Androidクイズアプリ「やる助じいさんの八戸藩のこと，教えてやるすけ！」の設計と実装

小久保 温[†]・小玉 成人[†]・伊藤 智也[††]

Development of Quiz Application "Grandpa Yarusuke, Please Tell Me About the Hachinohe Domain!" for Android OS

Atsushi KOKUBO[†], Naruhito KODAMA[†] and Tomoya ITO[††]

ABSTRACT

We developed an Android quiz application in collaboration with Hachinohe City Museum. There are 4 quiz genres, with 20 questions in each genre. A user first selects a genre, and then 10 questions are randomly selected without duplication from the selected genre. The graphic design of the application's screen was created with accessibilities. Processing IDE was used to develop the program. The program was designed with the MVC architecture as a reference. Since the state transition of the screen for answering questions became complex, we decided to create a state class to manage it. The application was displayed at the Hachinohe City Museum's autumn special exhibition, "The Abandoned Domain: The Last Lord of the Domain, Nobuyuki Nanbu," held from October 2 to November 23, 2021. A logging function was implemented in the application to record the usage. In terms of usage, the quiz results were displayed a total of 138 times, which is 4.31 times per opening day. The percentage of correct answers to the quiz was found to vary from 10% to 100%.

Key Words: Android Application, quiz, accessibility, Processing, Design Pattern, log analysis

キーワード : Androidアプリ，クイズ，アクセシビリティ，Processing，デザインパターン，ログ解析

1. はじめに

八戸市博物館との共同研究でAndroidクイズアプリ「やる助じいさんの八戸藩のこと，教えてやるすけ！」をProcessing[1)]で開発した。八戸市博物館がクイズの問題やアプリの素案を作成し，八戸工業大学が具体的なデザインやプログラムを開発した。アプリは2021年10月2日から11月23日に開催された八戸市博物館秋季特別展「今般廃藩之儀－最後の藩主・南部信順－」で展示された。同展覧会には八戸工業大学システム情報工学科から「幕末八戸藩ぐるぐる」「参勤交代！奥州ウルトラクイズ」も出展された。

令和3年12月6日受付

† 工学部システム情報工学科／大学院工学研究科電子電気・情報工学専攻・教授

†† 工学部システム情報工学科／大学院工学研究科電子電気・情報工学専攻・准教授

2. 八戸市博物館の要件

本アプリに関して，博物館からの要望は次のようなものであった。

1. 使用する端末
   過去の共同研究で開発したアプリを展示するのに使っていた約10.1インチ型ワイドタブレット NEC LAVIE Tab E の 2019 年春モデル (TE510/JAW)である。
2. クイズのジャンル
   「八戸藩の人物」「八戸藩の内政」「八戸藩の外交と軍事と文化」「八戸藩の産業」の4つ
3. 問題と出題と結果の表示
   回答者は最初にジャンルを選択する。各ジャンルには20問の問題があり，そのうち10問をランダムに重複なく抽出して出題する。10問回答すると正答数に応じて結果が表示される。各問題は，問題文と3つの選択肢から構成される。各問題の回答時間は15秒である(後に30秒)。
4. 八戸藩に関する紹介
   クイズ以外に八戸藩に関する紹介があり，「八戸藩とは」「八戸藩のおわり」の2ページから構成されている。
5. アプリのデータなどは8月下旬に提供する。開発期間は9月下旬までである。

このうち，1から4までは要望通りに実装した。

5の納期に関しては無理があった。この案件は可能であれば学生の成長のために学生が，困難な場合は教員が開発するという設定であった。しかし，通常，アプリの開発はそれを専門に行なっている事業者でも3ヶ月から半年程度かかる。大学の教員は学生の教育や研究が本業でアプリの開発ではない。また，学生がはじめて開発する場合には未知の事項を調べて試行錯誤しながら開発することになるため，熟練者の数十倍から百倍程度時間がかかる。そこで，展覧会には高速に開発できる教員の小久保が開発したものを仮に納品する。そして，学生は年内にアプリを開発して納品し，それを展覧会以降の機会では使うことにした。それでも開発期間が短すぎるため，素案とサンプルデータを7月中旬に提供していただいて開発をはじめ，本番のデータを8月下旬に提供してらうことにした。

3. 八戸市博物館作成のアプリの画面案

八戸市博物館が作成したアプリの画面案を図 1〜8 に示す。これは白黒で作られたあくまでも素案であった。

図 6 の結果画面は全問正答した場合のもので，これ以外に 70，40，0 点の 4 段階の画面案があった。結果の算出は 1 問正答で 10 点が入り，10 問出題されるので 0〜100 点の範囲になる。

