**File names and functionality descriptions:**

AV-study/stim\_delivery/ (Folder)

AV-study/stim\_delivery/Results\_Analysis (Folder)

* FindMaxMinSingle.py
  + Analyse the results from the the interp frame, key logs and time
  + Find all the local maxes in the interp frame and time excel file
  + Find all the indexes where spacebar is pressed on (key log is first pressed)
* FindMaxMinsManyAutomate.py
  + Same features as FindMaxMinSingle.py, except this runs the script for all the files in the folder and finds the files which contain the string ‘frames’ and the corresponding trial files with keylogging
  + Make sure to comment out the plot and graph section of the code or else the program will continuously generate and display graphical pop-ups.
* [Cross\_correlation.py](#kdk6mfbn6lg7)
  + pearson and numpy correlation coefficient calculations
* Tompy.py
  + Necessary imports and column checks for coefficient calculations by Tom Irvine

AV-study/stim\_delivery/animationplayer (Folder)

* CreateHdfOnsets/ (Folder)
  + [hdfOnsetReflectedNoBlanks.py](#ema836qsx0p5)
    - Run hdfOnsetReflectedNoBlanks.py to create new HDF files with input specified integers to be “shifted or onset” the specified number of frames. The number of frames
  + hdfOnset.py
    - Simpler features of hdfOnsetReflectedNoBlanks.py, except instead of reflecting the number of frames, blanks (zeros) are added by the input number of positive integers or frames are removed if the input number is a negative integer
  + Other HDF files are the generated output files from either of the two .py scripts, one HDF file must be the file generated from the animationplayer which is used in the script to generate the other HDF files (adds the input integer into the created file name to the name of the original HDF file) (original: 0\_niceBeat.hdf, created HDF file with input integer of 10 0\_niceBeat\_offset\_by\_10.hdf).
* [stimDeliveryWithKeylog.py](#1hplezwieiex)
  + Records time and frames during trials and generates an excel sheet
  + Frames from the HDF files are scaled to a range between 0 to 1 for more efficient and easier use with results analysis and for generating more comparable graphs and data with key logging. This is because keylogging is recorded with 0 to 1, so graphs or other data analysis methods to compare the frames with the same range from 0 to 1, would make analyzing the results more convenient.
* stimDeliveryWithKeylogTwoFiles.py
  + Same functionality as above, except the frames is measured differently.
  + Frames are recorded as the same value as the frame inputs in the given HDF files. If this .py file was used, it can be converted to the scaled values in the above file, but requires more work by using more scripts and reorganizing data in folders.
* stimDelivery.py
  + Backup of earlier renditions of stimDelivery with the added screens and trial messages during the experiment.
* Keylog\_data.py
  + Script for generating a csv (keylog\_data.csv) which records the time and the frequency of the user pressing the spacebar and holding the spacebar during the experiment. (pressing the spacebar indicates the user detected the ball bounced up to the peak)
* AV-study/stim\_delivery/data/trials (Folder)
* hdfConditions (Folder)
  + Place HDF files in this folder to be used for the trials in the experiment
  + Played in the order they are numerically arranged
* wavConditions (Folder)
  + Place wav files in this folder to be used for the trials in the experiment
  + Played in the order they are numerically arranged, the same wav files can be played multiple times when setting up the experiment for the participant

stimDeliveryWithKeyLog.py

* **Functionality**: Runs the experiment with the different trials, displays the interface for the different screens and the trials
* **Input**: takes in the HDF files and the WAV files found in the "AV-study/stim\_delivery/data/participant\_data” folder. Iterates throughout the entire list
* **Output**: two CSV files, one CSV file with the time and keylog data, another file with time and frames displayed per trial. The frame rate can either be converted to be from a scale from 0 to 1, or from the input HDF file which was use to create the trial

FindPeaks.py (DOUBLE CHECK, is this same as findMaxMin?)

* **Functionality**: does the calculations to find where the crests are from the participant recorded data.
* looks at the frames from the HDF excel files and finds the local maximas
* finds all the locations where the key logs are initially pressed (when the keylog was 0 initially which means the space bar was not pressed, until the the spacebar is pressed. (records the indexes of where the the key ogs are from 0 to 1)

cross\_correlation.py

* **Functionality**: calculates the cross-correlation and allows user to select to separate files to analyze what their cross correlation is and multiple statistical results as well as display a chart of the two input files and a third file of the cross-correlation.
* **Input:** select the two files used to calculate the cross-correlation between
* **Output**: on console the statistical calculations and graphs.

hdfOnsetAddBlanks.py

* **Functionality**: adds the user input n number of blanks in front of the given HDF file
* **Input**: HDF file path and the number of frames you would like to offset the HDF file by. Can input a negative number to remove frames and a positive number to add blank frames before the given HDF file
* **Output**: new hdf file with the user input number of blank frames before the original HDF file (positive input value) or the user input number of frames removed from the original HDF file (negative input value)

hdfOnsetReflectedNoBlanks.py

* **Functionality**: creates HDF files with user input amount of frames to offset the input file by.
* **Input**: HDF file created from the animation player or a regular HDF file.
* **Output**: new HDF file with the user indicated amount of offset. Based on the n input from the user, the tail or the head of the file will be reflected by the same number of n frames

Participants

* 3 pilot studies performed, with the same trial conditions
* The first participant did not complete all trials and only obtained up to trial 31,while the other two participants completed up to trial 48.
* All participants were over 20 years old

Independent and dependent variables

The independent variables were the audio files with different frequency (faster, normal, slower) but the same pitch and the visual animations with different amounts of offset (0 (no offset, original), ±20, ±55, ±150, ±200, ±300, ±400 and ±600).

The dependent variables were the time key logs which recorded the time, trial and the times of when the participants responded to the stimuli by pressing the spacebar to indicate the ball was at a crest.