

これからを見据えた
「プログラミングで教える」
をテーマにした
プログラミング実践

東京都立石神井高等学校
情報科 小松 一智

komatsu@komatsu-no-site.net

2022年

新學習指導要領 年次進行

情報 I

情報 II

全生徒が
プログラミングを！？

でも、何をやるの？

みなさん、
何をやりますか？

「プログラミングで教える」
ってどうですか？

でも、プログラミングも大変
でしょ？

今までやってきたこと (プログラミング関係)

- ExcelでVBA
- ドリトル
- ビュートレーサー
- HTML & CSS
- レゴマインドストームEV3
- Scratch

毎회가試行錯誤

□ ExcelでVBA

補完機能があるため、扱いやすい
しかし、何となくで動いてしまう

アルゴリズム → コーディングにならない
とりあえずコーディングしていく生徒多数

□ ドリトル

日本語で命令を書けるので初めてでも取り組みやすい

全角の記号 (——) の違いでエラーに
長音、環境依存、マイナス

□ ビュートレーサー

動かす対象があるので取り組みやすい
ブロックプログラミングなのでわかりやすい

安価であるため強度が . . .

台数を確保するのが . . .

□ HTML & CSS

メモ帳さえあれば取り組める
(Webサーバも必要となるが)

必要性を感じる生徒が少ない

□ Scratch

中学校で扱ってきた生徒がいる！

年々増えている！

ブロックプログラミングなので取り組みやすい

□ レゴマインドストームEV3

昨年初めて実践

使い方を理解できていない（自分が）

ループなどがきちんと動いているのかわかりにくい

ファームウェアアップデートのお知らせで文鎮化することが。。。。

ここ数年の取り組み

Blocklyでブロックに慣れる 1時間

(<https://blockly-games.appspot.com/>)

Scratchで基本要素(順次処理、判断分岐、
繰り返し)を学ぶ 2~5時間

昨年から

ドリトルでプログラミング

正しくコーディング(文字入力を間違えない)
オブジェクトを作ってから配置など順序を理解させる

兼宗先生、久野先生ありがとうございます

今までは

プログラミング = 写経

タイピングが早い生徒 = ヒマになる、
意欲をなくす

タイピングが遅い生徒 = 終わらない、発狂
する、

意欲をなくす

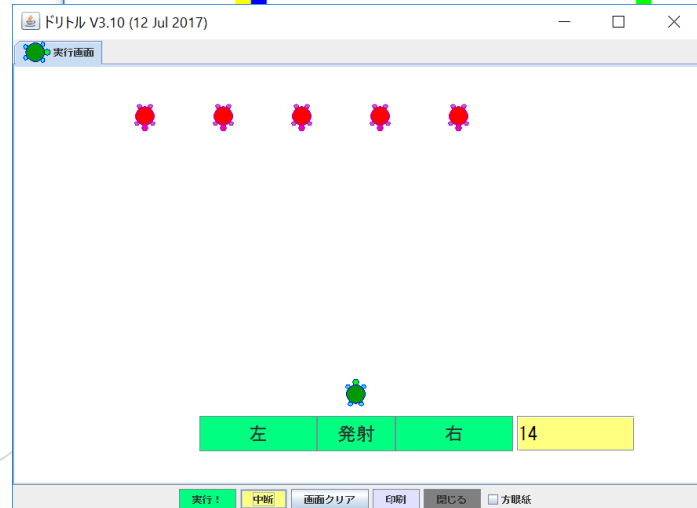
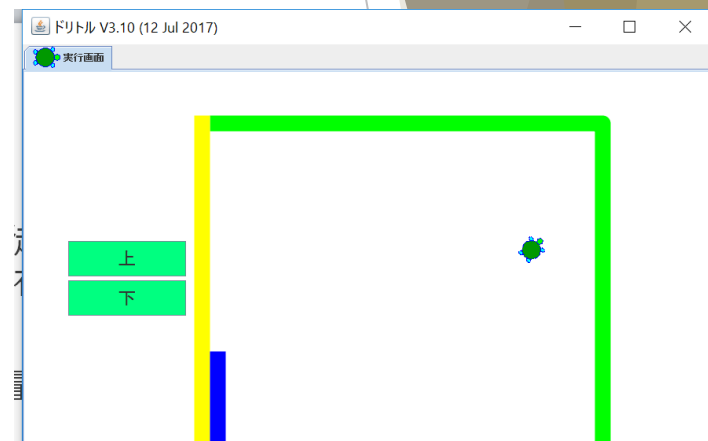
ドリトルに慣れる