図 7，8 が八戸藩に関する紹介である。この 2 枚の画面はレイアウトが異なるため，それぞれ別に開発する必要があった。

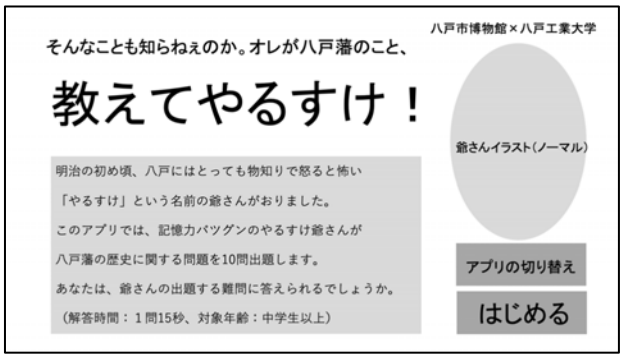

図1 ホーム画面

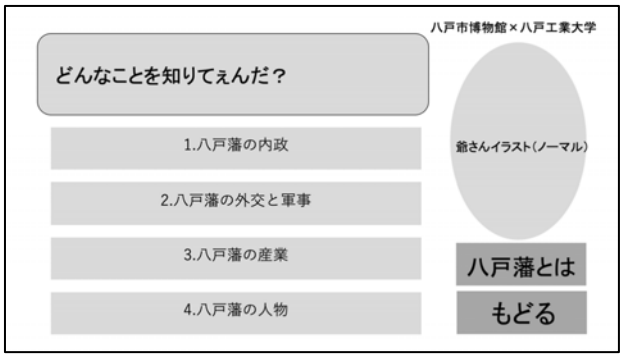

図2 クイズのジャンル選択画面

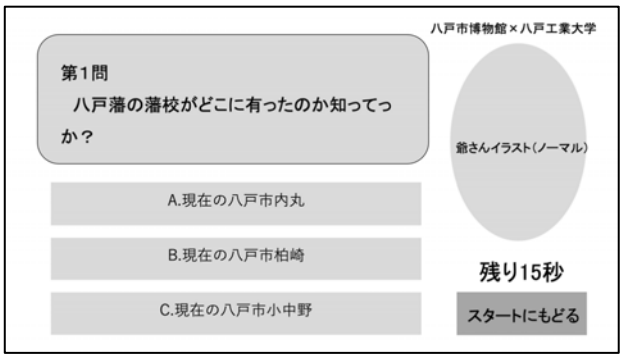

図3 クイズの問題画面

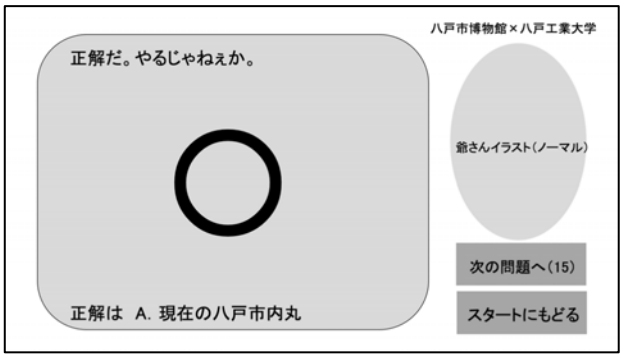

図4 クイズに正答した場合

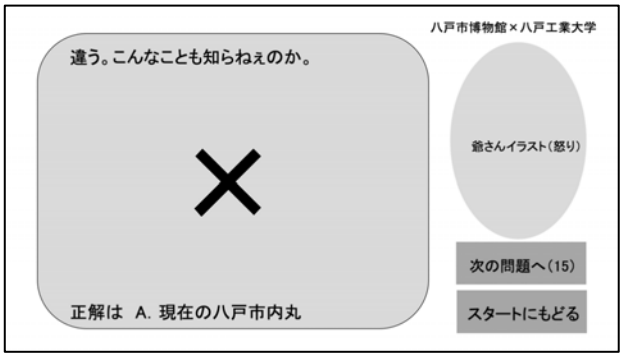

図5 クイズに誤答した場合

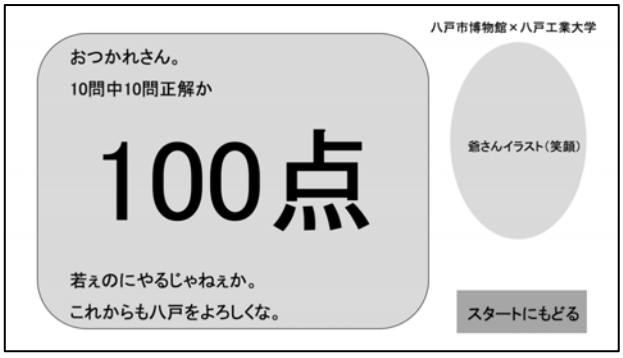

図6 結果画面

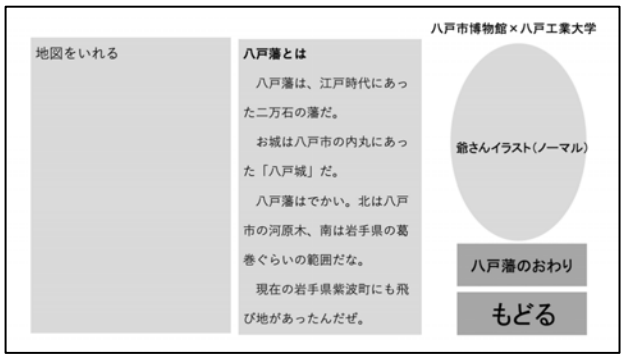

図7 八戸藩紹介画面「八戸藩とは」

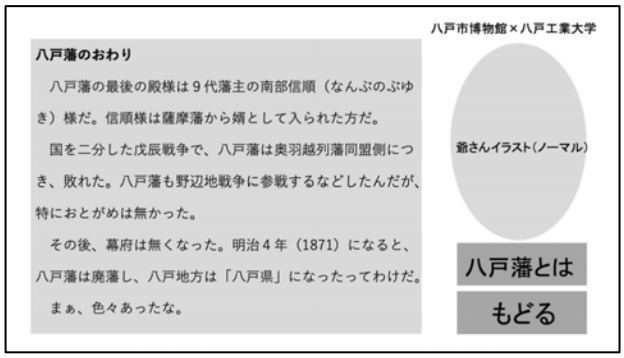

図8 八戸藩紹介画面「八戸藩のおわり」

4. 結果のレベル分けとやる助じいさん

このクイズは八戸藩に詳しい「やる助」じいさんが出題するという設定になっている。画面の案(図 1～8)の「爺さんイラスト」の部分に「やる助じいさん」の画像が入る。八戸市博物館が作成した「やる助」じいさんの画像のうち 2 枚を図 9，10 に示す。さまざまな表情とアクションをしている画像があり，計 7 種類提供された。この 7 種類のうちには，結果の画面で使用する画像が 5 レベル分用意されていたが，画面案では 4 レベルで異なっていた。そこで八戸市博物館と相談して表 1 の 5 レベルで開発することにした。

