

論文 | Articles

アジア諸国からの留学生のためのIT人材育成教育

IT Human Resource Development Education for Foreign Students from Asian Countries

川本 勝

KAWAMOTO, Masaru

尚美学園大学

総合政策学部非常勤講師

Shobi University

2019年6月

Jun.2019

アジア諸国からの留学生のための IT 人材育成教育

川本 勝

IT Human Resource Development Education for Foreign Students from Asian Countries

KAWAMOTO, Masaru

Abstract

In response to the situation that there will be around 800,000 shortages of IT personnel in 2030, it is opened a new class called the latest programming exercise for foreign students from Asian countries at a business vocational school in Tokyo.

As a result, it was revealed that IT human resource development education is effective even for foreign students from Asian countries.

要 約

「2030年にはIT人材がおおよそ80万人不足する」という情勢を受けて、アジア諸国からの留学生に向けた最新プログラミング演習と称する新たな科目を都内のビジネス専修学校で開講した。

その結果、アジア諸国からの留学生でも、IT人材育成教育が効果のあることが解った。

キーワード

アジア諸国からの留学生 (Foreign Students from Asian Countries)

IT人材教育 (IT Human Resource Development Education)

プログラミング教育 (Programming education)

フローチャート (Flow Chart)

スクラッチ (Scratch)

1.はじめに

経済産業省（2016）の平成27年度調査研究レポート「IT人材の最新動向と将来推計に関する調査結果」によれば、「IT（Information Technology：情報技術）人材は2030年にはおよそ80万人が不足する」とされている。

これを受けて、文部科学省（2018）は2020年度から小学校でもプログラミング教育を実施する施策を始めた。

一方、文部科学省（2016）の「小学校段階におけるプログラミング教育の在り方について（議論の取りまとめ）」では、「プログラミング教育とは、子供たちに、コンピュータに意図した処理を行うよう指示することができるということを体験させながら、将来どのような職業に就くとしても、時代を超えて普遍的に求められる力としての『プログラミング的思考』などを育むことであり、コーディングを覚えることが目的ではない」と、提言されている。

しかしながら、2020年度に小学校に入学した児童が大学を卒業して社会に出るのはおよそ16年後である。しかも、小学校でプログラミング教育を受けた児童の全てがITエンジニアに育って社会に出るわけではない。

詰まり、文部科学省が新たに進める小学校段階におけるプログラミング教育には、経済産業省が報告した「IT人材は2030年にはおよそ80万人が不足する」に対する即効性が有るわけではない。

一方、文部科学省ほか（2008）は早くから「留学生30万人計画」を実施しており、独立行政法人日本学生支援機構（2017）の「平成29年度外国人留学生在籍状況調査結果」によれば、「平成29年5月1日現在の留学生数は、267,042人で、その93%はアジア系であり、また、同じく、73%は人文・社会・教育の文系である」と、報告されている。

そこで、筆者は、2030年までの短期間にIT人材を大量養成する必要性を考えた場合、

文系のアジア系留学生をIT分野における人材源の一つとする考えも有り得る

と考える。

詰まり、IT系ではない文系のアジア系留学生が日本での留学期間中にIT業界に職業としての興味を持ち、IT人材として必要な基礎知識や基礎技術を習得して、IT業界に就職してくれるようになれば、IT人材不足解消の即効性の有る一助になる、と筆者は考える。

上記のような情勢を受けて、筆者の出講先である東京都内の2年制ビジネス専修学校（専門課程）でもカリキュラムを見直し、Word、Excel、PowerPoint、Access、コンピュータ概論、HTML、JavaScriptなどといった既に開講していたIT系科目に加えて、2017年度から筆者を担当者として新たに最新プログラミング演習と称する科目を開講した。

この授業では、文部科学省の提言に従って、コンピュータ言語の言語教育では無く、プログラミング的思考を育むことを目的とし、処理の流れに着目した思考力を育む道具としてフローチャートを用い、使用するコンピュータ言語には敢えてコーディングの不要なScratch（スクラッチ）

を用いた。

ところで、文部科学省が進める新たなプログラミング教育に関して、ビジネス専修学校のアジア系留学生に対する先行研究は、特に、筆者が所属する日本教育工学会にジャストマッチするものは無い。

日本教育工学会では、ようやく、「小学校プログラミング教育」(日本教育工学会2018)というセッションが設けられて研究が花盛りになり始めたところである。

筆者が担当する、この最新プログラミング演習と称する新たなプログラミング教育についての当初の研究成果の一部分は2018年度の日本教育工学会(川本勝2018)にて既に報告済みであるが、学会発表では、その紙面に制限(2ページ)があるため、その報告は研究全体の極一部分のみに止まらざるを得なかった。また、筆者が所属する日本教育工学会の場合、発表原稿のページ制限は学会誌でも同様(8ページ)である。

