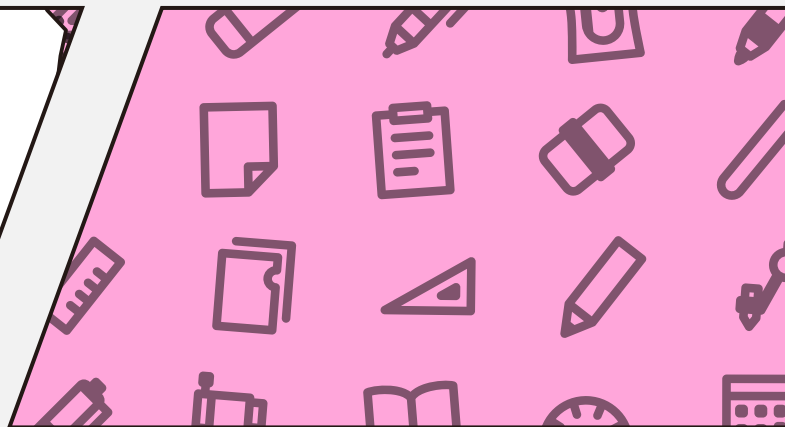


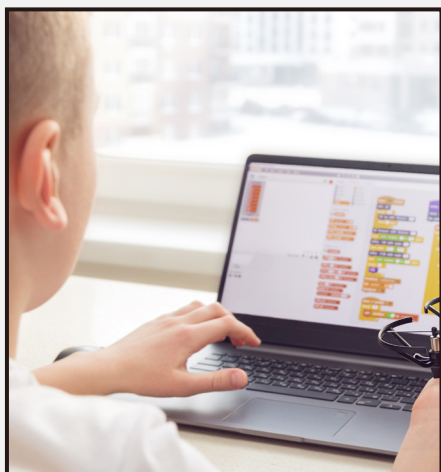
みんな
で
は
じ
め
る



小中学生のための ITエンジニア 体験教育

プログラミング
教育

キャリア教育



入門者～初心者



NTTテクノクロス株式会社

はじめに

本書はNTTテクノクロス株式会社の中で技術を愛し、人に教えることが大好きな有志が、昨今社内外で開催している「ITエンジニア体験教育」について、実践している内容を概要から具体的な方法までまとめた本です。

執筆に至った背景

NTTテクノクロス株式会社では、本書の執筆者が中心となり、CSR活動として学生・児童に対してITエンジニア体験教育を提供しています。提供の場は社内でのイベントだけでなく、学校に伺って講演やクラブ活動の形で実施しています。

このような活動を行う中で、習得した技術・教育方法について当社内でとどめるのではなく、さらに多くの世の中のエンジニア・教員に広めることで、各々の独自色を持ったITエンジニア体験教育がそれぞれの教育の現場で実施されるきっかけにつなげたいと考えようになりました。

今回は当社独自、すなわち **NTTテクノクロス流のITエンジニア体験教育** を具体的な活動内容レベルでお伝えし、同じような活動をお考えの皆様のきっかけとしていただければと思い、本書を執筆しました。

ITエンジニア体験教育について

本書では、実際に手を動かしながらプログラムを作る体験（＝プログラミング教育）と、ITエンジニアとして第一線で活躍する人の話を聞きながら職業の理解を深めつつ自分の将来を想像する体験（＝キャリア教育）を合わせて「**ITエンジニア体験教育**」という名前を定義しました。

お問合せ先

本書に対するお問合せは下記メールアドレス宛にお願いいたします。

- xmobile.support-ml@ntt-tx.co.jp

注意事項

本書は情報の提供のみを目的としています。

免責事項

本書の内容を用いた教育活動は、必ずご自身の責任と判断によって行ってください。

教育活動の結果について、著者はいかなる責任も負いません。

なお、この本は2023/05/20時点の内容です。開発環境やツールのバージョンアップ等により内容に差異が生じる可能性があります。その点は了承願います。

商標について

本書に掲載するシステム名や製品名は、関係各社の標章または登録商標です。
本書では、™、®等は省略しています。

著作権について

本書に掲載するソフトウェア、アプリ、プログラム等に関する著作権は、関係各社に帰属します。
本書では、©等は省略しています。

対象読者

この本は以下のような皆様を想定して書いています。

- プログラミング教育やキャリア教育を新たに教育現場に取り入れたい先生
- 学校と連携した活動を行おうと考えている企業の社員

本書を読み進めるに際して、前提となる知識はありません。ぜひそれぞれの現場でご活用いただければと思います。

目次

• はじめに	1
• 対象読者	3
• 1章 NTTテクノクロス流 学生向けプログラミング・キャリア教育	5
◦ 1-1.活動方針	6
◦ 1-2.活動区分	7
◦ 1-3.活動歴	8
• 2章 プログラミング教育(Scratch)	10
◦ 2-1.動作環境	11
◦ 2-2.ゲームアプリ	12
◦ 2-3.アニメーション	29
◦ 2-4.トラブルシューティング	34
◦ 2-5.コラム	35
• 3章 プログラミング教育(Androidアプリ)	37
◦ 3-1.動作環境	37
◦ 3-2.サンプル	42
◦ 3-3.トラブルシューティング	56
◦ 3-4.コラム	59
• 4章 キャリア教育	64
◦ 4-1.活動の目的	65
◦ 4-2.コンテンツ例	66
◦ 4-3.コラム	72
• おわりに	73
• 表紙デザインについて	74
• 著者紹介	75

1章 NTTテクノクロス流 学生向けプログラミング・キャリア教育

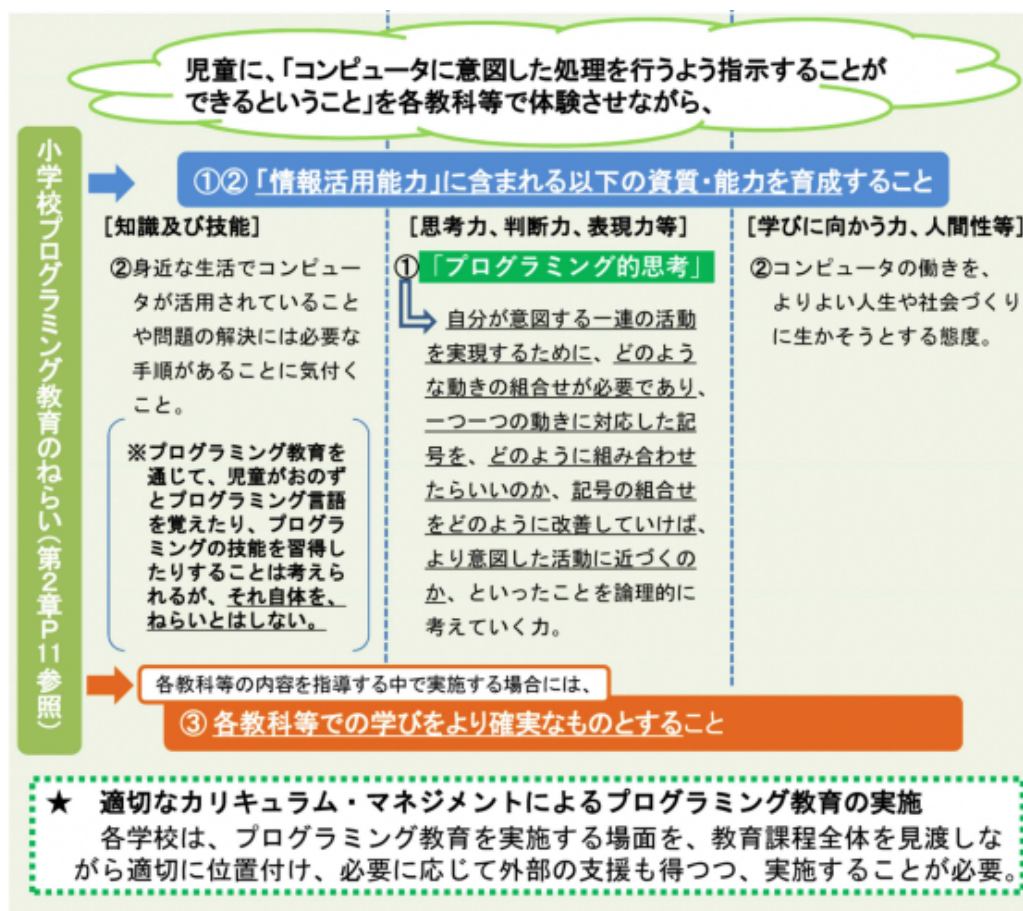
この章ではNTTテクノクロス流 学生向けプログラミング・キャリア教育について紹介します。

- [1-1.活動方針](#)
- [1-2.活動区分](#)
- [1-3.活動歴](#)

1-1.活動方針

2017年度からNTTテクノクロス株式会社でCSR活動(社会貢献活動)の一環として取り組んでいる『プログラミング・キャリア教育』は、文部科学省『小学校プログラミング教育の手引き 第三版』で定義されている『「情報活用能力」を構成する資質・能力の育成』を活動方針の根幹としています。

なお、『「情報活用能力」を構成する資質・能力の育成』とは、以下イメージの通り、『知識及び技能』・『思考力、判断力、表現力等』・『学びに向かう力、人間性等』で構成されています。(イメージは、文部科学省『小学校プログラミング教育の手引き 第三版』より引用)



活動方針

1-2.活動区分

NTTテクノクロスで行なっている活動は、文部科学省『小学校プログラミング教育の手引き 第三版』で定義されている活動区分における『C・D・E』に該当し、『D』の割合が最も多いです。(イメージは、文部科学省『小学校プログラミング教育の手引き 第三版』より引用)

図5 小学校段階のプログラミングに関する学習活動の分類

-
- A 学習指導要領に例示されている単元等で実施するもの
 - B 学習指導要領に例示されていないが、学習指導要領に示される各教科等の内容を指導する中で実施するもの
 - C 教育課程内で各教科等とは別に実施するもの
 - D クラブ活動など、特定の児童を対象として、教育課程内で実施するもの
 - E 学校を会場とするが、教育課程外のもの
 - F 学校外でのプログラミングの学習機会

活動区分

1-3.活動歴

1-3-1.プログラミング教育誕生のキッカケ

2017年度 筆者(神長貴博)の長男と長女が、後に「プログラミング教育」を行う学校：川崎市立上丸子小学校で一緒に過ごす最後の年でした。その年の始めから長男の卒業前に記憶に残るイベントをやりたいと思い、業務の合間にアレコレ考えを巡らせていました。そうした中、とあるニュースサイトで **2020年度から「小学生向けプログラミング教育」が開始される** ことを知りました。

私が業務の一貫で社員向けの技術研修の講師や勉強会を主催していたこと、仕事に関わる技術分野以外でも発信の場を求めていたこともあり、「小学生向けのプログラミング教室」を開催したら面白いのでは？と考えるようになりました。

そして、この考えを町内会で知り合った上丸子小学校のPTA関係者に相談したところ、川崎市教育委員会の方を紹介頂き、**地域の寺子屋事業**の一貫として、**2018年3月に川崎市立上丸子小学校で「第1回 小学生向けのプログラミング教室」**の開催することになりました。こうして、NTTテクノクロス株式会社のCSR活動 **プログラミング教育**が誕生しました。



小学生向けのプログラミング教室

1-3-2.キャリア教育誕生のキッカケ

2017年度 筆者(神原健一)が、ある中学校と縁がありキャリア教育をお願いできないかという依頼を受けました。実は私の元同期である友人が転職し、その中学校(横浜市立東山田中学校)で教員として働いていました。私は当時、ITエンジニアとして、アプリ開発に加え、講演や執筆など外部活動に取り組み、これら活動の状況を社外に向けて情報発信していました。この取り組みを見てくれていた、その友人から、キャリア教育の一環として、その経験を子供たちの前で話してくれないか?と相談されたのがきっかけでした。

自身のこれまでのキャリアを振り返ってみるとともに、未来を担う子供たちに向けて、自分の経験を語ることで何か貢献できることがあるなら挑戦してみようと考え、この依頼をお受けしました。

