

## **Second language learning using eBooks**

**Tiago Vaz Duarte Soares**

Thesis to obtain the Master of Science Degree in

### **Electrical and Computer Engineering**

Supervisors: Prof. Isabel Maria Martins Trancoso  
Dr. Xavier Anguera

#### **Examination Committee**

Chairperson: Prof. António Manuel Raminhos Cordeiro Grilo

Supervisor: Dr. Xavier Anguera

Member of the Committee: Prof. Ralph Rose

**July 2017**



*This thesis is dedicated to my parents  
for their continued support and  
encouragement.*



## Acknowledgments

First and foremost, I would like to thank my thesis supervisors Isabel Trancoso and Xavier Anguera for the expertise, patience and guiding extended to me.

I am also grateful to INESC-ID's Spoken Language Systems Laboratory for the resources provided without which development would not have been possible.

And finally, I would like to thank my parents for their continued support and encouragement; my brothers and sister Raquel, Jorge and Tó for always being supportive throughout the writing of this thesis; and to my nephews and nieces Henrique, António, David, Miguel, Isabel, Vitória, Glória and Maria da Luz for the happiness they have brought into my life during this time.

A todos, muito obrigado.



## Resumo

As ferramentas actuais de aprendizagem de linguagem assistida por computador (CALL), que permaneceram em grande parte inalteradas nos últimos anos, são de um modo geral consideradas inadequadas para alcançar a fluência numa língua estrangeira. O presente projecto tem como objectivo construir uma ferramenta CALL baseada em eBooks bilíngues, seguindo o formato EPUB3. A ferramenta deve ter uma interface de utilizador intuitiva, incluir definições de palavras e também incorporar um conjunto de jogos sérios.

O projecto englobou o desenvolvimento de um script Python que produz um único documento eBook com recursos dinâmicos dados os textos paralelos em ambos os idiomas e áudio pré-gravado na língua alvo. Estes recursos oferecem ao utilizador um controlo melhorado sobre a reprodução de áudio e não estão incluídos por defeito na plataforma alvo, o iPad da Apple neste trabalho. O script também carrega definições de cada palavra no eBook e inclui-as no documento final. Além disso, o projecto também incluiu o desenvolvimento de uma aplicação Javascript que implementa os jogos sérios sem depender de recursos externos.

Realizou-se um estudo preliminar com um eBook de demonstração gerado por este sistema, para melhorar suas características e entender as expectativas que os utilizadores têm desta ferramenta. Apesar do parecer positivo do grupo de teste sobre a utilidade desta ferramenta, o interesse revelado por funcionalidades que exigem uma análise mais profunda do livro bilingue suscita preocupações sobre a viabilidade da aplicação ibooks da Apple e o formato EPUB3 face a uma implementação da solução como uma aplicação standalone.

**Palavras-chave:** Aprendizagem de línguas assistida por computador, jogos sérios, EPUB, livros bilíngues, eBooks interactivos.





## Abstract

Current Computer Assisted Language Learning (CALL) tools, which have largely remained unchanged over the past few years, are often inadequate for attaining fluency in a second language. The goal of this work was to build a CALL tool based on bilingual eBooks, following the EPUB3 format. The tool should have an intuitive user interface, include word definitions, and also incorporate a set of serious games.

The project comprised the development of a Python script which took the parallel texts in both languages and pre-recorded audio in the target language and merged them into a single structured eBook document, complete with dynamic features. These features offer the user an improved control over the audio playback, and are not included by default in the target platform, Apple's iPad in this work. The script also retrieves dictionary definitions for every word in the eBook and includes them in the final document. Additionally, the project also comprised the development of a Javascript application that implements the serious games without relying on online external resources.

A preliminary user study was conducted with a demonstration eBook generated by this system in order to improve its features and understand the expectations users would have for the eBook. Although positive feedback reassures the usefulness of this application, the demand for features that require deeper analysis of the dual language book raises concerns over the viability of the chosen reading platform, Apple's iBooks application, and the EPUB3 format versus implementing the solution as a standalone mobile application.

**Keywords:** CALL, serious games, EPUB, bilingual books, interactive eBooks.



# Contents

Acknowledgments . . . . .	v
Resumo . . . . .	vii
Abstract . . . . .	ix
List of Figures . . . . .	xiii
List of Acronyms . . . . .	xv
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Topic Overview . . . . .	2
1.3 Related work . . . . .	3
1.4 Objectives . . . . .	4
1.5 Contributions . . . . .	4
1.6 Organization of the thesis . . . . .	4
<b>2 Background</b>	<b>7</b>
2.1 Theories for language learning . . . . .	7
2.1.1 Computer-assisted language learning . . . . .	7
2.1.2 Serious games . . . . .	8
2.2 Text and audio processing algorithms . . . . .	11
2.2.1 Parallel text alignment . . . . .	11
2.2.2 Forced alignment and speech recognition systems . . . . .	13
2.3 EPUB 3.0.1 . . . . .	14
<b>3 Specification and Design</b>	<b>21</b>
3.1 Specification . . . . .	21
3.2 Design . . . . .	22
3.2.1 Parallel books interface . . . . .	22
3.2.2 Serious games . . . . .	26
<b>4 Implementation</b>	<b>31</b>
4.1 EPUB generator . . . . .	31
4.2 Dictionary manager . . . . .	33

4.3	SMIL emulator . . . . .	34
4.4	Serious games . . . . .	36
<b>5</b>	<b>Results and Evaluation</b>	<b>39</b>
5.1	Survey hypotheses . . . . .	40
5.2	Informal discussion . . . . .	41
5.2.1	Current functionalities . . . . .	41
5.2.2	Desired implementations . . . . .	42
5.2.3	Other applications . . . . .	43
5.3	Formal survey . . . . .	43
5.3.1	General demographic questions . . . . .	43
5.3.2	Survey results for eBook features . . . . .	45
5.3.3	Survey results for serious games . . . . .	46
<b>6</b>	<b>Conclusions and Future Work</b>	<b>49</b>
6.1	Conclusions . . . . .	49
6.2	Future Work . . . . .	50
	<b>Bibliography</b>	<b>51</b>

# List of Figures

2.1	The gamification process. . . . .	9
2.2	Example gamification elements. . . . .	10
2.3	Overview of a typical HMM-based ASR system. . . . .	14
2.4	Files typically found in EPUB archives. . . . .	15
2.5	Example content of a container.xml file. . . . .	15
2.6	Example of a Publication Document file . . . . .	16
2.7	Example of a PLS lexicon document. . . . .	17
2.8	Linking PLS lexicons to a content document. . . . .	18
2.9	Example of SSML usage. . . . .	19
2.10	Example of a .xhtml text file. . . . .	20
2.11	Example of a .smil file. . . . .	20
3.1	High-level overview of the EPUB Generator. . . . .	24
3.2	Detail of eBook graphical user interface. . . . .	24
3.3	Location of the graphical user interface. . . . .	25
3.4	Detail of a word definition in the eBook. . . . .	26
3.5	Overview of a word definition in the eBook. . . . .	26
3.6	Sample from <i>Fill in the blanks</i> game. . . . .	27
3.7	Overview of the <i>Fill in the blanks</i> game. . . . .	27
3.8	Overview of the <i>Word blitz</i> game. . . . .	28
3.9	Detailed view of the hint system. . . . .	28
4.1	Sample from query for the word " <i>mestre</i> ". . . . .	33
4.2	Simplified activity diagram of <code>rb_smil_emulator.js</code> . . . . .	34
4.3	CSS redefinition for non-permanent word definitions. . . . .	35
4.4	Workaround to the timing issue. . . . .	36
5.1	Survey: Participants' ages. . . . .	44
5.2	Survey: Participants' genders. . . . .	44
5.3	Survey: Participants' country of origin. . . . .	44
5.4	Survey: Number of spoken languages. . . . .	44
5.5	Survey: Number of self taught languages. . . . .	44

5.6	Survey: Opinion of dual language eBooks in general. . . . .	45
5.7	Survey: Opinion of efficacy of dual language eBooks. . . . .	45
5.8	Survey: eBook question 1. . . . .	45
5.9	Survey: eBook question 2. . . . .	46
5.10	Survey: eBook question 3. . . . .	46
5.11	Survey: Prior usage of gamified language learning approaches. . . . .	46
5.12	Survey: Evaluation of gamified language learning approaches. . . . .	46
5.13	Survey: Serious games question 1. . . . .	46
5.14	Survey: Serious games question 2. . . . .	46
5.15	Survey: Serious games question 3. . . . .	47
5.16	Survey: Serious games question 4. . . . .	47

# List of Acronyms

<b>ASR</b>	Automatic speech recognition.
<b>CALL</b>	Computer-assisted language learning.
<b>HMM</b>	Hidden Markov model.
<b>IPA</b>	International phonetic alphabet.
<b>L2</b>	Second language.
<b>PLS</b>	Pronunciation lexicon specification.
<b>SMIL</b>	Synchronized multimedia integration language.
<b>SSML</b>	Speech synthesis markup language.
<b>TTS</b>	Text-to-speech.





