令和 4 年度　　卒業論文

幼稚園児の運動と集中力を関係を解析するためのシステムの開発

指導教員 熊澤典良 准教授
上谷俊平 教授

鹿児島大学 機械工学科
パクテウォン

目 次

第 1 章

序論

1.1　研究の背景と目的

一般的に運動と集中力の間には関係性があると知られいる．得に運動後には集中力が向上されるため，保育や教育には運動は積極的に採用されている．幼稚園児の運動量と集中力を計測し，個人の園児に適した運動内容と運動強度を提案することで，より効率的な集中力向上の効果を得ることができる．

先行研究では，Fig.1.1に示したような NFC 通信活動量計 MTN240 を用いて運動量の測定を行った．活動量計は小型の機器であり，着用することで METs，補数，消費カロリー等の測定ができる．集中力の測定は読み聞かせ時に行い，Fig.1.2 に示したような三つの AR マーカを付着した帽子を用いて頭部動揺の状況を測定することで集中度合いを評価した．

運動量と集中力の測定は幼稚園で実行される為，測定したデータの保存・表示・解析を自動化することができるシステムが必要である．本研究では，幼稚園児の運動と集中力データの自動保存と表示を行うことでデータの可視化と運動と集中力の関係性を判断することを目的とする．

Fig.1.1: NFC 通信活動量計 MTN240

Fig.1.2: AR マーカを付着した帽子

1.2 本研究の概要

本論文は全 4 章で構成されており，第 2 章では，本研究の実験環境について図とともに示す.

第 3 章では，活動量計のデータの MySQL に自動保存するシステムの開発と読み聞かせ時間帯の判定と保存について述べる.

第 4 章では，自動保存したデータを基に，運動と集中力の関係性を解析するために必要なデータの選定と選定したデータを Web 上に表示した結果を示す.

第 2 章

実験環境

2.1 活動量計

本研究では，吉田南幼稚園の年長児童 (5，6 歳児)29 名を対象に実験を行った．運動量の測定は Fig.1.1 に示したようにアコース社の NFC 通信活動量計 MTN240 を用いた．

活動量計は，最大 10 日分の 1 分毎の METs 値を保管する．METs は metabolic equivalents(代謝当量) の略であり，運動強度を示す指標である．安静時を 1 とした時と比較して何倍のエネルギを消費するかを示す身体活動量である．

また，最大 1ヶ月分の 1 時間毎の歩数 [歩]，アクティブ歩数 [歩]，総消費カロリー [kcal]，活動消費カロリー [kcal]，アクティブ活動時間 [秒]，活動時間 [秒] を保管する．

最後に，最大 90 日分の 1 日毎歩数 [歩]，アクティブ歩数 [歩]，総消費カロリー [kcal]，活動消費カロリー [kcal]，アクティブ活動時間 [分]，活動時間 [分] 距離 [km]，Ex[METs×時間] を保管する．アクティブ活動の基準は事前に閾値を設定することができる．本研究では 4METs を閾値として設定した．

活動量計の保管されているデータを csv ファイルで出力するには，Fig.2.1示したようなソニー社の非接続 IC カードリーダー/ライター RC-S380 をパソコンに接

続し，機器の上に活動量計を載せる．その後にアコース社が開発したアプリケーションを起動することで，csv ファイルに保存することができる．

Fig.2.1: 非接続 IC カードリーダー/ライター RC-S380

2.2 読み聞かせ時の計測カメラ

読み聞かせ時の計測は Fig.2.2に示したような H.VIEW 社の POE IP カメラ (HV-PTZ501) を用いて計測する．カメラは読み聞かせが実施される部屋の角に 1 台ずつ設置されている．各自のカメラには Fig.2.3，Fig.2.4，Fig.2.5，Fig.2.6に示したように 1 から 4 までの番号を付与した．

Fig.2.2: POE IP カメラ (HV-PTZ501)

Fig.2.3: カメラ 1

Fig.2.4: カメラ 2

Fig.2.5: カメラ 3

Fig.2.6: カメラ 2

第 3 章

データの自動保存

3.1　運動量データの自動保存

データの保存にはFig.3.1に示すように，幼稚園に設置されているSkynew社の小型パソコンW4を使用した．パソコンにインストールされているMySQLというデータベースに収得したデータを保存する．

Fig.3.1: 小型パソコン

活動量計データの計測と保存は幼稚園で先生方によって実行される為，幼稚園で使用できる自動保存システムの開発を行った．システムの開発にはPythonのTkinterというGUIライブラリを使用した．運同量データは名前毎のフォルダに分けている為，Fig.3.2に示したように複数のフォルダのドラッグ＆ドロップ機能

を作成した．次に，重複のファイルを保存させない為，Fig.3.3に示したようにログファイル作成することで，更新ファイルの検知を行った．最後に scp というファイル転送プロトコルを用いて小型パソコンに原本の csv ファイルを転送した．

データベースに保存する内容は，時間，活動量計 ID，1 分毎のデータである METs，1 時間毎のデータである歩数 [歩]，アクティブ歩数 [歩]，総消費カロリー [kcal]，アクティブ消費カロリー [kcal]，アクティブ活動時間 [秒]，活動時間 [秒] を保存した．

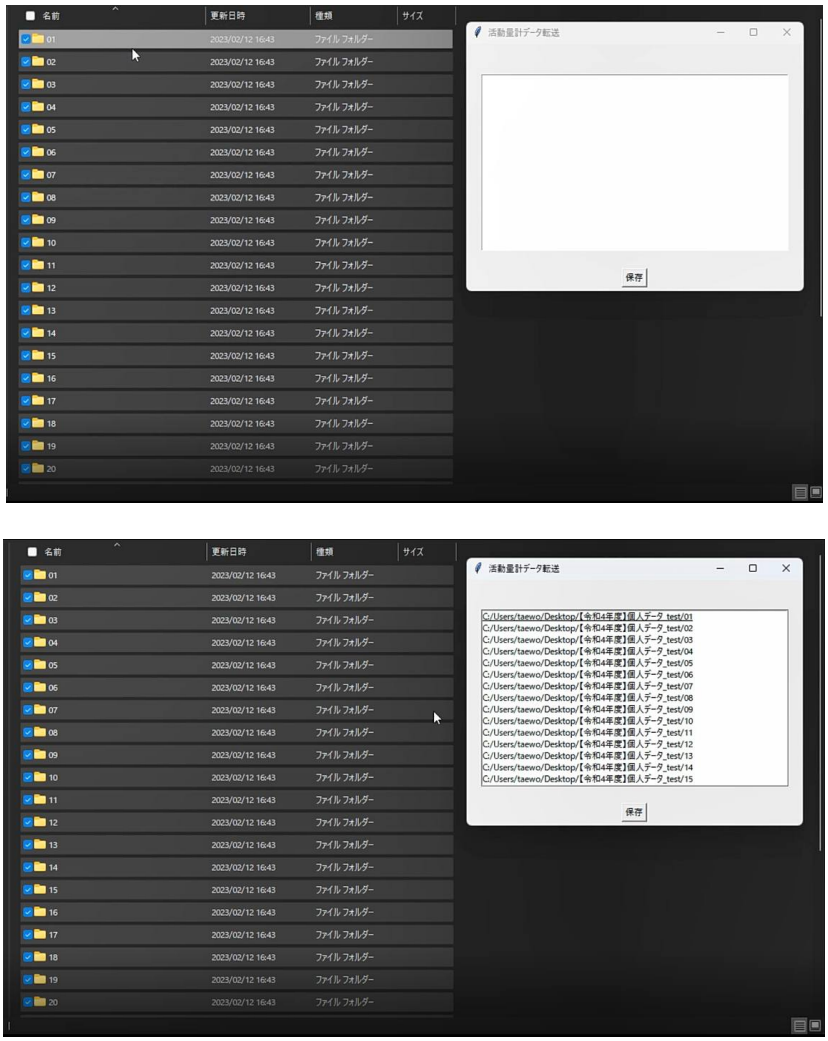

Fig.3.2: ドラッグ＆ドロップ機能

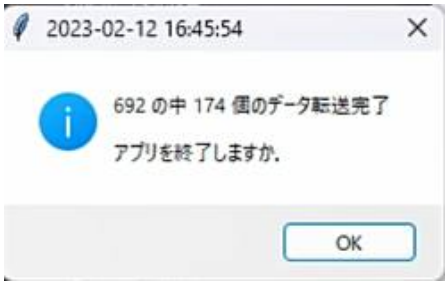

Fig.3.3: 更新ファイルの検知

3.2 読み聞かせの映像保存

吉田南幼稚園では一週間に 2 回，10 時から 11 時の間に約 10 分間読み聞かせを実施している．ランダムな時間帯に読み聞かせが実施されている為，cron と呼ばれる利用者の設定したスケジュールに従って指定されたプログラムを定期的に起動してくれるプログラムを用いて月曜日から金曜日までの 10 時から 11 時までのカメラ 4 台の映像を Fig.3.1に示したような小型パソコンに保存を行った．

各自のカメラの RTSP(Realtime Time Streaming Protocol) サーバという IP ネットワークを利用し，動画をリアルタイムで配信するサーバに接続し映像を保存する．

映像の保存には GStreamer と呼ばれるマルチメディアを扱うためのライブラリを用いた．4 つの RTSP サーバに同時に接続するためには最大 30 秒が必要である．正確に 10 時から 11 時の間の映像を保存する為に，9 時 59 分から 11 時 1 分までの 4 台のカメラ映像の保存する．保存が終わった後に ffmpeg と呼ばれる動画を記録・変換・再生するためのソフトウェアを用いて最初の 1 分と最後の 1 分を切ることで正確に 10 時から 11 時までの動画を保存することができた．映像サイズは 2592×1944[px] であり，フレーム率は 5[fps] である．

読み聞かせはランダムな時間帯に実施される為，保存した 1 時間の映像を基に読み聞かせの判定を行う必要がある．判定には Fig.2.6 に示したようなカメラ 4 を使用しフレーム単位で解析した．

