

英語教育ソフトの開発

Developing a computer program for Eiken test-takers

小林 健一郎

法月 健

Ken-ichiro KOBAYASHI

Ken NORIZUKI

(平成25年10月15日受理)

Windowsアプリケーションを開発するためのライブラリを制作した。それによって、「本学学生が制作した英語教育ソフトを本学学生が使用する」という環境が構築できると考える。本小論では、試作品を制作し、プログラミング教育と英語教育の双方を進める可能性も論じる。

1. はじめに

本学情報学部において、小林はプログラミング教育を行い、ソフトウェア開発用のライブラリを研究している。プログラミング教育の最終課題は実用ソフトの制作であるだろう。「実用も可能なソフト」「実用的ソフト」ではなく、「実用するソフト」である。実用するソフトを自作し、ユーザのフィードバックをもとにそのソフトを改良していくことができれば、学校教育の限界内においては最高のプログラミング経験になるのではないだろうか。一方、法月は英語教育法を研究し、指導を行っている。語学教育は個別学習者のニーズが大きく異なるため、これまでもゼミ等で個別に英語の資格試験の勉強を学習者に対して、PCの統計ソフトを使って、基礎的な学習履歴の記録を残してきた。

本研究では、両者が協力し、「本学学生が制作する英語教育ソフトを、本学学生が使用し自習すること」の可能性を探る。英語教育ソフトとしては、「英語検定2級を学習するもの」を取り上げる。

2. 英語検定2級について

実用英語技能検定（英検）とは、公益法人日本英語協会が主催し、文部科学省が後援する英語検定試験である^[1]。グレードは5級、4級、3級、準2級、2級、準1級、1級があり、それぞれに筆記試験とリスニング（3～1級は1次試験に合格すると面接試験がある）があるが、ここでは2級の筆記試験を取り上げる。

その問題の構成は以下のようにになっている。

第1問 語い・文法力

短文の語句空所補充。

文脈に合う適切な語句を補う。

第2問 作文力

短文中の語句整序。

短文または会話文を読み、前後関係に合うように与えられた語句を並べ替える。

第3問 読解力

長文の語句空所補充。

パッセージの空所に文脈に合う適切な語句を補う。

第4問 読解力

長文の内容一致選択。

パッセージの内容に関する質問に答える。

3. Windows SDK用のライブラリ

小林は、これまでに、「ゲーム制作用ライブラリ」をC++で開発している^[2]。その内容は、Direct Xをカプセル化するものである。Direct Xは、マイクロソフト社が提供する「高速な描画や途切れない音声を使うためのライブラリ」であり^[3]、これを利用する英語教育ソフトには魅力を感じる。Direct X自体の扱いは煩雑だが、小林制作のライブラリを使えば、それほど難しくはない。実際、これまで多くの学生が、ゲームプログラムを開発している。しかし、Windowsのバージョンやスペックの違いのため、パソコンによって同じソフトが使えないということが多く発生する。そのため、今回は、Windows SDKのみを使って制作する方法を模索することにした。Windows SDKは、マイクロソフト社が提供するソフトウェア開発キット（主に、ライブラリ）であり、Windowsパソコンでの汎用性が高い^[4]。これにより、ほとんどのWindowsパソコン上で動作するプログラムを、比較的容易に制作できると考えられるのである。

Windows SDKの扱いはDirect Xより簡単である。しかし、「英語教育ソフト」に使われる機能は、定型的なものであり、やはりライブラリ化がふさわしいと思われる。そこで、Windows SDKをカプセル化するライブラリを開発した。ただし、これは、極めて単純なものであり、Windows SDKを直接使うのと大きな違いはない。また、単純であるが故に、興味を持った学生がその内容を調べ理解することも容易であると考えられる。

ライブラリには、ボタンやエディットボックス等のコントロールを表すクラス（Button、Edit）、アプリケーションを表すクラス（KApp）を置く。また、「整数を文字列にする関数」（itos）などもある。そのヘッダファイルの概要はAppendixに示す。

