Table of Contents

```
%clc
clear all
```

read song files

```
% choose directory containing songs and read all files
dir_in = uigetdir();
files = dir(dir_in);

% create output text file
f_out = fopen('playlist_out.txt', 'w');
fprintf(f_out, 'reading files from %s \n', dir_in);

tic
fprintf(f_out, 'detecting keys and tempos\n');
availableCount = 0;
for i = 1:length(files)
    % assign each file an id and get file name
    songs(i).id = i;
    songs(i).name = files(i).name;

    % check filetypes and reject if not audio
    if length(songs(i).name) > 4
        if (strcmp(songs(i).name(end - 3:end), '.mp3')
 || strcmp(songs(i).name(end - 3:end), '.m4a') ||
 strcmp(songs(i).name(end - 3:end), '.wav'))
            songs(i).available = true;
            availableCount = availableCount + 1;
        end
    else
        songs(i).available = false;
    end

    % classify key, find tempo, and assign harmonic matches to valid
 songs

    if songs(i).available
        songs(i).tempo = getTempo(dir_in, songs(i).name);
        songs(i).tempoMatches = [0.8*songs(i).tempo
 1.2*songs(i).tempo];
        songs(i).key = getKey(dir_in, songs(i).name);
        fprintf(f_out, '%s is in %s and is %d BPM\n', songs(i).name,
 songs(i).key, songs(i).tempo);
```

```
       %major
       if strcmp(songs(i).key, 'A')
           songs(i).harmonicMatches = {'D', 'E', 'F#m'};

       elseif strcmp(songs(i).key, 'E')
           songs(i).harmonicMatches = {'A', 'B', 'Dbm', 'C#m'};

       elseif strcmp(songs(i).key, 'B')
           songs(i).harmonicMatches =
{'E', 'F#', 'Gb', 'Abm', 'G#m'};

       elseif (strcmp(songs(i).key, 'F#') |
strcmp(songs(i).key, 'Gb'))
           songs(i).harmonicMatches =
{'B', 'Db', 'C#', 'Ebm', 'D#m'};

       elseif (strcmp(songs(i).key, 'Db') |
strcmp(songs(i).key, 'C#'))
           songs(i).harmonicMatches =
{'F#', 'Gb', 'Ab', 'G#', 'Bbm', 'A#m'};

       elseif(strcmp(songs(i).key, 'Ab') |
strcmp(songs(i).key, 'G#'))
           songs(i).harmonicMatches = {'Db', 'C#', 'Eb', 'D#', 'Fm'};

       elseif(strcmp(songs(i).key, 'Eb') |
strcmp(songs(i).key, 'D#'))
           songs(i).harmonicMatches = {'Ab', 'G#', 'Bb', 'A#', 'Cm'};

       elseif(strcmp(songs(i).key, 'Bb') |
strcmp(songs(i).key, 'A#'))
           songs(i).harmonicMatches = {'Eb', 'D#', 'F', 'Gm'};

       elseif strcmp(songs(i).key, 'F')
           songs(i).harmonicMatches = {'Bb', 'A#', 'C', 'Dm'};

       elseif strcmp(songs(i).key, 'C')
           songs(i).harmonicMatches = {'F', 'G', 'Am'};

       elseif strcmp(songs(i).key, 'G')
           songs(i).harmonicMatches = {'C', 'D', 'Em'};

       elseif strcmp(songs(i).key,'D')
           songs(i).harmonicMatches = {'G', 'A', 'Bm'};

           %minor

       elseif strcmp(songs(i).key, 'Am')
           songs(i).harmonicMatches = {'Dm', 'Em', 'F#'};

       elseif strcmp(songs(i).key, 'Em')
           songs(i).harmonicMatches = {'Am', 'Bm', 'Db', 'C#'};

       elseif strcmp(songs(i).key, 'Bm')
```

```
            songs(i).harmonicMatches =
 {'Em', 'F#m', 'Gbm', 'Ab', 'G#m'};

        elseif (strcmp(songs(i).key, 'F#m') |
 strcmp(songs(i).key, 'Gbm'))
            songs(i).harmonicMatches =
 {'Bm', 'Dbm', 'C#m', 'Eb', 'D#'};

        elseif (strcmp(songs(i).key, 'Dbm') |
 strcmp(songs(i).key, 'C#m'))
            songs(i).harmonicMatches =
 {'F#', 'Gb', 'Ab', 'G#', 'Bbm', 'A#m'};

        elseif(strcmp(songs(i).key, 'Abm') |
 strcmp(songs(i).key, 'G#m'))
            songs(i).harmonicMatches =
 {'Dbm', 'C#m', 'Ebm', 'D#m', 'F'};

        elseif(strcmp(songs(i).key, 'Ebm') |
 strcmp(songs(i).key, 'D#m'))
            songs(i).harmonicMatches =
 {'Abm', 'G#m', 'Bbm', 'A#m', 'C'};

        elseif(strcmp(songs(i).key, 'Bbm') |
 strcmp(songs(i).key, 'A#m'))
            songs(i).harmonicMatches = {'Ebm', 'D#m', 'Fm', 'G'};

        elseif strcmp(songs(i).key, 'Fm')
            songs(i).harmonicMatches = {'Bbm', 'A#m', 'Cm', 'D'};

        elseif strcmp(songs(i).key, 'Cm')
            songs(i).harmonicMatches = {'Fm', 'Gm', 'A'};

        elseif strcmp(songs(i).key, 'Gm')
            songs(i).harmonicMatches = {'Cm', 'Dm', 'E'};

        elseif strcmp(songs(i).key, 'Dm')
            songs(i).harmonicMatches = {'Gm', 'Am', 'B'};
        end
    end

end
toc
```

```
Elapsed time is 76.698871 seconds.
```

