

はじめてのプログラミング

メンター用資料

この資料は、株式会社 Innovation Power（代表：宮島衣瑛）がScratchを用いた初心者向けワークショップを実施する際に行っているファシリテートの中身を記録したものです。タイムライン形式になっており、どのタイミングでどのような発問を子どもたちに投げかけるのかなどが記載されています。適宜参考にしながらワークショップを進めてください。

最終更新日：2018年8月14日

■ 前提条件

ワークショップは2時間想定です。参加者はコンピュータの操作は可能だが、プログラミングをしたことはない初心者を想定しています。

- ・使用ツール：Scratch 2.0 オンライン版（オフラインでも可）
- ・想定人数：20名前後
- ・年齢層：小学校3年生～中学3年生

■ ワークショップ前の準備

【ファシリテータ（進行を進める人）】

①：Scratchの教員用アカウントを取得する

Scratch の教員用アカウントは、自分のクラスを作る機能やクラスに追加する生徒のアカウントを一括で追加できる機能が実装されています。ワークショップのファシリテータをやる方にはマストな機能ばかりですので、作成してください。

作成方法は、次の公式サイトを参考にしてください。

<https://scratch.mit.edu/educators/#teacher-accounts>

※申請にはMITチームへのリクエストが必要となるため、少し時間がかかる場合があります。余裕を持って申請を行うようにしてください。

②：クラスに追加するためのアカウント

Scratchの教員用アカウントには、生徒アカウントを一括で追加するための機能があります。次の手順で行ってください。

・ <https://scratch.mit.edu/educators/classes/> にアクセスして、自分のクラスを作る。（すでに作成済みの場合はやらなくてよい）

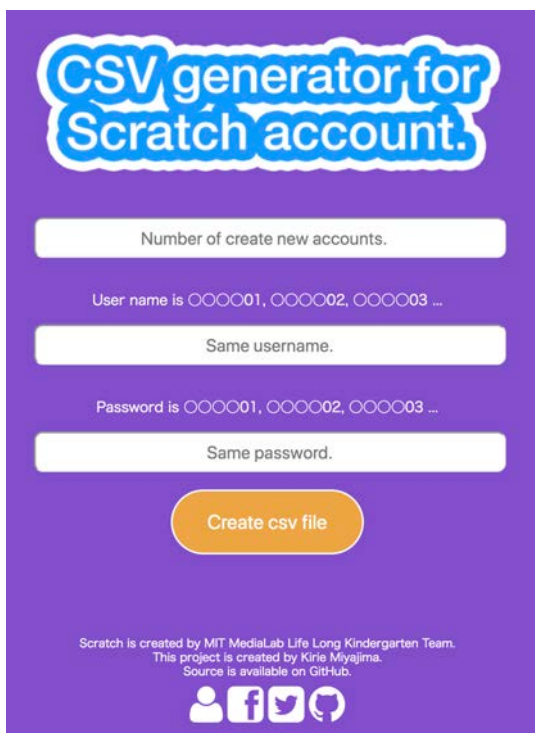
・ 「生徒」リンクをクリックして、「複数の生徒を追加する」の「CSVをアップロード」を確認する。



userName1	password1
userName2	password2
userName3	password3
userName4	password4
userName5	password5

csvファイルはこのような形式で用意しますが、作るのは大変なのでジェネレーターを使用します。

・ <http://kirie.me/apps/scratch-csv/> にアクセスして、CSVファイルを作成する。



Number of create new accounts.

