Celestia: A Vocal Interaction Music Game

Yang Shi
Carnegie Mellon University
Entertainment Technology Center
Pittsburgh, PA 15219 USA
yangshi@andrew.cmu.edu

Cheng Yang
Carnegie Mellon University
Entertainment Technology Center
Pittsburgh, PA 15219 USA
chengyan@andrew.cmu.edu

CHI 2013 Extended Abstracts, April 27–May 2, 2013, Paris, France.
ACM 978-1-4503-1952-2/13/04.

Abstract
Voice is one of the most natural means of expression and the vocal interaction is gaining popularity in game development field. In this paper, we present Celestia, a vocal interaction music game that detects different pitches to trigger specific visual events, and explain the design and development phases of it.

Author Keywords
Vocal Interaction; Pitch Detection; Music Visualization; Game Design

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

Introduction
A great deal of research in HCI and CSCW has explored vocal interaction as an emerging interaction style and examined its implications for the design of interactive systems [1]. One mainstream is in game and entertainment community, Sing Party (Fig. 1), for example, is a karaoke-style video game in which players' vocal qualities are visualized as a waveform and scored as the song is performed. In Cello Fortress (Fig. 2), a cellist plays different notes to defend fortress from tanks controlled by players. Despite advances in the diversity of vocal interactive game, the majority

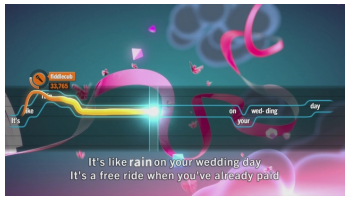

Figure 1. "Sing Party," 2012, by Nintendo for wii U

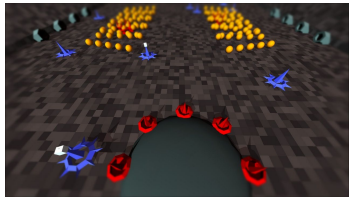

Figure 2. "Cello Fortress," 2012, by Joost van Dongen

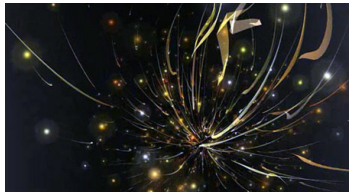

Figure 3. "Magnetosphere," 2007, by Robert Hodgin.

focuses on the task instead of visualizing complex, numerical pitch data as a form of aesthetic art. Reflecting on musical artwork by Robert Hodgin (Fig. 3), we created Celestia. Incorporating visualization technique, the game uses voice input based on pitch detection as a primary means of control, and provides insight into innovation of vocal interaction.

Game Design

From the viewpoint of User-Centered Design [2], we look at the essential elements of Celestia's design through the lens of usability and suitability.

Usability

Celestia detects three specific pitches that are the roots of major chord of the background music. The purpose of the setting is to help users tune by guiding their ear to the tonic chord of background music. Users can improvise a melody by connecting these pitches together as game progresses. In addition to identifying pitch, Celestia responds to amplitude, the louder the volume is, the more powerful the effect will be on the premise that users hit the given pitch. We introduced Celestia to a vocalist who volunteered to improvise the game for a live audience (Fig. 4). The show turned out to be a great success and we received positive feedbacks like "it's visually and aurally appealing".

Suitability

We optimized Celestia around how users can or want to play with it rather than forcing them to change their behavior for the game. We adopted two different pitch ranges to accommodate both female and male voices (pitches D3, G3, D4 for female, pitches A2, C3, D#3 for male). Users have the option to choose the register they feel more comfortable in the calibration settings.

Additionally, Celestia is not confined to human voice, users can play instruments, such as guitar, piano or harmonica to run the game, or more flexibly, make customized vocal tool like whistle or water bells by filling drinking glasses with different amounts of water.

Vocal Interaction

In regard to gameplay, the purpose is to guide a newborn star through the universe with melody. The user's voice can enlarge the star to absorb smaller planets and survive encounters with comets, nebulae. Through the journey, the star gains more energy, displayed through series of changing colors. Every element of the experiential aesthetic is tied to the music; the background constellation is the music visualization with three different colors reacting to high, mid and bass range of the soundtrack in real-time.

When the user sing a D3 (or A2 for men), the diameter of the star's corona expands to absorb nearby planets (Fig. 5). When successfully, the star scatters splendid sparks, providing the user with positive feedback.

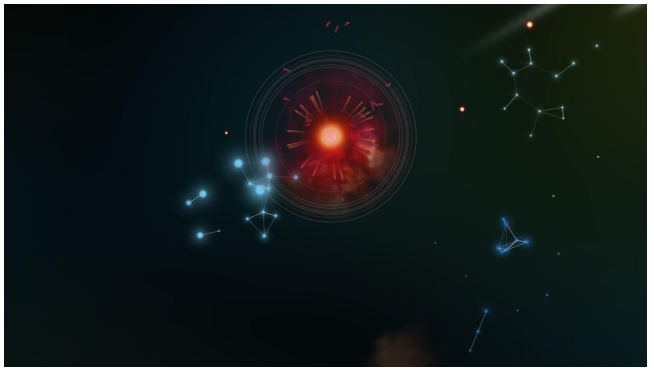

Figure 5. Absorbing Planets

Figure 4. Celestia Live Performance, Dec. 2012

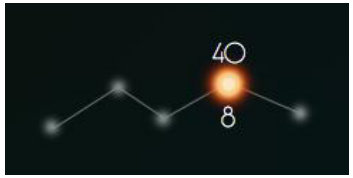

Figure 10. Progress map

When the user sings a G3 (or C3 for men), the star dodges the comets coming from either side (Fig. 6). If the star is hit by a comet, it burns for a few seconds, telling the user that their score has been lowered.