従って、筆者の研究の詳細な現状を以下に報告する。

2. フローチャート

ここでいうフローチャート(flow chart)とは、コトバンクのASCII.jpデジタル用語辞典(2019)によれば、「データの流れや問題解決の手順を表す図式。流れ図とも呼ばれる。プログラムの設計や、業務の流れなどを図示する場合に利用される。複雑なプログラムや業務の工程を作成するときは、どの順番でどんな処理をするのかを明確にする必要がある。フローチャートを作成することで情報の流れを視覚化し、整理・把握しやすくなる。フローチャートでは、基本となる処理を長方形、条件分岐する処理は菱形など、特別な処理を意味する記号を組み合わせで表現する。どの処理に対してどの記号を使うかはJIS規格に定められている。」とある。

また、コトバンクのブリタニカ国際大百科事典小項目事典(2019)によれば、「流れ図。コンピュータのプログラムを書くときは、まず問題を分析して答を求めるための一連の手順(アルゴリズム)を明確にし、次にこれを決められた記号を用いて図式的に表現する。それをフローチャートという。これには精粗さまざまなレベルがあるが、最終的にはできるだけ詳しいフローチャートを書き、それからほとんど一対一にプログラムが書けるようなものを作る。こうすることによりプログラムのための能率が上がり、誤りが少なく、またドキュメンテーション保持のために役立つ。」とある。

下記の図1は、2017年度の後期に筆者が留学生に制作させたフローチャートの課題の1例である。

3. スクラッチ (Scratch)

ここでいうScratch(2018)とは、2006年頃にマサチューセッツ工科大学のメディアラボで開発されたコーディングレスなプログラミング言語の学習アプリのことである。

コーディングレスにプログラミング学習が出来ることで、学習者は、所謂、デバッグ(debug)作業と呼ばれるコーディング時の誤字・脱字・用法の間違いなどの煩雑な訂正作業から解放され、プログラミング言語自体の理解に集中できるというメリットがある。

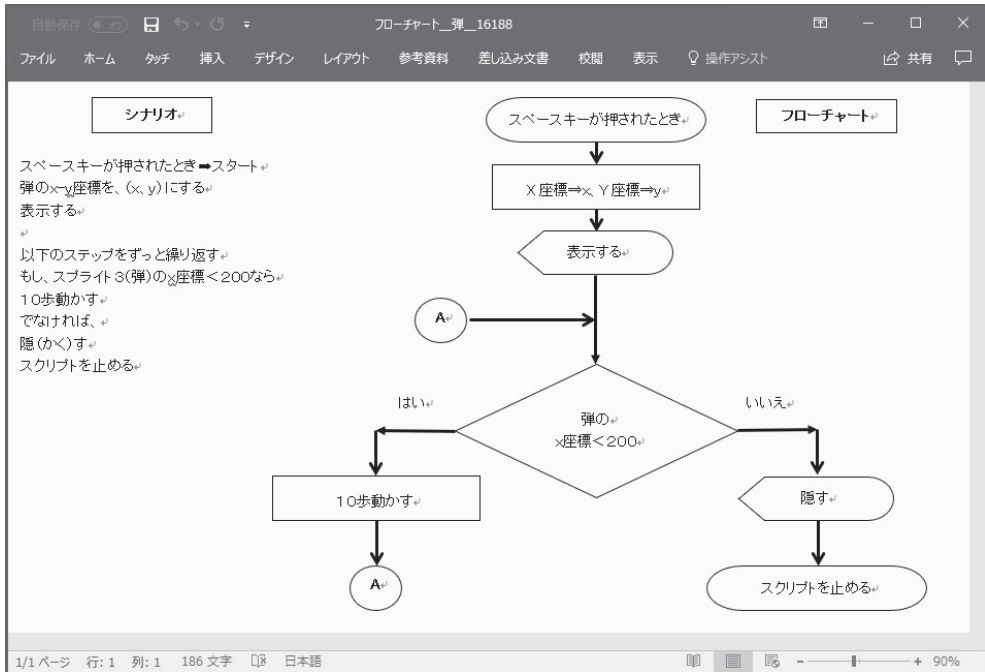


図1 筆者が留学生に制作させたフローチャートの課題の1例

図2は、2017年度の後期に筆者が留学生に制作させた課題の1例で、図1のフローチャートを基にプログラミングしたもので、PCの画面に表示されるScratchのWindow画面の全体である。図2の例では、Scratchのバージョンは1.4（オフライン・エディター版）である。

Scratchのプログラムに登場するキャラクタ（ないしはオブジェクト；Object）は、Scratchではスプライト（Sprite）と呼ばれ、図2に示されたScratch画面の右上部分にあるステージ（Stage）上で動作する。

スプライトはシナリオ（Scenario）をプログラム化したスクリプト（Script）と呼ばれるプログラムに従って動作する。Scratchのスクリプトは、図2に示されたScratch画面の左部分にあるブロックパレットから処理のブロックを選んで画面中央のスクリプトエリアにドロップして組み合わせることでコーディングレスに作成することが出来る。

スプライトは複数の設定が可能で、スプライト別にスクリプトを組む。Scratchのスクリプトは、イベントドリブン（Event-driven）型である。

また、各スプライトには複数のコスチューム（Costume）を設定することが可能で、コスチュームを入れ替えることによって登場キャラを交代させたり、複雑なアクションを表現したりすることができる。

