

令和元年度卒業論文

Tello を用いたプログラミング教育

京都教育大学 理科領域専攻

基礎物理学研究室

161186 倉橋 佑弥

目次

はじめに.....	3
1.1 研究の背景.....	3
1.2 研究の目的.....	4
1.3 研究の流れ.....	4
2. 理論.....	5
2.1 プログラミング教育とは.....	5
2.2 プログラミング教育の目的.....	5
3. 研究.....	6
3.1 Scratch(JavaScript)による Tello の操作.....	6
3.2 Python による Tello の操作.....	13
3.3 Pygame を用いた Joystick Controller による操作.....	16
3.4 Tello に関するアンケート調査.....	22
4. 研究成果.....	24
4.1 Scratch(JavaScript)による Tello の操作.....	24
4.2 Python による Tello の操作.....	24
4.3 Pygame を用いた Joystick Controller による操作.....	24
4.4 Tello に関するアンケート調査.....	24
5. 考察.....	28
5.1 Scratch(JavaScript)による Tello の操作.....	28
5.2 Python による Tello の操作.....	29
5.3 Pygame を用いた Joystick Controller による操作.....	29
5.4 Tello に関するアンケート調査.....	30
6. 課題と検討.....	30
7. 謝辞.....	30
8. 参考文献.....	31
9. 付録.....	32

はじめに

1.1 研究の背景

人工知能などによって第四次産業革命（Society5.0）が起こり、社会の在り方が変わることが予想されている。2018年6月、文部科学省は「Society5.0に向けた人材育成～社会が変わる、学びが変わる～」を発表した。ここには、AIとの共存社会を生き抜くために、どのような教育が必要で、生徒はどのような資質・能力を身につけなければいけないのかが書かれている。また、平成29年・30年に学習指導要領が改訂され、小学校でプログラミング教育が新たに導入されることとなった。プログラミング教育は一教科として教科化されるわけではないため、単位数や授業時間などの規定はない。他の教科と連携して組み込むこととなっている。上記の文部科学省による Society5.0 に関する報告書、新学習指導要領の二つを読み、今後プログラミング教育がいかに重要であるのかとも考えさせられ、プログラミング教育にとっても関心を持った。

また最近ドローンに関するニュースをよく目にする。例えば平昌オリンピック開幕式では、ドローン約1800台によるライトショーが行われていた。海外では、ドローンによる山岳救助や海難救助の事例がいくつかある。他にも消化用ドローン、警備用ドローンやAED輸送などの医療用ドローンなど今後ドローンがより活躍していくことが期待される。一方で、最近ではイランのソレイマニ司令官がアメリカの軍事用ドローンによって殺害されるなど、悪用されることも予想される。そのような良い面、悪い面の二面を持つドローンに私自身とても興味を持っている。また子どもたちが使い方も含めドローンのことをよく知り、どのように自らの生活に生かしていくのか考えさせる機会を設ける必要があると考える。

そのように考えていた時、偶然 iPhone のアプリケーションでドローン×プログラミングで学ぶ、プログラミング学習アプリ、DRONE STAR というアプリを見つけた。学校現場でスマートフォン端末を用いることはできないため、私は PC を用いて同じようにドローンを用いたプログラミング学習ができないかと考え、本研究を行うに至った。

1.2 研究の目的

1.1 研究の背景でも既述したように昨今の社会は変化が激しく、そのような変化に応じて子どもたちの学びも変化していかなければならない。人工知能の登場により、人は人工知能にはない読解力や創造力など、人間的な部分を育てることが求められる。また、1.1 研究の背景でも述べたようにドローンが様々な場所、用途で活躍している。このような背景を踏まえ、私の研究の目的は主に二つある。

一つ目は、ゲーム感覚で楽しみながら、子どもたちにプログラミング的思考を育むことである。文部科学省は“2030年の社会と子供たちの未来”において「グローバル化、情報化、技術革新等といった変化は、どのようなキャリアを選択するかにかかわらず、すべての子供たちの生き方に影響するものであるという認識に立った検討が必要である」と述べている。子どもたちは常に社会の変化に影響を受けるため、学校現場では子どもたちに時代にあった教育を提供することが求められる。現代の社会では人工知能などの登場により、人間的な創造性や思考力を子どもたちに育ていく必要がある。そのためプログラミングを通して、子どもたちに論理的思考力を身につけさせ、プログラミング的思考を育むことができる“プログラミング教育”が時代にあった教育であるとして、小学校段階で必修化されるようになったと考える。上述したようにプログラミング的思考を育むことは非常に重要であるが、私自身プログラミングが難しいというイメージが強いため、ゲーム感覚で楽しみながら、プログラミング的思考を育むことを目的とした。

二つ目は、ドローンとプログラミングを子どもたちが自らの生活に生かそうとする姿勢を育てることである。昨今の社会は変化が激しく、先行き不透明であるため、子どもたちは日々知識・情報を更新していかなければならない。とりわけドローン比較的新しい技術であり、ドローンやプログラミングは今後さらに私たちの生活に深く関わってくる事が予想される。そのため、子どもたちはドローンやプログラミングについて知り、自らの生活に生かせないか考えていく必要がある。単にプログラミングを通してドローンを飛ばして終わりではなく、1.1 研究の背景でものべたようなドローンの活躍を学び、また正確に動作させることができるという良さをどのように生かせるのか考えさせることもしていきたい。

1.3 研究の流れ

いくつかのプログラミング言語を用いて比較的安価なトイドローンである Tello を操作する。それぞれのプログラミング言語の特性を理解した上で、どのプログラミング言語を用いるのが適切か考察する。次に、実際にプログラミング言語を用いて操作する体験をさせ、アンケート(本学生理科領域四回生 8 名対象)を取る。実際に Tello を扱って感じたことやアンケートにより得られた情報をもとに考察でどのように学校現場に導入できるのか検討していく。課題と検討で改善点などについて考える。

