

Scratchワークショップ型の 「教えない」プログラミング教育

高校でもプログラ
ミング教育が始ま
るけど、どうした
らいいの？

太田 剛:千葉県立袖ヶ浦高等学校/市川南高等学校
南雲 智:千葉県立袖ヶ浦高等学校
森本 得憲:千葉県立袖ヶ浦高等学校
加藤 浩:放送大学



[http://multiassist.sakura.ne.jp/0Presen/
Zenkoujo2018_GoOta_Net.pdf](http://multiassist.sakura.ne.jp/0Presen/Zenkoujo2018_GoOta_Net.pdf)

はじめに: 背景

新学習指導要領:2022年より

高校情報科の現行科目「社会と情報」と「情報の科学」が「情報I」に統合されて、全高校生が2022年度よりプログラミングを学ぶ。



問題:

- ・ 情報科の教師にプログラミング教育できるか。
(免許外3割、他の教科との兼任5割)
情報の先生でも多くは「情報と社会」のみ担当
- ・ 生徒のプログラミング経験のばらつきが多くなる。
- ・ 生徒が授業についてこられるかの不安感。



教師が情報科でプログラミングの授業をしないことの危惧
どうする!!

対応方法

子供向けのプログラミング・ワークショップで現在行われている、指導者が教えることなく、参加した子供が自主的に自由なプログラミングを行う学習形式 =

「教えない」プログラミング教育

↓ やって見た

時数	生徒の活動	教師の活動・教材
フェーズ1: Scratchの入門と自由な作品の開発		
1限	ScratchIDの作成 Scratchの入門教材の実施(必要に応じて)プログラムの開発	ScratchのID作成と基本操作の説明 入門教材とサンプルプログラムの提示
2限	プログラムの開発:	グラフィックの作成方法
3限	・他のプログラムの実行	著作権上の注意点
4限	・友達との相談	他のプログラムの活用方法)
5限	・自分のプログラム開発 ・教師・友達への質問	デバッグ方法
6限	発表会(デモ形式)	

授業の様子

授業の様子

発表会の様子

・生徒は発表会で他の生徒にみせることだけを目指し、資料をみて自分でScratchのプログラミングするだけ。

実施科目	クラス数		学年	人数	小計
情報の科学	普通科	2	3年生	49	80
アルゴリズムとプログラム	普通科	2	3年生	31	
	情報科	1	2年生	41	41
				合計	121

情報科は1年生の3学期にScratchの授業有

実習の進め方

協働学習/協同学習 21世紀型スキル/能力の一つです。

広い意味のグループ学習。



例：陸上部
基本、個人競技
本時はこっちでいきます。



例：サッカー部
チーム競技

ルール

- ・分からなかったら、出来ている人のやっていることを見る。
- ・分からなかったら、出来ている人に相談する。
- ・出来て、余裕のある人は、わからなかった人を助ける。
- ・分かる・分からないに関係なく、他の人がどんなことをしているか見る、話す。

あくまで、**この授業の実習時間の話**ですが。

- ・実習時間中は他の友達との話は自由です。
- ・実習時間中は、席を立て、うろついていいです。
- ・実習時間中に、自分で凄いことやったと思ったら、声出して友達を呼んで見てもらっていいです。

ただし、先生が話す時は着席。

生徒用の資料

コード忍者の里

はじまりの書
- 90分でScratch入門 -

これから、プログラミングを始める人のための巻物です。
Scratch(スクラッチ)を使って初めてのプログラミングを90分程度作成していきます。

内容:

1. スクラッチを稼う前に
2. スクラッチを使ってみよう:初めての修業
3. 次の修業:プログラムを少し追加しよう。
 - 3A. ゲーム・アクション編
 - 3B. ダンス・ミュージック編
 - 3C. デジタル・アート編
4. コード忍者 白幕への修業
5. 付録