同じく、Scratchでは、ステージの背景画像と背景音楽ならびにキャラクタが発する声や台詞なども自由に設定することが出来る。



図2 筆者が留学生に制作させたScratch課題の1例

更に、Scratchでは、図3のように、そのWindowに表示される文字の言語を自由に設定できる機能（以下、この論文では、これをScratchの「多言語機能」と呼ぶ）があり、利用者の母国語に合わせる事が可能である。この場合、入力文字の言語を表示言語に合わせれば、利用者の母国語に合わせた台詞の表示も可能である。

同様に、背景音乐ならびにキャラクターが発する声なども、利用者自らが母国語のものを使用することができる。

4. 授業の設計

筆者が新たに担当した最新プログラミング演習という科目では、先にあげた文部科学省の提言「プログラミング的思考を育む教育」に従って、コーディングを排除し、プログラミング的思考を育む授業を目指すために、プログラミング言語としては、コーディングが不要なプログラミン



図3 中国語簡体字で表示したScratchのエディター画面の1例
(註) この図は、筆者が独自に作成したものである。

グ言語の学習アプリである Scratch を選んだ。

一方、実習課題には、ゲーム・プログラミングを選ぶことで、アジア系の留学生にIT業界に対する職業としての興味を持たせたい、と筆者は考えた。

更に、コンピュータ言語の文法から始めるという授業展開はせずに、IT系ではない文系のビジネス専修学校生でも短期間で興味を引き出せるように、例えば、

スペースキーを押せば弾が発射されるには、どのような処理をプログラムすれば良いか、というように、ゲームの機能毎に1回に1テーマで、しかも、学習した内容が同じ授業時間内に直ちに目に見える形に出来上がるように授業時間の配分と授業の展開に留意し、授業が進むに従ってゲームが出来上がって行くというように、実習課題を連続物として設計した。

特に、プログラミング的思考を育むために、

シナリオの制作→シナリオのフローチャート化→Scratchを用いたプログラミングという工程（具体的には、図1から図2を作る）を最重要視して、

フローチャートを使ったシナリオの間違いチェック

を通じて、プログラミング言語の文法やコーディングなどは全く意識せずにプログラミング的思考力が育成されることを狙った。

なお、そのための便利な道具として、フローチャートの制作には、特に、Wordに内蔵されているフローチャートのテンプレート（図4）を利用させ、フローチャートの作図についての課題も設けて学習させた。

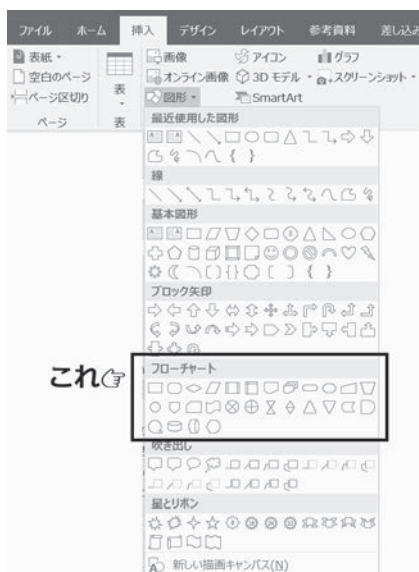


図4 Wordに添付されたフローチャートのテンプレート
(註) この図は、筆者が独自に作成したものである。

2017年度に筆者が担当した最新プログラミング演習の授業は、週1回90分のPC実習授業を前期・後期に各15回の合計30回、ただし、前期・後期ともに各15回目は定期実技試験と授業アンケートの実施、という設定で2年生のアジア系留学生4クラスに開講した。

一方、上述のような授業内容にマッチした市販の教科書は見つからなかったため、PC実習教室内のLAN上でのみ閲覧可能なWeb形式のテキスト（筆者は、これをe-Textと呼んでいる）を、筆者がオリジナルに制作してPC実習教室内のWebサーバーに設置し、授業に使用した。

なお、受講学生が制作した課題の成果物は、PC実習教室内のLANを介して教卓下にあるLANサーバー内の課題提出フォルダに提出され、教卓上の教師用PCで担当教師である筆者によりチェックされるが、その様子はPC実習教室内の大型モニターTV（4K-TV）に映し出されて、受講している留学生全員がリアルタイムにその場で見る事が出来る。

授業の効果は、前期・後期の各15回目に定期実技試験時に合わせて実施した授業アンケートの回答を分析することで評価した。授業アンケートの集計と分析にはマイクロソフト社の表計算ソフトであるExcelを用いた。

5. 実習課題の設計

実習課題では、数多いゲームの種類の中から、所謂、シューティングゲームを例にとりて制作させた。ここにいうシューティングゲームとは、弾を敵に当てて得点するゲームの事である（図2）。

シューティングゲームを選んだ理由は、単に筆者が実習課題を開発できるプログラム能力の範囲内であったからである。教師の開発能力以上のプログラムを実習課題に設定すると、受講している留学生のフォローに難が生じる場合があるからである。

一般に、ゲームソフトを開発するには、

- ① キャラクタ（character；ゲームのヒーローないし主人公や敵のキャラクタ）
- ② シナリオ（scenario；ゲームが進む台本、スプライトの台詞や音声も含む）
- ③ 背景画像（background picture；ステージに設定される）
- ④ BGM（background music；ステージに設定される背景音楽）

