Cosmic Scale

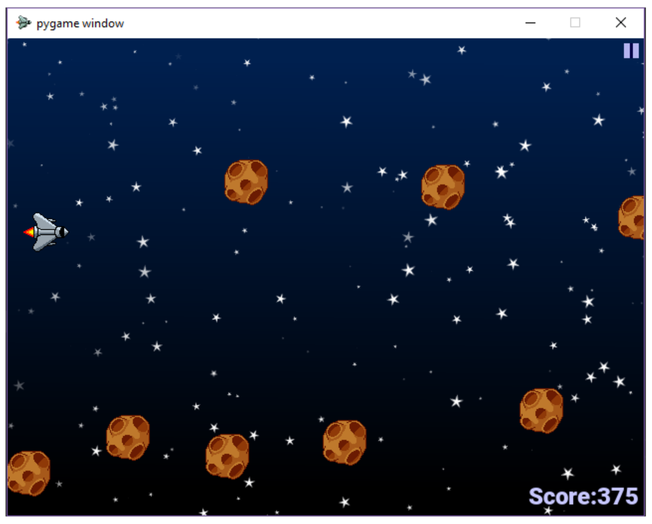
A Pitch-Controlled Endless Runner Game

Chloe Brown | chloebrown2020@u.northwestern.edu || Katrina Parekh | katrinaparekh2020@u.northwestern.edu || Jack Warshaw | jackwarshaw2020@u.northwestern.edu
Final project for EECS 352 at Northwestern University | Instructor: Bryan Pardo

Our Project

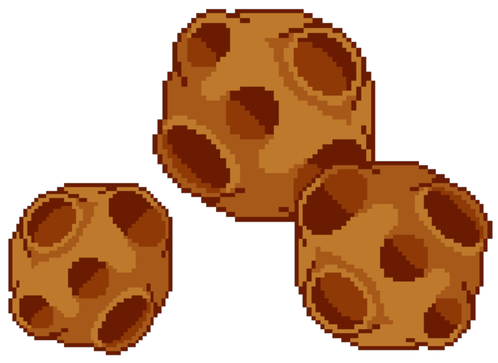
Our goal for this project was to create a simple runner game wherein the player can control their avatar with the pitch of their voice. We aimed above all to create a fun, responsive, challenging game. The nature of the game also means that those with motor disadvantages (ie, shaky hands or poor coordination) can still play.

Design



A screenshot from the in-game window of Cosmic Scale, showing the plitch-controlled player avatar (the rocket) and several asteroid obstacles that the player is trying to avoid.

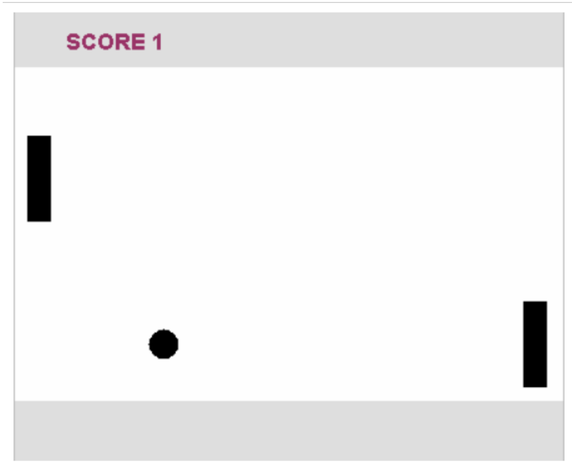
In order to make the game work for all human vocal ranges (and even some instrument ones), the game calibrates before each play instance by asking the user to make a low pitch and then a high pitch. Then, as the game plays, it takes input from a microphone, detects the pitch of the input in real time, and translates that pitch to a position on the game screen relative to the calibrated low and high pitches. The player's avatar follows that position, allowing real-time input based on pitch.



We coded this game using Python 3.7 with the pyaudio [4], aubio [5], and pygame [6] packages. Pyaudio allows us to take input from a microphone as a stream while aubio allows us to detect the pitch in real time. Pygame is a Python library for making games in Python. We picked these packages because they let us work all in Python.

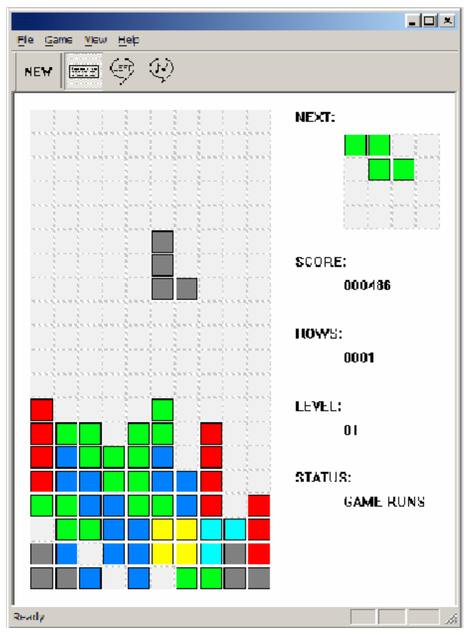
Related Work

Some work has been done in the space of audio-controlled games, though none is exactly like Cosmic Scale.