こうして **2017年9月に横浜市立東山田中学校で実施したのが、キャリア教育の初回となりました。**

1-3-3.活動の歴史

年度	活動概要
2017年 度	<ul style="list-style-type: none"> ・横浜市立東山田中学校 キャリア教育 ・川崎市教育委員会 寺子屋事業 第1回 プログラミング教育イベント開催(川崎市立上丸子小学校)
2018年 度	<ul style="list-style-type: none"> ・川崎市教育委員会 寺子屋事業 第2、3回 プログラミング教育イベント開催(川崎市立上丸子小学校) ・全日本教育工学研究協議会 全国大会 川崎大会登壇(外部カンファレンス初登壇) ・新島村立新島小学校 プログラミング体験イベント・キャリア教育開催 ・横浜市立東山田中学校 キャリア教育 ・大学向けキャリア教育
2019年 度	<ul style="list-style-type: none"> ・川崎市立上丸子小学校 プログラミングクラブ発足 ・川崎市立上丸子小学校 5学年・6学年 プログラミング授業開催 ・全日本教育工学研究協議会 全国大会 島根大会登壇 ・横浜市立東山田中学校 キャリア教育
2020年 度	<ul style="list-style-type: none"> ・川崎市立上丸子小学校 プログラミングクラブ(Scratch公式サイトにアプリ公開) ・全日本教育工学研究協議会 全国大会 鹿児島大会登壇 ・みなとみらい本町フェスティバル SDGsブース出展(プログラミング体験ブース) 横浜市立本町小学校
2021年 度	<ul style="list-style-type: none"> ・川崎市立上丸子小学校 プログラミングクラブ ・社内オンライン・オフライン プログラミング体験開催 ・横浜市立東山田中学校 キャリア教育(オンライン開催)
2022年 度	<ul style="list-style-type: none"> ・川崎市立上丸子小学校 プログラミングクラブ ・社内オフライン プログラミング体験開催 ・横浜市立東山田中学校 キャリア教育 ・全日本教育工学研究協議会 全国大会 愛知・春日井大会登壇 ・Salesforce社主催イベント 『Trailblazer Communityサステナビリティイベント#2』 私達ができる「教育」に関する取り組み事例』登壇 ・第10回学生スマートフォンアプリコンテストキャリア教育(スペシャルセッション)

2章 プログラミング教育(Scratch)

この章ではプログラミング教育(Scratch)について紹介します。

- [2-1.動作環境](#)
- [2-2.ゲームアプリ](#)
- [2-3.アニメーション](#)
- [2-4.トラブルシューティング](#)
- [2-5.コラム](#)

2-1.動作環境

Scratch 3.0は、[Scratch公式](#)から入手できます。

対応する環境は、以下になります。(Scratch公式サイトより引用)

- Windows 10+
- macOS 10.13+
- ChromeOS
- Android 6.0+



動作環境

2-2.ゲームアプリ

ここでは『色々な場所から落ちてくるリンゴを捕る』ゲームを活用したプログラミング教育の狙いと作成方法を説明します。

なお、このアプリは[Scratch公式サイト](#)で数多く公開されているリンゴキャッチゲームを参考に作成したものです。

2-2-1.プログラミング教育の流れと目標

このアプリは、2019年度5～10月に川崎市立上丸子小学校のプログラミングクラブ(4～6学年 34名が所属)で生徒が開発したものです。開発は、生徒を異学年・2～3名のチームに分け、プログラミング経験に応じて機能を分担、1人・1PCで進める方式を採用しました。

なお、アプリは3つのプログラムで構成されており、難易度は以下の通りです。

- 難易度高：リンゴ(Apple)
- 難易度中：ボウル(Bowl)
- 難易度低：背景(Background)

アプリの完成は、Scratchのエクスポート・インポート機能を使って、以下の様に機能の結合を行いました。

- リンゴ(Apple)とボウル(Bowl)の開発完了後、Scratchのエクスポート機能を使い、プログラムをエクスポート。
- 背景(Background)の開発をしているPCにエクスポートしたプログラムをScratchのインポート機能を使って完成。

プログラミング教育の目標は、活動方針に則って以下を設定しました。

- 知識及び技能：プログラムは人が作成していることを理解する
- 知識及び技能：コンピュータに意図した処理を行わせるには必要な手順があることを理解する
- 知識及び技能：コンピュータにはできること・できないこととがあることを理解する
- 思考力、判断力、表現力等：「プログラミング的思考」の育成
- 学びに向かう力、人間性等：他者(異学年の生徒)と協働しながらねばり強くやり抜く態度の育成
- 学びに向かう力、人間性等：集団の一員として自主的・実践的な態度の育成

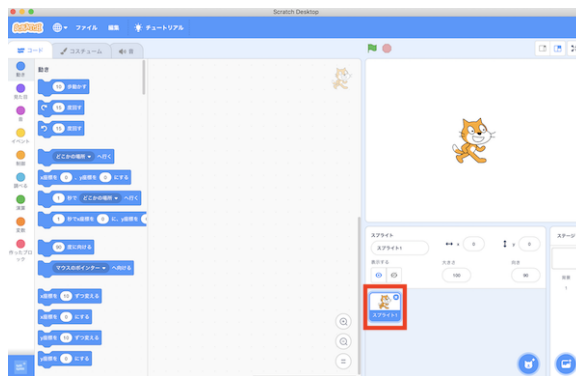
2-2-2.作り方

ここでは、機能毎のプログラムの作り方を説明します。

2-2-2-1.ボウルのプログラムの作り方

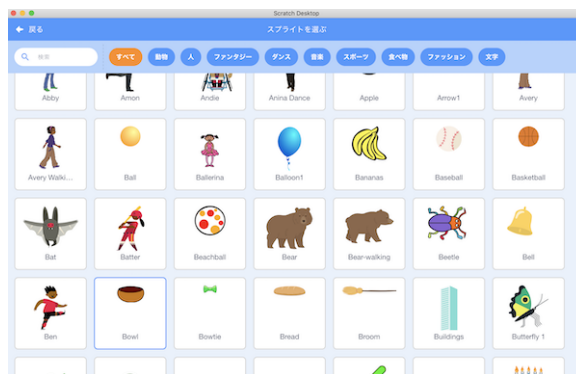
プログラムを作る前の準備

- スプライト1を削除する。(スプライト1を選択→×をクリック)



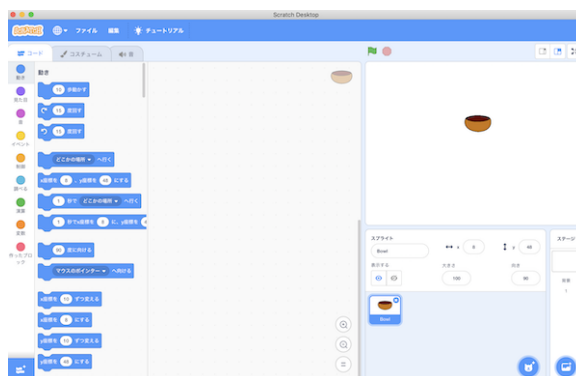
ボウルのプログラムを作る前の準備1

- ボタンをクリックする。
- ボウル(Bowl)を選択、クリックする。



ボウルのプログラムを作る前の準備2

- スプライトが設定されていることを確認する。



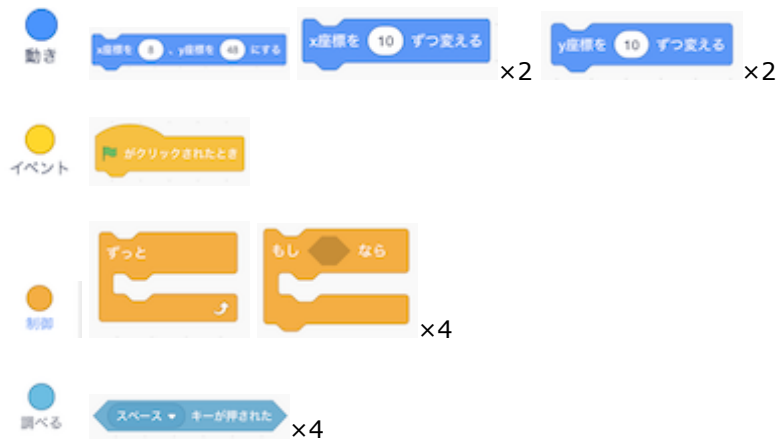
ボウルのプログラムを作る前の準備3

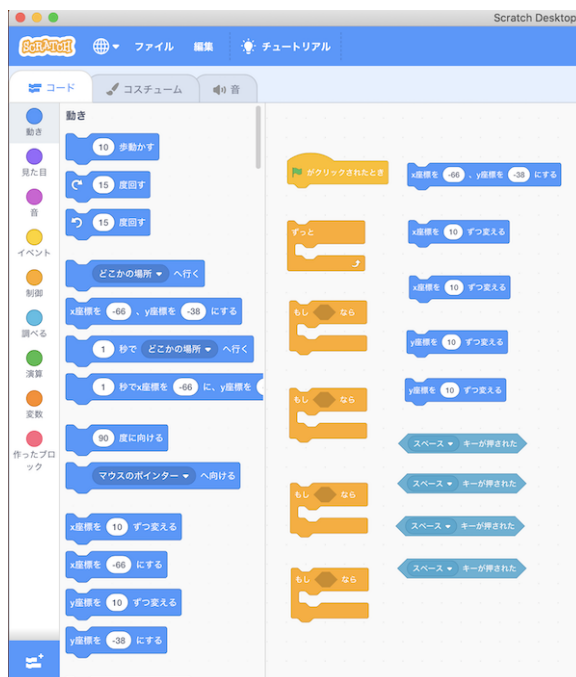
プログラムの作り方






ボウルのプログラム完成イメージ

- 以下のブロックを画面中央にドラッグ&ドロップします。






ボウルのプログラムの作り方1

-  の  に  をドラッグ&ドロップします。これを4回(ブロック数分)実行します。



ボウルのプログラムの作り方2

-  の▼を押して、表示されるメニューで『上向き矢印』を選択します。残りのブロックに『下向き矢印』、『右向き矢印』、『左向き矢印』を設定してください。



ボウルのプログラムの作り方3

- 『上向き矢印』と『下向き矢印』を設定したブロックに  をくっつけます。



ボウルのプログラムの作り方4

- 『下向き矢印』を設定したブロックの数字を **-10**に変更します。(数字をダブルクリックすることで、数字を編集できる状態になります。)