2. 理論

2.1 プログラミング教育とは

プログラミング教育とは、平成 29.30 年度学習指導要領の改訂によって、小学校段階で必修化されたものである。一教科としてではなく、他の教科等と連携して行うものとして位置づけられている。プログラミング言語の使い方を教えるのではなく、プログラミング的思考を育むために実施される。また、毎回 PC やタブレット端末を用いて授業を行うのではなく、“アンプラグド”と呼ばれる、カードやパズルなどを用いて PC が動作する仕組みを学習する電子機器を使用しないプログラミング教育などもある。現在は 2020 年度からのプログラミング教育必修化に向けて教育現場では準備がなされている。

2.2 プログラミング教育の目的

プログラミング教育が小学校で必修化された理由として文部科学省から出された“【総則編】小学校学習指導要領(平成 29 年告示)解説”には「子供たちが将来どのような職業に就くとしても時代を越えて普遍的に求められる「プログラミング的思考」(自分が意図する一連の活動を実現するために、どのような動きの組合せが必要であり、一つ一つの動きに対応した記号を、どのように組み合わせたらいいのか、記号の組合せをどのように改善していけば、より意図した活動に近づくのか、といったことを論理的に考えていく力)を育むため、小学校においては、児童がプログラミングを体験しながら、コンピュータに意図した処理を行わせるために必要な論理的思考力を身に付けるための学習活動を計画的に実施することとしている。」pp85 と記述されている。つまり子どもたちにプログラミング体験を通して論理的思考力を身につけさせることが目先の目的であると言える。そして、最終的にはプログラミング的思考を育むことを目的としている。また同文献に、プログラミングのねらいについて以下のように記されている。「小学校段階において学習活動としてプログラミングに取り組むねらいは、プログラミング言語を覚えたり、プログラミングの技能を習得したりといったことではなく、論理的思考力を育むとともに、プログラムの働きやよさ、情報社会がコンピュータをはじめとする情報技術によって支えられていることなどに気付き、身近な問題の解決に主体的に取り組む態度やコンピュータ等を上手に活用してよりよい社会を築いていこうとする態度などを育むこと、さらに、教科等で学ぶ知識及び技能等をより確実に身に付けさせることにある。」pp85-86 つまり、プログラミング自体を深くまで学習する必要はないのである。上記の記述からもわかるようにプログラミングの目的は、どのように学んだかに関わらず、プログラミング的思考を持った子供の育成にあると言える。

3. 研究

3.1 Scratch(JavaScript)による Tello の操作

① Scratch のインストール(windows10)

<https://scratch.mit.edu/download/scratch2> から Windows 対応の Scratch オフラインエディターをダウンロードする。(図 1 参照)

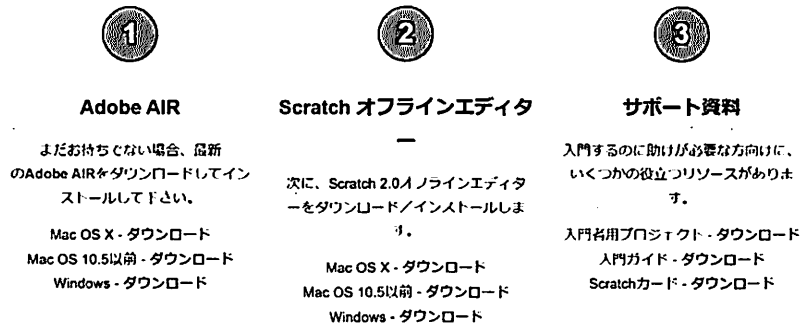


図 1 Scratch ダウンロード画面①

ファイルがダウンロードされると図 2 のような画面が出てくるので、“続行”をクリックする。

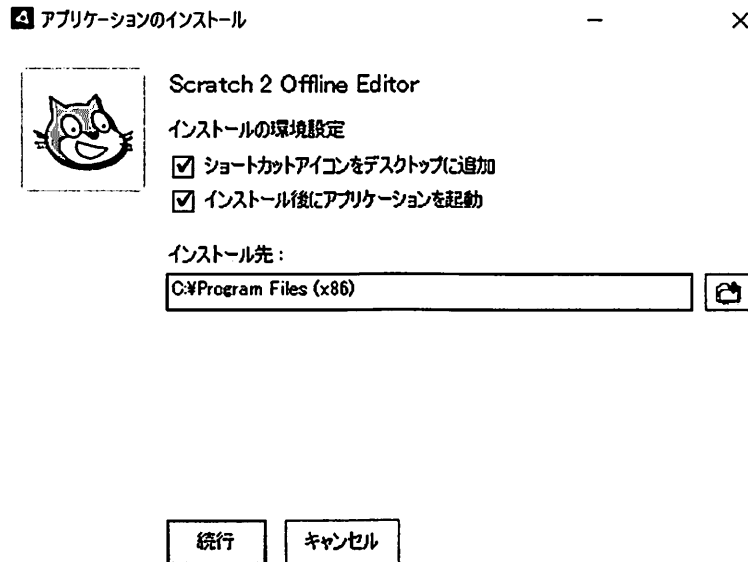


図 2 Scratch のダウンロード画面②

“このアプリがデバイスに変更を加えることを許可しますか?”という画面が出てきたら、“はい”をクリックし、ダウンロードが開始する。

② 必要なファイルのダウンロード

Tello 公式サイト <https://www.ryzerobotics.com/jp/tello/downloads> から Scratch 用 サンプルファイル(図 3 の“Scratch README”)をダウンロードする。

DOCUMENTS & MANUALS

Tello Release Notes

User Manual

Disclaimer and Safety Guidelines

Quick Start Guide - Multi

Scratch README

Tello SDK

図 3 Scratch README のダウンロード

開くと図 4 のようなファイルが表示されるので、“Click here to download”をクリックしてデスクトップに名前を付けて保存する。

Tello Scratch README

1. Visit <https://scratch.mit.edu/download> and follow the instructions to install the Scratch 2.0 Offline Editor.
2. Download and install node.js from <https://nodejs.org/en/>.
3. [Click here to download](#) Tello.js and Tello.s2e, open the terminal, go to the file directory where you saved the previous files, and type "node Tello.js"
4. Open Scratch 2.0, hold the "Shift" key, click the "File" menu, click "Import Experimental HTTP Extension," and select "Tello.s2e" file in the file directory.
5. The Tello interface will be shown in Scratch under "More Blocks."

