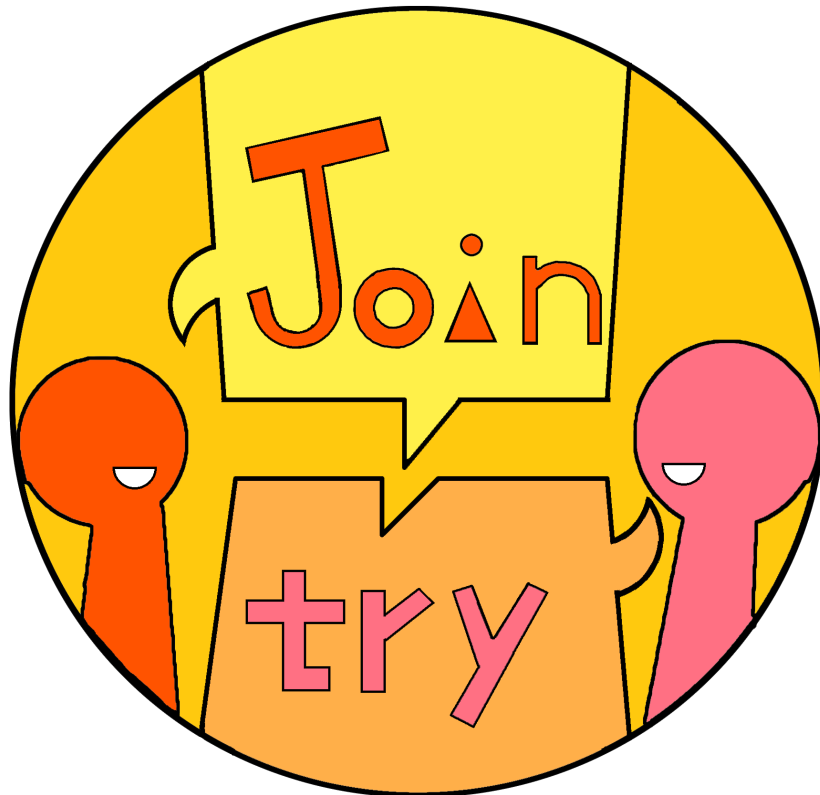


2013秋口忠三PBL 子ども向けプログラミング学習ツール 「Jointry」開発計画書



2013年5月11日（土）
秋口PBL
メンバー：権田・金本・角
担当教員：秋口教授
副担当教員：加藤教授、土屋助教

目次

- 1. [本プロジェクトの概要](#)
 - 1-1. [エレベーターピッチ](#)
 - 1-2. [プロジェクトの目的](#)
 - 1-3. [メンバーの目的](#)
- 2. [現状分析](#)
 - 2-1. [現状の課題](#)
 - [環境の課題](#)
 - [動機の課題](#)
 - [成果の課題](#)
 - 2-2. [類似プロダクト](#)
 - 2-3. [類似プロダクトの課題](#)
 - [環境の課題](#)
 - [動機の課題](#)
 - [成果の課題](#)
- 3. [想定する利用者と価値](#)
 - 3-1. [利用者層](#)
 - 3-2. [利用者の価値](#)
 - [「環境の課題」の解決](#)
 - [「動機の課題」の解決](#)
 - [「成果の課題」の解決](#)
- 4. [システム要件](#)
 - 4-1. [やりたいことリスト（ユースストーリー）](#)
 - [学習者向け](#)
 - [「環境の課題」の解決](#)
 - [「動機の課題」の解決](#)
 - [「成果の課題」の解決](#)
 - [その他（プラスアルファ）](#)
 - [講師向け](#)
 - [その他（プラスアルファ）](#)
 - 4-2. [やらないことリスト](#)
 - 4-3. [使用技術](#)
- 5. [品質](#)
 - 5-1. [機能的性](#)
 - 5-2. [信頼性](#)
 - 5-3. [使用性](#)
 - 5-4. [効率性](#)
 - 5-5. [保守性](#)
 - 5-6. [移植性](#)
- 6. [その他の成果物](#)
- 7. [プロジェクト体制](#)
 - 7-1. [開発メンバー](#)
 - 7-2. [ステークホルダー](#)
 - 7-3. [開発プロセス](#)

[7-4. コミュニケーション手段](#)

[8. スケジュール](#)

[8-1. 全体スケジュール](#)

[8-2. 直近のタスク](#)

[1Qのタスク](#)

