

卒業論文 2017 年度 (平成 29 年度)

Learning Style Inventory を用いた
プログラミング初学者のための
教材レコメンドシステムの提案

Proposal of an Learning Style Inventory-based
Programming Teaching Material
Recommendation System for Beginners

慶應義塾大学 総合政策学部

押見 太雄

Learning Style Inventory を用いた プログラミング初学者のための 教材レコメンドシステムの提案

本研究では、プログラミング初学者が教材選択や学習スタイルの確立の困難さから、入門初期段階で挫折してしまう問題に対して、個人に合ったプログラミング学習教材を提案する、教材レコメンドシステムの提案と開発を行った。

プログラミング初学者が入門初期段階で挫折してしまう原因として、初学者のリテラシレベルや習熟度に合わない「学習教材」と「学習手法」を選んでしまうことが挙げられる。これらの問題は、初学者がどのようにしてプログラミングを学習していけばいいかわからないことに起因している。

そこで、本研究では、プログラミング入門時に挫折する学習者を減らすために、システムによって学習者の学習スタイルを分類し、個人に最適な学習手法と学習教材を提案することを目的とする。先行研究より、学習スタイル別に学習方法を分ける手法は、他科目において短期間で習熟度を向上させることに効果があるとされている。

本システムでは、学習スタイルの判別手法である LSI(Learning Style Inventory) を使用して、学習者の学習スタイルを、「Visual タイプ」、「Auditory タイプ」、「Kinesthetic タイプ」に分類し、判別された学習スタイルに沿った教材を学習者に提供する。

本研究の有効性に関して、プログラミング学習における学習スタイルは LSI で判別可能であること、初学者に対して適切な教材と学習手法を提供することで学習効果を向上させることが可能であることの検証を行うため、プログラミング初学者を中心とした大学1年生以上の28名に対し、本システムを用いた実証実験を行った。本実験では、本システムおよび教材を用いて、被験者に問題を出してその正答率を取得し、実験終了後に被験者に対して、実験に関するアンケートを行った。

結果として、学習者における、LSI を用いて判定された学習スタイル及び提示された教材の納得度、満足度共に肯定的ではあるが、とても好評とは言えない。原因としては、「Auditory タイプ」に分類された被験者の納得度が低かった。しかし、初心者がプログラミング学習をする場合、学習スタイルの納得度と学習の満足度と正答率が相関していたため、正しい学習スタイルに判定された被験者には効果があると言える。

キーワード

1. プログラミング教育, 2. プログラミング初学者, 3. 学習スタイル,
4. Learning Style Inventory, 5. 教材のレコメンド

慶應義塾大学 総合政策学部

押見 太雄

Proposal of an Learning Style Inventory-based Programming Teaching Material Recommendation System for Beginners

This research proposes and develop a teaching material recommendation system that is suitable for each learner in order to dealing with the problems where beginners find it difficult in selecting materials and learning styles.

One of the reasons why the first scholar of programming is frustrated at the initial stage is to choose learning materials and learning methods that the first scholars do not agree with. These problems are caused by the fact that the first scholar does not know how to learn programming.

In this research, in order to reduce learners who are frustrated at the beginning of programming, we aim to classify the learning style of learners by the system and to propose learning methods and learning materials optimum for individuals. A method of separating learning methods from previous studies by learning style can improve proficiency level in other subjects in a short period of time.

In this system, learning styles of learners are classified into Visual type; Auditory type; and Kinesthetic type by using LSI which is a learning style discrimination method, Provide learning materials to the learning style according to the learning style.

Regarding the effectiveness of this research, it is verified that learning styles in programming learning can be discriminated by LSI and that it is possible to improve learning effect by providing appropriate learning materials and learning methods for beginners. In order to do this, we conducted a demonstration experiment using this system against 28 first-year college students, primarily programming initial scholars. In this experiment, using this system and teaching materials, we got problems to the subjects, acquired their correct answer rate, and after completing the experiment, we conducted a questionnaire about the experiment for the subjects.

As a result, both learners' learning styles determined using LSI and the degree of satisfaction and satisfaction of the presented teaching materials are positive, but they are not very popular. The cause is that the degree of satisfaction of the subjects classified as Auditory type is low, and we confirmed the degree of comprehension by actually programming against what we learned without actually writing the program, so that the lecture and practical training. It is thought that the difference in the properties of the two is affecting. However, when a beginner performs programming learning, the degree of satisfaction of the learning style, the degree of satisfaction of learning and the correct answer rate were correlated, so it can be said that the test subjects judged to be in the correct learning style have an effect.

Keywords :

1. Programming Education, 2. Programming beginner, 3. Learning Style,
4. Learning Style Inventory, 5. Teaching Material Recommendation

Faculty of Policy Management, Keio University

Tao Oshimi

目次

第1章 序論	1
1.1 背景	1
1.1.1 プログラミング学習の課題点	1
1.1.2 プログラミング学習の重要性について	2
1.2 本研究の目的	2
1.3 本研究により期待される効果	3
1.4 本論文の構成	3
第2章 先行研究・現状のプログラミング学習について	4
2.1 プログラミングの初学者の学習に関する先行研究	4
2.2 学習スタイルに着目した学習方法について	5
2.2.1 学習スタイルの定義	5
2.2.2 学習スタイルについての先行研究	6
2.2.3 Learning Style Inventory(LSI) について	9
2.3 現状におけるプログラミング学習について	10
2.3.1 プログラミング学習環境と初学者の挫折について	10
2.3.2 インターネット上におけるプログラミング学習サービスについて	11
2.3.3 具体的な既存サービスについての分析	12
第3章 提案：個人に適したプログラミング教材のレコメンド	20
3.1 課題定義	20
3.2 アプローチ	21
3.3 LSI の活用方法について	21
3.4 仮説	22
第4章 設計・実装	23
4.1 システム概要	23
4.1.1 システム設計	23
4.1.2 Learning Style Inventory のアンケートについて	24
4.1.3 分類手法について	25
4.2 実装	25
4.3 教材概要	26
4.3.1 Visual 教材について	28
4.3.2 Auditory 教材について	30

4.3.3	Kinesthetic 教材について	32
第 5 章	評価実験	35
5.1	評価概要	35
5.1.1	評価目的	35
5.1.2	評価手法	35
5.2	実証実験	35
5.2.1	実験概要	35
5.2.2	被験者について	36
5.2.3	実験後アンケートについて	36
5.3	実験の結果	37
5.3.1	定量評価: 被験者の問題の正答率	37
5.3.2	定性評価: アンケート結果	39
5.4	考察	44
第 6 章	結論	45
6.1	本研究のまとめ	45
6.2	今後の課題と展望	45
	謝辞	47
	参考文献	53

目 次

4.1	システム設計図	23
4.2	システム内部実装図	26
4.3	各教材のカリキュラム詳細	28
4.4	Visual 教材の一部	30
4.5	Auditory 教材として使用したドットインストールの画面	32
4.6	Kinesthetic 教材の一部	34
5.1	プログラミング経験区分における被験者の問題正答率	37
5.2	Visual 判別された被験者のプログラミング経験区分における問題正答率	38
5.3	Auditory 判別された被験者のプログラミング経験区分における問題正答率	38
5.4	Kinesthetic 判別された被験者のプログラミング経験区分における問題正 答率	39
5.5	判別した学習スタイルの納得度と他のスタイルのほうが学習が効率的で あるか	40
5.6	全アンケート結果	43

表 目 次

2.1	各サービスの評価表	18
5.1	実験後アンケート項目	36
5.2	各学習スタイルの判別人数	37
5.3	判別された学習スタイルと学習の納得度のスコア	40
5.4	判別された学習スタイルと学習満足度のスコア	41

第1章 序論

本章では、本研究の背景として、プログラミング学習方法とプログラミング初学者の学習に関する課題およびプログラミング学習の重要性について述べる。また、本研究の目的と本研究により期待される効果について説明し、最後に、本論文の構成について述べる。

1.1 背景

1.1.1 プログラミング学習の課題点

2020年から始まる小学校でのプログラミング教育必修化に伴い、日本国内におけるプログラミング教育に関する教育活動が活発化している。多様な進路選択に関わらず、プログラミングに関する知識や技術力が必要となる時代となり、教養としてのプログラミングの学習の必要性が高まっている。しかし、プログラミングの学習について課題がいくつか存在している。本研究ではそのうち学習途中で挫折してしまう問題の要因である「学習教材の選び方」と「学習手法」を課題とした。

- 学習教材の選び方

プログラミングの学習の必要性が高まるに伴い、習熟度に応じた教材や多様な形態の教材が世の中に出てきている。プログラミング入門時において、プログラミングの初学者が、幅広い選択肢から教材を選ぶことは困難と言える。誤った学習教材を選んでしまうと初学者が応用分野の技術に取り組んでしまうことがある。自分に適していない教材を使用した結果、内容が理解できず「自分にはプログラミングの学習は無理だ」と思い、諦めてしまうことが発生する。

- 学習手法

プログラミングの学習は他の一般教養科目と比較すると、教養科目として広く学習されるようになってから年数が浅い。そのためプログラミングの学習における学習手法が確立されておらず、自分に適した学習手法を選ぶことが困難である。自分に適さない学習手法で学習することで学習者のモチベーションの低下や学習効果が悪くなるといったことが発生する。

上記のように学習する意欲があるのにも関わらず学習教材の選び方や学習手法が誤って、プログラミングの学習を諦めてしまうことが多いといえる。しかし、これは逆に学習教材の選び方や学習手法を改善することによりプログラミングの学習を諦める人

が減少すると言える。そのため、学習手法や学習教材の選び方のサポートを改善していく必要がある。

1.1.2 プログラミング学習の重要性について

ICT の発達により IoT やビッグデータ、AI といった分野の技術が発達してきていることがあげられる。これらの技術は経済効果に大いに関係があるとされている [1]。つまり、これは現在発達している最中である技術が社会に密接に関わってくるということがわかる。そのため、今後プログラミングがどのような分野においても必要不可欠である。また、このように技術と社会の関わり方が変わっていく中、エンジニアの不足ということも問題となっている。経済産業省の IT 人材の最新動向と将来推計に関する調査 2020 年に 36.9 万人、2030 年には 78.9 万人の IT 人材が不足するとされている [2]。

エンジニア不足が今後の社会が直面する問題であることに伴い、プログラミングを学習することの必要性について重要視されている [3]。

上記のことから、今後の社会において仕事や進路の選択をするにあたりプログラミングについて理解していることや実際にプログラミングができることが必要となる。また、できないことによって様々なことに支障をきたすためプログラミングを学習することは重要であると言える。

また、この重要性についてはバラク・オバマやスティーブ・ジョブズ、マーク・ザッカーバーグといった世界的な著名人もコメントしている [4]。

バラク・オバマは生活を変えてくれるようなツールや技術を習得した若い人が必要になると語っている。

また、スティーブ・ジョブズは 1995 年のインタビューで以下のようにコメントしている。

大事なのは思考プロセスを知り考える方法を知ることさ。米国人は全員プログラミングを学ぶべきだ。

コンピューター科学は私にすれば一般教養さ。人生の中の 1 年をかけて皆が学ぶべきだと思うんだ、プログラミングの方法をね。

また同様にマーク・ザッカーバーグもプログラミングの学習の必要性について述べるコメントしている。

1.2 本研究の目的

本研究は、プログラミング入門時に挫折する学習者を減らすために、学習者の学習スタイルを分類することで個人に最適な学習手法と学習教材を導くことを目指す。初学者は自分にとって適切な学習教材を見分けることは非常に困難である。そこで学習

スタイルの分類をおこない、学習者に適している学習手法を判別する。判別した結果をもとに教材をRecommendするシステムを提案する。また、学習スタイルの分類手法として、Learning Style Inventoryを用いる。Learning Style Inventoryについては2章にて詳しく述べる。

1.3 本研究により期待される効果

本研究による期待される効果は初学者がプログラミング学習を行う上で、Learning Style Inventoryを利用したシステムを使い、学習スタイルによる適切な教材のRecommendができることである。適切な教材のRecommendができることによってプログラミング学習に至るまでの準備や、学習初期段階に達成可能性が低いと感じることや苦手意識を抱くことを和らげるのに効果があると期待できる。また、それに伴いプログラミング学習における挫折が減少することである。

1.4 本論文の構成

以下に本論文の構成を示す。第2章では先行研究と現状のプログラミング学習の課題点と現在存在しているプログラミング学習サービスについて述べる。第3章では本研究における提案事項について述べる。第4章では本研究における実験に際し作成し、使用したシステムについて述べる。第5章では評価・実験における評価手法や実験概要、実験結果について述べる。第6章では本研究のまとめを述べる。

第2章 先行研究・現状のプログラミング学習について

本章ではプログラミング学習についての先行研究と学習スタイルに着目した学習についての先行研究について述べる。

2.1 プログラミングの初学者の学習に関する先行研究

プログラミングの初学者の学習方法に関する研究は多く行われており、学習手法や学習教材、学習支援システムに着目したものがある。

学習支援システムに着目したものでは、学習用のエディターを開発することによって、学習者のサポートを行うものが多い。学習支援システムの開発に関する論文では、杉浦らや猪狩らは問題提起として初学者が学習を始めるまでの学習環境の構築でつまずいてしまう場合やデバッグの難しさについて述べている。[5, 6] これに対する解決手法として西田らはプログラミング初学者の学習環境について研究を行っている。そこではプログラミング初学者を対象とした学習環境「PEN」についての評価に取り組んでいる[7]。また、同様に兼宗らは学校教育を目的とした研究を行っている。そこでは初等中等教育を目的としたプログラミング言語「ドリトル」の開発に取り組んでいる[8]。この上記の二つの論文[8, 7]ではプログラミングのコードを書く上で必要な要素はプログラムの動作を理解することが重要であり、そこに焦点を当てられている。

また、ブロック型の命令を組み立てることによってプログラムを実装することができるヴィジュアル型プログラミング言語が存在する。このブロック型言語を用いたプログラミング教育やプログラミング教材の開発が行われている。ブロック型言語はテキスト記述型言語と異なり、文法のエラーがないためプログラミング初学者がプログラムのアルゴリズムを学習するのに使いやすい言語であるとされている。松澤らはブロック型の言語とテキスト型の言語に関する研究を行なっている。そこでは文科系大学生を対象とし、ビジュアル言語とテキスト言語に関する研究が行われており、プログラミングに苦手意識を持つ学生ほどビジュアル型のブロック型言語を選ぶ割合が高く、プログラミング言語の相互変換環境が学習促進に効果をもたらす結果を出している[9]。

プログラミングの学習手法に関する研究としては、土肥らはプログラミング学習においてつまずきやすい箇所の研究を行っている。この研究では学生を対象としてプログラミング学習においてつまずきやすい箇所の研究をしており、この研究ではプログラミングの学習者がつまずきやすい点をモデル化することで、それに応じての対策を

講じることができると述べている [10]. また、田口らは他には学習者の学習意欲や学習状況に合わせた研究をしている. この研究では、初学者を対象に初学者が学習内容の難易度による挫折がどこで発生するのか、その場合に学習者の理解できていない点に応じた演習方法などの提案がなされている [11].