などを最初に考える。

今回は、著作権などのトラブルを予防するために、あえて、スプライトのコスチュームや音声、背景画像、背景音楽などの全てはScratchに最初からフリーで添付されているものを使った（図2）。

同様に、留学生が自分で制作したゲームを不用意にインターネット上に公開してトラブルになることを防ぐために、あえて、Scratch 1.4（オフライン・エディター版）を使用した（図2）。

従って、留学生が独自に制作する部分は、ゲームのシナリオだけである。ただし、シナリオには各スプライトの台詞や音声も含まれる。このシナリオは、Word文書で作成し、それを元に、Wordに添付されているフローチャートのテンプレート（図4）を使用してフローチャートを作成させ、シナリオの校正はフローチャートの校正に転嫁できる事を学習させた。

シナリオからフローチャートを作成した後にフローチャートを校正することでシナリオを校正するという課題（具体的には、図1を制作する課題）を繰り返し実習することで、この科目の目的であるプログラミング的思考を育むことを試み、その課題の得点を時系列に並べることでプログラミング的思考を育むことの計量化を試みた。

今回の課題であるこのシューティングゲームの内容は、図2のように、

- ①キャラクターが登場して、
- ②弾を投げ、
- ③敵に当たれば得点する。
- ④敵は、弾が当たれば、「ミャー」と鳴いて、消滅する。
- ⑤敵を全てやっつければ、ゲームは終わる。
- ⑥このゲームの主人公（ヒーロー）はプレイヤー。

とした。

ここで、図1は、図2を制作した留学生がWordで制作した同じスプライト3のシナリオとフローチャートである。従って、図2中のスプライト3のスクリプトは、図1のフローチャートから制作されたものである。

6. アンケート結果

筆者が2017年度に最新プログラミング演習の授業を担当したクラスは、IT系ではないビジネス専修学校のアジア系留学生クラスの2年生の4クラスで、留学生の合計は84名であった。

授業の効果を評価するために、前期・後期各15回目の定期実技試験時に合わせて授業アンケートを実施した。

授業アンケートの集計と分析にはマイクロソフト社の表計算ソフトであるExcelを用いた。

そのアンケートの回答率が表1である。ここで、「回答無し」とは、アンケート当日に欠席したことを意味している。従って、以下の各アンケートの集計値は、これらの数値が母数となる。