[9. 考えられるリスク](#)

[10. 今後の展望](#)

[10-1. 開発者の増加](#)

[10-2. 利用者の増加](#)

[付録A. 参考文献](#)

[付録B. 類似プロダクトの紹介](#)

[Scratch](#)

[Blockly](#)

[VISCUIT](#)

[アルゴロジック](#)

[プログラミン](#)

[前田ブロック](#)

[PenCode](#)

[App Inventor](#)

[Snap!](#)

1. 本プロジェクトの概要

1-1. エレベーターピッチ

「Jointry」は、小学校高学年向けのプログラミング学習ツールである。キーボードを使わずに簡単にプログラミングの学習を可能とする。既存のプログラミング学習ツール（Scratchなど）とは違い、友達や先生と一緒に学習できる共同プログラミング機能が備わっている。

1-2. プロジェクトの目的

秋口PBLでは「特定ドメインのアプリケーションを開発するための基盤となるソフトウェアを開発する」ことを目標としている。本プロジェクトでは、「特定ドメイン」を「プログラミング学習」に設定し、これからソフトウェア開発を始める人を対象にしたプログラミング学習環境の構築を目的とする。

また、開発プロセスには「反復型開発手法」と「顧客開発」の手法を導入し、実際にユーザーに利用してもらいながら、そのフィードバックを反復的にソフトウェアに反映 / 改善していくものとする。

1-3. メンバーの目的

金本：JavaFXとそれを用いたコーディング手法、様々なプロセス手法を学ぶ。

角：GUIプログラミングの効果的な設計について学習する。実際のユーザーからのフィードバックを取捨選択しながら、プロダクトに反映する開発方法を体験する。

権田：JavaFXのプログラミング、チームでのアプリケーション開発の手法

2. 現状分析

2-1. 現状の課題

まず、プログラミング初心者の課題を「仮説」として以下に挙げる。

環境の課題

タブレットが普及している今、パソコンを持っていないこともある。プログラミングを学習するにはパソコンを使いこなす必要があるが、キーボードやマウスといった道具を使いこなすスキルが身についていない。

動機の課題

プログラミングは難しいといった先入観がある。何から始めればよいのか分からず、興味が持てない。

成果の課題

誰かと一緒にプログラミングしたり、作ったものを発表 / 公開したりする機会がないため、興味を持ったとしても続かない。あるいは、楽しさを実感できない。

2-2. 類似プロダクト

プログラミング初心者向けの既存の学習ツールを以下に挙げる。

| 名称 | 動作環境 | 登場時期 | 対象者 | 開発元 | 特徴 |
|--------------|------------------------------|-------|--------------------------|-----------------------------|-----------------------------|
| Scratch | Squeak (Windows、Mac、Linux) | 2006年 | 文法的に正しい構文を読んでも未だに書けない子供達 | MITメディアラボ | ツールや機能が安定したブロック言語環境 |
| Blockly | JavaScript | 2012年 | | Google | Python等を生成できる機能を持ったブロック言語環境 |
| VISCUIT | Flash | 2003年 | 素人や子供 | NTT | メガネ型ビジュアルプログラミング |
| アルゴリズム | Flash | 2010年 | プログラミング経験の無い中学生・高校生 | 一般社団法人 電子情報技術産業協会 (JEITA) | ブロック言語を使ったアルゴリズム体験ゲーム |
| プログラミング | Flash | 2010年 | プログラミングをこれから始める子供 | 文部科学省 | ブロックが絵になっているブロック言語環境 |
| 前田ブロック | JavaScript | 2012年 | 小学生 | ユビキタスエンターテインメント (UEI) | Blocklyを参考にした環境 |
| PenCode | iOSアプリ | 2013年 | N/A | Toshinori Hanya | ブロック言語を題材にしたゲーム |
| App Inventor | JavaScript | 2010年 | プログラミングを専門的に学んだことがない学生 | Google→MITメディアラボ | androidアプリを生成できる環境 |
| Snap! | JavaScript | 2009年 | 高校生・大学生 | カリフォルニア大学バークレー校 | Scratchがブラウザで動いている |

2-3. 類似プロダクトの課題

環境の課題

FlashやJavaScriptで作成されたものであればブラウザを使って操作可能だが、タブレット向けに特化して開発されたものは少なく、キーボードを使いこなさなければいけないという課題がさほど解消されていない。