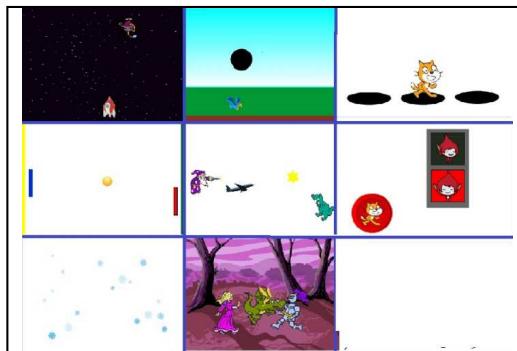
名前

バージョン:番号 (Ver0.90) 1ページ

入門用資料



Scratch 標準チュートリアル



サンプルプログラム(写経用)



<http://multiassist.sakura.ne.jp/0Presen/index.html>

結論というか、これからへの提言

教師がプログラミング授業を実施しないんだったら
➡ 教師が教えない、生徒が自由にプログラミングの方がましかも

- ・事後質問 (5件法 (5. そう思う～ 3. 普通～ 1. そう思わない))

略称	事後質問	全体 (n=112)	
		M	SD
(楽しさ)	従来の授業に比べて、楽しく授業ができた。	3.86	1.32
(主体的)	従来の授業に比べて、自分で学習したと思う。	4.11	0.97
(考える)	従来の授業に比べて、自分で考えたと思う。	4.33	0.93
(集中)	従来の授業に比べて、集中して学習したと思う。	4.14	0.93
(協働)	授業中プログラミングについて友達と相談した。	4.27	1.10
(保持)	自分でプログラミングした内容は、時間がたっても覚えていると思う。	3.75	1.11