# Chapter 1

## Introduction

Effects of globalization in recent years have created a demand for bilingual and multilingual individuals. Several languages such as English and Spanish have become highly sought after by employers for their value in international communication, the tourism industry and in trade. Other emergent languages, such as Mandarin Chinese, are also sought for the value it is perceived of them from times to come. Migration has also influenced these demands as it shapes countries into multilingual and multicultural societies, and so has the Internet with its accessibility to readily connect individuals anywhere around the globe. It is for these reasons that people and countries as a whole seek to prepare themselves to deal with the challenges of a modern bilingual world by instructing themselves and their youth in a second language (L2). These efforts translate into years of educational work in classrooms for the youth, usually supplemented by enrollment in specialized language schools in order to develop a wider knowledge of the foreign language, the latter generally being one of the few options for adults.

### 1.1 Motivation

Despite the vast accessibility to foreign media, which supplies a plethora of means of study and opportunities for enriching oneself with new knowledge of the language to advanced students, most of the tools available to beginner and intermediate L2 learners have remained unchanged over the past few years. These instruments are generally textbooks, audio compact discs and other similar media, which allow students to study and learn some new insights into the L2 on their own. Proficiency in L2, however, comes from achieving a degree of fluency, complexity and accuracy. These are attained from acquiring a competence in understanding formulaic expressions and knowledge of the L2 grammatical rules (Skehan, 1998). For a student to obtain these skills, studies suggest that instruction is required to revolve around an extensive input in the L2 as well as focusing on the meaning of its formulaic expressions, but to also give the student the possibility of output (Ellis, 2005).

This work proposes a method to automatically generate bilingual content for L2 learners to immerse themselves in input from the second language. This content will be conceived in the form of eBooks, a form of media readily available in contemporary electronic devices, such as tablets and smartphones,

thus allowing the L2 learner the freedom to access their learning tools at their convenience.

## 1.2 Topic Overview

Dual language books have been used successfully in past research to improve literacy of kindergarten students from different language backgrounds in regard to knowledge of printed letters and words in contrast to a control group of students who only heard the stories in English (Naqvi et al., 2013b,a; Ulanoff and Pucci, 1999). Such books are written in two languages, in which typically one page is written in one language and the adjacent page in a second language. They are designed in such a way as to allow the reader to simultaneously read in both languages. An earlier study had determined that dual language books are fundamental for these children in regard to maintaining their cultural identity, interest in the written form of their own language and stimulated the use of their mother tongue in a home setting. Beforehand they were becoming subtractive bilingual to the problematic point of replacing the use of their mother tongue with English at home (Sneddon, 2008). It was due to the use of these books that teachers managed to create a space for the children in an overwhelmingly English environment, in which they could explore their heritage and multicultural identities as opposed to focusing entirely on their development of English literacy (Creese et al., 2006).

Computer-assisted language learning (CALL) is a concept that encloses all applications of computer-based solutions in the field of language learning and instruction. Generally such systems employ either speech synthesis or prerecorded audio to supplement existing reading material with phonetic correctness. Speech synthesis systems are applications which permit generating speech which closely resembles human speech. They are of the utmost interest to CALL and our study as they allow the creation of any utterance given any correct input. Extensive research has been done in regard to the use of speech synthesizers in the fields of reading assistance for middle-school students, college students and adults with learning disabilities (Elkind et al., 1996; Elbro et al., 1996; Elkind, 1998; Anderson-Inman, 1999; Dolan et al., 2005). These studies have shown to produce reading comprehension, clarity and attention span gains in all learning impaired individuals who had reading rates, in words per minute, that were lower than those of their peers. In addition, the expected gains from the use of speech synthesizers is in an inverse correlation with the subjects reading rate, meaning that the lower the reading rate of the user the more relevant the effects of Text-to-speech (TTS).

Nevertheless, studies on the efficacy of TTS in CALL have demonstrated that the state-of-the-art speech synthesizing technologies used at the time were of inconclusive utility (Handley, 2009). Participants in this study evaluated four different TTS synthesis systems for their utility in CALL. Their assessment was completed on each system's utility in the following applications: as a reading machine, phonetic pronunciation model, prosodic pronunciation model or conversational partner. For each application a score was given for adequacy, comprehensibility, intelligibility and expressiveness. Of the four systems, only one received favorable ratings. In fact, most aspects related to quality of speech obtained a close to top rating, which would indicate that this one system in particular was not far from meeting the requirements needed for CALL.

Studies on serious games have demonstrated that by keeping its participants engaged and motivated they can be a robust tool in the study of Second language (L2). One such study revealed that primary-school-age children benefited from the problem solving and game activities in serious games as opposed to traditional textbook based materials (Sørensen and Meyer, 2007). Another study demonstrated that among adults, those who have a keen interest in the subject matter of the serious game felt they had acquired a functional ability in the language in the scope of the game. Those who did not consider the game significant or of particular interest to them, however, failed to further their L2 knowledge (Johnson, 2007). Considering, however, that L2 learners are able to choose which subject matter they wish to read about, serious games could prove to be a valuable addition to bilingual eBooks as a tool to provide L2 learners an opportunity of output and self-assessment.

### 1.3 Related work

Second language learning using eBooks has been the target of several applications as of recently. Upon reviewing these existing applications, it is clear that a categorical improvement is possible.

FarkasTranslations<sup>1</sup> and InLector<sup>2</sup> offer simple aligned books. Their method matches the corresponding paragraphs of each language. Whereas FarkasTranslations presents a two-column table which establishes this equivalence, InLector displays only the text of one given language at the time. Tapping a paragraph in these books will redirect the user to the corresponding paragraph in the second language, and vice versa. Both these applications lack audio support, a comprehensive user interface or the opportunity for user assessment.

Doppeltext<sup>3</sup> also provides aligned books. These, however, are enhanced by an interactive user interface. By default, only the text in the second language is shown. Upon pressing any given sentence, its translation is neatly displayed in an inline text box.

Bilinguis<sup>4</sup> offers parallel bilingual eBooks with a simple user interface. Some, complete with audio support. The user is free to navigate the audio by use of four buttons. These buttons allow to move forward/rewind the audio by 10/30 seconds. It does not allow for precise control, as the user is unaware of how far ahead or back they must forward/rewind the audio to match the text they are reading. There is also a distinctive lack of visual support, as there is no sentence or even paragraph highlighting during audio playback.

In summary, the work presented in this dissertation aspires to build upon the aforementioned applications in order to provide a richer learning experience and a complete user interface to the end user.

---

<sup>1</sup>[http://farkastranslations.com/bilingual\\_books.php](http://farkastranslations.com/bilingual_books.php)

<sup>2</sup><https://inlector.wordpress.com/>

<sup>3</sup><https://www.doppeltext.com/en/>

<sup>4</sup><http://bilinguis.com/>

## 1.4 Objectives

The present work aims to deliver a computer-based solution to the autonomous creation of bilingual eBooks. Specifically, the solution will create and enhance eBooks by means of an intuitive user interface, complete with word definitions, and a set of serious games.

This solution comprised of the development of a Python script which took the raw inputs of text and audio and merged them into a single structured eBook document, complete with dynamic features that allow the user an improved control over the audio playback. These features are not included in the target system, Apple's iPad, by default, and thus required development. The script also retrieves, from an open-source API, dictionary definitions for every word in the eBook and includes it in the final document.

Additionally, the development of a Javascript application was required to implement the serious games included in the eBook. This solution was required to use no external resources as the chosen eBook format specification does not allow for online content.

The development of the system was supported by feedback from a test group during a user study who reviewed the system and indicated which features were of their preference and offered further development suggestions.

Finally, this work aspires to lay a foundation for future work, namely in the field of read aloud, which could be a topic of significance in CALL. Despite the numerous research done involving TTS in addition to the benefits of reading aloud in L2 learning, its benefits have not been studied in CALL utilizing bilingual books as a basis. Such a system could enjoy the benefits of dual language books in conjunction to those of TTS.

## 1.5 Contributions

An article regarding the study developed by this thesis has been submitted to INForum 2017. INForum<sup>5</sup> is an annual Portuguese event which aims to be forum of choice for the dissemination, discussion and recognition of scientific works. It promotes the growth of computer science in Portugal by encouraging the participation of a vast array of subjects.

## 1.6 Organization of the thesis

This thesis is arranged by four main chapters. Each chapter is largely independent from one another, save for a few occasional references to previous chapters, to avoid excessive redundancy.

**Summary of Chapter 2** - This chapter provides an overview on the theoretical background of this thesis. It covers theories for language learning, such as computer-assisted language learning and serious games, text and audio processing algorithms, as found in parallel text alignment and forced alignment and speed recognition systems, and finally the EPUB 3.0.1 specification.

---

<sup>5</sup><http://inforum.org.pt/>

**Summary of Chapter 3** - This chapter regards specifications, design and implementation of the software system created. The specification section provides an overview of what the software system is required to do, and what capabilities are necessary. The design section describes a top-level analysis of how the software system meets these requirements, and discusses the design constraints on the software solution.

