

# プログラミング能力の 発達段階と要因に関する定量的分析

小学校でプログラミング教育が始まるけど、  
何年生に何を教えたら  
いいの？



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

このスライドはネットに  
アップしてあります。  
最後にURL示します。

1

## 背景と目的

約900+100のScratchプログラムの内容を分析しまし  
た。

スクリプト



カスタムブロック



ブロック(コマンド/ステートメント相当)を組み合わせて  
プログラミング

2

## はじめに：背景

文部科学省：

“発達の段階に即して、「プログラミング的思考」を育成すること”



プログラミングの発達段階について論議されることは無い。  
何年生でどのようなプログラムが作れるのかも明確になっていない。

○いろいろ実践が行われているが、そこで作成するプログラムが対象の学年の子供にとって適切なものなの？

○「プログラミングは楽しかった」「プログラミングを理解できた」など定性的な評価が多いのでは？

○小中高校でプログラミング教育の系統性が曖昧になり、現在の情報教育のワープロや表計算のように各学校段階で同じ学習内容を繰り返すんじゃないの？ 高校でもScratchのネコ歩き。

プログラミングをちゃんと分析して評価する。

3

## 先行研究の概要

従来のプログラムの評価、分析：

主に高等教育で、正しいプログラムや正しい操作の判定。  
(実行結果の判断や、正答プログラムとの比較等)



Scratch等のビジュアル言語を対象に、  
自由課題場面などの多様なプログラムの質を評価。

- ・使用ブロックの有無や使用ブロックの数
  - ・ブロックのカテゴリー
  - ・プログラミング環境のコミュニティの活動
- などで評価基準で分析し
- ・プログラミング能力の発達段階
  - ・プログラミング能力の獲得の要因

現状：

- ・評価基準や分析方法を試行錯誤で構築している状況
- ・必ずしも明確な発達段階や獲得要因が明らかになっていない

4

# 分析調査の目的

1. 子供のプログラミング能力の発達段階を明らかにする。  
何年生でどんなプログラムが作れるの  
実際のプログラムから、教材から  
おまけ: 子供と大人のプログラムはどこが違うの
2. プログラミング能力の獲得に影響する要因は?  
先行研究: リミックス(他のプログラムをもとに作成)  
開発期間、作成数  
コミュニティとのかかわり
3. 適切な評価基準は何? 分析方法は?  
新しい2種類の基準  
コンピューティショナル・シンキング概念(CTC)評価項目  
機能分析

5

## 分析・評価基準

### 分析ツール:コード忍者の里 for Scratch クラスルームシステム

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

本来、子供が会話型で、プログラムの診断を行う。

Excelシートに記載したプログラム情報をもとに大量一括分析も可能

6

# 本分析調査の評価基準の考え方

## 1. プログラムの基本的特徴

ブロック数(通常のプログラムのステートメント数)  
変数の数等

## 2. ブロック種別:CTC評価項目

MITのScratch用プログラミング評価フレームワーク(分岐・ループ等)に分類。

### コンピューターシヨナル・シンキング概念(CTC)

問題点: 基本的なブロックは小4~5で習得できるみたい  
-> 本分析では発達段階に基づく評価基準を作成

## 3. 機能分析

子供達は機能を中心に作成しているようである。

-> 本調査では63種類の機能を想定し、その機能を検出

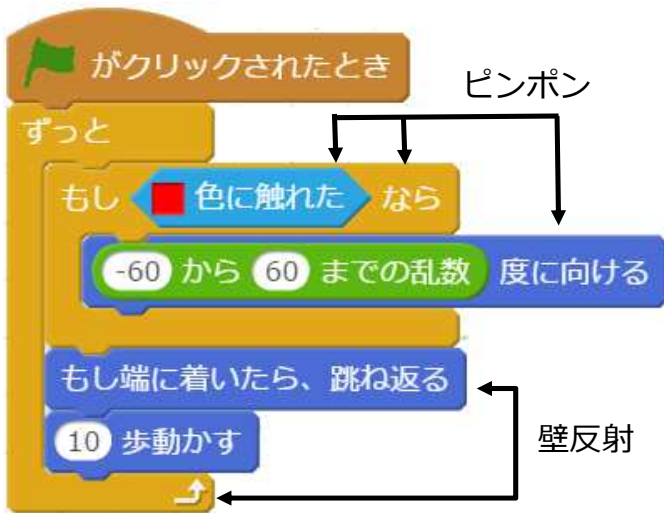
7

## 評価基準: CTC評価項目(新規作成)

	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]	カスタムブロック [サブルーチン]	引数のあるカスタムブロック	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カスタムブロック	異なるスクリプトでカスタムブロックの利用	異なるスプライトで同一カスタムブロック利用	カスタム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	同ー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

8

# 評価基準：機能分析(新規作成)



子供同士のプログラミング場面での教えあいでは機能で会話する。

子供の初心者がよく使用する機能を63種類作成し、そのパターンを検出する

先行研究の多くは、個々のブロックの有無のみ分析

使用例	機能	Scratch内部プログラムの検出パターン
壁反射	画面枠での反射	"['doUntil'   'doForever'   'doRepeat' ...['bounceOffEdge'"
ピンポン	特定の色に触れた場合の反射	"['doIf'   'doIfElse' ['touchingColor:' ... ['heading: ... ]"

## 分析1: 教材の分析: 教員実践者の経験知



これらの教材の違いが判りますか?