学習者にフォーカスした研究として学習者のモチベーションと学習状況の関係性の分析も行われている. 土肥らはこの研究で学習者のモチベーションと学習状況の関係性を見つけており、学習者のプログラミング学習に対するモチベーションが高ければ高いほど、学習効果が良くなるという結果を述べている [12].

学習教材に関する研究としてはプログラミングの学習教材の問題の中の空欄補充問題について研究を柏原らは行っている. この研究では空欄問題はプログラミングにおいて必要となる問題を体系化することにより、問題の自動生成が可能であると述べている [13]. また、岡本らは自主学習を中心とした学習方法についての研究を行っている. ここでは初学者を対象として初学者の躓きを考慮したテキスト型の自学自習用の教材を開発し、学習効果を得ている [14]. 辻らはプログラムによる計測・制御のための Raspberry Pi 用「写経型学習教材」の開発を行なっている. この研究では初学者を対象にし、体験型の学習手法の一つとして参考資料として書いてあるコードを写す学習を想定した「写経型学習教材」の開発と評価を行なっており、評価では、学習者に対して一定のプログラミング学習の評価を得ることができている [15].

上記の先行研究から現在は以前と比べ、初学者がプログラミングを学習しやすいようにする環境ができてきていると言える.

2.2 学習スタイルに着目した学習方法について

学習方法は様々な手法が存在している. IT やインターネットの普及により、学習者ができる学習手法というものも以前に比べ飛躍的に増えていると言える. 個人が自主学習をしようとした場合、以前は参考書などの紙媒体を使った学習方法か先生や教授の講義を紙にメモをし、それを復習するといった学習方法が主流であった. しかし、現在では講義を録画しておくことにより、オンデマンドで授業を受けることができる. また、Web 上にて検索することにより、自身が望む単元のみの問題を解くことや解説をみることが可能である.

各学習者が選択できる学習方法の幅が広がったことにより、個々に適した学習方法の判別がつきにくくもなっている. そのため、学習者の学習効率を上げるにはどのような学習手法が適切であるかという、学習者の最適なスタイルを見つけ出さなければならない.

2.2.1 学習スタイルの定義

岡田らによると学習スタイルとは「学習スタイルとは学習の方法や環境についての好みのことで、学習者は学習に関して自分が好む方法や環境を持っている」と述べている [16].

また、青木らによると学習スタイルとは「学習スタイル研究は、30年ほど前から欧米、特にイギリスとアメリカで盛んに行われており、これまで、教育・心理学・経営等様々な分野の研究者が様々な学習モデルの理論・測定方法を打ち出している。一方、画一的な教育が行われてきた日本では、個人差に目を向ける学習スタイルの研究は、欧米に比べるとその歴史は浅い」と述べている [17]。

本研究における学習スタイルとは「個人の経験や習慣に基づき、形成されてきたものを鑑みて、学習者にとって効率的に学習するための最適な手法」である。これは本研究が学習者にとっての最適な学習効果をもたらす学習スタイルの分類と学習手法の提案であることに基づいている。

2.2.2 学習スタイルについての先行研究

学習スタイルについての先行研究は大きく分けて2つのパターンに分けることができる。学習スタイル自体の研究と学習スタイルを活用した学習方法についての研究である。

まず学習スタイル自体の先行研究について述べる。学習スタイルは日本よりもイギリスやアメリカで30年ほど前より研究が盛んに行われてきた。一方、日本では近年になって研究が行われるようになってきているが、歴史はまだ浅い。イギリスやアメリカでは数多くの研究がされており、その中でも71個の異なる学習スタイル理論・モデルが提唱されている。しかし、多くある学習スタイルについて全て研究を進めると、学習スタイルの研究が滞ってしまう。そのため、イギリスの学習スキルセンターではCurryの学習スタイルのオニオンモデルを基として考え、学習スタイルの研究で大きく影響を与えた13個の理論・モデルに絞って研究を進められている [18]。青木らは各種学習スタイルについて調査結果を報告している [17]。そのため、本節ではその13個の学習スタイルについて整理する。

- Gregorc の学習スタイル Gregorc の学習スタイルでは学習スタイルを「人が環境をどのように学習し、それに適応するかを示す独特の行動から成るもの」と定義されている。この学習スタイルでは「Perception」「Ordering」の二つの軸から成立している。「Perception」では具体性と抽象性が両端になっており、「Ordering」では連続性と任意性が両端としている。また、ここから具体的連続性と具体的任意性、抽象的連続性と抽象的任意性の4つの学習スタイルに分類されるとしており、人はそれぞれ上記の一つか二つの学習スタイルに当てはまり、それは生来のものである述べている。 [19, 17]
- Dunn & Dunn の学習スタイル

Dunn & Dunn の学習スタイルでは「環境的」「感情的」「社会的」「心理的」「身体的要素」の5つにて分類されている。心理的刺激以外の4つの刺激が、学習者の学習スタイルを決定する要素であると述べており、また、個人の学習スタイルにあった学習環境で学習することによって学習効果が向上すると述べられている。

また、Dunn の学習スタイルモデルは、アメリカの初等教育に影響がかなりあり、イギリスにも徐々に広まっている。[20]

- Riding & Rayner の認知スタイルと学習スタイル

Riding & Rayner の認知スタイルは、考え方、及び、情報を整理して表現するやり方、と定義づけられている。学習スタイルは学習する際に個人が学習していく際の手法の選択する過程の一部であると定義づけられている。学習スタイルと認知スタイルの差は認知スタイルは生来のものであり、学習スタイルは時と場合によって変わることであると述べられている。また、Riding & Rayner は約 30 個の学習スタイルを評価した上で Cognitive Styles Analysis(CSA) の開発が行われた。CSA はテスト形式であるため、パソコンでテストができるため、イギリスでは学習スタイルの研究によく使われている。[21]

- Myers-Briggs Type Indicator

Myers-Briggs Type Indicator はユングの心理的タイプ理論を基としたものであり、性格を診断する際に心理学者によく使われている心理テストである。また、Myers-Briggs Type Indicator は心理テストであるが、日常的行動を測定する質問も含まれているため学習スタイルへの関連性があるとされている。[18]

- Apter の動機スタイルの反転理論

Apter の動機スタイルの反転理論は心理的な理論である。Apter の動機スタイルの反転理論では精神生活は、目的手段、規則、取引、関連性の 4 つの領域が変化することから考察されるとしている。その上で、「それぞれの領域内で「反転」が起こったとき、人は精神的満足感を得られる」と述べられている。この反転理論に関しては学習に関する動機、感情といった態度に利用することが可能であるとされている。[22]

- Jackson の学習スタイルとプロファイリング Learning Styles Profiler (LSP) は神経心理学的学習スタイルであるとされている。LSP では学習スタイルは個人の個性の一部であるとされており、個人の活動の上で形成されていくとされている。また、LSP では、「Initiator」、「Reasoner」、「Analyst」、「Implementer」の 4 つの学習スタイルに分類している。この分類からわかるように、LSP はビジネスと教育のためのものであるとされている。また、Jackson は一人一つの学習スタイルにする必要はないと述べている。[23]

- Kolb の学習スタイル

学習スタイルの研究において、最もよく使われているのが Kolb の学習スタイルモデルである。Kolb の学習スタイルでは、個人の学習スタイルは生まれ持った性格とは異なり、学習内容や理解状況、個人の状態によって変化するものであるが、そう簡単に変化するものでもないと述べている。Kolb が開発した学習ス

イルの判別手法である Learning Style Inventory (LSI) から判別された学習スタイルは変化がなかなかしないとされている。詳しくは次節にて後述する [24]。

- Honey & Mumford の学習スタイル

Kolb のモデルを基に派生した「管理職の人がどうやって学ぶか」という方向に発展させたものである。そして、その中で Kolb が開発した学習スタイルの判別手法に似た Learning Styles Questionnaire (LSQ) という質問集を製作された。この判別手法より、個人の学習スタイルは活動型、熟考型、理論型、実践型の4つの学習スタイルに分類されるとしている。また、これは個人の優劣をつけるものではなく、すべてのスタイルを使いこなせる方が良く、また、各学習スタイルは状況に応じて他の学習スタイルのほうがよいといったことがあり得ると述べられている。また、Mumford は、LSQ は学習スタイルを判別するだけではなく、学習効率を改善するためのものでもあると述べている。[25, 26]

- Herrmann の whole brain モデル

Herrmann は Hermann Brain Dominance Instrument (HBDI) と脳波の測定による右脳と左脳の活動状況との相関関係を多少なりとも実証している。脳の働きを考慮し、人の脳の働き方を「Theorists」「Organisers」「Innovators」「Humanitarians」の4つに分類をした。この4つの分類の一つにしか該当しないというわけではなく、いくつもの分類の働きを脳がしている場合も存在している。Herrmann は、状況に応じて脳の働き方をを使い分けられるようになることが学習の目的であり、それによって創造力を伸ばすことが出来ると述べている。[27]

- Allison & Hayes の認知スタイルパターン

Allison と Hayes は LSQ を用いた研究をすることで、行動と分析の二つの要素から Cognitive Style Index (CSI) を開発した。CSI は組織における自己開発をする上の研究ツールとして作成された。CSI は認知学的な見地より作成されており、Allison と Hayes は「認知学的な見地の有用性は個人の認知スタイルを職場の要求によって変えることができるところにあるのではなく、個人の認知スタイルに適合した職種を選ぶ、あるいは、職場環境を作り出すことができるところにある」と述べている。[28, 17]

- Entwistle の学生のための学習サポートに関してのアプローチの手法

Entwistle は大学などの教育機関内で学生の学習の質が良くなる学習環境を提供するため、教育の指導法の創出することが研究目的である。Entwistle は心理学的要素が学習方法に密接に影響していると述べている。心理学的なアプローチとしては学生の学習段階が高くなると深層学習を行い、逆に学習の積み重ねのみを主体としている場合、表層学習を行う傾向にあるされている。そのため学生の学習段階に合わせて、深層学習と表層学習を行うことができるように学生の学習をサポートする必要があると述べられている。この学生へのサポートのアプローチは場合によって変わるとされている。また、学生の性格によってサポートの仕方

が変わることから学生の性格を「無責任型」「活動型」「突入型」の3つに分類している。Approaches and Study Skills Inventory for Student ではこの性格の3分類からのアプローチの仕方のパターンが述べられている。[29]

- Vermunt の学習スタイル分類についてと学習スタイルのパターン (ILS)
Vermunt は学習スタイルと学習アプローチは同じものだとしており、そして様々な学習状況において一貫性をもって学習することに対して着目している。また、Vermunt は学習スタイルは性格などの変化のないものに比べて、状況によって変化するものであるとしている。学習スタイルは「meaning-directed」「application-directed」「reproductiondirected」「undirected」の4つの学習スタイルを提案しており、またそれぞれの学習スタイルの差異として「知覚」「動機」「感情」「精神」「規制」の5つの要素にて論じている。しかし、ILS では「感情」については排除されている。[30]
- Sternberg の思考スタイル理論とそのパターン
Sternberg は、思考スタイルと教育、思考スタイルとその評価方法について研究を行っていた。Sternberg は、スタイルは「どのようにしたいか」であり能力は「どれだけ出来るか」とであると定義している。そのため、学習スタイルを「人がどのようにして学びたいか」であり、思考スタイルは「人がどのように学習内容を考えるか」と述べている。Sternberg の学習理論は自己管理理論より学習方法自己管理方法の延長線上であるとしている。そのため、管理方法として述べられている政治形態4種類と政府機能の3種類と二つのレベル、内外の2種類、2種類の思考的傾向を合わせることで13種類の思考スタイルに分類している。[31, 32, 17]

2.2.3 Learning Style Inventory(LSI) について

Learning Style Inventory(以下 LSI) は Kolb の経験学習理論 [33] から派生したものである。Kolb は学習者が自分の学習スタイルを理解すること、また、自分の学習スタイルと他人の学習スタイルとの差を理解することによって円滑にコミュニケーションをとることができる」と述べている。その上で Kolb は個人に内包する学習スタイルとして「適応型」「収束型」「発散型」「同化型」の4つに分類をしている。この4つの分類は具体性と抽象性の軸と能動性と考察性の軸の2軸で形成することができるとしている。Kolb は学習について「経験を変換して知識が形成されるまでの過程である」と定義している。Kolb はこの経験学習は6つの特徴を有していると述べている。

1. 学習はプロセスであり、結果ではない
2. 学習は経験による断続的なプロセスである
3. 学習は社会に適応する中で、4つの分類の学習スタイルの軸となっている2軸の要素をを融合していくことによって生まれるものである。

4. 学習は社会に適応するための全体性的なプロセスである。
5. 学習は個人と学習環境との兼ね合いがある。
6. 学習は知識を生成する上でのプロセスである。

Kolb は学習の過程を学習スタイルを形成する 2 軸から表すことができるとしている。具体性と抽象性の軸では、学習内容の具体的な部分から学ぶか概念的な部分から学ぶかを表している。能動性と考察性の軸では学習内容を実際に体験をすることで学ぶか、考えることによって学ぶかを表している。

ここから Kolb は学習方法として学習内容を視覚的に学ぶ「Visual スタイル」と学習内容を聞くことによって学ぶ「Auditory スタイル」と学習内容を体験することによって学ぶ「Kinesthetic スタイル」の 3 つのパターンがあるとし、LSI のアンケートを作成し、アンケートに回答することによって回答者の学習スタイルを 3 つのスタイルに分類することを可能にした。