図 4 Scratch README

デスクトップに図 5 のようなアイコンのファイルができるので 7zip でデスクトップに展開し、“Scratch_For Tello”というファイルを作成する。



図 5 ファイルのアイコン

③ Node.js のインストール

Node インストールサイト <https://nodejs.org/en/download/> から Windows Installer64-bit 版をインストールする。インストールしたアプリケーションを開くと図 6 のような画面が出てくる。



図 6 node のインストール

Next をクリックし続けると Install が表示されるので Install をクリックする。

① の時と同様に“このアプリがデバイスに変更を加えることを許可しますか?”という画面が表示されるので“はい”をクリックする。

④ Scratch の動作

Scratch2.0 を開き左上のファイルを、SHIFT キーを押しながらクリックすると図 7 のような画面が表示される。

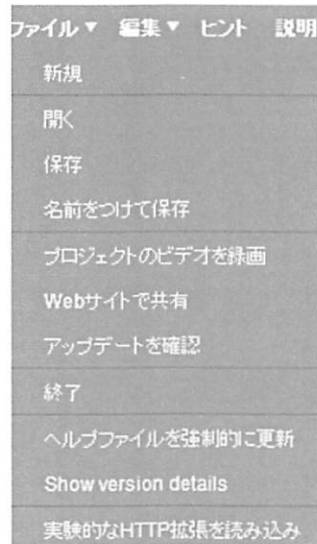


図 7 Scratch ファイル

一番下の“実験的な HTTP 拡張を読み込み”をクリックし、②でダウンロードしたファイルの Tello.s2e を読み込む。

読み込みが成功すれば Scratch2.0 スクリプトのその他に図 8 のように表示される。

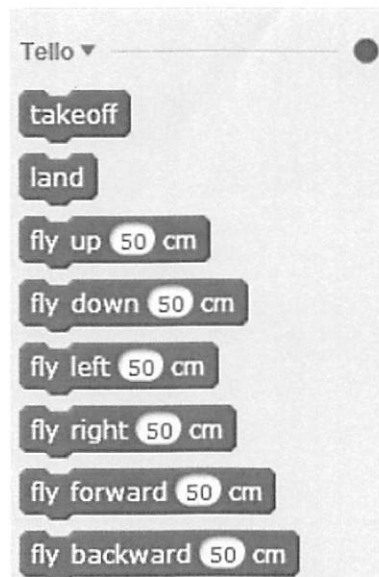


図 8 スクリプト その他

⑤ Tello の接続と操作

Tello の電源を入れ、Wi-Fi で Tello と PC を接続する。

図 9 のような Wi-Fi があるので接続する。

585143 の数字は Tello によって変わるので、Tello が複数台ある場合、どの Tello がどの番号であるか把握しておく必要がある。



図 9 Tello の Wi-Fi

Tello と PC を Wi-Fi で接続後、コマンドプロンプトというアプリケーションを開く。node と入力し、スペースキーを押し、②で展開した JavaScript ファイル、“Tello.js” をコピーしてペーストする。(ドラッグアンドドロップでも可) プログラミングが正しく動作すると図 10 のように表示される。

```
C:\> コマンド プロンプト - node C:\Users\yuya\Desktop\Scratch\Tello.js
Microsoft Windows [Version 10.0.17763.805]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\yuya>node C:\Users\yuya\Desktop\Scratch\Tello.js
-----
Tello Scratch Ext running at http://127.0.0.1:8001/
-----
```

図 10 Tello.js の起動

Tello と PC の接続が良好であれば、Scratch の図 8 の上の赤丸だったところが緑丸に変わるので、赤丸のままであれば、どこかでミスが生じていると考えられるので④の手順からもう一度やり直す。

図 11 のように緑丸であればいつでも Tello を飛ばすことができる状態である。

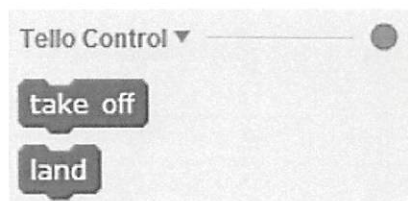


図 11 接続確認

tello.js のミスと解決を
付録 (appendix) に入れて
どうなのかな？

接続状態が良好であれば、図 12 のようにコマンド(ブロック)を組み Tello に PC から指示を与え、Tello を動作させる。

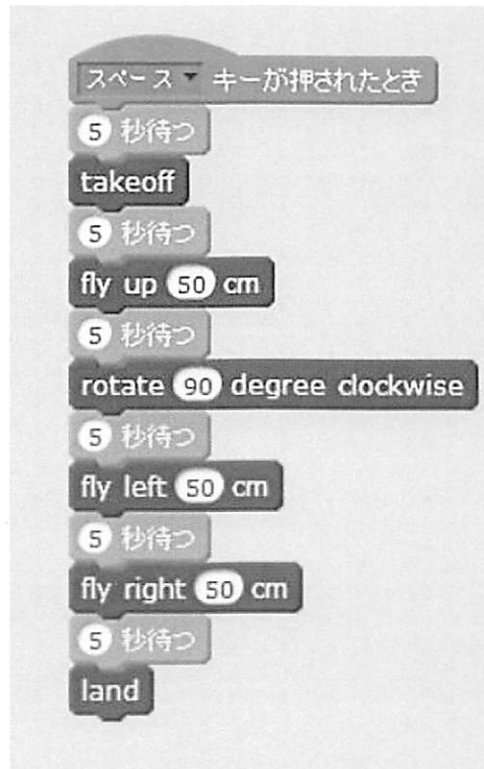


図 12 Scratch コマンド

様々なコマンドがあり、一つのコマンドに対して一つの動きをする。
コマンドを自由に組み替えることでドローンに意図した動きをさせることができる。
次ページにすべてのコマンドを記す。

Scratch のコマンド

takeoff	離陸する
land	着陸する
fly up cm	cm上昇する
fly down cm	cm 上昇する
fly left cm	cm左に移動する
fly right cm	cm右に移動する
fly forward cm	cm 前に移動する
fly backward cm	cm 後ろに移動する
rotate 90degree clockwise	時計回りに 90° 回転する
rotate 90degree counter clockwise	反時計回りに 90° 回転する
flip f,b,l,r	1 回転する
set speed to cm/s	速度を設定する
emergency stop	緊急停止する

