

探究的にプログラミングを学ぶ 情報Ⅰ授業

岡山理科大学

高橋 信幸

京都府立乙訓高等学校

河端 梓

概要

探究的なプログラミング教育
→プログラミング的思考力への影響

順次・分岐・繰り返し・変数などの
知識の獲得が促進された

コーディング指導
探究的プログラミング教育
を併せて実践

プログラミングの楽しさや達成感を実感
主体的に取り組む態度育成に寄与

1. 問題の所在

プログラミング教育

これまで

専門的に学びたい
一部の生徒が対象

これから

全児童・生徒
が対象

視点が重要

プログラミングに対する興味関心を高め、
プログラミング的思考力を養う

理想！ プログラミング教育が創造性の育成に資する
Jeanette Wing(2006)

探究的な学びは教授者による講義を聞くよりも
主体的・協働的で深い学びで、思考力育成にも効果的

現実！ プログラミングが嫌いな生徒を量産??

中学までに学んだ「変数」や「逐次・分岐・繰り返し」
の概念を獲得できていたのは約1割 高橋ら(2024a)

2025年1月 大学入学共通テスト情報Ⅰ 第3問
コーディングに関する問い(2~3) 正答率は約50%
情報Ⅰ全体の平均69%よりも低かった

小・中学校:

Scrach等のビジュアルプログラミング環境

プログラミング環境に依存せずに活用できる
「プログラミング的思考に関するスキーマ」
を想起・活用して学ぶ

高校:

Python等のテキストプログラミング環境

プログラミング的思考に関するスキーマ
を意識した学習が効果的であると推測

共通テストで問われるプログラミング学習項目

項目	詳細
基本構造	逐次, 分岐, 繰り返し (ネスト含む)
変数	変数 (型, スコープ含む)
関数	関数 (組み込み, 作成, 再帰活用)
配列	配列 (2次元含む)

スキーマ

問題解決の際に用いられる
一連の知識や経験・手続きのまとめり

問題解決の流れとプログラミング的思考

プログラミング的思考に関するスキーマ
を意識した学習が効果的であると推測

問題解決の流れ	プログラミング的思考
課題の発見	条件分け
アルゴリズムを考える	手順の創造
デバッグ	規則性の発見
課題の解決	アルゴリズムの精査
	構造の最適化
	データの抽出

2. 目的及び方法

目的

次の2つを意識して教育実践を行い、成果と課題を分析・考察する

- ・身につけさせたいスキーマを明確にして指導する。
- ・必要な知識・技能を学んだ後に、探究的な学びを実施

方法

授業実践

対象: 京都府内の高校1年生6クラス235名
内容: 基礎的な学び6時間, 探究的な学び2時間

- (1) 基礎的な学び(実施時期: 2024年10月~11月)
オンラインプログラミング教材
「Life is Tech! 情報 I Python」を用いた学習
- ・アルゴリズムとは(フローチャート含む)
 - ・プログラミングの基礎(変数, 演算子, 条件分岐)
 - ・繰り返し, 配列, 関数

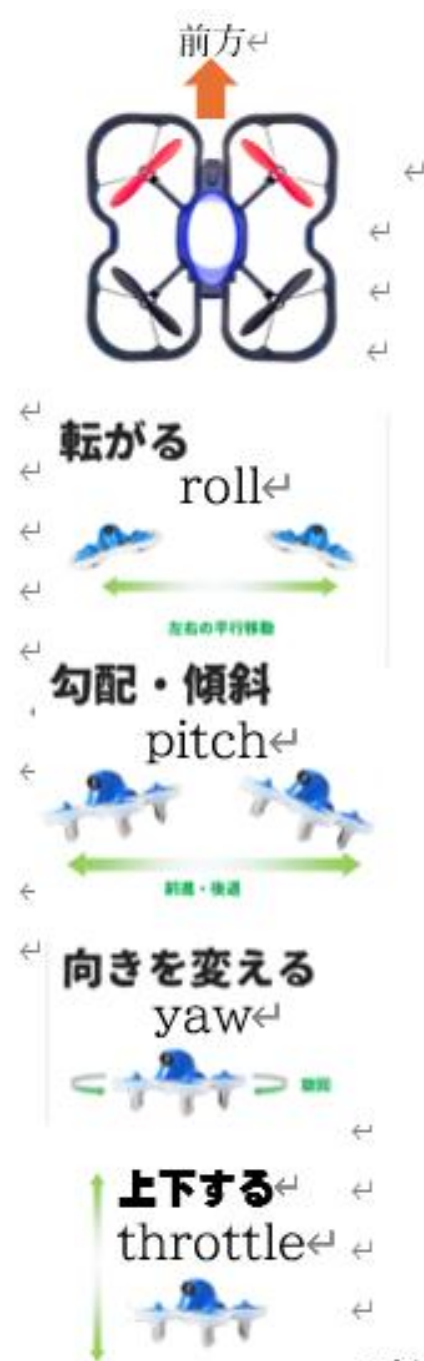
- (2) 探究的な学び(実施時期: 2025年1月~2月)
ドローンを用いた探究的なプログラミング学習
- ・変数のスキーマ
 - ・逐次・分岐・繰り返しのスキーマ についての解説