ボウルのプログラムの作り方5

- 『右向き矢印』、『左向き矢印』を設定したブロックに  をくっつけます。




ボウルのプログラムの作り方6

- 『左向き矢印』を設定したブロックの数字を **-10**に変更します。数字をダブルクリックすることで、数字を編集できる状態になります。



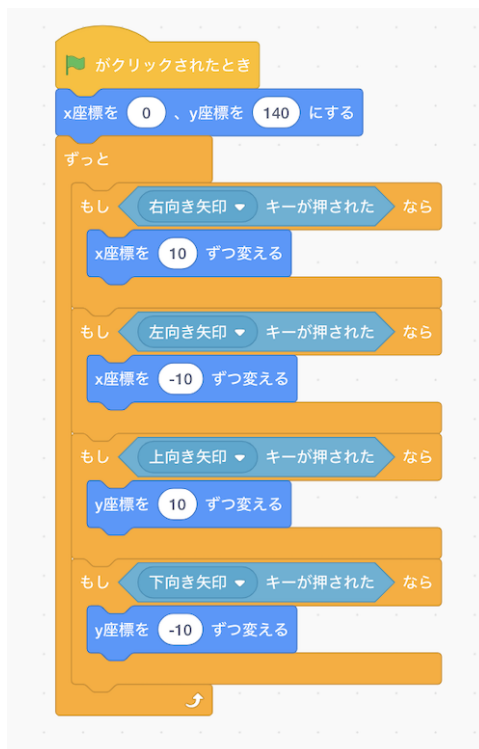
ボウルのプログラムの作り方7

-  の **x座標とy座標の数字を以下の画像の内容に変更** します。数字をダブルクリックすることで、数字を編集できる状態になります。



ボウルのプログラムの作り方8

- ブロックをくっつけてください。これでプログラムは完成です。

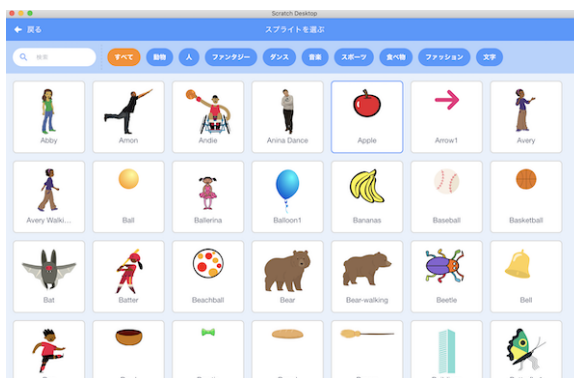


ボウルのプログラムの作り方9

2-2-2-2.リンゴのプログラムの作り方

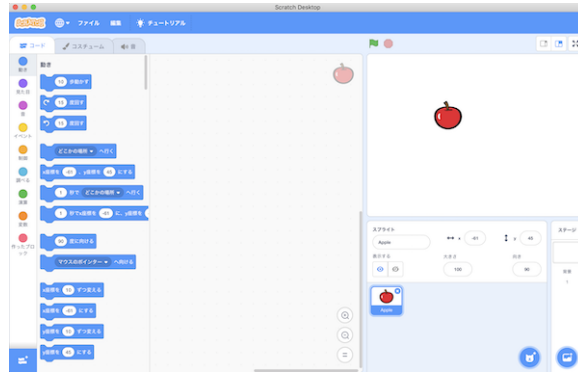
プログラムを作る前の準備

- ボタンをダブルクリックする。
- 表示される画面でリンゴ(Apple)をクリックする。



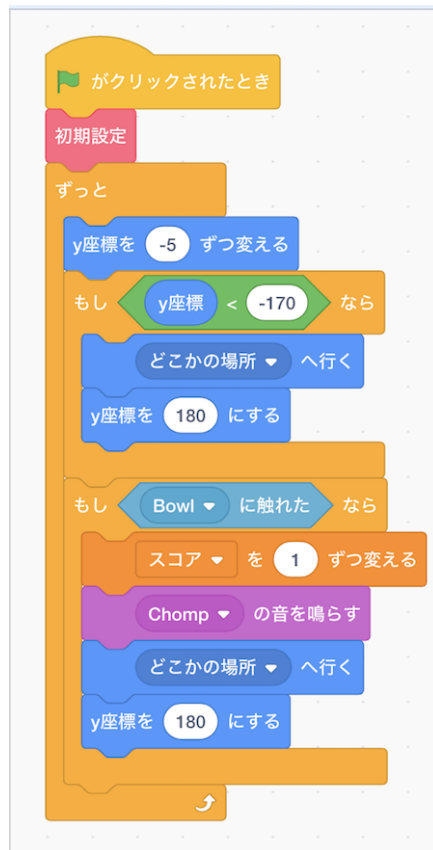
リンゴのプログラムを作る前の準備1

- スプライトが設定されていることを確認する。





リンゴのプログラムを作る前の準備2

プログラムの作り方



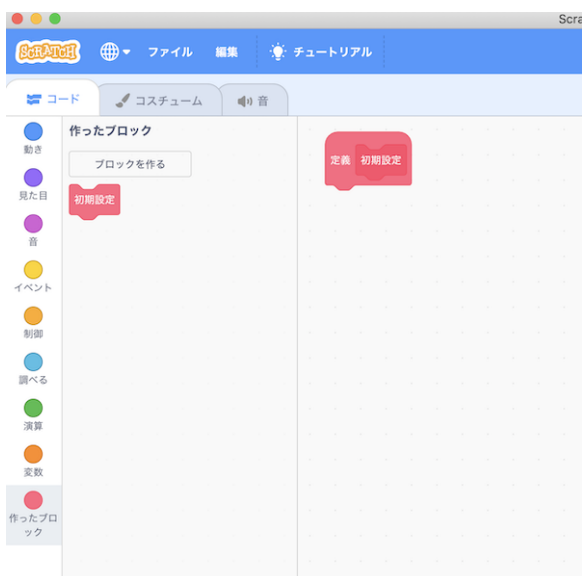
リンゴのプログラム完成イメージ

-  を押してください。
-  を押してください。
- 以下の画面が表示されるので、『**ブロック名**』を『**初期設定**』に変更してから、OKボタンを押してください。





リンゴのプログラムの作り方1

- 以下の画面が表示されることを確認してください。



リンゴのプログラムの作り方2

-  を押してください。
-  を押してください。
- 以下の画面が表示されるので、『新しい変数名:』に『スコア』と入力してから、OKボタンを押してください。



リンゴのプログラムの作り方3

- 変数『スコア』が追加されていることを確認してください。




リンゴのプログラムの作り方4

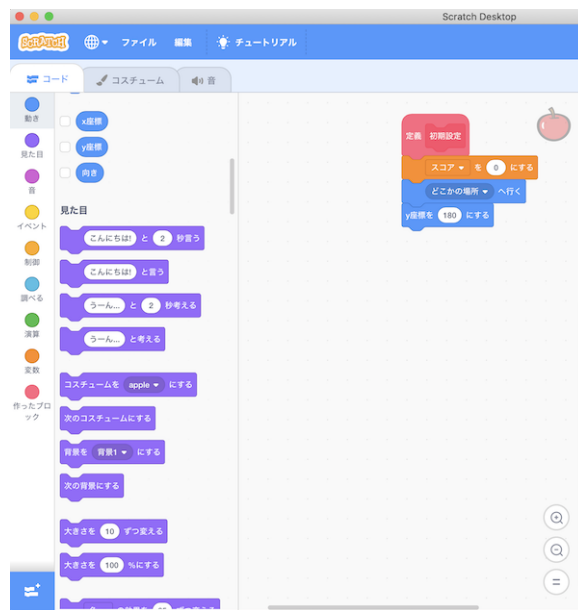
- 以下のブロックを画面中央にドラッグ&ドロップします。





リンゴのプログラムの作り方5

-  の数字を **180に変更** します。数字をダブルクリックすることで、数字を編集できる状態になります。
- 変更後、ブロックをくっつけてください。



リンゴのプログラムの作り方6

- 以下のブロックを画面中央にドラッグ&ドロップします。





リンゴのプログラムの作り方7

-  の  に  をドラッグ&ドロップします。






リンゴのプログラムの作り方8

-  の▼を押し、『マウスのポインター』から『Bowl』に変更します。



リンゴのプログラムの作り方9

-  の  に  をドラッグ&ドロップします。






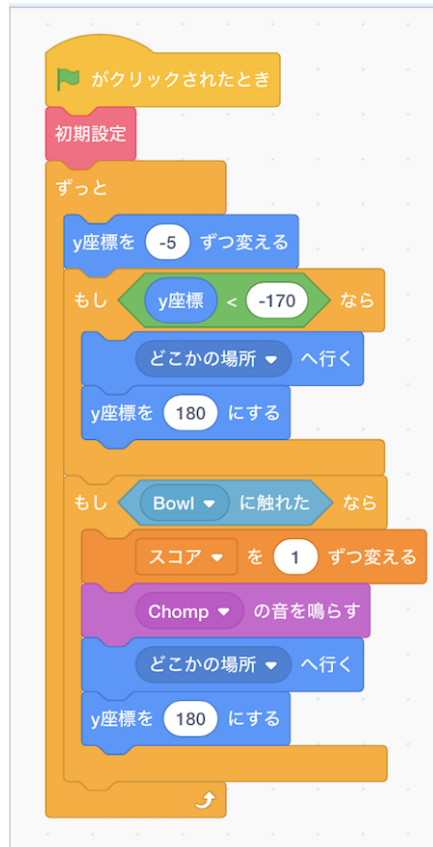
リンゴのプログラムの作り方10

-  の  に  をドラッグ&ドロップします。



リンゴのプログラムの作り方11

-  の数字を **-170に変更** します。数字をダブルクリックすることで、数字を編集できる状態になります。
-  の数字を **180に変更** します。数字をダブルクリックすることで、数字を編集できる状態になります。これを2回(ブロック数分)実行します。
-  の数字を **-5に変更** します。数字をダブルクリックすることで、数字を編集できる状態になります。
- ブロックをくっつけてください。これでプログラムは完成です。

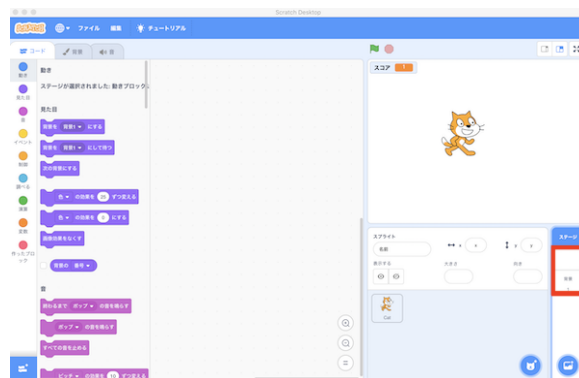


リングのプログラムの作り方12


2-2-2-3.背景のプログラムの作り方

プログラムを作る前の準備

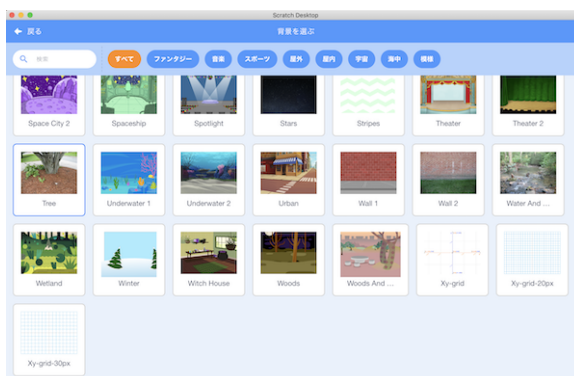
- Scratch 3.0を起動し、画面右下にある背景1をクリックします。



背景のプログラムを作る前の準備1

- 背景のタブをクリックし、を押します。

- 以下の画面が表示されるので、背景を選択します。(例：Treeを選択。)



背景のプログラムを作る前の準備2

- 選んだ背景が追加されたことを確認後、『背景1』を選択し、右上の×ボタンを押して削除します。




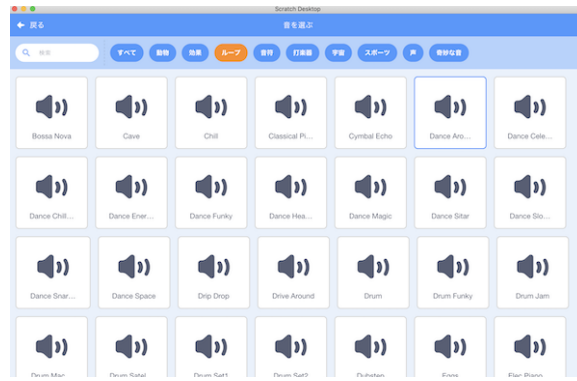
背景のプログラムを作る前の準備3

- 音のタブをクリック後、『ポップ』を選び、右上の×ボタンを押して削除します。



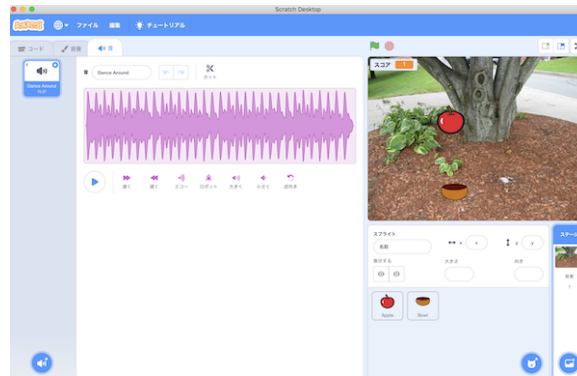
背景のプログラムを作る前の準備4

- を押すと、以下の画面が表示されるので、ループをクリックし、音楽を選択します。(例：Dance Aroundを選択)



背景のプログラムを作る前の準備5

- 以下の画面が表示されれば、準備は完了です。



背景のプログラムを作る前の準備6

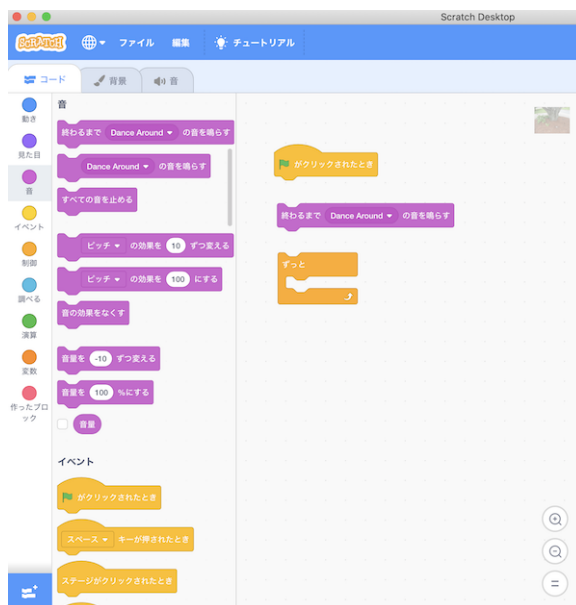
プログラムの作り方



背景のプログラム完成イメージ

- コードのタブを選択し、以下のブロックを画面中央にドラッグ&ドロップします。





背景のプログラムの作り方1

- ブロックをくっつけてください。これでプログラムは完成です。



背景のプログラムの作り方2

2-2-3. プログラムの実行・停止方法

2-2-3-1. プログラムを実行

旗のボタンを押してください。



プログラムを実行

2-2-3-2.プログラムを停止

赤いボタンを押してください。



プログラムを停止

2-3.アニメーション

ここでは『飛び魚のアニメーション』の作成方法を説明します。

2-3-1.プログラミング教育の流れと目標

アニメーションは、2019年度11月～2月に川崎市立上丸子小学校のプログラミングクラブ(4～6学年 34名が所属)で生徒が作成したものです。作成は、1人・1PCで進める方式を採用しました。

アニメーションの完成は、以下の様に行いました。

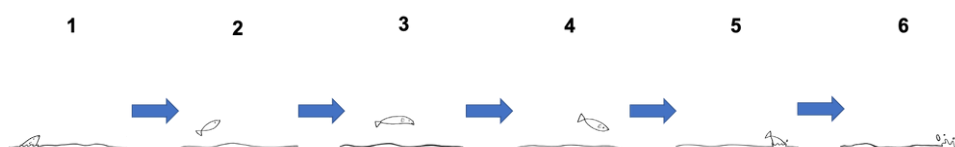
- 紙でアニメーションの素材を作成。
- 素材をフリーソフトを利用して画像ファイル化。
- 画像ファイルをScratch 3.0がインストールされたPCに保存。
- Scratch 3.0でプログラミング。

プログラミング教育の目標は、活動方針に則って以下を設定しました。

- 知識及び技能：コンピュータが日常生活の様々な場面で使われ、生活を便利にしていることを理解する
- 知識及び技能：コンピュータに意図した処理を行わせるには必要な手順があることを理解する
- 思考力、判断力、表現力等：「プログラミング的思考」の育成
- 学びに向かう力、人間性等：他者(異学年の生徒)と協働しながらねばり強くやり抜く態度の育成
- 学びに向かう力、人間性等：集団の一員として自主的・実践的な態度の育成

