

情報技術科2年生対象C++言語プログラミングに関する教材の見直し

情報技術科 久保 雅俊

1 はじめに

従前の情報技術科のカリキュラムにおいて、プログラミング言語教育は1年次3Qから2年次2QまでC++言語とJava言語を学生の希望による選択制としていた。しかしながら、平成25年度より当時ソフトウェア開発系企業の多くが開発にJava言語を採用していたことから、1年次3Qに全員がJava言語を学習し、4Qから選択制とするように変更した。その際に、1年次の4Qの授業では、3Q4Qを通じて利用していた教材から抜粋し、基礎的な部分を学び、後半の応用的な部分は進捗状況により学生が個々に進めていく形とした。

ところが、年々の学生の学力低下に伴い、1年次に予定している授業内容の理解が及ばず2年次の授業にも影響が出始め1年次の復習に1か月を要するようになってしまった為、1年次の4Qから2年次の2Qまでの教材をすべて見直し、学生がしっかり理解できるような改定が必要となった。

平成24年度以前

1年次		2年次	
3Q	4Q	1Q	2Q
C++言語	→	→	→
Java言語	→	→	→

平成25年度以降

1年次		2年次	
3Q	4Q	1Q	2Q
Java言語	C++言語 →	→	→
	Java言語 →	→	→

図1 情報技術科における言語系選択授業

2 研究の目的

昨年度、主として学生の学力低下の為、情報技術科のC++言語選択コースの言語教育における実習内容の見直し、教材作成を計画した。しかしながら、時間の都合で1年次のものだけに留まってしまった。そこで、今年度は2年次の前期に行う授業で使用する教材を作成し、1年次から学習内容を連携させることにした。

3 授業内容の検討

従来は「図形処理」を題材として、フォトレタッチソフトウェアのような画像処理アプリケーションの作成を行ってきた。これには、写真加工の仕組

みの理解はもちろんのこと、中規模なアプリケーション開発の手法を見せる狙いがあった。更に教材は、ネットワーク通信による画像の送受信をアプリ内で行えるよう解説しているが、そこまで到達できる学生が皆無になった。(以前は、成績上位の学生が到達していた。)

以前は仕組みを理解して作成している学生が大半であったが、最近では言われるままになんとか作成する学生も増えてきた。それは、中規模なアプリケーション開発の前に、基本的なGUIの仕組みが理解できていない為と考えている。

そこで、最初は1年次に小規模なWindowsアプリケーションの開発を繰り返し行い、ゲーム要素も取り入れて興味を持たせ、GUIアプリケーション開発に少し慣れさせた上で、2年次に中規模アプリケーション及び図形処理に入っていくように実習内容を変更した。

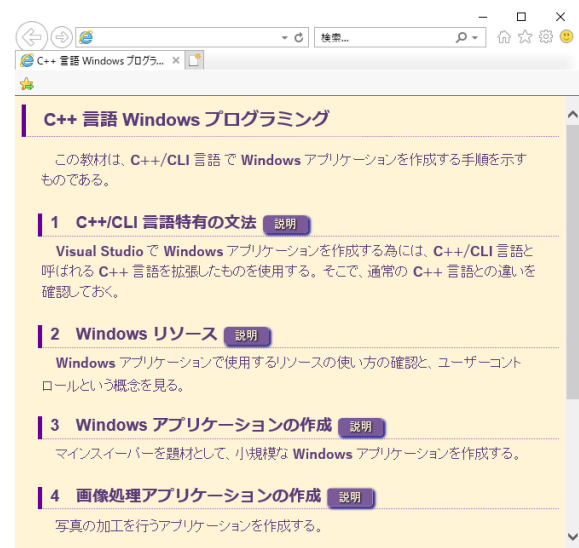


図2 教材の章立て

4 教材の特徴と学生の使用例

今回作成した教材の特徴を次に挙げる。

4.1 ゲームを題材とした中規模アプリケーション

ゲームを題材とすることで完成品に興味を持ってもらい開発の意欲を掻き立てることを狙った。具体的には次のような理由によりマインスイーパーにした。

① 頭脳系 (パズル系) である。

シューティング系のゲームは、ビジネス系 GUI アプリケーションを作成する上で参考になることが少ない。

② オブジェクト指向の良い適用例である。

ブロックをオブジェクト化して隣同士を接続することで、オブジェクト指向プログラミングが実現できる。

③ 外部仕様がある程度知られている。

今年度はコロナ禍ということもあり、5月下旬まで学生に自宅学習を強いることになった。その際に、Google マインスイーパーが誰でもプレイできるので、その外部仕様をまとめる課題を出題した。