またチェックマークを入れることで左側の画面に値を表示できる。

図 13 はチェックマークを入れるコマンドの一覧である。



図 13 Scratch コマンド

3.2 Python による Tello の操作

① Python のインストール

Python の公式サイト <https://www.python.org/downloads/> から Python の Windows 対応のものをインストールする。図 14 の“Download Python (Version)”をクリックし、実行する。図 14 では Version は 3.8.1 である。



図 14 Python のインストール①

図 15 のような画面が出てくるので“Install Now”をクリックする。“このアプリがデバイスに変更を加えることを許可しますか？”という画面が表示されるので“はい”をクリックするとインストールが始まる。



図 15 Python のインストール②

② Python用公式サンプルのダウンロード

Tello 公式サイト <https://www.ryzerobotics.com/jp/tello/downloads> を開く。

図 16 の“Tello SDK”をクリックする。

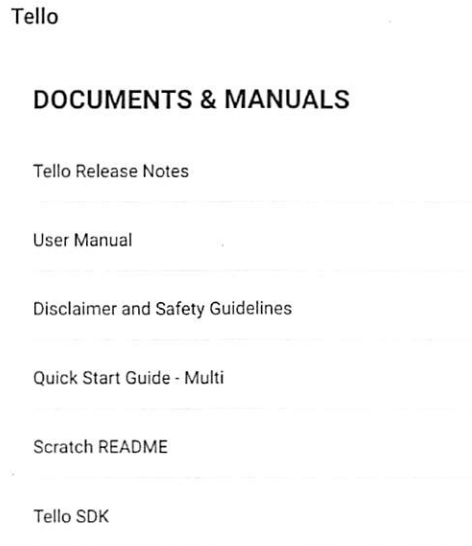


図 16 Tello Python 用サンプル

PDF ファイルの 2 ページ目の“Click here”をクリックする。開かれたファイルが今回用いる公式サンプルである。上記の URL からダウンロードするよりも、下記の URL からダウンロードする方が簡単であるので表記する。

<https://qiita.com/fumimatsu/items/5efd4cea02b40a517d69>

このサイトから Tello3.py をダウンロードする。ダウンロード後、Tello3.py を起動する。図 17 のような画面が表示される。



図 17 Tello3.py の起動

⑥ Tello の動作

Tello3.py を起動した後、3.1⑤と同様に Tello と PC を Wi-Fi で接続する。

図 18 のように command と入力し、ok が表示されれば、接続が完了している。

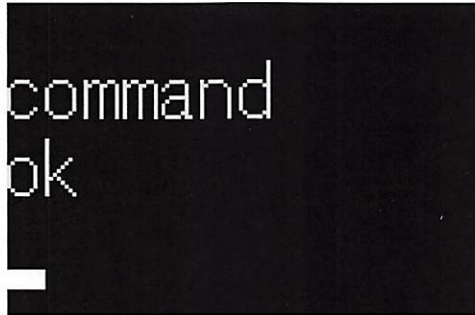


図 18 Tello の接続確認

コマンドを打ち込み、ドローン进行操作する。コマンド一つにつき一つの動作を行う。

ドローンの動きを見ながら、コマンドを打ち込むことができる。

以下にコマンドを示す。

Python のコマンド

command	接続確認
takeoff	離陸する
land	着陸する
flip l,r,f,b,bl,rb,fl,fr	1 回転する
forward 20-500	前に移動する
back 20-500	後ろに移動する
left 20-500	左に移動する
right 20-500	右に移動する
up 20-500	上昇する
down 20-500	下降する
cw 1-3600	時計回りに回転する
ccw 1-3600	反時計回りに回転する
speed 1-100	速度を設定する
speed?	速度を確認する
time?	飛行時間を確認する
battery?	Tello の充電を確認する

このコマンドは、Tello の動作を確認するためのコマンドです。
Tello の動作を確認するためのコマンドは、command と入力し、ok が表示されれば、接続が完了している。



[Faint, illegible text in the upper central area of the page, possibly a header or title block.]

[Faint, illegible text in the middle section of the page, possibly a main body of text.]

[Faint, illegible text in the lower section of the page, possibly a footer or concluding text.]

3.3 Pygame を用いた Joystick Controller による操作

① Miniconda のインストール

Miniconda の公式サイト <https://docs.conda.io/en/latest/miniconda.html> を開く。

図 19 の Miniconda3 Windows 64-bit をクリックする。

Windows installers

Windows			
Python version	Name	Size	SHA256 hash
Python 3.7	Miniconda3 Windows 64-bit	51.5 MiB	f18060cc0bb50ae75e4d602b7ce35197c0e31e81288d069b758594f1tb4e
	Miniconda3 Windows 32-bit	54.0 MiB	7c30778941d2bba03531ba269a78a108b01fa366530290376e7c3b467f3c
Python 2.7	Miniconda2 Windows 64-bit	50.9 MiB	8647c54058f11842c37854edeff4d20bc1fbdad8b88d9d34d76fda1630e6
	Miniconda2 Windows 32-bit	48.7 MiB	0d106228d6a4610b599df965dd6d9bb659329a17e3d693e3274b20291a7c

図 19 Miniconda のインストール①

Miniconda のインストールを実行すると図 20 のような画面が表示される。

Next をクリックする。“I agree”や“Next”をクリックし続けると“Install”が表示されるのでクリックする。



図 20 Miniconda のインストール②

“このアプリがデバイスに変更を加えることを許可しますか？”という画面が表示されるので“はい”をクリックするとインストールが始まる。

② Visual Studio のインストール

<https://visualstudio.microsoft.com/ja/downloads/?rr=https%3A%2F%2Fdocs.microsoft.com%2Fja-jp%2Fvisualstudio%2Finstall%2Finstall-visual-studio%3Fview%3Dvs-2019> 上記の URL から Visual Studio 2019 をダウンロードする。図 21 の Community 用の無料ダウンロードを実行する。