動機の課題

大半が初心者や子ども向けに作成されており、操作風景が動画で解説されているものもあるので、プログラミングを始めるためのハードルは低くなっていると言える。しかし、自由度の高いアプリケーションが多く、子どもによっては「そもそも作りたいものがない」という状態が起こりうる。できれば、事前に講師が適切な目標を設定できるといいだろう。また、プログラミングそのものが楽しくなるような仕掛け（ゲームなど）が必要である。

成果の課題

1人で学習することを目的に作られているものが多く、他の人と一緒にプログラミングをするという体験が得られにくい。また、作成したアプリケーションをトレンドな市場に出すことができない。

3. 想定する利用者と価値

3-1. 利用者層

- 小学高学年（4・5・6年生）
 - 「8～10歳ぐらいになると、予測し、計画をたてる能力が徐々に発達（中略）11、12歳になると、一般的な構造をつくり、汎用的な道具を工夫する能力が大きく進歩する」（参考文献1）
 - 「5年生という子どもたちの特性は、多少抽象的な考え方の入り口が見えてきた段階である。したがって具体物を扱ったり、動きが確認できたりするような素材を学習に取り入れていくことが、理解につながる」（参考文献2）
 - 中学生になると英単語を理解できるようになり、汎用言語が使えるので、対象から外す。
- 上記を対象にプログラミングを教える講師

3-2. 利用者の価値

「環境の課題」の解決

パソコンやキーボードを使わずに、タブレットのタッチ操作だけでプログラミングが可能になる。

「動機の課題」の解決

お絵かきやゲームのような機能があるので、最初のプログラミングのハードルが低い。

「成果の課題」の解決

双方向通信ライブラリを利用した「ペアプログラミング」機能を使って、誰かと一緒に楽しみながら学習できる。また、作成したプログラムを「本物のアプリケーション」として公開できる。学習機能を搭載することで、小さな目標から達成できる。

4. システム要件

4-1. やりたいことリスト（ユースタートリー）

学習者向け

「環境の課題」の解決

- タッチパネルを使ってプログラミングをビジュアルに学ぶ事が出来る

「動機の課題」の解決

- 自分の描いた絵を動かす事が出来る
- マクロのようにキャンバスの画像を動かしたらプログラムを作ることが出来る
- カメラで撮影したものをプログラミングに利用出来る
- ゲーム感覚でプログラミングを学ぶ事が出来る

「成果の課題」の解決

- 作成したプログラムをトレンドな市場に出せるプログラムに生成出来る
- プログラミングを友達や講師と一緒に学ぶ事が出来る

その他 (プラスアルファ)

- 作ったプログラムに名前を付けモジュールに出来る
- Webに公開されている他の人が作ったプログラムのソースを見ることが出来る

講師向け

その他 (プラスアルファ)

- プログラミングを講師や友達と一緒に学ぶ事が出来る
- スクリプト言語で問題を作ることが出来る

4-2. やらないことリスト

- iOS対応
- グラフィックデザインに関係すること (デザイナが必要になること)

4-3. 使用技術

- JavaFX
 - GUI全般。
 - JVMに標準装備されたメインストリームのGUIツールキットであるため。
 - 開発自体はデスクトップで行う。最終的にはJVMの動作するタブレット端末でも動作するように、タッチ操作などのイベントにも対応する。
 - Oracleとして正式にタブレット端末に対応する時期が未定である (リスク参照)。
- 双方向通信ライブラリ
 - 秋口PBLのライブラリを利用する。
- JDK8の新機能
 - Lambdaによる処理の抽象化。
 - リリースが2014年に延期されたため、OpenJDK8を使用する予定。

5. 品質

5-1. 機能性

プログラミング学習ツールとしての基本的な機能を提供する。

5-2. 信頼性

学習履歴の維持を優先的に考え、障害発生時にも履歴からのデータ復帰を可能にする。

5-3. 使用性

ターゲットである小学生でも扱えるように、できるだけキーボード操作を行わないグラフィカルに富んだUIにする。

5-4. 効率性

共同プログラミングの機能では、ネットワークや端末に負荷がかからないような設計をする。

5-5. 保守性

オープンソース化を行い開発者の増加を狙っているので、再利用可能であり機能拡張の行いやすい設計をする。

5-6. 移植性

JVMを搭載した端末であれば動作するように、JavaFXを使ったアプリケーションを作成する。

6. その他の成果物

- 開発計画書
- 機能仕様書
- 開発設計書 (アーキテクチャレベルのクラス図)
- ユーザー操作マニュアル (子ども向け、講師向け)