**Summary of Chapter 4** - This chapter describes the system at a finer detail, as it goes over the realization of the concepts and ideas developed in chapter 3.

**Summary of Chapter 5** - This chapter elaborates on the hypothesis of the user study and discusses the results of its execution, complete with a demographic analysis and informal feedback.



# Chapter 2

## Background

This chapter provides an overview on the theoretical background of this thesis. Bilingual eBooks are a topic that demand the study of an assortment of fields. Firstly, we will review computer-assisted language learning and how it applies to this case study. Secondly, the subject of text alignment arises, which pertains to the idea that the text of either language of the bilingual eBook must remain as parallel as possible throughout the entire document. Then comes forced alignment technologies which provide a means to synchronize the audio to match the text presently being read. Finally, the current EPUB standard must be understood as it is the preferable format with which to create and manipulate eBooks.

### 2.1 Theories for language learning

#### 2.1.1 Computer-assisted language learning

Computer-assisted language learning (CALL) is an approach to language learning and teaching which includes computer technologies as a means to promote educational learning through a substantial interactive element.

Study of CALL can be traced back to the 1960s, when it first was introduced on university mainframe computers by the PLATO project at the University of Illinois in 1960. CALL continued to be confined mainly to universities up until the late 1970s, before the arrival of personal computers. With these, a wider interest surged in the development of programs by the industry, which prompted a surge of new CALL programs and study articles in the 1980s.

Since then, innumerable CALL programs have been developed, ranging from simple to robust systems. For instance, some older programs consisted of drill-and-practice materials, in which the computer presented a series of challenges for the learner to attempt to solve. The computer would then analyse the students' input and evaluate their performance, including branching to word definitions and structural rules screens should the learner fail to correctly answer a particular question. In the early stages of CALL this process could only be accomplished through text, but since then the use of TTS and Automatic speech recognition (ASR) have allowed for more possibilities. In virtue of this, students are currently able to find systems in which they can not only practice their use of English, but also learn and

practice pronunciation with CALL (Davies, 2002).

L2 learning is usually segregated into individual language skills, and this is reflected in modern L2 learning research segregation into these different skills. Nonetheless, CALL can be used to help students attain fluency, complexity and accuracy in oral skills, literacy and formulaic expressions (Hubbard, 2009).

Owing to audio features, CALL programs provide their users with a profusion of different ways in which they can practice their listening skills. For instance, videos supporting L1 or L2 subtitles or explanatory notes, along with an option to quickly and easily control its playback can improve both immediate comprehension and acquisition (Borrás and Lafayette, 1994).

With the vast accessibility to foreign media, a wide range of L2 audio and video to listen to can be readily found, which often unintentionally creates great CALL opportunities. Furthermore, a wealth of dedicated CALL web content can be easily discovered to be used by free or for a price. Any of these options allow the L2 learner to gain a deeper understanding of pronunciation.

Also, thanks to ASR systems, L2 learners are no longer restricted to speech practice with a partner, but are able to talk to a computer integrating ASR in a CALL program to either simply have it evaluate their performance or to have a short dialog, with more advanced systems. These systems can then evaluate the L2 learner on their pronunciation through a Goodness of Pronunciation (GOP) algorithm, as the one proposed by Witt (Witt and Young, 2000; Kanter et al., 2009). Furthermore, using the internet it is entirely possible to find practice partners to engage in an audio discussion, using VoIP applications.

For writing skills, L2 learners can use CALL programs to enrich their vocabulary and benefit from the now widespread spell and grammar checking systems. Such systems enable L2 learners by quickly giving them feedback which develops the L2 writing accuracy of the learner.

### **2.1.2 Serious games**

Gamification is the concept of applying game design mechanics, techniques and game principles to improve user engagement, productivity and motivation. Its efficiency is a topic of interest, as while gamification has been tied to producing positive results, some confounding factors exist, namely the role of the context being gamified and the nature of the users (Hamari et al., 2014).

Serious games can be defined as a method of gamification which brings entertainment and education together, but does not have fun or enjoyment as a primary purpose. In summary, they focus instead on a carefully thought-out topic or material of interest around which the game is built. They can also enrich previous existing materials such as training films and books, by allowing players to not only learn, but to also apply and test their new knowledge (Michael and Chen, 2006).

Games, be it computer games, board games or social games, require the players to learn, even if just the rules of the game. Upon mastering the rules, the player then has the creative freedom of applying and navigating the rules. For instance, when developing a new game, the game designer must take into consideration how to convince the player to learn how to play the game and use good methods and principles to ensure that it is a fun experience (Michael and Chen, 2006).



There is very little available research on the subject of efficacy of most gamification methods and techniques, despite the number of proposed methods. Most sources agree this process should revolve around identifying a meaningful frame and implementing a points system (Mekler et al., 2013) and (Huang and Soman, 2013).

Mekler et al. assessed the efficacy of points/currency and a meaningful frame within the scope of gamification with positive results. The study gauged the efficacy through two indicators: participant performance and intrinsic motivation. It revealed that using either method on its own or combined achieves equal levels of learner motivation. However, users who achieved better overall results for both performance and motivation were those using systems which implemented both approaches.

The points system also distinguished itself as not only a scoring tool, but also as a form of immediate feedback. No clear indicators were found regarding the reason as to why participants enjoyed the feature, but the authors hypothesized it may have been due to points establishing a correlation between user effort and performance. Such a feedback may have been the motivating agent that drove participants to make a greater effort towards achieving goals.

According to Huang and Soman, gamification should be taken as a five-step process as follows:

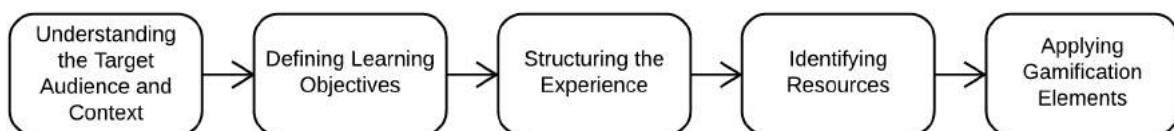


Figure 2.1: The gamification process.

- **Understanding the Target Audience and Context** is the first step of the gamification process. In order to create successful serious games one must clearly identify who the target audience will be, as different audiences will naturally have different difficulties and targets. A successful serious game will then be one that fully empowers the student to derive the richest possible learning experience.
- **Defining Learning Objectives** represents, as the name implies, the stage during which the developer unequivocally defines the end goal for students at the end of the learning program.
- **Structuring the Experience** is defined as the milestones used to sequence learning. With these one can quantify what students must achieve by the end of each stage. Students themselves find these milestones agreeable as it makes the end goal more achievable and progress towards it measurable.
- **Identifying Resources** is done once the experience is structured, and entails ascertaining which stages can be gamified, and how. The resources may be interpreted as a collection of tracking mechanisms, with which the instructor or program may evaluate how a student is faring and possibly purchase their tendencies and common pitfalls; currency or point reward mechanisms, which the student may use in some particular way, used purely as a motivational agent; and, finally, feedback systems. This last mechanism is one of the reasons games are so appealing, as students are

promptly given feedback on their performance, and so have the chance to make another attempt with their newly found knowledge.

- **Applying Gamification Elements** is the moment when the game-like-elements are finally added to the subject at hand. If one were to adhere to the definition set by Huang and Soman, game mechanics may then be classified in two ways.

Firstly, we have self-elements. These elements can range from points to achievement badges, and are, in essence, any elements that promote a student's desire to develop oneself, and with which set markers that may drive them further. Such elements can bring upon the student a sense of self-achievement.

Then come the social-elements. These, as opposed to self-elements, are motivated by competition or cooperation. Such an element, as for example a leaderboard, places the students in a community with other students, and make their progress public to others.

Self-Elements	Social Elements
Points	Leaderboards
Levels	Interactive Cooperation
Achievements	
Time Restrictions	

Figure 2.2: Example gamification elements.

Moreover, a plethora of gamification methods, resources, embeddable games and case studies are available at the Gamification Wiki<sup>1</sup>, a free online resource for those interested in the subject of gamification.

However, despite all stated above, when developing either a serious game or a gamified learning program, it is important to recognize that good performance, such as good grade results, cannot be gamified. Instead, the gamification mechanics drive students to perform well. Even if during the second stage of the gamification process, *Defining Learning Objectives*, the objective determined is a specific learning goal, the developer must use gamify the process in such a way that it influences the student's behavior in regards to learning. This does not mean, however, that gamification aims to reshape student behavior perpetually, since if the student finds himself in a new context, in which gamification is not existent, the extrinsic reward is gone and the new habits will not be preserved.

In conclusion, in traditional learning environments, a student may feel unmotivated due to various reasons. This lack of motivation derives from negative emotions usually encountered in traditional forms of education, such as embarrassment and humiliation. Gamification serves as a means of lessening these emotions, and, when adequately deployed, it delivers a powerful positive and captivating context within which students feel motivated to learn and devote more time towards achieving their goals. This

---

<sup>1</sup><https://badgeville.com/wiki/>

behavior is stimulated by the use of gamification elements which provide immediate feedback and a sense of gratification.