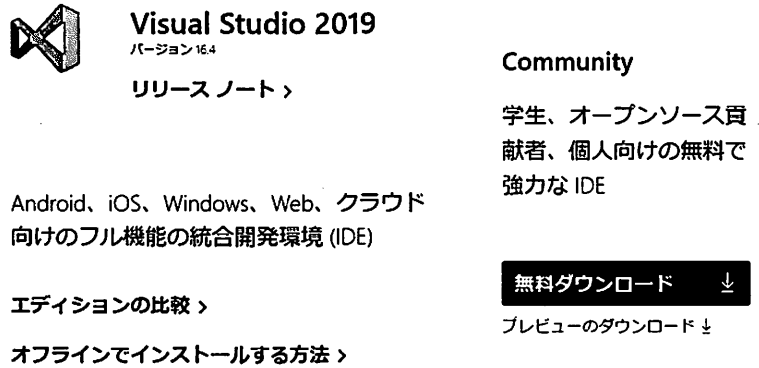


図 21 Visual Studio のダウンロード①

“このアプリがデバイスに変更を加えることを許可しますか?”という画面が表示されるので“はい”をクリックする。インストールが完了すると、図 22 のような画面が表示される。

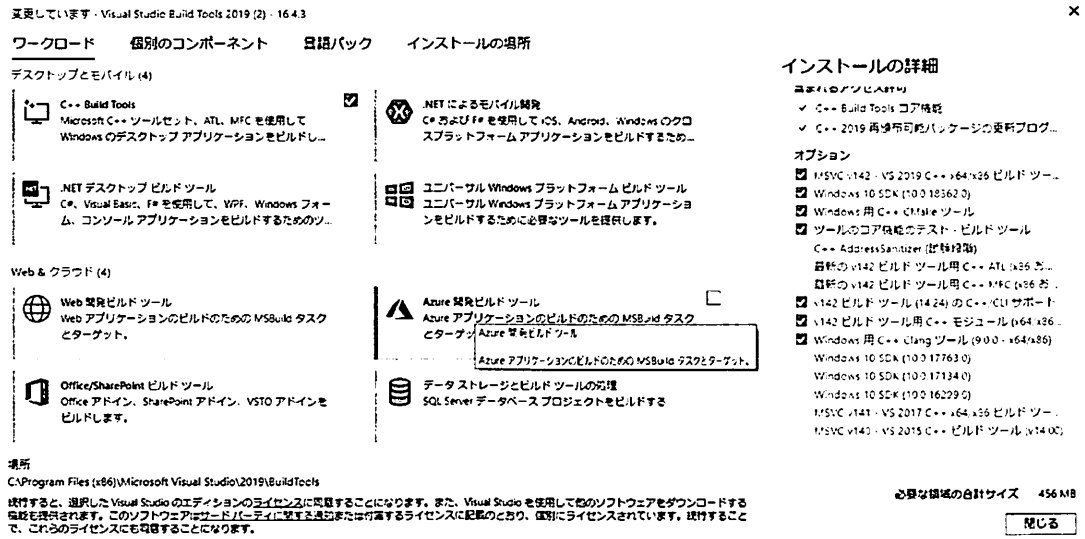


図 22 Visual Studio のダウンロード②

インストールの詳細のチェックマークが入っているところすべてにチェックを入れ、インストールする。

③ ライブラリのインストール

Anaconda Powershell Prompt(Miniconda3)を開き、以下のコマンドを実行する。

pip install tellopy

conda install av

pip install opencv-python

pip install image

図 23 のようにライブラリがインストールされる。

```

Anaconda Powershell Prompt (Miniconda3)
(base) PS C:\Users\yuya> pip install tellopy
Collecting tellopy
  Using cached https://files.pythonhosted.org/packages/fc/bf/03b1faed7cc401f6c508c2aafd739c66d43b9679279e996a63d7e2b56f2b/tellopy-0.6.0-py2.py3-none-any.whl
Installing collected packages: tellopy
Successfully installed tellopy-0.6.0
(base) PS C:\Users\yuya> conda install av
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\yuya\Miniconda3

  added / updated specs:
    - av

The following packages will be downloaded:

  package | build | size
  -----|-----|-----
  conda-4.8.1 | py37_0 | 2.8 MB
  -----|-----|-----
  Total: | | 2.8 MB

The following packages will be UPDATED:

  conda | conda-forge::conda-4.7.12-py37_0 --> pkgs/main::conda-4.8.1-py37_0
  openssl | conda-forge::openssl-1.1.1d-hfa6e2cd_0 --> pkgs/main::openssl-1.1.1d-he7
74522_3

The following packages will be SUPERSEDED by a higher-priority channel:

  ca-certificates | conda-forge::ca-certificates-2019.11.~ --> pkgs/main::ca-certificates-20
19.11.27-0
  certifi | conda-forge --> pkgs/main

Proceed ([y]/n)?

Downloading and Extracting Packages
conda-4.8.1 | 2.8 MB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

```

図 23 ライブラリのインストール

ライブラリがすでにインストールされている場合は、図 24 のように Requirement already satisfied: ~ in 保存場所と表示される。

```
Anaconda Powershell Prompt (Miniconda3)
(base) PS C:\Users\yuya> conda install av
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.

(base) PS C:\Users\yuya> pip install opencv-python
Requirement already satisfied: opencv-python in c:\users\yuya\miniconda3\lib\site-packages (4.1.2.30)
Requirement already satisfied: numpy>=1.14.5 in c:\users\yuya\miniconda3\lib\site-packages (from opencv-python) (1.17.4)
(base) PS C:\Users\yuya> pip install image
Requirement already satisfied: image in c:\users\yuya\miniconda3\lib\site-packages (1.5.27)
Requirement already satisfied: pillow in c:\users\yuya\miniconda3\lib\site-packages (from image) (6.2.1)
Requirement already satisfied: django in c:\users\yuya\miniconda3\lib\site-packages (from image) (2.2.7)
Requirement already satisfied: pytz in c:\users\yuya\miniconda3\lib\site-packages (from django->image) (2019.3)
Requirement already satisfied: sqlparse in c:\users\yuya\miniconda3\lib\site-packages (from django->image) (0.3.0)
```

