

プログラム機能の自動分析機能とプログラム概念の自動評価機能を持つScratch用プログラミング学習支援システム

小学校からプログラミング教育が始まるような話があるけど、どうしよう。機械音痴の私にプログラムを教えるなんて無理無理……



- 太田 剛:放送大学 修士課程
- 森本 容介:放送大学
- 加藤 浩 :放送大学



私たちがお手伝いします。

1

1.背景: 出発点と現状

初等教育でのプログラミング教育の動き

- 英国、オーストラリア等で、初等教育でのプログラミング教育を含む新しい情報科目の必修化。
- 政府が2016年6月に産業競争力会議が新しい成長戦略で2020年からの初等中等教育でのプログラミング教育の必修化が明記。
- 文部科学省も「小学校段階における倫理的思考力や創造性、問題解決能力等の育成とプログラミング教育に関する有識者会議」を立ち上げ。
- 文科省の問題指摘(2015)
 - ①プログラミング学習を担当する**教員の指導力**
 - ②プログラミング**学習に適した教材**
 - ③社会の変化に伴うプログラミング学習の**目標・内容**

2

1.背景: 国内のプログラミング教育の基本的な方針 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)

- 「プログラミング的思考」などを育むこと
- 「プログラミング的思考」 (いわゆる「**コンピュータショナル・シンキング**」の考え方を踏まえつつ、プログラミングと論理的思考との関係を整理しながら提言された定義である。) <= 情報教育の三本柱(情報活用の実践力、情報の科学的な理解、情報社会に参画する態度)と理数教育の推進(論理的思考の育成)からのしがらみか?
- 各教科等で育まれる思考力を基盤としながら基礎的な「プログラミング的思考」を身につけること => **専門の教科は作らない** = 平成10年の小学校の学習指導要領の時の情報教育といっしょでは、デジャブ?

3

ちょっとデモ: 対象言語

スクラッチはMITのメディアラボが開発した子供向けのビジュアル言語であり、スタンドアロン及びWebで使用が可能である。国内を含め全世界で広く利用され、Webのコミュニティには1100万人以上が登録し、1600万以上のプログラムが共有されている。



例えば上手は簡単なシューティングゲームであるが、スプライトと呼ばれるキャラクターを画面上に置き、それぞれのスプライトがどのように動作するかブロックを使ってプログラムするものである。

4

使ってみよう: コード忍者の里 for Scratch



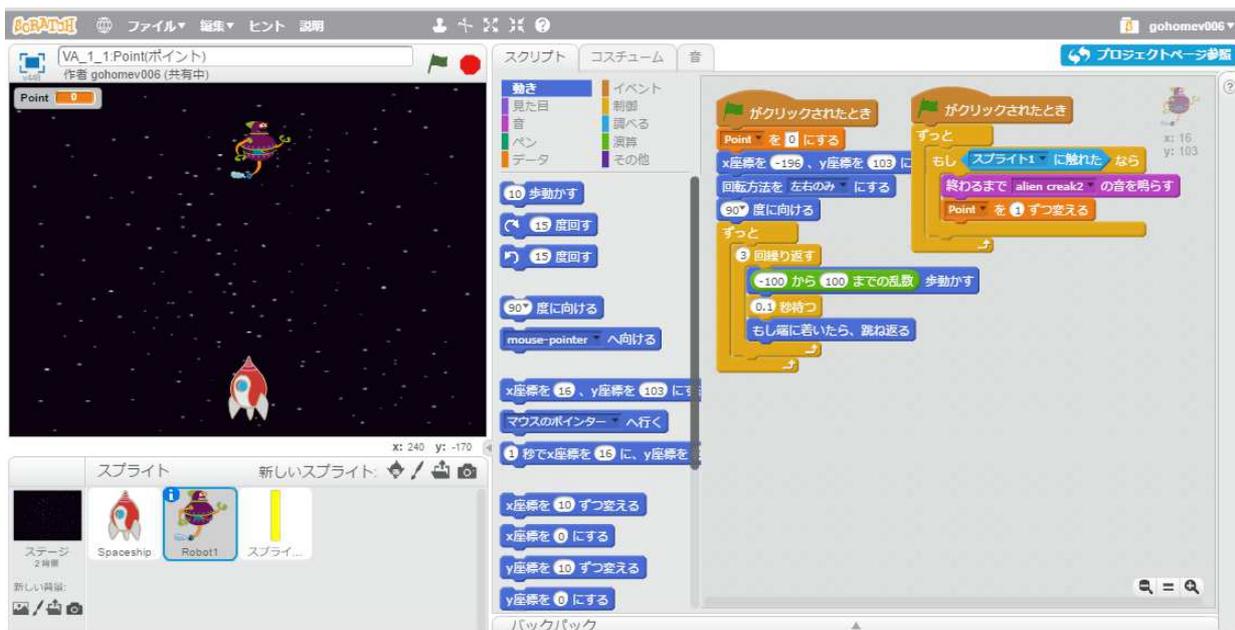
検索 コード忍者の里

「コード忍者の里 for Scratch」はPC上で作成又はダウンロードしたファイル、又はプロジェクトIDを指定すると分析できる。自分のプログラムが無い場合でもプロジェクトIDは、Scratchの個々のプログラム(プログラム)につけられた番号で、上手のような、画面に表示される番号を入力することで診断することもできる。

“コード忍者の里 for Scratch”. <http://tk2-249-34225.vs.sakura.ne.jp/ncv4s/>

5

簡単なシューティングゲームを診断



術の巻物 データ:ポイントプロジェクトID:109784985

6

コード忍者の里で診断

NinjaCode Village
術の巻物
プログラム診断
修業の記録シート
サイトについて

コード忍者の里 for Scratch



プログラム診断
君や友達のプログラムが、どのようなプログラムの考え方を使っているか、どのような術(じゆつ)を組み合わせているか自動的に診断(しんだん)します。Scratch 2.0に対応です。プロジェクトIDも指定できます。複雑なプログラムの場合は診断に時間がかかることがあります。

プロジェクトIDを指定する時

プロジェクトID

何年生ですか
選んでください???