表1 アンケートの回答率

	回答有り(出席)		回答無し(欠席)	
	人数	比率	人数	比率
前期	82	97.6%	2	2.4%
後期	80	95.2%	4	4.8%

(註) この表は、授業で実施した授業アンケートの結果から筆者が独自に作成したものである。

同じく、留学生の母国別シェアは、下記の図5の通りである。

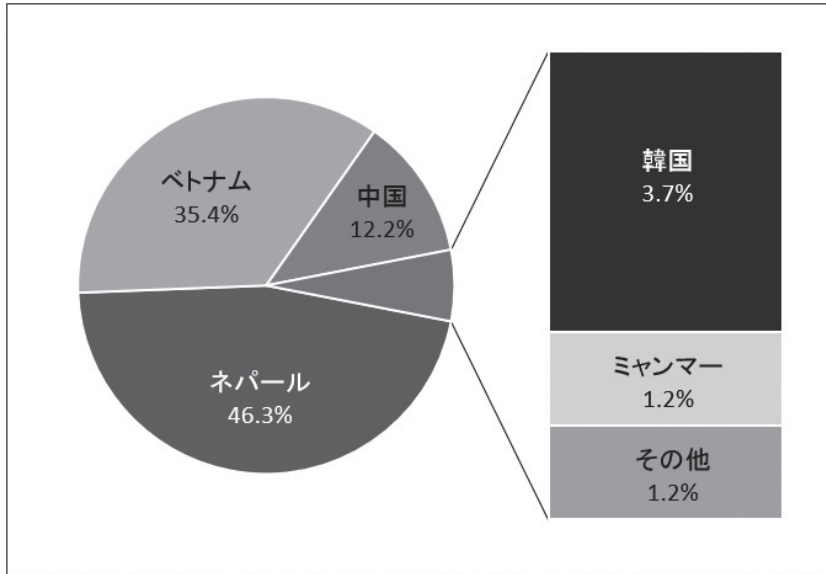


図5 留学生の母国別シェア

(註) この図は、授業で実施した授業アンケートの結果から筆者が独自に作成したものである。

6.1. Scratch (最新プログラミング演習) の授業は楽しかったか

前期と後期のアンケートで、それぞれ、「最新プログラミング演習 (Scratch) の授業は楽しいか」、尋ねた結果が下記の図6である。

前期 (図中の「前回」) では、回答した留学生の内、スゴク楽しい：43.6%、少し楽しい：44.9%で、合わせて88.5%が肯定的な回答であった。

同じく、後期 (図中の「今回」) では、回答した留学生の内、スゴク楽しい：55.1%、少し楽しい：37.2%で、合わせて92.3%が肯定的な回答であった。

特に、前期に比べ後期の方が、「いいえ」や「少し楽しい」が減って、「スゴク楽しい」が大幅に増えた。

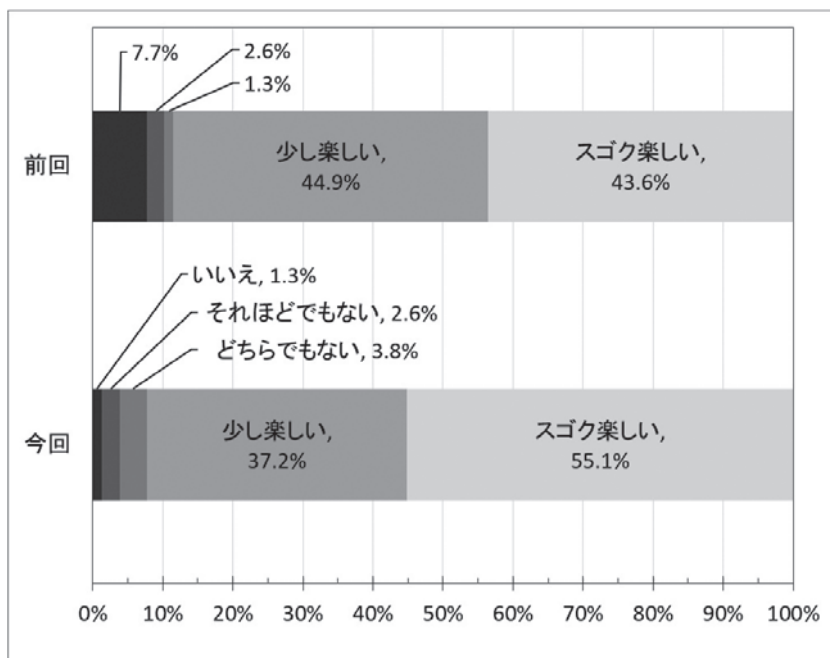


図6 最新プログラミング演習 (Scratch) の授業は楽しいか

(註) この図は、授業で実施した授業アンケートの結果から筆者が独自に作成したものである。

同じく、後期のアンケートで、Word、Excel、PowerPoint、HTML、JavaScript、Scratchなどの、この学校で開講されているIT系科目の中で、「どの科目が一番楽しかったか」を択一式で質問した結果が表2である。その結果、Scratch（最新プログラミング演習）が全体の2番目で、JavaScriptやHTMLのコーディングを必要とする科目よりも上位になった。

この結果から、IT系ではない文系のアジア系留学生にとっては、HTMLやJavaScriptのプログラミング・コードを手入力してWebページを制作するよりもプログラミング・コードの手入力が無いScratchでゲーム・プログラミングをする方が楽しいと感じていることが解った。

表2 IT系科目の好感度順位表

No	科目名	人数	比率
1	Excel	24	30.00%
2	Scratch	21	26.25%
3	JavaScript	13	16.25%
4	PowerPoint	11	13.75%
5	Word	5	6.25%
6	HTML	4	5.00%
7	わからない	2	2.50%
合計		80	100.00%

(註) この表は、授業で実施した授業アンケートの結果から筆者が独自に作成したものである

6.2. Scratchの認知度

前期のアンケートでは、「この授業で使用したScratchを事前知っていたかどうか」、質問してみたところ、回答者82名中、1.いいえ；51名（62.2%）、2.はい；31名（37.8%）の結果であった（図7）。つまり、留学生の過半数は、事前にScratchを知らなかった。

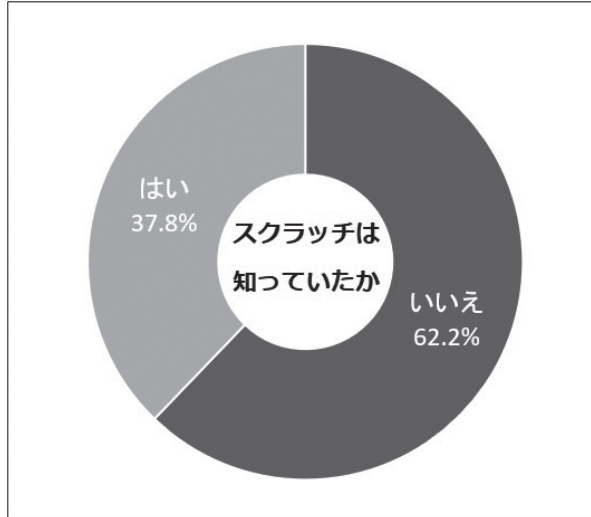


図7 スクラッチの認知度

（註）この図は、授業で実施した授業アンケートの結果から筆者が独自に作成したものである。

6.3. ゲームソフトの仕事の経験は有るか

前期のアンケートで、ゲームソフトの仕事の経験は有るか、質問してみたところ、回答者82名中、

1. いいえ；72名（87.8%）、
2. 母国で経験有り；3名（3.7%）、
3. 日本で経験有り；2名（2.4%）、
4. 現在、日本でアルバイト中；3名（3.7%）、
5. 回答無し；2名（2.4%）、