このライブラリを使えば、WinMainおよびウインドウプロシージャには触らずに、アプリケーションクラスを書いていくことで、Windowsアプリケーションを制作することができる。

```

class App : public KApp
{
    // ボタン等はここに置く
public:
    App();
    void init(); // 初期化
    void onButton(int b); // ボタンが押された時の動作
    void onKeyDown(int k); // キーが押された時の動作
    void onLButtonDown(int x, int y); // マウスの左ボタンが押された時の動作
    void onMouseMove(int x, int y); // マウスが動いたときの動作
    void onLButtonUp(int x, int y); // マウスの左ボタンが離された時の動作
    void changeTo(int s); // 状態番号がsに変わるときの処理
    void show(HDC hdc); // アプリケーションの表示
};

```

たとえば、ボタンとエディットボックスを使いたい場合、上記の「// ボタン等はここに置く」に

```

    Button b_[5]; // 5つのボタン
    Edit ed_; // エディットボックス

```

と書き込み、initで次のように初期化すれば、ウィンドウに5つのボタンとエディットボックスが表示される。

```

void App::init() {
    for(int i = 0; i < 4; ++i) {
        // x座標、y座標が60 + 120*i、270、幅が100、高さが22のボタン
        b_[i].create(getHWnd(), 60 + 120*i, 270, 100, 22, "問題" + itos(i+1));
    }
    // x座標、y座標が240、310、幅が100、高さが22のボタン
    b_[4].create(getHWnd(), 240, 310, 100, 22, "全部");
    // x座標、y座標が60、70、幅が100、高さが22のエディットボックス
    ed_.create(getHWnd(), 60, 70, 100, 22);
}

```



図 2. 1 アプリケーションの表示

ボタンを押したときなどの処理は、

```
void App::onButton(int b) {
    if(ed_.checkEnter()) {
        // エディットボックスでエンターキーが押された時の処理
    }
    if(b == b_[0].getID()) {
        // b_[0]ボタンが押された時の処理
    }
    redraw(); // 画面の再描画
}
```

のように書くことができる。

また、アプリケーション画面への文字列の表示は、

```
void App::show(HDC hdc) {
    // x座標、y座標が0、0の場所に、「こんにちは」を表示
    mDrawText(hdc, 0, 0, "こんにちは");
}
```

のように行うことができる。

本ライブラリを使うと、「状態を持つオブジェクト」をSNumControlerの派生クラスにすることで、簡便に扱える。これが、本ライブラリの特徴の1つと考える。

状態は状態番号で区別する。特に、SNumControlerには、

```
void setSNumTimer(int s1, int s2, time_t w);
```

という関数をつけた。この関数を実行すると、状態番号が即座にs1に変わり、wミリ秒後にs2に変わる。このような処理は、一般に、ウィンドウプロシージャを使い、異なるオブジェクト間でデータのやり取りをしなければ実現できない。これを学生が実装する場合、多くの時間を費やすことになる。したがって、ライブラリにカプセル化したのである。

4. 試作品

学生諸君による実用ソフトの実装は2014年度よりはじめる予定である。その最終形は、おそらく、問題別演習、本番形式でのテスト演習、採点、成績処理、統計、弱点の指摘などの機能が盛り込まれることになるだろう。しかし、語学系学生のフィードバックを待ちながら、情報系学生が行うことになるため、開発プロセスとしては、漸進的、あるいは、スパイラルな方式を考えている。そのため、特に重要なことは、「英語学習ソフト」の全体構成を拡張可能にしておくことである。

ここでは、上記を踏まえた上で、「どのようなものができるか」を見るための試作品(プロトタイプ)を説明する。

プロトタイプでは、英検に合わせて問題を4種類に分類し、それぞれを問題1～問題4とする。これらの分類別に「英検1回分の(その種類の問題のみの)問題演習とその採点」を1セットとし、ユーザが望む回数分のセットを実行できるようにする。ただし、このような操作のはじめに問題選択画面が現れるようにする。