自分で作ったプログラム?
選んでください???

プログラムのタイプ
選んでください???

[診断開始](#)

プログラムのファイルがPCにある時

ファイル名
ファイルを選択 (VA_1_1-P...イント).sb2

何年生ですか
選んでください???

自分で作ったプログラム?
選んでください???

プログラムのタイプ
選んでください???

[診断開始](#)

二つの診断結果

プログラマーの技

#	1	2	3	4
分岐	★			
ループ	★	★	★	★
モジュール	★			
モデル/モジュール共有				
データ利用	★			
発動・トリガー	★			
連携・同期				
ユーザインタフェース				

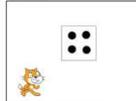
プログラム内の術

MO_1_2: 壁反射
OP_1_1: 逃げる
VA_1_1: ポイント
RN_1_1: サイコロ

術の表示



術のリスト

歩く 壁反射	操り 逃げる	データ ポイント	ランダム サイコロ
			
端までいくと、方向を変えます。 プロジェクト ID:105754746	方向キーで操作します。 プロジェクト ID:108728181	ゲームの得点を計算します。 プロジェクト ID:108794985	サイコロ振ります。 プロジェクト ID:108970730

プログラミング概念

	1	2	3	4
分岐	If	If - else	論理演算子	If (- else)の入れ子
ループ	無限ループ	ループ回数指定	終了条件付きループ	ループの入れ子
モジュール	複数のスプライト [オブジェクト]	ブロック[サブ ルーチン]	引数のあるブ ロック	クローン[オブ ジェクトのコ ピー]
モデル/モ ジュール共有	1つのスクリプト で複数のブロック	異なるスクリプト でブロックの 利用	異なるスプライト で同一ブロッ ク利用	ブロックの再帰 的呼び出し
データ利用	変数利用	異なるスプライト で変数共有	リスト型変数利 用	クラウド変数利 用
発動・トリ ガー	特定キーにより起 動	マウス操作によ る起動	背景変化による 起動	タイマー等によ る起動
連携・同期	背景変化を介した 複数スクリプトで の連携	同一スプライト でのメッセージ の使用	他のスプライト へのメッセージ での連携	メッセージ後の Waitでの連携
ユーザインタ フェース	文字入力の使用	キーの検出	マウスクリック の検出	マウスの座標位 置の利用

9

情報教育・プログラミング教育の目標

[日本]

● プログラミング的思考

自分が意図する一連の活動を実現するために、どのような動きの組合せが必要であり、一つ一つの動きに対応した記号を、どのように組み合わせたらいいのか、記号の組合せをどのように改善していけば、より意図した活動に近づくのか、といったことを論理的に考えていく力

[英国・オーストラリア等]

● コンピューショナル・シンキング

“複雑かつ雑然で、一部しか見えない現実問題を、知能のないコンピュータだけで問題に取り組めるような指示に変換する、いくつかの知的能力の集合(英国)”



本システムではこちらの考え方をベースに作成する

プログラミング教育の目標

● 原田

「コンピュータとはなにか」ということを知るためにプログラミングを学ぶべき

太田の感想:
バスケットは一般の言語に比べて知能がある。

● コンピュータショナル・シンキング

現実問題を, 知能のないコンピュータやらせるためのプログラムを作るにはどうしたらよいか学ぶ



その考え方や手法はコンピュータを使わない場合でも役に立つ

プログラマー的思考

11

コンピュータショナル・シンキングをもとにした各国カリキュラムのキー概念 ≡ 知的能力の集合

UK Computer Progress Pathway	Australian Curriculum	CSTA(USA) *1
Abstraction	Abstraction	Abstraction
Decomposition		
Generalization		Modeling and Simulation
Algorithmic thinking	Specification, algorithms	Algorithms
		Problem solving
	Data collection, representation and interpretation	Data representation
Evaluation		

Wing(2006)が明確に定義していない、広範囲に言及

12

本システムでの コンピューティショナル・シンキングの考え方

プログラミング的思考の具体的な能力は、コンピューティショナル・シンキングの中にあると考え、本システムの目標として以下の具体的なキー概念を選んだ。

- **抽象化 (Abstraction)** : 問題解決では、問題を解きやすくするため、問題や事象の適切な側面・性質だけを取り出し、他の部分を捨てること。
- **モデル化 (Modeling)** : 抽象化の結果であり、問題解決では問題や事象を簡潔に理論的に表現すること。
- **デコンポジション (Decomposition)** : 大きなまたは複雑な問題や事象を理解または解決可能な細部に分割すること
- **アルゴリズム (Algorithms)** : 問題を解決するための明確な手順で、同様の問題に共通して利用できるものである。

