Everyone Can Do Magic: An Interactive Game with Speech and Gesture Recognition

Chris Wang, Zhiduo Liu, and Sidney Fels

Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, BC Canada V6T 1Z4
{chrisw,zhiduol,ssfels}@ece.ubc.ca

Abstract. This paper presents a novel game design that allows players to learn how to cast magic spells that combine hand gestures and speech. This game uses the imperfect recognition performance in speech and gesture recognition systems to its advantage to make the game challenging and interesting. Our game uses a Wii remote encased in a wand and a microphone to track player's gestures and speech which are then recognized to determine if they have performed the spell correctly. Visual feedback then provides confirmation of success. Through the game, players learn to adjust their speaking and movement patterns in order to meet the requirements of the recognition systems. This effectively mimics the characteristics of casting spells correctly such that players are trying to adjust their performance so that an "oracle" recognizes their speech and movement to have a magical outcome. A user study has confirmed the validity of the idea and establishes the accuracy required to create an interesting game based on the theory of channels of flow.

Keywords: Interactive game, speech recognition, gesture tracking system, magic.

1 Introduction

Computer game design has been long investigated through the decades. The main goal of game design usually focuses on engagement and entertainment. The ideal achievement of a role playing game or an experimental learning game is to make the player experience a growth in skill and a series of discoveries throughout the gameplay. The main aspects involved in accomplishing this goal are game plot, challenge level, and game interaction [1]. An attractive game plot is the first part of a successful game design. Inspired by the popularity of magic related entertainment products such as books, movies and TV shows, developing a wizard game was our initial motivation of this work. Motivated by the Harry Potter book series [2][3], we conveyed a preliminary survey of 25 participants mostly consisting of graduate students aged between 23 and 35 years old, located across the world including Canada, USA, Singapore, and Norway (14 females and 11 males). Out of all participants, 17 of them were Harry Potter fans and they all showed great interest in games relating to this popular story plot. Additionally, 6 out of the remaining 8 non-Harry Potter fans were also

H.S. Yang et al. (Eds.): ICEC 2010, LNCS 6243, pp. 32–42, 2010.

willing to try a magic-themed game. Among all the participants, 80% thought a spell learning activity in such games met their expectations. This large population of Harry Potter fans in the world provides great potential for this design.

A magic spell consists of a synchronized gesture made with a verbal command. Building a wizard game capturing these features of a spell requires a system with both speech recognition to identify the spell's incantation and gesture recognition to detect the player's motions. The recognition needs to be synchronized as both correct speech and gesture are simultaneously necessary for a successful spell.

Speech recognition has been an active research area for over two decades. Recent work has shown success for speaker dependent, small vocabulary speech recognition system such as voice dialing on cell phones [4], however, due to the complexity of the human language, the accuracy for general speech recognition is still not perfect. Gesture tracking also faces similar problems. Despite being in development for years, there are no non-encumbering systems that can report positions with perfect accuracy and position, while not constricted by various interfering media. To greatly oversimplify, the continuous works are encouraging, but we are still waiting for improved speech and gesture recognition technology to be created.

Yet, the imperfectness within these technologies can be incorporated into our game as challenges, which can actually benefit the enjoyment of games. As shown in the three channel model of game design [5], challenge is an essential element of an attractive game. It was observed that the unpredictable nature of magic matches well with the flaws of these two technologies. The reasonable error rate in the speech and gesture recognition systems can be used to impose challenges on the players. The players are then required to develop their skills throughout the game to overcome the technological difficulties in order to accomplish in game goals.

The remainder of the paper is organized into four sections. Section 2 presents the background about game design and section 3 explains the implemented system. Next, user study results and discussion are presented in Section 4 and future work is presented in Section 5.

2 Background

2.1 Game Design

Studies of experimental learning games by Kiili el. introduced a three channel model to describe game design [5]. Figure 1 shows the diagram referred to as a three channel model of flow. The axes of the diagram are challenges and skills, which are the most important dimensions of the flow experience. As P represents a player, the state of a player may fall into boredom region when challenges are significantly lower than the player's skill level. On the other hand, if the challenges that a player faces are beyond their capabilities, they may feel of anxious, which forms the upper region of the diagram. Flow emerges in the space between anxiety and boredom, which implies that challenges players face in the game closely match their skill level. A well designed game should keep the player within the flow state, where both the player's skill level and challenge level grow at the same pace. Additionally, the flow channel can be extended by providing more guidance to the player or by providing the possibility of solving problems. Our design is able to accomplish this control of flow experience by providing variety of spells with different difficulties.