2-3-2.準備:アニメーションの素材を作る

- A5サイズ(148×210mm)の紙を6枚用意する。
- 1～6の順番でアニメーションするように絵を描く。

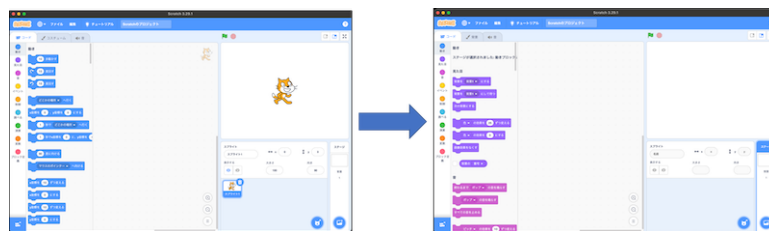


アニメーションの素材を作る

- 絵を『**Adobe Illustrator Draw**』(現在は、入手不可)と『**Adobe Capture**』を使用して画像ファイルへ変換する。
- 画像ファイルをScratch 3.0がインストールされたPCに保存する。

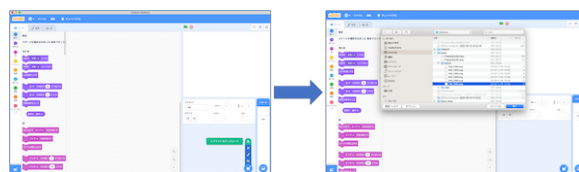
2-3-3.準備:プロジェクトの作成

- Scratch 3.0を起動し、スプライト1を削除する。(スプライト1を選択→×をクリック)



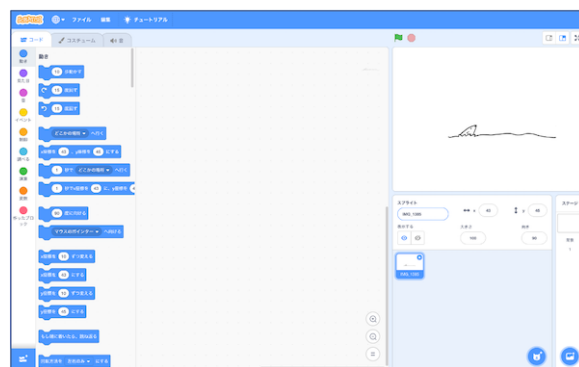
プロジェクトの作成1

- 『**スプライトをアップロード**』をクリックし、カメラアプリで撮影した画像を選択する。



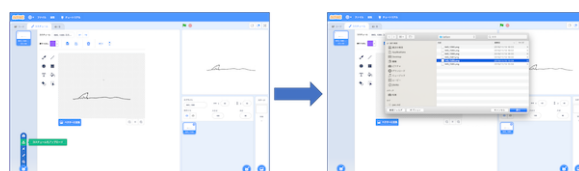
プロジェクトの作成2

- スプライトが設定されていることを確認する。



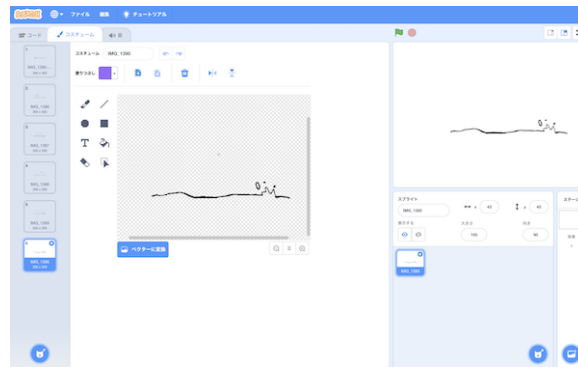
プロジェクトの作成3

- 『**コスチュームをアップロード**』をクリックし、カメラアプリで撮影した画像を選択する。準備した6枚の画像分、この手順を行う。



プロジェクトの作成4

- コスチュームが設定されていることを確認する。



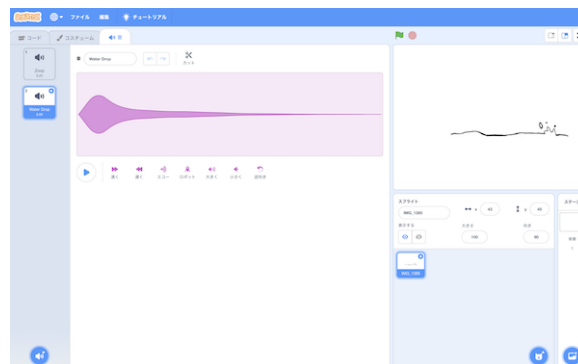
プロジェクトの作成5

- ボタンをクリックし、効果のカテゴリから、『Zoop』、『Water drop』を選択する。



プロジェクトの作成6

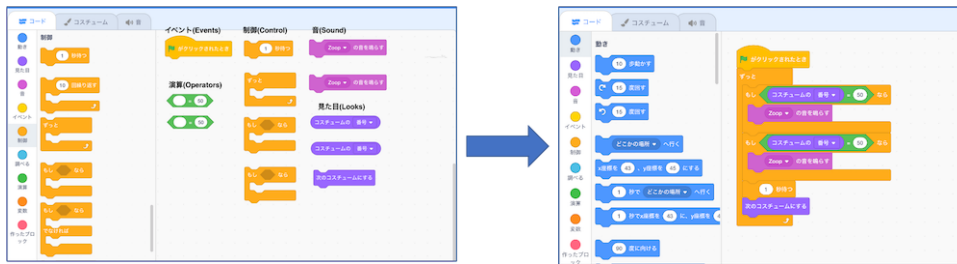
- 選んだ音が、設定されていることを確認する。



プロジェクトの作成7

2-3-4.アニメーションの作り方

- 以下の図のとおり、カテゴリ毎のブロックをドラッグ&ドロップし、ブロック同士をつなげる。



アニメーションの作り方1

- ブロックの白枠をクリックし、50から1に変更する。



アニメーションの作り方2

- ブロックの白枠をクリックし、50から5に変更する。



アニメーションの作り方3

- Zoopの右側にある▼ボタンを押し、音をWater Dropに変更する。



アニメーションの作り方4

- ブロックの白枠をクリックし、1から0.1に変更する。



アニメーションの作り方5

- 緑旗ボタンを押し、プログラムが動くことを確認する。
- 赤丸ボタンを押し、プログラムが止まることを確認する。

2-3-5.作成したアニメーションへ着色

コスチュームタブの機能を利用し、画像に色をつけることで、アニメーションがより良くなります。



作成したアニメーションへ着色

2-4.トラブルシューティング

2-4-1.学校所有のPCから外部サイトへ接続

現在(2023年時点)、学校所有のPCからScratch公式サイトへの接続は問題無くできますが、活動開始時(2018年)は接続が禁止されていました。接続する場合は、**学校側と調整(使用目的や頻度、対象となる学年と生徒数)**を行い、**学校側から市の教育委員会に申請、許可がおりると接続可能**となりました。(筆者の場合、1週間程度で許可がおりました。)

本書を読んで同様の活動を行おうとする時に『Stretch3』の様なインターネット接続を必要とするツール、それ以外のツールを使用する場合等、学校側との調整が必要なので注意してください。(計画を立てる際、調整・申請期間を加味する必要があります。)

2-4-2.プログラムの管理

複数人で1つのアプリを開発する場合、プログラムの管理はとても重要です。

2019年度5~10月に川崎市立上丸子小学校で複数人(1チーム:2~3名 × 12チーム)でゲームアプリを開発した際、「PCに保存したプログラムを誤って消してしまう。」「ブラッシュアップするつもりでプログラムを改変したけど、やはり元に戻したい。」といったトラブルが頻発しました。

大人の世界であれば、Gitの様なバージョン管理ツールを導入といった話になると思いますが、生徒が運用するにはハードルが高いので、以下2つの方法を適用しました。

- 共有サーバー上にチーム毎のディレクトリを作成し、プログラムを保存する。
- 学校の先生と協力して学校所有の外部記憶媒体にもプログラムを保存する。

実際やってみた結果ですが、『共有サーバー上にチーム毎のディレクトリを作成』は『**保存するディレクトリを間違える、誤って他の人のプログラムを消してしまう**』トラブルが頻発した為、効果がなかったです。『**学校の先生と協力して学校所有の外部記憶媒体にもプログラムを保存する。**』は、前述の問題もあり非常に効果がありました。長いスパンで生徒達にプログラミングに取り組んでもらう場合、指導を行う側がプログラムの管理を行うことで、生徒達が多くの成果を上げる事ができます。

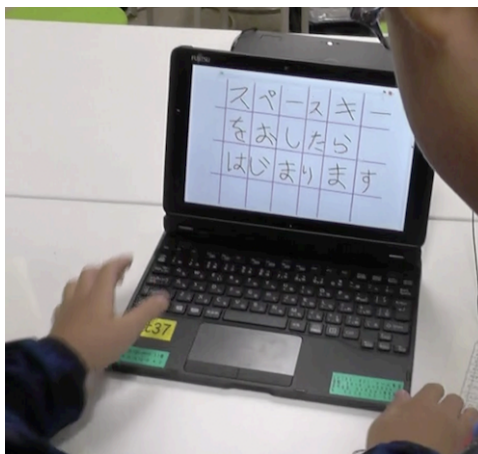
2-5.コラム

2017年度からプログラミング教育を始め、今年で7年目に突入します。

活動を始めた当初は、キーボードが打てない、英数とかなの切り替えができない、クリック、ディレクトリ・ファイルといったPCを使う上で基本的な用語も分からない生徒が大半だったのですが、現在はこういった生徒がいない状態になりました。生徒達の理解するスピード、理解したものを発展させるスピードに毎年驚かされています。

活動の過程で印象的だった事は、ゲームアプリを開発する中で4年生の生徒が以下の改造を独力で行なったことでした。

- タイトル画面・ゲームオーバー画面の追加(背景や音楽の切り替え処理)
- スコアに応じてリンゴの落下スピードを変化(徐々に早くする)
- スコアに応じてリンゴ以外のスプライト(ペイントソフトを使ってオリジナルのスプライトを作成)を落下
- リンゴ以外のスプライトにアニメーション処理を追加



コラム

その生徒が、家でインターネットを駆使してアレコレ調べたり、クラブの中で上手くいかない部分を私に質問して問題を1つずつ解決しながらプログラムを完成させていく姿を見る度、プログラミング教育をやって本当に良かったと思いました。

Scratchは『**ビジュアルプログラミングツール**』と呼ばれるものです。通常のプログラミング言語と大きく違う点は、直感的且つ簡単、作ったものを直ぐに動かして確認できる点です。プログラミング教育の現場でも生徒がブロックを繋げては動かしてを繰り返し、プログラムを少しずつ完成に近づけていく姿を何度も目にしました。こうした事から『**「情報活用能力」を構成する資質・能力の育成**』に含まれる『**「プログラミング的思考」の育成**』を自然に行える非常に優れたツールと言えると私は思っています。

今後もScratchは勿論、[Stretch3](#)や[MIT App Inventor2](#)といったビジュアルプログラミングツールも使い、プログラミング教育を通じて『**「情報活用能力」を構成する資質・能力の育成**』を行なっていきたいと思っています。

3章 プログラミング教育(Androidアプリ)

この章では1章でも記載した、上丸子小学校でのプログラミングクラブの活動で実施したプログラミング教育(Androidアプリ)について紹介します。

以下、この活動のことを「クラブ活動」と表現しています。

- [3-1.動作環境](#)
- [3-2.サンプル](#)
- [3-3.トラブルシューティング](#)
- [3-4.コラム](#)

3-1.動作環境

ここでは、プログラミング教育(Androidアプリ)で利用したハードウェア、ソフトウェアについて紹介します。

アプリ開発環境(ハードウェア)

今回のプログラミング教育で利用したハードウェアは、上丸子小学校において各生徒用に配布されたChromebookとなります。GIGAスクール構想で配布されるマシンの性能は各学校での調達時の選択により異なりますので参考値となりますが、下記の参考情報から大きな乖離はありません。

参考：[GIGAスクール構想の実現パッケージ P.3](#)

- CPU: Intel Celeronプロセッサ同等以上
- メインメモリ: 4GB
- ストレージ: 32GB

アプリ開発環境(ツール類): MIT App Inventor2

Androidアプリの開発環境についてはさまざまな選択肢があります。アプリ開発に知見のある方であれば、Android Studioを開発マシンに導入して開発を進める、という方法が思いつくかもしれませんが、しかし、小学校の活動の中でアプリを開発を行う場合、以下の制約が考えられます。