=> 作成するアカウントの総数

Same username

=> 作成するユーザ名の共通部

Same password

=> 作成するパスワードの共通部

例)

Same username = innovation_

Same password = power_

innovation_01	power_01
innovation_02	power_02

- ・ 生徒アカウントCSVをアップロードする
- ・ 参加者にユーザ名とパスワードを教える

※このジェネレータで作成したアカウントの連番は、数字が一桁の場合は 01, 02 のように先頭に0が付きます。

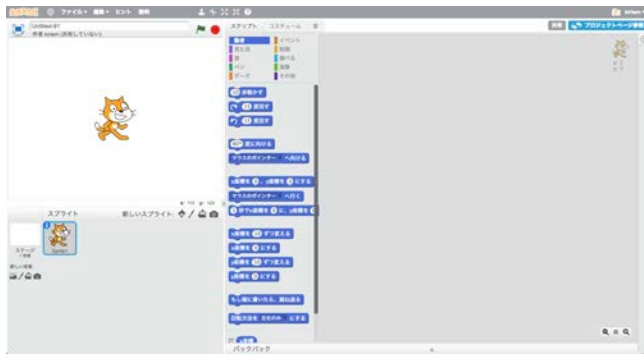
■ ワークショップの流れ

時間	内容	備考
0:01	・ワークショップスタート！ ☐ ファシリテータ挨拶 ☐ メンター紹介 ☐ 本日のテーマなどの発表 // もし、ほとんどの子が初めて参加する場合や入場時に静かすぎた場合には、アイスブレイクを行う。 /* ・アイスブレイク 目的：ワークショップ参加者の関係性を柔らかくする。 アイスブレイクの手法はたくさんあるが、ここではいくつか紹介しておく。 ①：サイレントバースデーゲーム 一言も話さずに、身振り手振りだけを使って教室の端から順に1月1日に近い順から順番に並ぶゲーム。 子どもたちには以下の指示を出す。 ・喋ること以外の動作をつかって、自分の誕生日をうまく伝えよう。 ・教室の前方から1月1日に近い人の順番になって並びましょう。時間は3分間です。どうやったら素早く、正確にできると思いますか？ 実際にやってみて、正解しようがしまいがみんなで👏をする。「今日はこのメンバーで最後までやっていきます。仲良く取り組んでください！」などの声かけをする。 */	
ワークショップ本編①：ねこあるき		
	☐ ブラウザを開かせる ・ Google Chrome または Microsoft Edge を推奨 ・ Internet Explorer 以外であればなんでもよい ☐ https://scratch.mit.edu にアクセス ・ "Scratch"とGoogleで検索してもよい。 ☐ Scratchの説明をする	

- ・Scratchはアメリカのマサチューセッツ工科大学メディアラボが開発した教育用のプログラミングツールである
- ・全世界で使われており、いま世界で最も有名なプログラミングツールのひとつ

☐ さっそくやってみる

・画面左上の「作る」ボタンを押して、エディタ画面に飛ぶ。画面の説明をする。



左側が「スマートフォンやゲームの画面と同じ、作ったものが動く画面」

真ん中のブロックたちを右側の灰色のエリアにドラッグ
アンドドロップして組み合わせることでプログラミングを
していく

漢字が難しくて読めない場合は、画面左上地球儀マークをクリックして、下にスクロールすると「ひらがな」モードがあるので、これを利用する。

☐ **<10歩動かす>** をクリックする

□ **<10歩動かす>** を右側のプログラミングエリアにドラッグアンドドロップさせて、同じようにクリックする。

☐ ネコも左側に移動させ、全員でレースをする。

・誰が一番早くこのネコを右側に移動させることができるか勝負をしよう

・やっていいことは1つだけ。10歩動かすをひたすらに連打するだけ

☐ いまやっていたことは楽しかったか？

・ネコを何度もクリックして右側に向かわせることはめんどくさくないか？

・ 人類がプログラミングをするのはどうしてか？

-> 楽をするためである

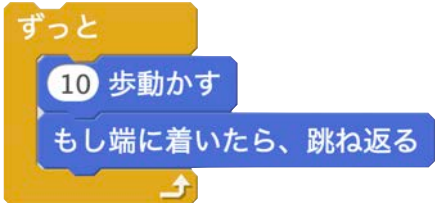
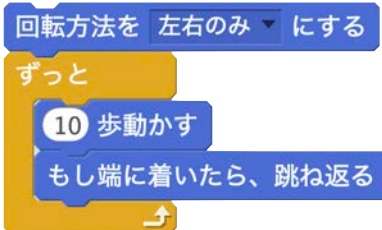
このとき、実行画面のネコがどのような動きをしたかを確認させる