の結果であった（図8）。

この結果を、「はい／いいえ」で集約すると、

- いいえ；72名（87.8%）、
- はい；8名（9.8%）、
- 回答無し；2名（2.4%）、

に集約され、ほとんどの留学生はゲームソフトの仕事の経験は無かった。

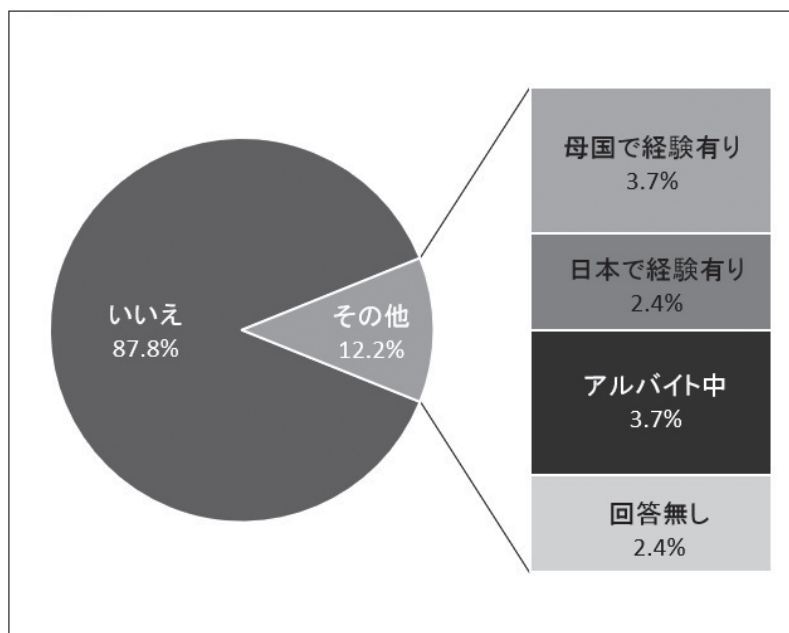


図8 ゲームソフトの仕事の経験はあるか

(註) この図は、授業で実施した授業アンケートの結果から筆者が独自に作成したものである。

6.4. 職業としてIT業界に興味があったか

前期・後期のそれぞれに、IT業界についての職業としての興味度を、「①いいえ、②知りたい、③体験したい、④アルバイトしたい、⑤就職したい」、の5段階に区分した択一式で尋ねた(図9)。回答数は、前期が82名、後期が80名であった。

「就職したい～知りたい」の肯定的な回答が前期・後期共に大勢を占める(前期：80.5%、後期：85%)中で、前期に比べて後期の方が、「就職したい(11.0%→16.3%)」や「体験したい(19.5%→32.5%)」が更に増加した。

この結果は、IT系ではない文系のアジア系留学生に対するIT人材養成のための新たなプログラミング教育の授業としてフローチャートとScratchを選び、課題の題材としてゲーム・プログラミングを選んだ組み合わせに少なくとも基本的な間違いは無かったことを示しているものと、筆者は考える。

ただし、HTMLやJavaScriptで同様のアンケートを行っていないので、HTMLやJavaScriptとの直接的な比較は出来ない。

6.5. Scratchの多言語機能の効果

今後の授業にフィードバックさせるために、前期のアンケートでは、Scratchの多言語機能の利用度についても質問した(図10)。

その結果は、

「日本語：71.3%、英語：20.0%、母国語：7.5%、その他：1.3%」

であった。

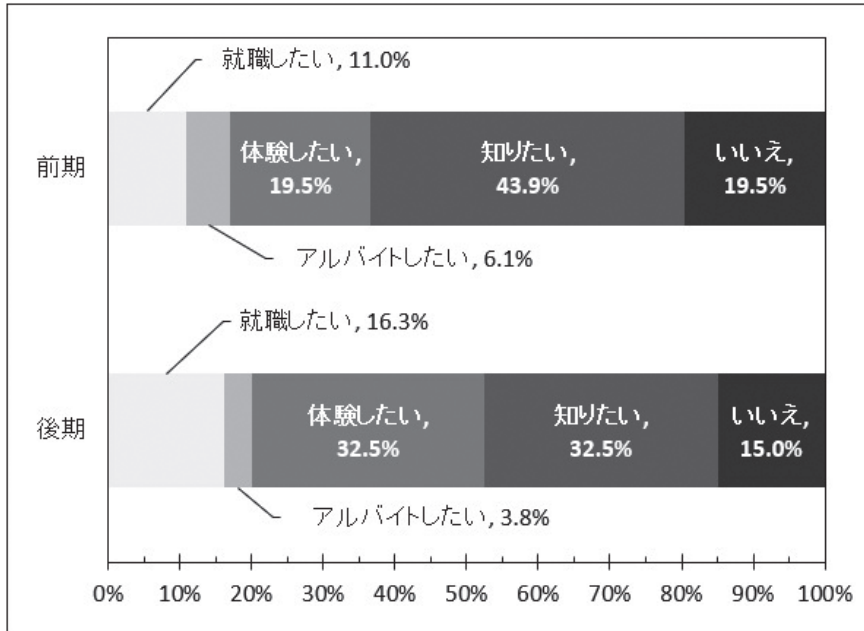


図9 IT業界への興味度

(註) この図は、授業で実施した授業アンケートの結果から筆者が独自に作成したものである。

詰まり、今回の受講生である留学生達の大半は、母国語よりも日本語でScratchを使用していた。今回のアンケートに特に限定されたものではある（広く一般的にいえる結果ではない）が、この結果から、特に、留学生の母国語が使用できなければITエンジニア養成の授業が成立しないわけではない可能性はあるといえる。特に、アジア系の留学生に対し日本語のみでITエンジニア養成の授業が可能であるとなれば、その効果は大きい可能性がある。