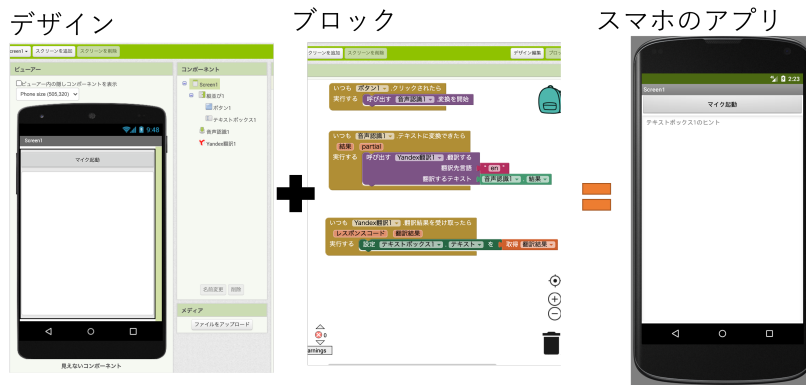
- 1.開発マシンの性能が潤沢ではない。(前項、アプリ開発環境(ハードウェア)参照)
- 2.プロの利用する開発環境は小中学生が使うには難しい。
- 3.環境を整えることについても知識を必要となり、教員にとってもハードルが高い
- 4.PCの操作に対する慣れ、熟達度が個々で差がある。

制約に対し、以下のアプローチを取りました。

- 1.と2.については、ブラウザ上で動作する軽量な開発環境を用いることとし、かつビジュアルプログラミングツールを前提とすることで、直感的にプログラミングが進められるようにする。
- 3.についてはいくつか取れる手段のうち環境構築に手間がかからない方法を選択する。
- 4.については最近では低学年の授業の中でもScratchを用いたり、算数などでChromebookを使う機会があるなど、学校生活のなかでPCに触れる機会増えつつありますが、それにつけても得意・不得意はありますので、可能な限り手での入力を減らすことを前提としました。

これらの制約の大部分を解消する方法として、[MIT App Inventor2](#) (日本語化プロジェクト版)を利用しました。

App Inventorの機能を端的に表すと下記の図の通りとなります。



App Inventorのイメージ

二つの機能についてももう少しだけ説明します。

デザイン編集

デザイン編集画面はスマホ上での見た目を調整するための機能です。ボタンの配置や色、大きさなどを編集する画面です。

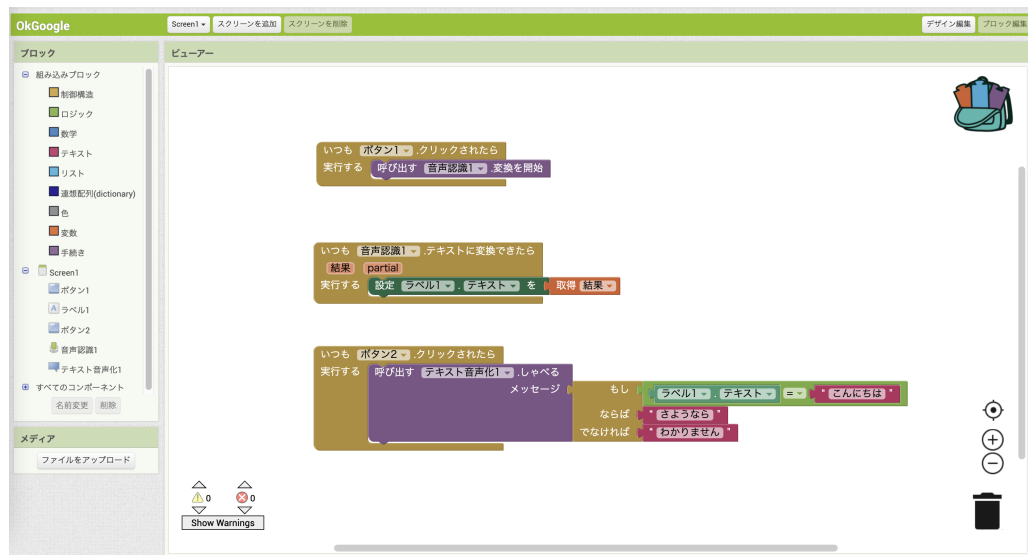


デザイン編集1

- 画面の要素は以下の通り
 - パレット: 画面に表示するボタンやテキストなど(コンポーネントと呼びます)の一覧です。パレットからビューアにドラッグ&ドロップする事でコンポーネントを追加します。
 - ビューア: 端末上でアプリを動かした時の見た目を表します。
 - コンポーネント: アプリに追加されたコンポーネントがツリー構造で表示されています。
 - メディア: アプリで画像や動画などを使う場合にここからファイルをアップロードします。
 - プロパティ: コンポーネントの大きさや色など見た目を変更する場合に使います。

ブロック編集

ブロック編集画面はアプリのボタンなどをタップした時の動きを設定するための画面です。処理の動き始めるきっかけこそ異なりますが、Scratchで旗が押された時に開始する動きを作り上げるのと同様です。



ブロック編集1

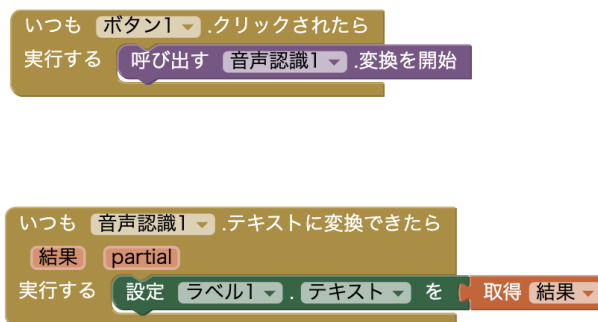
- 画面の要素は以下の通り
 - ブロック: 処理の始まりや動きを表すパーツ(ブロック)の一覧を表示します。
 - ビューア: ブロックを組み合わせることで処理を組み立てる箇所です。

例えばボタンに関するブロックについては以下のような一覧が表示されます。



ブロック編集2

このブロックをドラッグ&ドロップでビューアに配置し、ブロックの凸凹をつなげることで、さまざまな動きを作り上げる事ができます。



ブロック編集3

アプリの作成

前述二つの画面の機能を使ってアプリの見た目と動きを作り上げ、下記のメニューからAndroid端末上で動作するように仕上げます。



アプリの作成メニュー

このツールで作成したアプリはPlayStoreで配布する事も可能です。

アプリ動作環境

アプリを作成した後は、それが実際に動く様を見る事で子供たちはより楽しみを感じるものです。前述の通り、開発マシンの性能が潤沢ではないためアプリを動かす環境として開発環境を使うことが難しいこと、また実際に実機で動いていることの方がより面白みが増すであろう、などの理由により、アプリの動作確認はAndroid端末実機を利用することにしました。

実機は全て当社であらかじめ保有していたものです。あくまでも今回使ったものですので、MIT AI2 Companionアプリが動けばバージョンが古くても概ね問題はないと思われます。

- ハードウェアおよびスペック
 - YOGA TAB 3 PLUS
 - OS 6.0.1
 - RAM 3GB
 - Storage 32GB
- その他必要な事前準備・環境
 - [MIT AI2 Companionアプリ](#)のインストール
 - 2023/4時点で最新バージョンは2.66 (2023/2/16更新)
 - ブラウザ
 - Android端末は通常Chromeがインストールされている
 - インターネット接続可能な環境の準備
 - モバイルWifiルーター
 - ルーターへの接続設定
 - カメラ
 - QRコードの読み取りを実施するため
 - ※この他、端末が作成したいアプリで利用する機能を備えていることが必要です。

3-2. サンプル

ここではサンプルアプリの作成を例に、プログラミング教育の中での実際の取り組み方について示します。

作成するアプリ

今回はスマホの音声認識機能を使ったアプリを作成するまでの流れを示します。

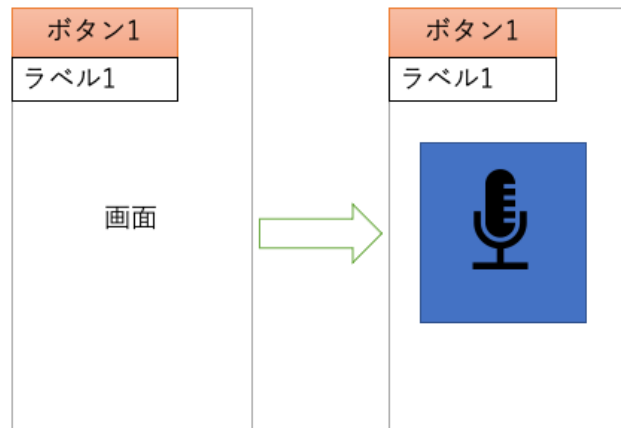
- 画面のイメージ



画面イメージ

- 画面の構成

画面に表示される要素を図にすると以下の通りとなります。



画面構成

- アプリの動作
 - ボタン1を押すとスマホのマイクの機能が起動する
 - マイクに何かを話すと話した言葉がラベル1に表示される

アプリのレベルに関する考え方

アプリの作成については以下の順番で段階ごとに複雑度を上げていくことで、無理なく進めていくことが可能です。今回は下記のうち4.までを想定した記載となります。

- 1.画面に要素を加えて表示する
- 2.画面の要素の並び、色味などをこだわって表示する
- 3.ボタンをクリックすることで何かが発生する
- 4.スマートフォンの機能を1つ使う
- 5.スマートフォンの機能を2つ使う
- 6.スマートフォンの機能を組み合わせて使う
- 7.複雑なロジックを加える

全体的な流れ

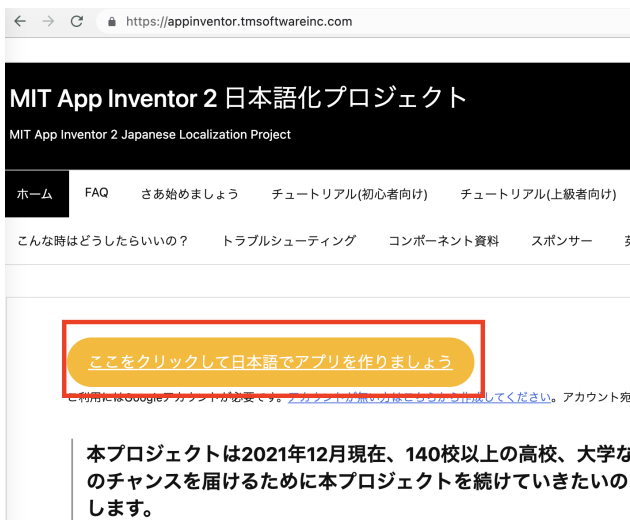
今回クラブ活動の中では、下記の流れでアプリの作成を進めました。

- App Inventorへのログイン
- 新規プロジェクトの作成
- App Inventorの画面説明
- デザイン編集でアプリの画面作成
- ブロック編集でアプリの動き作成
- アプリの実行

App Inventorへのログイン

- 1.ブラウザで、MIT App Inventor 2 日本語化プロジェクトのサイト (<https://appinventor.tmssoftwareinc.com/>)を開く

- 2.下記の黄色ボタン「ここをクリックして日本語でアプリを作りましょう」をクリックする



- 3. Google認証のページが表示されるので、あらかじめ用意したアカウントでログインを実施する。(あらかじめ学校側で対応しておくともスムーズかもしれません。)



- 4. 初めてログインする場合、以下の通りの画面が出てくるので、「許可する」を選んでください。



- 5. 「MIT App Inventor 日本語」のページが表示されます。

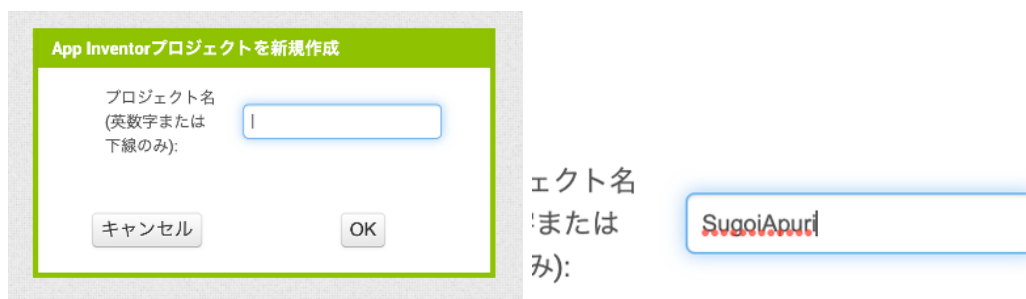
新規プロジェクトの作成

- 1.画面のメニューから「プロジェクト」をクリックし、その中の「プロジェクトを新規作成」をクリックします。



プロジェクトの新規作成1

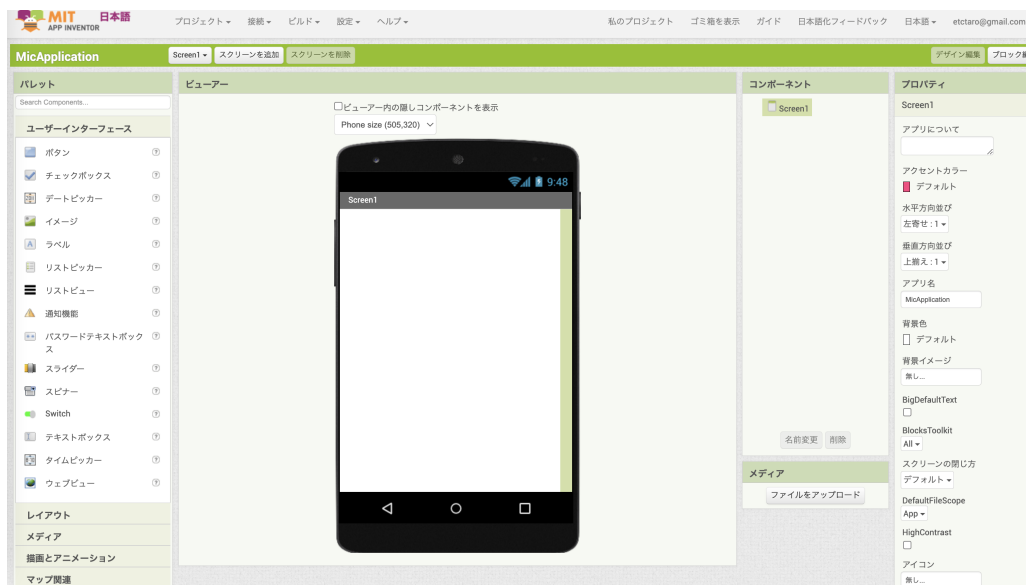
- 2.下記のダイアログが表示されますので、英数字と でこれから作成するアプリに名前をつけてOKをクリックします。**全角文字（ひらがな、カタカナ、漢字）はつかえません。**