ドラッグアンドドロップの動きを確認する

スプライトも移動させることができることを確認する

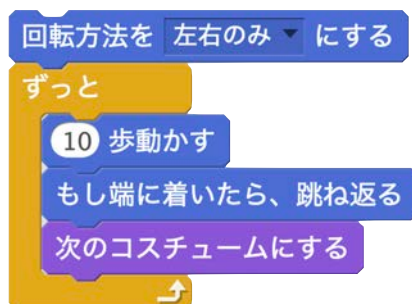
ひたすら連打する = めんどくさい
という感覚を持ってもらいたい

この結論を子どもたち自身た出せるようなヒントを出す

・自動改札機も自動販売機も、人間がやらなくてもいいようなことはすべて機械に任せようとした。それを作る力があるのがプログラミングである。 ・いま私たちが抱えている問題は、ネコを右側に持っているのが大変だということ。これを解決しよう。  ・制御の中にある <ずっと> を右側に持ってくる。 ・<10歩動かす> ブロックを<ずっと>の中に入れる。 ・<ずっと>を動かして、ネコの動きを観察する。 ・さっきよりだいぶ楽になったが、まだめんどくさい部分がある。それはどこだろうか？ ☐ もっと楽ができる部分を探す ・しっぽを掴んで戻すことがめんどくさい ・自動で戻ってくるとよい  ・<もし端についたら、跳ね返る> ブロックを追加する ※この状態ではネコがひっくり返ったままになっているので、これを修正するプログラムを追加する  ・<回転方法を左右のみにする> ブロックを<ずっと>の上に挿入する。これによりひっくり返る現象が修正される。 ・これでネコを動かすプログラムは完成した。でもこれはスケートをしているみたい。次はネコを歩かせてみよう。	プログラミングには、目の前にある大変な問題を解決できる力があることを紹介する。 <ずっと>ブロックの中に他のブロックが入ることを理解する ネコが画面の右側に行ってしまった場合は、しっぽを掴んで左側へ移動させる。 回答が出なかった場合は、何度か発問を繰り返す ここは子どもたちが気づくので、自然な流れで修正まで持っていく。
---	--

□ ねこあるきプログラムを作る

・ネコを歩かせるためには、【見た目】の中にある＜次のコスチュームにする＞ブロックを追加する。



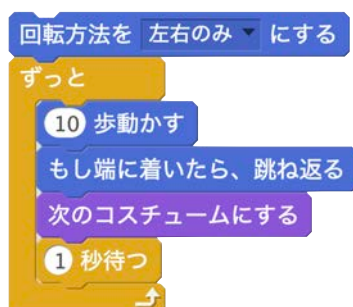
・しかし、これでは早すぎる。足が高速回転してしまっているのを、修正する。

・そもそも、＜次のコスチュームにする＞とはなにかを説明する。

・画面中央にある コスチューム タブをクリックすると、右図のような画像が出現する。これが次のコスチュームの正体である。

・両足が地面についている写真と、片足が上がっている写真がある。この2枚の写真を次々と交互に表示することで、歩いているかのように見えるアニメーションを作っている。これはパラパラ漫画の仕組みと同じ。