図 24 既存のライブラリのインストール

④ 環境構築の確認・テスト飛行

① ~ ③の手順が正確にできていることを確認する。

Scratch(JavaScript)による Tello の操作の⑤の時と同様に Tello の電源を入れ、Wi-Fi で Tello と PC を接続する。接続後、Anaconda Powershell Prompt (Miniconda3) を開き、以下のコマンドを実行する。

```
python -m tellopy.examples.simple_takeoff
```

環境構築が正しく行われていれば、Tello が離陸する。Tello が離陸しなければ、どこかでミスをしている可能性が高いので、もう一度①~③の手順をやり直す。正しくコマンドを実行すれば、図 25 のように表示される。

```
選択 Anaconda Powershell Prompt (Miniconda3)
(base) PS C:\Users\yuya> python -m tellopy.examples.simple_takeoff
Tello: 10:34:25.519: Info: start video thread
Tello: 10:34:25.519: Info: send connection request (cmd="conn_req:9617")
Tello: 10:34:25.519: Info: video receive buffer size = 524288
Tello: 10:34:25.519: Info: state transit State::disconnected -> State::connecting
Tello: 10:34:25.535: Info: connected. (port=9617)
Tello: 10:34:25.535: Info: send time (cmd=0x46 seq=0x01e4)
Tello: 10:34:25.535: Info: state transit State::connecting -> State::connected
Tello: 10:34:25.535: Info: set altitude limit 30m
Tello: 10:34:25.535: Info: takeoff (cmd=0x54 seq=0x01e4)
Tello: 10:34:25.581: Info: recv: ack: cmd=0x54 seq=0x0000 cc 60 00 27 b0 54 00 00
00 00 a1 81
ALT: 0 | SPD: 0 | BAT: 17 | WIFI: 0 | CAM: 0 | MODE: 6
ALT: 0 | SPD: 0 | BAT: 17 | WIFI: 0 | CAM: 0 | MODE: 6
ALT: 0 | SPD: 0 | BAT: 17 | WIFI: 0 | CAM: 0 | MODE: 6
```

図 25 Tello のテスト飛行

⑤ Joystick Controller による Tello の操作

Joystick Controller と PC を USB ケーブルで接続する。また、Tello も Wi-Fi で PC と接続する。コマンドプロンプトというアプリケーションを開き下記のコマンドを実行する。

```
python -m tellopy.examples.joystick_and_video
```

コマンドがうまく実行されていれば、図 26 のような画面が表示される。

```
Anaconda Powershell Prompt (Miniconda3)
(base) PS C:\Users\yuyua> python -m tellopy.examples.joystick_and_video
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
joystick name: Wireless Controller
tello: 20:04:29.293: Info: start video thread
tello: 20:04:29.294: Info: send connection request (cmd="conn_req:9617")
tello: 20:04:29.294: Info: video receive buffer size = 524288
tello: 20:04:29.294: Info: state transit State:disconnected -> State::connecting
tello: 20:04:29.294: Info: start video (cmd=0x25 seq=0x01e4)
tello: 20:04:31.294: Info: send connection request (cmd="conn_req:9617")
tello: 20:04:33.295: Info: send connection request (cmd="conn_req:9617")
tello: 20:04:34.723: Info: set_roll(val=-0.13)
tello: 20:04:34.723: Info: set_pitch(val=0.09)
tello: 20:04:34.733: Info: set_roll(val=-0.26)
tello: 20:04:34.733: Info: set_pitch(val=0.15)
tello: 20:04:34.743: Info: set_roll(val=-0.34)
tello: 20:04:34.743: Info: set_pitch(val=0.20)
tello: 20:04:34.753: Info: set_roll(val=-0.45)
tello: 20:04:34.753: Info: set_pitch(val=0.27)
tello: 20:04:34.763: Info: set_roll(val=-0.52)
tello: 20:04:34.763: Info: set_pitch(val=0.28)
tello: 20:04:34.774: Info: set_roll(val=-0.59)
tello: 20:04:34.774: Info: set_pitch(val=0.31)
tello: 20:04:34.784: Info: set_roll(val=-0.60)
tello: 20:04:34.784: Info: set_pitch(val=0.33)
tello: 20:04:34.805: Info: set_roll(val=-0.61)
tello: 20:04:34.805: Info: set_pitch(val=0.32)
tello: 20:04:34.815: Info: set_roll(val=-0.60)
tello: 20:04:34.826: Info: set_roll(val=-0.54)
tello: 20:04:34.826: Info: set_pitch(val=0.33)
```

図 26 Joystick Controller 接続良好

コマンド実行後に、Joystick Controller の R1 ボタンを押すと Tello が離陸する。離陸しない場合は、Joystick Controller か Tello のどちらかの接続が悪いため、もう一度接続からやり直す。接続がうまくいっていないときには図 27 のような画面が表示される。図 27 は no supported joystick found からわかるように、Joystick Controller が接続されていない状態である。

```
Anaconda Powershell Prompt (Miniconda3)
(base) PS C:\Users\yuyua> python -m tellopy.examples.joystick_and_video
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
no supported joystick found
```

図 27 Joystick Controller 接続不良

Joystick Controller のボタンとコマンド

R1	離陸する
L1	着陸する
△	前進する
×	後退する
○	右に進む
□	左に進む
↑	上昇する
↓	下降する
→	右に回転する
←	左に回転する
右スティック	△×○□に対応
左スティック	↑↓→←に対応

3.4 Tello に関するアンケート調査

① アンケートの作成

3.1～3.3 でおおむねどのようなプログラミング×ドローンの授業が現場で実践できそうか検討がついたため、その授業が現実的にできるかどうかを確認するためのアンケートを作成した。本学理科領域四回生の学生 8 人を今回のアンケートの対象とした。学校現場で用いるためにはドローンに慣れていない人対象でもできることが条件であるので、ドローン経験がない学生 8 人を選抜した。作成したアンケートは別途付録に添付する。