13

機能サンプルプログラム集と自動機能分析

- 生徒がプログラムを作る場合に必要な個々の機能を含んだ小さなプログラムをサンプル集として提供することで、プログラム作成作業を手助けする
- プログラム分析機能により、生徒はプログラムがいろいろな機能から構成されることを気づかせ、子どもは作りたいプログラムに必要な機能をサンプル集の中から探して組み込むことで、大きく複雑なプログラムも容易に作成できる
- 子どもがお手本となるプログラムを見つけた時、それがどんな機能から作られているか知ることができる

14

プログラミング教育の教材・学習方法

[スモールステップ]

「特に、教員は目標を細分化し、小さな目標を達成する体験を積み重ねながら最終目標に近づくようスモールステップで課題を設定することで、児童生徒の『プログラミングは難しい』という思い込みを払拭させ、自分にもできるという自己効力感を高めさせているようです」(文部科学省のプログラミング教育実践ガイド)

[メイキング]

「単なる操作できる教具を与えるだけでなく、学習者が何か意味あるものを作り出す時に教育は最も効果がある」(Papertの構築主義)
 「簡潔でゆるい課題設定において、コンピュータを一つの素材として、“いじくりまわし”，自分の好みで、考えて作っていく活動」(Sylvia and Gary)

↑
 本システムでは創造性の杭性の観点等から、この学習方法を支援

15

アクティブラーニングから見たスモールステップとメイキングの比較

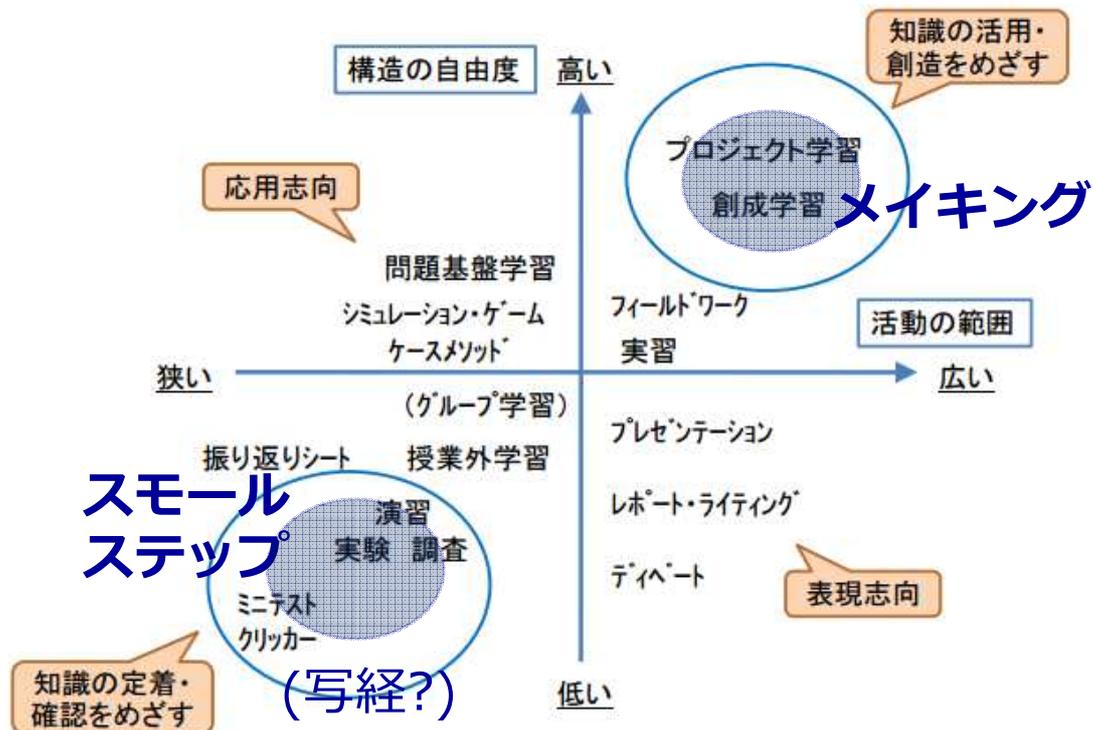


図1 アクティブラーニングの多様な方法

16

プログラミング教育の教材・学習方法:問題点

[スモールステップ]

- 「(Hour of Codeの) パズル自体を自分たちで作らだせるようなメタな仕組みは提供されていない。そのため、解き終わった子供たちは短い時間で飽きてしまった。」(阿部)
- 今までの教育と同じで落ちこぼれやプログラミング嫌いを作るのでは(太田の考え)

[メイキング]

- 「習得度にばらつきがでる」や「基本的なテクニックを理解していない」(みどりっ子クラブ)
- Scratchの場合、ぬり絵や電子絵本のレベルで留まる子供
- 「ブロックを適当に組み合わせて不規則に動く作品を子供の創造性と誤認(教師側の過去の失敗原因)」(兼宗, 阿部, 原田)

本システムでは、この問題点に対応

17

メイキングの問題点の解決方法

みどりっ子クラブ

- 各習得レベルに対応しプログラミング方法のサンプル提供
- プログラミング方法を「技」として技一覧表を作り、技の習得度合を判定する検定の実施

Scratchの自動評価

- MITのプログラミング場面での評価フレームワーク
- Wolz(2011) Scratchのブロックの自動分析
- Moreno-León(2015) Dr.Scratch プログラム概念の自動評価ツール(評価結果の得点化)
数週間Scratchを学習した生徒に対して1時間のワークショップを実施。ワークショップの前後で得点上がる。