※PCが気付かぬうちに全角モードになっている場合がありますので、児童が自由に設定する場合はご注意ください。

App Inventorの画面説明

新規にプロジェクトを作成すると、以下の通り新しいアプリの画面が表示されます。



App Inventorの画面

ここでデザイン編集の画面、ブロック編集の画面、アプリの動作についてそれぞれ大まかな機能について紹介すると、この後の進め方がイメージつきやすくなると思われます。

詳細について、[動作環境](#)の節を参照してください。

デザイン編集でアプリの画面作成

ここから実際のアプリの作成について触れていきます。デザイン編集の画面では、アプリを端末で動かした時に表示される見た目を作り上げていきます。

今回は、スマホの機能を使うことを前提として、見た目の複雑さには拘りません。

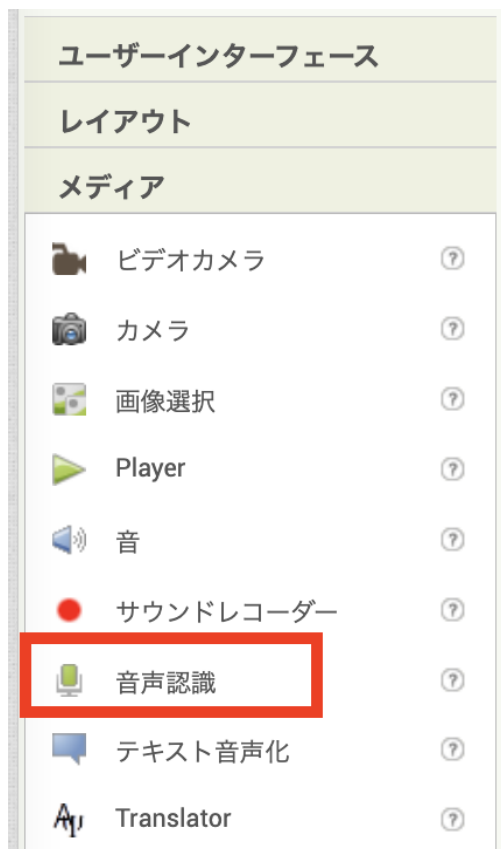
- 1.パレットからビューアのスマホの画面にボタンとラベルをドラッグ&ドロップする 以下のように左から右にドラッグ&ドロップして必要な要素（コンポーネント）を追加することができます。



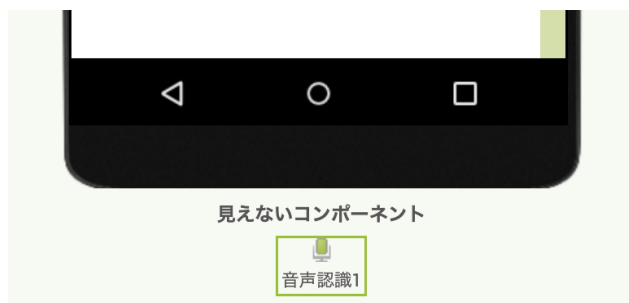
以下のようになります。それぞれボタン1、ラベル1という名前で追加されました。



- 2. 音声認識の機能を追加する パレットのうち、メディアの中に「音声認識」があります。これもビューアにドラッグ&ドロップします。



画面上で見えないですが、以下の通り、「見えないコンポーネント」として追加されています。これで、このアプリから音声認識の機能を使う準備が整ったということになります。



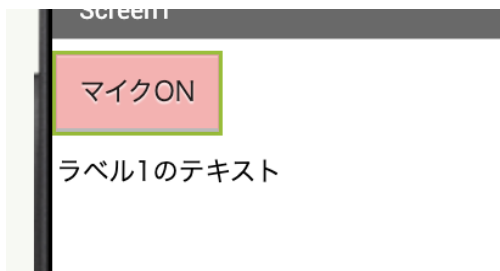
- 3. ボタンの見た目調整 ビューアのボタン1を選択すると右側にプロパティが表示されます。これはボタンに関する見た目を設定するための情報です。



例えば、背景色をピンク、テキストを「マイクON」とします。



ボタンの動きが想像しやすくなりました。



ブロック編集でアプリの動き作成

さて、見た目が完成したら、次はアプリを操作した時の動きを作ります。ブロックを見ていただくと下記のようになっています。



ブロックの一覧

この中でScreen1となっているブロックは、今回追加したコンポーネントに関するブロックです。例えばボタン1をクリックするとこのように表示されます。



ボタン関連のブロック

- 黄色いブロックは何かの動きを始めるきっかけのブロック。
- 濃い緑は見た目などに何かの変更をするためのブロック。
- 薄い緑は変更をする時の変更後の値を表すブロックとなります。

他にもブロックはありますが、これらのブロックの凸凹をつながるようにつなげることで、アプリの動作を組み立てていきます。

- 1. ボタンをクリックした時の動きを作る

- ボタンをクリックしたことをきっかけとした動きを作る場合は、そのボタンに関するブロックを使います。



- ボタン1のブロックのうち、「いつも ボタン1 .クリックされたら」というブロックを使います。



- 2. ボタンをクリックしたら音声認識の機能(マイク)を起動する
 - 音声認識に関する動きを作りたいので、音声認識1のブロックを使います。



- 音声認識の機能のうち、マイクを起動する場合は、「呼び出す 音声認識1 .変換を開始」というブロックを使います。

呼び出す 音声認識1 .変換を開始

- 二つのブロックを組み合わせます。

いつも ボタン1 .クリックされたら
 実行する 呼び出す 音声認識1 .変換を開始

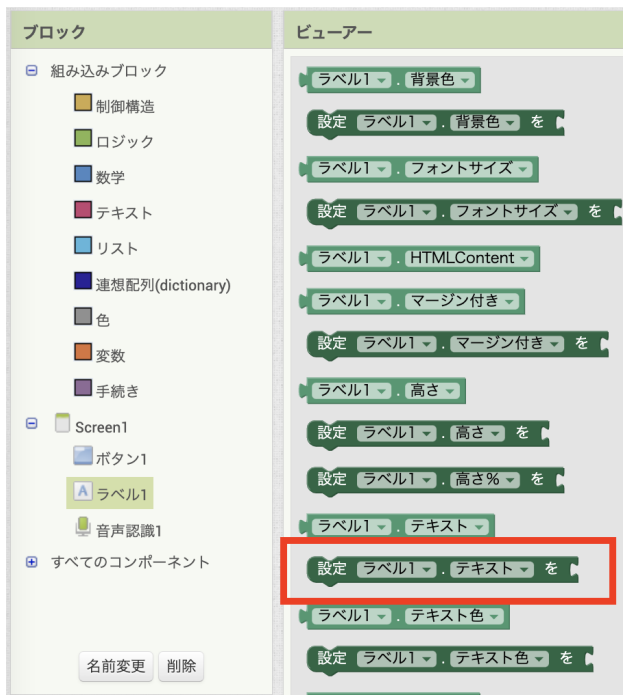
3. マイクで聞き取った結果を使う

- スマホの音声認識機能では聞き取った結果をテキストに変換しますが、その結果が返却されるまで時間がかかる可能性があります。
- このようにスマホの機能を使う場合は『スマホの機能を使って変換ができれば、動きを始める』ためのきっかけとなるブロックが必要です。
- 音声認識1のブロックのうち「いつも 音声認識1 .テキストに変換できたら」というブロックを使います。

いつも 音声認識1 .テキストに変換できたら
 結果 partial
 実行する

4. 音声認識の結果を画面に表示する

- 音声認識の結果を表示したいので、ラベル1に関するブロックを使います。



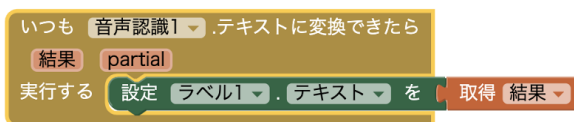
- ラベル1のなかから「設定 ラベル1 .テキスト を」というブロックを使います。



- これだけだと右側が不足しており、×印がつきます。変換結果を入れたいところです。
- 「いつも 音声認識1 .テキストに変換できたら」ブロックの「結果」という部分をクリックすると以下のようなブロックが表示されます。この「取得 結果」ブロックを使います。



- 全てのブロックを組み合わせます。



これでアプリとしては完成です。

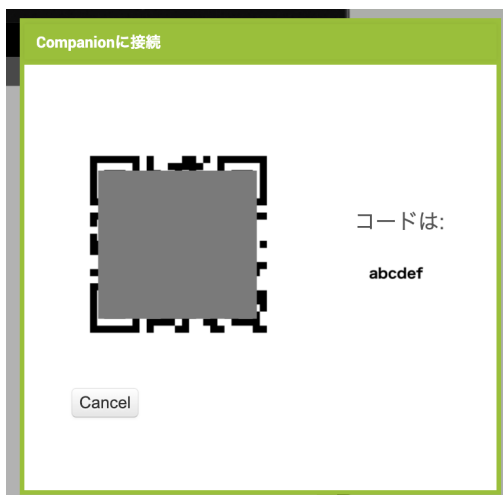
アプリの実行

作成したアプリを端末で実行するための方法について、App Inventorではいくつか方法がありますが、以下の流れでアプリを動かします。

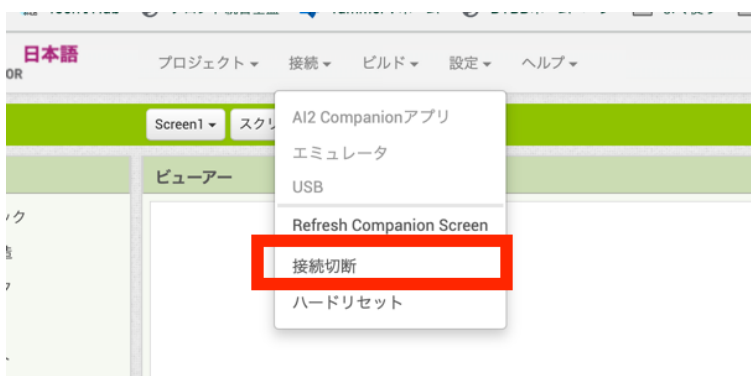
- 1. 接続 > AI2 Companionアプリ をクリックする



- 2. QRコードが表示されるので、これを端末の AI2 Companion アプリで読み取る



- 3.少し時間を経過した後、端末上でアプリが起動する。
- 4.動作確認が終わった後、接続切断を実行し、アプリを終了する。



- 5.他の人に端末を渡す。(端末を複数人で使い回す場合。)

この方法は短時間でアプリが実行できると共に、複数台の端末でそれぞれアプリを実行する場合、この方法では台数の制限がない（または制限が緩い）ため本書では実行手段として記載を行いました。

ただし、ネットワークの設定（ネットワーク機器の設定によるアクセス制限など）によってはこの手段は取れない場合がありますので、事前に動作確認した上で対応した方が良いでしょう。

3-3.トラブルシューティング

以下、Androidアプリの開発を進める上で生じたトラブルとその解消について記載します。

App Inventor2のサイトへアクセスできない

生徒が利用するPC(今回はChromebook)は安全のため接続可能なURLが制限されています。App Inventor2もこの制約により初期状態ではアクセスができませんので対応が必要となります。

学校側の対応として以下を実施しました。

- App InventorのURLのアクセスを許可する。
- 具体的には http://ai2.tmssoftwareinc.com/login/google?locale=ja_JP

