**Release Plan**

**Heading:**

* Product name: AutoAudio
* Team name: The Awesome Team
* Release name: AutoAudio
* Release date: July 25, 2019
* Revision number: 1
* Revision date: June 29, 2019

**High level goals:**

1. Make a device that can detect sounds and their frequencies
2. Output this information in the format of sheet music, to be used for music composition
3. Implement user interaction to fix/modify the music once it has been written

**User stories:**

* Sprint 1:
  + “As someone who is new to piano, I need to see where the note I’m currently playing shows up on the score, so I can learn to read music.”
  + As a deaf and aspiring singer, I want to be able to classify frequencies to make the learning process much more efficient (Goal: I want an easy interface to compose music so that I am less impeded by my disability)
  + As a music composer, I want a device to automatically transcribe music for me, to simplify a difficult and tedious process. (Goal: a faster way to transcribe music)
* Sprint 2:
  + As a band member, I want a way to turn the notes I play into sheet music so that other members can read my music. (Goal: have music written in a format humans normally read in)
* Sprint 3:
  + As someone who composes music for a living, I want to be able to clearly read the sheet music produced by the program

• User stories for release: A listing of all the user stories that are needed to implement the high level goals. Each user story must have a story point time estimate. Each user story must be assigned to a Sprint, and within each Sprint, listed in priority order from highest (top) to lowest (bottom). Recall that there are 3 Sprints in this class. Each high level goal should have one or more user stories associated with it. User stories that do not correspond to a high level goal, or a high level goal that has no associated user story, are both indications of a lack of project specification. Recall that a user story should take the form, "As a {user role}, I want {goal} [so that {reason}]". A user story should fit on an index card, and meet the "INVEST" criteria (independent, negotiable, valuable, estimatable, sized appropriately, and testable). The complete list of user stories will take the form of:

o Sprint 1 (story points) User story 1 (highest priority for Sprint 1) (story points) User story 2 ... (story points) User story N (lowest priority for Sprint 1, but might get bumped down into Sprint 2 if not implemented in Sprint 1) o

o Sprint 2 (story points) User story 1 (highest priority for Sprint 2) (story points) User story 2 ... (story points) User story N (lowest priority for Sprint 2, but might bump down to Sprint 3) o

o Sprint 3 (story points) User story 1 (highest priority for Sprint 3) (story points) User story 2 ... (story points) User story N (lowest priority for Sprint 3, and would be pushed into an independent study or another class if not implemented)

**Product backlog:**

* Implement user interaction to fix/modify the music once it has been written
* Displaying chords/complex notes
* Displaying key/time signatures and tempo
* Displaying rests (time when no music is played)

• Project presentation: See slides