一つ目の判定基準は Python の画像や動画の処理に使用される OpenCV と呼ばれるライブラリを用い Fig.3.4に示したように，AR マーカが 10 個以上検出された時を読み聞かせ時と判定した．しかし，読み聞かせ前に準備時間が存在するため読み聞かせの開始時間にずれが発生した．そこで，同じ先生方が壇上に座って読み聞かせを実施する情報を用いてもう一つの判定基準を設けた．

Fig.3.4: 一つ目の判定基準

二つ目の判定基準は Fig.3.5に示したように，壇上部分を抽出し物体認識で人が 1 名のみ検出された時，人の高さ h[px] を求める．また，事前に測定した人が座っている状態の高さは H[px] とする．測定した結果，$H = 390$[px] である．一人のみ壇上に座っている状態を表す式 (3.1) に示したような状態が 10 秒間 20 回以上検出された時を読み聞かせの開始時間と判定した．

$$H - 20 \leq h \leq H + 20 \tag{3.1}$$

Fig.3.5: 二つ目の判定基準

第 4 章

データの選定と表示

4.1　運動量データ

本研究では Fig.3.2 に示したように小型パソコンにインストールされている Apache2 という Web サーバを用いて Web 上にデータの表示を行った．Fig.4.1 に示したように，日付を選択し確認ボタンをクリックすることで全員の情報が確認できる．全員の情報は Fig.4.2に示したように出席番号順，運動強度が高い順，運動強度が低い順を選択することで整列することができる．

Fig.4.1: 日付選択

Fig.4.2: 整列基準の選択欄

全員の運動量データはFig.4.3に示した．3METs 以上の最初に計測された時から運動開始と判断し，出席番号，読み聞かせが始まる前までの総 METs，平均 METs，最大 METs，運動時間 [分]，消費カロリー [kcal]，運動強度，詳しい情報選択欄を表示した．運動強度は日本健康運動研究所の身体活動・運動の強度と量の指標を参考して Table.4.1に示したように，METs による運動強度を 7 段階に分けて評価した．

詳しい情報を選択することで Fig.4.4に示したように，運動開始時間から読み聞かせ終了時間までの METS 値を表すグラフを表示する．青い部分が運動時間で赤い部分が読み聞かせ時間である．

出席番号	総METs	平均METs	最大METs	運動時間 [分]	活動消費カロリー [kcal]	運動強度	詳しい情報
1	146.2	3.11	6.5	47	111.0	楽	◉
2	153.9	3.14	10.5	49	107.6	楽	○
3	107.3	3.46	8.5	31	102.2	楽	○
4	156.0	3.18	6.2	49	122.8	楽	○
5	68.4	2.07	4.8	33	82.4	非常に楽	○
6	72.0	2.06	3.5	35	84.0	非常に楽	○
7	0	0	0	0	0	x	
8	308.3	3.08	10.1	100	161.8	楽	○
9	233.0	3.33	8.9	70	149.0	楽	○
10	0	0	0	0	0	x	
11	67.8	2.12	5.0	32	78.0	非常に楽	○
12	107.6	2.83	7.4	38	83.2	非常に楽	○
13	105.2	3.29	9.2	32	110.2	楽	○
14	271.6	2.72	7.6	100	123.2	非常に楽	○
15	0	0	0	0	0	x	

出席番号	総METs	平均METs	最大METs	運動時間 [分]	活動消費カロリー [kcal]	運動強度	詳しい情報
16	109.9	3.33	6.4	33	87.8	楽	○
17	212.3	2.91	7.9	73	121.0	非常に楽	○
18	97.1	2.94	6.2	33	83.8	非常に楽	○
19	310.9	3.05	7.4	102	138.6	楽	○
20	0	0	0	0	0	x	
21	0	0	0	0	0	x	
22	292.4	2.92	8.3	100	141.8	非常に楽	○
23	0	0	0	0	0	x	
24	228.3	2.59	9.6	88	111.0	非常に楽	○
25	178.3	3.07	7.5	58	100.2	楽	○
26	119.7	3.52	8.2	34	89.0	楽	○
27	0	0	0	0	0	x	
28	197.5	2.82	8.4	70	104.2	非常に楽	○
29	139.5	2.97	7.2	47	91.2	非常に楽	○

Fig.4.3: 全員の運動量データ

Table.4.1: 運動強度の基準

METs	運動強度
2 以上 3 未満	非常に楽
3 以上 4 未満	楽
4 以上 5 未満	やや楽
5 以上 6 未満	ややきつめ
6 以上 7 未満	ややきつい
7 以上 8 未満	きつい
8 以上	非常にきつい

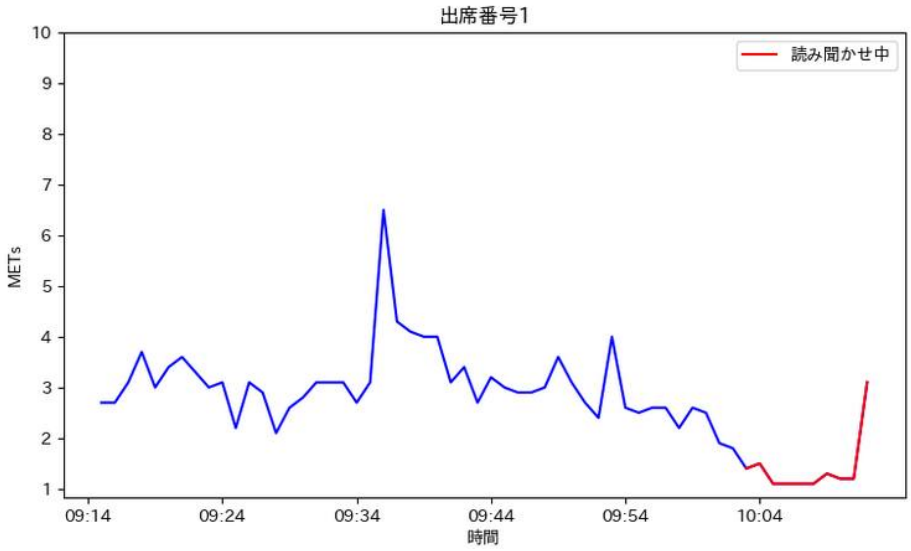

Fig.4.4: 個人の METs 情報

4.2 読み聞かせデータ

日付を選択すると，Fig.4.5に示したように，読み聞かせの時間と映像が表示される．Fig.4.6に示したように，全体のカメラ，カメラ 1，カメラ 2，カメラ 3，カメラ 4 を選ぶことができる．Fig.4.3に示したように，詳しい情報を選択した場合には選択した人の位置を Fig.4.7に示したように，矢印で表示する．また，Fig.4.8に示したように，横軸はフレーム，縦軸は x，y である選択した個人の頭部動揺が確認できる二つのグラフの動画を表示した．

Fig.4.5: 読み聞かせデータの表示

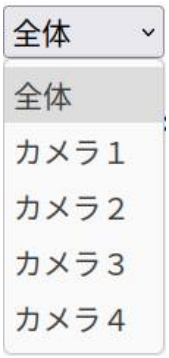

Fig.4.6: 読み聞かせカメラ選択欄

Fig.4.7: 個人の読み聞かせ動画

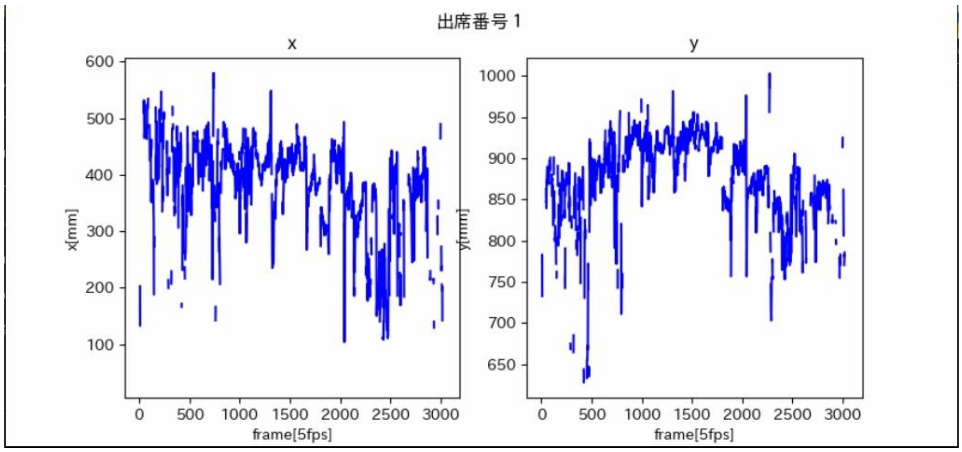

Fig.4.8: 個人の読み聞かせ時のグラフ

第 5 章

結論

本研究では幼稚園児の運動と集中力の関係を解析するためのシステム開発を行った．運動量と集中力の測定は幼稚園で実施される為，システムの自動化が必要である．

データの自動保存システムとしては，活動量計データを自動保存システムを開発した．システムは複数のフォルダをドラッグ＆ドロップ，ログファイルで更新ファイルの検出，scp 転送で原本データの保存の機能がある．活動量計の自動保存システムを用いることで，データベースに易しくデータを保存することとができる．読み聞かせの動画は 10 時から 11 時までのカメラ映像を保存し，保存した映像を基に読み聞かせ時間帯を判定する個とができた．