具体的な各学習スタイルについては下記のものである。

- Visual スタイル

Visual スタイルの学習者は見ることに本を読んだり、図形、実演を見ることによって学習するスタイルが得意である学習者である。

- Auditory スタイル

Auditory スタイルの学習者は音声を聴くといった聴くことによって学習するスタイルが得意である学習者である。

- Kinesthetic スタイル

Kinesthetic スタイルの学習者は問題を解くことや実際に体験をすることによって能動的に学習するスタイルが得意な学習者である。

2.3 現状におけるプログラミング学習について

本節では現状のプログラミング学習の環境や手法、学習サービスについて述べる。

2.3.1 プログラミング学習環境と初学者の挫折について

現在プログラミングの学習は様々な学習環境で行うことができる。学校環境における情報の授業や書籍による学習、インターネット上のプログラミング学習サービスなどが代表的である。これらの学習環境は主に高校生や大学生以上の年齢を対象としているが、中には小学生を対象としたプログラミング学習塾などもある。しかし、本研究は学習者の対象を初学者の高校生以上としているため、小中学生を対象としたプログラミング学習塾などは類似サービスの対象から除くものとする。

学校環境における情報の授業以外の書籍による学習やインターネット上におけるプログラミング学習サービスは初学者以外が対象である場合も存在している。中でも書籍による学習は対象者が初学者でない場合が多い。そのため、前章で述べた初学者がプログラミング学習を挫折する要因の一つとなる教材の選択を誤ることが多くなってしまう。

もともとプログラミングは専門的な学問であり、他の一般教養科目と比較すると学習している人は少なかったが、2017年現在は一般的に学習されるようになり、学習者が増加してきている。初学者は教材が初学者向けであるか否かの判別というのは非常につきにくい。しかし、世の中にはプログラミング学習用の教材が多々あり、の中には中上級者向けである書籍なども数多く存在する。書籍のレビューなどで良い評価がついていたとしても、それは初学者にとっていいかどうかは不明であり、ここで初学者にとって誤った教材を選ぶと初学者は書籍の内容が全くわからないということになってしまう。最近では初心者向けの書籍というのも増えてはいるが、全くコンピューターを知らない人を対象としているのか、それともコンピューターを多少は使ったことがある人が対象なのか、はたまた書籍に書いてあるプログラミング言語が初めてであり他の言語を書いたことがある人が対象なのかといったことは判別がしづらいという懸念は存在している。

また、書籍やインターネット上におけるプログラミング学習サービスは開発環境の構築などについて詳細に書いてあることは少ない。プログラミング学習における初学者が挫折してしまふこととしては開発環境の構築で挫折してしまうこともある。中にはプログラミングを勉強しようとしたが、プログラミングのコードを勉強する前に挫折してしまったというケースも存在している。[34]

2.3.2 インターネット上におけるプログラミング学習サービスについて

プログラミングの学習者の増加に伴い、様々なプログラミング学習サービスが以前に比べ、増加した。また、プログラミングを学習するという性質上コンピューターを使うことが前提となっており、インターネット上で安価に学習することが可能になってきている。多種多様なプログラミング学習サービスがあり、それぞれのサービスによりやり方や学習するプログラミングの言語の対象、学習者のレベルの対象に違いが存在する。

対象となる学習者のレベルは主に初心者から中級者を対象としているサービスが多くなっている。これは中級者や上級者は参考となる”GitHub”や書籍、対象言語やフレームワークのリファレンスを自分で理解し、実行することができるケースが多い為、1から10まで全てを教わる形は学習効率が落ちるからではないかと考えられる。しかし、初学者の場合に関しては前節で述べたように、環境構築やどのように学習したらよいかかわからないというハードルが存在する。そのため、1から10までを教えてくれるサービスが必要となり、その結果プログラミング学習サービスは1から10までを教えてくれる教材が多くなっている。

プログラミング学習サービスで学習するにあたり、どのようなプログラミング言語が学習できるかという点に着目すると、多く見られるのはHTMLやJavaScript, CSS, PHPといったホームページを作成するにあたり必要な言語が多くなっている。これは初学者の動機が”自分のホームページを作りたい”といったことや、なにかビジネスやサービスを作るにあたって”ホームページが必要になった”ということが動機になることが多いからではないかと推測される。最近ではスマホの普及により、スマホのアプリが活用される頻度が高い。サービスを活用するユーザーの需要がスマホのアプリを作成できる言語であるSwiftやAndroid Javaといった言語にも増えてきている。それに伴い、プログラミング学習サービスでもそれらを学習することができるようになってきている。プログラミング学習サービスで学習できる言語は非常に環境構築が楽なものが多く、Script言語が多い。これは初学者がプログラミングの学習を始めるにあたるハードルを下げるためではないかと考えられる。

また、プログラミング言語を学ぶのではなく、プログラムを作る上で必要になってくるアルゴリズムについて学ぶサービスも多い。これらのサービスでは、ゲーミフィケーションの手法を取られることが多く、ゲーム感覚でプログラミングの基礎を学べるようになっている場合が多い。

また、中にはサービスのWebページ上にてプログラムを組み、実行するようなサービスも多く存在した。このような手法を利用しているサービスは单元ごとにプログラムを作成し、実行し、Webページ上にてプログラムの結果が表示され、できていたら次へ、できていなかった場合はエラーが出てきてそれを直すという形で学習していくものが多かった。他には動画を活用し、各单元ごとの動画を用意し、それを学習者が自発的に見て、実際に実行する場合には自分のコンピュータで実行するといった手法を取っている場合も存在した。

2.3.3 具体的な既存サービスについての分析

今後のプログラミング学習手法を論じるためには現状のプログラミング学習環境について分析を行う必要がある。そのため、前節で述べたインターネット上におけるプログラミング学習サービスについての分析をおこなった。

分析する既存サービスの一覧

分析をおこなったプログラミング学習サービスは以下の25個のサービスである。

- コードモンキー (<https://codemonkey.jp/>)[35]
- paiza ラーニング (<https://paiza.jp/works>)[36]
- Schoo (<https://schoo.jp/guest>)[37]
- Progate (<https://prog-8.com/mypage>)[38]

- ドットインストール (<https://dotinstall.com/>)[39]
- CODEPREP (<https://codeprep.jp/>)[40]
- codecademy (<https://www.codecademy.com/>)[41]
- Udacity (<https://www.udacity.com/>)[42]
- Progra! (<https://progra.org/>)[43]
- シラバス (<https://cyllabus.jp/>)[44]
- CodeStudy (<https://jeek.jp/school/programming>)[45]
- コーディング道場 (<https://coderdojo.jp/>)[46]
- ミニツク (<http://www.minituku.net/>)[47]
- ShareWis (<https://share-wis.com/>)[48]
- 動学 (<http://dougaku.tv/html/faq/title004.html>)[49]
- Udemy (<https://www.udemy.com/jp/>)[50]
- Skill hub (<http://skillhub.jp/>)[51]
- CodinGame (<https://www.codingame.com/>)[52]
- Flappy Code (<https://studio.code.org/flappy/1>)[53]
- FIGHT CODE (<http://beta.fightcodegame.com/>)[54]
- MOONBlock (<http://www.moonblock.jp/>)[55]
- ウェブカツ!! (<https://webukatu.com/>)[56]
- Treehouse (<https://teamtreehouse.com/>)[57]

上記のプログラミング学習サービスに決定した理由としては、Web 上で初心者が学習する際には Web 検索を活用すると推測される。その場合、”プログラミング 初心者”といったキーワード検索がされると考えられる。この検索結果では実際にプログラミング学習サービスのホームページがでてくるともあるが、それ以外にプログラミング学習サービスのレビューサイトがでてくる。このレビューサイトにてでてくるものが利用されやすいことは明らかであるため、参考にした。[58][59][60]

各サービス概要

コードモンキー アルゴリズムの勉強・プログラミングの組み立て方の勉強・感覚的プログラミング勉強を目的としており、ゲーム感覚で学ぶことができる。

paiza ラーニング 3分程度の動画にて学習したのちに、演習として、ブラウザ上にてコーディングを行う。また、演習結果に関しては採点を行うシステムが存在する。

Schoo 講義の録画を配信している動画形式の学習サービスである。稀に生配信をしている場合も存在している。多種多様なプログラミング言語を学習することが可能である。

Progate 講義パートと演習パートにわかれており、演習パートはブラウザ上にてコーディングを行うことができる。一つの演習パートをクリアすると次の講義パートに進むことができる。ユーザーにレベルが存在しており、学習の進み方によりレベルが上がるというゲーミフィケーション的な要素が入っている。

ドットインストール 3分間の動画で学んでいく動画を公開しており、順番に聞くことによってプログラムを学習していくことができる。また、開発環境から動画で学習することができる。

CODEPREP 穴埋め形式の問題を解いていく形のサービスである。基本的に講義といったものは存在せず、知識はその問題ごとに必要なものを提示しており、問題を解くことにより、学習を進めることができる。

codeacademy 基本的なところから説明がされており、課題がその都度出てくるサービスである。エディタがブラウザ上に存在するため、環境構築は必要がない。また、ブラウザ上にてコーディングする場合、エディタの横に学ぶことについて解説が掲載されている。

Udacity 未経験者から上級者までを対象とする、動画形式の学習サービスである。1つの動画は短くまとまっている形である。

Progra! 穴埋めの問題を解いていく形の学習サービスである。基本的にサイト内で全て完結するため、学習環境の構築は必要がない。学習進度をヴィジュアル的に一目で見る画面も存在している。

シラバス 実際の Web サービスなどの開発の流れを沿うことによって学習を促すサービスである。非常に実務的であるが、学習環境の構築などについても画像による説明がされている。

CodeStudy ブラウザ上にてコーディングを行う為、環境構築をせずに、実践的に JavaScript を学ぶことができる。問題が提示され、それに対してコーディングを行なっていく、問題を解くとレベルが上がっていくというゲーミフィケーション的な要素も入っている。問題は小問題集のような形である。

コーディング道場 全 97 問の問題が存在しており、それぞれに難易度を設定していて、個人個人でそれを解いていく形のサービスである。プログラムの説明自体は少なく、問題を解かせることを重要視している。また、解く問題の挑戦者数などが提示されており、ゲーム感覚で学習に臨むことができる。

ミニツク Ruby を学ぶためのサービスである。環境国地区など、初心者にはつまづくところも多くあると思われるサービスである。Ruby に特化しているため、Ruby 技術者認定試験の対策問題集なども無料で利用が可能である。

ShareWis 各コースの単元で簡単な説明後、演習を行うという形のサービスである。また、演習問題では 3 択などの簡単な問題が多く、出題しており、コードを書かせるような演習問題は非常に少ない。

動学 動画を視聴することによって学習する形のプログラミング学習サービスである。演習などは一切存在しない。また、教材数は 500 超存在する。

Udemy 学習するコースを購入後、それを学ぶ形の学習サービスである。オンラインにて講師に質問することができる。コース概要のみ動画で説明がされている。

Skill hub Web サービス製作系の授業を動画で受けることでプログラミング学習サービスである。初心者の人から受けられるものであり、授業の動画をそのまま実践することでサービスを作成することができるものである。

CodinGame ブラウザ上にてゲームをプログラムして作成していく形のプログラミング学習サービスである。様々なプログラミング言語に対応しており、どの言語であってもミッションを成功させればクリアとなる。ミッションはゲームを主体としており、ゲームをする感覚でプログラミングを学習することができる。

Flappy Code 人気ゲーム「Flappy Bird」でビジュアルプログラミングの学習ができるサービスである。ヴィジュアルプログラミングであるため、アルゴリズムの学習に適している。

FIGHT CODE 「Fight Code」は、JavaScript でロボットの動作を記述して、他のユーザのロボットと対戦することができる。JavaScript をある程度学習していないとまらないが、ゲーム感覚で JavaScript の学習を進めることができる。

MOONBlock ブラウザ上にてヴィジュアルプログラミングができるサービスである。ブロックを組み合わせることによってプログラムを作っていくため、アルゴリズムの学習に適している。

ウェブカツ!! Web サービスの製作に必要な知識を動画にて学習するプログラミング学習サービスである。動画の中にはつまづきやすいポイントやエディタの使い方などがあることから初心者になりに向いている学習サービスである。

Treehouse 動画を見て学びつつ、演習を動画の中に挟んで、コードを書かせるように促すプログラミング学習サービスである。ブラウザ上にてコーディングが可能であるため、環境構築などは不要である。

分析軸について

前述したプログラミング学習サービスの分析軸は以下のものでおこなった。

- 対象者のレベル
- 学習できるプログラミング言語
- 学習スタイル

分析軸の評価手法

この節では分析軸の評価手法について述べる。

対象者レベルの評価 対象者のレベルは初心者・中級者・上級者(1 3)の3つに分類をする。レベルの基準としては初心者(1)のレベルはプログラミングを初めて学ぶ、もしくは少し触ったことがあるが学習するプログラミング言語の公式のリファレンスのみでは学習できないものとする。中級者(2)のレベルは一つのプログラミング言語を学んだことがあり、リファレンスが多少は理解できるものとする。上級者(3)のレベルはリファレンスのみで学習するプログラミング言語のプログラムを作ることができるものとする。

学習できるプログラミング言語の評価 分析するサービスにて学ぶことができるプログラミング言語がいくつあるか、何の言語であるかということを確認する。これは全体の分析をする際にどの言語を学習する環境が多かったかどうかを確認するために使用する。また、アルゴリズムについてゲーミフィケーションで学ぶなどといった場合にはその他に分類する。

学習スタイル 学習スタイルはサービス内の学習する手段である。これは後述する Learning Style Inventory(LSI)にて使われている3分類にて分析をする。3分類は Visual(V), Auditory(A), Kinesthetic(K)の3つに分類する。サービスによってはこの3分類の中で複数の手法を取っているものもあるため、各サービスを3分類のスコアを5段階にて評価する。評価軸としては、Visual, Auditory, Kinestheticの要素をどの程

度使っているかで評価している。その要素単独で教材が成り立つ場合は5であり、補助要素として使われている場合は2という形でどの程度教材にとって重要視されているかというスコア隣っている。