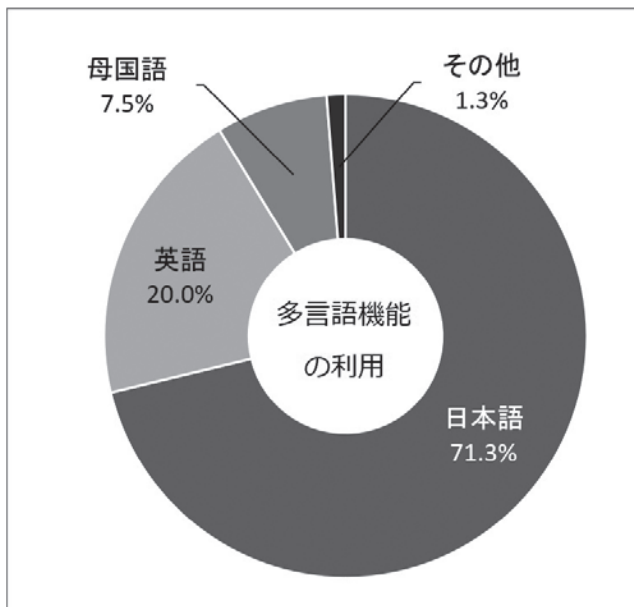


図10 Scratchの多言語機能の利用

(註) この図は、授業で実施した授業アンケートの結果から筆者が独自に作成したものである。

6.6. e-Textの内容は良かったか

同じく、今後の授業にフィードバックさせるために、後期のアンケートでは、授業に使用したe-Textの内容が良かったか聞いてみた。その結果を集計したものが、表3である。同じ結果をドーナツグラフで表したものが、図11である。

「すごく良かった：53.8%」、「少し良かった：32.5%」の、合わせて86.3%が肯定的な回答であったが、「プリントの方が良い：6.3%」、「市販本の方が良い：7.5%」という回答もあった。

表3 e-Textの内容は良かったか

評価	人数[人]	比率
すごく良かった	43	53.8%
少し良かった	26	32.5%
わからない	0	0.0%
板書の方が良い	0	0.0%
プリントの方が良い	5	6.3%
市販本の方が良い	6	7.5%
合計	80	100.0%

(註) この表は、授業で実施した授業アンケートの結果から筆者が独自に作成したものである。

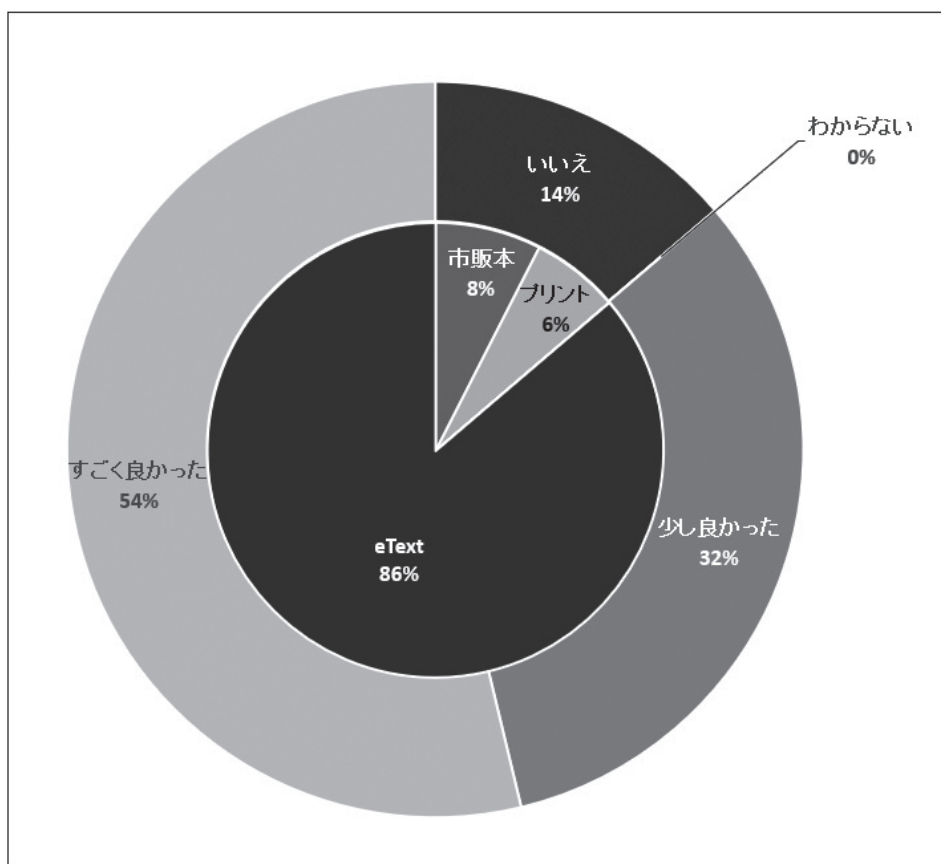


図11 e-Textの内容はよかったか

(註) この図は、授業で実施した授業アンケートの結果から筆者が独自に作成したものである。

7. 結論と今後の課題

今回の成果は、筆者が担当した留学生のみに限ったもので、広く一般的に検証されたものではないが、Scratchも知らず、IT業界やプログラミング言語には興味が無くてもおかしくないと思われる、IT系ではない文系のアジア系留学生にとっても、フローチャートとScratchを使用したゲーム・プログラミングの授業は楽しく感じていることが、今回のアンケートの結果（前期は88.5%が肯定的な回答、後期は92.3%が肯定的な回答で、前期に比べ後期の方が「スゴク楽しい」が大幅に増えた）から解った。