② アンケート調査

アンケート調査をする前に、まず大まかにどのような授業展開にしたいのか説明した。説明した内容は以下に記述する。

・高等学校と小学校が連携したプログラミング授業について

教育新聞の記事で「小学生が高校生からプログラミングを教わりながら、AI やロボットの可能性について体験的に学ぶ授業が 11 月 18 日、埼玉県立浦和北高校で行われた。」という記述を見つけた。この記事を受け、ドローン×プログラミングでも小高連携で授業をできるのではないかと考えた。

・ドローンの可能性について

アンケートを実施した際は、まだイランのソレイマニ司令官がアメリカの軍事用ドローンによって殺害される事件は起こっていなかったが、海外のドローンによる救助などはすでに成功例がいくつかあったので、ドローンの活躍について説明した。

・ドローンと生活とのつながりについて

ドローンの可能性を何も考えずに子供たちに伝えてもドローンと生活とのつながりについて考えるきっかけにならないと考えた。そのため、先にワークシートなどで、ドローンの正確に操作できるという良さをどのように生かすことができるのか考えさせる大まかな授業展開を説明した。

その後 3.3 の⑤の段階からドローンとプログラミングについて体験してもらい、アンケートを実施した。

③ アンケートの項目について

1. 今回の活動を通してプログラミングに興味を持つことができる。

ドローンを初めて体験した体験者がプログラミングに興味を持つことができたか確かめるためにこの項目を設定した。

2. 今回の活動を行うことで子どもたちがプログラミングに興味を持つきっかけになる。

プログラミングを意欲的に取り組む子どもたちを増やしたいためこのような項目を設定した。この項目でそう思わない、全くそう思わないなどの意見が出てきたとすれば、考え直す必要がある。

3. 高校生が小学生対象に本教材を用いることができる。
既述したように高等学校、小学校の連携のプログラミング教育の実例がある。そのためドローンを用いたプログラミング教育においても小高連携ができないかと考えこの項目を設定した。
4. 小学生がドローンを扱うのは簡単だ。
現在小学校のプログラミング教育では、先にコマンドを組み込んだロボットなどを動かすことが主流である。同様のことが Scratch による操作でも可能である。子どもたちが意図する一連のドローン動作を実現するために、一つ一つの動きに対応したコマンド(パズル)を、どのように組み合わせたらいいのか、コマンドの組合せをどのように改善していけば、より意図したところにドローンを動かすことができるのか、といったことを論理的に考えていく力が育つと考えられる。しかし、コマンドを組み込んだ後、ドローンを実際に動かすまでに若干のラグがあるため、ドローンを動かしたという実感がわきにくいと考えた。一方で JoystickController による操作では、押したボタンに応じてドローンが一つの動作をするため、子どもたちに実感がわきやすいと考えた。JoystickController によるドローンの扱いが小学生にとっても簡単であれば、指導案に組み込んでいきたいと考えたため、この質問項目を設定した。
5. 現場でもドローンを用いたプログラミング教育を導入していくべきだ。
どれだけ興味関心が惹けたとしても、現場に導入していくことができなければ本末転倒である。そのためこの項目を設定した。
6. プログラミングの仕組みを知るだけでなく、どのように生活に活かしていくか考えることができる。
1.2 研究の目的でも述べたように、ドローンの可能性に子どもたちが気づき、自らの生活に生かそうとする態度を育てたいと考えている。そのため、この項目を設定した。
7. ドローンの可能性について体験的に学ぶことができる。
プログラミング言語によって正確に動作する Tello を扱うことで、ドローンの可能性に気付くことができるかどうか確かめるためにこの項目を設定した。

4. 研究成果

4.1 Scratch(JavaScript)による Tello の操作

Scratch を用いて Tello は正常に動作した。しかし、狭い場所で動作させるには少し不安がある。Tello 自体が高さを認識するため、地面が平らではなく、机などいろいろなものがある環境で扱うと、コマンドに反して自動で高さが変動してしまう。先に決めたコマンド通りに動かすため、動作中の修正が効かない。

4.2 Python による Tello の操作

Python を用いて Tello は正常に動作した。しかし、Scratch の時と同様狭い場所で動作させるには不安がある。Scratch とは違い、コマンドを入力して、Tello を動作させるため、タイピングが遅い人が Python を用いて動作させると時間がかかる。一方でコマンドを入力するとすぐに動作するため、修正が容易にできる。

4.3 Pygame を用いた Joystick Controller による操作

Pygame を用いて Tello を操作することができた。JoystickController のスティックの反応が少し悪いが、スティックを押し倒しすぎると急に動き出すため、気を付ける必要がある。三つの中で一番軌道修正するのが容易である。

4.4 Tello に関するアンケート調査

1. 今回の活動を通してプログラミングに興味を持つことができる。

問①

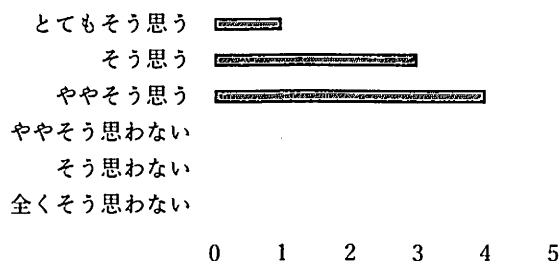


表 1

2. 今回の活動を行うことで子どもたちがプログラミングに興味を持つきっかけになる。

5. 考察

実験手順に沿って、様々なプログラミング言語を用いて Tello を操作したときに私自身が感じたことや、実際に学生に扱ってもらい頂戴した意見をもとに、考察ではどのようにプログラミングの教材としてトイドローンの Tello を用いることができるのかについて言及する。

・ ドローンについて

ドローンの下に手をかざしたときに、ドローンが手からの距離を一定に保とうと、上昇したことから、ドローン自体に高さを認識するセンサーがついているのではないかと考える。しかし、壁などにぶつかったりしていたため、高さ以外は認識していないと考える。このドローン自身の機能によってコマンドで意図した動作に反して、高さが上昇してしまうことがある。そのため、意図したい動作の中に高さに関わるコマンドを組み込むことは難しいと考える。またドローンの羽根は非常に早く回転しているため、子どもたちが扱うとき安全面に注意を払う必要がある。ドローンの充電は減るのがはやく、研究中ドローンが充電切れになってしまったことが多々ある。そのため、学校現場で用いるには予備のバッテリーを用意するか、授業前に充電を済ませておかなければならない。Python、Scratch をどちらのプログラミング言語を用いても、充電を確認することができるので、授業の導入段階で充電の残量確認の方法を伝えることも一つの手段として考えられる。

