Pitch Hero

21M.385 Final Project Proposal

# Team

Lisandro Jimenez, Edward Park

# Project Goal

This project is a rhythm and pitch-based game in which users can learn their musical intervals in a call-and-response fashion while also enjoying a fun game. Target users are those who which to become better at easily recognizing and singing intervals in a major scale.

# Project Description

The notes of the music are represented as a series of platforms scrolling towards the left. Each platform should represent a certain pitch of the scale with its height and color (ex: the tonic is green and at the bottom). The left side of the screen contains a ball that bounces on the platforms. If the user sings the note correctly, the ball will successfully bounce on the platform. If not, the ball will fall through, and another replacement ball will be introduced. Whatever pitch the user is singing will be indicated on the left by a picture of a microphone.

There will be two modes: a call-and-response mode, and a regular mode. In a call-and-response mode, the music will be split up into several sections (of four/eight/sixteen beats), and each section will be played twice – once with the notes and once without. In the regular mode, the notes will simply be played in succession.

We wish to use largely instrumental / electronic music, so that the music is still very catchy without being completely recognizable (which would be like karaoke). We will need to find appropriate music files from somewhere.

The user needs a microphone in which he can sing into, and a audio interface so that this microphone data can be sent via USB to the laptop.

# Major Risks / Challenges

We tried experimenting with the audio interface, but we were unable to get it working correctly. Using the Python software with the laptop’s internal microphone worked well, but connecting it to the audio interface didn’t get any input. Hopefully, we can fix this problem, since it is quite necessary.

Another potential problem is that the microphone / software seems to have a slight delay on pitch recognition, which could ruin the synchronization between the audio and the graphics. Hopefully, we can fix this, but if not then we can just unsync the audio and graphics.

# Division of Labor

Lisandro will focus on the audio half of the project – properly getting the audio interface to work with the Python software, finding appropriate songs to use, morphing songs into a series of notes, getting the audio detection and playback to work correctly.

Edward will focus on the graphics half of the project – scrolling the platforms, converting notes into blocks, timing issues, the ball bouncing mechanism, the pitch feedback system, the ball falling/reintroduction upon wrong notes.

Obviously, there is a lot of communication between the audio and graphics halves, so both of us will work on creating this interface when it comes up.

# Timeline / Milestones

For the first milestone, we want the audio and graphics halves to be largely working, though not necessarily together. We want a good song, and we want the audio detection and playback to be largely working. We want the beginnings of the platforms, scrolling, and ball.

For the second milestone, we want to combine these elements together to form a somewhat-working version of our game. This includes audio-graphics synchronization, communication, and input/output feedback. By this point, we should be able to play the game, but not necessarily well.

For the third milestone, we want to finish everything and have it all polished.