・今はこの切り替える時間が早すぎるから、走っているかのように見えてしまう。切り替える時間を少しゆっくりにしてみよう。【制御】の＜1秒待つ＞ブロックを追加。



・1秒待つだと遅いので、1より小さい数に変えてみる。

🎓 小数の説明

ホワイトボードなどに線分を書き、0と1をあえて離して書く。

0と1の間がすごく開いていることを確認し、この間にも数があることを説明する。

※子どもたちはすでに体温や気温などで小数を使っている場合が多い。同じことだという説明をしてもOK。



小数点の入力方法がわからない子どもがいるので、確認する。

00:30	□ 改造タイム 子どもたちに自由に制作する時間を与える。しかし、何も知らない状態で放り出されることは不自由なので、いくつかのブロックを例示し紹介する。 画像効果をなくす 大きさを 10 ずつ変える 色 の効果を 25 ずつ変える 色 魚眼レンズ 渦巻き ピクセル化 モザイク 明るさ 幽霊	このブロックがある。という紹介ではなく、実際に自分のコードに追加していくとウケがいい。
ワークショップ本編②：ねこたたき		
	・ YouTubeでもぐらたたきの映像を見る https://www.youtube.com/watch?v=ql7fytLHWRQ ・ もぐらたたきを作るが、Scratch にはモグラはいないのでネコで、実際に叩くとコンピュータが壊れてしまうのでクリックで代用した、「ねこたたき」ゲームを作る。 ・ 完成版を見る https://scratch.mit.edu/projects/238715493/ □ 要素分解を行う ・ 別紙のワークシートを配布する。 ・ 完成版を2-3度見せて、要素分解をさせる。 ・ まずは自分1人で考える（5分程度） ・ その後、周りの子どもたち同士で考える（5分程度） ・ ファシリテータは全体を見つつ、グループを指名して発表させる。クラス全体の同意が得られたら、開発に移る。	最近の子どもたちの多くは、もぐらたたきをやったことがない。映像を見ることでイメージをもたせる。 プログラミング的思考の、「構造を捉えて細分化」の部分

□ねこが隠れたり表示されたりする機能を作る。



ここで、緑の旗をクリックして実行すると、スプライトは1秒ごとに点滅していることがわかる。

これだとゲーム性がないので、ランダムに表示させるようにする。

1 から 10 までの乱数

【演算】の<1から10までの乱数>ブロックを使う。乱数とは、1～10まで順番に数えるのではなく、1から10までの範囲の数のどれかをランダムに出してくれるブロックである。

今回は、10秒も待つと長すぎるので、1から3までのの乱数を発生させることにする。

<乱数>ブロックを<1 秒待つ>の中に入れる



これにより、表示時間がバラバラになりゲーム性が高まる。

□ クリックされた時の処理を作る

ネコがクリックされたときの処理を作るためには、【制御】内の<このスプライトがクリックされたとき>ブロックを使う。

スプライトがクリックされたときには、ネコを隠して「ニャー」と鳴かせたいので、次のブロックを追加する。

子どもたちから意見を出させる。

「ランダム」という言葉は日常生活で使っていることが多いので、理解していることが多い。

ちなみに、乱数の数字を整数ではなく小数値にすると、乱数がとり得る範囲は小数も含まれるようになる。

このスプライトがクリックされたとき

隠す

meow の音を鳴らす

ここまでで一旦試してもらおう。ちゃんと音になっているかどうか確認する。

📢 音がならない場合のトラブルシューティング 📢

大体の場合、次の原因が考えられます。

- ・OSの設定により、スピーカーがオフになっている
- ・ブラウザの設定により、音がオフになっている

□ 変数を使って得点を作る

※変数は多くの子どもたちにとって超えなければいけない最初の大きな壁です。しっかりとフォローしましょう。

・得点を作るためには、【データ】の<変数>を使う。【データ】にはブロックがないので、改めて作らなければならない。



- ・変数名は「得点」にする。
- ・「すべてのスプライト用」になっているかどうかを確認して、OKを押す。

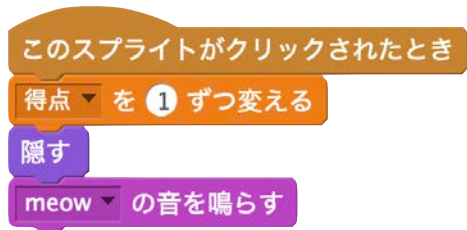
🎓 変数とはなにか？

変数とは、「数字」や「文字」を保存できる箱のこと。例えば、子どもたちに「年齢」はいくつですか？と聞くと、バラバラの答えが返ってくる。これが変数の本質である。同じ「年齢」という言葉に入っている数字は異なる。