## **2.2 Text and audio processing algorithms**

### **2.2.1 Parallel text alignment**

When dealing with large parallel texts, such as bilingual eBooks, the task of alignment can become very difficult. When a text is translated, particularly when considering how different two languages can be in terms of structure, grammar and syntax, some sentences may be split, merged, reordered, deleted or inserted by the translator. This increases the difficulty of finding a simple solution that will align any text with its translation.

Alignment may be executed on different levels. Some texts may benefit from paragraph alignment, while others from sentence alignment or even word alignment. Regardless, alignment methods are often categorized as either statistic or lexicon-based. Statistic approaches estimate alignment probabilities using a maximum likelihood whereas lexical level approaches use specific associative measures between texts such as mutual information in sentences and linguistic dictionaries (Haruno and Yamazaki, 1996).

One such statistic approach is that of Gale and Church (Gale and Church, 1993), who described a method of aligning sentences units in a bilingual corpus. The motivation for their model came from the observation that longer regions of text tend to have long translations, and shorter regions have shorter translations, finding that the correlation between the length of a paragraph, measured in number of characters, between the original text and its translation was exceedingly high.

There are instances, however, where this analysis can prove difficult. Some languages can be so far apart that a sizable number of sentences of highly different lengths, additions and deletions can occur from one language to the other. In these cases, this algorithm cannot be used effectively to align very large bodies of text. As such, it depends heavily on the paragraph delimiters, using them as synchronization aids that help the algorithm correct itself should it experience some deviation.

Gale and Church's algorithm is a two-step process. In the first step paragraphs are aligned, and only then are the sentences within each paragraph aligned. The paragraph alignment in this algorithm is fairly trivial, as it expects the commonly used paragraph boundaries that are usually clearly marked. As for the sentence alignment, the algorithm makes use of the fact that related sentences of both languages usually have a proportional length. Each pair of sentences is assigned a probabilistic score based on their character length ratio and variance. These scores are then processed by a dynamic programming framework which outputs the alignment configuration of maximum likelihood for all the sentences in the paragraph.

The distance is measured based on a probabilistic model so as to be able to combine this information later on in a consistent manner. The measurement is based on the assumption that each character in one language corresponds to a random number of characters in the other. Once all the data is gathered, the resulting model will be specified by the mean, which is the expected ratio between characters of

both languages, and its variance. We can represent the model by  $-\log \text{Prob}(\text{match}|\delta)$ , where  $\delta$  is  $(l_2 - l_1 c)/\sqrt{l_1 s^2}$ , with  $c$  representing the distance model's mean, and  $s^2$  its variance, and  $l_1$  and  $l_2$  the lengths of the two portions of text under analysis.

The dynamic programming algorithm uses the distance function calculated beforehand and processes the data recursively, in order to find the most likely sequence of aligned sentences.

Lexicon-based alignment approaches use associative measures rather than pure statistical measures. One example of such a method is the Clue alignment approach proposed by Jörg Tiedemann (Tiedemann, 2003). Most basic lexical level methods use the Dice-coefficient, which is a statistic used for comparing the similarity of two elements, to keep co-occurrence scores for each word pair, and the total independent occurrences for those words in each of their respective texts. Then it calculates the probabilities of alignment and selects the sentence combinations of maximum likelihood. Tiedemann's approach however, uses a segment to segment matrix to represent the possible alignments instead, and only then assigns a score to each available word alignment.

Tiedemann defines a word alignment clue  $C_i(s, t)$  as a probability which indicates an association between two lexical items  $s$  and  $t$ , where a lexical item is a set of words and its associated features. A clue can be further defined as a weighted association  $A$  between  $s$  and  $t$ :  $C_i(s, t) = P(a_i) = w_i A_i(s, t)$ , where  $w_i$  is used to normalize the association score  $A_i$ .

Alignment clues can be estimated from measurements of given empirical data, through, for example, co-occurrence and string similarity, shown below, respectively:

$$A_{Dice}(s, t) = \frac{2P(s, t)}{P(s) + P(t)} \quad A_{LC SR}(s, t) = LC SR(s, t)$$

Alignment training may also be used to estimate clues, as shown by the following equation, where  $f_s$  and  $f_t$  are sets of features.

$$C_j(s, t) = w_j * P(f_t|f_s) \approx w_j \frac{\text{freq}(f_s, f_t)}{\text{freq}(f_s)}$$

The clues are combined to find interlingual links, which can be found through defining the overall clue:

$$C_{all}(s, t) = P(a_{all}) = P(a_1 \cup a_2 \cup \dots \cup a_n)$$

It is of note that clues are not mutually exclusive. Furthermore, it is assumed that clues are independent, which must be taken into consideration during analysis of clue patterns.

Finally, to recover the word alignment, Tiedemann uses dynamic procedure. Firstly, it finds the link in the clue matrix with the highest relation value, and sets its value to zero. Then the link is compared to the previously accepted links, and should there be an overlap, it is ignored. Otherwise the link is added to the accepted link cluster. This process is repeated until no more nonzero value links are found, or when the best link is below a given threshold.

A combination of both statistic and lexicon-based alignment methods is the approach of most modern

parallel text aligners, such as the one used in the present work, Hunalign<sup>2</sup>.

Hunalign is an open-source parallel text aligner that aligns text on the sentence level. It uses both a dictionary and the Gale and Church algorithm to produce the alignment. Should no dictionary be available, Hunalign creates one by using the Gale and Church algorithm by itself, and then extract data from the alignment to create its own dictionary. After this process, it then executes a new alignment, this time using both the Gale and Church algorithm and the automatically formed dictionary.

## 2.2.2 Forced alignment and speech recognition systems

Forced alignment systems attempt to match the words of a given input text to those of an input audio file. The input text is mapped into a phone sequence by using a pronunciation dictionary, and phone boundaries are determined by algorithms such as the Viterbi algorithm. Forced alignment is very commonly used in large parallel corpora, video subtitling and live subtitling.

When initially doing research on forced alignment systems, the most frequently used approach was to create a Hidden Markov model (HMM) phonetic recognizer. HMMs are parametric and statistical models, in which the system being simulated is taken as a Markov process with hidden states. Although HMMs are no longer state of the art at the time of writing, these systems are still an acceptable support for the work developed.

Current forced alignment systems will usually be composed of two parts, a training stage and an alignment stage. During training, the system uses a given digitized speech to produce acoustic models of each phoneme, called monophone models. These models may take the phonemes context into account, using the preceding and succeeding phonemes. These models are called triphones and allow analysis of relaxed pronunciation. When attempting alignment, the system aligns the new speech with its acoustic models through the Viterbi algorithm to produce time stamps for the given audio file. The forced alignment process is identical to that of ASR, except that a phonetic transcript is known (Brognaux et al., 2012). Thus, in ASR the Viterbi algorithm is used to find the most likely path through the phoneme HMM and from that the system then generates the resulting word string.

Several ASR system variants using HMM, such as the deep neural network HMMs with many hidden layers or Gaussian mixture models are used on state-of-the-art systems with a plethora of recent studies attempting to compare their efficacy in relation to one another when under different conditions, such as context-dependent pre-training (Dahl et al., 2012; Hinton et al., 2012). In the case of Dahl et al., forced alignment is used during speech recognition through the Viterbi approach in an attempt to increase the confidence of the solution. Forced alignment is also used in speech recognition in a wealth of research, such as speech recognition with noise (Das et al., 2010), lyrics-to-audio alignment (Fujihara and Goto, 2012) and others.

A flowchart describing a typical HMM-based ASR system using the Viterbi algorithm may be seen in figure 2.3.

During training, context dependent phoneme HMMs are developed using a speech database from which mel-cepstral coefficients and fundamental frequencies (F0) are extracted at each analysis frame.

