

**8803 – Fall 2013 – Sonification Lab R&D Studio:**

**Audio Lemonade Stand Game**

Alan Zhang & Eunki Kim (+ Jonathan Schuett)

## Introduction

Everyone loves playing games. People relax, socialize, and learn by playing computer games. However, computer games are not yet accessible to everyone, especially people with vision impairment. Without seeing the cursor on the screen, they cannot play most games using traditional ways of interaction. On the other hand, most of the computer games focus on creating visually stunning graphic interfaces while sounds are just bonuses. For people who cannot see, sound is the main way to present contents of a game, to provide feedback for user interaction, and to determine whether the game is enjoyable or not.

*Lemonade Stand* is a basic economics game created in 1973. Players, acting as lemonade sellers, make choices of advertising and prices regarding different weather and temperature to make profits. Children learn basic economic principles of supply and demand through this game. In the multiplayer version, players compete with each other and gain pleasure from playing together. There are already many other versions of this “old” game on different platforms, but none of them is playable for visually impaired users.

Therefore, the goal of our project is to design and create a Lemonade Stand game, which is accessible and fun to all users regardless of having vision problem. The core of our work is sonification. Sounds can be narrations, sound effects and music. The game itself will also work as a research tool for analyzing how users comprehend and react to different sonifications.

## Methods

To achieve our goal of this project, we had two steps. First step was to code a text-based game in Java for building the basic frame of a game. Next step was to add all sound effects with a Text-To-Speech engine and voice recording to read out texts and present important data such as weather and news. Details of each step will be discussed in the following sections.

## System Design

**Text-Based Game.** Our first alternative to game design method was Unity. It is a game engine for creating interactive 3D and 2D content. It might be good for rendering detailed graphics, but we concluded that it included too many functions compared to our need which was presenting data mainly with sounds not visually stunning graphics. Our final decision was to code in Java and compile in Eclipse.

**JAVA Game Libraries.** The main two libraries used here are LWJGL and Slick2D. LWJGL means lightweight java game library, which is to enable java programmers develop commercial quality games alike by providing access to high performance cross-platform libraries such as OpenGL, OpenCL and OpenAL [1]. Slick2D is an open source, which includes a whole lot of extra functionality to LWJGL including a simple 2D API and a good framework to build games [2].

**State-Based Game.** This game is mainly implemented as a state based game using the Slick2D library. A state based game is a game designed to have several isolated different stages. At any point in time, the game should be in one of these states and when certain conditions are met, it can be transited from the current state to the other expected state. Our game has ten different stages. It includes start screen, menu, introduction, setting, and play. There are four states directly related to play stage. Except for the final report page, three states are looped until the last day of play, which is set by a player.

**Keyboard Input Method.** Main input method of this game is keyboard and interaction is a keystroke. The player continues to play with a space key and goes back to the main menu at any point with an enter key. With arrow keys, s/he can navigate options in main menu. The player needs to input numbers with number keys for deciding the number of lemonade glasses, the price of it and the number of advertising signs. All information is displayed with texts on screen, but it can be delivered by sounds to a player. Sounds are obviously main output method in this game and it will be discussed in the next section.

## Sound Design

**Voice Recordings.** Voice recordings are the most natural ways for text sonification. We let two native English speakers read out the introduction and news in our game. The most significant difference between human and machine voices is emotion - human can make the speech more expressive and convey the meaning more effectively using emotion. Therefore we believe human voices are better for these parts. We also let a number of kindergarten kids yell out “Lemonade Stand” at once to produce our title sound icon, which makes our game attractive to players.

**Text-to-Speech.** In addition to voice recordings, we also implemented some text-to-speech systems. The use of text-to-speech offers slightly more flexibility because they can be changed more often and don’t require us to record additional audio with someone doing the voice readings.

**MaryTTS.** MARY is an open-source, multilingual Text-to-Speech Synthesis platform written in Java. It was originally developed as a collaborative project of DFKI's Language Technology lab and the Institute of Phonetics at Saarland University and is now being maintained by DFKI. As of version 5.0, MARY TTS supports German, British and American English, Telugu, Turkish, Russian and Italian; more languages are in preparation. MARY TTS comes with toolkits for quickly adding support for new languages and for building unit selection and HMM-based synthesis voices [3]. While may not compete with commercial TTS systems, it is one of the most developed high-quality TTS free to public. We use MaryTTS to sonify numbers in the game, such as bank account balances, as they are difficult to sonify using recordings.