Scratchの教育実践者らは、経験知識して、何歳/何年生ぐらいの子供はどんなプログラムが出来るか知っていて、教材に反映されているだろう

# 分析対象1: 教材:教育実践者の経験値

略称	タイトル	種別	概要	プログラム数	対象学年
Why	Why! プログラミング (No.1~.10 + No.11~14)	TV Web	11回のテレビ番組で、楽しみながらScratchとプログラムの考え方を学習。視聴者がプログラムを投稿するWebサイトあり。	10	小中学生 [技術 小5~6・中]
Koka	Kokaプログラミング入門(子供の科学)	雑誌 Web	最終的にScratchでゲームを作ること为目标に、主にゲームを題材にしたプログラム。	27	雑誌の中心読者層は小5~6
Waku	小学生からはじめるわくわくプログラミング2	書籍	Scratchの全くの初心者を対象にして、各教科に対応した内容のプログラム。	7	小学生 初心者
Dojo	Scratch(スクラッチ)でつくる! たのしむ! プログラミング道場	書籍	世界的なプログラミング教室のCoderDojoの国内各地の主催者が実際に使用している初心者向けプログラム。	9	Scratch初心者
京陽	プログラミング学習実践事例集(品川区立京陽小学校)	実践 報告 書	京陽小学校での小1~6の各学年での実践報告。	6	小学生

11

## 教材分析結果 CTC評価項目(1)

	CTC評価項目	Waku Waku	Coder Dojo	Keiyo	Koka	Why
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	○	○	○	○	○
	カスタムブロック(Cブロック)				○	○
	引数のあるCブロック				○	○
	クローン		○		○	○
モデル/ モジュール共有	1つのスクリプトで複数のCブロック					○
	異なるスクリプトでCブロックの利用					○
	異なるスプライトで同一Cブロック利用					○
	ブロックの再帰的呼び出し					○

12

# 教材分析結果 CTC評価項目(2)

	CTC評価項目	Waku Waku	Code rDojo	Keiyo	Koka	Why
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	背景変化を介した連携				○	○
	同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					

「Why!プログラミングの歩き方」が公開されました。いろいろな教材の関係図を作ってみたい

## 評価基準：発達段階の参考情報

### 英国 教科Computing Pathwaysのプログラミング関係

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		イテレーシ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、個々の部分に対する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)			プロセスは、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、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

## 応用: 京陽小学校のカリキュラムを分析する

どのようなプログラム(レベルや学習するCTC評価項目)を作るかで、プログラミング学習の目標がはっきりするのでは。

教科でのプログラミング

- ・ 1～5年生

抑えられたコンピューショナルシンキング概念レベルの課題

=> 創造性や協働の重視しているという意味がわかる

(楽しみながら創造性の意味で自由度の高いプログラム)

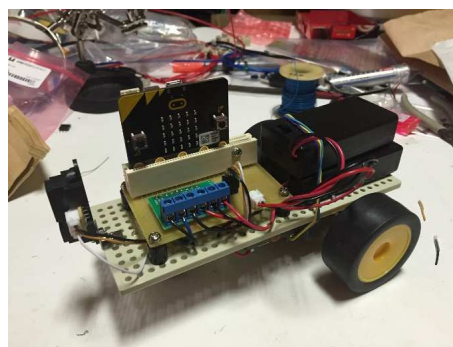
- ・ 6年生では高度なプログラム(リスト配列使って素数を求める)  
プログラミング能力の育成

補足:活用した授業(市民科:総合的な学習の時間)

- ・ テーマはあるがメイキング的な活動
- ・ プログラムの技術レベルは個々の生徒のレベルに合わせて
- ・ グループの協働制作

15

## 分析2: 子供のプログラムの分析



CoderDojo市川真間

16



## 分析対象2: 子供の実際のプログラム

Scratchコミュニティで公開されている小学校4～6年生のプログラムを対象に分析を行った。その中でコミュニティ利用が1年未満で10個以上のプログラムを公開している子供を対象とした。

	人数	総プログラム数	総自作数	総リミックス数	平均プログラム数	平均自作数	平均リミックス数
4年	8	232	137	95	29.0	17.1	11.9
5年	9	281	208	73	31.2	23.1	8.1
6年	11	358	283	75	32.5	25.7	6.8
全体	28	871	628	243	31.1	22.5	8.7