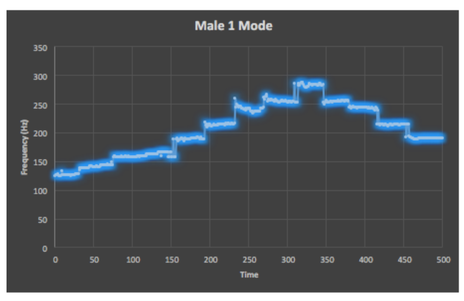


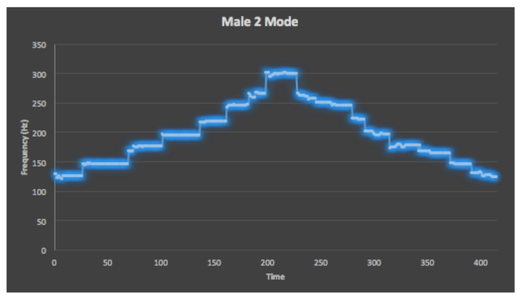
This image is a screenshot of a single-player Pong game built at Helsinki University of Technology. Here, the right paddle moves up and down according to the pitch of the player's voice, while the left paddle is controlled by the computer [1].

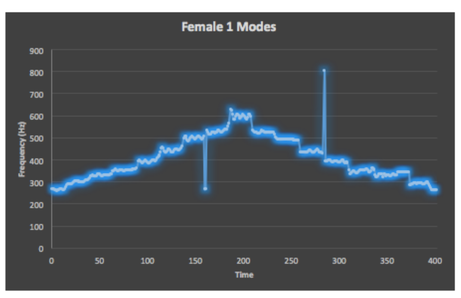
This image is a screenshot of a Tetris game built by Adam J. Sporka et al where the player controls their current piece using different tones in their voice. For instance, a noise that starts lower in pitch and moves higher over time would rotate the piece counter-clockwise [2].

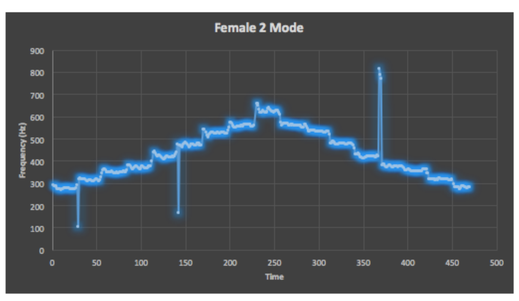


Evaluation









Four graphs of our pitch detection. Each graph shows the mode of our pitch detection over 10 plays through laptop speakers into a microphone. The four singers are all singing scales, the shape of which can be seen in the graphs.

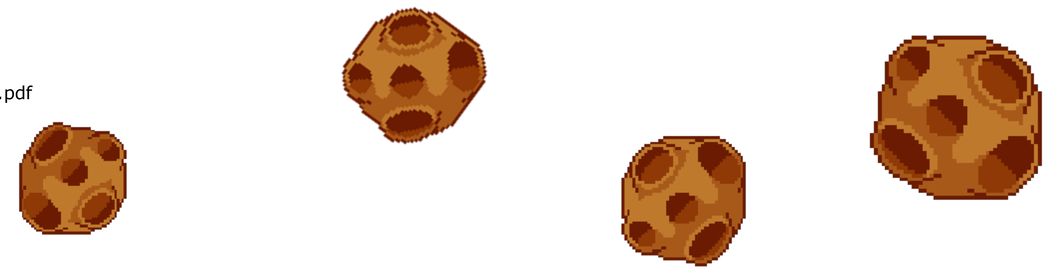
In order to analyze the performance of our pitch detection through aubio, we used the dataset of VocalSet [3], which contains scales by several different singers. We took 10 samples of each voice, recording with the same method each time for consistency. Overall, we found that the pitch processing worked fairly well, as each singer's mean and mode tracks ended up maintaining the same rough shape of their component recordings. Though the mean and mode of the tracks are both fairly accurate, we found the mode to follow the shape of a scale better than the mean, and have based our assesment mainly on those.

In terms of possible errors for data collection, one may be the technique used to get this testing data. The testing data was collected manually, meaning that the speaker quality will impact the pitch detection based on well how the microphone can intake the range and quality of sound output. Although these problems could be solved through automated testing, we wanted to test in the same way that we were taking in data: through the speakers.

Discussion

Cosmic Scale's input functions best in an environment where there is little background noise, as sudden or loud sounds in the background can be picked up by the microphone and cause the player avatar move erratically. It is more difficult than usual to filter out background noise because we are detecting pitch in real-time, so we can only minimally process the input stream before we pass it on to the game's position calculator. To partially mitigate this problem, we have imposed a minimum volume threshold on the input passing.

In the future, we believe it would be interesting to implement speech recognition in the game's menus so it could truly be a hands-free experience. This would allow even more people to play Cosmic Scale.



Sources Cited:
[1] P. Hämäläinen, T. Mäki-Patola, V. Pulkki, and M. Airas, “Musical Computer Games Played by Singing,” p. 5. http://mcpatola.com/publications/dafx2.pdf
[2] A. J. Sporka, S. H. Kurniawan, M. Mahmud, and P. Slavík, “Non-speech input and speech recognition for real-time control of computer games, ” in Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility - Assets ’06, Portland, Oregon, USA, 2006, p. 213. https://courses.cs.washington.edu/courses/cse590w/06au/resources/p213-sporka.pdf
[3] VocalSet Dataset. https://zenodo.org/record/1193957#.XHRjE5NKjOQ
[4] PyAudio. https://people.csail.mit.edu/hubert/pyaudio/docs/
[5] Aubio. https://aubio.org/
[6] Pygame. https://www.pygame.org/