---

<sup>2</sup><http://mokk.bme.hu/resources/hunalign/>

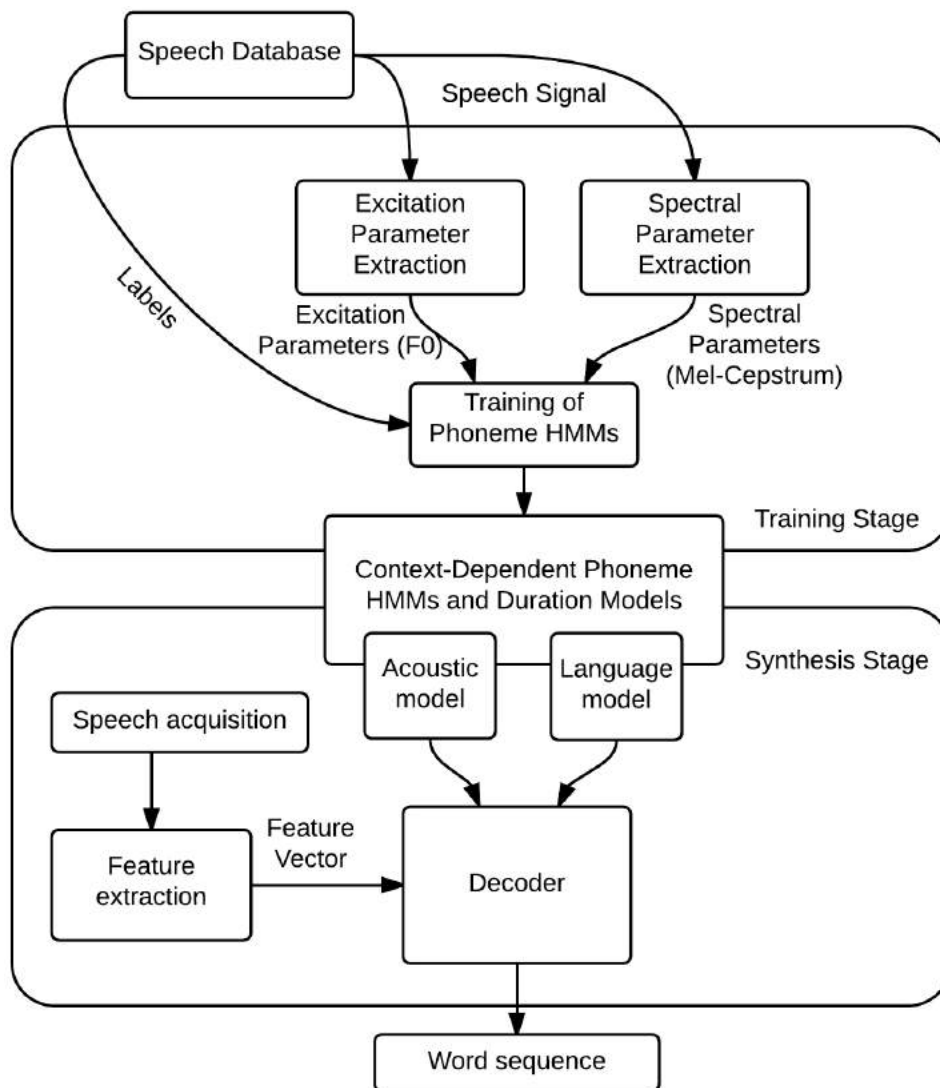


Figure 2.3: Overview of a typical HMM-based ASR system.

To model F0 properly, HTS resorts to multi-space probability distributions (MSD), since F0 patterns cannot be modeled by conventional discrete or continuous HMMs given that the values of F0 are not defined in unvoiced regions. Using MSD-HMM the system can model variations in spectrum and F0 by taking phonetic, prosodic and linguistic contexts into account, such as the number of morae and the type of accent in a sentence and in the preceding, current and succeeding breath groups, the position of the breath group in a sentence and the surrounding phonemes relative to each phoneme in a sentence.

## 2.3 EPUB 3.0.1

EPUB<sup>34</sup> is a widely adopted format for digital books, eBooks, due to its potential to comprise complex layouts and interactivity in the eBook. This is achieved by the use of the latest HTML5 standard by EPUB 3.0.1, which allows EPUB publications to contain video, audio and interactivity through scripting, much

<sup>3</sup><http://www.idpf.org/epub/301/spec/epub-overview.html>

<sup>4</sup><http://www.idpf.org/epub/301/spec/epub-contentdocs.html>

like modern web pages. EPUB reading systems paginate the content dynamically, adapting it to the display, rather than having the reader zoom in or out a fixed format page, as is common with other applications such as PDF. This behavior is not forced in any way, however, allowing for specific content that would benefit from a zooming and panning functionality to be manipulated in such a way.

Being comparable to a web page EPUB is, consequently, a simple ZIP archive file that contains the HTML files, CSS style sheets, metadata and any other assets necessary for EPUB reading applications or devices. As any other ZIP files, files with the extension .epub may be unzipped by most archive software. Doing so will reveal a set of files akin to that of figure 2.4.



Figure 2.4: Files typically found in EPUB archives.

Inside the first level of the EPUB archive one finds two folders and a typeless file. That file, `mimetype`, simply provides a way to determine that the zipped content is an EPUB publication, typically containing the line "application/epub+zip". One of the folders, in this case in particular the OEBPS (Open eBook Publication Structure) folder, contains the content of the eBook, with all related HTML, CSS and assets. Finally, the META-INF folder, which is so universally named in all EPUB archives, includes information regarding the publication.

Inside the META-INF folder is the pivotal `container.xml` file, which directs EPUB reading systems to the root files of the EPUB publication. The contents of an example file can be seen in figure 2.5.

```
<?xml version="1.0" encoding="UTF-8"?>
<container xmlns="urn:oasis:names:tc:opendocument:xmlns:container" version="1.0">
  <rootfiles>
    <rootfile full-path="EPUB/package.opf" media-type="application/oebps-package+xml"/>
  </rootfiles>
</container>
```

Figure 2.5: Example content of a `container.xml` file.

As can be seen in the `rootfile` tag, the path to the root files is fully specified, as is its media type. Specifically it provides the path to an .opf file, usually referred to as the Package Document, which is a unique file in the publication. It specifies all documents, images and other content that make up the publication. The Package Document contains its information in five different groupings:<sup>56</sup>

- **Package Name** - This element is the root container of the publication, and contains the four elements described below. It also specifies the EPUB specification version to which the EPUB reader system must conform and a unique identifier for the publication as a whole.

<sup>5</sup><http://www.idpf.org/epub/301/spec/epub-publications.html>

<sup>6</sup>[http://www.idpf.org/epub/20/spec/OPF\\_2.0.1\\_draft.htm](http://www.idpf.org/epub/20/spec/OPF_2.0.1_draft.htm)

- **Metadata** - Provides mechanisms for including and referencing metadata applicable to the publication or to their renditions in a EPUB reading systems. It must include at least the title, identifier and language of the publication, and a modified property that states when the publication was last edited. Furthermore, to include the metadata a child element must be used, "meta".
- **Manifest** - Lists all files that form the publication. As with metadata, to produce the manifest the child element "item" is required. It specifies a unique ID for its object, its location in the root EPUB file directory and its media type.
- **Spine** - Defines an ordered sequence of ID references from the manifest resources, arranging the documents in their default linear reading order. This is done using the child element "itemref". Each element that the developer chooses to be read linearly in their publication from the manifest is referenced this way, and their order will be that of the "itemref" element order. Fallback elements are also defined in this way so that deprecated EPUB reading systems may still use some features of the publication.
- **Guide** - A list of references to fundamental parts of the publication, such as a table of contents, glossary or bibliography.

The above elements can be quickly observed in the example Package Document from figure 2.6.

```
<?xml version="1.0" encoding="utf-8"?>
<package version="3.0" xml:lang="en" unique-identifier="uid" prefix="cc: http://creativecommons.org/licenses/by-sa/4.0/">
  <metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
    <dc:title id="title">A First Course in Linear Algebra</dc:title>
    <dc:identifier id="uid">code.google.com.epub-samples.linear-algebra</dc:identifier>
    <dc:language>en</dc:language>
  </metadata>
  <manifest>
    <item id="css0" href="css/epub.css" media-type="text/css"/>
    <item id="css7" href="css/commonltr.css" media-type="text/css"/>
    <item id="c1" media-type="application/xhtml+xml" href="xhtml/titlepage.xhtml"/>
    <item id="c2" media-type="application/xhtml+xml" href="xhtml/copyright.xhtml"/>
  </manifest>
  <spine>
    <itemref idref="c1"/>
    <itemref idref="c2"/>
  </spine>
</package>
```

Figure 2.6: Example of a Publication Document file

This example Package Document demonstrates the definition of a simple EPUB publication with two pages, their ordering, and their respective CSS code. Despite the ordering imposed by the Package Document, it is possible to bypass it when such a demand arises from the publication, such as it would from a cookbook or a journal. Since EPUB publications are built upon HTML, it is entirely possible to use hypertext to have the user navigate the publication in a non linear way.

In addition to CSS, EPUB enjoys all the benefits of HTML5, including scripting. However, one should note that scripting support is entirely optional for EPUB reading systems, and should it be present it could be disabled for security reasons. Nevertheless, EPUB incorporates several mechanisms that facilitate



development and, in some cases, remove the need for scripting. One such mechanism is the "trigger element", which enables image or textual entities to function as controls for audio and video playback.

Moreover, EPUB3 is outfitted with TTS facilities that facilitate some aspects of speech synthesis, including pronunciation and prosody. The inclusion of generic pronunciation lexicons using the W3C pronunciation lexicon specification (PLS)<sup>78</sup> enables developers to provide pronunciation rules that apply to the entire publication. Furthermore, the additional speech synthesis markup language (Speech synthesis markup language (SSML))<sup>9</sup> phonemes functionality enables developers to fine tune pronunciation control over the default pronunciation rules or the referenced PLS.