7. プロジェクト体制

7-1. 開発メンバー

権田、角、金本

7-2. ステークホルダー

- 秋口教授
- 加藤教授
- 土屋助教
- 外部評価者
- ユーザーとなる小学生
- ユーザーとなる講師

7-3. 開発プロセス

1Q: プロトタイプ作成を通して工数や規模感を把握することを目的とするので、開発プロセスは特に設定しない。

2Q: プロトタイプの成果をもとにして、本番用のアーキテクチャを設計する。機能はBacklogの「課題」に登録し、それを2週間ごとにリリースする。各イテレーションでは、実工数を計測して、今後の予測を立てられるようにする。また、可能な限り実際の利用者にインタビューして、その意見をアプリケーションにフィードバックする「顧客開発」の手法を導入する。

7-4. コミュニケーション手段

- Backlog
- Chatwork
- メーリングリスト
- コアミーティング

8. スケジュール

8-1. 全体スケジュール

- 1Q：仕様の構想、使用技術の調査
- 2Q：プロトタイプの実成、アーキテクチャの設計
- 夏休み：アーキテクチャの洗練
- 3Q：反復的開発の開始
- 4Q：反復的開発の継続

8-2. 直近のタスク

Backlogの「課題」で管理する。現在（2013-04-27）時点の課題を以下に挙げる。

1Qのタスク

- プログラミングの仕組みの調査
- 画面仕様のイメージ作成
- 画像のドラッグドロップ
- Netbeansプロジェクトをマシンに依存させないようにする
- 音を鳴らしてみる
- 10pxオブジェクトを動かしてみる
- 背景の切り替え
- 透過処理
- ブロック部品の合体
- JavaFXのマルチタッチプログラミング
- JavaFXはカメラ画像を扱えるのかを調査する

2Qのスケジュール

- プログラミングをまとめる（関数）
- プログラミングが動く
- プログラミングになっている
- 任意の画像を書くためのペイントツールの作成
- マクロ機能の実成
- メニュータブを作る
- JavaFXがandroid or iOSで動く条件や制約を調査する
- Brokerの解析

9. 考えられるリスク

- JavaFXがAndroidで動かない（対応されない）
 - Javaの動作するWindowsタブレット端末を検討する
- インタビューできる小学生が見つからない