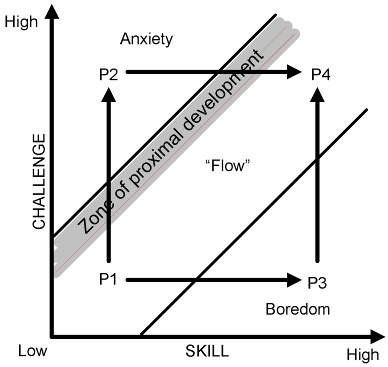


Fig. 1. Three channel model of flow

2.2 Related Work

2.2.1 Interactive Games

We reviewed the literature, and did not find a game which allows the player to be involved with both their speech and body movements except for the unreleased Kinect by Microsoft, which is advertised to have capabilities of natural speech and gesture interaction with players [6]. Although there are many video games related to magic or based on the Harry Potter books and films, such as the Harry Potter video game series developed by EA games and Lego Harry Potter games developed by Traveller's Tales, most are restricted to the visual interface only [7]. The most recently released EA game - Harry Potter and the Half Blood Prince allows very limited simple gesture interactions using the Wii remote when played on the Wii platform [8] but does not incorporate any speech interaction.

2.2.2 Gesture Recognition

The common gesture recognition technique is to equip the hand with a number of sensors which provide information about hand position, orientation, and flex of fingers. Recently, different researchers [9] have congregated to a new device for this problem: the Wii Controller. The key hardware contained in the Wii Controller that has attracted the researchers is its 3-axis accelerometer: Analog Device ADXL330. [10] This hardware emits a timed triplet $(a_x, a_y, a_z)_t$ which represents the acceleration in X,Y and Z axis at sample rates up to 80 Hz. [11]

A recent work on gesture recognition [10] has shown that a 97% recognition rate on user dependent study and 72% recognition rate on user independent study can be obtained using Dynamic Time Wrapping on ten different gestures. The gestures used in that work are similar to gesture patterns used in our game, and therefore, their results can also be applied to our game. This means that through certain user training, high recognition rates can be achieved. Although it is not a perfect system, it is sufficient for the game to be operational since it takes advantage of imperfect recognition as a source of challenge. This user training process is exactly what players in our game would strive for – to match the reference gestures through practice.

2.2.3 Speech Recognition

A typical speech recognition system consists of a language model, dictionary and acoustic model. [12] The language model and dictionary are responsible for splitting up words into their base senones (sub-words) after which the pronunciation of each senone is stored in the acoustic model. There are three possible acoustic models: discrete, semi-continuous and continuous hidden Markov model (HMM). The discrete HMM is the fastest but least accurate [13]. The continuous HMM is the most accurate, but at the sacrifice of recognition speed, as it is the slowest. [14] Finally, there is the semi-continuous HMM which fits between the two previous HMM models, with good accuracy while maintaining speed.

Sphinx4 [13] is an open source speech recognition system that has an approximately 97% recognition rate [16]. Although Sphinx4 does not have perfect recognition rate, it is more than sufficient for the purpose of our magic game, where the 3% error can be used as randomness to mimic the field of magic.

3 Design and Implementation

3.1 Game Design

We selected 7 sample spells with different levels of difficulty to emulate the 7 years of Hogwarts School in our prototype design. This demo version offers the user a sample experience of the game; a complete game would likely include an interactive storyline.

Two modes of play are provided are the adventure mode and free play mode. The adventure mode is a flow mode proceeding through all seven spells with increasing difficulty. The free play mode allows the user the flexibility to choose from any spell to learn and practice.

A typical spell learning process starts with a clear and short tutorial showing how to pronounce the spell, how to move the wand, and how to synchronize the two. A cue is used to signal the user to start casting the spell and the recognition system will output one of four outcomes (success, marginal success, failure, and no effect) based on the closeness of the user speech and gesture against the pre-trained database. All spells are extracted from the original Harry Potter series by J. K. Rowling [2] [3]. According to our survey, this is more desired than randomly creating spells. Corresponding gestures of the spells are creatively designed due to the lack of description of these gestures in the book.

3.2 System Overview

The implemented system has two main goals: to accurately capture and interpret the spells casted by the user. As the spells targeted by our system consisted strictly of speech and gesture, a sound capturing device and an accelerometer are adequate to satisfy the need. The inputs provided to the two aforementioned devices are interpreted by recognition software to determine the accuracy of each casted spell. Based on the evaluation, a graphical output is sent to display the results to the user. An overview of the complete system is shown in Figure 2 and is discussed in detail below.