- ・プログラムの内容も85%が分岐、95%のループを使用していた。

ワークショップ型プログラミング教育の実現 詳細の結果/考察/課題

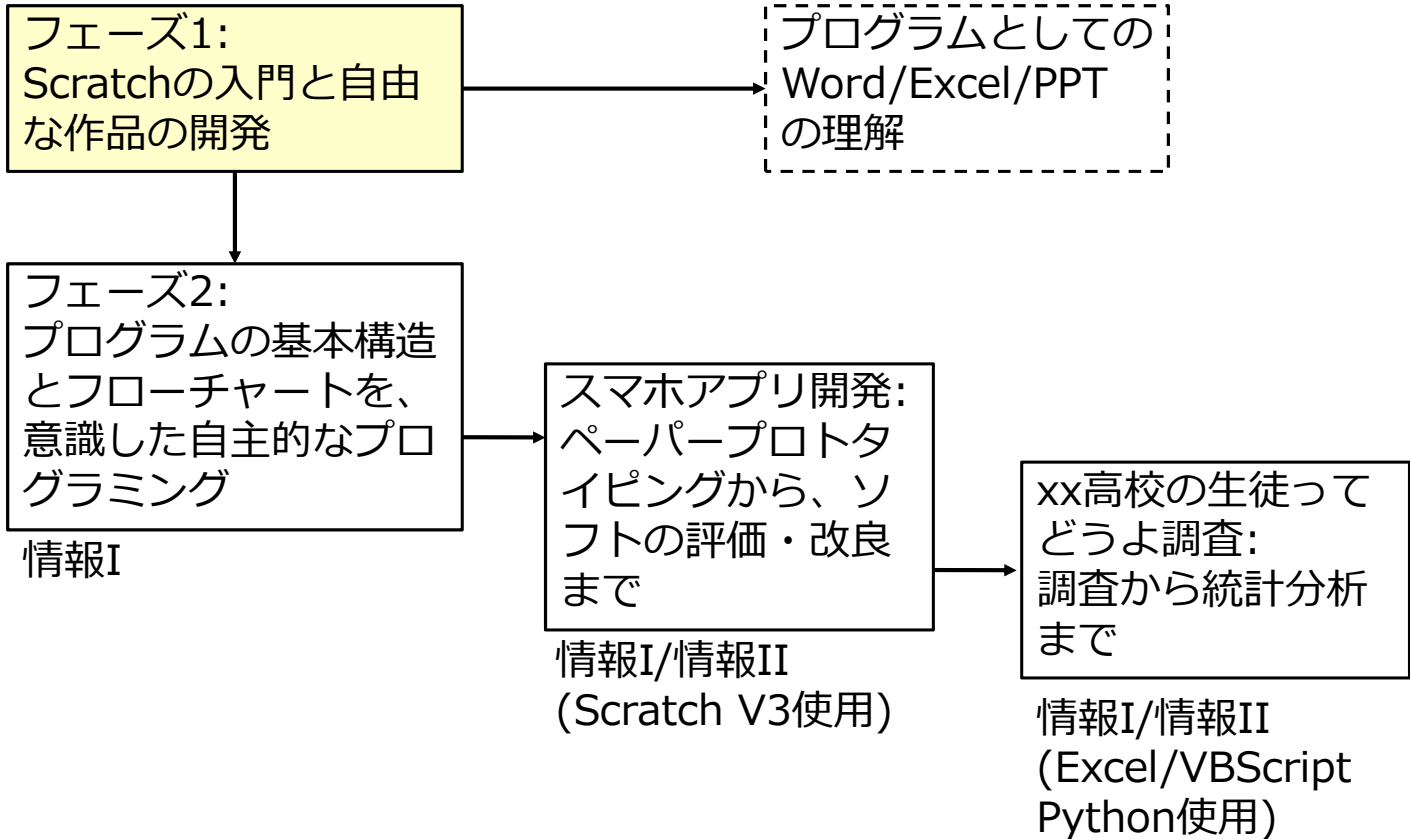
細かく見ると、

- ・生徒がプログラミングをどのように学ぶか
- ・ワークショップ型のプログラミング教育の課題

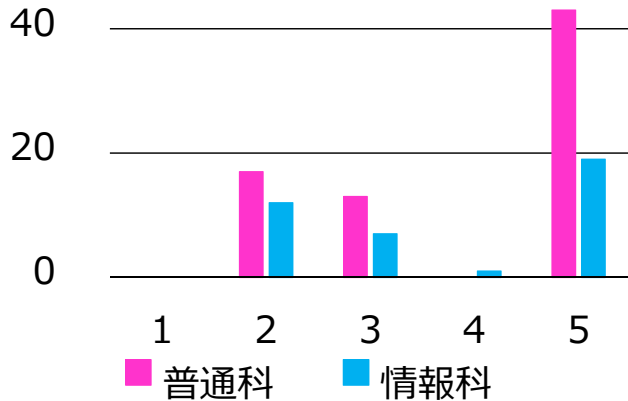
どうしたら、普通の先生がワークショップ型で
プログラミング教育を実施するか

見えてくるかも

あらためて、本実践の位置づけ



(楽しさ)について二極化?



「(楽しさ) 従来の授業に比べて、楽しく授業ができた」の人数分布

単純な統計処理(平均等)は危険かも

もしかしたら、このような自由にプログラミングを作ることが、楽しいと思う生徒と思わない生徒に二極化するかもしれない。

良いプログラムを作っている生徒でも、楽しくないと回答した生徒もいる。

補足: 同様の二極化は現行指導要領から導入された体育の「ダンス」でも見られる

以降、(楽しさ)の4/5を上位群、3以下を下位群として分析

(楽しさ)低い生徒もアクティブラーニング

略称	事後質問	上位群 (N=63)		下位群 (N=49)		p
		M	SD	M	SD	
(主体的)	従来の授業に比べて、自分で学習したと思う。	4.59	0.68	3.49	0.93	< 0.01
(考える)	従来の授業に比べて、自分で考えたと思う。	4.79	0.57	3.73	0.96	< 0.01
(集中)	従来の授業に比べて、集中して学習したと思う。	4.62	0.70	3.53	0.84	< 0.01
(協働)	授業中、プログラミングについて友達と相談した。	4.71	0.65	3.69	1.28	< 0.01