表1 得点のレベル分け

得点	やる助じいさんの様子
100	笑顔をする
70，80，90	サムズアップする
40，50，60	微笑する
10，20，30	考え込む
0	怒り

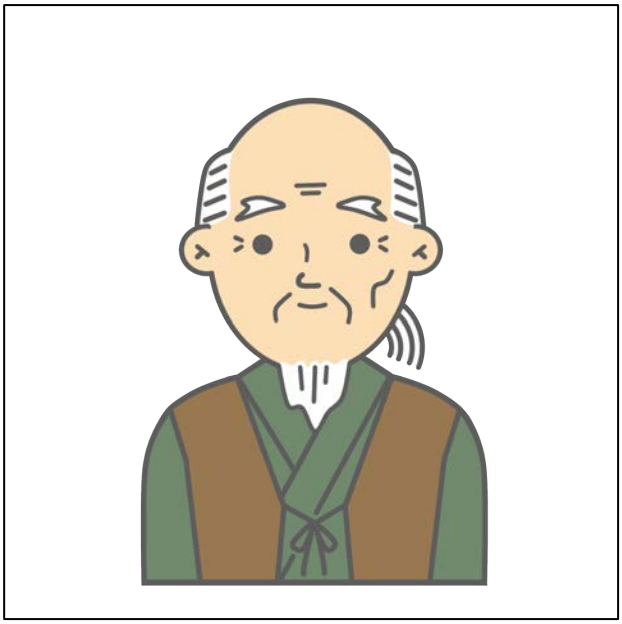

図9「やる助じいさん(ノーマル)」

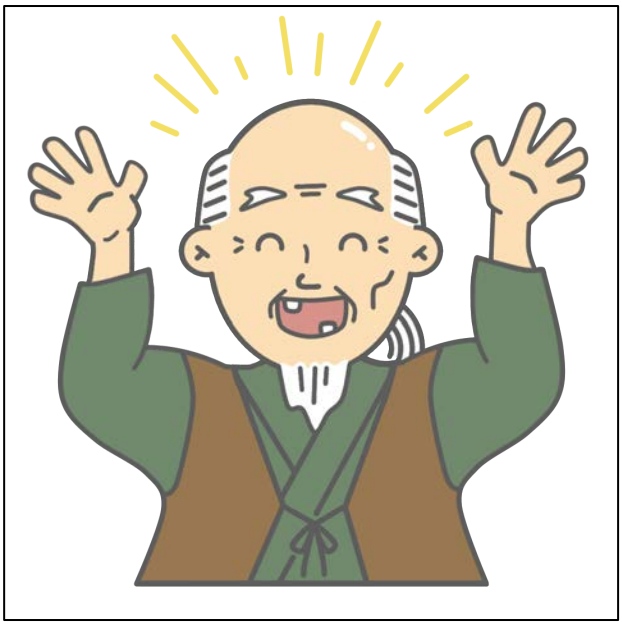

図 10「やる助じいさん(笑顔)」

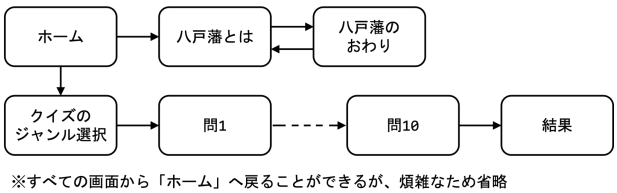

図11 画面遷移図

5. 実装した画面遷移

八戸市博物館が作成したアプリの画面案(図 1～8)，「やる助」じいさんの画像(図 9，10)などを元に，実装するアプリ画面遷移を図 11 のように設計した。ホーム画面で八戸藩に関する紹介にすすむ「八戸藩とは」と，クイズにすすむ「クイズのジャンル選択」に分岐する。また，すべての画面から最初のホーム画面に戻ることができるが，遷移の線が煩雑になるために図 11 では省略している。

6. 画面の状態遷移

図 11 のそれぞれの画面について，状態遷移を検討した。「問題」の画面以外では，利用者の入力(ボタンのタップ)を待って，入力が行われるとサウンド(効果音)を鳴らして，次の画面に遷移する。これを状態遷移図にすると図 12 になる。

「問題」の画面の場合，状態遷移は複雑である。実装した状態遷移を図13に示した。「入力待ち」の状態の間，残り時間がカウントダウンされる。入力が行われると次の状態に遷移する。入力が正答か誤答かによって，○か×を表示してサウンドを鳴らして，次の画面に遷移する。

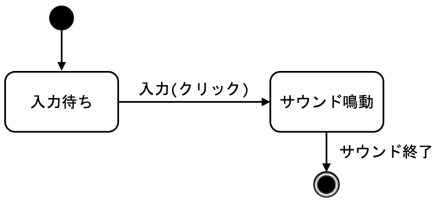

図12「問題」以外の画面の状態遷移図

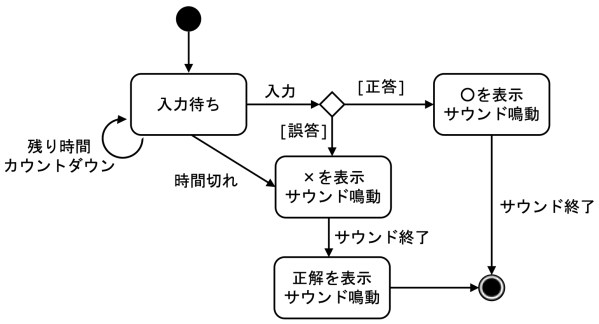

図13「問題」の画面の状態遷移図

八戸市博物館の画面の案には含まれていないが，×の場合は正答を表示してサウンドを鳴らしてから次の画面に遷移することにした。また，同様に案には含まれていないが，入力待ちの画面で時間切れになると，誤答の場合と同様の状態遷移をさせることにした。