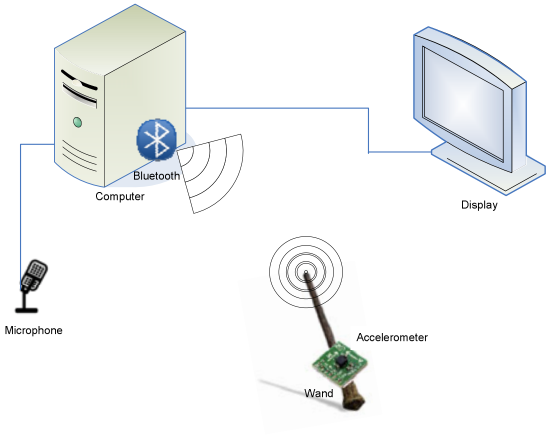


Fig. 2. System overview

3.3 Hardware

Our system consists of a microphone, accelerometer, display and a computer as depicted in Figure 2. For simplicity, we use a laptop containing the microphone, computer, display, and a separate Wii controller with a built-in accelerometer (along with Bluetooth communication components). The Wii controller is used to capture the gestures performed by the user and the microphone is responsible for capturing the speech. The Wii controller and the computer communicate through a Bluetooth connection. A C++ driver created by gl.tter [15] was utilized to facilitate the transmission of data between the Wii controller and the computer. Acceleration triplets, one for each axis{x,y,z}, are used to represent the motion performed by the Wii Controller. The laptop ran Windows 7 with support for C++ and Java programs. In addition, a built-in Bluetooth adapter is included to establish the connection with the Wii remote controller.

3.4 Software

This section discusses the interpretation software employed and the method used to synchronize between the speech and gesture recognition systems.

3.4.1 Speech Recognition

Sphinx-4 developed by CMU [16] was used as the speech interpreter, and was written in Java. This software was chosen because it contains a large vocabulary database, is speaker independent, and has a continuous speech recognition function. Sphinx-4 is based on semi-continuous Hidden-Markov-Model, which matches the input with the most probabilistic word in the database. Our system makes use of the pre-trained Wall Street Journal acoustic models and a custom dictionary was created for each word to indicate the phoneme sequence to pronounce each word.

The speech recognition system saves the matched word and the time which speech was first detected to a text file prior to termination.

3.4.2 Gesture Tracking System

The gesture recognition system is written in C++ and built on top of the Wii remote [15] library which was also written in C++. The Wii remote is sampled at 5 Hz and the result is compared against reference data trained in advance. As gestures vary in length and speed, the total number of data points for each move also varies. This number is fixed during training stage. Sampled user data is clipped from starting time when speech was first detected to the end of gesture length. Comparison is done by calculating error using the Euclidean distance between the two sets of data. The overall error is the arithmetic average of the entire set of data representing the respective gesture.

3.4.3 Integration and Synchronization

The integration between gesture and speech recognition systems was done through synchronized writes and reads from the same text file to allow for communication between the C++ and the Java program. The speech recognition system writes the results to a text file which is read by the C++ program at a later time.

Synchronization information is retrieved by tracking the start time of gesture and speech using the operating system's clock in units of milliseconds. Once the start of speech time is determined, the gesture recognition will search through the accelerometer data and set the closest matching data time as start and proceed with its recognition. Once the start time is determined, the speed of gesture is guaranteed since sampling rate and the number of data points evaluated is constant and the speed of speech recognition system is constant, creating a synchronized overall system. In order to cope with minor system error and human reaction time, an experimentally determined offset of up to 600 ms is accepted between the gesture and speech recognition start time.

3.4.4 Visual Display

There are four possible visual outputs for each spell casted depending on the result of the recognition software. Table 1 presents the visual output possibilities.

Table 1. Visual output decision chart

Speech correct?	Gesture error:	Visual output
Yes	< 0.6	Success
	0.6 – 1.2	Random between shrunk effect and no effect
	> 1.2	Fail
No	< 0.6	Random between shrunk effect and no effect
	0.6 – 1.2	
	> 1.2	Fail

Each spell comes with a corresponding tutorial and each tutorial is an animation consisting of spell pronunciation, wand trajectory, and temporal relation of phonemes in the voiced commands. Figure 3 shows a non-animated picture of the tutorial. The solid black line indicates the wand's path. Phonemes along this line indicate desired synchronization between speech and gesture. After the tutorial, a real scenario would appear as a cue to signal the player to start casting the spell. In the case of 'accio' shown in Figure 4, the scenario is a toad to be magnified, and a successfully casted spell will be an enlarging toad as shown in Figure 6. All animations are made in Adobe Flash CS4 and the GUI is written with Visual Basic. The graphics are quite different from a real video game but is enough for constructing a primitive framework of the entire design.