同じく、コーディングのあるHTMLやJavaScriptよりもコーディングの無いScratchでプログラミングをする方が楽しいと感じていることも解った。

更に、ほとんどの留学生はゲームソフトの仕事の経験は無かったにもかかわらず、「就職したい〜知りたい」の肯定的な回答が前期・後期共に大勢を占めた（前期：80.5%、後期：85%）中で、前期に比べて後期の方が、「就職したい（11.0%→16.3%）」や「体験したい（19.5%→32.5%）」が更に増加した。

以上の結果から、IT業界への興味を引く上でも、フローチャートとScratchを用いてゲーム・プログラミングする授業に、少なくとも基本的な間違いは無かった、と筆者は考える。

一方、大半の留学生は母国語（7.5%）よりも日本語（71.3%）でScratchを使用していることから、今回の留学生に限定された結果ではある（広く一般的にいえる結果ではない）が、特に、留学生の母国語が使用できなければITエンジニア養成の授業が成立しないわけではない可能性はあるといえる。特に、アジア系の留学生に対し日本語のみでITエンジニア養成の授業が可能であるとなれば、その効果は大きい可能性がある。

なお、筆者が独自に制作して使用したe-Textについては、86.3%が肯定的な回答であった。

しかし、フローチャートを用いてプログラミング的思考を育むことの計量化については、現在、集計と分析の途中で、その結果は、今後を追って発表する予定である。

一方、この授業は未だ開講間もなく、特に、就職の実績が上がっているわけではない。従って、今後は、この授業を通じて、IT系ではない文系のアジア系留学生が実際にIT業界に就職の実を上げるには授業をどのように展開するのがより妥当で効果的であるのかを更に追求し続けなければならない。

そのためには、HTMLやJavaScriptといったコーディング系の科目も含めて、アンケートの設問内容の熟成だけでなく、前期・後期の期末に実施している定期実技試験や期中に行っているミニテストないしは実習課題などの結果も計量化して授業にフィードバックする手法を開発することなども今後の重要な課題である。

文部科学省が進める新たなプログラミング教育について、筆者は、まだ研究の途についたばかりであるが、この報告が今後と同様の授業を展開される諸先生方の参考になれば幸いである。

参考文献

- ASCII.jp デジタル用語辞典、「フローチャート」、『コトバンク (ASCII.jp デジタル用語辞典)』、(株) 朝日新聞出版、2019、[https://kotobank.jp/word/ フローチャート](https://kotobank.jp/word/フローチャート) (accessed 2019.2.20)
- ブリタニカ国際大百科事典小項目事典、「フローチャート」、『コトバンク (ブリタニカ国際大百科事典小項目事典)』、(株) 朝日新聞出版、2019、[https://kotobank.jp/word/ フローチャート](https://kotobank.jp/word/フローチャート) (accessed 2019.2.20)
- 川本 勝、「アジア諸国からの非IT系留学生のためのIT人材育成教育」、『日本教育工学会第34回全国大会』、日本教育工学会、2018、p.481-p.482
- 経済産業省、「IT人材の最新動向と将来推計に関する調査結果」、2016、https://www.meti.go.jp/policy/it_policy/jinzai/27FY/ITjinzai_report_summary.pdf (accessed 2018.02.24)
- 文部科学省ほか、「留学生30万人計画 骨子」、2008、https://www.jsps.go.jp/j-kokusaika/data/meibo_siryou/h21/06_kosshi.pdf (accessed 2018.02.26)
- 文部科学省、「小学校段階におけるプログラミング教育の在り方について (議論の取りまとめ)」、2016、http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/074/siryu/_icsFiles/afieldfile/2016/07/07/1373891_5_1_1.pdf (accessed 2018.02.25)
- 文部科学省、「未来の学びコンソーシアム 小学校プログラミング教育必修化に向けて」、2018、https://miraino-manabi.jp/assets/data/info/miraino-manabi_leaflet_2018.pdf (accessed 2019.02.25)
- 独立行政法人日本学生支援機構、「平成29年度外国人留学生在籍状況調査結果」、2017、https://www.jasso.go.jp/about/statistics/intl_student_e/2017/index.html (accessed 2018.02.26)
- 日本教育工学会、「SIG-14 小学校プログラミング教育」、『日本教育工学会第34回全国大会』、日本教育工学会、2018、http://www.jset.gr.jp/taikai34/announcement_app/announcement_app.html (accessed 2018.09.01)
- Scratch、<https://scratch.mit.edu/> (accessed 2019.02.25)