・(主体的)(考える)(集中)の項目について(楽しさ)上位群は非常に肯定的な反応を示し、下位群でも従来の授業に比べて肯定的である。

自由なプログラムの構想の難しさ

略称	事後質問	上位群 (N=63)		下位群 (N=49)		全体 (N=112)	
		<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>
(困難: 構想)	どんなプログラムにするか考えるのが難しかった.	4.63	0.62	3.90	1.27	4.31	1.03
(困難: Scratch)	Scratch自体が難しかった.	3.90	1.16	3.47	1.11	3.71	1.16

Scratchのプログラム自体より、どのようなプログラムを作るか難しいと感じた生徒が多い。自由な創造に慣れてない？



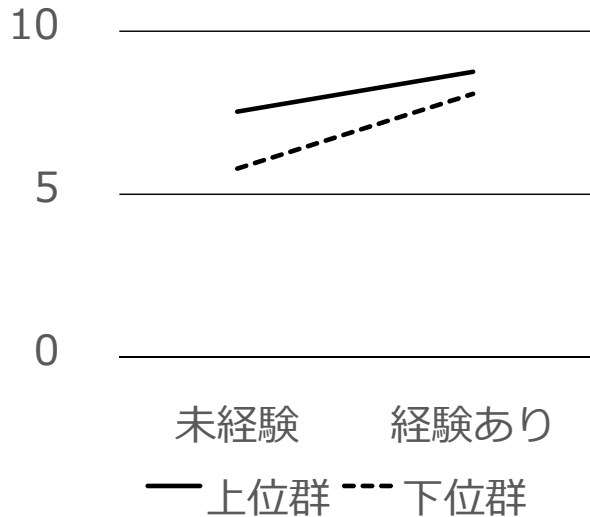
・吉田ら(2016)はどのように「手がかり」を教師が適切に与えるかという課題も示す。

プログラム内容の分析

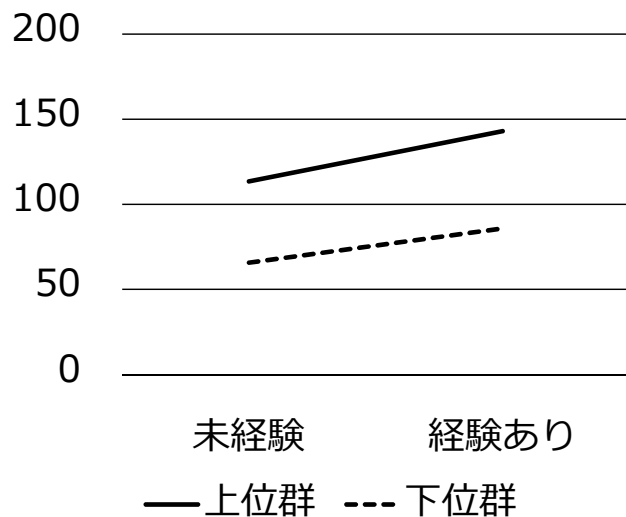
	if	if-else	論理演算子	無限ループ	ループ回数指定	終了条件付きループ	変数利用
未経験	76.0%	20.0%	16.0%	94.0%	52.0%	22.0%	38.0%
経験あり	91.9%	12.9%	16.1%	95.2%	62.9%	46.8%	71.0%
合計	84.8%	16.1%	16.1%	94.6%	58.0%	35.7%	56.3%

- ・ 32種類のプログラムの構成要素で分析(上図は抜粋)
- ・ 無限ループはScratchの場合、入門でも使用する。プログラムを動作させるための基本的な要素。