- 誰もいなければ、秋口先生の娘さんを対象にする

10. 今後の展望

10-1. 開発者の増加

オープンソースライセンスを適用し、github等のインターネットリポジトリに掲載する。

10-2. 利用者の増加

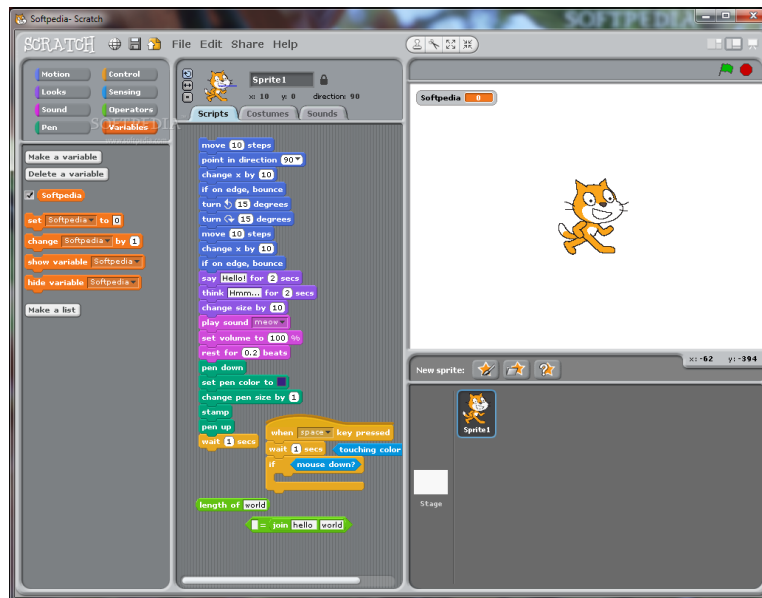
講師向けのワークショップを開催するなどして、草の根的に利用者を増やしていく。

付録A. 参考文献

1. アラン・ケイ, 鶴岡雄二 [翻訳], 浜野保樹 [監修] “アラン・ケイ”, アスキー出版社, 1992.
2. 佐藤和浩, “学習指導要領移行期における小学校情報教育の取り組み,” 2010.
3. 辰己丈夫, 兼宗進, 久野靖, “ドリトルと「情報教育の音楽化」,” 2005.
4. 佐藤和浩, 紅林秀治, 兼宗進, “小学校におけるプログラミング活用の現状と課題,” 2005.
5. 柿内達真, 取越翔太郎, 桜打彬夫, 大東和忠幸, 野口尚吾, 高田秀志, “SnowBoy : 教室
内でのプログラミング作品共有による共同創作が可能な初等教育向け協調学習支
援システム,” 2009.
6. 軽野宏樹, 木實新一, 上林弥彦, “ALAN - Kプロジェクト : Squeak を活用した創造的
な情報教育の試み,” 2003.

付録B. 類似プロダクトの紹介

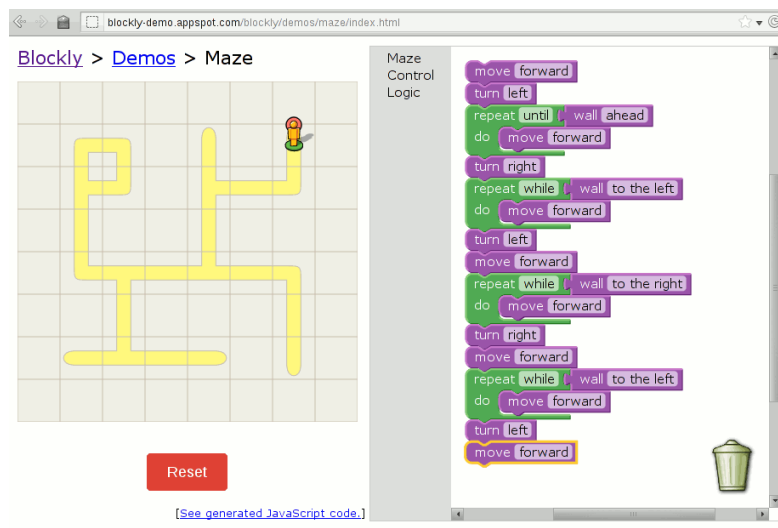
Scratch



アラン・ケイ博士が作ったオブジェクト指向プログラミング言語「スクイーク (Squeak Etoys) 」をベースにした小学生向けプログラミング環境。スクイークのよさを生かしつつ、より使いやすくするための様々な工夫がされている。

<http://etoys.jp/scratch/scratch.html>

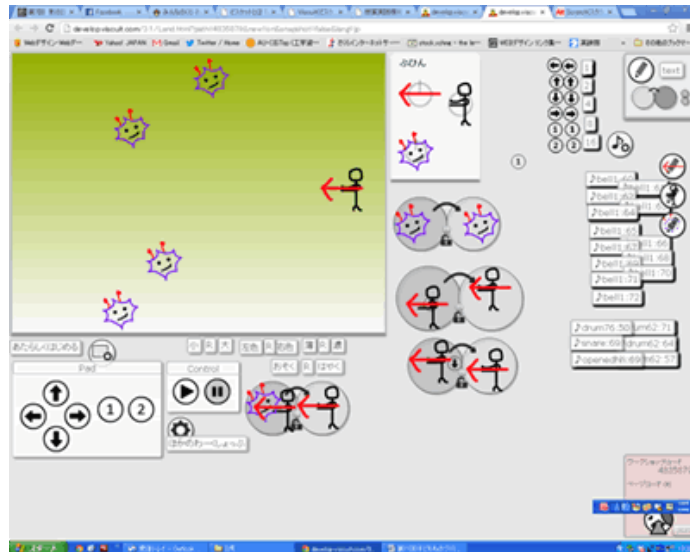
Blockly



Googleの開発するウェブベースのプログラミング学習環境。ブロックを使ってプログラムを構築する。「Javascript」「Dart」「Python」など他のプログラミング言語に変換可能であるところが特長。