分析結果

前述した評価軸について、分析したものを表にまとめた。

サービス名	難易度	学習可能なプログラミング言語	V	A	K
コードモンキー	1	その他			5
paiza ラーニング	1-2	C, Java, Python, PHP, Ruby, HTML, CSS, JavaScript, SQL		5	3
Schoo	1-2	Ruby on Rails / Ruby / PHP / SQL / PHP / MySQL / JavaScript / HTML / CSS / JAVA / Python / AWS / C		5	
Progate	1	HTML / CSS / JavaScript / jQuery / Ruby Ruby on Rails / PHP / Java / Python / Swift	2		5
ドットインストール	1-2	HTML, CSS, JS, PHP		5	
CODEPREP	1-2	HTML / CSS / Ruby / PHP / jQuery / Bootstrap / Java	3		5
codeacademy	1-3	HTML / CSS / JavaScript / jQuery / Ruby Ruby on Rails / PHP / Python / Java / SQL	4		5
Udacity	1-3	Java / JavaScript / HTML / CSS / SQL / Swift Unity / Python / Ruby	4	4	
Progra!	1	PHP / Ruby / Python	3		5
シラバス	1	HTML / CSS / Ruby on Rails	4		
CodeStudy	1	HTML / CSS / JavaScript / jQuery	2		5
コーディング道場	1	PHP / Ruby	1		5
ミニツク	1-2	Ruby / Ruby on Rails	4		
ShareWis	1-2	C / CSS / HTML / JavaScript / jQuery / MySQL / PHP / Unity	3		3
動学	1-3	C / CSS / Dreamweaver / HTML / Java / Linux		5	
Udemy	1-2	CSS / HTML / Swift / Java / Javascript / jQuery / PHP / Python / Ruby / Python / C	4		1
Skill hub	1	HTML / CSS / JavaScript / jQuery / Ruby		4	2
CodinGame	2-3	JavaScript, Go, C++, ObjectiveC, Haskell, PHP, Perl, Python, Scala, Ruby			5
Flappy Code	1	その他			5
FIGHT CODE	2	JavaScript			4
MOONBlock	1	その他			5
ウェブカツ!!	1-2	JavaScript / HTML / CSS / PHP / MySQL / jQuery		5	
Treehouse	1-2	HTML / CSS / Javascript / Ruby / PHP / Android Java		4	4

表 2.1: 各サービスの評価表

現状におけるプログラミング学習についてのまとめ

まず学習内容から大きく分けて二つに分けることができる。ヴィジュアルプログラミングによるプログラミング学習と通常のプログラミング言語の学習に分けることができる。

ヴィジュアルプログラミングに関しては完全にプログラミングをしたことがない初学者をターゲットにしており、プログラミングのアルゴリズムを学ぶことを重要視している。ヴィジュアルプログラミングの現状のサービスではゲームの要素が必ず入っていた。

通常のプログラミング言語の学習サービスでは、Web サービスを製作する上で必要な言語が学習教材として多く使われている。中でも HTML や CSS, JavaScript は学習内容に用いられている割合が多く、調査サービスのうちの約 73%にも及ぶ。

また、C 言語や C#, JAVA といったコンパイル言語を教材として扱っているものは少なく、初学者のみをターゲットとしたサービスには学習内容としてはいっていなかった。

次に LSI にて分類される Visual, Auditory, Kinesthetic について述べる。まず、要素が多かったものとして、Kinesthetic タイプの要素が全 23 件のうち、16 件と、約 70%と多くなっていた。Kinesthetic タイプの要素のスコアの平均は約 4.2 と高い数値であった。しかし、Kinesthetic タイプの要素のみのものは 5 件であり、そのうち 3 件はブロック型のプログラミングによるアルゴリズムの学習を目的としたものであり、コードを書くプログラミング教材では 2 件であり、その 2 件は中級者向けとなっているため、初学者がプログラムのコードを書くための学習をする上では Kinesthetic タイプの要素のみのものは 0 件であった。

次に Visual タイプについても単独のものは少なく、2 件という結果が出ている。しかし、Visual タイプの要素についてはスコアの平均が 3.1 と Kinesthetic タイプの要素に比べて低く出ており、ここから Visual タイプの要素は Kinesthetic タイプの要素と比較して初学者のプログラミング学習においては補助的な使い方をされていることがわかった。

単独の要素で成り立っているものの多くは、Auditory タイプの要素を含んだものであった。単独で成り立っている学習教材は 3 件であるが、Auditory タイプの要素を含んだ学習教材が 6 件しかないことから半分が単独で成り立っている。また、Auditory タイプの要素の平均スコアは約 4.6 と 3 つの要素の平均スコアの中で最大のものであった。このことから、学習教材として使う場合は Auditory タイプの要素はメインの要素として使われることが多いとわかる。また、Auditory タイプの要素が入っている学習教材は全て、初学者向けの学習教材として利用が可能であった。

第3章 提案：個人に適したプログラミング教材のレコメンド

本章ではプログラミングの初学者が挫折せずプログラミングを学習する目的に対する本研究における提案を論じる。

3.1 課題定義

プログラミングの初学者がプログラミングを学習するにあたり、幾つかの壁が存在する。壁を以下のものと本研究では定義する。

1. 学習言語の選定
2. プログラムの実装環境構築
3. プログラミング学習における自身の学習スタイルの見極め
4. 教材の選定
5. フォロー環境の有無

1の学習言語の選定は難しい。本人が作りたいと考えているものが難解な言語を用いなければならない場合があり、その際にやりたいことをやるためと言って難解な言語を選択してしまうと、初学者は「プログラミングは難しい」と感じて挫折してしまう場合がある。

2のプログラムの実装環境構築はコンピューターを普段から使っている人であってもターミナルやコマンドプロンプトを使ったことがある人は少なく、コンピューターを普段からあまり使わない人にとっては非常に高い壁になり、プログラミングを実際に行う前にやめてしまう人もいる。

3のプログラミング学習における自身の学習スタイルの見極めは他の一般科目とプログラミングの学習は違う点が多いため難しい。まず、一般科目であっても自身に最適な学習スタイルを完全に確立している人は少ない。その上、プログラミングの学習は他の一般教養科目（この場合、数学や英語といった科目）と違いがある。これはコンピューター上でしか実際にプログラミングを実行できない点である。最近ではスマートデバイス上でプログラミングが可能になってきてはいるが、プログラマーの大部分がコンピューターに類するものでプログラムを書いている。これにより、プロ

プログラミングの学習は一般教養科目とは違った形の学習手法をとることができる。そのため、プログラミング学習における個人に最適な学習方法というのは一般教養科目と同じであるかは不明である。また、プログラムを組む上ではプログラムの文法を覚えるだけではなく、プログラミング特有のアルゴリズムを理解し、使えなければならず、上記と同様に一般教養科目と同じ学習手法がいいのかというのは不明である。

4の教材の選定については現在プログラミングに関するサービスや書籍が数多くあり、どれが初学者にとっていいものであるのかは判別するのは難しい。これは初学者以外の学習者にも言えることであり、自分の持っている知識が書いてあるだけの書籍からは学習することは少ない。しかし、書籍が自分の学習に役立つものであるかは読まない限り不明である。そして、これはなんの知識もない初学者であれば特に顕著に出てしまう。

5のフォロー環境の有無についてだが、これは学習者がわからないところなどがあった際にその問題点の説明や解説をしてくれる人がいるかいないかでは学習者の学習効率が大きく変わることは明らかである。

1の学習言語の選定や4の教材の選定についてはインターネット上の情報や学習者の友人などから解決ができる課題である。2のプログラムの実装環境構築については書籍やインターネット上の情報を検索し、取得することで解決をはかることができる。また、5についてはこの問題を解決するようなサービスが存在している。そのため本研究の課題は3のプログラミング学習における自身の学習スタイルの見極めを課題とした。

3.2 アプローチ

本提案のアプローチとしては個人の学習方法を分類し、学習者にとって最適な学習方法をリコメンドするシステムがあればいいのではないかと考えた。個人の学習方法の分類をするために、LSIという個人の学習方法の分類するアンケートに着目した。LSIは元が英語のアンケートであるが、これを日本語に訳して、学習者に対して実施する。そのアンケートをLSIの分類手法を用いて、学習者に適した学習方法を判別する。学習者に対して、判別した学習方法とともに学習教材を提案する。

3.3 LSIの活用方法について

LSIのアンケートをブラウザ上にて回答し、回答結果から学習スタイルのスコアを算出する。算出した結果を元にアンケート回答者の学習スタイルを決定する。スコアの算出方法については5.1.3にて後述する。学習スタイルの決定後、学習スタイルごとに用意してあるプログラミング学習教材をブラウザ上に表示する。

3.4 仮説

本研究の仮説は以下のものである。

Learning Style Inventory を用いて，学習者に対してアンケートを実施する．その結果をもとに学習者に対して Visual, Auditory, Kinesthetic の3つの学習手法の中から最適な学習手法を判別し，その学習手法に沿った学習教材を提案する．上記の手法をとることによって初学者の学習効率が向上するのではないかと考えられる．

第4章 設計・実装

4.1 システム概要

本研究の仮説に基づき，Learning Style Inventory の日本語訳アンケートを Web 上にて行う．アンケート結果をもとに Visual，Auditory，Kinesthetic の3つの学習スタイルにアンケート回答者を分類する．分類結果をアンケート回答者に対して表示し，アンケート回答者の判別された学習スタイルの教材へ誘導する．

4.1.1 システム設計

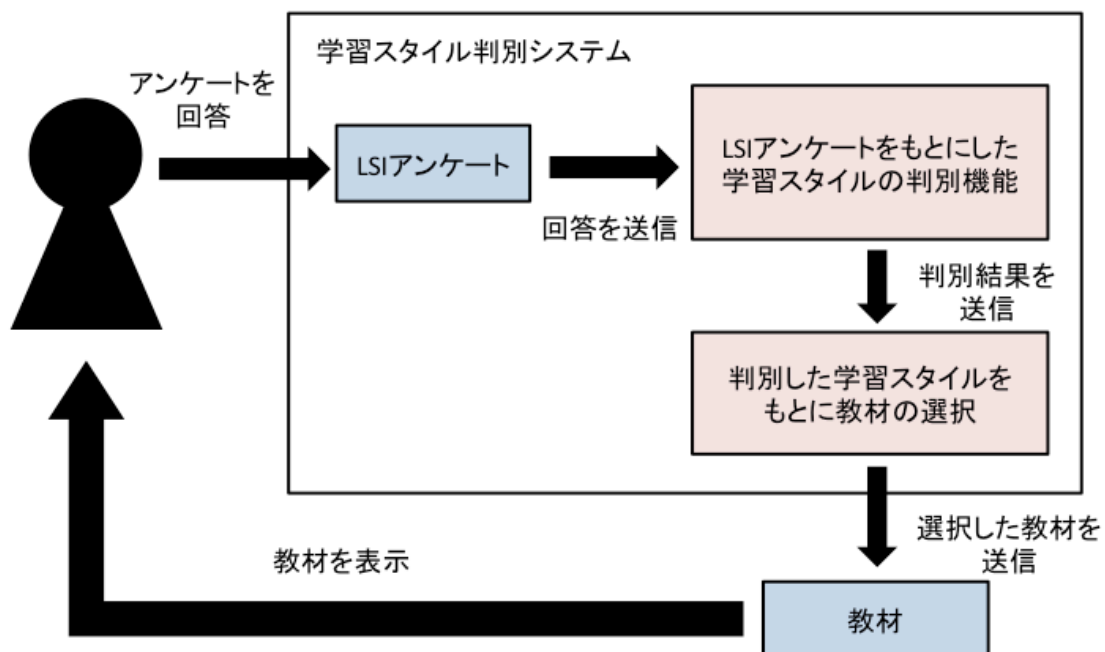


図 4.1: システム設計図

4.1.2 Learning Style Inventory のアンケートについて

アンケート項目は以下の 30 項目である。

1. 本を読むよりビデオ（テレビ）を観る方が好き
2. ラジオや CD を聴きながら歌っている時、その歌詞を理解できる
3. 運動神経が良い
4. 本を読みながら、物語の場면을思い浮かべることができる
5. 音楽を聴きながら勉強するのが好む
6. 実際に体験しながら学習するのが楽しい
7. スポーツは、観るよりもするほうが好きだ
8. 声に出して読むことで覚えやすくなる
9. 何かをする前に、まず他の人がするのを見てから実行する
10. カラーコーディネートを考えながら洋服を選ぶ
11. ライミング（押韻）やラップが得意だ
12. 感覚的に感じた時に「できている」や「厳しい」と言うことがある
13. 物事を理解するのに何度も見返す必要がある
14. 教科書よりも講師から口頭で教わる方が良い
15. 長時間じっとしていることが苦手だ
16. なにかを見た時に、「あなたの言っていることはわかる」、「かつこいい」「疑問が解けた」などと言う
17. 物事がどのようになっているのか解決するのが得意だ
18. 映像授業でも理解をすることができる
19. 頭の中で映画のシーンを思い返すことが容易である
20. 外国語学習が楽しい
21. 誰かがするのを見るよりも、自分で実験をして確かめる方がよい
22. 写真を撮るよりも家などを塗る方がよい
23. グループで勉強するのが楽しい

24. 人の家に行くときは、手書きの地図を好む
25. ものを見て、目を閉じた時にそのもののイメージを覚えている
26. 音楽のセンスがある
27. 新しい単語を学ぶ時は何度も書いて覚える
28. 何かをする際は、それをしている姿を想像してからとりかかる
29. 話を聞いた時に「思い出した」、「わかった」、「いい感じ」といったことを言う
30. 道具を使って何かを作るのが楽しい

以上のアンケート項目に対して Yes, No で回答してもらう。アンケートの質問は Visual, Auditory, Kinesthetic の 3 つの学習スタイルに沿った質問になっている。各質問の学習スタイルは以下のようにになっている。

- Visual : Questions 1, 4, 9, 10, 13, 16, 19, 24, 25, 28
- Auditory : Questions 2, 5, 8, 11, 14, 18, 20, 23, 26, 29
- Kinesthetic : Questions 3, 6, 7, 12, 15, 17, 21, 22, 27, 30