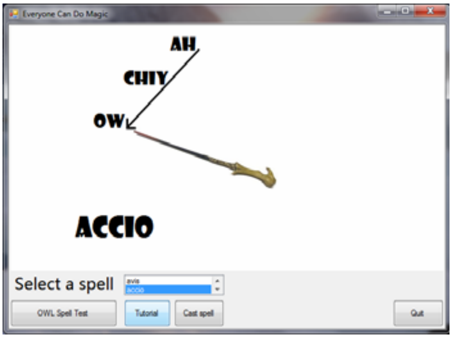
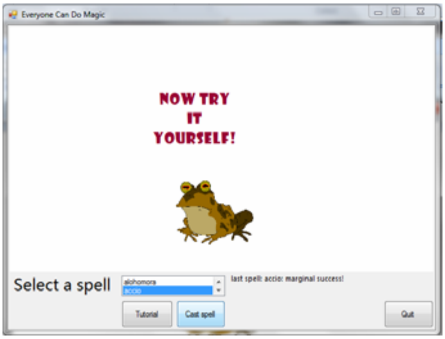
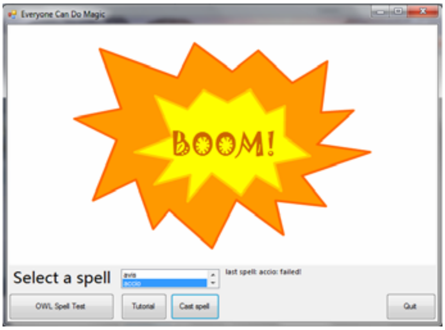


Fig. 3. Sample tutorial



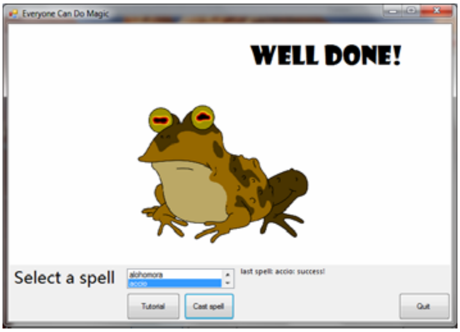
last spell: accio: marginal success!

Fig. 4. Cue for user to being casting



last spell: accio: failed!

Fig. 5. Failed Spell Cast



last spell: accio: success!

Fig. 6. Success Spell Cast

4 User Study

We adopted a common approach: play testing to evaluate our game design. Twenty-nine subjects (17 males, 12 females) were recruited to participate in the volunteer study (UBC ethics code H07-03063). The principle dependent variable is whether the user enjoyed the game while the independent variable is percentage of spells cast that were successful based on adjusting decision thresholds to make it harder or easier to cast successfully. We call this our challenge level.

4.1 Test Procedure

The user survey was conducted in three different groups based on the success rate of all spells, which were categorized as low, medium, and high. The procedure for each group was the same.

The experimental procedure was first explained to the survey participant. The game then started with a practice mode allowing subjects to manually choose from a list of spells and practice each one at most three times. We allowed players to proceed and see all spells even if they did not pass in the three trials to control the experiment time.

After the user exhausts the maximum number of tries for each spell or upon request, the game will enter the test mode, which loops through all seven spells and records the subject's performance. Our challenge level was adjusted randomly to control the overall success rate in the testing mode for different subjects. Each participant was required to fill out a questionnaire regarding to their experience at the end. We note the number of spells passed on the paper for result analysis purposes. In particular, each subject was inquired about any suggestion for improvement they have, as well as their impressions on the game. For this, they were given choices of anxiety, enjoyment or boredom to describe their feelings through the game.

4.2 Experimental Results

Figure 7 shows the distribution of total performance in test mode for all subjects incorporating their emotions. It can be seen that the highest percentage of each group of participants that felt the game was enjoyable were participants who achieved success rate between 30%-84%. 67% of participants in the high success group were bored, and there was an approximate even distribution of emotions from the subjects in the low success group.