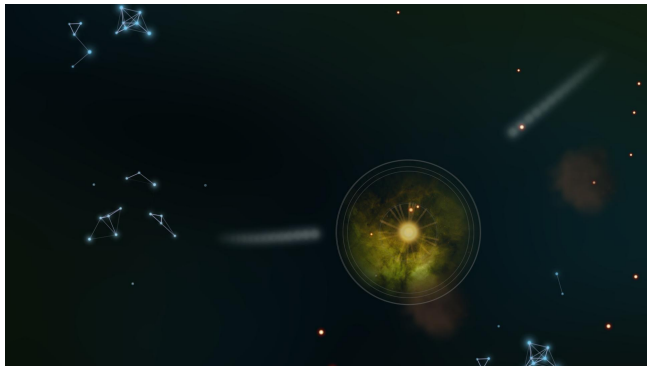

Figure 6. Dodging a Comet

When the user sings the third pitch D4 (or D#3 for men), a burst of energy is released from the star's core to explode the nebulae in its path (Fig. 7).

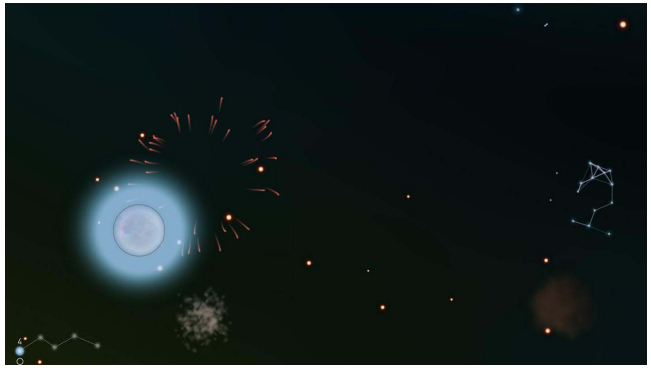

Figure 7. Exploding a Nebula

User Interface

By balancing technical functionality and visual appeal, the user interface of Celestia corresponds to the context of use, making the user interaction as efficient and intuitive as possible. The information presentation, such as navigation element in the Menu (Fig. 8) is consistent with the style of the game, as its layout resembles that of a constellation.

Figure 8. Navigation Menu

The user guidance information, such as the feedback information in the Tutorial (Fig. 9) is displayed as a tree, indicating and evaluating the users' input timely and perceptibly as they build the learning curve. During the game, status information is shown as a progress map (Fig. 10) at the bottom from which users can tell which of the five phases they are in. The figure above shows the how many planets have been absorbed, and the figure below shows how many times they have been hit by comets.

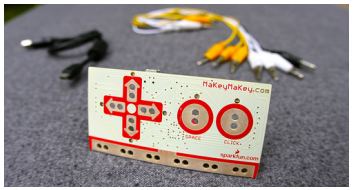

Figure 11. "Makey Makey", 2012, by Eric Rosenbaum, Jay Silver

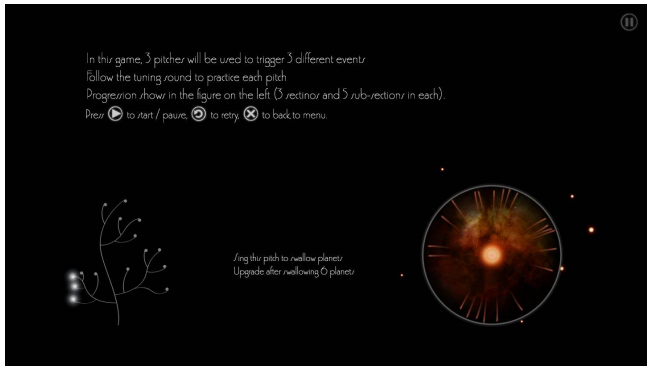

Figure 9. Tutorial

System Description

Celestia is a PC Console Game coded in Processing [3], an open-source programming framework extended from Java. Contributed libraries, Ani and Minim, are included to extend the core functionality.

Celestia adopts Fast Fourier Transform (FFT, which provides fast way to transfer signal into frequency domain) [4] class in the built-in minim library of Processing for spectrum analysis. After buffering the line-in audio signal, we used logarithmic averages to group the spectrum into 10 octaves, and then divided each octave into 12 bands according to the Equal Temperaments. Through individual volume detection from 120 bands, we got the volume changes of every single semitone. We selected featured pitches as triggers according to the music chord feature and human voice range.

Innovative Interface

Celestia's innovation lies in its artistic and illuminating visualization of acoustic input, conveying informative feedback to shape users' appreciation of their vocal quality. And like the Makey Makey (Fig. 11), which extends keyboard functionality to daily objects, Celestia encourages users to explore other musical input possibilities to control the gameplay.

Additionally, pitch can be technically perceived as an intermediate signal between discrete and continuous. While Celestia defines one set of pitches according to the background music choice, more selections can be made based on different preferences of background music for future developing purpose.

Acknowledgment

We would like to express our gratitude towards faculty of Carnegie Mellon University's Entertainment Technology Center.

References

[1] Sri H. Kurniawan and Adam J. Sporka. 2008. Vocal interaction. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '08). ACM, New York, NY, USA, 2407-2410. DOI=10.1145/1358628.1358695 http://doi.acm.org/10.1145/1358628.1358695

[2] Donald A. Norman. 2002. *The Design of Everyday Things*. Basic Books, Inc., New York, NY, USA.

[3] Fry, Ben. 2008. *Visualizing Data*. O'Relly Media, Inc., CA, USA.

[4] Rohde and Schwarz, Dr. Florian Ramian. 2011. *Implementation of Real-Time Spectrum Analysis*.