また、本アンケートは Learning Styles Inventory のアンケート項目 [61] が英語であったため、日本語訳したものである。翻訳するにあたり、英語特有の言い回しが 12, 16, 29 項目に存在していたため、イディオムと質問に対応する学習スタイルより意識をしている。

4.1.3 分類手法について

アンケートの Yes, No の回答をもとに回答者の学習スタイルを分類する。分類する際には学習スタイルごとにスコアをつけて、最もスコアが多かったものが学習者の学習スタイルとなる。スコアは各学習スタイルに対応する質問の Yes の数である。

4.2 実装

本研究で実装したものは、LSI の Web 版日本語アンケートページ、アンケート結果をもとにアンケート回答者の学習スタイルを判別するシステム、Visual スタイル用学習教材、Kinesthetic スタイル用学習教材である。下記の図はアンケート回答者の学習スタイルを判別するシステム内部実装図である。

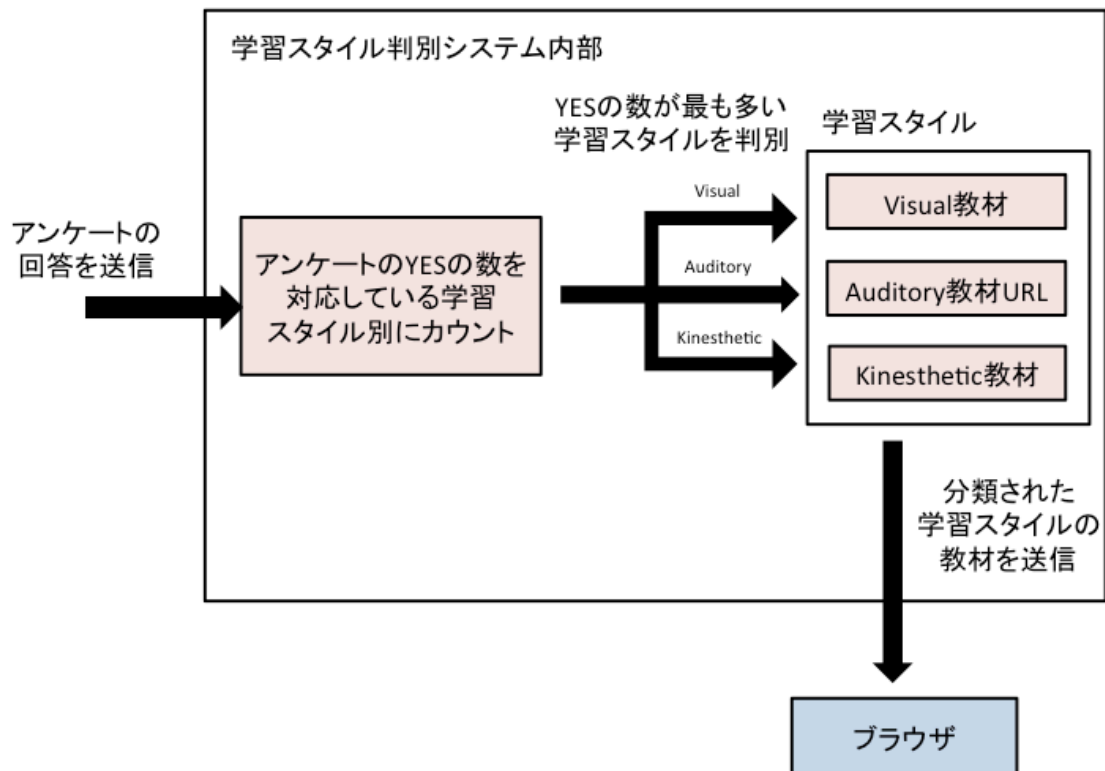


図 4.2: システム内部実装図

4.3 教材概要

本研究はプログラミングの初学者がLSIを活用することで学習効率が上がるかの測定が目的であるため、学習するプログラミング言語をJavaScriptに設定し、学習内容もプログラミング初学者がつまずきやすい「if文」と「for文」に関してのみと設定した。そのため、教材も「if文」と「for文」の項目の教材をVisualスタイル教材とKinestheticスタイル教材にて製作した。Auditory教材は製作が困難であったため、既存のサービス（ドットインストール）で同一の単元を使用した。製作した教材のif文のカリキュラムは下記の流れに沿ったものである。

1. if文とは
2. 基本的なif文について
3. if文の条件について
4. 練習問題1
5. else, else if文について

6. else 文について
7. else if 文について
8. 練習問題 2
9. if 文に関する総まとめ問題

また，製作した教材の for 文のカリキュラムは下記の流れに沿ったものである．

1. for 文とは
2. 基本的な for 文の書き方について
3. for 文の処理について
4. 練習問題 6
5. break について
6. continue について
7. 練習問題 7
8. for 文に関する総まとめ問題

各教材ごとの詳細なカリキュラムは下表のようになる．練習問題はいくつかの選択肢の中から正解を選ぶ選択式の問題を提示しており，総まとめ問題では実際にコードを書かせる問題と選択式の問題を提示している．全ての教材で

	Visual	Auditory	Kinethetik
基本方針	各章ごとに説明をした後に練習問題を出す。最後にはまとめの問題を出す。	動画の特定の時間ごとに練習問題を出す。最後にはまとめの問題を出す。	各節ごとに練習用のプログラムを用意し、それを見ながら打ってもらいながら進めていく。Visualと同じタイミングで練習問題を出す。
章節分け			
if文	1.if文とは	動画を1分まで閲覧	1.if文とは※2
	2.基本的なif文の書き方		2.基本的なif文の書き方※2
	3.if文の条件の書き方		3.if文の条件の書き方※2
	4.練習問題1	練習問題3※1	4.練習問題1
	5.else,else if文の説明	動画を1分50秒まで閲覧	5.else,else if文の説明※2
	6.else文の書き方	練習問題4※1	6.else文の書き方※2
	7.else if文の書き方	動画を最後まで閲覧	7.else if文の書き方※2
	8.練習問題2	練習問題5※1	8.練習問題2
	9.総まとめ問題	総まとめ問題	9.総まとめ問題
for文	1.for文とは	動画を1分10秒まで閲覧	1.for文とは※2
	2.基本的なfor文の書き方		2.基本的なfor文の書き方※2
	3.for文の処理の流れの説明		3.for文の処理の流れの説明※2
	4.練習問題6	練習問題6	4.練習問題6
	5.breakについての説明	動画を最後まで閲覧	5.breakについての説明※2
	6.continueについての説明		6.continueについての説明※2
	7.練習問題7		7.練習問題7
	8.総まとめ問題	総まとめ問題	8.総まとめ問題
備考	※1に関して 動画の内容の都合上、練習問題1,2よりそれぞれの範囲に沿ったものを抜粋する。全体の総問題は変わらない。		
	※2に関して		
	※2が入っているものに関しては、Visualよりも軽い説明のスライド+実際のコードの写経である。Progateを参考としている		

図 4.3: 各教材のカリキュラム詳細

4.3.1 Visual教材について

Visual教材は文章を主体としてそれに補足として図を足す形で学習をしたのちに学習状況を確認するために練習問題を解くという形の流れで学習を進められるものを製作した。if文の各項目と内容は以下の通りである。

1. if文とは
if文の概念的な説明をチャート図と文章を使い説明している..
2. 基本的なif文の書き方
if文の一番基本的な文法や書き方について文章と図を使用して説明している.
3. if文の条件の書き方
if文を設定する上で利用される比較演算子や論理演算子について文章と図を使用して説明している.
4. 練習問題1
ここまでで説明されているif文の書き方と条件文に関する練習問題を3題出題している.
5. else, else if文とは
else, else if文の例外や別条件におけるプログラムの実行について文章と図を使用して説明している.

6. else 文の書き方
else 文の書き方や文法的な説明を文章と図を使用して説明している。
7. else if 文の書き方
else if 文の書き方や文法的な説明を文章と図を使用して説明している。
8. 練習問題 2
else 文や else if 文に関する練習問題を 3 題出題している。
9. if 文に関する総まとめ問題
if 文の書き方や条件における演算子、例外処理について総括した問題を 3 題出題している。

for 文の各項目と内容は以下の通りである。

1. for 文とは
for 文とはどのようなことをするものであるかの概要の説明をおこなっている。
2. 基本的な for 文の書き方
for 文の書き方や構文についての説明をおこなっている。
3. for 文の処理について
for 文におけるプログラムの処理の流れ（ループ処理）について説明をおこなっている。
4. 練習問題 1
for 文の書き方に関する問題と処理の概念に関する練習問題を 3 題出題している。
5. for 文における break について
for 文にて使われる break に関する文法的説明と概念的説明、実際の使用例について説明をおこなっている。
6. for 文における continue について
for 文にて使われる continue に関する文法的説明と概念的説明、実際の使用例について説明をおこなっている。
7. 練習問題 2
for 文にて使われる break と continue に関する練習問題を 3 題出題している。
8. for 文に関する総まとめ問題
for 文の書き方や break, continue を含んだコードに関する総括した問題を 3 題出題している。



図 4.4: Visual 教材の一部

4.3.2 Auditory 教材について

Auditory 教材は開発が困難であったため、Auditory の要素のスコアが高く、ユーザーが多いドットインストールを使用した。教材の進み方が他の学習スタイルの教材と異なっているが、練習問題は他の学習スタイルの教材と同じものだが、順序が違うものを問題として用意した。練習問題を解くタイミングについては動画を何分まで見たら問題を解くのかを Web ページ上にて指定した。動画内容と問題を解く流れの流れは下記に記述した。if 文の各項目と動画・出題問題内容は以下の通りである。

1. 条件分岐としての if 文についての説明と if 文の書き方の説明
ここでは条件分岐として活用される if 文の説明をしつつ、実際に if 文の例をコーディングすることで説明している動画である。また、時間は 1 分 10 秒まで見るように指定した。
2. 練習問題 1
if 文の書き方に関する選択式の問題を 1 題出題している。
3. else, else if 文に関する説明とそれぞれの構文的書き方の説明
ここでは else, else if 文についての説明を実際にこの二つを使ったコーディングの例を用いることで説明している動画である。また、時間は 1 の続きから 1 分 54 秒まで見るように指定している。

4. 練習問題 2

else 文, else if 文に関する選択式の問題を 3 題出題している.

5. 比較演算子と論理演算子

ここでは if 文で使用される比較演算子と論理演算子について説明をしている. また, 時間は最後まで見るように指定している.

6. 練習問題 3

比較演算子と論理演算子に関する選択式の問題を 2 題出題している.

7. if 文に関する総まとめ問題

if 文の書き方や条件における演算子, 例外処理について総括した問題を 3 題出題している.

for 文の各項目と動画・出題問題内容は以下の通りである.

1. ループ処理の一種である for 文についての説明

実際の for 文の例をコーディングし, その上で for 文でどのようなことができるのかということを説明している動画である. また, 時間は 1 分 7 秒まで見るようにしている.

2. 練習問題 1

for 文の書き方と for 文の概念にに関する選択式の問題を 3 題出題している.

3. for 文を抜ける処理の際に使う break と continue についての説明

ここでは else, else if 文についての説明を実際にこの二つを使ったコーディングの例を用いることで説明している動画である. また, 時間は最後まで見るように指定している.

4. 練習問題 2

for 文にて使われる break と continue に関する練習問題を 3 題出題している.

5. for 文に関する総まとめ問題

for 文の書き方や break, continue を含んだコードに関する総括した問題を 3 題出題している.

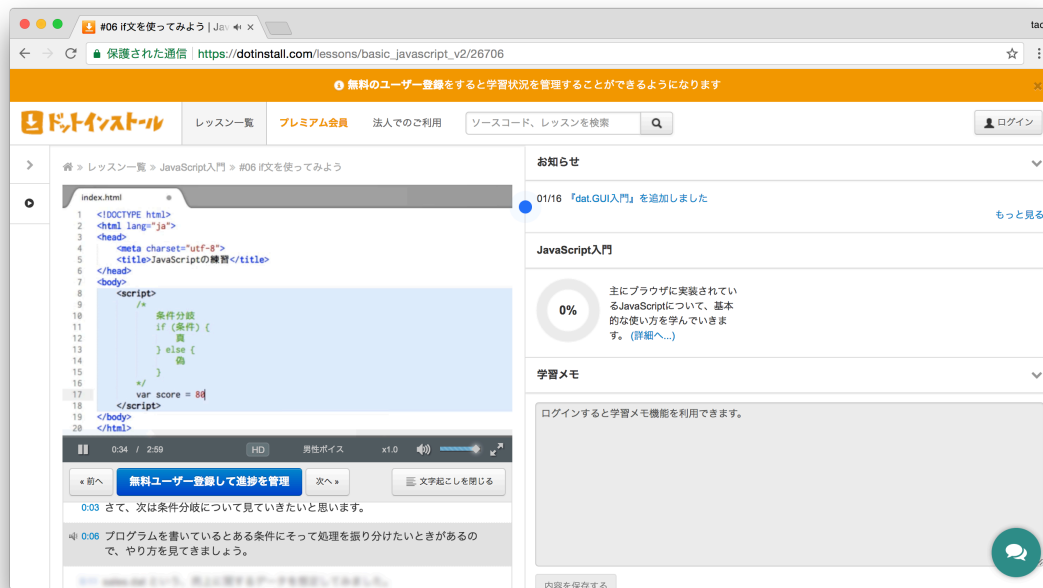


図 4.5: Auditory 教材として使用したドットインストールの画面

4.3.3 Kinesthetic 教材について

Kinesthetic 教材は文章による少量の説明とサンプルプログラムを写すことによって学習した後に、Visual 教材と同様の練習問題を解くという流れで学習を進められるものを製作した。if 文の各項目と内容は以下の通りである。

1. if 文とは
if 文の概念的な説明をチャート図と文章と図を使用して説明している。
※ここでは説明のみでプログラムを写させていない。
2. 基本的な if 文の書き方
if 文の一番基本的な文法の説明を少量の文書と図にて行い、その後サンプルプログラムを写させている。
3. if 文の条件の書き方
if 文の条件を設定する上で利用される比較演算子や論理演算子についての説明を表にて行い、その後サンプルプログラムを写させている。
4. 練習問題 1
ここまでで学習した if 文の書き方と条件文に関する練習問題を 3 題出題している。
5. else, else if 文とは
else, else if 文の例外や別条件におけるプログラムの実行についての説明を少量の文書と図にて行い、その後サンプルプログラムを写させている。

6. else 文の書き方
else 文の書き方や文法的な説明を少量の文書と図にて行い、その後サンプルプログラムを写させている。
7. else if 文の書き方
else if 文の書き方や文法的な説明を少量の文書と図にて行い、その後サンプルプログラムを写させている。
8. 練習問題 2
else 文や else if 文に関する練習問題を 3 題出題している。
9. if 文に関する総まとめ問題
if 文の書き方や条件における演算子、例外処理について総括した問題を 3 題出題している。

for 文の各項目と内容は以下の通りである。

1. for 文とは
for 文とはどのようなものであるかの概要の説明をおこなっている。
※ここでは説明のみでプログラムを写させていない。
2. 基本的な for 文の書き方
for 文の書き方や構文についての説明を少量の文書と図にて行い、その後サンプルプログラムを写させている。
3. for 文の処理について
for 文におけるプログラムの処理の流れ（ループ処理）について文章と図で説明している。
※ここでは説明のみでプログラムを写させていない。
4. 練習問題 1
for 文の書き方に関する問題と処理の概念に関する練習問題を 3 題出題している。
5. for 文における break について
for 文にて使われる break に関する文法的説明を少量の文章と図にて説明後、サンプルプログラムを写させている。
6. for 文における continue について
for 文にて使われる continue に関する文法的説明と少量の文章と図にて説明後、サンプルプログラムを写させている。
7. 練習問題 2
for 文にて使われる break と continue に関する練習問題を 3 題出題している。

8. for 文に関する総まとめ問題

for 文の書き方や break, continue を含んだコードに関する総括した問題を 3 題出題している.