**AT&T Natural Voices™.** AT&T Natural Voices™ provides high-quality conversion of text to audio for US English and US Spanish. The system was introduced in 1998 with the name of Next-Generation TTS, featuring naturalness of speech synthesis significantly while maintaining good intelligibility. Now called AT&T Natural Voices™, it is still broadening the range of applications in which TTS can be deployed [4]. We use AT&T Natural Voices to read out texts such as commands and facts in the game, which are more objective and does not require so much expressiveness.

**Weather Sonification.** We use sound clips instead of human voices for sonification to improve the gaming experience. Those sound clips are collected on FreeSound [5], a website aiming to create a huge collaborative database of audio snippets, samples, recordings, bleeps, ... released under Creative Commons licenses that allow their reuse. The sound clips we use are all recordings of natural sounds; including bird and cricket chirping sounds, thunderstorms, and drizzly rains. Recordings of thunderstorms and drizzly rains were expected to be easily perceived as those weather states by the user. Sunny and Cloudy weathers, however, do not have direct sound effects. Therefore we use bird chirping to indicate sunny weather, and cricket chirping sound to indicate a muggy and cloudy weather. These will have to be tested further to determine if users perceive weather states correctly.

**Music.** Music is a fundamental part of computer games. We carefully chose our music from the Internet and used it in our game to make the game interesting to play. This is another area where we intend to make updates as we collect user data.

### Discussion

We skip the result section because we didn't do any study or usability testing by target users yet. Instead, we have lots of things to discuss including what we have learned from this one semester project, what challenges we faced, how we solved it, and what problems are left to the future work.

**What we learned.** The first thing we got from this project is experiencing the whole process of game design. It includes deciding development tools, defining variables, steps in game flows, implementing it, and so on. Of course, the main theme of this project is not to make the most creative game but to make the existing game accessible to visually impaired people, so we don't know about the general design process and tools of developing game such as using graphics. However, we learned the way to understand users and think from their aspects when we designed.

We also learned how to present information with different types of sounds - human voices, machine-generated voices, natural and artificial sounds, and music. In making these sonifications possible, we learned how to find audio clips from the Internet,

how to use TTS libraries, how to play sounds in Java, how to edit audio, etc. In addition, we learned a lot about movie making as we produced the video of our game.

**Challenges.** The challenging issue in developing a game was how to make the game more engaging. The main task of this lemonade stand game is to make more profits by selling as many glasses of lemonade as possible with reasonable price. It is related to the development problem which is how to decide the difficulty of the game by modeling a sale function with several variables such as weather states, temperature, news, and so on. One of main things to be considered was the number of possible people to come and the other one was the probability of buying a glass of lemonade. Since it included more than four variables, it was not easy to model a stable function. This issue still needs to be more discussed.

As for the sonification part, the greatest challenge is to determine what to use to sonify different parts in the game, which can greatly affect comprehensibility of the game and the level of joy which users receive in the game. For example, both artificial and natural sounds could be used to sonify weather. There are also difficulties of choosing technical tools. Free TTS systems do not produce satisfying voices, while commercial ones are not distributable. Lastly, there are problems of sound making and music composition, which is highly demanding, especially for non-music professionals.

**Future Work.** There are a few aspects we want to work on in the future. Usability testing and evaluation are the major parts of our future works. At this time, we only had few feedbacks from normal vision people in the poster session. They found some minor errors in game and told about what impressions they got from our project. Since they are also secondary users of our game, it was helpful. However it didn't include many details about problems of interactions in playing game and something needed to be improved. We would like to receive feedbacks from our main target users, especially visually impaired users, to make the game better.

The game has one difficulty level now. We can add more levels by adjusting the relationship between the factors and the profits. Currently, the game is only for a single player mode, but we would like to add in multiplayer mode and artificial intelligence for players to compete with. Also we would like to research voice input by using speech

recognition and the sonifications can be always improved by adding more sound effects and music.

### References

- [1]. LWJGL, <http://www.lwjgll.org/>
- [2]. Slick, <http://slick.ninjacave.com/>
- [3]. MaryTTS, <http://mary.dfki.de/>
- [4]. AT&T TTS, <http://www2.research.att.com/~ttsweb/tts/demo.php>
- [5]. FreeSound, <http://www.freesound.org/>