保存したデータを基に Fig.5.1に示したように，Web 上に運動と集中力の関係を解析するためにデータを選定と表示を行った．

正確な運動強度と集中力の判断基準がない問題点が見られた．基準の設定等の対策が必要であると考えられる．

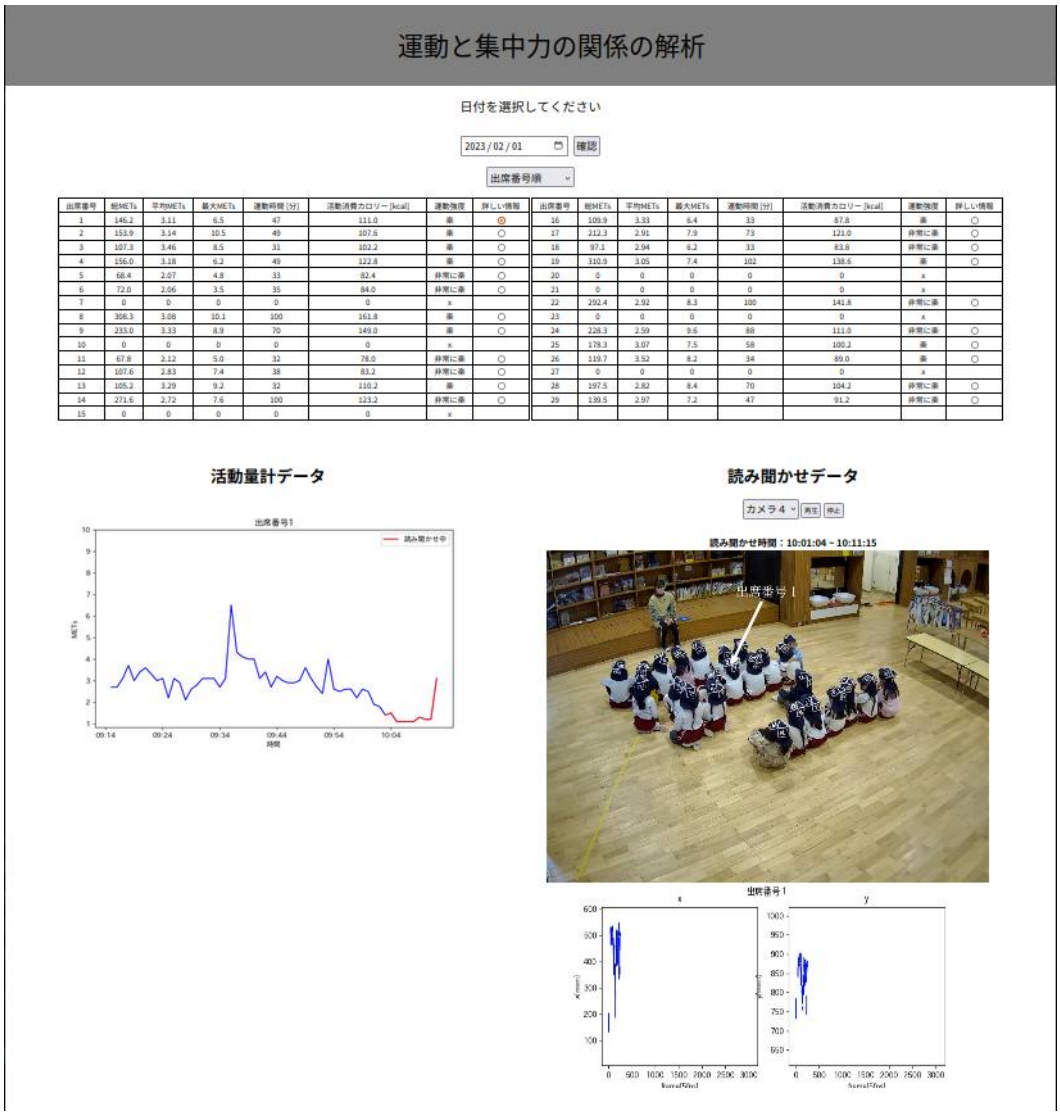

Fig.5.1: 全体の Web

謝辞

本研究を進め，論文を作成するにあたり，ご指導いただきました指導教員の熊澤 典良 准教授，上谷 俊平 教授に心からの感謝の意を表します．特に，担当教員である熊澤 典良 准教授には多くの助言をいただきました．深く感謝申し上げます．本研究の測定にご協力頂きました，吉田南幼稚園の橋口園長先生をはじめとする先生方や幼稚園児の皆様にも深く感謝申し上げます．また，本研究を進めるにあたりお世話になりました全ての方々に心よりの御礼を申し上げ，謝辞とさせていただきます．

Appendix A 章

付録 A 章の見出し

2023/02/23 ParkTaewon title : データの保存・表示 by k2519210977@kadai.jp

A.1　活動量計データの指導保存システムのプログラム

```
import os
import datetime as dt
import paramiko
import scp
from TkinterDnD2 import *
from tkinter import filedialog
from tkinter import messagebox
import csv
try:
    from Tkinter import *
    from ScrolledText import ScrolledText
except ImportError:
    from tkinter import *
    from tkinter.scrolledtext import ScrolledText

import mysql.connector
import pandas as pd

ip_address = '192.168.1.23'

root = TkinterDnD.Tk()
root.withdraw()
root.title('活動量計データ転送')
root.grid_rowconfigure(1, weight=1, minsize=250)
root.grid_columnconfigure(0, weight=1, minsize=250)
#root.grid_columnconfigure(1, weight=1, minsize=250)

def print_event_info(event):
    print('"nAction:', event.action)
    print('Supported actions:', event.actions)
    print('Mouse button:', event.button)
    print('Type codes:', event.codes)
    print('Current type code:', event.code)
    print('Common source types:', event.commonsourcetypes)
    print('Common target types:', event.commontargettypes)
    print('Data:', event.data)
    print('Event name:', event.name)
    print('Supported types:', event.types)
    print('Modifier keys:', event.modifiers)
    print('Supported source types:', event.supportedsourcetypes)
    print('Operation type:', event.type)
    print('Source types:', event.sourcetypes)
    print('Supported target types:', event.supportedtargettypes)
    print('Widget:', event.widget, '(type: %s)' % type(event.widget))
    print('X:', event.x_root)
    print('Y:', event.y_root, '"n')
```

```
Label(root).grid()

list_frame = Frame(root)
list_frame.grid()

listbox = Listbox(list_frame, selectmode="extended", width=70, height=15)
listbox.grid(row=1, column=0, columnspan=2, padx=20, pady=20, sticky='news')

text = Text()

def btnclick():
    list_file = []
    files = filedialog.askopenfilenames(initialdir="/",title = "select file",filetypes = (("*.csv","*csv"),('all files', '*.*')))
    for f in files:
        if os.path.exists(f):
            listbox.insert('end', f)
    if files == '':
        messagebox.showwarning("Warning", "ファイルを選択してください")

def read_log():
    result = []
    try:
        f = open('log.csv','r', encoding='utf-8')
        rdr = csv.reader(f)

        for line in rdr:
            result +=line
        f.close()
        return result
    except:
        return result

def write_log(list_name):
    f = open('log.csv','a', newline='', encoding="utf-8")
    wr = csv.writer(f)
    for i in range(0,len(list_name)):
        wr.writerow([list_name[i]])
    f.close()

def drop_enter(event):
    event.widget.focus_force()
    return event.action

def drop_position(event):
    return event.action

def drop_leave(event):
    return event.action

def drop(event):
    if event.data:
        if event.widget == listbox:
            files = listbox.tk.splitlist(event.data)
            global path_name
            path_name = sorted(listbox.tk.splitlist(event.data))
            path_name_folder = []
            for i in range(0,len(path_name)):
                path = path_name[i].split('/')
                file_path = path[-1]
                if file_path[-3:] == "csv":
                    break
                else:
                    file_list=os.listdir(path_name[i])
                    file_list_csv=[file for file in file_list if file.endswith(".csv")]
                    file_list_csv_all= [path_name[i]+ '/' + file_list_csv[j] for j in range(0,len(file_list_csv))]
                    path_name_folder += file_list_csv_all
            path_name = path_name_folder
            global all_file
            all_file = len(path_name)
            already_saved = read_log()
            if already_saved:
                path_name = list(set(path_name) - set(already_saved))
            for f in files:
                if os.path.exists(f):
                    listbox.insert('end', f)
        elif event.widget == text:
            bd = text['bd'] + text['highlightthickness']
            x = event.x_root - text.winfo_rootx() - bd
            y = event.y_root - text.winfo_rooty() - bd
            index = text.index('@%d,%d' % (x,y))
            text.insert(index, event.data)
    return event.action
```

```
listbox.drop_target_register(DND_FILES, DND_TEXT)
text.drop_target_register(DND_TEXT)

for widget in (listbox, text):
    widget.dnd_bind('<<DropEnter>>', drop_enter)
    widget.dnd_bind('<<DropPosition>>', drop_position)
    widget.dnd_bind('<<DropLeave>>', drop_leave)
    widget.dnd_bind('<<Drop>>', drop)

def drag_init_listbox(event):
    print_event_info(event)
    data = ()
    if listbox.curselection():
        data = tuple([listbox.get(i) for i in listbox.curselection()])
    return ((ASK, COPY), (DND_FILES, DND_TEXT), data)

def drag_init_text(event):
    print_event_info(event)
    data = ''
    sel = text.tag_nextrange(SEL, '1.0')
    if sel:
        data = text.get(*sel)
    return (COPY, DND_TEXT, data)

def drag_end(event):
    print('Drag ended for widget:', event.widget)

class DatabaseConnector:
    def __init__(self):
      self.conn = self.cur = None
      self.connect2db()


    def __del__(self):
      self.conn.close()


    def connect2db(self):
        self.conn = mysql.connector.connect(
            host=ip_address,
            port='3306',
            user='kindergarten',
            password='Chi-kuma3',
            database='kindergarten',
            auth_plugin="mysql_native_password"
        )

        self.cur = self.conn.cursor(buffered=True)

    def insert_test_data(self):
        for i in range(0,len(path_name)):
            path = path_name[i].split('/')
            file_path = path[-1]

            if file_path[11:17] == "Hourly" and file_path[11:16] != "Daily":
                file = pd.read_csv(path_name[i], encoding="shift-jis", skiprows = 6, "
                      names=['date','steps','active_steps','total_calories','active_calories','active_time','time','Nan'])

                date1 = file['date'].tolist()
                id1 = [file_path[0:10] for i in range(0,len(date1))]
                step1 = file['steps'].tolist()
                act_step1 = file['active_steps'].tolist()
                tot_cal1 = file['total_calories'].tolist()
                act_cal1 = file['active_calories'].tolist()
                act_time1 = file['active_time'].tolist()
                time1 = file['time'].tolist()
                data = []
                for dates, ids, steps, act_steps, tot_cals, act_cals, act_times, times in "
                zip(date1, id1, step1, act_step1, tot_cal1, act_cal1, act_time1, time1):
                    data.append((dates, ids, steps, act_steps, tot_cals, act_cals, act_times, times))
            sql = f"INSERT INTO `activity_tracker` (`date`,`id`,`steps`,`active_steps`,`total_calories`,`active_calories`,`active_time`,`time`) "
                      VALUES (%s, %s, %s, %s, %s, %s, %s, %s) ON DUPLICATE KEY UPDATE steps = VALUES(steps), ac-
tive_steps = VALUES(active_steps), "
                          total_calories  = VALUES(total_calories), active_calories = VALUES(active_calories) , active_time = VAL-
UES(active_time), time = VALUES(time);"

                file2 = pd.read_csv(path_name[i], encoding="shift-jis", nrows=4, names=['get','data','nan'])
                datas = file2['data'].tolist()
                height = int(datas[0])
                weight = int(float(datas[1]))
                age = int(datas[3])
                if datas[2] == "女性":
                    sex = 'f'
                else:
```

```
                sex = 'm'
            sql2 = f"INSERT INTO `student_information` (`activity_id`, `sex`, `age`, `height`, `weight`) VALUES (%s, %s, %s, %s, %s) "
                   "ON DUPLICATE KEY UPDATE age = VALUES(age), height = VALUES(height), weight = VALUES(weight);"
                self.cur.executemany(sql, data)
                self.cur.execute(sql2,[file_path[0:10], sex, age, height, weight])
                self.conn.commit()

            elif file_path[11:17] != "Hourly" and file_path[11:16] != "Daily" and file_path[0:2] != ".":
                file = pd.read_csv(path_name[i], encoding="shift-jis", skiprows=1, names=['date','METs'])
                date2 = file['date'].tolist()
                met2 = file['METs'].tolist()
                id2 = [file_path[0:10] for i in range(0,len(date2))]
                data = []
                for dates, ids, mets in zip(date2, id2, met2):
                    if mets != 0:
                        data.append((dates, ids, mets))
                  sql = f"INSERT INTO `activity_tracker` (`date`, `id`, `mets`) VALUES (%s, %s, %s) ON DUPLICATE KEY UP-
DATE mets = VALUES(mets);"
                self.cur.executemany(sql, data)
                self.conn.commit()


def scp_transmit_file():
    with paramiko.SSHClient() as ssh:
        ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())

        print('[ssh 接続]')
        ssh.connect(ip_address, port=22, username='ubuntu', password='chi-kuma')

        print('[SCP 転送開始]')
        for i in range(0,len(path_name)):
            with scp.SCPClient(ssh.get_transport()) as scpc:
                path = path_name[i].split('/')
                file_path = path[-1]
                file = fr"-path_name[i]"".replace('"u3000', '  ')
                scpc.put(file, f'/home/ubuntu/activity_data/-file_path[0:10]"')

def main():
    database_connector = DatabaseConnector()
    database_connector.insert_test_data()
    scp_transmit_file()
    write_log(path_name)
    now = dt.datetime.now()
    MsgBox = messagebox.showinfo(f'-now.strftime("%Y-%m-%d %H:%M:%S")"',f'-all_file" の中 -len(path_name)" "
             個のデータ転送完了"n"n アプリを終了しますか. ')
    if MsgBox == True:
        root.destroy()

buttonbox = Frame(root)
buttonbox.grid(row=2, column=0, columnspan=2, pady=5)
Button(buttonbox, text='保存', command=main).pack(side=LEFT, padx=5)

listbox.drag_source_register(1, DND_TEXT, DND_FILES)
text.drag_source_register(3, DND_TEXT)

listbox.dnd_bind('¡¡DragInitCmd¿¿', drag_init_listbox)
listbox.dnd_bind('¡¡DragEndCmd¿¿', drag_end)
text.dnd_bind('¡¡DragInitCmd¿¿', drag_init_text)
root.update_idletasks()
root.deiconify()
root.mainloop()
```