これらを意識してプログラムコードを読み解き、
目的に応じたプログラムの作成・実行

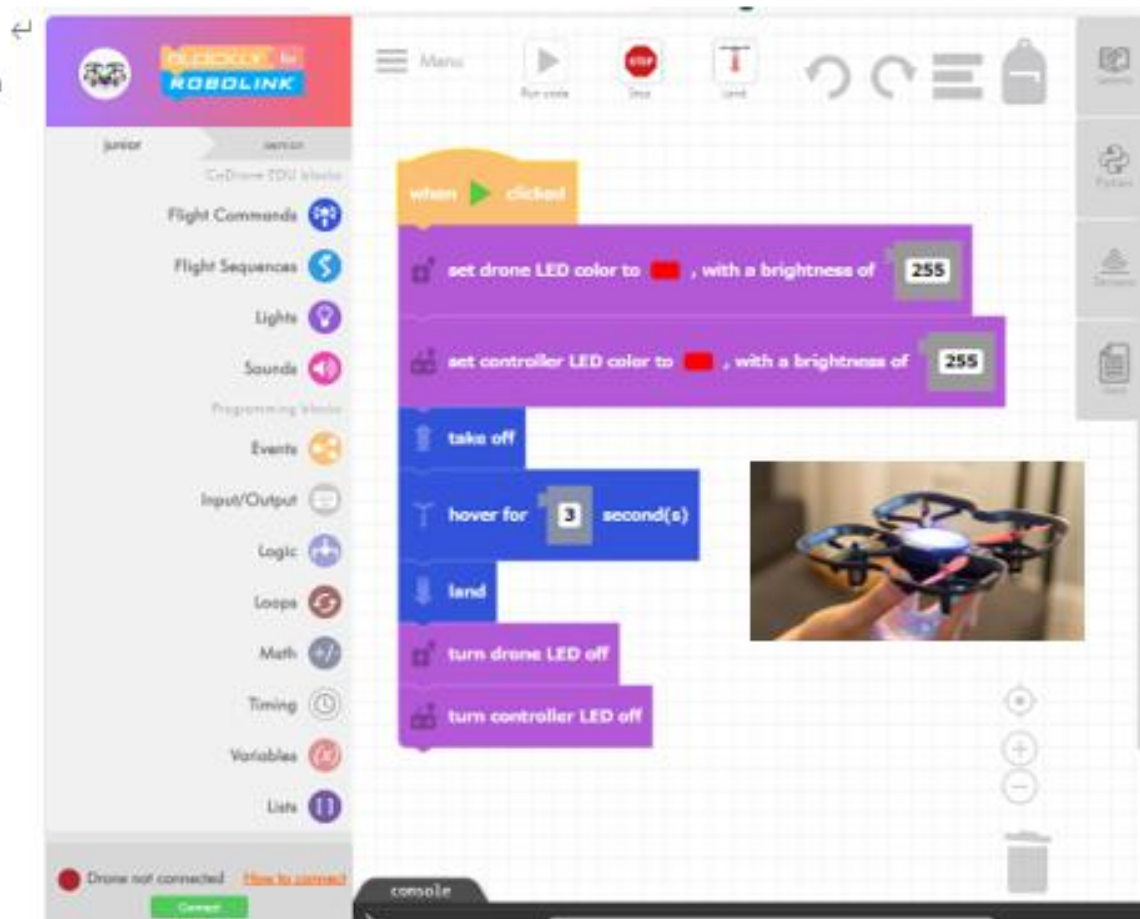
定期考査、ワークシート、行動観察 → 学習内容の定着や生徒の変容を分析



ドローンを用いた 探究的な プログラミング学習



プログラミング画面



コントローラをUSBケーブルでつなぐと、ドローンがプログラム通りに動く。

ドローンを用いた 探究的な プログラミング学習

Lesson2 ドローン操作に挑戦！

- (1) 右のプログラムを作成しよう
(色は自由に変更可)
- (2) コントローラをUSBケーブルで接続
- (3) プログラムを実行してみよう
- (4) 班員全員がやってみよう
(プログラムは変えていい)