<https://code.google.com/p/blockly/>

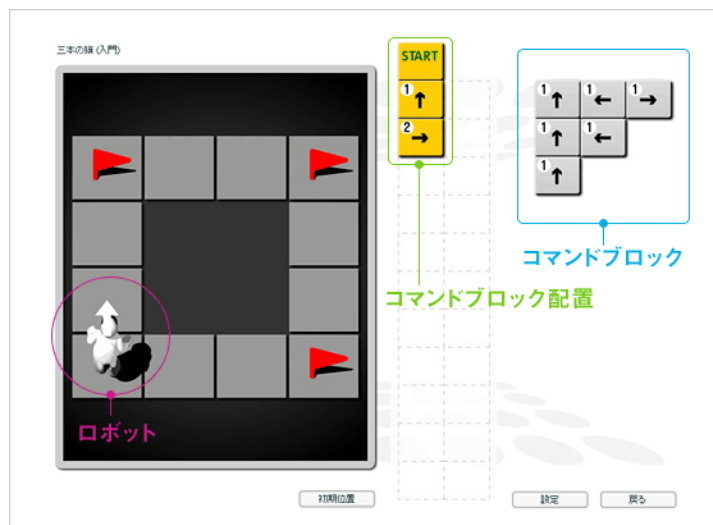
VISCUIT



マウスだけで絵を動かすことができる。お絵かきソフトの延長線上にあると言える。小さな子どもでも操作可能。InfoTalk第37回でも発表されている。

<http://www.viscuit.com/>

アルゴロジック



ゲーム要素が盛り込まれているところが特長。ブロックが単なる四角なのでプログラミングを構築しやすい。パズルゲームの一種とも言える。

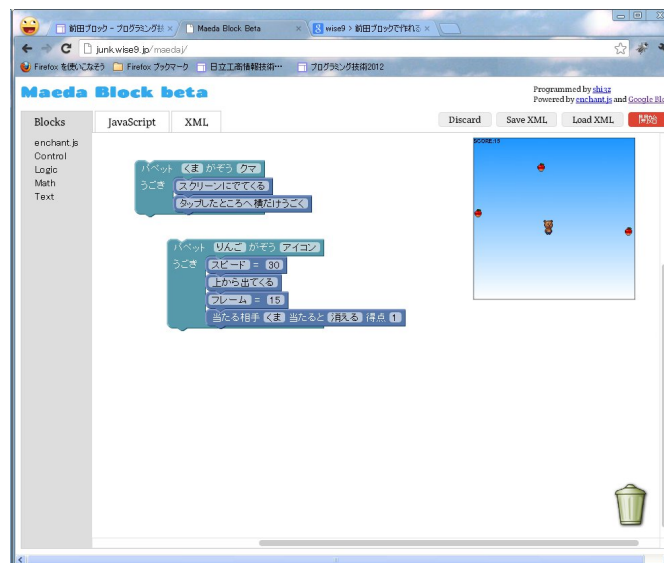
<http://home.jeita.or.jp/is/highschool/algo/index.html>

プログラミン



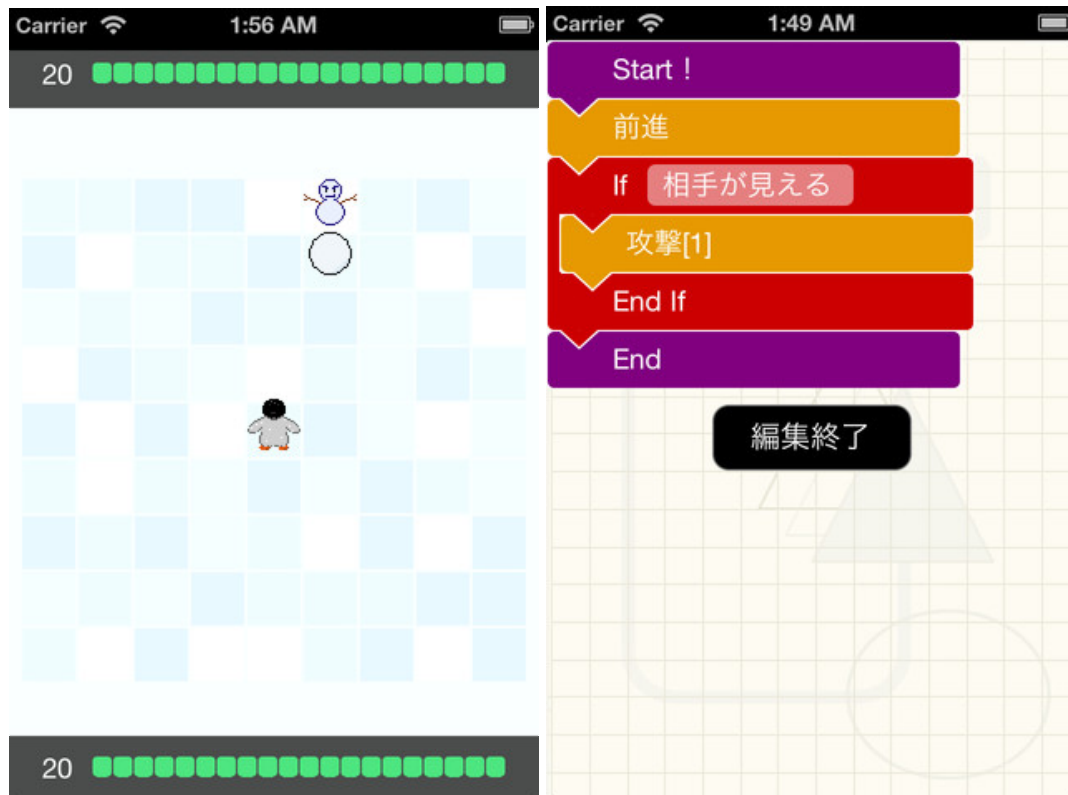
Scratchとよく似ている。文科省が作っているところとウェブで動くところが特長。
<http://www.mext.go.jp/programin/>