1. if 文とは

プログラミングの構文の一つであり、
「もし〇〇の場合、××してね。」
といった条件分岐を書く時に使う構文です。
これを使う理由は、「ある場合にのみこうしてほしい」という場合があるためです。
例えば
「天気雨が降ったら傘をさす」
といったものです。
多くのプログラミングで使われており、
条件に応じて処理を書くものを if 文と言います。

基本的な if 文の動作

```
graph TD
    Start([スタート]) --> Condition{条件}
    Condition -- true --> Process[処理]
    Condition -- false --> End([終了])
    Process --> End
```

2. 基本的な if 文の書き方

上の「天気雨が降ったら傘をさす」の例を使って、if 文を書く以下ようになります。

```
if (天気 == 雨){
  傘をさす
}
```

図 4.6: Kinesthetic 教材の一部

第5章 評価実験

5.1 評価概要

5.1.1 評価目的

本評価実験の目的は学習者がLSIを用いた学習スタイル判別システムを利用して学習スタイルを判別し、学習者にとって適切な教材をレコメンドすることができるかを測定することである。これには教材に含まれている練習問題の正答数と全学習終了後に行うアンケートを用いて分析する。アンケートでは学習スタイルの判別システムについての是非と学習内容についての項目が存在し、それらの項目と練習問題の正答数を用いて、学習スタイルの判別システムが与えた被験者への効果を測定する。

5.1.2 評価手法

本実験中に9つの練習問題を被験者に解いてもらい、その結果によって被験者の学習内容の理解度を定量的に評価する。全学習終了後にユーザーアンケートを行う。このアンケートにはLSIを用いた学習スタイルの判別システムについてと学習内容や教材についての質問が存在している。アンケートの結果より、LSIを用いた学習スタイルの判別システムについてと学習内容や教材についての2軸で分析を行い、定性的評価を行う。本実験全体のLSIを用いた学習スタイルの判別システムについてと学習内容や教材についての評価はそれぞれを掛け合わせた形で分析を行う。また、プログラミング初学者とそうではない人が本実験に参加しているため、上記の分析はそれぞれの枠でも評価を行う。

5.2 実証実験

5.2.1 実験概要

本実験は実装したLSIを用いた学習スタイルの判別システムを利用し、被験者の学習スタイルを判別する。その後、判別した学習スタイルの教材を使用し、学習をしてもらう。教材には一定の項目ごとに練習問題と単元の最後にまとめの練習問題が存在する。各単元でまとめの練習問題が3問、それ以外の9問ずつ合計18問の練習問題が、それを解いてもらい、正答数のスコアを集計する。実験時間は全部で約1時間にて実施した。全学習終了後に被験者に対してアンケートを行い、回答してもらう。

5.2.2 被験者について

被験者はプログラミング初学者や初心者を中心とした大学1年生以上の28名である。アンケートの被験者のステータスに関する質問事項として学年とプログラミング学習の有無を設けた。学年は大学1年生～大学院2年生、その他の項目を設けた。プログラミング学習の有無については「ない」「あまりない」「少しさわったことがある」「しっかりと学んだことがある」の4段階で項目を設けた。また、その上でプログラミング初心者は「ない」と「あまりない」を選択した被験者とした。この中にはプログラミング初学者も含める。プログラミング経験者は「少しさわったことがある」と「しっかりと学んだことがある」を選択した被験者とした。被験者の各ステータスごとの人数は以下の通りである。

学年

大学1年生：3名
大学2年生：5名
大学3年生：9名
大学4年生：5名
大学院1年生：2名
大学院2年生：1名
その他：3名

プログラミング学習の有無について

ない：11名
あまりない：6名
少し触ったことがある：6名
しっかりと学んだことがある：5名

5.2.3 実験後アンケートについて

実験後のアンケートは以下の表5.1項目にて実施した。

表 5.1: 実験後アンケート項目

質問内容	回答方法
氏名	必須記述
学年	必須選択回答
プログラミングは学んだことがありますか？	必須選択回答
本実験における学習に満足しましたか？	必須選択回答
判別された学習スタイルは何でしたか？	必須選択回答
判別した学習スタイルの納得できましたか？	必須選択回答
他の学習スタイルのほうが学習が効率的だと思いますか？	必須選択回答
本研究のシステムについてどう思いますか？	自由記述回答
テストの難易度は納得できましたか？	必須選択回答
本実験における学習内容についてどう思いましたか？	必須選択回答
if 文は理解できましたか？	必須選択回答
for 文は理解できましたか？	必須選択回答

5.3 実験の結果

本節では実験結果について定量評価と定性評価にて評価を述べる。

5.3.1 定量評価: 被験者の問題の正答率

被験者全体とプログラミング初心者、プログラミング経験者の正答率は以下のグラフ 5.1 の通りであった。

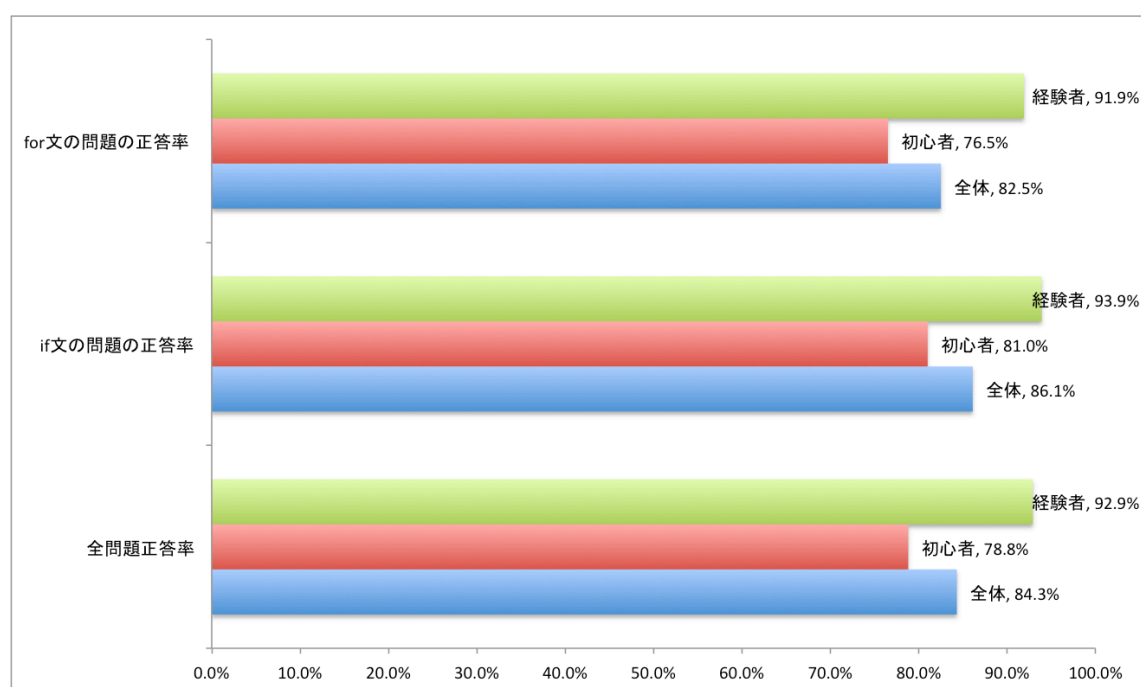


図 5.1: プログラミング経験区分における被験者の問題正答率

上記の結果より、for 文は if 文よりも学習難易度が高いことがわかる。プログラミング経験者は if 文、for 文どちらも平均 90%以上の正答率があることからプログラミング経験者にとっては今回の問題は簡単であったことがわかる。また、プログラミング初心者にとっては if 文よりも for 文が難しいことが for 文の問題の正答率が if 文の問題の正答率よりも約 5%低いことからわかる。

各学習スタイルの判別人数の内訳は以下の表 5.2 である。

表 5.2: 各学習スタイルの判別人数			
被験者カテゴリー	Visual	Auditory	Kinesthetic
全体	12 名	8 名	11 名
初心者	5 名	5 名	6 名
経験者	7 名	3 名	5 名

また、各学習スタイルと学習者のレベルに応じた正答率は以下の図 5.2, 5.3, 5.4 の通りであった。

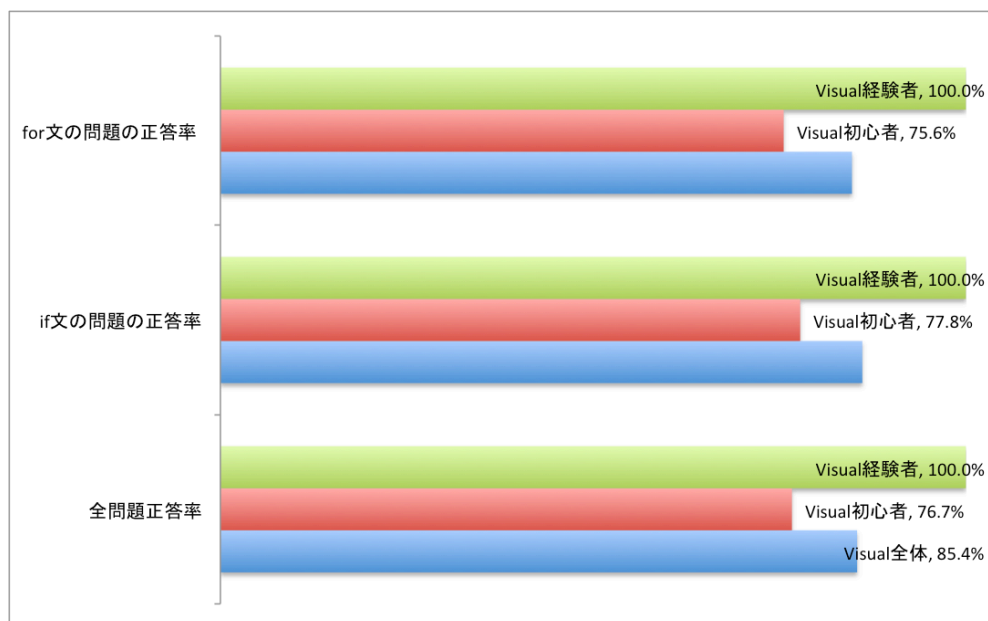


図 5.2: Visual 判別された被験者のプログラミング経験区分における問題正答率

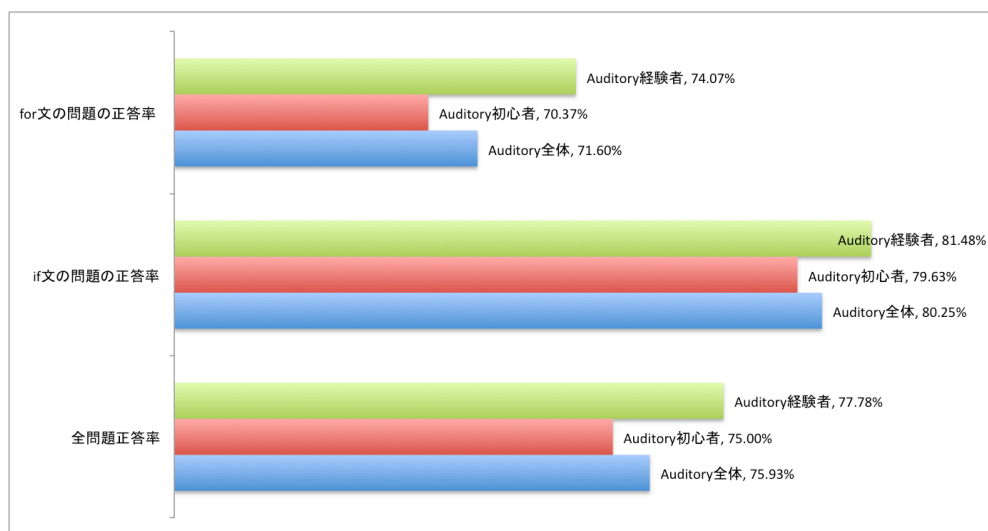


図 5.3: Auditory 判別された被験者のプログラミング経験区分における問題正答率



図 5.4: Kinesthetic 判別された被験者のプログラミング経験区分における問題正答率

上記の結果より，Visual スタイルの学習スタイルは他の学習スタイルよりも被験者全体の学習効果が高いことがわかる．逆に Auditory スタイルの学習スタイルでは全体，初心者，経験者どの被験者カテゴリーであっても正答率が他のものや全体の平均よりも低いことから学習効果が他の学習スタイルと比較して悪いことがわかる．また，プログラミング初心者は Kinesthetic スタイルの教材を利用したほうが学習効果がいいことが全体平均が一番高い Visual 教材よりも正答率が高いことから言える．