どんな動作をするか予測してみよう

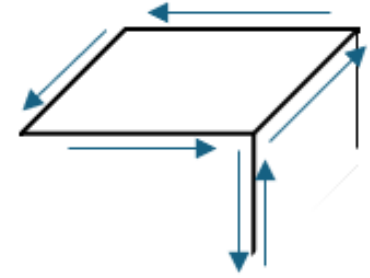


```
when clicked
  play this note F3 for 1 second(s) on drone
  set_drone_LED( 0, 255, 0, 255 )
  repeat 2 times
    do
      move( 0, 25, 0, 0, 0.6 sec)
      move( 0, -25, 0, 0, 0.6 sec)
      move( 0, 0, 25, 0, 0.6 sec)
      move( 0, 0, -25, 0, 0.6 sec)
    do
  flip( back )
  drone_LED_off()
  take off
```

ドローンを用いた 探究的な プログラミング学習

Lesson 3 四角形飛行に挑戦！
次のプログラミングに挑戦しよう

- (1) ドローンを飛ばしてホバリング
- (2) 水平移動で四角形を描く
- (3) 元の場所に着陸



〔ヒント〕プログラミングの知識

(1) 逐次実行

プログラムは上から順番に実行される

(2) 繰り返し

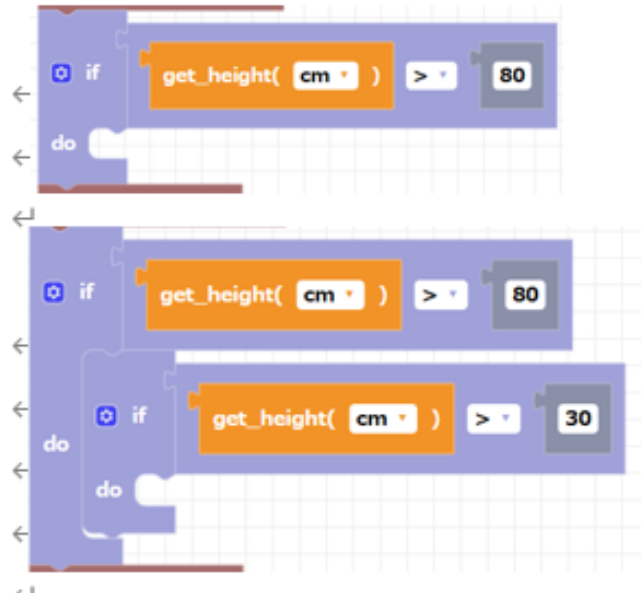
ある部分を繰り返して実行するには繰り返しコマンドを使う



←

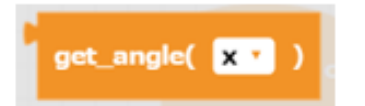
(3) 分岐

条件を満たすときと、そうでないときで実行するプログラムを変える。



(4) センサー

ドローンには7つのセンサーがある。
値を取得して利用できる。



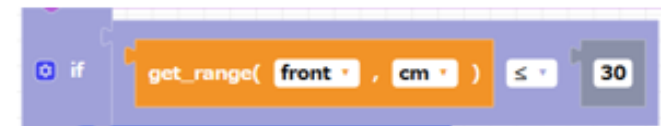
水平 x 軸からの角度を取得



前方物までの距離[cm]を取得

(5) 変数

プログラムは、変数に入れられた値を用いて計算・比較・実行する。



前方物までの距離[cm]を変数に代入し、
30cm 以内かを判断



ドローンを用いた 探究的な プログラミング学習



Lesson 4 プログラムの解説

次のプログラムを実行するとドローンはどんな動きをするだろうか？予想してみよう。

```
when clicked
  take off
  repeat 100 times
    if (get_range(front, cm) >= 25 and get_range(front, cm) <= 70)
      do move(0, 30, 0, 0, 0.4 sec)
    else if (get_range(front, cm) < 25)
      do move(0, 30, 0, 0, 0.3 sec)
    else if (get_range(front, cm) > 70)
      do hover for 0.5 second(s)
  land()
```

[予想：ドローンの動き]

予想できたら、実際にプログラムして動かしてみよう。←

予想通りでしたか？ はい いいえ←

どこが難しかったですか？（

〔結果：実際の動き〕←

Lesson 6 オリジナルのプログラミングに挑戦！←

自由に発想して、オリジナルのプログラムでドローンを動かしてみよう。←

←

〔チャレンジ結果報告〕チャレンジの内容と結果を書いて報告しよう。←

←

〔感想〕プログラミングを学んでの感想を書こう。←



3. 結果及び考察 3.1 基礎的な学びの評価

オンライン教材について

✓ 変数を使って計算し、足し算の結果を表示しよう

変数`total`を作り、変数`num1`と`num2`の足し算を代入しよう
`print`関数を使って、変数`total`の中身を表示しよう

```
# 変数num1と変数num2を足して変数totalに代入
total = num1 + num2

# 画面に計算結果を表示
???(???)
```