7. 実装したアプリの画面

画面と状態の遷移(図 11～13)を元に，画面を作成した(図 14～23)。画面を作成する上では，クイズの問題と選択肢の最長の文字数を調べ，画面におさまるようにフォントサイズと文字の描画領域を決めた。問題文の最長の文字数は「八戸城下で馬の売買・仲介を仕事とする馬喰（ばくろう）が多く住んでいたのはどこ？」の 39 文字，選択肢は「町奉行日記・勘定所日記・用人所日記」の 17 文字であった。

図14 ホーム画面

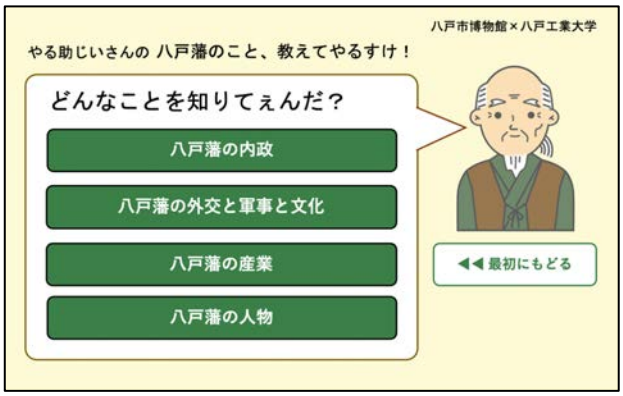

図15 クイズのジャンル選択画面

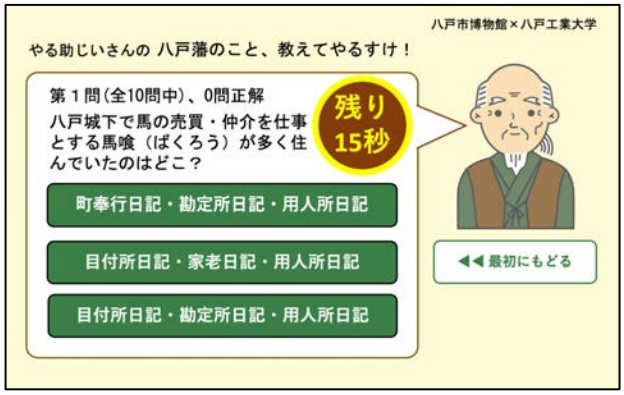

図16 クイズの問題画面

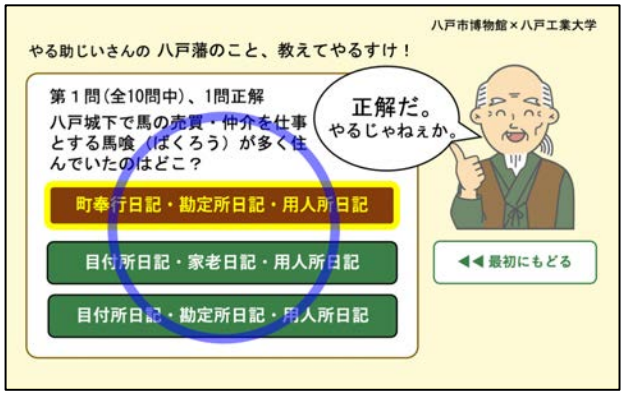

図17 クイズに正答した場合

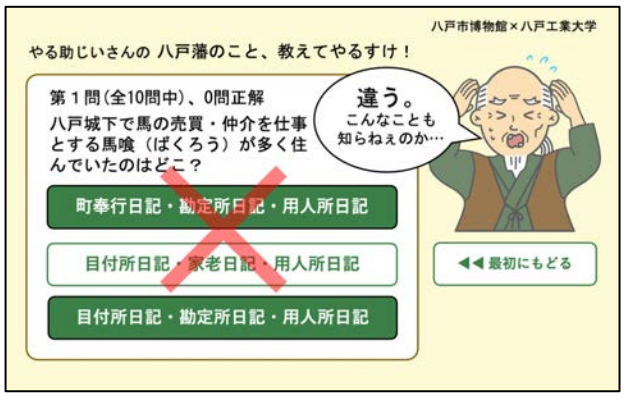

図18 クイズに誤答した場合

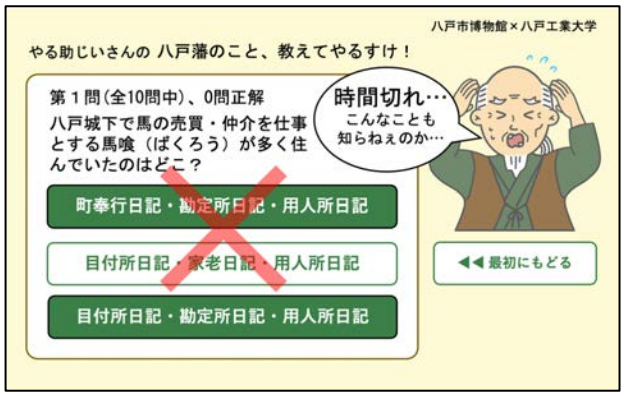

図19 時間切れになった場合

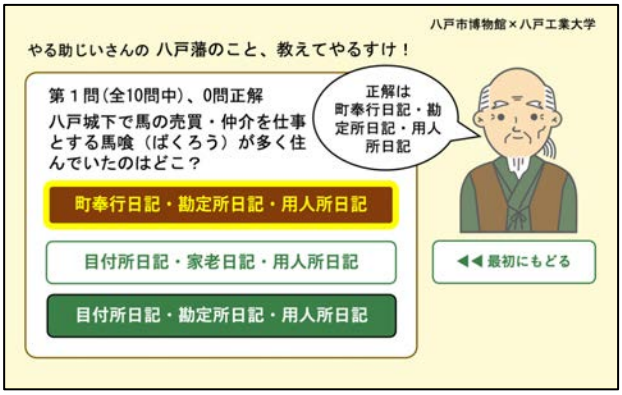

図20 正答以外では正解を示す

図21 結果画面

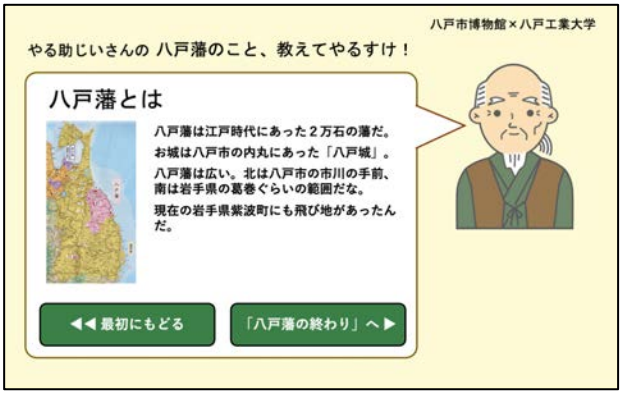

図22 八戸藩紹介画面「八戸藩とは」

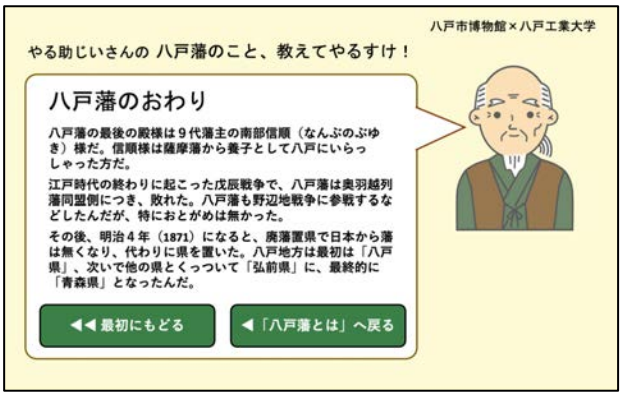

図23 八戸藩紹介画面「八戸藩のおわり」

8. 配色のアクセシビリティ

画面のデザインは，アクセシビリティに配慮して表示される文字と背景色の配色を選んだ(図24)。障害者差別解消法が 2013 年に成立し公共機関での取り組みが義務化され，2021 年の改正では民間事業者についても義務化された。これに伴い Web コンテンツなどにアクセシビリティへの配慮が求められることになった。たとえば WCAG(Web コンテンツアクセシビリティガイドライン) 2.1[2]には，さまざまなガイドラインが示されている。今回のアプリは Web コンテンツではないが，ディスプレイで見るものとしては共通である。多岐に渡るガイドラインのうち，今回開発するのは時間制限のあるクイズゲームであることを考慮して，視覚障害者や高齢者にも見やすい配色にすることにした。WCAG 2.1 では，文字と背景色のコントラスト比について最低限 4.5 以上であることが望ましいとされる。これに沿って図 24 のようにカラースキームを決めた。コントラスト比はディスプレイや人間の視覚特性を考慮して算出される値で最小 1，最大 21 となる。