seed playlist

choose random song to start remove chosen song from list of available songs

```
fprintf(f_out, 'ordering songs \n');
seed = randi(length(songs), 1);
while songs(seed).available == false
    seed = randi(length(songs), 1);
```

```
end
songs(seed).available = false;
availableCount = availableCount - 1;

playlistIDs = [seed];
```

search list for harmonic match

get current key use camelot to determine possible next keys ie if song.key = C song.nextKey = F G or Am find harmonic match in list if multiple matches compare tempos and choose nearest tempo match

```
while availableCount > 0
    currentSong = playlistIDs(end);
    searchKeys = songs(currentSong).harmonicMatches;

    nextCandidates  = [];
    for i = 1:length(songs)
        if songs(i).available == true
            for j = 1:length(searchKeys)
                if strcmp(searchKeys{j}, songs(i).key)
                    nextCandidates = [nextCandidates, i];
                end
            end
        end
    end


    if length(nextCandidates) > 1
        tempoInd = 0;
        for i = 1:length(nextCandidates)
            if (songs(nextCandidates(i)).tempo >=
 songs(currentSong).tempoMatches(1) && songs(nextCandidates(i)).tempo
 <= songs(currentSong).tempoMatches(2))
                tempoInd = i;
            end
        end
        if tempoInd > 0
            nextInd = tempoInd;
        else
            nextInd = randi(length(nextCandidates), 1);
        end
        playlistIDs = [playlistIDs, nextCandidates(nextInd)];
        songs(nextCandidates(nextInd)).available = false;
        availableCount = availableCount - 1;

    elseif length(nextCandidates) == 1
        playlistIDs = [playlistIDs, nextCandidates];
        songs(nextCandidates).available = false;
        availableCount = availableCount - 1;

    else
        hailMary = randi(length(songs), 1);
        while songs(hailMary).available == false
```

```
            hailMary = randi(length(songs), 1);
        end
        playlistIDs = [playlistIDs, hailMary];
        songs(hailMary).available = false;
        availableCount = availableCount - 1;

    end

end
fprintf(f_out, 'enjoy your playlist: \n');
for k = 1:length(playlistIDs)
    fprintf(f_out, '%d: %s \n', k, songs(playlistIDs(k)).name);
end
fclose(f_out);

type playlist_out.txt;
```

```
reading files from C:\Users\mtk\Documents\Undergrad\f2020\eecs
 351\project\songs
detecting keys and tempos
01 Adult Diversion.mp3 is in Am and is 88 BPM
01 Jackie Onassis.mp3 is in C# and is 139 BPM
01 Little Trouble.mp3 is in F and is 141 BPM
01 Somewhere Only We Know.mp3 is in A and is 86 BPM
01 The 2nd Most Beautiful Girl In the World.mp3 is in A# and is 161
 BPM
02 Addictions.mp3 is in D# and is 135 BPM
02 Anagram.mp3 is in G and is 179 BPM
03 First Love _ Late Spring.mp3 is in C and is 200 BPM
03 Kyoto.mp3 is in E and is 130 BPM
03 On + Off.mp3 is in E and is 130 BPM
04 All These Things That I_ve Done.mp3 is in F# and is 118 BPM
08 Direct Address.mp3 is in D and is 70 BPM
ordering songs
enjoy your playlist:
1: 01 Jackie Onassis.mp3
2: 04 All These Things That I_ve Done.mp3
3: 02 Anagram.mp3
4: 03 First Love _ Late Spring.mp3
5: 01 Adult Diversion.mp3
6: 01 Little Trouble.mp3
7: 01 The 2nd Most Beautiful Girl In the World.mp3
8: 02 Addictions.mp3
9: 08 Direct Address.mp3
10: 01 Somewhere Only We Know.mp3
11: 03 Kyoto.mp3
12: 03 On + Off.mp3
```

Published with MATLAB® R2020a