難しいこと知らないけど、がんばる生徒



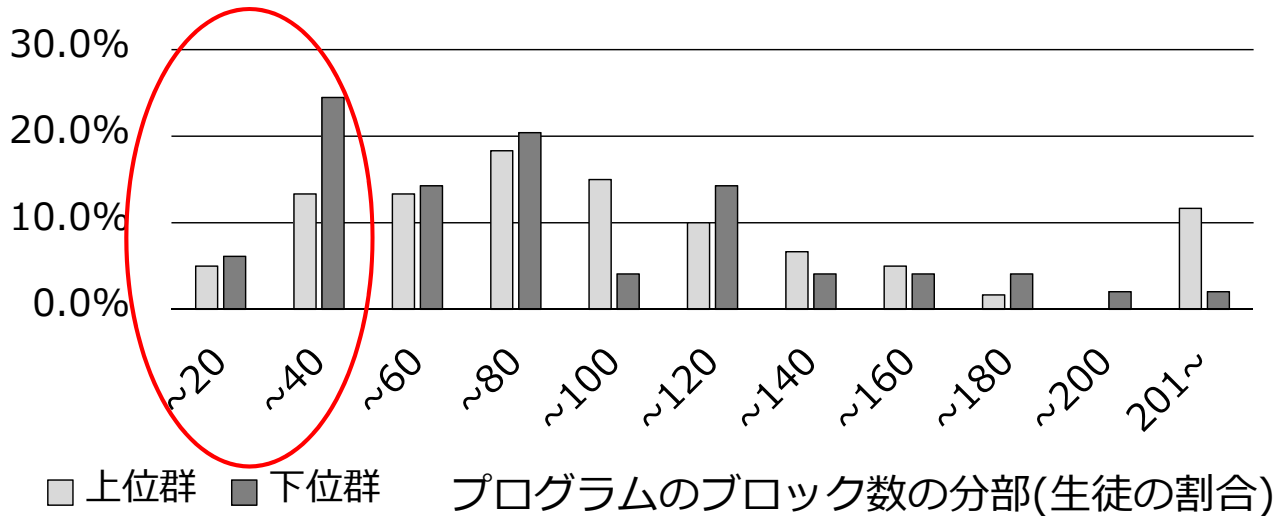
プログラムの構成要素数(分岐・ループ・変数等)



プログラムのブロック数の平均

- ・構成要素数は、Scratchの経験ありが有意に多い
 - ・ブロック数は、(楽しさ)上位群が有意に多い
- 上位群の中にはプログラムの知識は無いけど、どんどん大きなプログラムを作っていた生徒がいるかも。

サンプルプログラム程度の簡易なもの



サンプルプログラムでも30ブロック、40以下は簡易なもの

・(楽しさ)下位群で30%程度、全体で25%程度

プログラミング能力がないのでできないのか、創造性がないのでできないのか、はっきりしない。

今後の課題：全員が楽しく確実に修得するため

1. プログラミング能力の高い生徒が、本実践のような場面を「楽しくない」と感じる要因の明確化。
たぶん、少し上のレベルの課題の設定するかも
2. 自由なプログラムを作成するための必要最低限なプログラミングの知識と、その確実な習得の支援。
全くの初心者にはある程度、入門的なもの個人写経で作成することを示すかも
3. 創造性の観点から、自由なプログラミング作成の方向性を与える手がかりの支援方法の確立。
とりあえずの対応としてベースとなるプログラムの絞り込みあたりが有効かも

「情報I」のプログラミング教育に向けて

とりあえず、教師にプログラミング経験が無くても「教えない」プログラミング教育を先行期間にでもやってみたら (9月までには教材再整備の予定)

- ・生徒はわからないことがあると、どうしても教師に聞きたがる。そして、教師は教えたがる。
ワークショップ型の敵は、たぶん
 - ・日本の教師の一斉授業方法での教授能力の高さ
 - ・生徒の教師が丁寧に説明・指示してくれるという態度
- ・教師自身が授業中Scratchのプログラミングを学習する機会になります。そのうち、生徒の支援ができるようになります。
- ・すでにScratchを知っている生徒も数名いるはずなので、当面、それらの生徒に、他の生徒の支援してもらいましょう。

おまけ:フェーズ2の結果(1)