試作品は、次のようなものにした。

まず、起動すると名前を入力する画面となる。

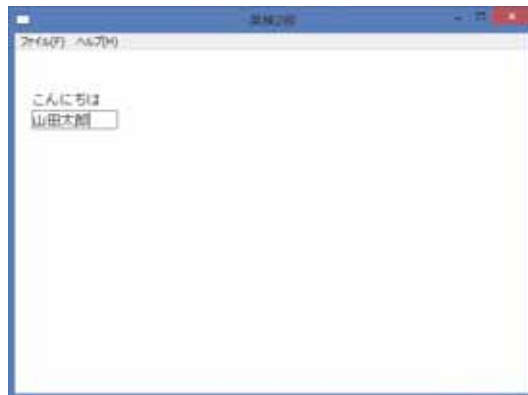


図3. 1 プロトタイプの起動と名前入力

この画面で名前を入力しエンターキーを押すと、問題選択画面になる。



図 3. 2 問題選択画面

ここから問題を選べる。たとえば、問題 1 を選ぶと「問題 1 に分類される問題のセットの番号」が複数現れ、そのどれかを選択すると、そのセットがはじまる。以下は、問題 1 のあるセットの画面である。

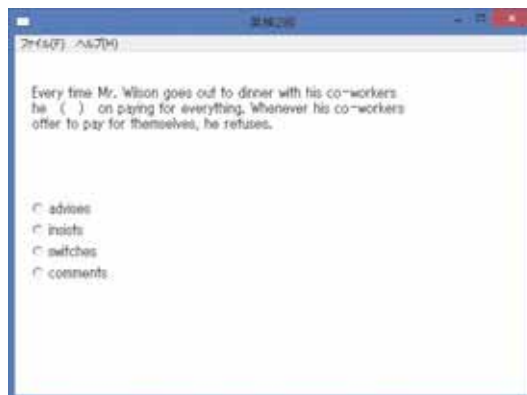


図 3. 3 問題1の画面

ここで選択肢を選ぶと、すぐに次の問題に進むようにした。その際、正誤判定をその場で表示することも、あとで表示することもできる。このような動作は他の問題でも同様である。また、1 セット終了すると、再び問題選択画面（図 3. 2）に戻るようにした。

問題 2 は、並べ替え問題であるが、プログラムでは、選択肢をドラッグアンドドロップして並べ替えを行うようにした。

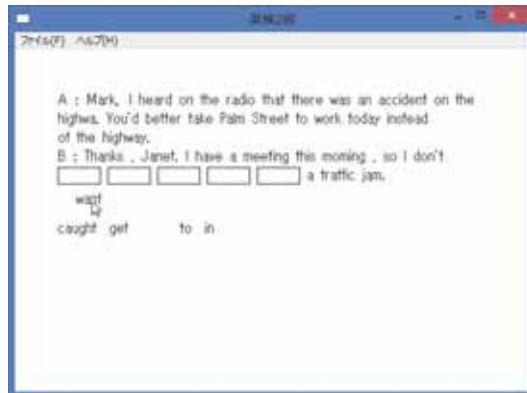


図 3. 4 問題 2 の画面 (「want」をドラッグ中)

問題 3 は長文中の単語選択であるため、長文と選択肢を別のウィンドウで表示するようにした。



図 3. 5 問題 3 の長文ウィンドウ



図 3. 6 問題 3 の選択肢ウィンドウ

問題 4 は問題 3 と同様である。

5. まとめ

Windowsアプリ用のライブラリを作成した。これにより、「本学学生が制作した英語教育ソフトを本学学生が使用する」という環境を構築し、プログラミング教育と英語教育の双方を進めることが可能になったと考える。

実際の運用は来年度からになるが、システム設計分析受講生、および法月ゼミのゼミ生に試作品を見てもらい感想を聞いた。統計的な集計にはいたらないが、「プログラマ」となるシステム分析受講生からは、「おもしろそうである」といったポジティブな反応が多かった。また、「ユーザ」である法月ゼミ生からも、「勉強に有用そうである」といったポジティブな反応があった一方、改善点の指摘も受けた。その主なものは、

- 1問ごとに正しい答を示してほしい。
(試作品では、正誤判定の事実しか示さない。)
- 1問ごとに正誤判定を表示しないでほしい。
- ユーザが望んだときにはヒントを出してほしい。
- ユーザが望んだときには解説を出してほしい。