図24 背景色と文字の配色とコントラスト比

9. アプリのプログラム

アプリのプログラムは，Processingで開発した。ProcessingはJavaで作られているため，JavaのAPIを使用することができるため，JavaのCollectionsフレームワークを活用した。開発には，MVCアーキテクチャ[3]を参考にした。MVCとは，プログラムを整理する方法の一つで，データを操作するモデル(Model)，表示を行うビュー(View)，ユーザーの入力に応じてモデルとビューを操作するコントローラー(Controller)と，プログラムを役割ごとに分けて開発するものである。

9.1 モデルのクラス図

モデルのクラスが図25である。質問Questionは，問題文sentenceと選択肢Choiceのリストchoicesから構成される。リストListとは，集合を表すインターフェイスの一つで，これを実装したものには連結リストなどがある。今回のプログラムではJavaのCollectionsフレームワークのArrayListを使用している。このような設計にすることで，次の形式で，ある質問のインスタンスのquestionが持っている選択肢のリストを取り出すことができ，プログラムがわかりやすくなる。

```
question.choices
```

選択肢Choiceは，選択肢の文sentenceとその選択肢が正しいか否かを示すisCorrectedから構成される。

質問はジャンルGenreに属する。Genreはジャンルの名前nameと質問Questionのリストquestionsから構成される。

質問全体を表すのがQuestionsPoolで，これはジャンルGenreのマップgrenresを持っている。マップとは連想配列を表すインターフェイスで，添字とオブジェクトで構成される要素の集合であり，添字はユニークで同じ添字を複数使うことはできない。今回はJavaのCollectionsフレームワークのMapの一つHashMapを使用している。QuestionPoolにはメソッドrandomChoice()を用意した。このメソッドの引数にジャンルと抽出する

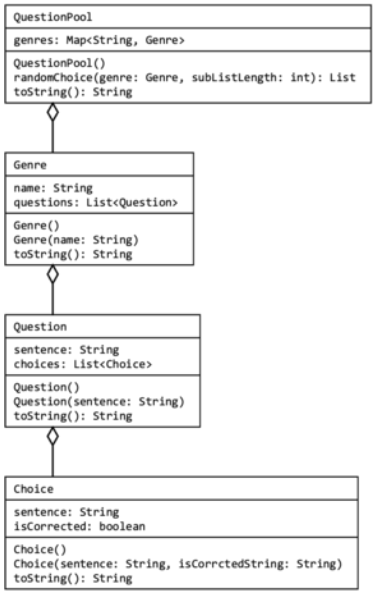

図25 モデルのクラス図

問題の個数を指定すると，そのジャンルから指定した個数の質問をランダムに重複なく抽出する。内部では，JavaのCollectionsクラスのsuffle()メソッドで要素の集合をシャッフルし，Listインターフェイスのsublist()メソッドで要素の集合の部分集合を取り出している。