Implementing PLS in EPUB is quite simple. Firstly, one requires a PLS file, which defines pronunciation of given words using the International Phonetic Alphabet (International phonetic alphabet (IPA)), as shown in figure 2.7.

```
<?xml version="1.0" encoding="UTF-8"?>
<lexicon version="1.0"
  xmlns="http://www.w3.org/2005/01/pronunciation-lexicon"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/2005/01/pronunciation-lexicon
    http://www.w3.org/TR/2007/CR-pronunciation-lexicon-20070814"
  alphabet="ipa" xml:lang="en-US">
  <!-- Defining pronunciation for one word -->
  <lexeme>
    <grapheme>eBook</grapheme>
    <phoneme>'i:buk</phoneme>
  </lexeme>
  <!-- Defining multiple pronunciations for
    the same orthography -->
  <lexeme>
    <grapheme>address</grapheme>
    <phoneme>ə'dres</phoneme>
    <phoneme>'ædres</phoneme>
  </lexeme>
  <!-- Homophones should be represented as
    different lexemes -->
  <lexeme>
    <grapheme>sent</grapheme>
    <phoneme>ssnt</phoneme>
  </lexeme>
  <lexeme>
    <grapheme>scent</grapheme>
    <phoneme>ssnt</phoneme>
  </lexeme>
  <!-- Multiple orthographies -->
  <lexeme>
    <grapheme>defence</grapheme>
    <grapheme>defense</grapheme>
    <phoneme>dI'fEns</phoneme>
  </lexeme>
</lexicon>
```

Figure 2.7: Example of a PLS lexicon document.

After creating or editing such a file to suit their needs, the developer then must reference the Pronunciation lexicon specification (PLS) document in the manifest of the Package Document, using the "item" element, and define its media-type as "application/pls+xml". Lastly, the PLS document must also

<sup>78</sup><http://www.w3.org/TR/2008/REC-pronunciation-lexicon-20081014/>

<sup>8</sup><http://www.idpf.org/accessibility/guidelines/content/tts/pls.php>

<sup>9</sup><http://www.w3.org/TR/speech-synthesis/>

be linked to each content document that requires it, as can be observed in figure 2.8.

```
<html>
  <head>
    <link
      rel="pronunciation"
      href="lex/example.pls"
      type="application/pls+xml"
      hreflang="en" />
  </head>
</html>
```

Figure 2.8: Linking PLS lexicons to a content document.

The SSML provides control over a larger spectrum of synthetic speech generation. A TTS system that supports SSML uses it to define the form of the output speech. As seen in section ??, speech synthesis is done in five phases. SSML allows the developer to have control over each step and ultimately the final voice output.

Firstly, in phrase segmentation, the "s" (sentence) and "p" (paragraph) elements are used as an explicit indication of the documents' structure. Their use is optional, however, and should text occur without an enclosing p or s element, the speech synthesizer must attempt to determine the texts structure on its own.

As for text normalization, "sub" elements allow the developer to indicate an alternate text for pronunciation in its alias attribute, which supersedes the text of the publication.

In the text to phoneme conversion, SSML can control pronunciation of specific elements of text, be it one word or word sequence, through the "phoneme" element, akin to PLS. The "lexicon" element may also be used to reference external pronunciation lexicon documents.

"Emphasis", "break" and "prosody" elements can be used to manipulate prosody analysis. "Emphasis" and "break" provide the functions given by their names, the first having a level adjustment to the emphasis the system must attempt to render, and the second having the option be adjusted with either a fixed timer attribute or a strength attribute, similar to the level attribute of the "emphasis" element. "Prosody" is an element that allows control over pitch, speaking rate and volume of the synthesized speech. Attributes available in "prosody" are all optional, and are described below.

- **pitch** - allows control over the F0 of the synthesized voice.
- **contour** - sets the actual pitch contour.
- **range** - allows the developer to increase or decrease the dynamic range of the synthesized speech.
- **rate** - creates a relative change in speaking rate in relation to the default rate.
- **duration** - can be used to force the synthesizer to read the text in the scope of the element in a specific time, given in seconds or milliseconds.
- **volume** - controls the sound volume.

Finally, in waveform production, the developer has access to both the "voice" and "audio" elements. The first allows the developer to define a particular voice or specific voice characteristics, such as gender, age, name and variants. The latter element can be used to insert recorded audio into the output stream and TTS is not performed on the text in its scope.

A very small example of an SSML file is represented in figure 2.9.

```
<?xml version="1.0"?>
< speak version="1.0" xmlns="http://www.w3.org/2001/10/synthesis"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2001/10/synthesis
      http://www.w3.org/TR/speech-synthesis/synthesis:
      xml:lang="en-US">
  <p>
    <voice gender="female">
      <s>This is a
        <emphasis level="strong"> great </emphasis>
        <phoneme alphabet="ipa" ph="ˈiːbʊk">eBook</phoneme>
      .</s>
    </voice>
    <voice gender="male">
      <s>Example
        <sub alias="Speech synthesis markup language ">SSML</sub>
        file.</s>
      </voice>
    </p>
  </speak>
```

Figure 2.9: Example of SSML usage.

In addition to TTS, using synchronized multimedia integration language (Synchronized multimedia integration language (SMIL)) one can synchronize the text of a publication with a corresponding recorded audio file. To do this, in the HTML file in which the text is documented, one must declare a "span" element, and give it a unique ID number. Then, in the SMIL file, this ID can be used to indicate that the position in the text where the "span" element is found is the starting moment of a given audio clip in the EPUB publication, and the closing of that "span" element is where it ends. One is able to assign a starting and ending point for the audio file, thus dividing a whole audio file into several sentences, and assigning a unique "span" ID to each sentence in the text file. Doing so will open the opportunity to skip entire portions of the audio file, and repeat select sentences at will without having to listen to the whole file. This could also be done for each individual word, or any other way the developer sees fit.

A small example of how such a functionality is implemented and how the IDs are referenced specifically is given in figures 2.10 and 2.11.

```

<?xml version="1.0" encoding="utf-8"?>
<html xmlns:epub="http://www.idpf.org/2007/ops" xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:
  <head>
    <meta name="viewport" content="width=1200px, height=900px"/>
    <meta http-equiv="default-style" content="text/html; charset=utf-8"/>
    <link rel="stylesheet" href="../css/epub.css" type="text/css"/>
    <link rel="stylesheet" href="../css/commonltr.css" type="text/css"/>
    <title>Example</title>
  </head>

  <body class="example">
    <span id="a0"/>    <!-- First audio tag -->
    <span style="position:absolute; left:100px; top:100px; font-size:14px;">A </span>
    <span style="position:absolute; left:140px; top:100px; font-size:14px;">Short </span>
  </span>
    <span id="a1"/>    <!-- Second audio tag -->
    <span style="position:absolute; left:260px; top:100px; font-size:14px;">Example</span>
  </span>
</body>
</html>

```

Figure 2.10: Example of a .xhtml text file.

```

<?xml version="1.0" encoding="UTF-8"?>
<smil xmlns="http://www.w3.org/ns/SMIL" xmlns:epub="http://www.idpf.org/2007/ops"
  <body>
    <par id="a0"><text src="titlepage.xhtml#a0"/>
    <audio clipBegin="0.000" clipEnd="2.000" src="../Audio/audio_file.m4a"/>
  </par>
    <par id="a1"><text src="titlepage.xhtml#a1"/>
    <audio clipBegin="2.000" clipEnd="5.000" src="../Audio/audio_file.m4a"/>
  </par>
  </body>
</smil>

```

Figure 2.11: Example of a .smil file.

## Chapter 3

# Specification and Design

This chapter regards specifications, design and implementation of the software system created. Initially, the specification section will provide an overview of what the software system is required to do, and what capabilities are necessary. Then, the design section describes a top-level analysis of how the software system meets these requirements, and discusses the design constraints on the software solution.

### 3.1 Specification

This section regards the demands of the project, around which the software design is built. The objective of the software system is twofold. Firstly, it must create the eBook and its user interface, complete with a dictionary mode. The user interface must allow control over both playback and speech rate. Secondly, it must include serious games in the eBook, which are to be understood as gamified self-assessment tools, repeatable by nature, regardless of the content of the eBook.

It is intended that the creation of the EPUB formatted eBooks be fully autonomous, requiring no external resources other than the text given in two distinct languages, its corresponding audio file and parallel text alignment and forced alignment tables. Due to the use of the EPUB format, any resource or data required by the eBook must be self-contained, as it is not possible for the eBook to fetch any online resource. Static resources available online required by the eBook must be fetched by the eBook generator script.

Regarding the specifications of the parallel book section, the eBook must provide the user with a sentence highlighter which establishes a correlation between the sentence being read by the narrator and the corresponding text segment in the second language. The user interface must allow the reader to seamlessly control audio playback, through commonplace play and pause functionalities and pace and flow control processes as well.

The dictionary mode is expected to be able to return the definition of every word in the eBook that has an entry in the given source. In the case of this study, the open source dictionary Wiktionary is to be used. A few select words are to be underlined at all times, meaning their definitions may be accessed

without the use of dictionary mode. These correspond to the words deemed most important and relevant for the readers learning process. When generating the eBook, the script must allow for a word list to be provided to be used as the important word list. Should this list not be provided, a default method must determine the important words through a basic algorithm that uses word length and recurrence as indicators.