おまけ

専門家	10	95	95			9.5	
-----	----	----	----	--	--	-----	--

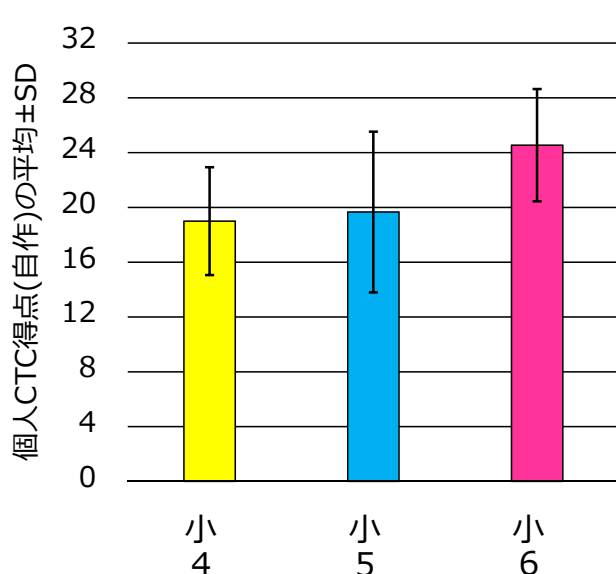
### 制約事項

- ・ Scratchコミュニティに参加している時点でプログラムに興味があると考えられ、一般の子供とは乖離している
- ・ 公開しているプログラムなので、実際に作成した全体ではない
- ・ 3学年分の被験者数が28名と十分でない

17

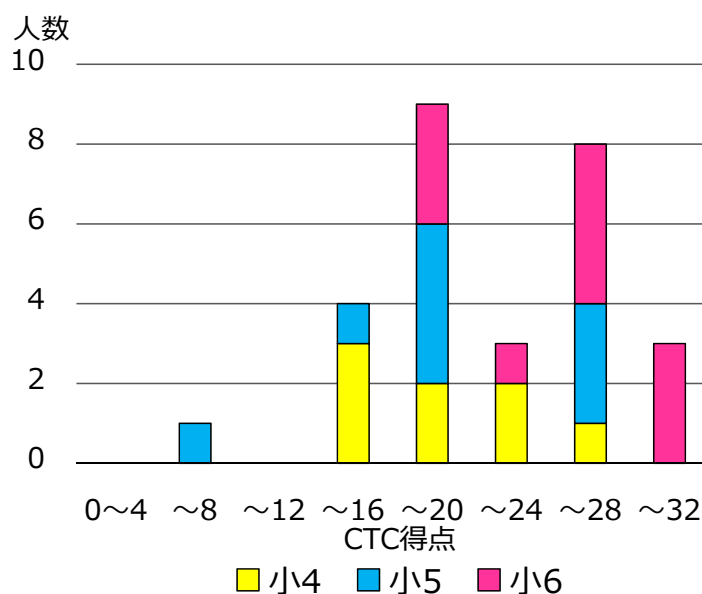
## 発達段階に関する分析結果(1) CTC得点

学年間のCTC評価項目(自作)の獲得状況の比較



学年の効果は有意  
( $F(2,25) = 3.641, p < .001$ )

\*CTC得点: 32個の個々のCTC評価項目を使用していた場合、1点として得点化(Max 32)