前田ブロック



Javascriptベースのブロック型学習ツール。これをもとにenchant.jsで書き直されたものが「MOONBlock」の名前でAndroidをベースにしたタブレット「enchantMOON」に搭載されている。
<http://enchantmoon.com/>

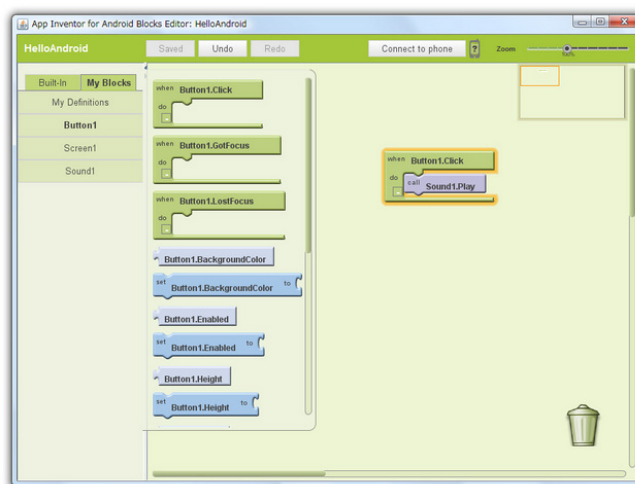
PenCode



iPhone用の学習ツール。ゲームを使うところはアルゴリズムと同じ。

<https://itunes.apple.com/gb/app/pencode/id624675093?mt=8>

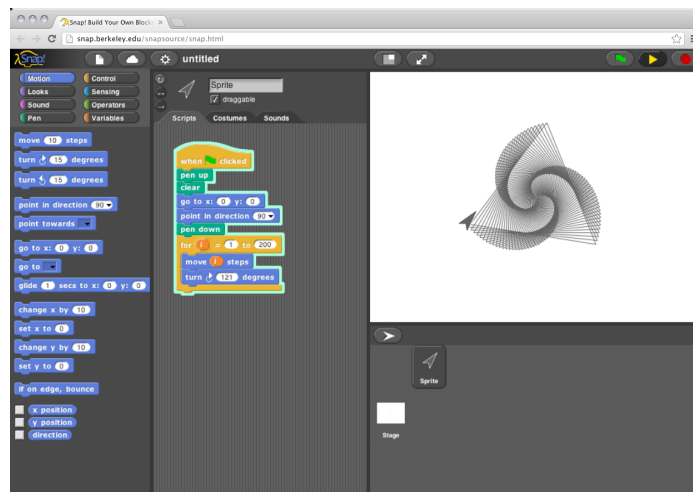
App Inventor



androidアプリの開発ツール。GUIをブラウザ上で組み立て、jarファイルのブロック言語のエディタを使いアプリを作成する。作成したアプリは開発環境上から公開が可能。このツールの使用にはgoogleアカウントが必要。

<http://appinventor.mit.edu/>

Snap!



ScratchをJavaScriptで実装して、独自のブロックを作成できるようにしたもの。リスト・手続き・継続などがファーストクラスに定義できるので、より上級者（高校生や大学生）を対象にしている。

<http://byob.berkeley.edu/>