Regarding the serious games, there are a few restrictions imposed on their design. As mentioned beforehand all content in the eBook must be self-contained. In this manner, it is assumed that the games revolve around the text in the eBook, requiring the user to translate words into the second language given their context in the original transcript. The games are also expected to utilize gamification elements to bolster the users motivation and engagement levels.

Finally, the resulting eBooks are expected to be fully functional within the iBooks application in any iOS device, with the iPad being their primary target, and as such correct operation on other systems is not to be guaranteed. iOS devices with smaller screens, such as smartphones, are, however, not supported either. The text would be too small to read comfortably due to the specification that both languages must be displayed in parallel at all times. Future development of a zooming functionality that acts equally on both texts, i.e., dividing the page in half and mirroring the zooming and panning on both sides, could mitigate this problem.

## 3.2 Design

### 3.2.1 Parallel books interface

The software system may be logically divided into three main components, which work independently from each other, but are equally vital to its functionality:

- **EPUB generator** - Responsible for merging the input texts into single files, copying independent resources, adding definitions and serious games to the EPUB package.
- **Dictionary manager** - In charge of storing, fetching and parsing dictionary data from Wiktionary<sup>1</sup>.
- **Serious games** - This component is responsible for the real-time generation of gamified self-assessment tools using the eBook's pages as its only resource.

The eBook is to allow the user to play the audio starting from any given sentence in the eBook. To do this, the user should tap the text corresponding to the sentence he wishes to start from. In addition, the eBook is to provide means that help better comprehend the audio. It is the responsibility of the EPUB generator script to include this functionality in all eBooks. On running, this first component expects a number of eBook independent resources, namely icons, images, text fonts, scripts and optionally a list of important words to underline in the eBook that is going to be processed. The images and icons are needed to make a small graphical user interface, which is required to be non intrusive, so as to not disrupt reading. Tied to these is a script which is the backbone of all interactivity in the eBook. It handles

---

<sup>1</sup><https://www.wiktionary.org>

all touch requests and button presses. On the iPad, it also ensures the iBook's default UI is only brought up on demand, overriding all default behavior.

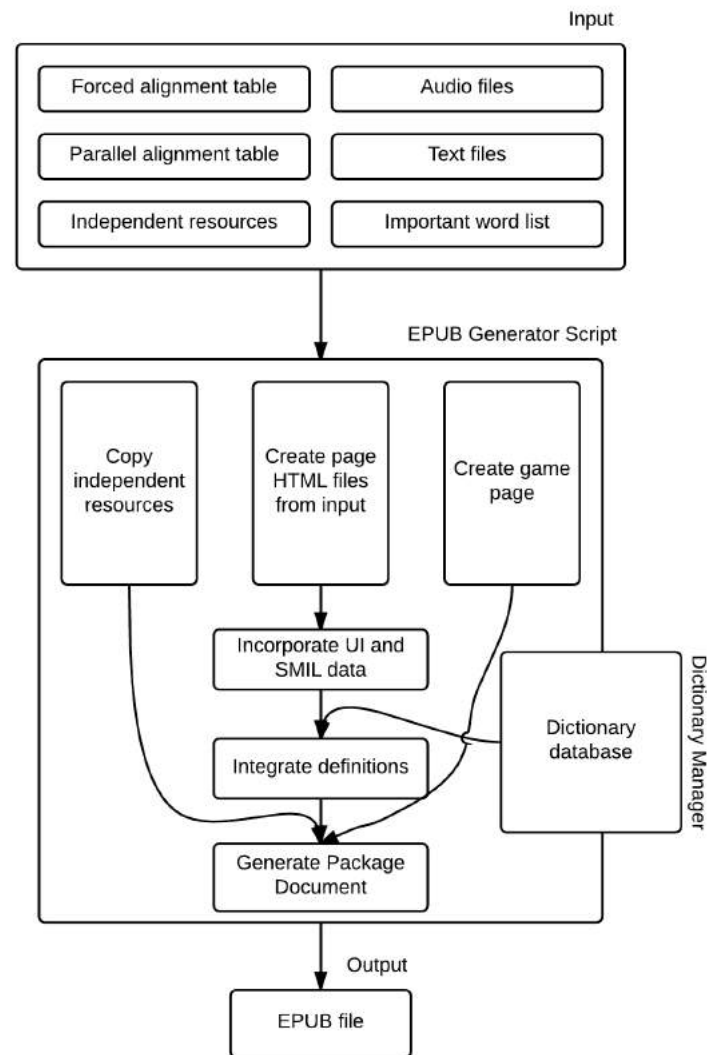


Figure 3.1: High-level overview of the EPUB Generator.

In addition to the dictionary mode, the user interface includes tools which both ease reading along to the audio and rewind the audio without blocking the subject's view of the text. These functions are displayed in figure 3.2. In order, they are play/pause, rewind, repeat, one-by-one mode (in which only one sentence is played, at the end of which the audio is paused), slow/normal pace audio and the dictionary mode. The exact location of the UI can be seen in figure 3.3.



Figure 3.2: Detail of eBook graphical user interface.



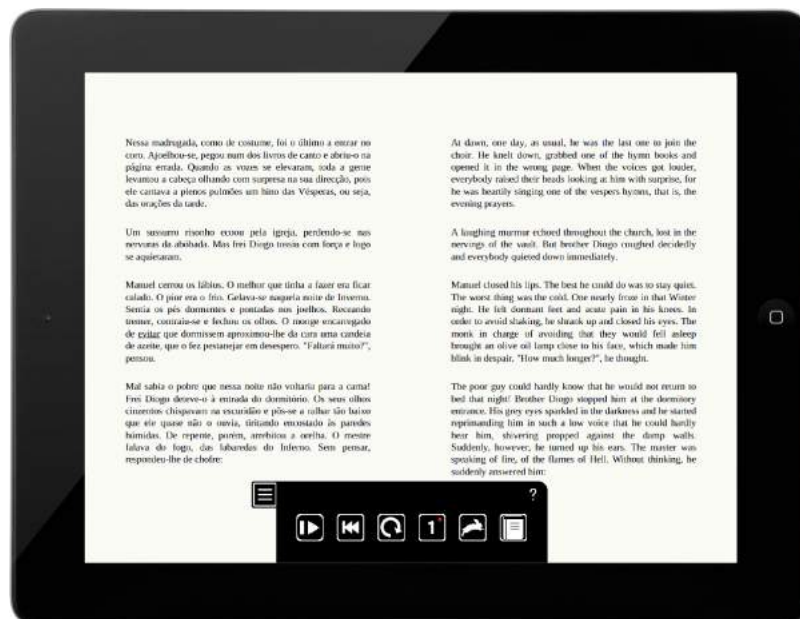


Figure 3.3: Location of the graphical user interface.

The important word list is optional. If none is present, important words are selected from the text on an occurrence frequency basis. Should there be a word list to read from, it is expected that this list contains all words that are regarded to be of interest to the level of difficulty of the input text. These words are to be underlined by default on the eBook, and pressing them will bring up a short definition. The EPUB generator script expects the dictionary, an external resource, to incorporate all words in the text, and enhances the eBook with their definitions. Regardless of importance, however, every word in the eBook will have a corresponding definition which can be accessed through the dictionary mode in the user interface.

Managing and updating this dictionary is the responsibility of the second component, the dictionary manager. All available dictionary entries are stored in a CSV<sup>2</sup> file. Considering the initial scope of this software system, all dictionary information is gathered from Wiktionary, the free dictionary, through the MediaWiki API<sup>3</sup>. New words acquired by the dictionary manager are stored and organized between the following entries: *raw*, *parsed* and *timestamp*. *Raw* stores the raw data acquired from the MediaWiki API, and *timestamp* registers when this raw data was collected. *Parsed* is the string of characters resulting from the dictionary manager's parser, which transforms the raw data from the MediaWiki API to an intelligible word definition.

<sup>2</sup>Comma separated values files allow data to be saved in a table structured format.

<sup>3</sup><https://en.wiktionary.org/w/api.php>

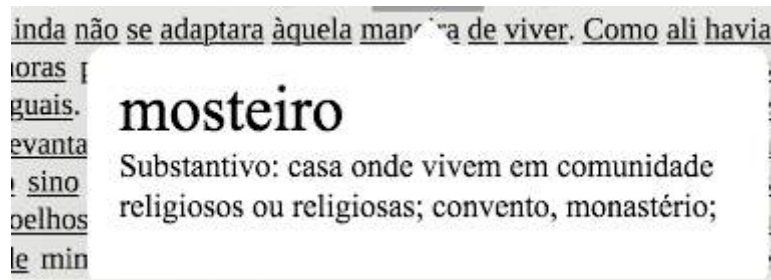


Figure 3.4: Detail of a word definition in the eBook.

The following image showcases how the word definition is displayed in a full page.