5.3.2 定性評価: アンケート結果

アンケート結果からは学習スタイルの判別システムについてと学習内容・教材について分析することができる．

学習スタイルの判別システムについて分析を行う上で以下の手法で評価を行った．判別した学習スタイルの納得できたかという質問に対して他の学習スタイルのほうが学習が効率的であると思うかという質問の回答者の分布は以下の 5.5 となった．

		他の学習スタイルのほうが学習が効率的だと思いますか？					総計
		1 そう思う	2 まあまあそう思う	3 普段と変わらない	4 あまりそう思わない	5 そう思わない	
判別した学習スタイルの納得度	1 できなかった	4	0	0	0	0	4
	2 あまりできていない	2	0	0	0	0	2
	3 普通	0	3	1	1	0	5
	4 まあまあできた	0	1	0	4	0	5
	5 できた	0	0	2	4	6	12
	総計	6	4	3	9	6	28

図 5.5: 判別した学習スタイルの納得度と他のスタイルのほうが学習が効率的であるか

これより、判別された学習スタイルに納得できなかった人（アンケートで「1 できなかった」と「2 あまりできていない」を選択した人）は全員が他の学習スタイルのほうが効率的に学習できるとしており、これらの人は全体の約 21. 4%である。逆に学習スタイルに納得できた人（アンケートで「4 まあまあできた」と「5 できた」を選択している人）のうち約 70. 6%の人が他の学習スタイルのほうが学習が効率的ではない（「4 あまりそう思わない」と「5 そう思わない」を選択している人）としており、これらの人は全体の 50%である。

学習スタイルと判別された学習スタイルの納得度について初心者と経験者と全体の区分を使用し、分析を行う。分析にて使用する数値として納得度のスコアを 5 段階の質問から測定する。質問の選択肢の「1 できなかった」から「5 できた」までを順番に 1 5 のスコアをつけた。（平均値は 3 となり、数値が低いほど納得されていないということになる）判別された学習スタイルの納得できたかという質問による納得度のスコアと実際に判別された学習スタイル、被験者のレベルを掛け合わせたデータは以下の表 5.3 となった。

表 5.3: 判別された学習スタイルと学習の納得度のスコア

	全体	Visual	Auditory	Kinesthetic
初心者	3. 64	3. 60	3. 17	4. 17
経験者	3. 83	4. 00	1. 33	5. 00
全体	3. 68	3. 75	2. 56	4. 55

表 5.3 より、初心者の納得度は全ての学習スタイルにおいて平均値を越している。中でも Kinesthetic の学習手法をとった被験者の平均スコアは 4 を超えており、Kinesthetic の学習分類に分類された人は学習スタイルに納得していることがわかる。経験者の納得度のスコアでは、経験者の Auditory のスコアは初心者、経験者のどの学習スタイルの納得度のスコアと比較しても最も低い結果になっている。また逆に Kinesthetic のスコアは最大のスコアをとっており、Visual のスコアも 4. 00 を獲得しており、分類された学習スタイルに対して納得していることがわかる。被験者全体のスコアは平均値よりも Visual と Kinesthetic が高く、Auditory が下回る結果になっている。下回った原因として

は母数が少ないことによって経験者の Auditory のスコアが大きく影響してしまっていることにある。

学習スタイルと本実験における学習の満足度について初心者と経験者と全体の区分を使用し、分析を行う。分析にて使用する数値として満足度のスコアを5段階の質問から測定する。質問の選択肢の「1 できなかった」から「5 できた」までを順番に15のスコアをつけた。(平均値は3となり、数値が低いほど納得されていないということになる) 本実験の学習内容に満足できたかという質問による満足度のスコアと実際に判別された学習スタイル、被験者のレベルを掛け合わせたデータは以下の表5.4 となった。

表 5.4: 判別された学習スタイルと学習満足度のスコア

	全体	Visual	Auditory	Kinesthetic
初心者	3. 71	3. 60	3. 50	4. 00
経験者	2. 48	3. 00	1. 33	4. 20
全体	3. 46	3. 38	2. 78	4. 09

表5.4 より、初心者の満足度は全ての学習スタイルでスコアの平均値である3よりも高い結果となり、一定の満足度が得られていることがわかる。3種類の学習スタイルの中で Kinesthetic のみスコアが4. 00を越している結果であり、Kinesthetic に判別され、Kinesthetic の教材をやった被験者の満足度が高いことがわかる。経験者の満足度は経験者全体の平均スコアではスコアの平均である3を下回っている。経験者全体の平均スコアが低くなった原因としては母数が少ないことによって Auditory のスコアの低さの影響を受けていることが原因である。一方で Kinesthetic に分類されて学習した人の満足度は4. 20と4を越す結果になっており満足度が高いことを示している。被験者全体の満足度はスコアの平均値である3よりも全体の平均値が高いことから被験者が本実験における学習である一定の満足度を得ていることがわかる。Kinesthetic に分類された人は全体を通して満足度が高い結果になっているが、逆に Auditory に分類された人の満足度のスコアは3をして回っており、全体を通して Auditory の満足度が低いことがわかった。

学習スタイルの種類に注目して、表??と表5.3と表5.4より学習スタイルごとの被験者の実験満足度と学習スタイルの納得度と問題の正答率に関して分析を行う。被験者が Visual に分類され、学習を行った場合は納得度は初心者、経験者共に平均スコアである3を越す結果になっており、被験者は納得している人が多いことがわかる。しかし、学習満足度は納得度と比較すると経験者の場合のみ低い値となっている。これはアンケートの回答で Visual に分類された経験者全員が学習内容についての質問で「1 かなり簡単であった」と回答していることから学習内容が簡単であったことに起因すると言える。また、Kinesthetic と比べ、実際に行うことがすくないため、実験中の学習に学習がつまらなくなってしまうのではないかと考えられる。また、正答率は満足度と関係性がないと考えられる。

被験者が Auditory に分類され、学習を行った場合は納得度と満足度の比較をすると初心者、経験者共に似た値になっているため、関連性があると考えられる。しかし、経

験者に限っては満足度や納得度が正答率に影響を与えないことが言える。

被験者が Kinesthetic に分類され、学習を行った場合には満足度、納得度、正答率全てにおいて高い効果を出しており、それぞれに対して良い影響を与えていたことがわかる。

被験者のプログラミング経験について注目して表 5.3 と表 5.4 と表 ?? より学習スタイルごとの被験者の実験満足度と学習スタイルの納得度と問題の正答率に関して分析を行う。被験者が初心者の場合、学習スタイルの納得度と学習の満足度は似通ったスコアになっており、それぞれのスコアと正答率を比較するとスコアの上下が正答率の上下に連動すると考えられる。

被験者が経験者の場合、学習スコアの納得度のスコアよりも学習の満足度のスコアは全体的に下がることがわかる。今回 Auditory に関してのみ同一のスコアであるが、母数が 3 件と少なすぎるため、明確なことは不明である。また、学習スタイルの納得度や学習の満足度のスコアと正答率を比較すると正答率に影響しないことがわかる。

5.4 考察

LSIにて判別された学習スタイルに対して Visual タイプと Kinesthetic タイプに関しては学習スタイルの判別に被験者は納得している。しかし、Auditory タイプに関しては納得度のスコアが低く出ており、被験者は納得していない。スコア的な原因としては母数が少ないことによってプログラミング経験者の低いスコアの実験結果の影響が大きいと考えられる。

しかし、全体を通して Auditory タイプの納得に関して考えた際に納得できたというアンケート結果は Auditory に分類された被験者 9 名のうち 2 名のみであった。これは Web 上における動画による学習に被験者たちの慣れがないことなどが原因なのではないかと推察される。また、Kinesthetic に分類された人の学習満足度スコアが初心者、経験者共に高いことからプログラミング学習における学習満足度を上げるためには Kinesthetic 型の学習手法が効率よく学習しつつ、満足度を得られている。そのため、実験人数を増やし、実験を行うことで Auditory とその他の学習スタイルのスコアの差を縮められるのではないかと推察される。

また、LSI にて分類された学習スタイルに納得できなかった初心者の学習内容の理解度は高いものではないことがアンケートからわかる。LSI にて分類された学習スタイルに納得している初心者は学習内容の理解度は高いことからプログラミング学習において LSI にて分類する学習スタイルと学習内容の理解度に相関性があることがわかる。このことより、学習者に対する学習スタイルの判別の精度を上げることによって、学習内容の理解度を上げることが可能ではないかと考えられる。また、プログラミング初心者の学習の満足度を上げるには学習スタイルの精度を上げることが必要であり、学習者に対する学習スタイルの判別の精度を上げることで学習満足度、学習効率を上げることにつながるのではないかと考えられる。そのため、学習スタイルの判別の精度を上げるには既存の LSI の質問項目をプログラミング学習する上での学習スタイルの判別を目的した質問項目に変えていく必要がある。変えていく際には他の学習スタイルの判別用に質問を使う手法や現在の LSI の質問項目を変えていく手法が考えられる。

また、プログラミング経験者に関しては、学習の満足度と学習スタイルの関連性についてアンケート上の自由コメントから、プログラミングの学習をする際に、実際にコーディングをしながら学ぶことが学習の効率と満足度の向上に繋がると判明した。プログラミング初心者とプログラミング経験者でプログラミング学習のアプローチが変わるのだと考えられる。経験者はプログラミングの学習手法が確立しており、そのためプログラミング学習をする上での動作や思考が学習手法に適するようになっているのだと推察される。そのため、プログラミング以外に使われている学習スタイルを、プログラミング学習に使用しても学習の納得度が下がった上、学習効率が落ちるのではないかと考えられる。逆に初心者はプログラミングの学習手法が確立されていない。学習者自身にとっての最適な学習方法をプログラミング学習で行うことにより、プログラミング学習をする上で改めて学習手法に動作や思考の最適化の手間が省けるため、効率的に最初の学習が進む。その結果、効率的に学習内容の理解度を高めることができ、プログラミング学習で挫折する可能性が低くなるのではないかと考えられる。

第6章 結論

6.1 本研究のまとめ

本研究では学習スタイルに注目し、LSIを用いた教材レコメンドシステムの提案を行った。本システムはLSIにて学習スタイルを判別し、その結果に沿った教材を学習者にレコメンドすることによってプログラミング初学者にとって個人に最適な学習手法の提案を行った。

本研究でLSIがプログラミング学習に有効であるかの評価を行うためにプログラミング初学者を中心とした大学1年生以上の被験者（28名）に、JavaScriptのif文とfor文に関する学習をしてもらい、評価実験を行った。評価実験では「LSIを用いたシステムにて学習者の学習スタイルを判別し、判別された学習スタイルの教材を使うことで、プログラミング初学者の学習効率の向上が可能ではないか」という仮説のもと、実験を行った。評価実験の効果測定の一環として問題を一定数（合計18問）出題し、その正答率を測定することと全学習終了後にアンケート調査を行った。この問題の正答率とアンケート調査の結果をもとに、プログラミング初心者と経験者の回答結果をもとに分析を行った。

分析の結果、プログラミング初心者と経験者では学習の満足度を得る要因が異なっていることがわかった。プログラミング初心者は学習スタイルの判別の納得度と学習の満足度と問題の正答率がほぼ同じように推移するため、学習スタイルの判別の精度を上げることによって学習の満足度や学習の効率が上がることが明らかとなった。

学習スタイルの判別の精度を上げるには既存のLSIの質問項目をプログラミング学習する上での学習スタイルの判別を目的した質問項目に変えていく必要がある。変えていく際には他の学習スタイルの判別用に質問を使う手法や現在のLSIの質問項目を変えていく手法が考えられる。

また、一方でプログラミング経験者は実際にコーディングを行いながら学習することで学習の満足度や効果が上がることがわかった。これより、プログラミング初心者と経験者は学習するにあたって違ったアプローチが必要だと判明した。

6.2 今後の課題と展望

本研究ではプログラミングの初学者の学習効果が学習スタイルの判別システムを導入することで学習スタイルに納得がいった人は満足度や学習効果の向上をさせることができることがわかった。その上で本研究の課題点を述べる。

本研究では実験環境の都合により JavaScript の教材の一部の単元でしか実験をすることができなかったため、一番最初からある程度のプログラムが組めるようになるまでで実験を行う必要があると考えられる。それによって、学習スタイルの判別したスタイルによる勉強手法が最初から最後まで有効であるか測定することができる。そして、被験者を増やすことでより正確なデータが取れるため、実験のスケールを大きくした実験をする必要性がある。

また、学習スタイルの判別も別の手法が幾つも存在するため、それらを組み合わせることで学習スタイルの判別の精度を上げていく必要がある。

謝辞

本論文の執筆にあたり、ご指導頂きました環境情報学部教授村井純博士、同学部教授楠本博之博士、同学部教授中村修博士、同学部准教授 Rodney D. Van Meter 博士、植原啓介博士、同学部教授三次仁博士、同学部教授武田圭史博士、同学部講師齊藤賢爾博士、政策・メディア研究科特任准教授鈴木茂哉博士、同研究科特任准教授佐藤雅明博士、Achmad Husni Thamrin 博士、同研究科特任助教中根雅文博士、同研究科特任助教工藤紀篤博士、同研究科特任助教佐藤貴彦博士、同研究科特任講師空閑洋平博士、同研究科特任講師松谷健史博士、同研究科元特別招聘教授 下村健一氏、に感謝致します。

特に研究について日頃からご指導賜りました Achmad Husni Thamrin 博士、齊藤賢爾博士、工藤紀篤博士、下村健一氏に重ねてお礼申し上げます。徳田・村井・楠本・中村・バンミーター・植原・三次・中澤・武田合同研究プロジェクトにおいてご指導頂きました、村井純研究会の諸氏皆様、及び、環境情報学部客員教授徳田英幸博士、同学部准教授高汐一紀博士、同学部准教授中澤仁博士、加えて、研究室で声をかけてくださったり、金沢大学総合メディア基盤センター教授 (SFC 研究所上席所員) 大野浩之博士に感謝致します。様々な場面で研究生活を支えて頂きました、JUNSEC の皆様に感謝致します。村井純研究室の先生方には村井純研究室に所属して3年半の間、様々なご指導をいただきました。