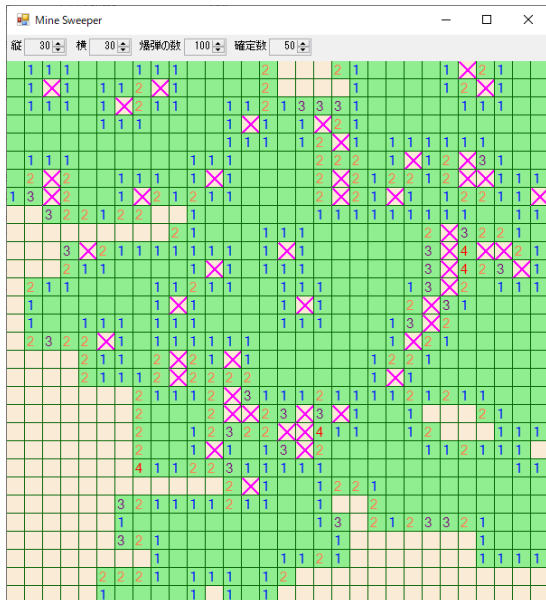


図3 完成したゲームアプリケーション

4.2 ヒント機能の多用

教材に関して、学生のレベル差をオプション課題による課題量で調整する形から、昨年度以降はヒントを段階的に提示する形に変えた。例えば、プログラミングにおけるヒントは次のように提示している。

第1段階 取っ掛かりとなる作業内容を列挙

第2段階 処理内容を言葉で説明

第3段階 ソースコードの穴開き問題を出題

この教材を授業で使用したところ、学生がヒント機能を上手に使う事例と悪く使う事例が見られた。

4.2.1 上手く使う例

現在の統合開発環境のエディターには単語補完機能があり、ソースコードの前後関係から類推し、コードを入力しているそばから単語単位での候補を列挙して選択できるようになっている。

そこで、穴開き問題のソースコードを穴が開いた状態のままエディターを使って打ち込み、その入力中に単語補完機能の支援を受けて答えを見つける学生が幾人か見受けられた。

そしてエディターに教わった候補の内、最適と思われるものを今度は教材側に入力して答え合わせを行う。

相互に補完しながら使う使い方はこちらが想定していなかったものであるが、これはこれで良いと考える。

プログラムを作成する際に関数等の仕様がうろ覚えの状態でもエディターの力を借りて見つけ出すことはよくあることで、そのような使い方が身につけば学生が単独で作成できるようになることが期待できる。

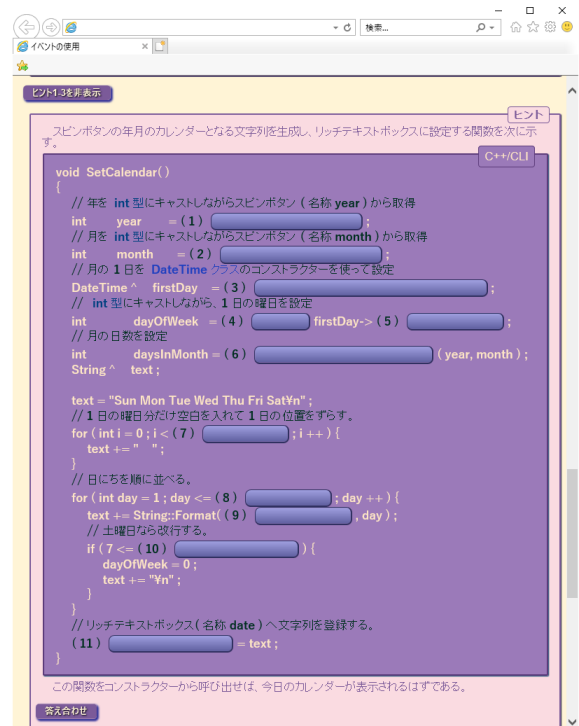


図4 ヒント機能の穴埋め問題例

4.2.1 悪く使う例

ヒントは最初すべて伏せた状態であり、ボタンを押すことで提示される。そして、最初のヒントがわからなければ次のヒントを開けることを想定している。しかしながら、学生の中にはとりあえず全部のヒントボタンを押して開けてしまう者がいる。

このように自分の力で作れるようになりたいという願望が希薄な学生を想定はしていたが、最初のヒントを読まずに直ぐに次のヒントを開けるようでは第2段階までのヒントの意味がなくなる。

よって、最初のヒントを開けてからある程度の時間が経たないと次のヒントが開かないようにしようと考えている。JavaScriptなら実現は可能であろう。

5 おわりに

今年度、授業で使用した反省点としては、前段となる説明や課題が多くなってしまったことと、新型コロナウイルスの影響で授業時間数が少なくなったことで例年より図形処理のアプリケーションに実装できた機能が少なくなってしまった点が挙げられる。

来年度は内容を精査し、最終目標となる図形処理アプリケーションに昨年度並みの機能を持たせるところまで授業を行いたい。