り返ってみると，なんで苦労していたのか不思議．)ただ，'実装しよう'と思っていた機能が曲がりなりにも全て実装でき，個人的には満足だ．

9 円グラフ生成器

文責：中山凌一

9.1 作品の概要

初めの画面 (図 1) で、表示したい円グラフのタイトル、円グラフで表示したいデータを 4 つ、各テキストボックスに打ち込む。(図 2) その後、画面下部にある「Start」ボタンをクリックすることで打ち込んだデータをもとに円グラフを生成する。(図 3)

また、初めの画面 (図 1) で、データの値を打ち込む際に数字以外または負の値を打ち込む (図 4) と打ち直しを求める画面を表示する。(図 5)

9.2 作品に対する想い

今回のアプリケーションでは、事前に表示された例がボタンを用いたアプリケーションであったため、他の班員が使いそうに無い機能を使ってみようと思い、PieChart(円グラフ)の機能を利用し製作した。Java 自体初めて触る言語であったため、そのライブラリ関数である JavaFX を使う際躓くことが多く、参考サイトについても今回使用した JavaFX Scene Builder を使っているサイトがなかなか無く、試行錯誤しては、先輩に尋ねるといった結果になってしまった。しかし、その甲斐もあり、まだまだ不十分であるが、形となるアプリが作成できたと考える。

図 1: データ入力画面 : 入力前

AkiChannel	710000
Kaguya Luna Official	900000
Miwa Akari Project	730000
Siro Channel	590000

図 2: データ入力画面 : 入力後



図 3: 円グラフ画面

図 4: データ入力画面:入力ミス

右側の空欄には
正しく数字を入力してください

図 5: 打ち直し要求画面

おわりに

本プロジェクトは、活動頻度も高く、プロジェクト活動として成功したと考えている。また、メンバーは毎回の活動に精力的に活動し、多くの人が無らかなの技術を身につけられたと考えている。

しかしながら、初心者が多い中で説明を急いでしまったり、飛ばしてしまったことは今後の課題である。プロジェクト活動が終わった後もサポートをしていきたいと考えている所存である。最後になったが、本プロジェクト参加してくれたメンバーには最大限の感謝と謝辞を伝えて報告書とする。