A.2　10 時から 11 時のカメラ映像保存するプログラム

```
#!/bin/sh

DATE=`date +"%Y%m%d%H%M%S"`

DAY=`date +"%Y%m%d%H"`

LOG_DATE=`date +"%Y%m%d"`

cam111="cam111"
cam112="cam112"
cam113="cam113"
cam114="cam114"
cam_all="4in1"

FNAME111=$DATE$cam111
FNAME112=$DATE$cam112
FNAME113=$DATE$cam113
```

```
FNAME114=$DATE$cam114
FNAME_ALL=$DAY$cam_all
FNAME_LOG=$LOGDATE

date_now=`date +"%Y"/"%m"/"%d"/"%H"`
date_now_log=`date +"%Y"/"%m"/"%d"`

path111="/hdd/kindergarten/$date_now/camera111"
path112="/hdd/kindergarten/$date_now/camera112"
path113="/hdd/kindergarten/$date_now/camera113"
path114="/hdd/kindergarten/$date_now/camera114"
path_storytelling="/hdd/kindergarten/$date_now/storytelling"

if [ ! -e $path111 ]; then
  mkdir -p $path111
fi

if [ ! -e $path112 ]; then
  mkdir -p $path112
fi

if [ ! -e $path113 ]; then
  mkdir -p $path113
fi

if [ ! -e $path114 ]; then
  mkdir -p $path114
fi

gst-launch-1.0 rtspsrc location=rtsp://admin:chi-kuma@172.16.0.111:554/live/main latency = 10000 ! "
 rtph264depay ! h264parse ! matroskamux ! filesink location=$path111/$FNAME111.mkv & "
gst-launch-1.0 rtspsrc location=rtsp://admin:chi-kuma@172.16.0.112:554/live/main latency = 10000 ! "
rtph264depay ! h264parse ! matroskamux ! filesink location=$path112/$FNAME112.mkv & "
gst-launch-1.0 rtspsrc location=rtsp://admin:chi-kuma@172.16.0.113:554/live/main latency = 10000 ! "
rtph264depay ! h264parse ! matroskamux ! filesink location=$path113/$FNAME113.mkv & "
gst-launch-1.0 rtspsrc location=rtsp://admin:chi-kuma@172.16.0.114:554/live/main latency = 10000 ! "
rtph264depay ! h264parse ! matroskamux ! filesink location=$path114/$FNAME114.mkv & "

sleep 1870
killall -2 gst-launch-1.0 && "

ffmpeg "
-i $path111/$FNAME111.mkv "
-ss 00:01:00 -t 1800 -vcodec copy -acodec copy -an -f mp4 $path111/$FNAME111.mp4 && "

ffmpeg "
-i $path112/$FNAME112.mkv "
-ss 00:01:00 -t 1800 -vcodec copy -acodec copy -an -f mp4 $path112/$FNAME112.mp4 && "

ffmpeg "
-i $path113/$FNAME113.mkv "
-ss 00:01:00 -t 1800 -vcodec copy -acodec copy -an -f mp4 $path113/$FNAME113.mp4 && "

ffmpeg "
-i $path114/$FNAME114.mkv "
-ss 00:01:00 -t 1800 -vcodec copy -acodec copy -an -f mp4 $path114/$FNAME114.mp4 && "

echo "start: "$DATE "END: "`date +"%Y%m%d_%H%M%S"` && "
echo ""

exit 0;
```