ヒントを見る

ヒントを閉じる

プログラムの実行結果

```
Python
1 #. 変数
2 num1 = .5
3 num2 = .100
4
5 #. 変数num1と変数num2を足して変数totalに代入
6
7
8 #. 画面に計算結果を表示
9
10
```

基礎的な学びでの様子

- ・意味を理解せず入力して動作確認する生徒が4割
- ・変数を扱う練習で、計算結果を自分で計算して、変数に直接代入させる生徒が2割いた
→恐らく変数の概念が獲得できていない

3. 結果及び考察

3.1 基礎的な学びの評価

A: 画像の表示の仕方などは今日の授業では勉強でできてへんから、文字で書けるようになるまでシンプルにゲームを作成してみよう
 B: まずは初期設定として、勇者とモンスターの設定が必要だね。
 A: ゲームの展開としては、勇者とモンスターが交互に攻撃、攻撃が当たれば体力から攻撃者の攻撃力の値を引く、そしてどちらかの体力 (HP) が0になった時点で終了でええな？
 B: ちょっと待って、それやとあかんや！ [] にしとかな。
 A: ほんまやな、そこはちゃんと終了条件を書きな。
 B: はな作ってこが！

(1) 文中下線部についてAが考えた判断終了条件だと不具合が生じる場合がある。改善案をBが空欄で述べている。空欄部分にははまる最も適切な選択肢を選び、【61】にマークしなさい。(2点)

- ～選択肢～
- ① 体力 (HP) が0よりも大きくなった時点で終了
 - ② 体力 (HP) が0以上になった時点で終了
 - ③ 体力 (HP) が0よりも小さくなった時点で終了
 - ④ 体力 (HP) が0以下になった時点で終了

```

# 戦闘ループ
while h_hp > 0 【65】 m_hp > 0:
    # 勇者のターン
    print("ゆうし+の攻撃！ らすばすに" + 【66】(h_atk) + "のダメージ！")
    m_hp = m_hp - h_atk
    if m_hp <= 0:
        print("ラスボスを倒した！ ゆうし+の勝利！")
        break # mにモンスターの負けが確定したらループから出る命令
    print("らすばすの残りHP:" + 【66】(m_hp))
    # モンスターのターン
    :
  
```

【65】の選択肢

- ① and
- ② or
- ③ ==
- ④ !=

【66】の選択肢

- ① str
- ② choice
- ③ for
- ④ randint



	出題内容	正答率
繰り返しのアルゴリズムの出題	1) 繰り返し条件	83%
	2) 繰り返し回数	70%
	3) 実行結果	92%
プログラムコードを完成させる出題	4) 繰り返しの条件	34%
	5) 文字列への変換	54%
	6) 関数	36%
	7) 条件分岐	73%

3. 結果及び考察 3.1 基礎的な学びの評価

定期考査での評価

	出題内容	正答率
繰り返しに関する アルゴリズムの 出題	1)繰り返し条件	83%
	2)繰り返し回数	70%
	3)実行結果	92%
プログラムコード を完成させる出題	4) 繰り返しの条件	34%
	5) 文字列への変換	54%
	6) 関数	36%
	7) 条件分岐	73%

アルゴリズムを構築する
能力は
身につけている？

繰り返し、文字列変換、関数、条件分岐
をプログラム中で扱う **応用的な問題**

- ・約7割の生徒・・・完答できていない
- ・**知識が断片的な状態**のままになっている
- ・応用可能な**スキーマの獲得に至っていない**

3. 結果及び考察 3.1 基礎的な学びの評価

A: 画像の表示の仕方などは今回の授業では勉強できてへんから、文字で展開が分かるようなできるだけシンプルなゲームを作成してみよか

B: まずは初期設定として、勇者とモンスターの設定が必要やね。

A: ゲームの展開としては、勇者とモンスターが交互に攻撃、攻撃が当たれば体力から攻撃者の攻撃力の値を引く。そしてどちらかの体力 (HP) が0になった時点で終了でええな？

B: ちょっと待って、それやとあかんで！ にしとかな。

A: ほんまやな。そこはちゃんと終了条件を直すわ。

B: ほな作ってこか！

(1) 文中下線部についてAが考えた戦闘終了条件だと不具合が生じる場合がある。改訂版が空欄で述べている。空欄部分に当てはまる最も適切な選択肢を選び、【6 1】にマークしなさい。(2点)

～選択肢～

- ① 体力 (HP) が0よりも大きくなった時点で終了
- ② 体力 (HP) が0以上になった時点で終了
- ③ 体力 (HP) が0よりも小さくなった時点で終了
- ④ 体力 (HP) が0以下になった時点で終了