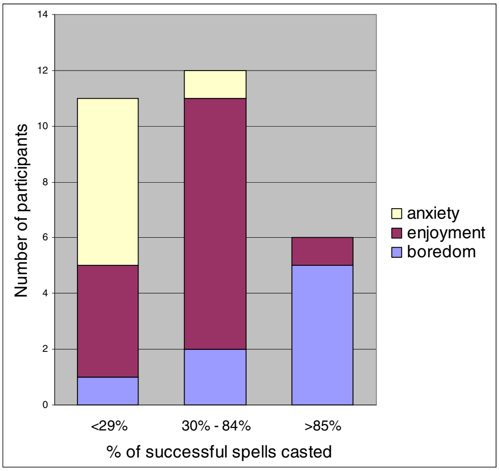


Fig. 7. Game enjoyment vs spell success rate

Figure 8 shows the user enjoyment level of the game with respect to whether the participant is a frequent gamer or not. It can be seen that only 21% (3 out of 14) of frequent gamers didn't enjoy the game, whereas 53% (8 out of 15) of the non-gamers did not enjoy the game.

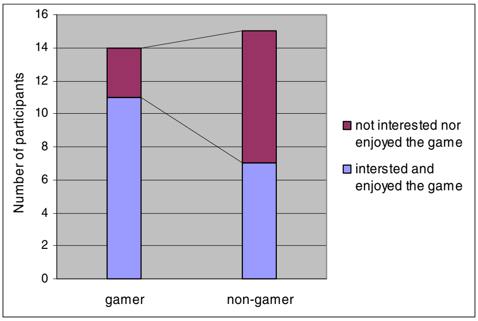


Fig. 8. Game enjoyment vs participant's gaming background

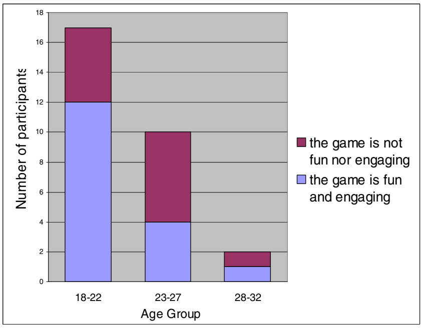


Fig. 9. User feeling vs age

Figure 9 shows user engagement in the game with respect to the participant's age. The majority (71%) of participants in the age group between 18 and 22 years old felt the game was fun and engaging and 40% of the participants in the age group between 23 and 27 years old gave positive feedback. An even split was seen amongst the third group which includes participants between 28 and 32 years, however, this data is inconclusive since the sample size is too small.

4.3 Discussion

Results from Figure 7 suggest that a success rate between 30%-84% provides a significant chance of flow experience where players are most likely to be intrigued by the game. On the other hand, excessive success rates depreciate the enjoyment of the game, which leads to boredom. The result for the low success rate group suggests that participants can still enjoy the game by aggressively improving their skills and striving into the "flow" region despite falling transiently into the anxiety zone.

Figure 7 also demonstrates the correlation between challenge level and user experience, thus, verifying the hypothesis of the three channel model of flow presented in Figure 3 is applicable to our design. One way to obtain the desired success rate on users is to setup an adaptive system where the evaluation criterion becomes more stringent as the player's success percentage exceeds a particular threshold, such as 84%. This way, all players, regardless of skill level can feel challenged without changing the spells. On the contrary, the evaluation criterion could also become lenient for players with a success rate below a specified threshold, such as 30%. The adjustment of evaluation criterion in the background should not be detected by the players, and this adaptive system could be used to maintain the engagement level of all players. Naturally, more complex magical spells can be introduced to provide increasing complexity for advanced skill levels.

By observing the subjects' performance, we noticed that spells with simple speech and gesture are easier than those that consist of a relatively long speech with multiple phonemes and a more intricate gesture motion path, which confirms our preliminary assumptions on spell difficulty.

Results from Figure 8 infer that frequent gamers were much more engaged to the game than non-frequent gamers. Given that frequent gamers are likely to be exposed to various types of games, their past experience establishes a guide for comparing our game against the ones previously played. Positive feedback from them demonstrates that our game has great promise for future investments.

Based on the statistics from Figure 9, our game was more appealing to the age group between 18 and 22 than older age groups. Due to this trend, as well as the target age group for many popular magical paraphernalia, we believe there is likely to be an increasing trend on even younger populations, specifically between ages 13-17 who would be attracted by the game.