「得点」のローマ字が打てない子どもがいるので、フォローする



変数の説明が一通り終わったら、さっそく使ってみる。
ネコがクリックされたときのプログラムに＜得点を1ずつ
変える＞ブロックを追加する。



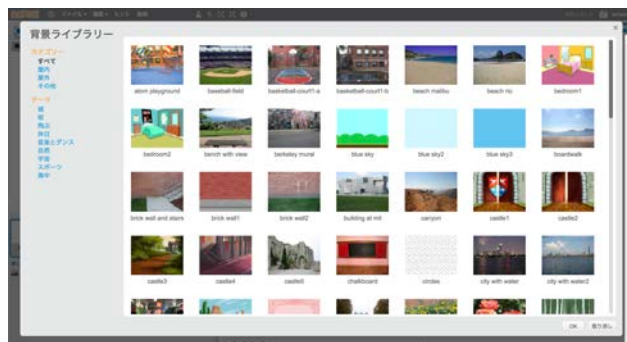
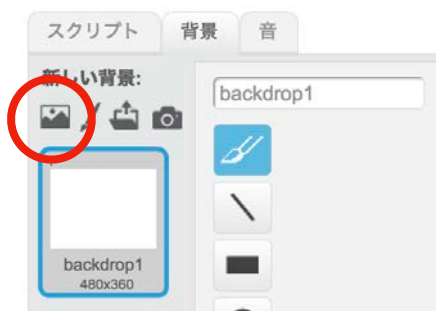
これで、得点が増えていくようになった。

□ 背景を変える

ゲーム画面が真っ白だとつまらないので、背景の画像を選ぶ。
画面左下の「ステージ」をクリックする。



画面中央上部の背景タブをクリックして、新しい背景を追加する。



ゲーム中、ゲームクリアの2つの背景を選ばせる。

変数は言葉や理論で覚えるより、
体で使い方を覚えたほうが良いの
で、どんどんやる。

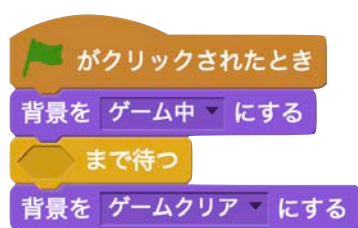


背景を選んだら、それぞれの背景に名前をつける。
「ゲーム中」及び「ゲームクリア」を入力する、

タイピングをフォローする

□ 背景を制御する

ゲームスタート時には「ゲーム中」の画面に、得点が10点を超えたら「ゲームクリア」の画面に変えるようなプログラムを作る。



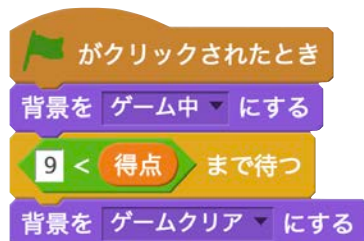
得点が10点を超えた場合の処理を作る。【演算】の
< A < B > ブロックを用意し、左側に9を、右側に
< 得点 > を追加する



🎓 なぜ、9 < 得点になるの？

A < B は、BがAより大きい状態 == 真 である状態を指します。数学では $\leq$ という～以上を表す記号がありますが、Scratchにはありません。「A以上」と言ったときにはAも範囲含まれますが、「Aより大きい」と言ったとき、Aは含まれませんので、1つ小さい数の9を代入しているのです。

説明が終わったらブロックを追加する。



また、2回目以降遊ぶときに変数をリセットしたいので、変数を0にする処理を追加する。



最後に、ネコのスプライトに戻って、空いている部分に評価式を入れる。最後に<隠す>も忘れずに。



ここまできたら、後はネコを増やすだけです。ネコのスプライトを右クリックして「複製」を押すと、プログラムを保持したままネコが複製されます。



以上でプログラミングは終了です。お疲れ様でした！

もし時間が余ったら、適宜改造の時間等に当てて調整する。

02:00

ワークショップ終わりの挨拶をする。