83%

```
(省略 初期設定)
print('ゆうしゃとらすぼすの戦いが始まった！')

# 戦闘ループ
while h_hp > 0 【6 5】 m_hp > 0:

    # 勇者のターン

    print('ゆうしゃの攻撃！ らすぼすに' + 【6 6】(h_atk) + 'のダメージ！')

    m_hp = m_hp - h_atk
    if m_hp <= 0:
        print('ラスボスを倒した！ ゆうしゃの勝利！')
        break #もしモンスターの負けが確定したらループから出る命令

    print('らすぼすの残り HP:' + 【6 6】(m_hp))

# モンスターのターン
    ...
```

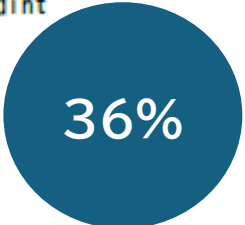
【6 5】の選択肢

- ① else
- ② and
- ③ or
- ④ ==



【6 6】の選択肢

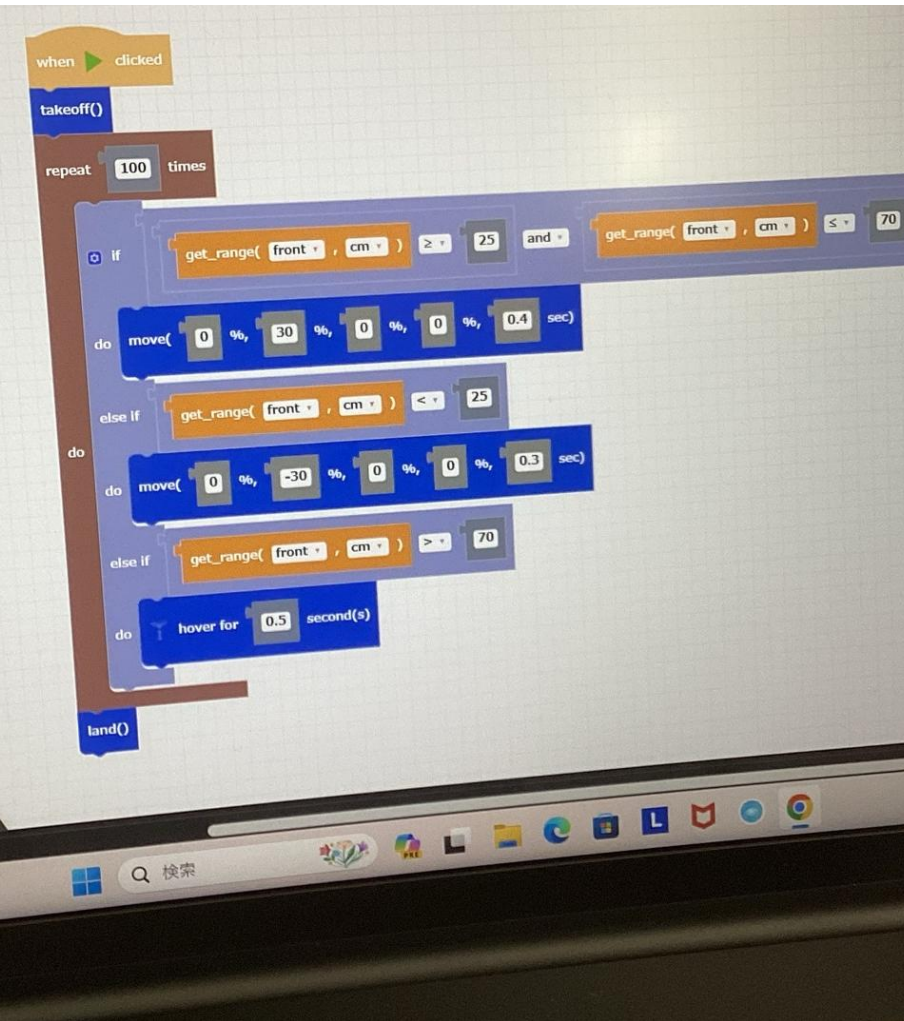
- ① str
- ② for
- ③ choice
- ④ randint



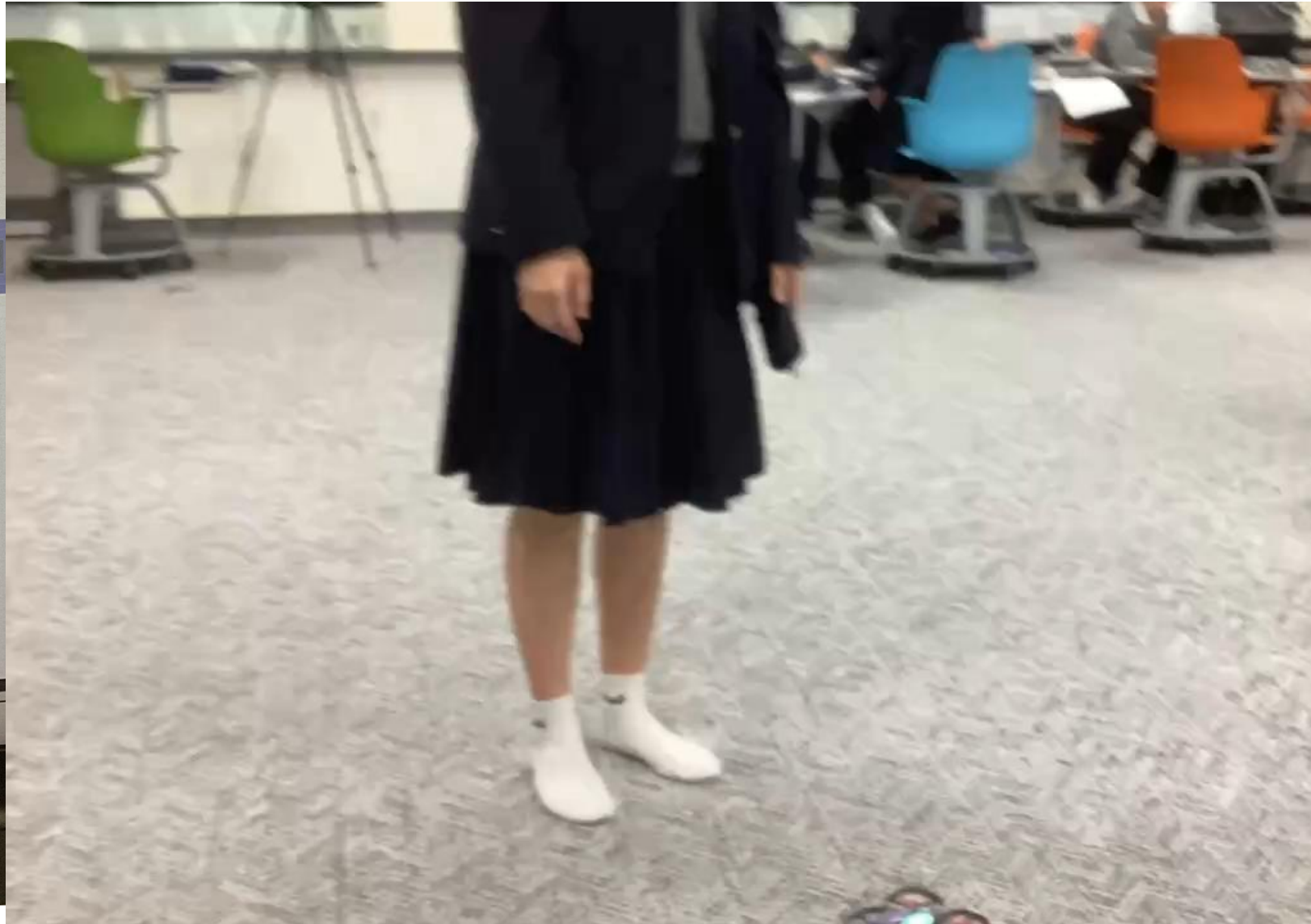
学習意欲の低い生徒たち
プログラミング = 写経作業
より深く考えさせるための工夫が必要