Overall, the subjects were fascinated by the game despite that 6 out of 29 testers thought the game's visual interface could be improved, which encourages further investigation on this type of game. The novelty of the design idea and integrated system with both speech and gesture recognition technology explores a new space for future game development. Many subjects recommended that a better plot, background music and costumes such as a pointy hat for the players would further improve game quality.

5 Conclusion

In this paper, we presented the design of an interactive game using speech and gesture recognition systems. We created a framework with a spell casting environment which took advantage of the imperfection of the aforementioned technology to portray randomness in the magical world. The user study showed that frequent gamers aged under 22 could be the main target audience for such a game. In order to maintain the player's interest, the design should follow the principles of the three channel flow model and the spell casting success rate should be kept within 30% and 84%. This success rate would avoid leaving players with feelings of anxiety or boredom.

6 Future Work

Current data suggests that most players were only able to successfully cast 2 or 3 spells out of the 7 with minimal practice. As we plan to introduce a more complete story plot, more spells are needed.

Due to ethical issues involved with running experiments on minor subjects, we were unable to obtain feedback from users in younger age groups. As seen from Figure 10, there is a clear trend of increasing interest for our game for younger participants. Based on this, more experiments on younger subjects could be done to obtain important user feedback.

Despite a suboptimal storyline, enhancing visual interface with more vivid animations and completing the storyline with more spells and versatile activities are the main future work of this design. Some details such as flexibility of skipping part of the introduction and tutorial more help information, and better wand design could also help make it more like a commercial game. Furthermore, user customizations such as selecting courses according to user interest and picking their own wands can be added to furnish the design.

Another aspect that is missing in our prototype system is providing audio clips for visual animations. A door that opens with a squeaking sound is much more satisfying than only seeing the door open on the visual output. The user would thus experience a more realistic environment and surely be more immersed in the game.

Unifying the programming language for speech and gesture recognition software can further improve the synchronization and the integration of the overall system. One possibility is to rewrite either the speech recognition in C++ or rewriting the Wii remote driver in Java. Either way, using one language would make it easier to access intermediate recognition information by the other system, allowing better synchronization schemes.

Acknowledgements

This work has been partially supported by The Institute for Computing, Information, and Cognitive Systems (ICICS) at University of British Columbia.

References

1. Pinelle, D., Wang, N., Stach, T.: Heuristic Evaluation for Games: Usability Principles for Video Game Design. In: CHI 2008, vol. 5, pp. 1453–1462 (April 2008)
2. Rowling, J.K.: Harry Potter and the Philosopher's stone (1997)
3. Rowling, J.K.: Harry Potter and the Goblet of Fire (2000)
4. Lecture 12: An overview of Speech Recognition, http://www.cs.rochester.edu/u/james/CSC248/Lec12.pdf (retrieved February 25, 2010)
5. Kiili, K.: Content creation challenges and flow experience in educational games: The IT-Emperor case. The Internet and Higher Education 8(3), 183–198 (2005)
6. http://www.xbox.com/en-US/kinect/default.htm (retrieved June 18, 2010)
7. http://en.wikipedia.org/wiki/Lego_Harry_Potter (retrieved March 12, 2010)
8. http://www.ea.com/games/harry-potter-half-blood-prince (retrieved March 12, 2010)
9. Buxton, B.: Chapter 14: Gesture Based Interaction (May 2009), http://www.billbuxton.com/input14.Gesture.pdf (Retrieved March 6, 2010)
10. Leong, T., Lai, J., Pong, P., et al.: Wii Want to Write: An Accelerometer Based Gesture Recognition System. In: International Conference on Recent and Emerging Advanced Technologies in Engineering 2009, Malaysia (2009)
11. Mlích, J.: Wiimote Gesture Recognition. In: Proceedings of the 15th Conference and Competition STUDENT EEICT 2009, vol. 4, pp. 344–349 (2009)
12. Q6.1: What is speech recognition?, http://www.speech.cs.cmu.edu/comp.speech/Section6/Q6.1.html (retrieved February 25, 2010)
13. Walker, W., et al.: Sphinx-4: A flexible open source framework for speech recognition (2004), http://research.sun.com/techrep/2004/smli_tr-2004-139.pdf
14. Huang, X., Alleva, F., Hon, H.W., Hwang, M.Y., Rosenfeld, R.: The SPHINX-II speech recognition system: an overview. Computer Speech and Language 7(2), 137–148 (1993)
15. http://wiiyourself.gl.tter.org/ (retrieved March 16, 2010)
16. http://cmusphinx.sourceforge.net/sphinx4/ (retrieved March 18, 2010)