本システムでは、これらを発展させシステム化したもの

18

教育の指導力の問題への対応

[人的リソースの強化]

- 教員研修による教師のプログラミング能力の育成
- 支援体制と支援要員の整備

「若年層に対するプログラミング教育の普及推進」事業(総務省)

[システムでの支援]

プログラミング能力の低い教師を支援

- スモールステップのパズル Hour of Code等
- 正答、正しいプログラムへの自動評価

従来の高等教育用等

- プログラミングレベルの自動評価

どのようなプログラムか評価し、生徒と教師にフィードバック



本システムでこの考え方をベースに作成する

おまけ:従来の自動評価との違い

真正(Authentic)な学習内容、学習場面、学習評価

- 従来は学校知の文脈で、全員に共通した目的を教師が教える。
生徒は正しい回答を覚えるだけかもしれない。



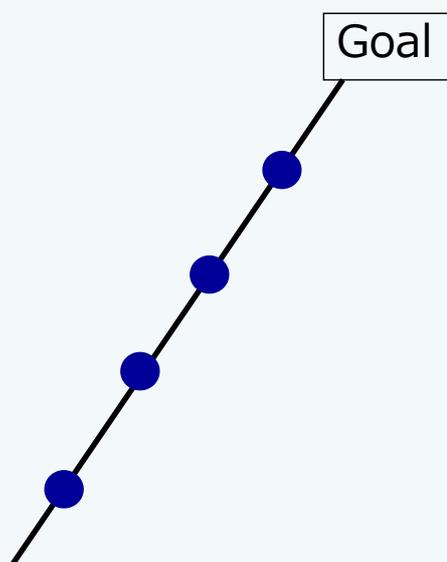
- 本物の、意味のある活動をしたとき、本当に学べる(より実社会に近い文脈で学習する。個々の生徒が異なる認知過程で学習をする。)



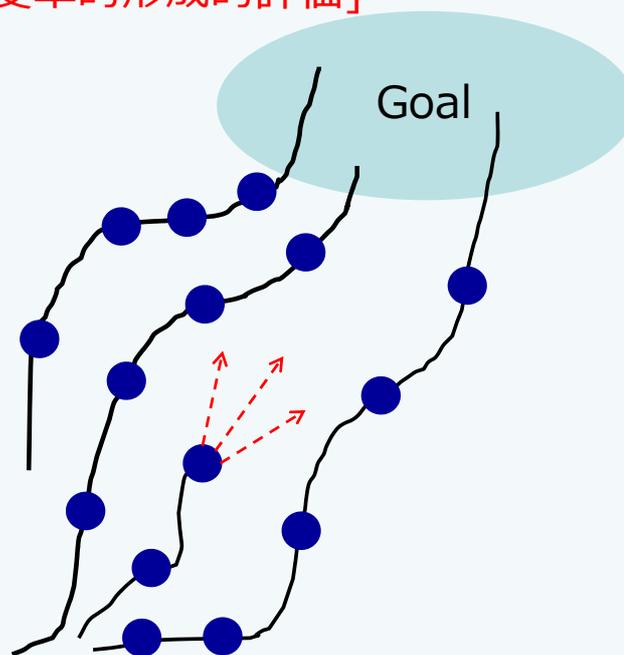
- ・ 「期待する要素」を含む答えを出せたかで評価する
- ・ ひとりひとり固有の過程を評価し、次の指導を考えていく。そのサイクルを回し続ける。:変革的形成的評価

おまけ:従来の形成的評価と変革的形成的評価

[従来の形成的評価]



[変革的形成的評価]



21

2. 本システムの開発思想

生徒にとって

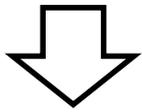
- プログラミングの奥深さを気づかせる。
- 高度なプログラミングを作る動機を与える。
- 自分のプログラムを客観的に評価する。
- プログラミング的思考、コンピューテショナル・シンキングに必要な能力を意識させる。

教師にとって

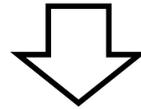
- プログラミング経験の浅い教師でも、生徒の作ったプログラムの内容を把握できる
- 短時間で多くの生徒のプログラムの内容を把握できる。
- 生徒のプログラム能力の変化を把握できる。
(特にクラスルーム用システムにおいて)

本システムの立ち位置

プログラミング教育用言語 プラットフォームの覇権争い?
ドリトル, VISCUIT, Scratch, アルゴロジック, プログラミン,
スモウルビー, Minecraft/Microsoft Block Editor, PEN,
MOONBlock, GLICODE, Tynker...



教材の作成
授業案の作成・実践
教員研修
指導員育成
指導支援体制の整備



システムによる支援

ここ:正しいプログラムの判定や正しいプログラム作成の支援ではなく、プログラミング内容/レベルや進歩の様子への支援

23

トップ画面

NinjaCode Village 術の巻物 プログラム診断 サイトについて

コード忍者の里 for Scratch

術の巻物
プログラムは、いくつかのパターンから出来上がっています。術の巻物(じゅつ)のまきものでも、役に立つパターンを知ることができます。

プログラム診断
君や友達のプログラムが、どのようなプログラムの考え方を使っているか、どのような術(じゅつ)を組み合わせているか自動的に診断しだんします。

診断するプログラムファイルの指定

ファイル名
[ファイルを選択] 選択されていません

何年生ですか
[選んでください]???

プログラムのタイプ
[選んでください]???