3. 結果及び考察

3.2 探究的な学びの評価



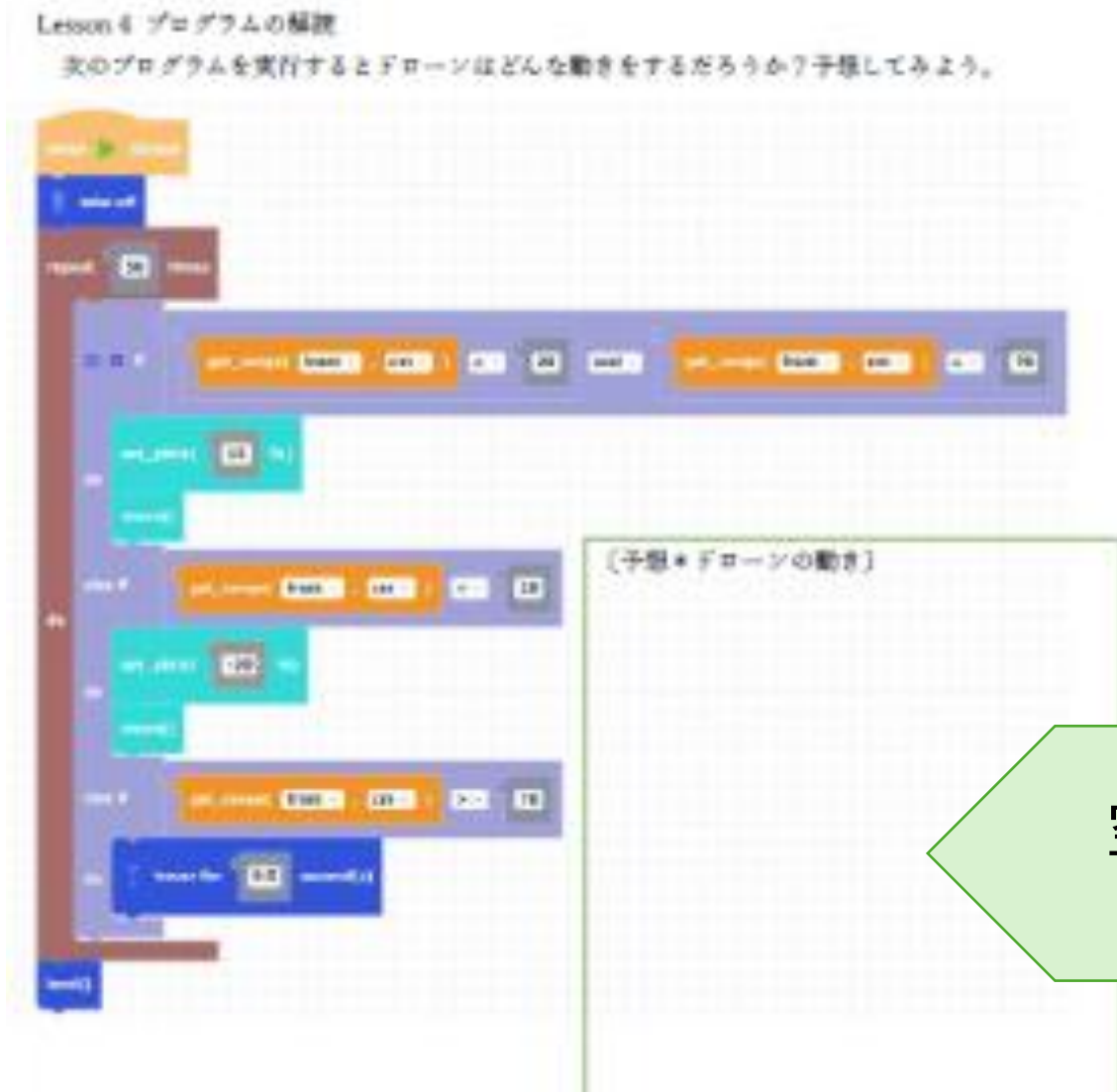
生徒が作成したプログラム



生徒の様子(左のプログラム実行中)

3. 結果及び考察 3.2 探究的な学びの評価

定期考査での評価



出題内容	正答率
1)繰り返しを使用	32%
2)条件分岐を使用	40%
3)条件の内容	34%
4)条件分岐動作	31%

空欄率12%

→プログラミングに対する興味**向上**

3. 結果及び考察

3.2 探究的な学びの評価

人生で初めてプログラミングが楽しかった。世界中の英語が分かる。
予想する困難という場面もあるけど、最終的に成功したのは良かったです。

何より楽しく頭をフル回転させている感じが良かったです。

思っていたよりもむずかしかったです。
おま本の通りに作れたと思っけていても上手くいかなかった。どこでミスしているのかを見つけるのもむずかしかったです。動かしてみてもちがうところを修整していったりやりたい動きに近づいていくのがうれしかったです。

1回目は、矢野君と一緒だったんですけど、どうやら直訳の力を、しっかりと考えました。

繰り返しが分岐を使い、フローをコントロールすることができました。
予想では分岐がなかった部分と実際に動かしてみたら、分岐がなかった部分も必要だった。roll, pitch, yaw, throttle を使う

たです。最終的に成功したから正解を見つけたのが楽しかったです。センサーが動くのも、くりしたし、人とドローンが繋がった感じがとてもおもしろかったです。

その通りに動いた時はとても達成感がありました。普段はパソコン上や画面上で動かすだけなのに、実際に目の前のドローンの動きと反映するのを見て、プログラミングの可能性の広さを改めて感じました。

プログラミングには様々な種類があり、センサーがつけることが出来るので、色々な事が出来るのが分かった。またドローンは、飛行プログラミングをした通りに動くので、それは許さずかいて分かった。