1時間でできるプログラミング

ピンポンゲーム

シューティングゲーム

ブロック崩し



試してみた

STEP毎に写経

ただし、早い生徒向けにその後のコードをQRコードで配布

コーディングの量が多いと一部の生徒しか終わらない

試してみた

各STEPのコードを予め配布

ただし、一部不完全な状態で配布

例) `//ステップ1 主役と左右の動き
かめた=タイトル!作る 90 左回り ペンなし 0 -150 位置。
左=ボタン!“左” 作る -200 -180 位置。
左:動作=「かめた! -20 0 移動する」。`

自然と自分で考えながらコーディング
様々なアレンジを加えることも

ラケットを作る（数字は半角で）

かめた=タートル！ 作る。

かめた！（緑）線の色 20 線の太さ。

かめた！ 500 歩く 90 右回り 420 歩く 90 右回り 500 歩く。

壁=かめた！図形を作る -200 200 位置。

かめた！ 90 右回り。

左壁=かめた！（黄）線の色 440 歩く 図形を作る -210 -230 位置。

ラケット=かめた！（青）線の色 120 歩く 図形を作る -190 -210 位置。



ステップ2

キー操作できるようにしよう

上ボタン=ボタン！ "上" 作る -380 50 位置。
上ボタン：動作=「ラケット！ 0 50 移動する」。
下ボタン=ボタン！ "下" 作る -380 0 位置。
下ボタン：動作=「ラケット！ 0 -50 移動する」。



ステップ2

キー操作できるようにしよう

上ボタン=ボタン！ "上" **"UP"** 作る -380 50 位置。

上ボタン：動作=「ラケット！ 0 50 移動する」。

下ボタン=ボタン！ "下" **"DOWN"** 作る -380 0 位置。

下ボタン：動作=「ラケット！ 0 -50 移動する」。

UP DOWN
半角



ステップ3

かめたの向きを変えよう

```
// ボールを動かす (ステップ3)
かめた！ペンなし。
かめた！ (乱数 (200) ) (乱数 (300) -150) 位置。
かめた！ 45 向き。
かめた：衝突＝タートル：跳ね返る。
時計＝タイマー！作る 60秒 時間 「かめた！ 20 歩く」実行。
```



ステップ4

ゲームの勝敗を判定する

ゲームクリア=はい。

左壁：衝突=「：ゲームクリア=いいえ。時計！中断」。

時計！待つ。

「ゲームクリア==はい」！なら「

ラベル！"ゲームクリア！" 作る (青) 文字色。

」そうでなければ「

ラベル！"ゲームオーバー！" 作る (赤) 文字色。

」実行。



さらに

「プログラミングで学ぶ」を取り入れた
ネットワークの導入～仕組み

「チャットを作ろう」

IPアドレス、ホスト名、DNS

「音楽を交換しよう」

MIDIとは

ダウンロードとストリーミングの違い

チャット

- ◆ Localhostに接続し、一人でチャット
- ◆ 教員のコンピュータに接続し、クラス全体でチャット
- ◆ グループ内でチャットができるように自分たちで考えてチャット

チャットを作ろう

//STEP1 名前を表示させる。メッセージ入力フィールドを作る。

名前="かめた"。

ラベル！（名前） 作る 200 45 大きさ -250 200 位置。

メッセージ=フィールド！ 作る 200 45 大きさ -250 150 位置。

チャットを作ろう

//STEP2 サーバーに接続。フィールドの動作を設定。

サーバー！ "localhost" 接続。

サーバー！ "msg" "" 書く。

フィールド：動作 = 「

サーバー！ "msg" (名前！ ": " (メッセージ！ 読む) 連結) 書く。

メッセージ！ クリア。

」。

ここは1行

チャットを作ろう

//STEP3 サーバーからメッセージを受信。

受信表示=リスト！ 作る 200 400 大きさ 0 200 位置。

直前メッセージ=""。

タイマー！ 作る 1 間隔 600 回数「

受信メッセージ=サーバー！ "msg" 読む。

「受信メッセージ!=直前メッセージ」！ なら「受信表示！（受信メッセージ）書く」実行。

直前メッセージ=受信メッセージ。

」実行。

ここは1行

音楽配信

- ◆ グループ内でサーバ役、クライアント役
- ◆ タイミングを合わせてダウンロード
- ◆ タイミングを合わせてストリーミング

「プログラミングで教える」
ってどうですか？

面白いですよね？