自分で作ったプログラム

診断開始

新春の術

<p>装着 まつし選択</p> <p>クリックした帽子をかぶり ます プロジェクトID:105760323</p>	<p>攻撃 単純攻撃</p> <p>星一個で攻撃します。 プロジェクトID:105309398</p>	<p>攻撃 渦巻き弾幕</p> <p>渦巻き状の星で攻撃しま す。 プロジェクトID:105311643</p>	<p>降る ひと粒の雪</p> <p>一粒の雪が降ります。 プロジェクトID:105203960</p>	<p>降る 雪乱舞</p> <p>いろいろな種類の雪が降 ってきます。 プロジェクトID:105202201</p>
---	---	--	--	--

教師・指導者の方へ | 開発思想

24

システムの2つの診断機能



診断結果

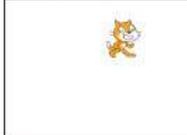
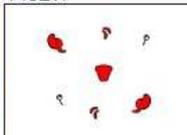
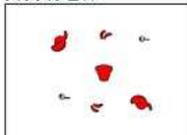
プログラマーの技					プログラム内の術
#	1	2	3	4	
分岐	★				MD.1.2: 壁反射 VA.1.1: ポイント RN.1.1: サイコロ 術の表示
ループ	★	★		★	
モジュール	★				
モデル/モジュール共有					
データ利用	★				
発動トリガー	★				
連携・同期					
ユーザインタフェース					

本システムの中核は以下に示す機能サンプルプログラム集と二種類の診断機能である。

本システムは、スクリプトを読み込み、左図の例ではプログラム概念として、分岐はレベル1, ループはレベル1, 2, 4のコーディングをしていることと、プログラムの機能としては、「壁反射」, 「ポイント」, 「サイコロ」のサンプルプログラムと同等の機能を使用していることを診断結果として表示する。

術一覧画面

術リスト

<p>歩く 四角動き</p>  <p>四角に歩きます プロジェクトID:105757239</p>	<p>歩く 壁反射</p>  <p>端までいくと、方向を変えます プロジェクトID:105754746</p>	<p>歩く ワープ歩きA</p>  <p>端までいくと、逆の端に移動します プロジェクトID:105757564</p>	<p>歩く ワープ歩きB</p>  <p>端までいくと、逆の端に移動します プロジェクトID:105758468</p>	<p>装着 ぼうし選択</p>  <p>クリックした帽子をかぶります プロジェクトID:105760323</p>
<p>装着 カタログ選択</p>  <p>カタログ上で帽子を選択できます。 プロジェクトID:105760537</p>	<p>装着 単体合体</p>  <p>合体してカニになります。 プロジェクトID:105761147</p>	<p>装着 クルクル合体</p>  <p>部品が回転しながら合体します プロジェクトID:105761373</p>	<p>変化 ブラックホール</p>  <p>宇宙船が消滅していきます プロジェクトID:105753748</p>	<p>ダンス&ミュージック ダンス</p>  <p>音楽に合わせてダンスします。 プロジェクトID:105740267</p>
<p>ダンス&ミュージック ペアダンス</p>  <p>二人が順番にダンスします。 プロジェクトID:105740749</p>	<p>ダンス&ミュージック ダンス舞台</p>  <p>ダンスステージの装飾が変わります。 プロジェクトID:105743192</p>	<p>ダンス&ミュージック ピアノ</p>  <p>けんぼんでピアノを演奏できます。 プロジェクトID:105743510</p>	<p>ダンス&ミュージック エレクトーン</p>  <p>演奏の音色を変えることができます。 プロジェクトID:105743962</p>	<p>感知 犬の散歩</p>  <p>犬がついていきます。 プロジェクトID:105736417</p>

3.2種類の診断機能

左側

[プログラム概念評価]

分岐、ループ、データ利用等のプログラミング概念の使用レベルについて評価



MITのプログラミング場面に特化したコンピューテシヨナル・シンキングの評価フレームワーク

目的: プログラミングの能力の育成と評価

右側

[プログラミング機能分析]

プログラムで使用している機能について分析



Wingのコンピューテシヨナル・シンキングをベースとしている

目的: プログラミングを通じたコンピューテシヨナル・シンキングの能力の育成

二つのレベルのコンピューテシヨナル・シンキング

27

3.1 左側 [プログラム概念評価] (プログラマーの技)

プログラマーの技				
#	1	2	3	4
分岐	*	*	*	*
ループ	*	*	*	
モジュール	*			*
モデル/モジュール共有				
データ利用	*	*		
発動トリガー				
連携・同期		*	*	
ユーザインタフェース		*		

- ループ、分岐、モジュール、データ等、通常のプログラミング教育でも対象としている内容
- 生徒がプログラミングを行う中で、自らこれらの概念があることを学び、その意味を考えて、プログラムをより良いものにしていく
- 教師はこれらの概念を生徒がプログラムの中で使用しているか把握できる



MITの評価フレームワーク等、ここ数年、Scratchを対象に自動評価の研究が始まっている。

28

プログラミング場面の評価フレームワーク (Brennen, K..(2012))

概念 Computational Thinking Concept	実践 Computational Thinking Practice	考え方・見方 Computational Thinking Perspective
<ul style="list-style-type: none"> ・ 順次 ・ ループ ・ イベント ・ 並列処理 ・ 分岐 ・ 演算子 ・ データ 	<ul style="list-style-type: none"> ・ 反復型開発 (Iterative and Incremental Development) ・ テストとデバック ・ 再利用とリミックス 	<ul style="list-style-type: none"> ・ 表現 ・ つながり ・ 質問すること