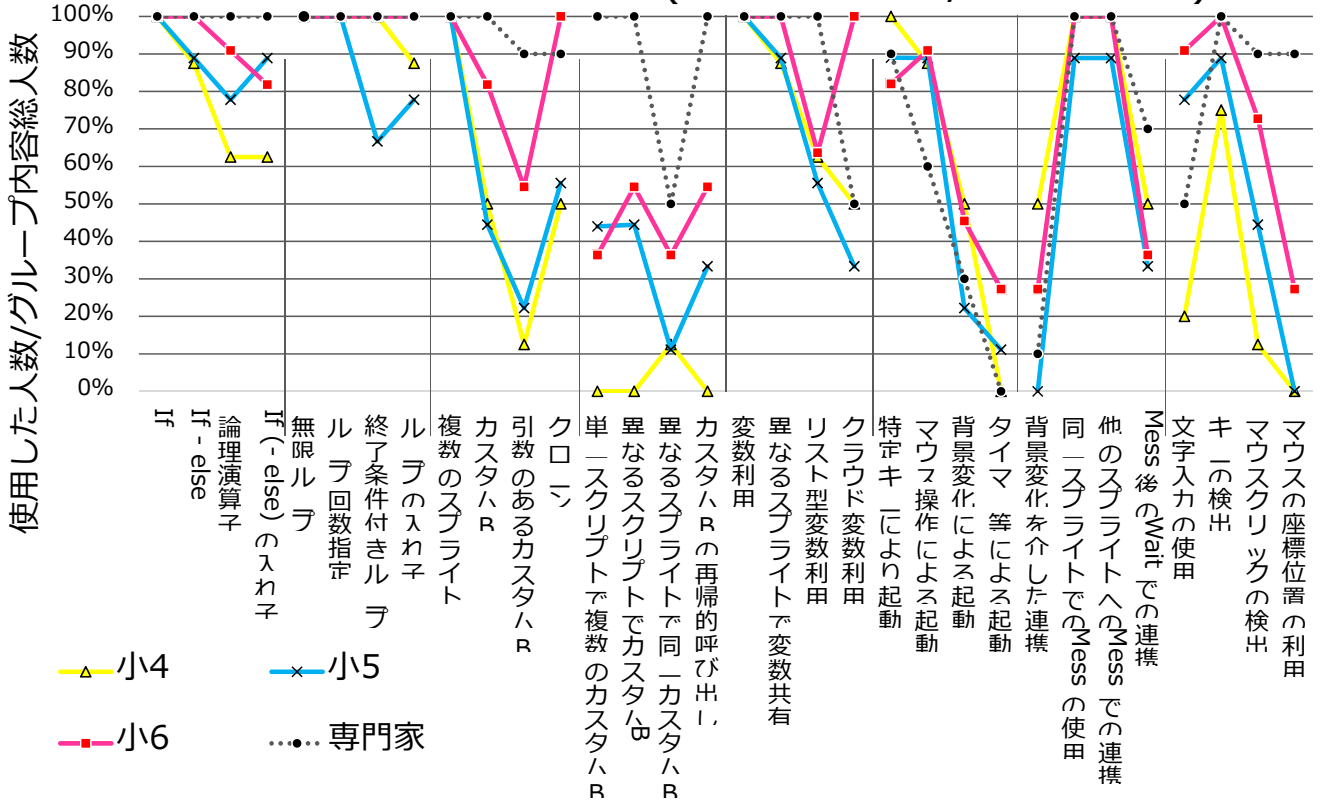
CTC得点(自作)の分布

但し。小学校4年生でも6年生と同様に高いプログラミング能力を獲得することは可能である。

18

# 発達段階に関する分析結果(2) CTC項目別

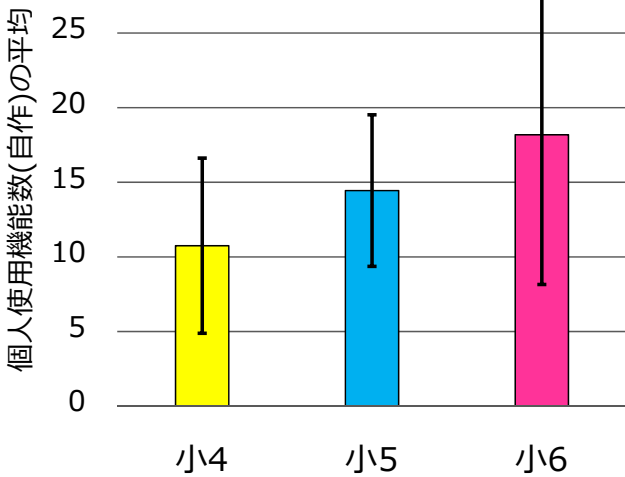
学年ごとのCTC評価項目の使用率(使用した子供/子供の人数)



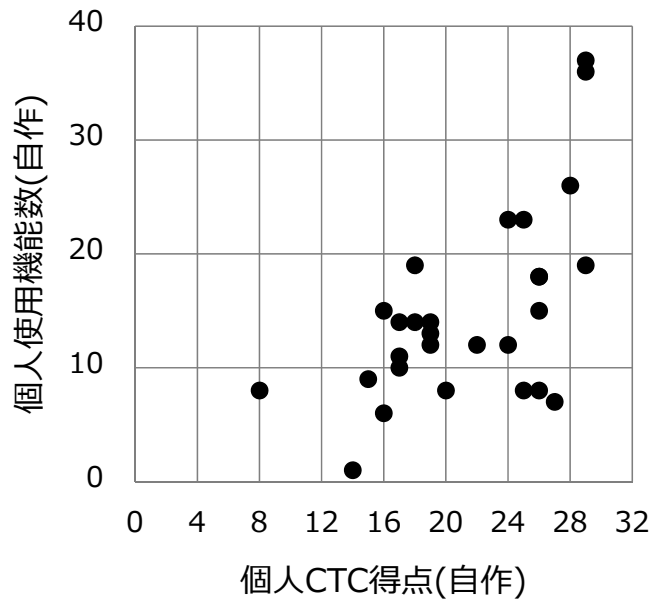
○ブロックの使用有無だけだと4年以上の違いを区別できないかも  
 ○教育実践者との経験値とも的一致しているのでは(Pathways含む)