5.1 Scratch(JavaScript)による Tello の操作

Scratch による Tello の操作では、実験結果にも既述したように Scratch の場合は先にすべてのコマンドを決めた後に、Tello に PC から指示を与え、組み合わせたコマンド通りに動作させるという仕組みで動いている。そのため、一連の動作中に新たにコマンドを与えることや、位置を修正することが不可能である。このような特徴を持つ Scratch を教材として扱う際には、“Tello を決められた場所に移動させる”などではなく、“指令通りに Tello を動作させるためにコマンドを組み立てる”といったような教材の方が望ましい。例えば、次のような例が考えられる。

「各班に”Tello を離陸後、上昇させて左に 90° 回転させる。”などの指令書を渡し、Scratch のアプリケーションで指令書通りに Tello が動作するようにコマンドを組み立てる。その後班ごとに、教室前に出て PC と Tello を接続し、コマンドを実行する。コマンドを実行後、他の班はどのような指令書であったかを当てる。」

このような流れで授業をすることで、生徒たちは指令書通り(決まった動作)にどのコマンドを組み立てるのが良いかなど考える。また、組み立てたコマンド通りに Tello を動作させることができたとき、他の班がどのような指令であったのか当てることができる。そして生徒達にプログラミング的思考が育まれると考える。

5.2 Python による Tello の操作

Python による Tello の操作では、Scratch による Tello の操作とは異なり、コマンドを入力するごとに一つの動作をする。そのため、細かな修正がきく。Python を用いたときには、Scratch を用いたときよりも決められた場所に Tello を移動させることが容易である。容易とはいえ、正確に決められた場所に移動するのは Joystick Controller を用いたときに比べると、その正確さは劣る。また、Python を用いた場合コマンドを入力するのに時間がかかる子どもたちにとっては少し扱いづらいのではないかと考える。以上のことを踏まえ、Python を用いて Tello を動作させることはあまりプログラミング教育に向かないのではないかと考える。Python を用いるとするならば、演示で用いるのが良いだろう。例えば、次のような使い方が挙げられる。

PC とプロジェクターをつなぎ、コマンドプロンプトのコマンド入力画面がクラス全体に見えるようにする。次に先生が前でコマンドを入力していき、入力したコマンド通りに動作した Tello を観察させる。一つのコマンドにつき一つの動作をする Tello を観察することで、プログラミングの仕組みについても少しわかるのではないかと考える。興味をひくための導入として用いることができると考える。

5.3 Pygame を用いた Joystick Controller による Tello の操作

Pygame を用いた JoystickController による Tello の操作は、3つの中で最も正確に操作することができる。この特性を生かし、以下のような実践方法が挙げられる。

設定 ドローン(Tello)は救助用もしくは補給物資運搬用のドローンとする。

ドローンを救助場所もしくは、補給物資届け先などの定めた場所に移動させる。

このように、Tello (ドローン) の正確さにふれることで、プログラミング的思考のみならず、研究目的であった Tello の可能性について考えることにもつながる。

しかし、はじめて使ったときは操作が少し難しいように感じたため、操作の工夫などについて少し検討していく必要がある。また目的の場所に移動させるためにコントローラをどのように動かせばよいかなど考えることがプログラミング的思考の育成につながるが、少しゲーム感覚の要素が強くなってしまふことが課題として挙げられる。

5.4 Tello に関するアンケート調査

研究成果のアンケート結果からもわかるように、“4.小学生がドローンを扱うのは簡単だ”の項目でややそう思わないという意見やそう思わないという意見があった。この意見から 5.3 でも既述したように Tello の操作が小学生にとって難しい可能性がある。また PS4 のコントローラを用いたため、自由記述のところで、「PS4 では移動はすべて左スティックで行うが、今回の Tello の操作は右スティックで行うため、普段ゲームをしている人からするとやりにくい」という指摘があった。この指摘通りに既存のプログラミングを書き換えて左スティックで移動するように組み替えることも必要になってくるのではないかと考えた。金銭面や安全面について危惧する意見もあったが、金銭面については班に一台とかではなく、クラスに一台で授業すること、または興味を持っている子どもたちを対象を絞り、行うことなどで解決できる。安全面については、事前に注意させることが必要である。

6. 課題と検討

同じ教室で数台の Tello を動作させることは現実的には厳しい。Tello 同士がぶつかってしまう危険性があることや、Tello の Wi-Fi 接続が難しいことが理由として挙げられる。また、Tello の価格は一万円程度であるため、何台も用意するのは予算的にも厳しい。そのため、授業に Tello を用いる際はクラスに一台が妥当であるだろう。つまり、JoystickController を用いた操作でプログラミング教育を行うことは難しい。そのため、クラスに一台 Tello があればプログラミング教育を行うことができるような工夫などを今後考えていきたい。

また実際に救助などに使われているドローンと今回用いたの Tello との最も大きな違いは操縦者がドローンを目視できているかどうかである。実際救助などに使われているドローンは、人が立ち入ることができない場所に行くこともあるため、操縦者がドローンを目視することができない。操縦者はドローンに搭載されているカメラから画像情報を PC で読み取り、操縦している。子どもたちにドローンの良さについてさらに知ってもらうために、Tello でも画像認識できるように今後取り組みを進めたい。Tello にもカメラが搭載されているため、PC で Tello 目線の映像をみることは可能である。画像認識ができるようになれば、もっとプログラミング教育の可能性も広がるだろう。

7. 謝辞

本研究を進めるにあたって様々な助言をしていただいた指導教員の高嶋隆一教授に感謝いたします。また、本研究のアンケートに協力していただいた本学学生の皆さんにも感謝しております。

8. 参考文献

「2030年の社会と子供たちの未来」文部科学省

「【総則編】小学校学習指導要領(平成29年告示)解説」文部科学省