それぞれのクラスには引数のないデフォルトコンストラクタも用意した。また，デバッグやログ出力などに使用するために，Javaのルートオブジェクトobjectクラスに定義されているオブジェクトを文字列化するtoString()メソッドを上書き(オーバーライド)して実装した。

引数のあるコンストラクタは，データファイルを読み込んでオブジェクトを生成する際に使っている。たとえば，選択肢Choiceのデータは，データファイルに文字列で記述されており，選択肢の文章とその選択肢が正解か否かで構成されている。引数のあるコンストラクタChoice(sentence: String, isCorrectedString: String)は，これらのデータの文字列を引数に指定すると，Choiceインスタンスを生成するようにしている。

9.2 ビューのクラス図

(1) 画面を表すSceneクラス

ビューは，GoFのデザインパターン[4]のStateパ

ターンを参考に開発した。GoFとは，Eric Gammaら4人の技術者のグループを指す。デザインパターンは，オブジェクト指向を活用した設計のサンプル集である。Stateパターンは，そのうち状態を交換可能なオブジェクトとして扱う方法である。開発したアプリでは，Sceneクラスを用意して，個々のシーンはクラスSceneを継承するようにした。個々のシーンは「ホーム」HomeScene，「問1」「問2」…「問10」を表すQuestionSceneなどである(図26)。これにより，さまざまなシーンが共通の親クラスSceneを持ったことで交換可能になる。

(2) Buttonとその関連のクラス

各シーンにはボタンがあり，ボタンをタップするとサウンドが鳴り，次のシーンに移動するなどの反応がある。ボタンを表すButtonクラスを用意し，属性としてボタンの形状geomety，色colorScheme，文字text，ボタンを押したときの反応behavior，効果音soudEffect，ボタンが反応中か否かを表すactivated，正解のボタンを表すisCorrectedなどを用意した(図27)。

また，ボタンの表示に関するクラスとして，形状Geometry，配色のセットColorSchemeを用意した。ColorSchemeは，タップされてアクティブになっている状態の配色activeと，アクティブなっていない状態の配色linkを持っている。activeやlinkは，ある状態における配色を表すColorsクラスのインスタンスで，文字text，背景background，枠線の色borderを属性として持つ(図28)。

画面のデザインはMicrosoft PowerPointで開発しているが，PowerPointの長さの単位はcmである。一方，プログラミングに用いているProcessingではピクセル単位で座標を指定する。この換算を画面にパーツを配置するたびに行うことは，大変に効率が悪い。そこで換算を行う関数px()を次のように用意した。これにより，ビューのプログラムの開発が効率化された。

```
int px(float cm) {
  return int(cm * 28.3);
}
```

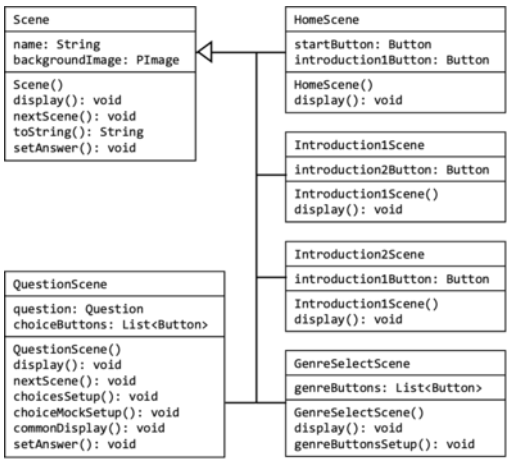

図26 Sceneクラス図

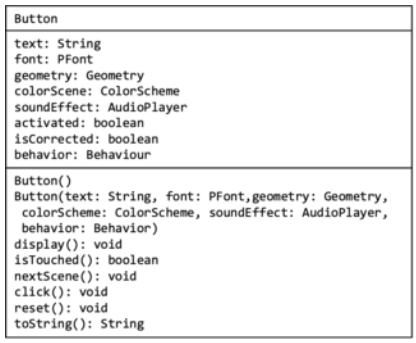

図27 Buttonクラス図

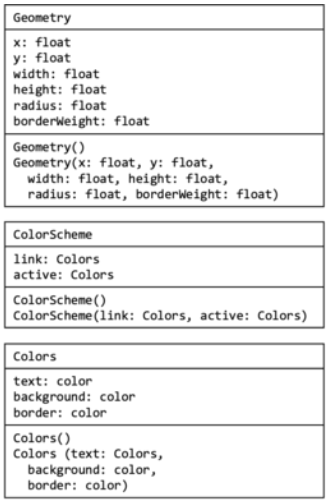

図28 ボタンの表示に関するクラス図

(3) 画面の状態を表すStateクラス

「問題」の画面には他の画面と異なり，複雑な状態遷移がある(図13)。「問題」の画面では，状態によらず表示しているものと，状態に応じて表示しているものがある。たとえば，選んだ選択肢が正解であったことを示している状態(図17)では，薄い青い丸や「正解だ。やるじゃねえか。」といったやる助じいさんのリアクションが，状態に応じて表示しているものである。状態によって変わる表示を扱うために，Stateクラスを用意した。これはさまざまな状態の共通部分で，親クラスである。そして，カウントダウンCountDownState，正答CorrectState，誤答InCorrectState，時間切れTimeoutState，正解の表示AnswerStateなどの個々の具体的な状態はStateクラスを継承する。これらのクラスでは状態に応じた表示を行うためにdisplay()をオーバーライドする。Stateは主に「問題」画面で使うために導入したが，画面でも使うことができて入力待ちIdleinState，画面遷移中TransitionStateなどのクラスも作った(図29)。Stateクラスは，Sceneクラスと同様にGoFのデザインパターンのStateパターンを利用している。