# 発達段階に関する分析結果(3) 使用機能数

学年間の使用機能数比較



学年の効果 は有意でない  
 $(F(2,25) = 1.999, p = 0.156)$



個人CTC得点(自作)と個人使用機能数(自作)の関係

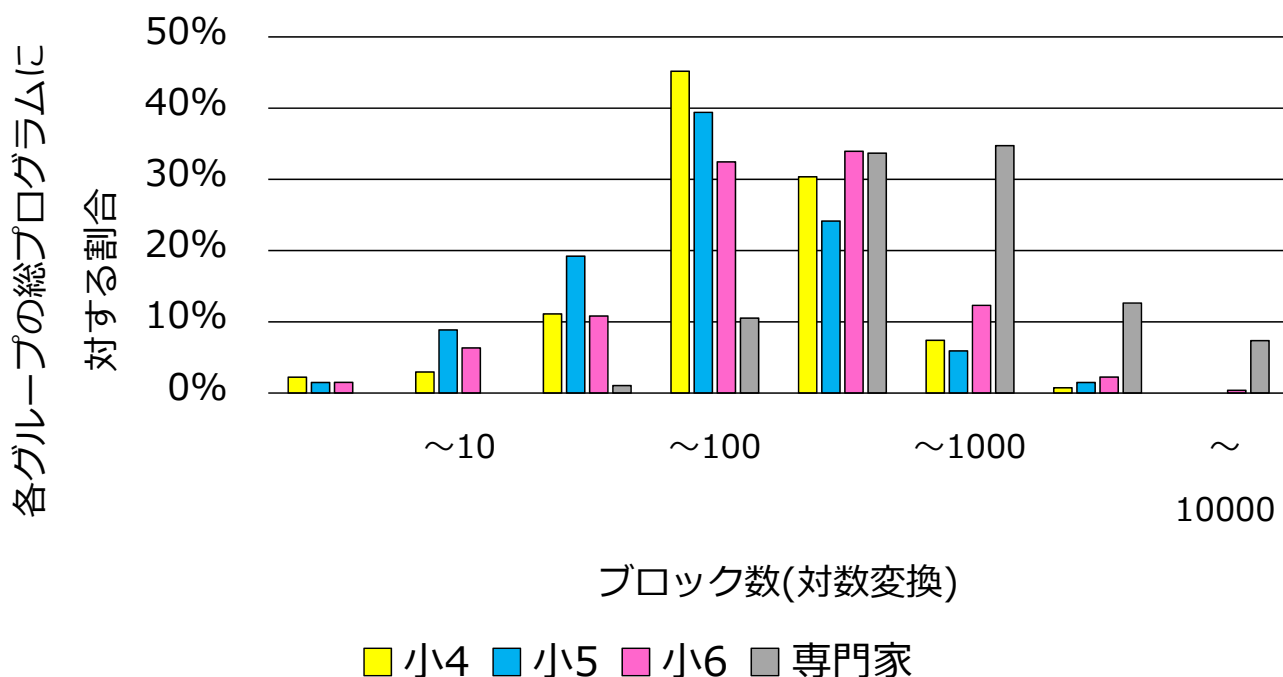
やや相関がみられる( $r = 0.60$ )

○ツールが機能を検出できない ○機能を使わないプログラム(100%Penなど)

# 発達段階に関する分析結果(4) プログラムサイズ

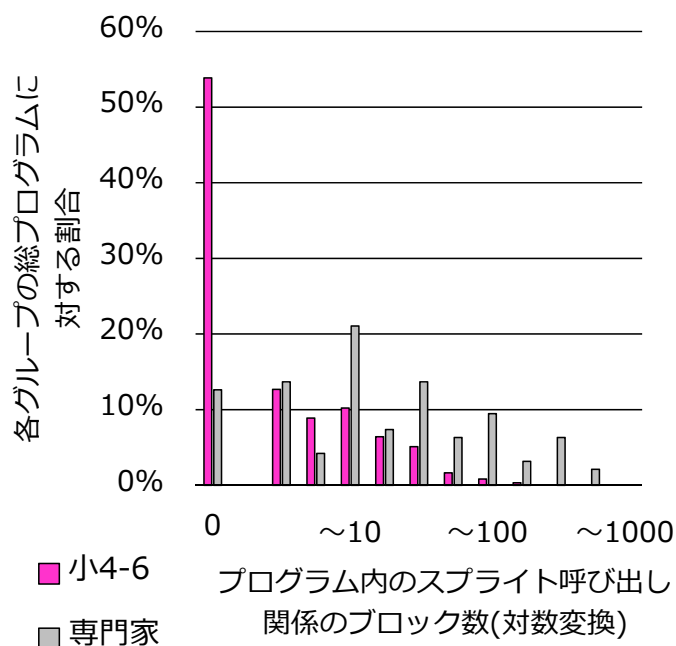
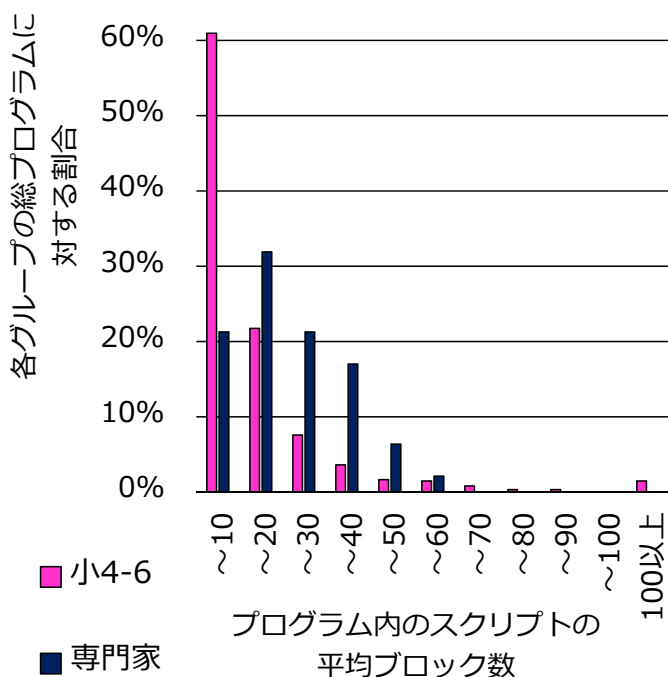
## 学年間のプログラムのブロック数の比較