A.3 読み聞かせ時間帯を判断するプログラム

```
import time
import cv2
import csv
import numpy as np
import glob
from datetime import datetime, timedelta
import cvlib as cv
from cvlib.object_detection import draw_bbox
import sys
import os
import math
from PIL import Image
import diagonal_crop

# ------------------------------------------
#                 初期設定部分
# ------------------------------------------
```

```
year = sys.argv[1]
month = sys.argv[2]
date = sys.argv[3]

video_directory = f"/hdd/kindergarten/-year"/-month"/-date"/10"
intrinsic_parameters = "/home/ubuntu/camera/calibration_param/camera4/mtx.npy" #カメラ行列
distortion_coeffcients = "/home/ubuntu/camera/calibration_param/camera4/dist.npy" #歪みパラメータ

people_num = 29 # 園児の数
ARmarkers_per_person = 3 # 園児一人当たりにつける ARマーカーの数
ARmarker_num_per_pix = 50 # ピクセルごとの ARマーカー生成数
marker_length = 0.08 # ARマーカーのサイズ

col_names = ["date", "id", "cam_number" ,
    "c1_x" ,"frame"]

col_dic = -col_name : num for num ,col_name in enumerate(col_names)"

# ------------------------------------------------
#                   関数設定
# ------------------------------------------------
class MakePathArray:
    def __init__(self, directory_name, extension):
        self.file_paths = glob.glob(directory_name + '/**/*' + extension)
        self.file_paths.sort()

class ArmarkerDetecter:
    def __init__(self):
        self.dictionary_4pix = cv2.aruco.getPredefinedDictionary(cv2.aruco.DICT_4X4_50)
        self.dictionary_5pix = cv2.aruco.getPredefinedDictionary(cv2.aruco.DICT_5X5_50)

    def setparam(self, video_path, marker_length,
        intrinsic_parameters, distortion_coeffcients"
    ):
        self.marker_length = marker_length
        self.video_path = video_path
        self.cam_number = self._get_cam_number(video_path)
        self.intrinsic_parameters = np.load(intrinsic_parameters)
        self.distortion_coeffcients = np.load(distortion_coeffcients)
        self.book_data = []
        self.person_data = []

    def _get_cam_number(self, path):
        self.file_name = (path.split('/'))[-1]
        cam_number = self.file_name[-5]
        return cam_number

    def get_4corner_datas(self):
        video = cv2.VideoCapture(self.video_path)
        frame_count = 0
        datetime_string = year+month+date+"_100000"
        datetime_format = "%Y%m%d_%H%M%S"
        date_time = datetime.strptime(datetime_string, datetime_format)
        frame = round(1/video.get(cv2.CAP_PROP_FPS),1)
        angle = (math.pi * 13) / 180
        base = (0, 210)
        height = 500
        width = 900

        while True:
            people = 0
            info = []
            y_1=0
            y_2=0

            ret, img = video.read()
            if not ret: break

            fourcorner_param = self._detect_markers(img)
            data_frame = self._make_data_frame(frame_count,date_time)
            data = self._put_param(data_frame, fourcorner_param)
            check_start = [i[3] for i in data]

            if check_start.count(None) ¡ 77:
                im = Image.fromarray(img)
                cropped_im = diagonal_crop.crop(im, base, angle, height, width)
                numpy_image = np.array(cropped_im)
                bbox, label, conf = cv.detect_common_objects(numpy_image)

                for _ in range(0,len(label)):
                    info.append([bbox[_],label[_],conf[_]])

                for _ in range(0,len(label)):
                    if "person" in info[_]:
```

```
                    people += 1

                if people == 1:
                    y`1=info[0][0][1]
                    y`2=info[0][0][3]
                    height` = y`2 - y`1
                if height` ¿= 370 and height` ¡= 510:
                    self.person`data.append(date`time)
            date`time += timedelta(seconds=frame)
            frame`count += 1
        video.release()

        return self.person`data

    def `detect`markers(self, img):
        corners`4pix, ids`4pix, rejectedImgPoints`4pix = cv2.aruco.detectMarkers(img, self.dictionary`4pix)
        corners`5pix, ids`5pix, rejectedImgPoints`5pix = cv2.aruco.detectMarkers(img, self.dictionary`5pix)

        return  –
            'corners' : [corners`4pix, corners`5pix],
            'ids' : [ids`4pix, ids`5pix],
            'rejectedImgPoints' : [rejectedImgPoints`4pix,
                rejectedImgPoints`5pix]
            ″

    def `make`data`frame(self, frame`count ,date`time):
        empty`data = [
            [date`time, id, self.cam`number,
            None, frame`count]
            for id in range(people`num * ARmarkers`per`person)]
        return empty`data

    def `put`param(self, data`frame, fourcorner`param):
        for pixel`num, (ids, corners) in enumerate(zip(fourcorner`param['ids'], fourcorner`param['corners'])):
            if corners:
                ids = np.squeeze(ids)
                if np.ndim(ids) == 0:
                    ids = np.array([ids])

                corners = np.squeeze(corners)
                if np.ndim(corners) == 2:
                    corners = np.array([corners])
                for id, corner in zip(ids, corners):
                    if pixel`num == 1:
                        id += ARmarker`num`per`pix
                    if id ¡ people`num * ARmarkers`per`person:
                        for values in enumerate(corner):
                            data`frame[id][col`dic["c1`x"]] = values[0]

        return data`frame

def main():
    start`time = 0
    start`time`title = 0
    end`time = 0
    end`time`title = 0
    index = 0
    index`start = 0

    video`paths = MakePathArray(video`directory, 'mp4')
    armarker`detecter = ArmarkerDetecter()

    armarker`detecter.setparam(video`paths.file`paths[3], marker`length,intrinsic`parameters, distortion`coeffcients)
    data`person = armarker`detecter.get`4corner`datas()

    if len(data`person) != 0:
        successive`time = []
        for i in range(1,len(data`person)-50):
            index = 0
            for ` in range(1,50):
             if float(data`person[i + `].time().strftime('%M%S.%f')) - float(data`person[i + ` - 1].time().strftime('%M%S.%f')) ¡= 1.0:
                    index +=1
            if index == 49:
                index`start= i-1
                break

        start`time = data`person[index`start].time().strftime('00:%M:%S.%f')
        start`time`title = data`person[index`start].time().strftime('%H%M%S.%f')
        end`time = (data`person[-1] + timedelta(seconds=2)).time().strftime('00:%M:%S.%f')
        end`time`title = (data`person[-1] + timedelta(seconds=2)).time().strftime('%H%M%S.%f')
        print(start`time,end`time)

        os.system(
            f" “
```

```
            ffmpeg -i -video'paths.file'paths[0]" -ss -start'time" -to -end'time" "
            -vcodec copy -acodec copy -video'directory"/storytelling/-year"-month" "
            -date"'-start'time'title"'-end'time'title"'cam11-video'paths.file'paths[0][-5]".mp4 && "
            ffmpeg -i -video'paths.file'paths[1]" -ss -start'time" -to -end'time" "
            -vcodec copy -acodec copy -video'directory"/storytelling/-year"-month" "
            -date"'-start'time'title"'-end'time'title"'cam11-video'paths.file'paths[1][-5]".mp4 && "
            ffmpeg -i -video'paths.file'paths[2]" -ss -start'time" -to -end'time" "
            -vcodec copy -acodec copy -video'directory"/storytelling/-year"-month" "
            -date"'-start'time'title"'-end'time'title"'cam11-video'paths.file'paths[2][-5]".mp4 && "
            ffmpeg -i -video'paths.file'paths[3]" -ss -start'time" -to -end'time" "
            -vcodec copy -acodec copy -video'directory"/storytelling/-year"-month" "
            -date"'-start'time'title"'-end'time'title"'cam11-video'paths.file'paths[3][-5]".mp4")

        os.system(
            f' "
            ffmpeg  "
            -i -video'directory"/storytelling/-year"-month"-date"'-start'time'title"' "
            -end'time'title"'cam11-video'paths.file'paths[0][-5]".mp4 "
            -i -video'directory"/storytelling/-year"-month"-date"'-start'time'title"' "
            -end'time'title"'cam11-video'paths.file'paths[1][-5]".mp4 "
            -i -video'directory"/storytelling/-year"-month"-date"'-start'time'title"' "
            -end'time'title"'cam11-video'paths.file'paths[2][-5]".mp4 "
            -i -video'directory"/storytelling/-year"-month"-date"'-start'time'title"' "
            -end'time'title"'cam11-video'paths.file'paths[3][-5]".mp4 "
            -filter'complex " "
            [0:v]scale=648*486[v0]; "
            [1:v]scale=648*486[v1]; "
            [2:v]scale=648*486[v2]; "
            [3:v]scale=648*486[v3]; "
            [v0][v1][v2][v3]xstack=inputs=4:layout=0'0|w0'0|0'h0|w0'h0[v]" "
            -map "[v]" -c:v libx264 -preset veryfast -acodec copy -an -f mp4 "
            -video'directory"/-year"-month"-date"'-start'time'title"'-end'time'title"'4in1.mp4')

# ------------------------------------------------------------
#                           実行
# ------------------------------------------------------------
if ''name'' == ''main'':
    main()
```

A.4 データを Web 上に表示する最初ページのプログラム

```
#!/usr/bin/python3
import mysql.connector
import matplotlib.pyplot as plt
import sys

# ------------------------------------------------------------
#                         初期設定部分
# ------------------------------------------------------------
print("Content-Type: text/html")
print()

def main():
  html1 = """
  <!DOCTYPE html>
  <html>
    <head>
      <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
      <link rel="stylesheet" href="./css/style.css?after">
      <script src= "./js/myfunction.js?after"></script>
      <title>test</title>
    </head>

    <body>
      <header>
        <a href="./index2.py">
          <p class="title">運動と集中力の関係の解析</p>
        </a>
      </header>
      <div id="date">
        <label for="date">日付を選択してください</label>
      </div>
      <div id="input">
        <form action="./cgi-bin/create2.py" method="POST">
          <input type="date" class="input'date" name="date" style="font-size:20px;">
          <input type="submit" class="input'submit" onclick="input()" value="確認" style="font-size:20px;">
```

```
        ¡/form¿
      ¡/div¿
    ¡/body¿
  ¡/html¿
  """
  print(html1)

# ————————————————————————————————
#                実行
# ————————————————————————————————
if ``name`` == ```main```:
    main()
```