「プログラム概念」と本システムでは呼んでいる

プログラム概念評価の評価基準の作成の流れ



項目	Year 1-5					Year 7-8		Year 10-12	
	1	2	3	4	5	7	8	10	12
分岐									
演算子									
ループ									
モジュール									
データ利用									



項目	Scratchの言語使用を定義し4レベルに分類			
	1	2	3	4
分岐	if	if - else	論理演算子	if (else)の入力
ループ	単純ループ	ループ回数指定	終了条件付きループ	ループの入れ子
モジュール	複数のスクリプト (アタチメント)	グローバル変数	変数のあるブロック	グローバル変数 (グローバル変数)
データ利用	1つのスクリプトで複数のデータの共有	異なるスクリプト間でデータの共有	異なるスクリプト間でデータの共有	グローバル変数の共有
演算・トリガー	数値利用	数値利用	リスト/変数利用	リスト/変数利用
演算・トリガー	特定のキーにより数値	特定のキーにより数値	特定のキーにより数値	特定のキーにより数値
演算・トリガー	特定のキーにより数値	特定のキーにより数値	特定のキーにより数値	特定のキーにより数値
演算・トリガー	特定のキーにより数値	特定のキーにより数値	特定のキーにより数値	特定のキーにより数値
演算・トリガー	特定のキーにより数値	特定のキーにより数値	特定のキーにより数値	特定のキーにより数値
演算・トリガー	特定のキーにより数値	特定のキーにより数値	特定のキーにより数値	特定のキーにより数値

英国Computing Progression Pathways (G1-10:約150項目)

プログラミングに関する28項目を抽出し、MITのプログラム概念で整理 (8レベル)

Scratchの言語使用をもとに評価内容を定義し4レベルに分類

UK computing pathwayをプログラム概念で再構成

学年	Year 1-6				Year 7-9		Year 10-11
発達段階	2	3	4	5	6	7	8
分岐	ループやif文などの分岐を使った簡単なアルゴリズムを設計。	反復やif-else文のような二分岐を使う解決方法（アルゴリズム）を設計	if文とif-then-else文の違いを理解し、それらを適切に使用		入れ子(ネスト)になった分岐文を使用		
演算子	ステートメントの中で算術演算子を使用	if-then-elseを含む分岐の流れ、をプログラムの中で使用	変数と比較演算子を、終了判断を制御するために使用	論理型等の演算子と数式をプログラムでの制御で利用	(ビット)反転の演算子を理解し使用		
ループ	プログラムの中でループを使用	"until"等の後判定ループを使用		イテレーターシジョン繰り返しの処理であることを理解		"前判定と後判定のループの違いを理解し、利用	"While"ループと"For"ループの違いを理解
モジュール			問題を分割し、個々の部分に対しての個別の解決方法を作成		引数を持つ関数の必要性を認識し、独自の関数を作成	引数の受け渡しについて理解して利用	同じ問題のより小さな部分の解決方法に依存する(再帰)の利用
モデル(抽象化)			プロシージャは、下位の解決方法の詳細を隠すために使用できることを理解	状況における類似性と差異を識別できて、それらを問題解決に利用(パターン認識)。	いくつかの問題が同様の特徴を共有し、同じアルゴリズムを使用することを認識	問題解決の一般化において、情報をどこで取り除くことができるか認識。	サブルーチンとして、どこでも再利用可能なモジュールプログラムをデザイン、作成
データ利用		変数を宣言したり割り当てる。		適切なデータの型を選択	一次元配列変数構造を使用し操作	変数の有効なスコープを確認	二次元配列構造を理解し利用

31

本システムの診断基準(プログラミング概念)

	1	2	3	4
分岐	If	If - else	論理演算子	If (- else)の入れ子
ループ	無限ループ	ループ回数指定	終了条件付きループ	ループの入れ子
モジュール	複数のスプライト [オブジェクト]	ブロック[サブルーチン]	引数のあるブロック	クローン[オブジェクトのコピー]
モデル/モジュール共有	1つのスクリプトで複数のブロック	異なるスクリプトでブロックの利用	異なるスプライトで同一ブロック利用	ブロックの再帰的呼び出し
データ利用	変数利用	異なるスプライトで変数共有	リスト型変数利用	クラウド変数利用
発動・トリガー	特定キーにより起動	マウス操作による起動	背景変化による起動	タイマー等による起動
連携・同期	背景変化を介した複数スクリプトでの連携	同一スプライトでのメッセージの使用	他のスプライトへのメッセージでの連携	メッセージ後のWaitでの連携
ユーザインタフェース	文字入力の使用	キーの検出	マウスクリックの検出	マウスの座標位置の利用

32

3.2 左側 [プログラム機能分析] (プログラム内の術)

プログラム内の術

ST.1.5: ブラックホール
FA.1.2: 雷
ST.1.1: 加速
ST.1.3: ボール投げ
AT.1.3: 大きな星で攻撃
CH.1.2: 変化
RN.1.1: サイコロ

術の表示



- プログラミングや問題解決において小さな機能や部分に分解して考えることを生徒が意識し実施する。
- プログラムは小さな抽象的な機能が集まってできることを生徒が意識し、プログラミングする。
- 教師は、使用している機能やその数から、生徒の作成しているプログラムの内容や複雑を判断する。