各学年のプログラムの分布(各学年の総プログラム数に対する割合)



○小5・6で明らかに大きなプログラムが作れるようになる

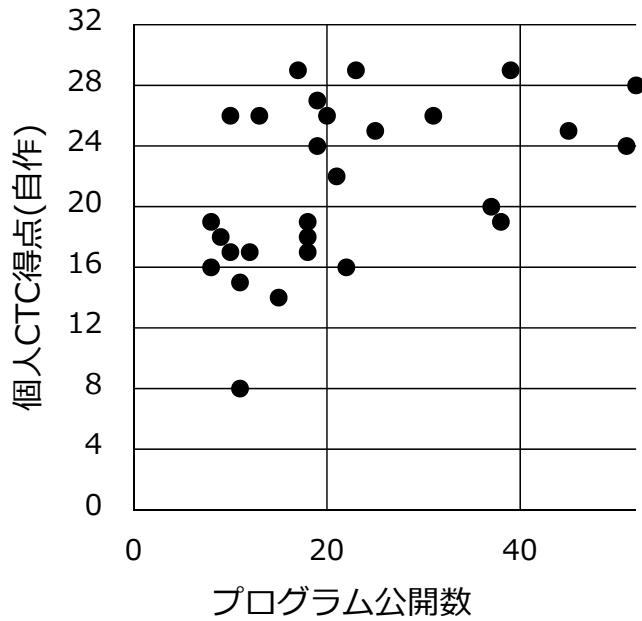
# 発達段階に関する分析結果(おまけ) モジュール



○専門家はスクリプトへのメッセージ送信(他のスプライトを起動する)や、カスタムブロックの呼び出しを多用している。  
 ○専門家の平均のスクリプトサイズがある程度の大きさに収まっている。

## 能力獲得の要因に関する分析結果(1)

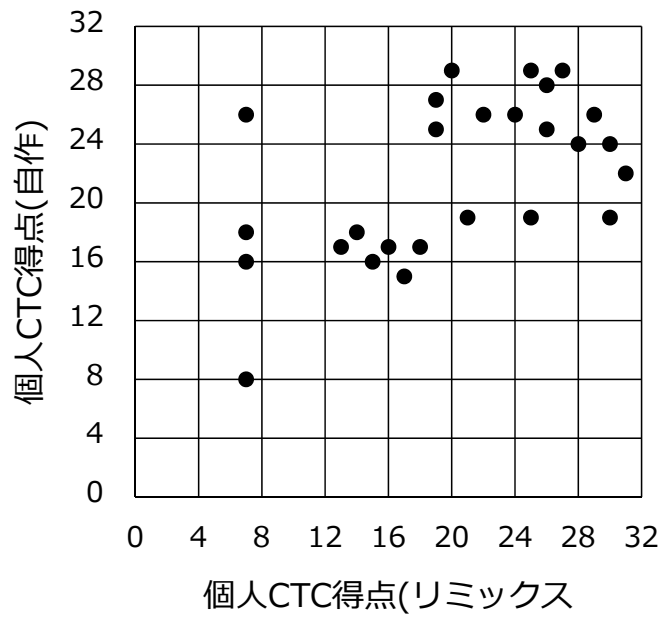
### プログラム作成数



やや相関がみられる( $r = 0.46$ )

- 作成した全部のプログラムではない
- 参加期間とCTC得点の相関は無し

### リミックス



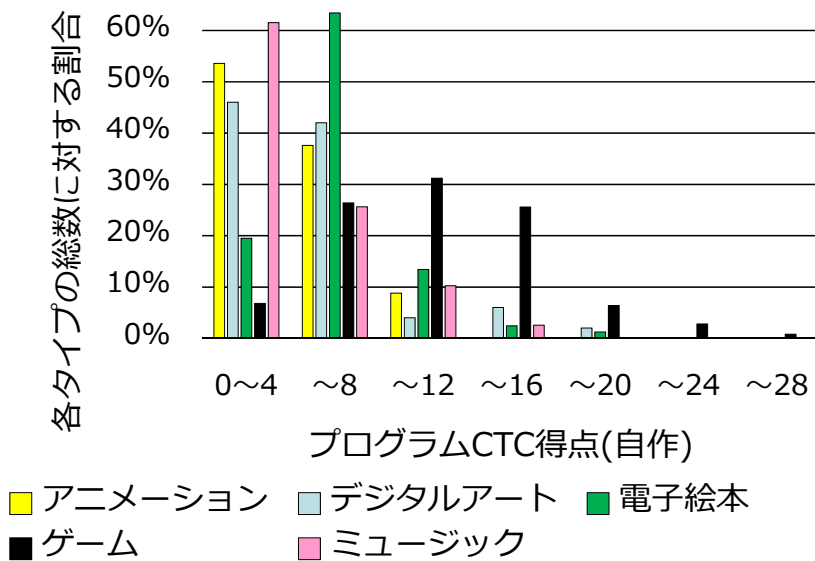
やや相関がみられる( $r = 0.55$ )