A.5 データを Web 上に表示するページのプログラム

```
#!/usr/bin/python3

import cgi
import cgitb
import mysql.connector
import matplotlib.pyplot as plt
from matplotlib import dates
import pandas as pd
import numpy as np
import datetime as dt
import japanize`matplotlib
import os
import glob

# ————————————————————————————————
#                初期設定部分
# ————————————————————————————————
cgitb.enable()
global date
global sort
global student`num
sort = "0"
cam`num = "all"
student`num = "0"
print("Content-Type: text/html")
print()

form = cgi.FieldStorage()
for data in form:
  if data == 'date':
    date = form[data].value
  elif data == 'sort':
    sort = form[data].value
  elif data == 'cam':
    cam`num = form[data].value
  elif data == 'choice':
    student`num = form[data].value

# ————————————————————————————————
#                関数設定
# ————————————————————————————————
class DatabaseConnector:
  def ``init``(self):
    self.conn = self.cur = None
    self.connect2db()

  def ``del``(self):
    self.cur.close()
    self.conn.close()

  def connect2db(self):
    self.conn = mysql.connector.connect(
      host='localhost',
      port='3306',
      user='kindergarten',
      password='Chi-kuma3',
      database='kindergarten',
      auth`plugin="mysql`native`password"
    )

    self.cur = self.conn.cursor(buffered=True)

  def select`student`information`data(self):
    sql = f"SELECT * FROM `student`information` ORDER BY number;"
    self.cur.execute(sql)
    self.conn.commit()
```

```
    informations = self.cur.fetchall()
    inforamtion = []

    for x in informations:
      inforamtion.append(x)

    return inforamtion

  def select`activity`tracker`data(self, activity`id, slect`date):
    sql = f"SELECT * FROM `activity`tracker` WHERE id=–activity`id" "
          AND `date` BETWEEN '–slect`date" 07:00:00.00' AND '–slect`date" 11:00:00.00';"
    self.cur.execute(sql)
    self.conn.commit()

    informations = self.cur.fetchall()
    inforamtion = []

    for x in informations:
      inforamtion.append(x)

    return inforamtion

class createGraph:
  def createMinGraph(self,info,start,end,stud`num):
    exercise`start=0
    exercise`start`index=0
    exercise`time = 0
    if start == 0 and end == 0:
      date = pd.date`range("08:00","11:00",freq="1min").time
      del info[0:60]
      mets = list(zip(*info))[2]

      fig = plt.figure(figsize=(9,6))
      pd.plotting.register`matplotlib`converters()
      ax = fig.add`subplot(1,1,1)
      ax.plot(date, mets, color='blue')

      ax.set`xticks(pd.date`range("08:00","11:00",freq="5min").time)
      ax.set`yticks([`+1 for ` in range(10) ])

      plt.xticks(rotation=90)
      plt.xlabel("時間")
      plt.ylabel("METs")
      plt.title(f'出席番号–stud`num"')
      plt.savefig('./graph1.png', format='png')

    else:
      date = pd.date`range("08:00","11:00",freq="1min").time
      start`min = ((int(start[0:2]) - 8) * 3600 + int(start[2:4]) * 60 + int(start[4:6])) // 60
      end`min = ((int(end[0:2]) - 8) * 3600 + int(end[2:4]) * 60 + int(end[4:6])) // 60
      del info[0:60]
      mets = list(zip(*info))[2]
      for ` in range(0,len(mets)):
        if float(mets[`]) ¿= 3.0:
          exercise`start = date[`].strftime('%H:%M:%S')
          exercise`start`index = `
          date = np.delete(date,slice(0,`+1))
          mets = np.delete(mets,slice(0,`+1))
          break
      exercise`time =  dt.datetime.strptime(start[0:4],"%H%M") - dt.datetime.strptime(exercise`start[0:5],"%H:%M")

      fig = plt.figure(figsize=(9,5))
      pd.plotting.register`matplotlib`converters()
      ax = fig.add`subplot(1,1,1)
      ax.plot(date[:end`min-exercise`start`index+1], mets[:end`min-exercise`start`index+1], color='blue')
      ax.plot(date[start`min-exercise`start`index+1:end`min-exercise`start`index+1], "
            mets[start`min-exercise`start`index+1:end`min-exercise`start`index+1], color='red',label='読み聞かせ中')
      ax.legend()
      ax.set`xticks(pd.date`range(f"–exercise`start[0:5]"",f"–end[0:2]":–end[2:4]"",freq="10min").time)
      ax.set`yticks([`+1 for ` in range(10) ])

      # plt.xticks(rotation=90)
      plt.xlabel("時間")
      plt.ylabel("METs")
      plt.title(f'出席番号–stud`num"')
      plt.savefig('./graph1.png', format='png')


class createHtmlCode:
  def createSelectCode(self, info):
    html`code = ""
    if info == "0":
      html`code = f""
        ¡select name="sort" onchange="this.form.submit();" style="font-size:20px;"¿
```

```
        ¡option value="0" selected¿出席番号順¡/option¿
        ¡option value="1"¿運動強度高い順¡/option¿
        ¡option value="2"¿運動強度低い順¡/option¿
      ¡/select¿
    """
  elif info == "1":
    html_code = f"""
      ¡select name="sort" onchange="this.form.submit();" style="font-size:20px;"¿
        ¡option value="0"¿出席番号順¡/option¿
        ¡option value="1" selected¿運動強度高い順¡/option¿
        ¡option value="2"¿運動強度低い順¡/option¿
      ¡/select¿
    """
  elif info == "2":
    html_code = f"""
      ¡select name="sort" onchange="this.form.submit();" style="font-size:20px;"¿
        ¡option value="0"¿出席番号順¡/option¿
        ¡option value="1"¿運動強度高い順¡/option¿
        ¡option value="2" selected¿運動強度低い順¡/option¿
      ¡/select¿
    """
  return html_code

def createSelectMovieCode(self, info):
  html_code = ""
  if info == "all":
    html_code = f"""
      ¡select id="cam" name="cam" onchange="this.form.submit();" style="font-size:20px;"¿
        ¡option value="all" selected="selected"¿全体¡/option¿
        ¡option value="cam1"¿カメラ 1¡/option¿
        ¡option value="cam2"¿カメラ 2¡/option¿
        ¡option value="cam3"¿カメラ 3¡/option¿
        ¡option value="cam4"¿カメラ 4¡/option¿
      ¡/select¿
    """
  elif info == "cam1":
    html_code = f"""
      ¡select id="cam" name="cam" onchange="this.form.submit();" style="font-size:20px;"¿
        ¡option value="all"¿全体¡/option¿
        ¡option value="cam1" selected="selected"¿カメラ 1¡/option¿
        ¡option value="cam2"¿カメラ 2¡/option¿
        ¡option value="cam3"¿カメラ 3¡/option¿
        ¡option value="cam4"¿カメラ 4¡/option¿
      ¡/select¿
    """
  elif info == "cam2":
    html_code = f"""
      ¡select id="cam" name="cam" onchange="this.form.submit();" style="font-size:20px;"¿
        ¡option value="all"¿全体¡/option¿
        ¡option value="cam1"¿カメラ 1¡/option¿
        ¡option value="cam2" selected="selected"¿カメラ 2¡/option¿
        ¡option value="cam3"¿カメラ 3¡/option¿
        ¡option value="cam4"¿カメラ 4¡/option¿
      ¡/select¿
    """
  elif info == "cam3":
    html_code = f"""
      ¡select id="cam" name="cam" onchange="this.form.submit();" style="font-size:20px;"¿
        ¡option value="all"¿全体¡/option¿
        ¡option value="cam1"¿カメラ 1¡/option¿
        ¡option value="cam2"¿カメラ 2¡/option¿
        ¡option value="cam3" selected="selected"¿カメラ 3¡/option¿
        ¡option value="cam4"¿カメラ 4¡/option¿
      ¡/select¿
    """
  elif info == "cam4":
    html_code = f"""
      ¡select id="cam" name="cam" onchange="this.form.submit();" style="font-size:20px;"¿
        ¡option value="all"¿全体¡/option¿
        ¡option value="cam1"¿カメラ 1¡/option¿
        ¡option value="cam2"¿カメラ 2¡/option¿
        ¡option value="cam3"¿カメラ 3¡/option¿
        ¡option value="cam4" selected="selected"¿カメラ 4¡/option¿
      ¡/select¿
    """
  return html_code

def createResultTableCode(self, student_info, activity_info, select, radio, start):
  #date, id, mets, step, activestep, totalcalorie, calories, activetime, time#
  html_code = f"""
    ¡div id= 'tb' style="margin: 1% 5% 3% 5%;" ¿
      ¡table border="2" align = "center" style="font-size:15px; text-align:center;
       border-collapse: collapse; border:solid;border-width:3px;margin:0;width:100%;"¿
        ¡tr¿
```

```
          ¡td¿出席番号¡/td¿
          ¡td¿総 METs¡/td¿
          ¡td¿平均 METs¡/td¿
          ¡td¿最大 METs¡/td¿
          ¡td¿運動時間 [分]¡/td¿
          ¡td¿活動消費カロリー [kcal]¡/td¿
          ¡td¿運動強度¡/td¿
          ¡td style="border-right:double;border-right-width:9px;"¿詳しい情報¡/td¿
          ¡td¿出席番号¡/td¿
          ¡td¿総 METs¡/td¿
          ¡td¿平均 METs¡/td¿
          ¡td¿最大 METs¡/td¿
          ¡td¿運動時間 [分]¡/td¿
          ¡td¿活動消費カロリー [kcal]¡/td¿
          ¡td¿運動強度¡/td¿
          ¡td¿詳しい情報¡/td¿
        ¡/tr¿
        """
    all_data = []
    mets = [' for ' in range(0,29)]
    max_mets = [' for ' in range(0,29)]
    index = 0
    end_index = int(start.split(".")[0][0:2])*60 - 420 + int(start.split(".")[0][2:4])

    for i in range(0,29):
      weight =  float(student_info[i][9])
      step = int(activity_info[i][60][3]) + int(activity_info[i][120][3])
      activestep = int(activity_info[i][60][4]) + int(activity_info[i][120][4])
      totalcalories = float(activity_info[i][60][5]) + float(activity_info[i][120][5])
      calories = float(activity_info[i][60][6]) + float(activity_info[i][120][6])
      activetime = int(activity_info[i][60][7]) + int(activity_info[i][120][7])
      total_time = int(activity_info[i][60][8]) + int(activity_info[i][120][8])


      for ' in range(0,end_index):
        if float(activity_info[i][''][2]) ¿= 3.0:
          index= '
          break
      mets[i] = [r[2] for r in activity_info[i][index:end_index]]
      average_mets = round(sum(mets[i])/len(mets[i]),2)
      all_mets = round(sum(mets[i]),2)
      max_mets[i] = max(mets[i])

      time = (float(activity_info[i][60][8]) + float(activity_info[i][120][8])) / 3600
      hour = total_time // 3600
      total_time = total_time - hour * 3600
      min = total_time // 60
      sec = total_time - min * 60
      # average_mets = round(calories / (eeight * time * 1.05),1)
      if activetime == 0:
        all_data.append([i+1,0,0,0,0,0,"x",""])
      else:
        if average_mets ¿= 2 and average_mets ¡ 3:
          if i+1 == int(radio):
            all_data.append([i+1,all_mets,average_mets,max_mets[i],len(mets[i]), round(totalcalories,1),"非常に楽" "
                                    ,f"¡input type='radio' name='choice' value='-i+1"' checked onclick='this.form.submit();' on-
change = 'outputVideo();'¿"])
          else:
            all_data.append([i+1,all_mets,average_mets,max_mets[i],len(mets[i]),round(totalcalories,1),"非常に楽" "