なお、教員に配布されたPCの場合制限がない場合もありますので、必ず生徒の利用するPCで確認してください。

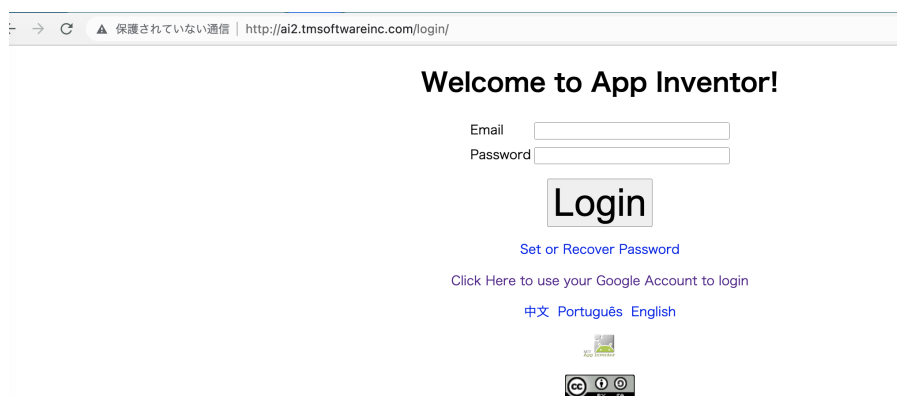
App Inventor2で利用するGoogleアカウントの準備

App Inventor2ではGoogleアカウントを利用します。アカウントの準備については、開発で利用する環境の状況により異なりますが、当社が直面したパターンは以下の通りです。

- パターン1: 開発マシンが生徒個人に紐づいていない
 - 例えばクラブ活動用にPCが用意されており、生徒が好きにとっていく場合です。
 - このパターンの場合、連続して同じマシンを使うとは限らないため1回きりのコンテンツとします。
 - Googleアカウントは当社側で用意しました。
 - クラブ活動の前にあらかじめ設定しておくなどの時間的な余裕があります。
- パターン2: 開発マシンが生徒個人に紐づくものである
 - GIGAスクール構想によりそれぞれの生徒にPCが配布されている場合が相当します。
 - 生徒個人用にGoogleアカウントを学校側で用意していただきました。
 - 複数回続けて開催できるコンテンツとすることが可能です。

App Inventor2でセッションが切れた場合の対応

一度App Inventorにログインしてからしばらく経ってからアプリの更新などを行うと、下記の画面が表示されます。これは、セッションが切れたという状態です。



セッションが切れた時の画面

このまま再びApp Inventorにログインすることも可能ですが、この場合日本語化プロジェクトのページが表示されず、英語などになります。

この画面が表示された場合は、再度日本語化プロジェクトのトップ画面(<https://appinventor.tmssoftwareinc.com/>)を開き直してください。

また、トラブルを防ぐため、授業やクラブ活動は毎回トップ画面から始めるようにすると良いでしょう。

複数台の端末を同時に利用したところインターネットアクセスができなくなる

モバイルWifiルーターを利用する場合、1つのルーターに対してアクセス可能な台数が制限される場合があります。今回は学校側でいくつかモバイルWifiルーターを保有していたため、一部の端末についてはそちらにアクセスする形で問題を回避しました。

ただし、一台ずつアクセス設定を行うことになるので、その手間を可能な限り減らせることが望ましいです。最近QRコードを使いルータに接続する設定を加えることも可能であるため、もし端末が対応している場合はそちらも利用すると接続がスムーズになります。

App Inventor2でアプリを実行するのに時間がかかる

App Inventor2を端末で動かす方法はいくつかあります。

- 方法1: デバッグ機能を利用する

[前節](#)で説明したアプリ実行方法です。方法2より迅速にアプリの実行が可能です。この方法の特徴は以下の通りです。

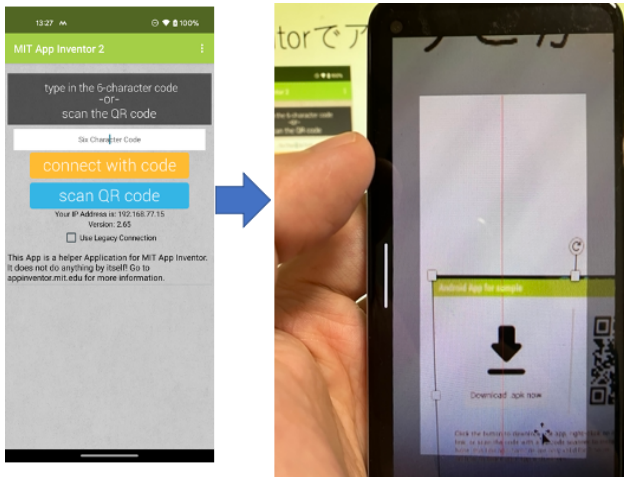
- 端末上でアプリ実行中にデザイン編集やブロック編集で変更を加えるとその変更が即座に反映される。
- アプリの実行までに要する時間が方法2より短い。
- 同時に実行可能な数の上限が方法2より緩い。このため、複数台の端末で同時に実行しやすい。
- ネットワーク環境の状況によっては利用できない場合がある。

- 方法2: apkを作成する

- (1)apkを作成



○ (2)インストール



実際のアプリ開発でも使われるapkの形式でアプリを作成し、それを端末にインストールする方法です。インストールにMIT AI2 Companionアプリを使います。この方法の特徴は以下の通りです。

- apkの作成が時間を要し、かつ同時に実行可能な数に制限がある。
- インターネットに接続可能な環境であればおおむね問題なく実施できる。

特に1点目の同時に実行可能な数の制約が強いため、ある程度の数の児童が相手となった場合、そこで待ちが発生します。当初はこちらの方法を取っていたものの、この待ちが許容できなくなったため、方法1を取りました。

3-4.コラム

本節では、クラブ活動でのプログラミング教育を実施した際に補足で説明した内容などについて記載します。ぜひ、プログラミング教育を実施する際に提供する話題として加えていただければと思います。

近年の児童のスマホと接する機会について

クラブ活動の初回に児童に対して、「家庭、または自分でスマホを持っていたり触ったりすることがあるか」というアンケートを取りました。

年度により多少の差はありますが、ほぼ全員が接する機会があるという回答をいただいています。

かく言う私の娘も幼稚園の頃からiPadを使いこなすなど、私自身が子供の頃からは想像もつかないほどに、スマートフォンというものは身近になっていると言えます。

とはいえ、スマートフォン上で動いているアプリはどこか自分の知らないところで大人が時間をかけて作っている、そのような浮世離れた印象をおそらくは持っているのだと思います。

そのスマートフォン上で動いているアプリを自分の手で作ることができる、という体験は、児童にとって強い興味を持てる材料だと言えます。

このようなアンケートと合わせて、「普段スマホで遊んでいるゲーム」、「知っているスマホの機能」など、スマホに関する質問も続けると、素早く反応が返ってくることも印象的でした。

活動の雰囲気づくりにもちょうど良いと思いますので、参考にしてみてください。

アプリをわかりやすくすることについて

App Inventorでアプリを作る場合、デザイン編集の機能で何も手を加えないと以下ようになります。



初期状態

この見た目のままブロック編集でボタンをタップした時の動きを作ることも可能ですが、ITエンジニアの仕事という観点で以下の説明を加えました。

- アプリを動かしてこのような見た目だとボタンを押して良いかわからない、使いづらい。
- 使いづらいと何が起るかというと、ユーザーさんはアプリの評価を下げてしまう。
- アプリの評価が下がるとアプリがダウンロードされなくなる。
- そのようになってしまうと、お仕事がなくなって困ってしまう。
- そのようにならないように、アプリのわかりやすさのために工夫をします。

たったのワンポイントですが、例えば以下のように文字を変えるだけでもわかりやすくなる、または押したくなる、という説明ができます。



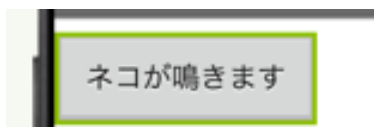
テキスト変更後

★初期状態だとどのような点に問題があるか、というテーマでディスカッションをするのも良いでしょう。

ボタンの色を変えるワンポイント

一つ前の項と同時に、色についての説明を加えることもできます。色から伝わる印象と、押したくなる、押すのに躊躇するボタンという観点で以下の3種類のボタンを紹介しました。

- 色は特に変えないが文字でボタンの機能を説明する例



デフォルトの色

- 押しやすくなる優しい色を使う例



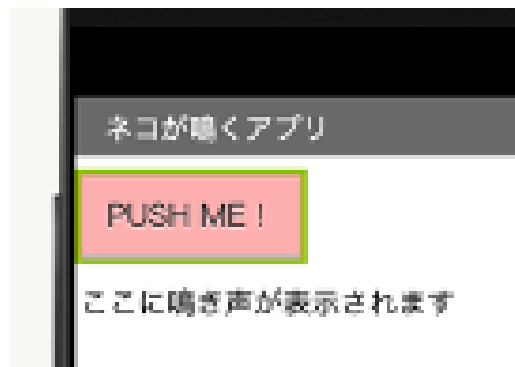
優しい色

- 文字は安全そうだが、危険そうな印象を持つ色を使う例



危険そうな色

適切にアプリに説明をすることで分かりやすいアプリという印象を持たせることが可能です。



アプリのタイトルとの組み合わせ

★色や文字は子供たちの独自性が特に発揮されやすいポイントですので、自由時間を挟み、それぞれの内容を発表するタイミングを設けると良いでしょう。

★色という話からD&I(ダイバシティー&インクルージョン)の観点で、自分の思っている色とは違う色として感じる人がいる、と言った話題にも広げられる可能性があります。

Androidのセキュリティについて

Android端末で動作するアプリでは、悪用されると危険な機能を使う場合、以下のようにダイアログが表示されます。(例：マイクで入力した音声を使う場合)



許可のリクエスト

ユーザがこのようなダイアログの内容を確認し、「端末の機能を使っても良い」という許可を与えなければ、アプリは機能を使うことはできません。

このように、安全性に関わる機能があることを示しながら、例えば以下のようなことを説明に盛り込むことができます。

- スマートフォンは便利だが、危険とも隣り合わせであること
- 危険に対する対策としてハードウェアやソフトウェアの開発者はさまざまな工夫をしていること

★例えば、スマホと隣り合う危険についてのディスカッションを行うなど、情報リテラシの教育に話題を広げられる可能性があります。

4章 キャリア教育

この章ではキャリア教育について紹介します。

- [4-1.活動の目的](#)
- [4-2.コンテンツ例](#)
- [4-3.コラム](#)

4-1.活動の目的

キャリア教育は、「ICT業界の魅力を子供たちに伝える」ことをテーマとして、活動しています。小中学校や高校・大学などを訪問し、ITエンジニアの仕事の内容を紹介しています。社会で働くことの意義ややりがいを伝えることで、子どもたちに自分の将来を考えるヒントにしてもらっています。対面での開催に加え、オンラインでも実施しています。写真は、中学校を訪問させていただいたときの授業の様子です。



キャリア教育の様子

4-2.コンテンツ例

本項では、キャリア教育として、ITエンジニアの仕事を紹介するときに使っているコンテンツを紹介します。対象とする学校や生徒の皆さんの学年や、興味・関心によって、伝える内容を変えています。今回は、中学1年の学生さんを対象として実施したときの内容を例として紹介します。また、**単純にコンテンツの内容を紹介するだけでなく、これら内容をなぜ含めているのか？** その背景についても触れます。

「お互いの」自己紹介

キャリア教育の対象となる学生の皆さんとは、初対面のケースが多いでしょう。当然、学生から見ても自分たちは初対面の大人です。学校の先生経由で、子供たちはキャリア教育の前に「どんな人が来るんだろう?」「話を理解できるか不安だけど楽しみ」「緊張してちゃんと話せるかどうか心配」など、楽しみな気持ちと不安な気持ちの両方を抱えているという話をよく聞きます。自分が子供だったときのことを振り返ってみると、(キャリア教育を受ける機会がそもそもなかったのですが)初対面の大人と話をするときには、同じような気持ちを持っていたような気がします。そこで **お互いの自己紹介を行う(教育の時間に対して、生徒の人数が多いなど全員に自己紹介をしてもらうのが難しいケースは除く)** 時間を設けるようにしています。また、ここでは、あえて仕事の内容には触れずに、それ以外の内容を話すように心がけています。

自己紹介

- 名前：神原 健一(かんばら けんいち)
- 趣味：ランニング、英会話、旅行
- 部活：卓球部(小中)
児童文化部・ソフトボール(大学)
- 職場：みなとみらい(横浜)・自宅

Step1_1

みなさんの自己紹介

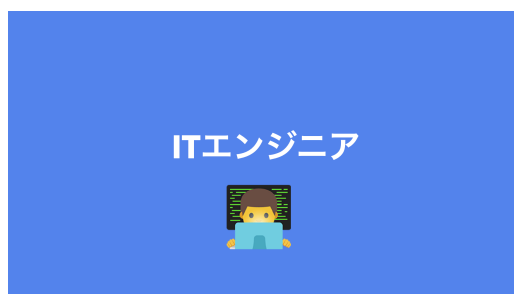
- 名前
- 部活動
- 最近、夢中になっていること(あれば)

Step1_2

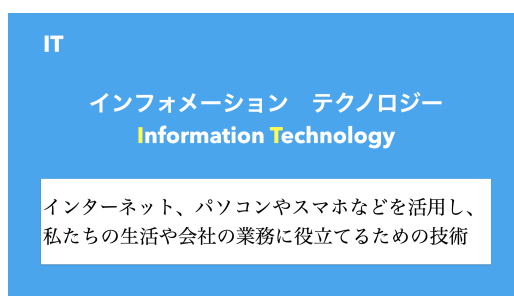
これにより、お互いの距離感が少なからず縮まるため、相互に理解し合うための **アイスブレイク** になります。