- より詳細な分析が必要
- リミックスの内容
- 自作プログラムへの反映

23

## 能力獲得の要因に関する分析結果(2)

### プログラムのタイプ

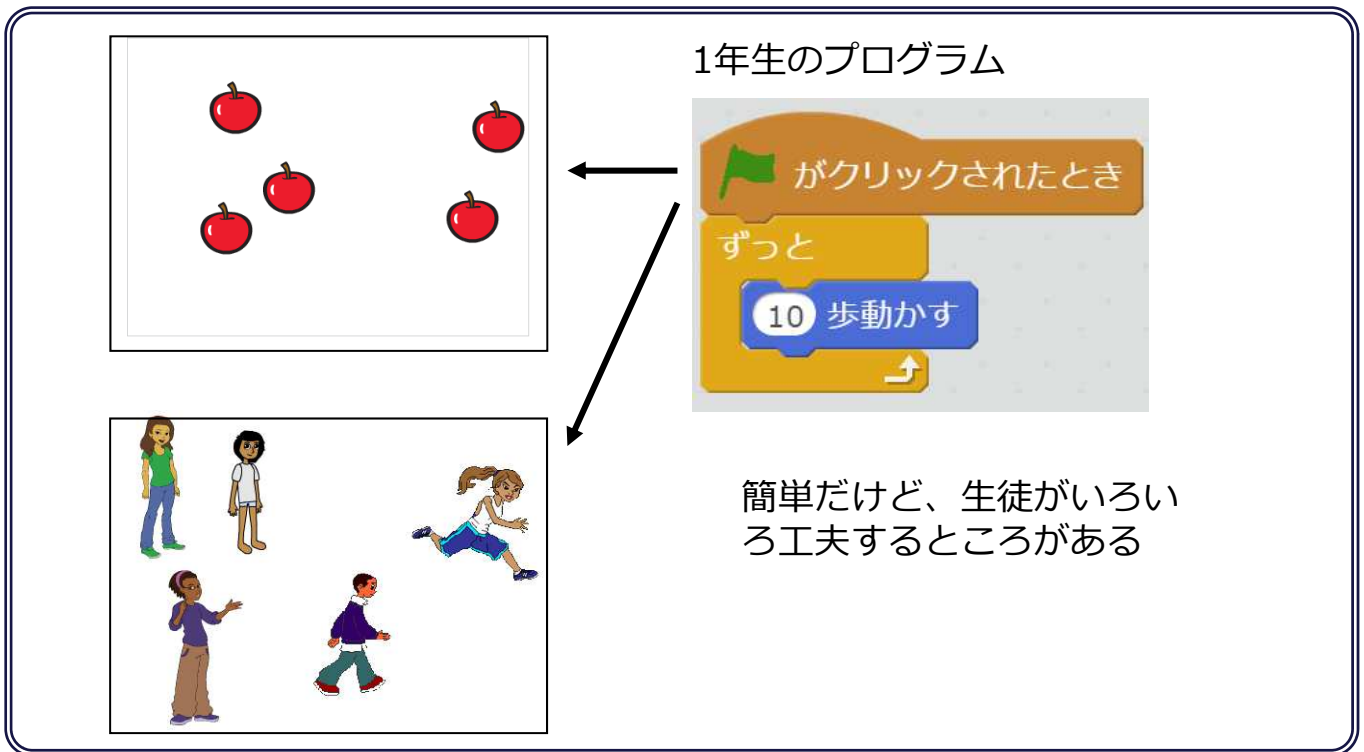


プログラムのタイプ別のCTC得点の分布

○アニメーションを対象にした先行研究では、学習が進んでもあまりプログラムの内容に差がでなかった。  
-> アニメーション自体がそんなに難しいことしないのでは?

○学校での実践において、比較的自由的な課題としてアニメーション的なもの(物語や説明など)を作ることがある。  
この場合、どのようなプログラミング能力が獲得されるか把握しておく必要がある。  
(アニメーション的なものについては、創造性などの育成については適切な課題という一面)