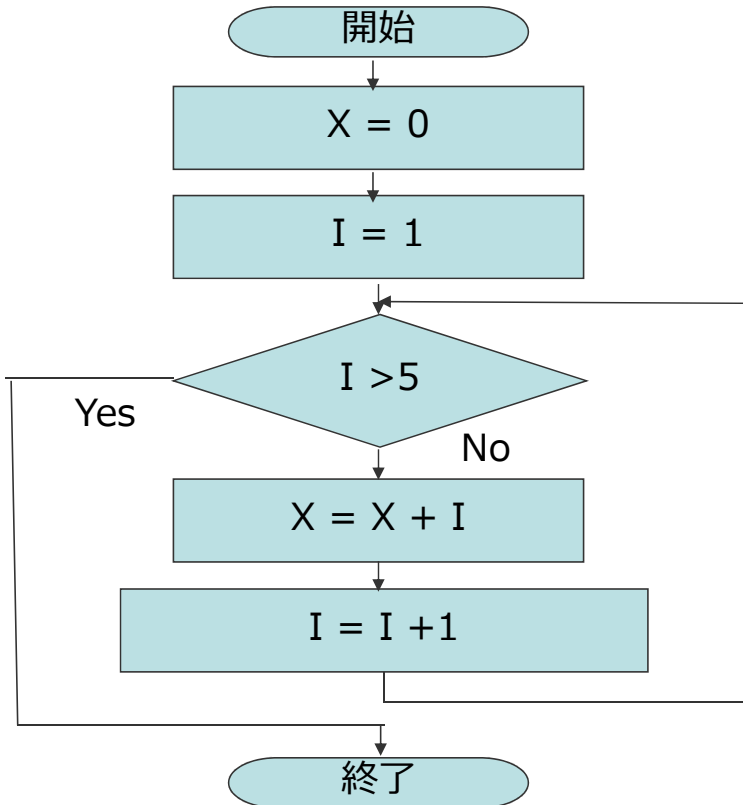
プログラムの基本構造とフローチャートを意識した自主的なプログラミング(資料の理解と課題プログラム)



配列/リストの添字に変数を使うと、理解できなくなる。

おまけ:フェーズ2の結果(2)

単純な1からn(5)の合計



ループ・カウンターとして変数を使った、繰り返し処理が理解できない。

例えば: このフローチャートに基づくScratchプログラムがあり、それを改修して、偶数の合計を求めるプログラムが作れない。

おまけ:ワークショップ型の体験するには 各地のCoderDojoに参加・見学

アイルランド発祥で、基本的に無償でプログラミング教育を提供するという憲章に賛同した人が、コーダー道場を開設(実施方法は各道場でバラバラ)。国内で現在150か所弱、

Coderdojo Japan

検索



全くプログラミングを知らない先生でも数回参加すると、メンターとしてプログラムを指導することができると思います。

お子さん(小学生~中学生)がいる方はいっしょにどうぞ

・ DojoCon Japan 2018
8月25日(日本橋) 全国のCoderDojoが主催する
年に一度の集まりです。

おまけのおまけ: 小学校プログラミング教育

ももとは、CoderDojo用に作った資料ですが、解説など作り小学校の先生向けの研修で2時間程度使いました。よかったら自由に使ってください (office系の元ドキュメントもあります)

・全体の説明資料

http://beyondbb.jp/TTEP/TTEP_GO20180809.pdf

・各資料のリンクトップページ

<http://beyondbb.jp/TTEP/index.html>



物語基本
書いた人 gohomev006



物語
書いた人 gohomev006



起承転結
書いた人 gohomev006



飛び出す地図
書いた人 gohomev006



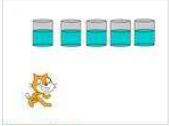
写真クイズ
書いた人 gohomev006



写真アニメ
書いた人 gohomev006



防犯ライト
書いた人 gohomev006



水溶液分類
書いた人 gohomev006



壺(写真)
書いた人 gohomev006



速度01
書いた人 gohomev006



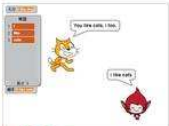
リスト_電卓
書いた人 gohomev006



都道府県人口比較
書いた人 gohomev006



都道府県統計データ
書いた人 gohomev006



AI英会話
書いた人 gohomev006



空気の粒05
書いた人 gohomev006