機能分析は新しい試み

33

本システムでの コンピューティショナル・シンキングの考え方

プログラミング的思考の具体的な能力は、コンピューティショナル・シンキングの中にあると考え、本システムの目標として以下の具体的なキー概念を選んだ。

- **抽象化 (Abstraction)** : 問題解決では、問題を解きやすくするため、問題や事象の適切な側面・性質だけを取り出し、他の部分を捨てること。
- **モデル化 (Modeling)** : 抽象化の結果であり、問題解決では問題や事象を簡潔に理論的に表現すること。
- **デコンポジション (Decomposition)** : 大きなまたは複雑な問題や事象を理解または解決可能な細部に分割すること
- **アルゴリズム (Algorithms)** : 問題を解決するための明確な手順で、同様の問題に共通して利用できるものである。

34

機能サンプルプログラム集と自動機能分析

- 生徒がプログラムを作る場合に必要な個々の機能を含んだ小さなプログラムをサンプル集として提供することで、プログラム作成作業を手助けする
- プログラム分析機能により、生徒はプログラムがいろいろな機能から構成されることを気づかせ、子どもは作りたいプログラムに必要な機能をサンプル集の中から探して組み込むことで、大きく複雑なプログラムも容易に作成できる
- 子どもがお手本となるプログラムを見つけた時、それがどんな機能から作られているか知ることができる

35

術(機能)リスト例と検出パターン

サンプルタイトル	機能	Scratchプログラムの検出パターン
サイコロ	乱数の使用	"['randomFrom:to:']"
壁反射	画面枠での反射	"['doUntil' 'doForever' 'doRepeat' ...['bounceOffEdge']"
ポイント	得点の計算	"['doIf' 'doIfElse', [' ...['changeVar:by:', ... , 1]]"
迷路	方向キーによる左右の移動	"['whenKeyPressed', 'left arrow']... ['changeXposBy:', - & "['whenKeyPressed', 'right arrow'] ... ['changeXposBy:', "
ピンポン	特定の色に触れた場合の反射	"['doIf' 'doIfElse' ['touchingColor:' ... ['heading: ...]"

36

4. 実装方法と検討項目

4.1 Scratchの対象バージョン

- 今後学習履歴をもとにした指導情報等の作成のため、ビッグデータ処理関係のライブラリーが充実していることと、Scratch関連のツールがあることからPythonで実装。
Python3.4を使用
- Scratch自体はVer 1.4と2.0があるが、今回はWeb上のコミュニティを利用することを想定しているため、現状2.0が対象。2.0自体のファイル形式は公開されていて、プロジェクトは使用している画像などをパッケージ化したZip形式のファイル。Zip内にプログラム自体はJSON形式のファイルで存在。(1.4 -> 2.0のコンバータはPython2.x系で利用可能。現状使用できない。Scratchサイトに1.4をアップロードして2.0に変換すれば、診断可能)

37

4.2 評価の得点化と評価基準の公表

- 診断結果はe-learningで使用されるバッジシステムと同様に学習者の動機づけになると考えられる。ただしバッジシステムは、時には学習よりバッジの収集自体が目的化される恐れがある。
Dr. Scratchのように評価結果を得点化して表示することは、本システムでは行わなかった。
- Dr.Scrachでは、プログラミング概念の評価基準内容をWebで公開。本システムでは生徒の目につくところには置いていない。評価基準を提示すれば、単純にバッジを獲得できる。提示しないことで、バッジの意味を考えさせる。

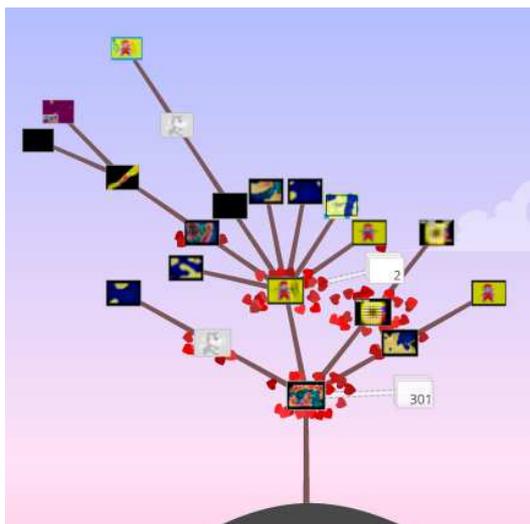
38

4.3 プログラム書法の評価

- Scratchの先行研究/システムではプログラム書法(変数名の初期値からの変更、無意味なブロック・変数)も評価.
- 本システムでも診断過程において、プログラム書法に関するチェックは行っている(Dr.Scratchより細かい)。但し、生徒に自由なプログラミングを楽しんでもらうため、プログラム書法に関する情報は表示していない。

39

4.4 リミックス、他人のプログラム評価の問題



リミックスツリー
他のプログラムをもとに
した開発の関係図

- リミックスで作ると、いきなり高度なプログラムが出来上がってしまう。

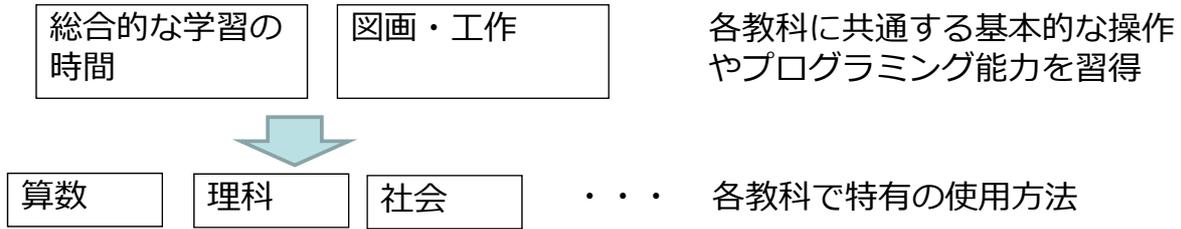


考え方・見方(Computational Thinking Perspective)のつながり(Connecting)から推奨協働作業だけでなく、他人の為に作るなど含む

