**Project Overview**

The music industry, as compared to itself half a century ago, has grown exponentially. It is an epidemic that is evident through the new genres being formed and the reinventions of old ones. A phenomenon fuelled by advances in technology and the ever-increasing interests of the public. Yet through the seemingly boundless variations that may ever exist, two things remain constant. Music is a series of sounds (or lack thereof) and music exists in time.

The program under consideration supports one of these core attributes. It is a rhythm synchronization tool which is intended to accept and analyze audio input from multiple sources. A source’s input (be it from a traditional musical instrument or person) will be received via the use of a distinct microphone. The audio input will have its properties analyzed to determine the timing of the individual notes/sounds recorded along with tempo of the piece as a whole. The program will go on to compare these properties with those obtained from another source and determine whether or not the sources are "in sync." Once determined, it should display all interpretations (tempo, individual timing of notes and comparison results) to the user in standard music notation, a clear indication of tempo and instructions to attain synchronization. The instructions will be presented as simple messages: "Slow down" if the source is too fast, "Speed up" if it is too slow, or "Just right" if in sync.

**Intellectual Merit**  
Current technologies, such as the instrument lesson tools Yousician and Pianu, implement similar tools. They compare the tempo (and notes) of an instrument to values stored in a preset file and give feedback in real time. However, the proposed program will attempt to simultaneously process and compare data between multiple live instruments. No preset files are needed. Unlike Pianu, the program will not use a MIDI instrument (which directly plugs into computers). In attempting this different implementation, additional verification is required to minimize errors.

To fully implement this project several challenges must be tackled that involve the use of third party APIs along with the appropriate selection & implementation of data structures. The grand scheme requires a mathematical replication of what sounds would seem synchronized to the human ear. However this may be too large a task to directly overcome.   
The first step, determining when to start recording audio input from all sources, would mainly depend on the last source’s preparedness. Then a single source’s audio input will be analyzed to determine timing. Upon surpassing a predefined amplitude, a timestamp must be placed to mark the sound being recognized virtually. The same will be done to record the last instant of the sound being "heard" and the difference will be calculated and stored. This threshold amplitude is an attempt to eliminate background noise from calculations.   
               The program may need to keep track of the largest and smallest differences to determine individual timing notation. From this notation, partitions can be made based on predicted time signatures and tempo may be determined (in beats per minute). As sound is continuously validated and analyzed, data concerning each recording must be stored in some easily expandable data structure (such as a linked list) seeing as the number of data entries would be unknown.

**Broader implications**

The program, as defined thus far, is simply a tool to be used by music enthusiasts to ensure live instrumentation is in sync. For this reason its main purpose is to assist in making harmonious musical pieces. It may shorten the time musicians take to arrange instruments and free them up to handle more creative aspects of production. An alternative social benefit derived from this program would its subtle ability to generate an interest in the theory of music. The notation that may be presented to the user will require some comprehension of the technicalities of the subject (a guide will be provided if such notation is presented).

The program as described could be used as a subset of a larger project where analog sound waves are analyzed in further detail for the purposes of transcription (like writing sheet music) or synthesis (like making program virtually replicate the sound on a different instrument). Such tasks would require an in depth look at the frequency and amplitude of the sound being recorded seeing as these are respectively responsible for pitch and volume.