授業後の感想

- ・57%「楽しかった, 面白かった」
- ・12%「ねらったとおりに動かすことができたときの達成感」

3. 結果及び考察

3.2 探究的な学びの評価

② (2) 不明瞭な文字は採点できません。丁寧に記述すること。

take offで動き始める。
場合① 前方障害物との距離が25cm~70cmの場合、40%の力で、0.5秒間前進する。
場合② 前方障害物との距離が25cmより短い場合、35%の力で0.4秒間後退する。
場合③ 前方障害物との距離が70cmより長い場合、0.3秒間どまる。
これを10回くり返した後にlandで着陸する。

とある生徒の経過

・基礎的な学び段階

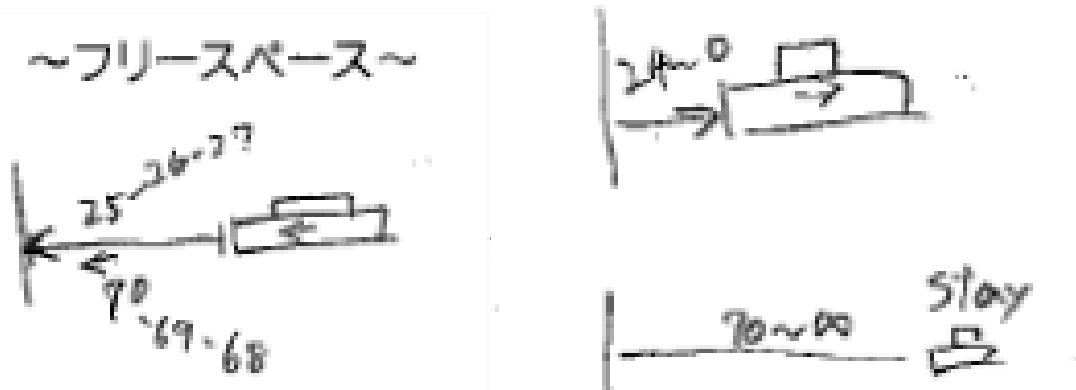
あまり理解できておらず、画面にでてくるヒントをもとに入力をして授業を進めていた

→定期考査ではあまり結果がふるわなかった

・探究的な学び段階

「最初はブロックの組み合わせや単語の意味が分からずあせっていたが、ゆっくり確認するとミスが見つかったり、スキがあったので、そこさえ見落とさなければ簡単だった」

左図の解答 → プログラミング分野満点



プログラミングの楽しさや達成感を
実感する生徒の割合増
主体的にプログラミング学習に
取り組む態度を育成できた

4. まとめ

基礎的な学び(コーディング指導) + 探究的な学び(教育用ドローンを用いた教育)

探究的な学びによって

- ・プログラミングの楽しさや達成感を実感する生徒増→主体的に取り組む態度の育成に寄与
- ・順次・分岐・繰り返しや変数などの知識の獲得が促進された

～今後の展望～ 分析対象とした実践は一事例のみ

集団の特性が影響している可能性有

→さらに多様な生徒を対象として教育実践を行い, 探究的なプログラミング教育がプログラミング的思考にどのような影響を与えるかについて明らかにしていく

参考文献

- (1)高橋信幸, 河端梓:”小中高をつなぐプログラミング的思考に関するスキーマ –プログラミング教育の中高接続に着目して–”, 日本情報科教育学会第17回全国大会論文集, pp.73-74 (2024b).
- (2)(株)ナガセ, “大学入学共通テスト「情報 I」体験模試 多くの高校生がプログラミング問題に苦戦”, NEWS RELEASE 2023.2.28, pp.1-2 (2023), https://www.toshin.com/news_release/uploadFiles/NewsReleases/2f5f2112fd98afaed02c6a0b45099203fd4cbc4db19968e468aa2f68a99feb1b20230228143738.pdf (最終アクセス2025.3.17).
- (4)Jeanette Wing; “Computational Thinking”, Communications of the ACM, 49, 3, pp.33-35 (2006).
- (5)高橋信幸, 河端梓:”情報I におけるプログラミング的思考力育成の取り組み”, 日本情報科教育学会第22回研究会研究発表論文集, pp.9-12 (2024a).
- (6)星千枝, 後藤義雄, 小田理代, 永田衣代, 赤堀侃司:”教科学習を横断するプログラミング的思考のパタン”, STEM教育研究, 1, 0, pp.19-30 (2018).
- (7)小高俊夫:“算数・数学に認知科学は役立つか:スキーマ形成の理論”, pp.1-199, 東洋館出版社 (1992).
- (8)北山浩司:”教育用ドローンを使ったプログラミング(python)教育の実践について”, 第17回全国高等学校情報教育研究会全国大会(愛知大会)大会冊子, pp.14-1 (2024)



ご静聴ありがとうございました

本研究は以下の助成により実施しました。この場を借り謝意を表します。

- ・京都府立高等学校情報教育研究会 令和6年度研究費補助
- ・科研費 課題番号23K18887