                  ,f"¡input type='radio' name='choice' value='-i+1"' onclick='this.form.submit();' onchange = 'outputVideo();'¿"])

        elif average_mets ¿= 3 and average_mets ¡ 4:
          if i+1 == int(radio):
            all_data.append([i+1,all_mets,average_mets,max_mets[i],len(mets[i]),round(totalcalories,1),"楽" "
                                    ,f"¡input type='radio' name='choice' value='-i+1"' checked onclick='this.form.submit();' on-
change = 'outputVideo();'¿"])
          else:
            all_data.append([i+1,all_mets,average_mets,max_mets[i],len(mets[i]),round(totalcalories,1),"楽" "
                  ,f"¡input type='radio' name='choice' value='-i+1"' onclick='this.form.submit();' onchange = 'outputVideo();'¿"])

        elif average_mets ¿= 4 and average_mets ¡ 5:
          if i+1 == int(radio):
            all_data.append([i+1,all_mets,average_mets,max_mets[i],len(mets[i]),round(totalcalories,1),"やや楽" "
                                    ,f"¡input type='radio' name='choice' value='-i+1"' checked onclick='this.form.submit();' on-
change = 'outputVideo();'¿"])
          else:
            all_data.append([i+1,all_mets,average_mets,max_mets[i],len(mets[i]),round(totalcalories,1),"やや楽"
                  ,f"¡input type='radio' name='choice' value='-i+1"' onclick='this.form.submit();' onchange = 'outputVideo();'¿"])

        elif average_mets ¿= 5 and average_mets ¡ 6:
          if i+1 == int(radio):
            all_data.append([i+1,all_mets,average_mets,max_mets[i],len(mets[i]),round(totalcalories,1),"ややきつめ" "
                                    ,f"¡input type='radio' name='choice' value='-i+1"' checked onclick='this.form.submit();' on-
change = 'outputVideo();'¿"])
```

```
        else:
          all_data.append([i+1,all_mets,average_mets,max_mets[i],len(mets[i]),round(totalcalories,1),"
                                ,"ややきつめ",f"¡input type='radio' name='choice' value='-i+1"' onclick='this.form.submit();' on-
change = 'outputVideo();'¿"])

      elif average_mets ¿= 6 and average_mets ¡ 7:
        if i+1 == int(radio):
          all_data.append([i+1,all_mets,average_mets,max_mets[i],len(mets[i]),round(totalcalories,1),"ややきつい" "
                                ,f"¡input type='radio' name='choice' value='-i+1"' checked onclick='this.form.submit();' on-
change = 'outputVideo();'¿"])
        else:
          all_data.append([i+1,all_mets,average_mets,max_mets[i],len(mets[i]),round(totalcalories,1),"ややきつい" "
                ,f"¡input type='radio' name='choice' value='-i+1"' onclick='this.form.submit();' onchange = 'outputVideo();'¿"])

      elif average_mets ¿= 7 and average_mets ¡ 8:
        if i+1 == int(radio):
          all_data.append([i+1,all_mets,average_mets,max_mets[i],len(mets[i]),round(totalcalories,1),"きつい" "
                                ,f"¡input type='radio' name='choice' value='-i+1"' checked onclick='this.form.submit();' on-
change = 'outputVideo();'¿"])
        else:
          all_data.append([i+1,all_mets,average_mets,max_mets[i],len(mets[i]),round(totalcalories,1),"きつい" "
                ,f"¡input type='radio' name='choice' value='-i+1"' onclick='this.form.submit();' onchange = 'outputVideo();'¿"])

      elif average_mets ¿= 8:
        if i+1 == int(radio):
          all_data.append([i+1,all_mets,average_mets,max_mets[i],len(mets[i]),round(totalcalories,1),"非常にきつい" "
                                ,f"¡input type='radio' name='choice' value='-i+1"' checked onclick='this.form.submit();' on-
change = 'outputVideo();'¿"])
        else:
          all_data.append([i+1,all_mets,average_mets,max_mets[i],len(mets[i]),round(totalcalories,1),"非常にきつい" "
                ,f"¡input type='radio' name='choice' value='-i+1"' onclick='this.form.submit();' onchange = 'outputVideo();'¿"])

  if select == "0":
    all_data.sort(key=lambda x:x[0])
  elif select == "1":
    all_data.sort(key=lambda x:-x[1])
  elif select == "2":
    all_data.sort(key=lambda x:x[1])

  for i in range(0,len(all_data)-15):
    html_code += f"""
      ¡tr¿
        ¡td¿-all_data[i][0]"¡/td¿
        ¡td¿-all_data[i][1]"¡/td¿
        ¡td¿-all_data[i][2]"¡/td¿
        ¡td¿-all_data[i][3]"¡/td¿
        ¡td¿-all_data[i][4]"¡/td¿
        ¡td¿-all_data[i][5]"¡/td¿
        ¡td¿-all_data[i][6]"¡/td¿
        ¡td style="border-right:double;border-right-width:9px;"¿-all_data[i][7]"¡/td¿
        ¡td¿-all_data[i+15][0]"¡/td¿
        ¡td¿-all_data[i+15][1]"¡/td¿
        ¡td¿-all_data[i+15][2]"¡/td¿
        ¡td¿-all_data[i+15][3]"¡/td¿
        ¡td¿-all_data[i+15][4]"¡/td¿
        ¡td¿-all_data[i+15][5]"¡/td¿
        ¡td¿-all_data[i+15][6]"¡/td¿
        ¡td¿-all_data[i+15][7]"¡/td¿
      ¡/tr¿
    """
  html_code += f"""
    ¡tr¿
        ¡td¿-all_data[14][0]"¡/td¿
        ¡td¿-all_data[14][1]"¡/td¿
        ¡td¿-all_data[14][2]"¡/td¿
        ¡td¿-all_data[14][3]"¡/td¿
        ¡td¿-all_data[14][4]"¡/td¿
        ¡td¿-all_data[14][5]"¡/td¿
        ¡td¿-all_data[14][6]"¡/td¿
        ¡td style="border-right:double;border-right-width:9px;"¿-all_data[14][7]"¡/td¿
        ¡td¿¡/td¿
        ¡td¿¡/td¿
        ¡td¿¡/td¿
        ¡td¿¡/td¿
        ¡td¿¡/td¿
        ¡td¿¡/td¿
        ¡td¿¡/td¿
        ¡td¿¡/td¿
    ¡/tr¿
  """
  html_code += "¡/table¿¡/div¿"

  return html_code
```

```
  def createVideoCode(self,date,select):
    year = date[0:4]
    month = date[5:7]
    day = date[8:10]
    path = f"../video/kindergarten/{year}/{month}/{day}/10/"
    path_each = f"../video/kindergarten/{year}/{month}/{day}/10/storytelling/"
    files_path = glob.glob(f'{path}*.mp4')
    files_each_path = glob.glob(f'{path_each}*.mp4')
    files_each_path = sorted(files_each_path)

    html_code = ''

    if None not in files_path and len(files_path) != 0:
      file_name = files_path[0].split('/')[-1]
      start_time = file_name.split('_')[1]
      end_time = file_name.split('_')[2]
      html_code = f"""
        <h3 style = "margin:3% 3% 0% 3%;">読み聞かせ時間：
        {start_time[0:2]}:{start_time[2:4]}:{start_time[4:6]} ~ {end_time[0:2]}:{end_time[2:4]}:{end_time[4:6]}</h3>
        """
      if select == "all":
        html_code += f"""
          <video style="height:648px;" id="video" controls muted><source src="{files_path[0]}" type="video/mp4"></video>
        """
      elif select == "cam1":
        html_code += f"""
          <video style="height:648px;" id="video" controls muted><source src="{files_each_path[0]}" type="video/mp4"></video>
        """
      elif select == "cam2":
        html_code += f"""
          <video style="height:648px;" id="video" controls muted><source src="{files_each_path[1]}" type="video/mp4"></video>
        """
      elif select == "cam3":
        html_code += f"""
          <video style="height:648px;" id="video" controls muted><source src="{files_each_path[2]}" type="video/mp4"></video>
        """
      elif select == "cam4":
        html_code += f"""
          <video style="height:648px;" id="video" controls muted><source src="{files_each_path[3]}" type="video/mp4"></video>
        """
    else:
      html_code = "<h3>動画データが存在しません</h3>"
      start_time = 0
      end_time = 0

    return html_code, start_time, end_time

def main():
  database_connector = DatabaseConnector()
  create_html_code = createHtmlCode()
  create_graph = createGraph()
  activity_tracker_information = []
  student_information = database_connector.select_student_information_data()
  student_id =list(zip(*student_information))[2]

  for index in student_id:
    activity_tracker_information.append(database_connector.select_activity_tracker_data(index,date))

  select_code = create_html_code.createSelectCode(sort)
  select_movie_code = create_html_code.createSelectMovieCode(cam_num)
  video_code, start_time, end_time = create_html_code.createVideoCode(date,cam_num)

 html_table_code = create_html_code.createResultTableCode(student_information, activity_tracker_information, sort, student_num,start_time)
 if student_num != "0":
   create_graph.createMinGraph(activity_tracker_information[int(student_num)-1], start_time, end_time,student_num)
   img_html = '<img id="active" src="image1.py" style="max-width:95%">'
 else:
   img_html = ''

 html1 = """
 <!DOCTYPE html>
 <html>
   <head>
     <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
     <link rel="stylesheet" href="../css/style.css?afsddd1231">
     <script src= "../js/myfunction.js"></script>
     <script src="../js/canvas-arrow.js"></script>
     <title>test</title>
   </head>

   <body>
     <header>
       <a href="../index2.py">
         <p class="title">運動と集中力の関係の解析</p>
```

```
      ¡/a¿
    ¡/header¿
    ¡div id="date"¿
      ¡label for="date"¿日付を選択してください¡/label¿
    ¡/div¿
    ¡form action="./create2.py" method="POST"¿
    ¡div id="input"¿
        ¡input type="date" class="input'date" name="date" value=%s style="font-size:20px;"¿
        ¡input type="submit" class="input'submit" onclick="input()" value="確認" style="font-size:20px;"¿
    ¡/div¿
    ¡div id="content"¿
      ¡div class = "all"¿%s %s¡/div¿
      ¡div class = "left"¿
¡h1¿活動量計データ¡/h1¿
        ¡div class = "up"¿¡/div¿
        ¡div class = "down"¿%s¡/div¿
      ¡/div¿
      ¡div class = "right"¿
       ¡div class = "title"¿
        ¡h1¿読み聞かせデータ¡/h1¿
          %s
          ¡button type="button" onClick="outputVideo()" id="btnResume"¿再生¡/button¿
          ¡button type="button" id="btnPause" onClick="Pause()"¿停止¡/button¿

       ¡/div¿
          %s
          ¡div id="cv"¿
            ¡canvas id="parent" width="864px" height="648px"¿¡/canvas¿
            ¡canvas id="children" width="864px" height="648px"¿¡/canvas¿
            ¡canvas id="children2" width="864px" height="648px"¿¡/canvas¿
            ¡canvas id="children3" width="864px" height="1048px"¿¡/canvas¿
          ¡/div¿
    ¡/div¿

  ¡/form¿
  ¡/body¿
¡/html¿
"""  % (date, select'code, html'table'code,img'html,select'movie'code,video'code)

print(html1)

# ————————————————————————————————————————–
#                                実行
# ————————————————————————————————————————–
if ''name'' == '''main''':
    main()
```