村井純研究室 CAL グループで共に研究をした政策・メディア研究科重田桂子氏、総合政策学部鎧坂文菜氏、環境情報学部土田悠輝氏、総合政策学部加藤新菜氏、環境情報学部椎葉瑠星氏に感謝いたします。特に重田桂子氏には卒業研究に関して大変お世話になっており、深く感謝いたします。CAL グループの皆様がいなければ本論文を書ききることはできませんでした。

本論文の実験に協力して下さった皆様に感謝致します。

また、2年の秋から2年間の間お世話になった村井純研究室 NECO の卒業生後上峻一氏、卒業生諸澤正樹氏、卒業生渡部貴博氏、卒業生塚越広彬氏、卒業生相川美菜子氏、卒業生加藤奈美氏、卒業生田中公人氏、卒業生板垣孝明氏、政策・メディア研究科阿部涼介氏、卒業生瀬川雅弘氏、総合政策学部尾崎周也、総合政策学部豊田安信氏、総合政策学部菅藤佑太氏に感謝致します。NECO の皆様がいなければ現在私はこの研究室にいなかったと思います。また、特に様々な相談に乗ってもらった阿部涼介氏に感謝致します。

研究室生活の中で共に過ごした重田桂子氏、阿部涼介氏、尾崎周也、豊田安信氏、菅藤佑太氏、政策・メディア研究科澤井優作氏、総合政策学部桑原誠尚氏、環境情報学部鈴木雄祐氏、総合政策学部森島隆成氏に感謝致します。皆様と楽しく研究生活を過ごせたことにより、本論文を書ききることができました。

アインクラインスオーケストラの同期、先輩、後輩に感謝致します。特に卒業生向美穂氏、卒業生沖幸太郎氏、卒業生中尾圭志氏、卒業生久島彩希氏、総合政策学部伊藤栞氏、卒業生加藤優佳氏、環境上法学部宮崎真和氏、総合政策学部田原宏基氏、総合政策学部大和田純平氏に特に感謝致します。皆様と共にオーケストラサークルで過ごしたことで非常に楽しい大学生活を送ることができました。何にも代えがたい時間を得ることができました。

TechKidsSchoolの社員の方、先輩、同期、後輩に感謝致します。特に松倉健悟氏、小宮山貴史氏、岩城拓海氏、後藤孟毅氏、石橋萌氏、永野亮介氏、大竹博斗氏、小林和輝氏、山岸明夢氏、野並将志氏、久畑凌氏、遠藤菜奈氏、金森郁花氏に感謝致します。皆様のおかげでプログラミング教育に関して興味を持つことができました。また、皆様と一緒に働けたことによって日々が充実感にあふれていました。

最後にこれまでの人生を支えてくれた家族に感謝致します。今まで様々な迷惑をかけ、人よりも大学卒業が遅くなりつつも、何一つ不自由なく過ごせたのは支えてくれた家族がいたからこそだと考えています。おかげで悔いのない大学生活を過ごすことができました。改めて深く感謝致します。

以上を持って本論文の謝辞とさせていただきます。

参考文献

- [1] 内閣府政策統括官（経済財政分析担当）. 日本経済 2016 – 2017 第 2 章 新たな産業変化への対応. http://www5.cao.go.jp/keizai3/2016/0117nk/n16_2_1.html, 1 2017. [Online; accessed 21-12-2017].
- [2] 経済産業相. It 人材の最新動向と将来推計に関する調査結果. <http://www.meti.go.jp/press/2016/06/20160610002/20160610002.pdf>, 6 2017. [Online; accessed 21-12-2017].
- [3] 林 晋, 黒川 利明. 日本の危機としての it 人材問題.
- [4] TechAcademy マガジン. オバマ、ジョブズ、ザッカーバーグが語る「プログラミングを学ぶ必要性」とは【動画紹介】. <https://techacademy.jp/magazine/1260>, 2013. [Online; accessed 21-12-2017].
- [5] 猪狩 知也, 速水 治夫. プログラミング入門者向けのオンラインジャッジシステムによるプログラミング学習支援. https://www.google.co.jp/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=0ahUKEwjWt-yBzuvYAhWLbbwKHVqoBGgQFgg-MAE&url=https%3A%2F%2Fipsj.ixsq.nii.ac.jp%2Ffej%2F%3Faction%3Drepository_action_common_download%26item_id%3D105003%26item_no%3D1%26attribute_id%3D1%26file_no%3D1&usg=AOvVaw12CNjph1r83uf-BV1yYjnN, 2014. [Online; accessed 21-12-2017].
- [6] 杉浦 学, 松澤 芳昭, 岡田 健, 大岩 元. アルゴリズム構築能力育成の導入教育. <http://crew-lab.sfc.keio.ac.jp/papers/data/Manabu%20Sugiura/IPSJ-JNL4910008.pdf>, 2008. [Online; accessed 19-1-2018].
- [7] 西田知博, 松浦敏雄. 初学者のためのプログラミング学習環境 pen. <http://www.cmc.osaka-u.ac.jp/publication/for-2006/11-16.pdf>. [Online; accessed 19-1-2018].
- [8] 兼宗 進, 御手洗 理英, 中谷 多哉子, 福井 眞吾, 久野 靖. 学校教育用オブジェクト指向言語「ドリトル」の設計と実装. <http://kanemune.eplang.jp/data/pro4211.pdf>. [Online; accessed 19-1-2018].

- [9] 松澤芳昭, 酒井三四郎. ビジュアル型言語とテキスト記述型言語の併用によるプログラミング入門教育の試みと成果. <https://ci.nii.ac.jp/naid/110009551344/>, 2013. [Online; accessed 19-1-2018].
- [10] 土肥 紳一, 宮川 治, 今野 紀子. プログラミング学習における学生のつまずき箇所の分析. https://www.jstage.jst.go.jp/article/jjsei/10/4/10_KJ00002152227/_pdf/-char/ja, 1995. [Online; accessed 19-1-2018].
- [11] 田口 浩, 糸賀 裕弥, 毛利 公一, 山本 哲男, 島川 博光. 個々の学習者の理解状況と学習意欲に合わせたプログラミング教育支援. <https://ci.nii.ac.jp/naid/110006207816>, 2007. [Online; accessed 19-1-2018].
- [12] 土肥 紳一, 宮川 治, 今野 紀子. プログラミング入門教育におけるモチベーションと成績との関係の分析. <https://ci.nii.ac.jp/naid/130005434365>, 2016. [Online; accessed 19-1-2018].
- [13] 柏原 昭博, 久米井 邦貴, 梅野 浩司, 豊田 順一. プログラム空欄補充問題の作成とその評価. https://www.jstage.jst.go.jp/article/tjsai/16/4/16_4_384/_article/-char/ja/. [Online; accessed 19-1-2018].
- [14] 岡本 雅子, 村上 正行, 喜多 一, 吉川 直人. 初学者を対象とした自習中心のプログラミング教育の教材開発と評価. <https://ci.nii.ac.jp/naid/170000082703>, 2001. [Online; accessed 19-1-2018].
- [15] 辻仁志, 喜多一. プログラムによる計測・制御のための raspberry pi 用写真型学習教材の開発. https://ipsj.ixsq.nii.ac.jp/ej/index.php?active_action=repository_action_common_download&page_id=13&block_id=8&item_id=107171&item_no=1&attribute_id=1&file_no=1, 2014.
- [16] 岡田 有加. 学習スタイル. https://opac.library.twcu.ac.jp/opac/repository/1/5426/OKADA_Yuka20120301.pdf, 2012. [Online; accessed 19-1-2018].
- [17] 青木 久美子. 学習スタイルの概念と理論 — 欧米の研究から学ぶ. <https://ci.nii.ac.jp/naid/120005486267/>, 2005. [Online; accessed 19-1-2018].
- [18] Coffield F., Moseley D., Hall E., Ecclestone K. Learning styles and pedagogy in post-16 learning: A systematic and critical review., 2004.
- [19] A.R Gregorc. Style delineator. maynard, ma: Gabriel systems., 1982.
- [20] R Dunn. Synthesis of the dunn and dunn learning styles model research: Who, what, when, where and so what? dunn and dunn learning styles model and its theoretical cornerstone. new york: St. john ' s university, 2003.

- [21] Stephen Rayner Richard Riding. Cognitive styles and learning strategies: Understanding style differences in learning and behaviour, 1998.
- [22] M.J. Apter. Motivational styles in everyday life: A guide to reversal theory. washington d.c.: American psychological association., 2001.
- [23] Chris J. Jackson. Understanding the learning styles profiler (lsp), 2016.
- [24] Saul McLeod. Kolb - learning styles, 2013.
- [25] Honey and mumford — university of leicester. <https://www2.le.ac.uk/departments/gradschool/training/eresources/teaching/theories/honey-mumford>.
- [26] Mumford A. Honey P. The manual of learning styles 3rd ed. maindenhead, peter honey, 1992.
- [27] Herrmannsolutions. <http://www.herrmannsolutions.com/what-is-whole-brain-thinking-2/>.
- [28] Hayes J. Allison C. The cognitive style index. journal of management studies,, 1996.
- [29] Assist pagemaker - enhancing teaching-learning environments.
- [30] Dianne Geraldine R. Gullas. Vermunt ’ learning styles model and its implication to teaching-learning activities.
- [31] 和田 裕一 落合 純, 真家 優子. 思考スタイル質問紙日本語版の信頼性・妥当性の検討, 2016.
- [32] Thinking styles by robert j. sternberg - cambridge university press, https://www.google.co.jp/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=0ahUKEwjrp4X9v0zYAhVPQLwKHSEnDjcQFggzMAE&url=https%3A%2F%2Fwww.cambridge.org%2Fcore%2Fbooks%2Fthinking-styles%2F1475ED3555FF4DA3DEA59DE0B635EF43&usg=A0vVaw1xSdZeD432huh_4c9C_xU2.
- [33] D.A. Kolb. Experiential learning theory and the learning style inventory.
- [34] TechAcademy マガジン. 始める前に要チェック！プログラミングで挫折する3つのパターンとその解決方法. <https://techacademy.jp/magazine/6342>, 6 2017. [Online; accessed 21-12-2017].
- [35] コードモンキー. <https://codemonkey.jp/>.

- [36] paiza ラーニング. <https://paiza.jp/works>.
- [37] Schoo. <https://schoo.jp/guest>.
- [38] Progate. <https://prog-8.com/mypage>.
- [39] ドットインストール. <https://dotinstall.com/>.
- [40] Codeprep. <https://codeprep.jp/>.
- [41] codecademy. <https://www.codecademy.com/>.
- [42] Udacity. <https://www.udacity.com/>.
- [43] Progra! <https://progra.org/>.
- [44] シラバス. <https://cyllabus.jp/>.
- [45] Codestudy. <https://jeek.jp/school/programming>.
- [46] コーディング道場. <https://coderdojo.jp/>.
- [47] ミニツク. <http://www.minituku.net/>.
- [48] Sharewis. <https://share-wis.com/>.
- [49] 動学. <http://dougaku.tv/html/faq/title004.html>.
- [50] Udemy. <https://www.udemy.com/jp/>.
- [51] Skill hub. <http://skillhub.jp/>.
- [52] Codingame. <https://www.codingame.com/>.
- [53] Flappy code. <https://studio.code.org/flappy/1>.
- [54] Fight code. <http://beta.fightcodegame.com/>.
- [55] Moonblock. <http://www.moonblock.jp/>.
- [56] ウェブカツ!! <https://webukatu.com/>.
- [57] Treehouse. <https://teamtreehouse.com/>.
- [58] プログラミング勉強ならコレ！無料学習サイトおすすめランキング top5. <https://www.sejuku.net/blog/796>.
- [59] 【エンジニアが教える】プログラミング入門におすすめ無料学習サイト 10 選. <https://tech-camp.in/note/engineer/231/>.

- [60] 【無料で学べる！】独学でプログラミングが学習できるサービス 10 選.
http://kojima1992.com/%E3%83%97%E3%83%AD%E3%82%B0%E3%83%A9%E3%83%9F%E3%83%B3%E3%82%B0%E7%84%A1%E6%96%99_%E7%8B%AC%E5%AD%A6%E5%AD%A6%E7%BF%92%E3%82%B5%E3%83%BC%E3%83%93%E3%82%B9.
- [61] Academic Success Services. Learning styles inventory.

卒業論文

題名：

Learning Style Inventory を用いたプログラミング初学者のための
教材レコメンドシステムの提案

氏名：押見 太雄

サブ親氏名：Achmad Husni Thamrin

押見 太雄君の卒論仮綴提出を認めます。

親：Achmad Husni Thamrin



コメント：

本研究は、学習スタイルを用いたレコメンドシステムと教材を提案、実装と、実証実験を行った。LSI はプログラミング学習にはあまり研究されていないと思うが、LSI はプログラミング学習分野で、卒論レベルで十分できている。

卒業論文

題名：Learning Style Inventory を用いたプログラミング初学者のための
教材レコメンドシステムの提案

氏名：押見 太雄

サブ親氏名：斉藤 賢爾

押見君の卒論本綴提出を認めます。

サブ親：斉藤 賢爾



コメント：

本研究は、学習者の学習パターンを予め把握することを通して最適なプログラミング入門の教材をレコメンドするというものであり、プログラミングという行為が一般に馴染みのあるものになりつつある現在、社会的な意義が大きい。

本研究は、そうした意義の大きな課題について取り組んだものであり、学術研究として実験の構成などに不十分な面もあるものの、卒業研究としては十分に研究活動を行ったと認められる。

卒業論文

題名：

Learning Style Inventory を用いたプログラミング初学者のための
教材レコメンドシステムの提案

氏名 : 押見 太雄

サブ親氏名 : 重田 桂子

押見 太雄君の卒論仮綴提出を認めます。

サブ親：重田 桂子

重田

コメント：

年明け前後の体調不良が心配でしたが、最終的には論文として仕上げる事ができて、
よく頑張りました。

教育課程にいる身で教育関連の研究を行うのが大変なことや、評価を行うときに、被験者
の人数が必要なことも身にしみて分かります。

ただ、小規模でも誰も試したことのないことをやったことに価値があるので、卒業論文で
取り組んだことに自信を持ってください。

お疲れ様でした。