であった。「1問ごとに正しい答を示してほしい」と「1問ごとに正誤判定を表示しないでほしい」は、矛盾する要求であるが、モードを切り替えられるようプログラムを書いていくことで解決できる。これらの要求は、「ほんもの」であり、それに答えるようプログラムを書いていくことは、プログラム系の学生にとってよい経験になるはずである。

Appendix Windows SDK をカプセル化するライブラリのヘッダファイル

```
// intをstringに変換する関数
std::string itos(int n);

// 文字列を表示する関数 (背景色デフォルト)
void mDrawText(HDC hdc, int x, int y, const std::string& t);

// 文字列を表示する関数 (背景色指定)
void mDrawText(HDC hdc, int x, int y, const std::string& t, COLORREF r);

// メッセージボックスを表示する関数
inline void mb(const std::string& m = "", const std::string& t = "") {
    MessageBox(0, m.c_str(), t.c_str(), MB_OK);
}
```



```

// コントロールの基底クラス
class Control
{
    ...
public:
    ...
    // ID番号を取得
    int getID() const { return id_; }
    // 表示する文字列のセット
    void setText(const std::string& n) { SetWindowText(hWnd_, n.c_str()); }
    // コントロールの移動
    void move(int x, int y, int w, int h) { MoveWindow(hWnd_, x, y, w, h, true); }
}

void show(bool s);           // 表示・非表示の切り替え
void enable(bool e);        // 有効・無効の切り替え
};

// ボタンを表すクラス
class Button : public Control
{
public:
    // コントロールを生成する
    void create(HWND hw, int x, int y, int w, int h, const std::string& n);
};

//ラジオボタンのグループを表すクラス
class RButtonGroup
{
    ...
public:
    // ラジオボタンを追加
    void addButton(HWND hw, int x, int y, int w, int h, const std::string& n);
    // n番目のボタンの文字列をセットする
    void setText(int n, const std::string& t) { rs_[n].setText(t); rs_[n].n_ = t; }
    void allOff();           // すべて非選択にする
    void show(bool s);       // 表示・非表示を切り替える
    void setOn(int n, bool o = true); // n番目のボタンのオン・オフを変える
    int getSelect() const;   // 選ばれてるボタンの番号を戻す
    std::string getSelectText() const; // 選ばれてるボタンの文字列を戻す
};

```

```

// 1行入力のエディットボックスを表すクラス
class Edit : public Control
{
    ...
public:
    void create(HWND hw, int x, int y, int w, int h, const std::string& n = "");
    std::string getText() const; // 書き込まれている文字列を戻す
    // Enterキーが押されたかチェックする (チェック後リセットする)
    bool checkEnter() const { bool b = enter_; enter_ = false; return b; }
};

// 複数行入力のエディットボックスを表すクラス
class MEdit : public Edit
{
public:
    void create(HWND hw, int x, int y, int w, int h, const std::string& n = "");
};

// リストボックスを表すクラス
class ListBox : public Control { ... }

// コンボボックスを表すクラス
class ComboBox : public Control { ... }

// ビットマップを表示するためのクラス
class BMOject { ... }

// アニメーションを表示するためのクラス
class AnimeObject { ... }

// wavファイルを再生する関数・止める関数
void playwav(const std::string& fn, bool lp = false);
void stopwav();

// 状態番号を管理するクラス
class SNumControler
{
    int snum_; // 状態番号
    ...
public:

```

```
...
int getSNum() const { return snum_; }
// 状態番号を即座にsにする
void setSNum(int s);
// 状態番号を即座にs1とし、wミリ秒後にs2にする
void setSNumTimer(int s1, int s2, time_t w);
};

//アプリケーションの基底クラス
class KApp : public SNumControler { ... }
```

謝辞

静岡産業大学情報学部の先生方に感謝致します。

文献

- [1] 日本英語検定協会 <http://www.eiken.or.jp/>
- [2] 小林健一郎『プログラミング教育におけるゲームプログラム』静岡産業大学情報学部紀要第15号（2013）p.1
- [3] マイクロソフト社
<http://msdn.microsoft.com/library/windows/apps/hh452744.aspx>
- [4] マイクロソフト社
<http://msdn.microsoft.com/ja-jp/windows/hardware/hh852363.aspx>