9.3 コントローラに相当する振舞いのクラス図

本アプリでは，コントローラーはボタンを押したときの振舞いに相当する。そこで，Buttonクラスには振舞いを表すBehaviorクラスのインスタンスbehaviorを用意している。Behaviorクラスは，さまざまな振舞いの親クラスで，振舞いの内容を実行するexec()メソッドと，振舞いの実行が終了した後で実行するnext()を用意した。そして，Behaviorクラスを継承して「ホームに戻る」HomeBehavior，「八戸藩とは」に行くIntroduction1Behavior，「八戸藩の終わり」Introduction2Behavior，「ジャンル選択」に行くStartBehavior，ジャンルを選択するGenreBehavior，問題で選択肢を選ぶChoiceBehaviorなどの振舞いを作った（図30)。これはプログラムの構成的には，SceneやStateクラスで採用したGoFのStateパターンと同じだが，状態というよりは挙動を表しているのでGoFのStrategyパターンと言えるだろう。

9.4 タイマーのクラス図

また，問題の画面では，タイマーが表示されている。これは1秒ごとにカウントダウンするものなので，Timerクラス(図31)を用意し、直前の画面での残り秒数before，現在の残り秒数nowなどの属性を整数として持たせ，現在の残り秒数が直前の画面の残り秒数よりも小さくなったらカウントダウンするようにした。また，解答結果の表示画面のようにタイマーは表示されていても，カウントダウンしないことがあり，その状態を表現するためにisActive属性を用意した。

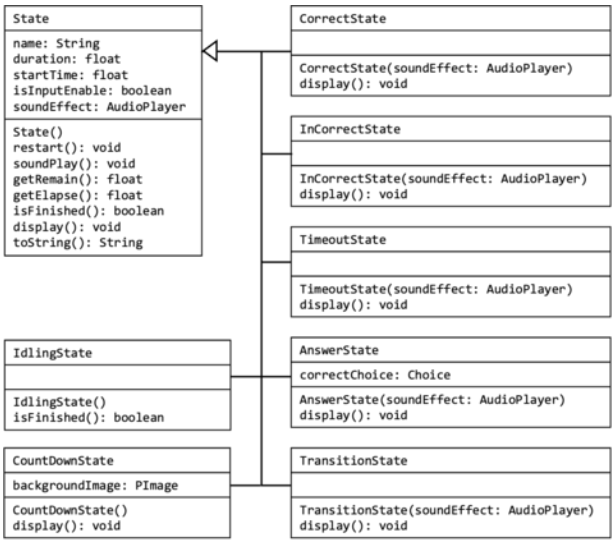

図29 Stateクラス図

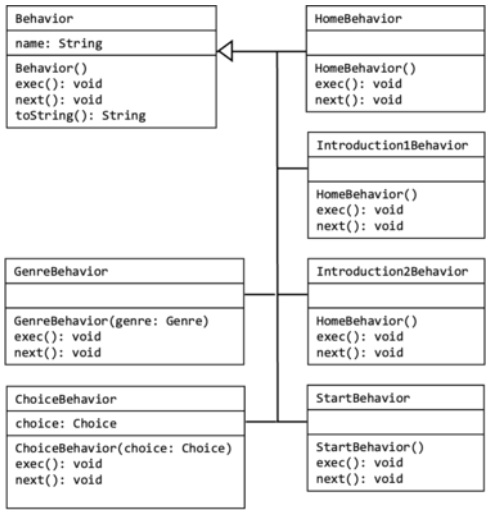

図30 Behaviorクラス図

9.5 仮想デバイスのクラス図

Android端末の画面サイズは，ものによってさまざまである。これに対応するため開発者が開発時に想定した端末を表す仮想デバイスVirtualDeviceクラスを用意した(図32)。そして，実際にアプリが動いている実機の画面サイズに合わせて，仮想デバイスの表示をアスペクト比を保ったまま拡大縮小し，画面の中央に表示できるようにした。属性として仮想デバイスの幅widthと高さheightを用意した。そして，実機の画面サイズを取得するメソッドgetRealSize()で，実機に表示する際に行う拡大縮小率scale，画面上(下)の余白paddingTop，左(右)の余白paddingLeftなどの属性を計算する。計算結果に基づき，実機の画面サイズに合わせて表示するメソッドapply()を用意した。また，実機のポインタの位置から，仮想デバイスのマウスの位置を計算するgetMouseX()，getMouseY()メソッドを用意して，実機を仮想デバイスにマッピングしている。

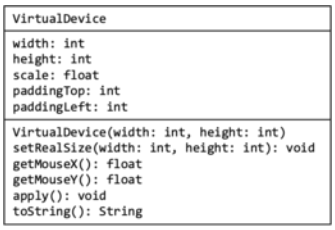

図31 Timerクラス図

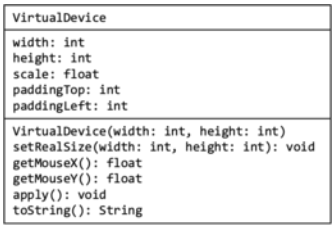

図32 VirtualDeviceクラス図

10. アプリの改修

展示後にアプリの改修を依頼された。改修内容は2つあり，一つは回答時間が短いということで15秒から30秒に増やした。もう一つは，アプリを切り替えるボタンを隠したことである。アプリを切り替えるボタンは，アプリを切り替えられるように，アプリのAndroidアクティビティをバックグラウンドにするものである。これまでの展示では他の研究室のアプリと切り替えるのに使用していたが，今回は切り替える機能が開発されていなかったため，単純にアプリのAndroidアクティビティがバックグラウンドなってアプリが見えない状態になってしまった。そこで，このボタンを非表示にした。

11. アプリの利用状況

アプリでは操作と画面に表示される情報のログをタブ区切りテキストで記録するようにした。展覧会終了後にログを関係データベースに入れ，SQLを使って解析してアプリの使用状況を調べた。ただし，10月18日にアプリを改修した際に小久保のミスでそれ以前のログを削除してしまった。そのため，ログは10月19日から11月23日まで取得されている。