初めは「極力シンプルに」職業紹介

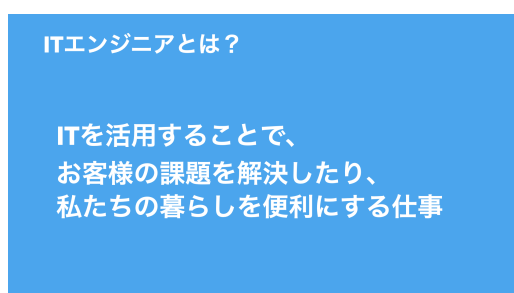
学生の中には、この仕事の内容について既にある程度知っている人がいる一方、初めて聞く人もいます。ITエンジニアとして取り組んでいることを詳しく伝えようとしても、なかなか伝わらないことが多い印象があります。そこで、前提の知識が全くなくても、どんな仕事なのか？を想像できるように、**理解してもらいやすそうな内容を選び、そして、難しい言葉はなるべく使わずに(どうしても使う時は必ず補足説明)、極力シンプルに分かりやすく** を心がけています。



Step2_1



Step2_2



Step2_3

これらに加え、**特に注力しているのが、ITエンジニアとして自分が実際に企画や開発に携わってきたサービスやプログラムを紹介すること**です。というのも、実際に自分が関わった内容を紹介することで学生の心に響き、仕事の取り組みをより具体的に理解してもらうことができます。ただ、守秘義務遵守の観点から、話す内容は既に公知となっている内容から選んでいます。

「工作中的1日の過ごし方や職場」の紹介

仕事の細かな説明に入る前に、働いている日はどんな1日を過ごしているかを紹介しています。コロナ禍以降、ITエンジニアとしてリモートワークする頻度も増えました。そこで、**世の中の仕事には、職場は会社だけでなく、自宅から仕事することがあることも、ありのままの実態として**合わせて伝えています。



Step3_1

先生から、**学生は学校の中だけに視界が限定されていることが多く、社会のことはまさに勉強している最中で、分かっていないことも多い**とよく耳にします。当然ながら、自分の過去を振り返っても、そうだったなあと思います。

そこで、職場の雰囲気をできるだけ伝えられるよう、職場の建物の中や、会社の仲間が働いている様子の写真も一緒に紹介しています。こうすることで、社会人が働いている様子を少しでも、より具体的にイメージしてもらうことができます。

ITエンジニアの仕事紹介(前編)

ここから、ITエンジニアの仕事の中身についての紹介に入っていきます。**一言にITエンジニアと言っても実に様々な業務**があります。

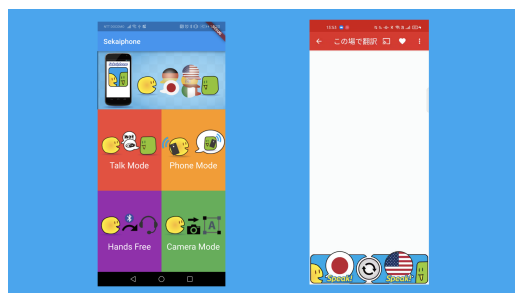
ITエンジニアのお仕事(例)

1. アプリ開発
2. 研修講師/ブログ執筆
3. 講演
4. CSR活動

Step4_1



Step4_2

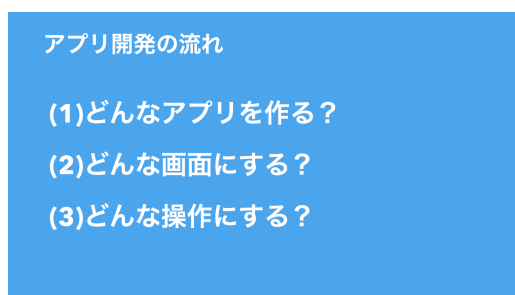


Step4_3

まずは、そのことに触れつつ、自分が中心的に取り組んでいる、スマホアプリ開発に重きを置き、掘り下げて紹介しています。

ITエンジニアの仕事紹介(後編)

ここからは、スマホアプリ開発の具体的な流れについて、紹介していきます。スマホやアプリについて、見たり聞いたりしたことがある学生は一定数いる一方で、1つのアプリが完成するまでにどんなステップがあるかまで知っている人は多くない状況です。そこでまずはその流れを簡単に説明しています。



Step5_1

(1)どんなアプリを作る？

- ある言語(例.日本語)で話すと、別の言語(例.英語)に翻訳して音声で読み上げる。
- 15ヶ国語(日本語、英語...)に対応する。
- Android・iPhoneをサポートする。

など

Step5_2

(2)どんな画面にする？



Step5_3

(3)どんな操作にする？




Step5_4

その後、アプリ開発の仕事について、掘り下げて紹介していきます。学生の反応や理解状況を見ながら、予備で準備しているスライドを見せることもあります。

こんなプログラムを書いてるよ

```
// 左上のボタンが押されたときの処理
aloneButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // 翻訳画面への移動
        Intent intent = new Intent(getActivity()
            .getApplicationContext(), PhoneScreen.class);
        startActivity(intent);
    }
});
```



Step5_5

例えば、もう少し踏み込んだ内容を伝えた方が理解が進みそうと感じたときは、ソースコードの断片を紹介することもあります(この場では、細かな処理を説明することはない、呪文のような何かを書いているということまで伝われば十分だと考えています)。

「気になったことを何でも聞いてもらう」質問タイム

学生は、自分の家族以外から仕事の話や生活を直接教わる機会が多くないと聞きます。そこで必ずQ&Aの時間を長めにとって、気になったことを何でも聞いてもらうようにしています。



Step6_1

ただ、積極的に質問の手が挙がることもあれば、そうでないことも多々あります。例えば、仲良くなった学生に授業のあとで聞いてみると、「色々聞いてみたかったけど、質問する勇気が出なくて手を挙げられなかった」という話を聞いたこともあります。

質問例

- 日々大切にしていること
- 一番大変だったこと
- 一番やりがいを感じたこと
- 休みの過ごし方
- 他に気になること何でも!!!

Step6_2

そこで、できるだけ、質問してもらいやすくすべく、**事前に、学生のみんがこんなこと気になるかな？**と思う質問例を準備しておいて、反応を見ながら、例えば、「こんなことが気になる人はいますか？」と促しています。

4-3.コラム

これまでキャリア教育を実施してきて、学生の皆さんがITエンジニアの仕事の内容以外に、特に興味を持ってくれたことを紹介します。まず、「なぜ、この仕事を選んだか？」です。どんな仕事を選ぶか？その理由は人それぞれですが、学生は仕事について、「周りの大人がなぜその仕事を選んだのか？」関心を持っている人が多そうです。

今の仕事に就いた理由

- ITが純粋に好き
- 世の中に貢献できる
- 話したり教えることが好き

仕事に就いた理由

そこで、あくまで自分の場合はという前提を置いた上で、その背景を伝えています。

最後に自分がキャリア教育を締めくくる時に、話している内容を紹介します。

みなさんへのメッセージ

- 挑戦は、人生を豊かにします。
- 学びは、人生の選択肢を増やします。
- 毎日を大切に過ごしてください！

みなさんへのメッセージ

実はこれら3つは、**学生から頻繁に聞かれてきた内容に対する「自分なりの答え」**だったりします。自分が子供の頃を振り返ってみると、まさにこれらについて、同じ疑問を持っていました。このメッセージには、3つの背景があります。

- (1)なぜ苦勞してまで挑戦する意味があるのか？
- (2)(一見すると将来、役に立たないかもしれなようなことを)なぜ勉強するのか？
- (3)何気ないと思える1日がどれだけ尊いものか？

この背景に対する答えが先に紹介したメッセージです。このようにキャリア教育を通じて、学生の皆さんが疑問に感じている(過去の自分が疑問に思っていた)ことを、ほんの少しだけ(?)長く生きている人間のあくまで1意見としてお話ししています。

おわりに

「みんなではじめる 小中学生のためのITエンジニア体験教育」を最後までお読みいただき、本当にありがとうございます。本書の企画が始まったのは、2022年の後半でした。今回も様々な企画案を出しつつ検討を行った結果、私たちが教育現場で実践している「ITエンジニア体験教育」をテーマにすることが決定しました。

2022年には今回本にしたプログラミング教育、キャリア教育の機会があり、執筆メンバーで[日本教育工学協会\(JAET\)](#)の全国大会に登壇したこともありました。

プログラミング教育の必修化など、教育現場を取り巻く環境は年々変化していますが、一方で教員がトレンドを取り入れて教育を実施する、という点についてはまだまだ発展途上にあると考えています。

そのような課題認識のもと、解決に導くための一つの礎となれば、ということで書き進めたのが本書となります。

本書を読んで頂いた後、面白い！役に立つ！！と感じていただけたなら、是非身近な他の人にも紹介していただけたら嬉しいです。誤りの指摘、こんな本も読んでみたい等のご要望も是非お寄せいただければと思います。

本書は[技術書典14\(2023/5-6 オンライン・オフライン同時開催\)](#)に向けて執筆しました。技術書典11に出典させて頂いた「ゼロから始めるFlutterアプリ開発入門」、技術書典13に出典させて頂いた「宣言的UIアプリ開発食べくらべ」とは全く毛色の違う企画となりました。そして、執筆は3月後半から開始・4月末に執筆完了、とかなりの短期スケジュールとなりましたが、無事執筆し終えることができました。

少しでも多くの皆様がこの本を閲覧し、ご自身の現場で「ITエンジニア体験教育」を実践するための手がかりにいただければと思います。

また、今回も社内の多くの皆様にご支援いただきました。活動を前向きに応援いただいた上長の皆様、社内の決裁を回して下さった皆様、社内の手続きでQA対応して下さった皆様、表紙のデザインを検討・作成していただいた皆様。こうして形にできたのは皆様のお陰様です。本当にありがとうございます。

今後も当社では本の執筆を通じ、多くの開発者に役立つ情報を発信する活動を続ける考えです。今回はITエンジニア体験教育に関する本でしたが、当社には他の分野に精通した社員も沢山いるので、この活動の分野を広げて行くことも考えています。

是非、当社の今後にご期待いただければと思います。

2023年5月 NTTテクノクロス株式会社 執筆メンバー一同

表紙デザインについて

今回は「ITエンジニア体験教育」をテーマとしました。

教育がテーマということで、まず真っ先に思いついたのが、「学校」そして「教室」「黒板」でした。子供らしさから連想される色味。そして、学校のあれこれのモノを並べる。

最後に、ITエンジニアという職業についての本なので、黒板には執筆者の子供の頃の夢を並べてみよう、ということで出来上がったのがこの表紙です。

食べ物ではないですが、「**黒板本**」などと呼んでください。

ところで、皆さんの子供の頃の夢はなんでしたか？

著者紹介

神長 貴博(Twitter:@onyankopon_0311)

会社でiOS・Androidアプリの技術支援、開発をしつつ、社外で小学生向けにプログラミングの授業、プログラミングクラブの監修・指導、講演活動を行っています。

趣味はランニング(横浜マラソン、川崎国際多摩川マラソン等の大会に出場)と食べ歩き。

子供の頃の夢は「ラーメン屋」。今もラーメンが大好きです。

本書では、1章(NTTテクノクロス流 学生向けプログラミング・キャリア教育)の全般、2章(プログラミング教育(Scratch))の執筆を担当。本プロジェクトでは、PL(プロジェクトリーダー)/社内調整などを担当。

戸部 雄太郎(Twitter:@etctaro)

Android、Kotlin、自動テストなどの技術をフォローしたり、社内の勉強会を開催したり、社内のちょっとした話題から技術的なノウハウまで様々な情報を社内展開しています。

趣味はボクシング観戦とゲーム。モンハンハンマー使い。

子供の頃の夢は「バスケの選手」。スラムダンク直撃世代でした。

本書では、全体構成と3章(プログラミング教育(Android))の執筆を担当。本プロジェクトでは、SL(サブリーダー)/デザイン調整などを担当。

神原 健一(Twitter:@korodroid)

FlutterやWear OSなど新しい技術大好きエンジニア。会社では技術支援・講師、技術/英語コミュニティ運営、社外では講演や執筆などエバンジェリスト活動を実施しています。

マイブームは、英語とランニングとアプリ開発。

子供の頃の夢は「お菓子屋さん(その後、医者)」。食べるのが大好きでした(今も変わらず)。

本書では、1章(NTTテクノクロス流 学生向けプログラミング・キャリア教育)のキャリア教育部分、4章(キャリア教育)の執筆を担当。本プロジェクトでは、PM(プロジェクトマネージャ)/全体統括などを担当。

★表紙デザイン： 足達 俊雅

みんなではじめる 小中学生のためのITエンジニア体験教育

2023年5月20日 初版発行

- 著者 NTTテクノクロス株式会社
- 編集 NTTテクノクロス株式会社
- 発行 NTTテクノクロス株式会社

(c) 2023 NTT TechnoCross Corporation