

به نکات زیر توجه کنید:

- کلاس‌های توجیهی پروژه و دیگر مطالب آموزشی با توجه به کتابخانه‌های پایتون انجام می‌شود. اما اجباری به استفاده از پایتون نیست و می‌توانید از متلب هم برای انجام پروژه استفاده کنید.
- فایل تحویلی شما باید شامل سه بخش باشد:
 - (۱) فایل یا فایل‌های کد
 - (۲) نتایج خواسته‌شده در هر بخش، با نام ذکر شده (برای زیاد نشدن حجم فایل ارسالی، زمان فایل‌های صوتی انتخاب شده برای هر بخش کمتر از ۲۰ ثانیه باشد)
 - (۳) گزارش (در صورت استفاده از Jupyter Notebook یا MATLAB LiveScript احتیاجی به تحویل گزارش جداگانه نیست و کافایت در همان فایل Notebook یا LiveScript توضیحات مربوط به کدهایتان را بنویسید).
- گزارش باید شامل توضیح کد و نتایج به‌دست آمده در هر بخش باشد. توضیحی کوتاه و خلاصه برای هر بخش کافایت و نیازی به توضیح بیش از حد (مثلا درباره‌ی جزئیات خط به خط کد) نیست.
- در ابتدای هر فاز پروژه یک کلاس توجیهی برگزار می‌شود. پس ترجیحا تا پیش از برگزاری کلاس فایل پروژه را مطالعه کنید تا با آمادگی بیشتر در کلاس حاضر شوید. می‌توانید ابهامات خود را از درگاهی که در CW درس ایجاد می‌شود مطرح کنید و یا با تی‌ای‌های پروژه به صورت ایمیل در میان بگذارید:

۱. سروین مؤتمن s.motamen@gmail.com

۲. امیررضا حاتمی‌پور arhp78@gmail.com

۳. کیمیا محسنیان mohseniankimia@gmail.com

مقدمه

فایل `GetStarted.ipynb` در اختیار شما قرار داده شده است. در این فایل تابع‌ها و تکه کدهایی که در انجام پروژه استفاده می‌شوند، آمده است. آن را تا پیش از جلسه‌ی توجیهی فاز دوم پروژه مطالعه کنید و سوالات خود را در این جلسه مطرح کنید.

بخش اول - حذف نویز

(۱) فایل‌های صوتی اغلب مقداری نویز دارند. شما در این درس با فیلترهایی مانند `bandpass` و `lowpass` آشنا شدید. ابتدا سعی کنید نویز فایل صوتی با نام `noisy` در فولدر `musics` را حذف کنید تا صدا شفاف‌تر شود. فایل جدید را با نام `noise_reduced` در همان فولدر ذخیره کنید. برای نویزگیری می‌توانید از فیلترهای آماده در کتابخانه‌های پایتون نیز استفاده کنید.

(۲) در این بخش، فایل صوتی `encrypted` در اختیار شما قرار داده شده است. شما باید با اعمال کردن فیلترهایی که تاکنون یاد گرفته‌اید، تلاش کنید که این فایل را رمزگشایی کنید و صدایی قابل فهم از آن استخراج کنید. فایل جدید را با نام `decrypted` ذخیره کنید.

بخش دوم - 3D effect

وقتی از هدفون استفاده می‌کنیم، فایل صوتی از دو کانال پخش می‌شود. اگر صدای پخش‌شده از هر دو کانال یکسان باشد، صدای شنیده‌شده **mono audio** خواهد بود. اگر صدای پخش‌شده از دو کانال متفاوت باشد، می‌توان صدای سه‌بعدی ایجاد کرد، یعنی کاری کرد که شنونده حس کند صدا از نقطه‌ای خاص در فضا به گوش می‌رسد.

(۱) فایل صوتی دلخواه خود را با نام **3D_before** ذخیره کنید. عدد **percent** که بین ۰ تا ۱۰۰ است را از کاربر ورودی بگیرید. سپس کاری کنید که در فایل صوتی خروجی، **percent%** از صدا در گوش راست و **(100-percent)%** از صدا در گوش چپ پخش شود و نام فایل جدید را **3D_after** بگذارید. با گوش دادن به فایل **3D_after** شنونده باید حس کند که صدا به اندازه **percent** درصد در سمت راستش قرار دارد. (اگر **percent** را ۵۰ در نظر بگیریم، صدا **mono audio** خواهد بود).

(۲) فایل **3D_before** را طوری تغییر دهید که صدا مرتباً بین گوش راست و چپ جابه‌جا شود و فایل خروجی را با نام **3D_moving** ذخیره کنید. با گوش دادن به این فایل، شنونده باید حس کند که منبع صدا مرتباً از راست به چپ و بالعکس جابه‌جا می‌شود. (راهنمایی: از تابع سینوسی کمک بگیرید).

* (۳) سعی کنید صدای سه‌بعدی معنادار بسازید. مثلاً مکالمه‌ای که صدای یک نفر از سمت راست به گوش برسد و صدای نفر دیگر از سمت چپ. یا صدای دو ساز که یکی از سمت راست شنیده شود و دیگری از سمت چپ. فایل خروجی را با نام **3D_sound** ذخیره کنید. در گزارش خود توضیح دهید که این فایل را چگونه تولید کردید.

Echo - بخش سوم

هنگامی که در سالنی بزرگ و خالی صدایی تولید شود، این صدا با برخورد به دیوارها با تاخیر دوباره به گوش شنونده می‌رسد. به این افکت پژواک یا اکو می‌گویند. فایل صوتی دلخواه خود را با نام `echo_before` ذخیره کنید. در این بخش عدد `n` و `t` را از کاربر ورودی بگیرید و با تکرار صدای ضبط شده‌ی `echo_before` به تعداد `n` و به فاصله‌های زمانی `t`، صوت دارای پژواک `echo_after` را تولید کنید. (توجه کنید که وقتی در سالن خالی صدا اکو می‌شود، به مرور زمان شدت آن کم می‌شود تا کم کم میرا شود).

برای اطلاعات بیشتر، می‌توانید به لینک های زیر مراجعه کنید:

https://en.wikipedia.org/wiki/Echo#In_music

<https://www.hackaudio.com/digital-signal-processing/echo-effects/>

Phaser Effect - بخش چهارم

از این افکت برای دادن حالتی مصنوعی و الکترونیکی به صدا استفاده می‌شود. مثلاً برای ساختن صدای `C3-PO` در فیلم `Star Wars` از این افکت استفاده شده! [لینک](#)

برای درست کردن این افکت، از `all-pass filter` های متوالی استفاده می‌شود که هر کدام تمام فرکانس ها را عبور می‌دهند، ولی در فاز، سیگنال را شیفت می‌دهند.

فایل صوتی دلخواه خود را با نام `phaser_before` ذخیره کنید. افکت `phaser` را روی آن اعمال کنید و فایل جدید را با نام `phaser_after` ذخیره کنید.

برای اطلاعات بیشتر، می‌توانید به لینک های زیر مراجعه کنید:

[https://en.wikipedia.org/wiki/Phaser_\(effect\)](https://en.wikipedia.org/wiki/Phaser_(effect))

[Analysis of Flanging and Phasing Algorithms in Music Technology](#)