# まとめ



1年生のプログラム

がクリックされたとき

ずっと

10 歩動かす

簡単だけど、生徒がいろいろ工夫するところがある

25

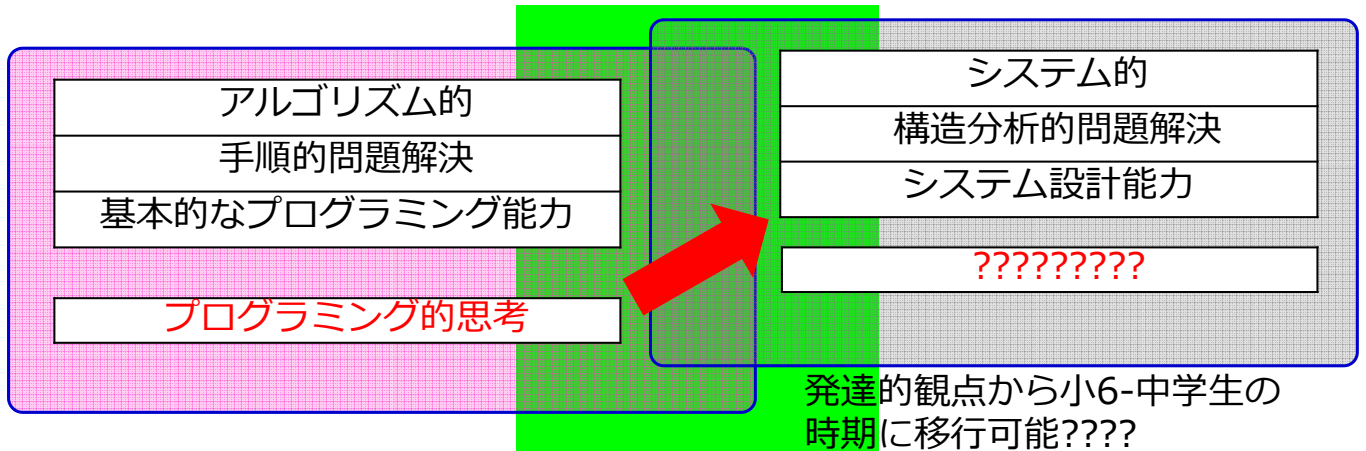
## まとめ

1. 小4でifやループ・変数などの使用が可能。小6で入れ子や論理演算などの使用が可能。(な子供もいる)
2. 新たに作成したCTC評価項目は、プログラミング能力の獲得の違いについて分析可能。機能分析は今後の改良が必要
3. 今後:さらなる分析  
中高を含めたプログラムの分析  
作成した全プログラムの分析
4. 今後: 応用  
教材作成や実践において、もっとプログラムの内容自体を検討 -> プログラミング学習の目標を再考する。
5. 子供と大人のプログラムの違い -> 次ページ

26

# おまけ: 「プログラミング的思考」のモヤモヤ

- ・小6程度で分岐・ループ・変数等の基本的なプログラミング能力の学習が可能ではないか : **アルゴリズム的なプログラム: 戦術的**
- ・専門家になるとプログラムをモジュール化して分割するなど、大きなプログラムを作成 : **システムのなプログラム: 戦略的**

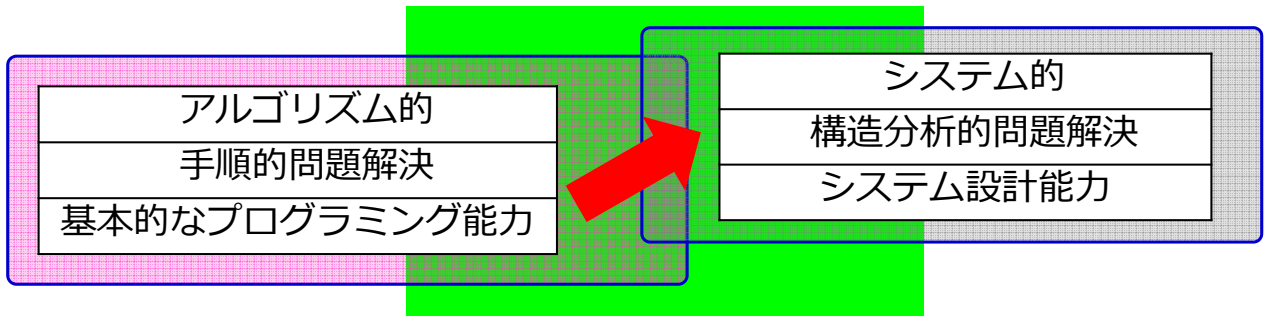


## 問:

- ・「プログラミング的思考」を小中高のすべての段階で育成?
- 次期指導要領: 中学技術・家庭解説「プログラミング的思考等を発揮して解決策を構想する力」

各国の情報科目(小中高を通じて):  
 オーストラリア: システム思考、デザイン思考、コンピューショナル・シンキング、プロジェクトマネジメント  
 英国: 抽象化、デコンポジション、アルゴリズム的思考、評価、一般化

# おまけのおまけ: 私的プログラミング教育の系統性



	小学校	中学校	高等学校
教科	すべての教科	技術・家庭	情報
考え方	プログラミング的思考	???????	???????
私的希望	プログラミング体験 プログラミング嫌いにならない		

海外	パズル->ビジュアル言語	テキスト言語	システム開発
----	--------------	--------	--------

## 本日のスライドのアップロード先

○[http://beyondbb.jp/Materials/SSS2017\\_GoOta\\_0818.pdf](http://beyondbb.jp/Materials/SSS2017_GoOta_0818.pdf)

○Web「高校「情報科」の教材・指導案作ってみました。」  
のトップページからアクセスできます。

高校 情報科 教材
-----------

検索
----

### お願い: 共同研究者/研究協力者の募集

今回の分析に使用したシステム「コード忍者の里」

本来は、プログラムの診断結果を学習記録として残す  
プログラム学習支援用のe-ポートフォリオシステム/ダッシュボード  
です。

主にScratchを使用する自由課題でのプログラミング学習を行う  
中学・高校・大学の先生のご利用お待ちしております。