A.6 個人の MET s グラブを Web 上に表示するプログラム

```
#!/usr/bin/python3

import sys
import os

src =  "//var/www/html/cgi-bin/graph1.png"
length = os.stat(src).st'size

sys.stdout.write("Content-Type: image/png“n")
sys.stdout.write("Content-Length: " + str(length) + "“n")
sys.stdout.write("“n")
sys.stdout.flush()
sys.stdout.buffer.write(open(src, "rb").read())
```

A.7 Web のデザインを設定するプログラム

```
header–
    width: 100%;
    background-color: gray;
"

body –
    margin: 0%;
"
```

```
header a-
    text-decoration: none;
"

header a:visited, a:hover-
    color: black;
"

header-
    padding:2%;
"
#home-
    margin:auto;
"
header a .title-
    margin: 0%;
    font-size: 50px;
    text-align: center;
    color: black;
"

#date -
    margin-top: 1%;
    font-size: 25px;
    text-align: center;
"

#input-
    margin: 1%;
    font-size: 40px;
    text-align: center;
"
#input .input'date -
    width: 10%;
"

#content -
    width: 100%;

"
#content .all -
    text-align: center;
"

#content .left -
    width: 50%;
    float: left;
    border-right: 5pt;
    text-align: center;
"
/* #content .left .west-
    width: 50%;
    float: left;
    border-right: 5pt;
"

#content .left .east-
    width: 50%;
    float: right;
    border-right: 5pt;
" */

#tb -
    text-align: center;
    font-size: 20px;
"

#content .right -
    width: 50%;
    float:right;
    text-align: center;
"
/* #content .title,h3-
  text-align: center;
" */

#cv-
    text-align: center;
    position: relative;
    margin: 0 5% 0 5%;
"

canvas-
    position: absolute;
```

```
    display: block;
    margin: 0 auto;
"
```

A.8 個人の読み聞かせ動画に表示するプログラム

```
let pause = false

function csvToArray()-
    let csv = new XMLHttpRequest();

    // CSV ファイルへのパス
    csv.open("GET", "../video/kindergarten/2023/02/01/10/ex/0.csv", false);

    // csv ファイル読み込み失敗時のエラー対応
    try -
    csv.send(null);
    " catch (err) -
    console.log(err);
    "

    // 配列を定義
    let csvArray = [];

    // 改行ごとに配列化
    let lines = csv.responseText.split(/"r"n—"n/);

    // 1 行ごとに処理
    for (let i = 0; i ¡ lines.length; ++i) -
    let cells = lines[i].split(",");
    if (cells.length != 1) -
        csvArray.push(cells);
    "
    "

    return csvArray;
"

function outputVideo()-
    var student = document.querySelector("input[name='choice']:checked")
    var cam = document.getElementById("cam");
    if (student != null && cam != null) -
        var cam_num = cam.options[cam.selectedIndex].value;
        var student_num = student.value;
    "
    console.log(cam_num,student_num)

    var a = csvToArray();

    var video = document.getElementById("video");
    var video2 = document.getElementById("video2");


    if(video.paused)-
        video.play();
        video2.play();
    "else-
        video.pause();
        video2.pause();
    "
    pause = false;

    var parent = document.getElementById("parent");
    var child3 = document.getElementById("children3");
    var child = document.getElementById("children");
    var context = child.getContext("2d");
    // graph_active
    // var active = document.getElementById("active");
    // var child2 = document.getElementById("children2");
    // child2.getContext("2d").drawImage(active, 0, 448, 300, 200);
    var x = 0;
    var y = 0;
    var i = 0;

    for(c=0;c¡a.length;c++)-
        if(a[c][3]!=0)-
            x=a[c][3]/3;
            y=a[c][4]/3;
            break;
        "
    "
```

```
    setInterval(function()–
        if(pause)–
            return
        ˝
        parent.getContext(”2d”).drawImage(video, 0, 0, 864, 648);
        child3.getContext(”2d”).drawImage(video2, 0, 648, 864, 400);


        if (student˙num == ”1” && cam˙num == ”cam4”)–
            context.clearRect(0, 0, 864, 648);
            context.font = ’25px serif’;
            context.fillStyle = ’white’;
            context.textAlign = ’center’;
            context.fillText(”出席番号1”, 432, 90);
            context.beginPath();
            context.arrow(420, 100, x, y, [0, 3, -20, 3, -20, 9]);
            context.fillStyle = ’white’;
            context.fill();

            if(a[i][3] != 0)–
                x= (a[i][3])/3;
                y= (a[i][4])/3;
            ˝
        ˝
        i++;
    ˝, 1000/5);
˝

function Pause()–
  var video = document.getElementById(”video”);
  var video2 = document.getElementById(”video”);
  video.pause();
  video2.pause();
  pause = true;
˝

window.onload = function() –
    var student = document.querySelector(”input[name=’choice’]:checked”)
    if (student != null) –
        target = document.getElementById(”video”);
        target2 = document.getElementById(”video2”);
    target.style.display =”none”;
    target2.style.display =”none”;
    ˝
    else –
        target = document.getElementById(”video”);
        target2 = document.getElementById(”video2”);
        target.style.display =””;
        target2.style.display =””;
    ˝

˝
```

A.9 個人の読み聞かせ動画のグラフを作成するプログラム

```
import numpy as np
import csv
import glob
import os
import matplotlib.pyplot as plt
import matplotlib.animation as animation
import math
import japanize˙matplotlib
import cv2

# ——————————————————————–
#                          初期設定部分
# ——————————————————————–
csv˙read˙file˙1 = ”/hdd/kindergarten/code/123/20230201˙100104.200000˙101115.000000˙cam114˙4corner.csv”

col˙names = [”date”, ”id”, ”cam˙number” ,
    ”c1˙x” ,”c1˙y” ,
    ”c2˙x” ,”c2˙y” ,
    ”c3˙x” ,”c3˙y” ,
    ”c4˙x” ,”c4˙y” ,
    ”frame”
    ,”is˙x”,”is˙y”
```

```
    ,"tf_x","tf_y"]

CD = {col_name : num for num ,col_name in enumerate(col_names)}
id_names = ["id0","id1","id2","id3","id4","id5","id6","id7","id8"]

# ————————————————————————————————
#                 関数設定
# ————————————————————————————————
class MakePathArray:
    def __init__(self, directory_name, extension):
        self.file_paths = glob.glob(directory_name + '/*.' + extension)

class CSVOperator:
    def __init__(self):
        pass

    def read_csv_as_2Darray(self, csv_path):
        with open(csv_path, "r") as f:
            reader = csv.reader(f)
            array = [row for row in reader]
        return array

    def save_2Darray_in_csv(self, csv_path, twoDarray):
        with open(csv_path, "w") as f:
            writer = csv.writer(f, lineterminator = '¥n')
            writer.writerows(twoDarray)
        pass

def intersection_point(line1,line2):
    a1,b1 = line1
    a2,b2 = line2
    x = (b2-b1)/(a1-a2)
    y = a1*x+b1
    return x,y

def make_line(x1,y1,x2,y2):
    #y=ax+b
    a = (y2-y1)/(x2-x1)
    b = y1-a*x1
    return a,b


def main():
    csv_operator = CSVOperator()
    csv_path1 = csv_read_file_1

    array_1 = csv_operator.read_csv_as_2Darray(csv_path1)

    arrays_1_casted = []
    for row in array_1:
        row_frame = [None,None,None,None,None,None,None,None,None,None,None,None,None,None,None,None,None]
        for col,value in enumerate(row):
            if (not value == '') and (not col == CD["date"]):
                row_frame[col] = float(value)
            elif col == CD["date"]:
                row_frame[col] = value
        arrays_1_casted.append(row_frame)

    arrays_2_casted = []

    targetid = 0
    #グラフ作成
    #targetid の時
    #arrays_1_casted の x-frame グラフ y-frame グラフ x-y グラフを作成
    #arrays_2_casted の x-frame グラフ y-frame グラフ x-y グラフを作成
    #6 つのグラフを一画面に表示

    #arrays_1_casted の x-frame グラフ y-frame グラフ x-y グラフを作成
    x1_frame = []
    y1_frame = []
    x1_y1 = []
    for row in arrays_1_casted:
        if row[CD["id"]] == targetid:
            x1_frame.append(row[CD["tf_x"]])
            y1_frame.append(row[CD["tf_y"]])
            x1_y1.append([row[CD["tf_x"]],row[CD["tf_y"]]])

    #arrays_2_casted の x-frame グラフ y-frame グラフ x-y グラフを作成
    x2_frame = []
    y2_frame = []
    x2_y2 = []
    for row in arrays_2_casted:
        if row[CD["id"]] == targetid:
            x2_frame.append(row[CD["tf_x"]])
```

```
            y2.frame.append(row[CD["tf.y"]])
            x2.y2.append([row[CD["tf.x"]],row[CD["tf.y"]]])

    fig = plt.figure(figsize=(8.64,4))
    ax1 = fig.add_subplot(1,2,1)
    ax2 = fig.add_subplot(1,2,2)
    ims = []
    for a in range(len(x1.frame)):
        line1 = ax1.plot(x1.frame[0:a],color='blue')
        ax1.set_title("x")
        ax1.set_xlabel("frame[5fps]")
        ax1.set_ylabel("x[mm]")

        line2 = ax2.plot(y1.frame[0:a],color='blue')
        ax2.set_title("y")
        ax2.set_xlabel("frame[5fps]")
        ax2.set_ylabel("y[mm]")
        ims.extend([line1+line2])
        fig.suptitle("出席番号1")
    ani = animation.ArtistAnimation(fig,ims)
    ani.save("anim.mp4",writer="ffmpeg")

    plt.show()

# ------------------------------------------------------------
#                          実行
# ------------------------------------------------------------
if __name__ == '__main__':
    main()
```