40

5. 授業形態/利用形態

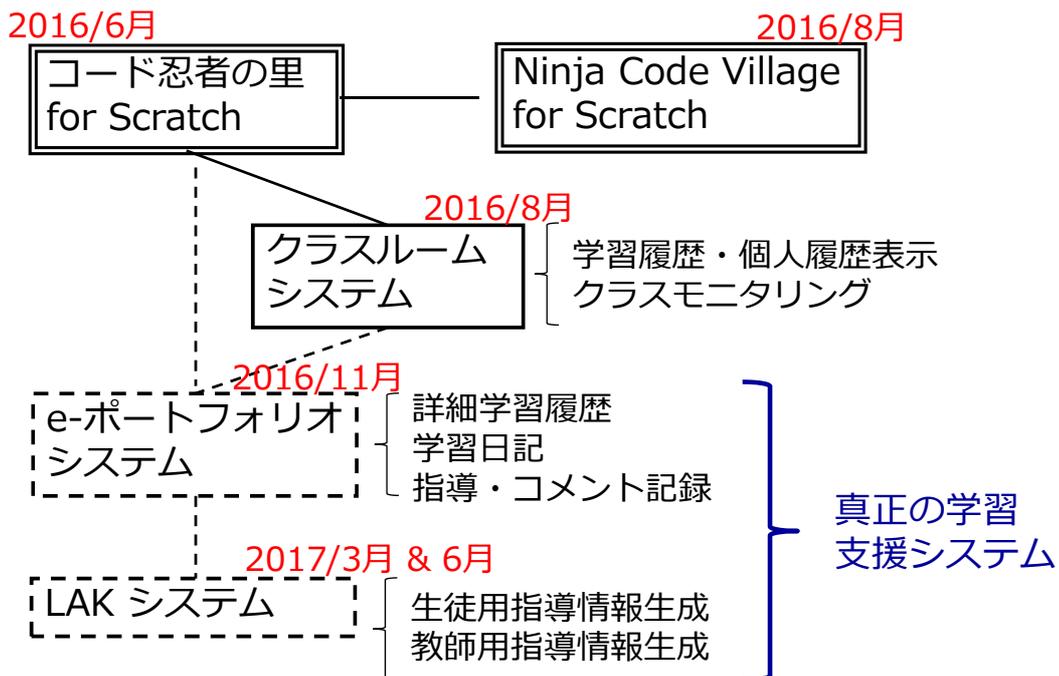
- プログラミング教育を実施する新しい教科を設けず，既存の教科内で学習することが提言された



共通する基本的な操作やプログラミング能力を習得はメイキング場面で使用することが有効であると考えられる。メイキングの考え方から自分の好きなものを作っていくことは，全く新しい授業の形態ではなく，すでに小学校の図画工作科で実践されてきて，図画工作科の学習指導要領の領域「A表現」には“「表したいことを絵や立体，工作に表す」は，およそのテーマや目的をもとに作品をつくらうとすることから始まる。”とある。

41

6. 開発計画



42

コード忍者の里 for Scratch (診断結果モニタ外アップ)

今までの診断結果(概要)

最終日時	名前	総診断	総自分	最大*の数	最大術の数	最大スプライト数	最大変数の数
2016年8月5日 13:49	あおちゃん	0	0	0	0	0	0
2016年8月5日 13:49	ごう	10	3	8	4	6	1
2016年8月5日 13:49	まこと	0	0	0	0	0	0
2016年8月5日 13:49	けんちゃん	0	0	0	0	0	0
2016年8月5日 13:49	まおい	0	0	0	0	0	0
2016年8月5日 13:49	みさお	0	0	0	0	0	0

© 2016 Go Ota (Powered by the concept model of Multi Assist)

43

今までの診断結果(概要)

日時	名前	自分	タイプ	*の数	術の数	スプライト数	スクリプト数
2016年8月17日 12:43	ユークリッドの互除法.stb2	0				1	1
2016年8月17日 12:44	VA_1_1-Point(ポイント).stb2	0				3	6
2016年8月17日 14:45	IN_1_1-Talk(会話).stb2	0				1	1
2016年8月17日 16:32	VA_1_1-Point(ポイント).stb2	0				3	6
2016年8月17日 16:35	VA_1_1-Point(ポイント).stb2	0				3	6
2016年8月17日 16:35	VA_1_1-Point(ポイント).stb2	0				3	6
2016年8月18日 5:13		0				3	6
2016年8月18日 6:12	VA_1_1-Point(ポイント).stb2	0		0	4	3	6
2016年8月18日 6:20	VA_1_1-Point(ポイント).stb2	0		8	4	3	6
2016年8月18日 11:21	VA_1_1-Point(ポイント).stb2	0		8	4	3	6

44

コード忍者の里

🔍 検索

黒板を超えて

🔍 検索