図33に日毎の結果の表示回数を示した。結果は全部で138回表示され，開館日1日あたり平均で4.31回表示されている。特に多かったのが10月19日(火)の9回，10月18日(土)の8回，10月31日(日)の15回，11月3日(水・祝)の11回，11月13日(土)の9回であった。

回答は全部で1,587件あった。このうち正答が748件(47.1%)，誤答が799件(50.3%)，時間切れになった回答が40件(2.5%)あった。

時間切れの回答のうち，回答中に時間が切れたと思われるのは3件で，1問が時間切れ，2問が時間切れとなった人が1人ずつであった。残り37件は6人によるもので，回答を途中でやめて端末を置いて立ち去ったものと思われる。というのも，途中から最後まで回答がずっと時間切れとなっているからである。

ジャンルごとに見ると，表2のようになる。「八戸藩の人物」「八戸藩の産業」は人気があったが，正答率や完答率が高かったのは一番人

気がなかった「八戸藩の内政」である。完答率は最後まで回答した人の割合である。

また，問題ごとに正答率は異なり，10〜100%までの開きがある。表3に正答率上位5問，表4に正答率下位5問を示した。

12. まとめ

Androidクイズアプリ「やる助じいさんの八戸藩のこと，教えてやるすけ！」を，八戸市博物館と共同研究で開発した。

八戸市博物館の仕様や画面デザインの素案をもとに，Androidアプリとしてのユーザビリティやアクセシビリティを検討して，画面遷移や画面を設計した。

開発にはProcessingを用いた。プログラムは，MVCアーキテクチャを参考に役割ごとにクラスに分けた。データを扱うModelではJavaのCollectionsフレームワークを使用した。表示を担うViewやユーザーの入力に対応するControllerでは，GoFのデザインパターンを活用した。

アプリは八戸市博物館の展覧会で展示された。アプリの操作や画面の状況はログに記録した。ログは関係データベースに入れて解析し，日毎に結果が表示された件数，ジャンルごとの人気や正解率，問題ごとの正解率などを調べた。

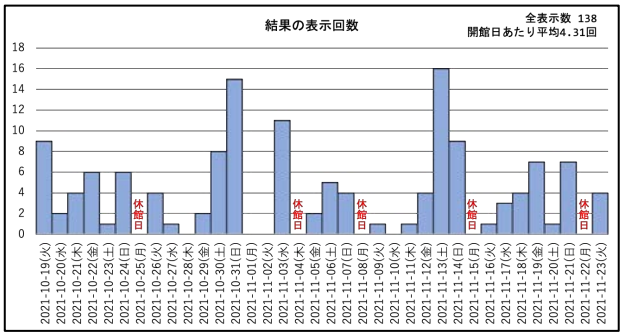

図33 結果の表示回数

表2 ジャンルごとの回答状況

ジャンル	選択	完答	完答率	正答率
八戸藩の人物	78	47	60%	50%
八戸藩の産業	75	49	65%	48%
八戸藩の外交と軍事と文化	33	20	61%	51%
八戸藩の内政	24	17	71%	56%

表3 正答率上位5問

問題	出題	正答	正答率
えんぶりの烏帽子（えぼし）のモデルになっているとされる動物は？	8	8	100%
明治４年７月 14 日の廃藩置県に伴い，八戸藩領に置かれた県の名前は？	11	10	91%
冷害の原因になる，春から夏に海から吹く冷たく湿った風は？	28	25	89%
「相馬大作事件」で相馬大作が暗殺しようとした人物は？	8	7	88%
駒踊りは何の様子を模したもの？	15	13	87%

表4 正答率下位5問

問題	出題	正答	正答率
漁業の税を集めるためにできた施設は？	30	3	10%
八戸藩ができたのはいつ？	13	2	15%
飛脚は江戸ー八戸間を通常約何日で走る？	30	5	17%
豪商・七崎屋の出身で「八戸浦之図」を描いた文人は？	24	4	17%
八戸藩が採用していた年貢の指数を示す制度は？	32	6	19%

参考文献

1) Reas, C., Fry, B., 中西 泰人, 安藤 幸央, 澤村 正樹, & 杉本 達應 (2015). Processing. ビー・エヌ・エヌ新社.
2) Kirkpatrick, A., Connor, J. O., Campbell, A., & Cooper, M. (2018). Web content accessibility guidelines (WCAG) 2.1. WWW Consortium (W3C).
3) Reenskaug, T. (1979). Mvc xerox parc 1978-79. Trygve/MVC.
4) Gamma, E., Helm, R., Johnson, R., Vlissides, J., 本位田真一, & 吉田 和樹. (1999). オブジェクト指向における再利用のためのデザインパターン 改訂版. ソフトバンク パブリッシング.

要　旨

八戸工業大学と八戸市博物館との共同研究で，Android クイズアプリ「やる助じいさんの八戸藩のこと，教えてやる助」を開発した。クイズのジャンルが 4 つあり，問題は各ジャンル 20 問ある。ユーザーは最初にジャンルを選択する。すると問題がランダムに重複なく 10 問抽出されて出題される。アプリの画面のデザインは，アクセシビリティに配慮して制作した。プログラムの開発には Processing を用いた。MVC アーキテクチャを意識して設計した。質問に回答する画面の状態遷移は複雑になったため，状態クラスを作って管理することにした。アプリは 2021 年 10 月 2 日から 11 月 23 日に開催された八戸市博物館秋季特別展「今般廃藩之儀－最後の藩主・南部信順－」で展示された。ログを取る機能を実装し，利用状況を記録した。利用状況を見ると，クイズの結果は全部で 138 回表示され，開館日 1 日あたり 4.31 回であった。問題によって正答率が 10%から 100%までの開きがあることがわかった。

キーワード : Androidアプリ，クイズ，アクセシビリティ，Processing，デザインパターン，ログ解析