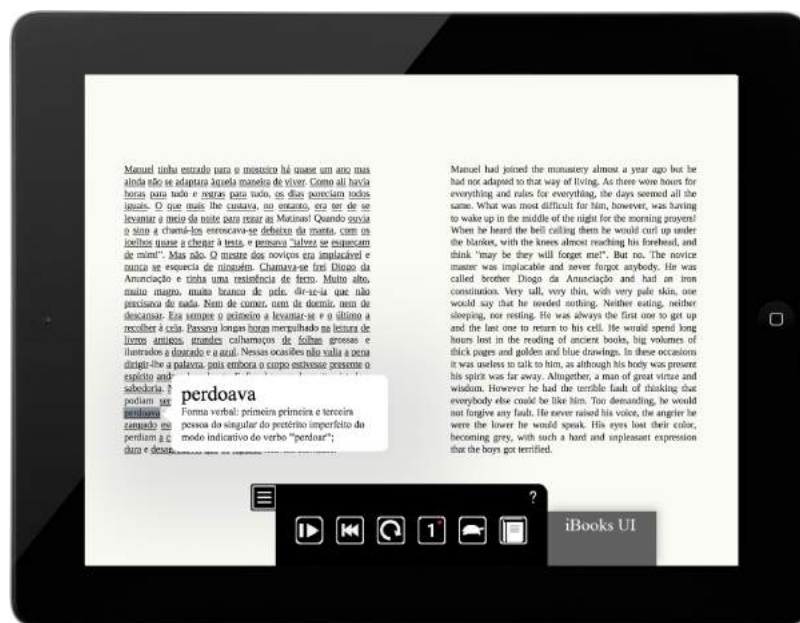


Figure 3.5: Overview of a word definition in the eBook.

Despite using Wiktionary as the source for word definitions, the dictionary manager is able to use definitions from any other source to the extent that it is able to read a well formatted CSV file. Other methods of storing the database could have been used, but ultimately it was decided that CSV files allow for a greater simplicity of data structuring.

Before the EPUB generator adds definitions to the eBook, it prompts the dictionary manager to query Wiktionary for any words that are not contained in the dictionary and parse them. When this task is concluded, the EPUB generator script is then able to add the word definitions and conclude working on the eBook's pages. Finally, it will include the serious games module into the eBook and generate a Package Document, ending the process.

### 3.2.2 Serious games

The serious game module is the last page of the eBook. It is devoid of any text, and contains only the necessary data to run the games. Upon loading this page the user is greeted with two games to choose

from, *Fill in the blanks* and *Word blitz*. The execution of these games is similar, but their presentation is fundamentally different. In both games, the eBook's text is used to create the games. Sentences or whole pages are selected, and from them, a random word or set of words is replaced with a blank which the user is then prompted to fill with the correct missing word, with the assistance of the corresponding audio.

Bernardo, em vez de fazer perguntas, mandou-o trabalhar para ao pé do forno onde os irmãos coziam grandes pães redondos. Um calorzinho bom devolve-lhe as forças. E deitou mãos à obra sem perceber logo por que motivo a cabeça a andar à roda e um vazio no estômago.

Figure 3.6: Sample from *Fill in the blanks* game.

In this first game, *Fill in the blanks*, whole book pages are used, and several words are removed from it. Initially hidden, one by one, the sentences are then revealed to the user, and the corresponding audio track is played. Should a revealed sentence have one or more of these blanks, execution is halted until the user fills all blanks. When the page is completely revealed and all game actions are exhausted, a score is given to the user depending on how they fare. This design results in prolonged games which test a subject's consistency.



Figure 3.7: Overview of the *Fill in the blanks* game.

The second game, *Word blitz*, however, uses a single sentence with a single blank per game, resulting in quicker, more relaxed games with which the player may spend any amount of time they see fit. This will both directly and indirectly aid them in *Fill in the blanks*, as playing *Word blitz* translates

not only into acquiring a greater closeness to the eBook, but also into hints that may be used to help achieve higher scores. The idea for *Word blitz* stemmed from initial testing of the then only game *Fill in the blanks*. It was clear that the latter was strenuous and very time consuming, which was likely to daunt and overwhelm users.

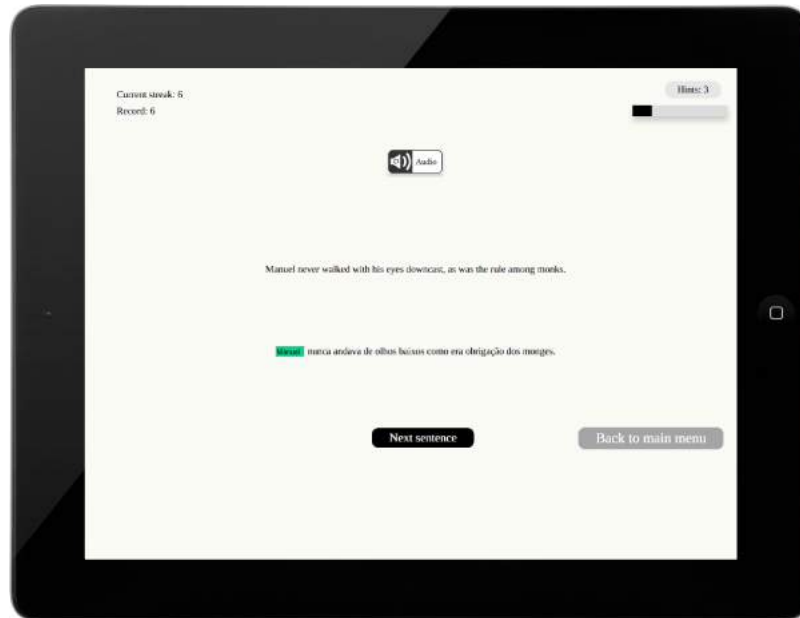


Figure 3.8: Overview of the *Word blitz* game.

The final feature, the hint system, came coupled with the idea for *Word blitz*. It is a gamification tool that rewards users for their time spent in self-assessment. By design, the hint system is only usable during *Fill in the blanks*, but hints may be earned by participating in either game. The hint menu, counter and progression bar are shown in detail in figure 3.9.

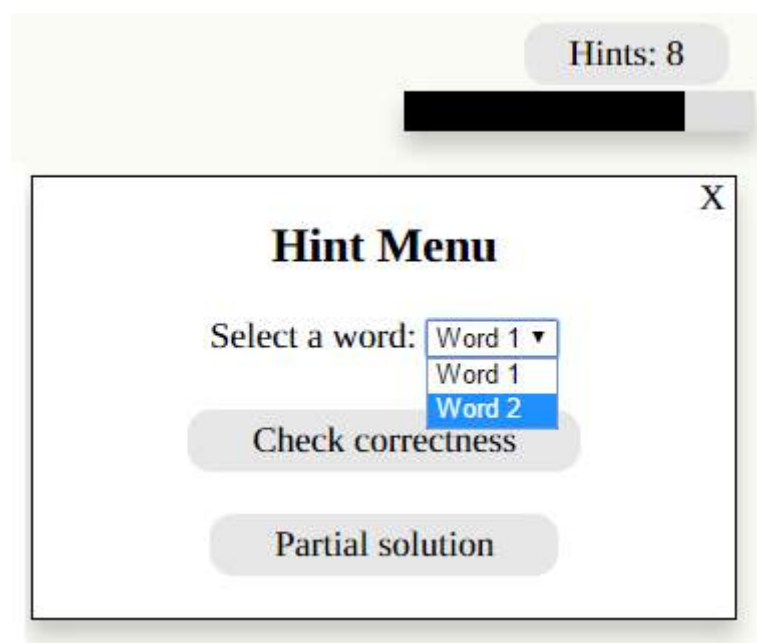


Figure 3.9: Detailed view of the hint system.

As players correctly identify the missing words in the games, the progression bar fills. Once entirely full, they earn a hint. In order to use the hint, for the current sentence, a blank has to be selected through the hint menu's drop down list. Once the blank is chosen, two types of hints are available, *Check correctness*, which, without detriment to their scores, reveals whether the text written in the blank is the correct solution; and *Partial solution*, which writes an initial portion of the word in the blank, useful for when the user has trouble ascertaining the exact word in the audio playback.

This style of game was chosen due to the limited resources offered by the EPUB specification. Taking this and the context of the learning environment into account, few of the game mechanics listed in section **2.1.2 Serious games** could be applied. The fact that all the data required for the chosen mechanics can be contained within the EPUB, through its local storage functionality, is the reason why they were ultimately chosen.



## Chapter 4

# Implementation

The implementation of the software system can be more easily described when divided into four modules: the three components described in 3.2 in addition to the SMIL emulator module imbued in the eBooks.

In addition to these modules it is worth noting that there are some mandatory input resources. Not all of these resources are given directly, and those include the audio files partitioned by page, slowed down audio files and parallel alignment tables. An optional important word list file may be used, but it is not required.

Manipulating the audio file was done through the open source software application Audacity<sup>1</sup>. Using the timing tables made it possible to manually cut the audio file into various segments corresponding to each of the pages in the book. Then, a replica of these segmented files can be edited with the change tempo effect, which can be used to either increase or decrease the speed of audio playback, without, however, affecting the overall pitch. Although this was achieved manually for this project, given the relatively short size of the eBook, it would be fairly trivial to automate this process with a shell script.

The parallel alignment tables are quickly and easily generated using Hunalign, as explained in section **2.2.1 Parallel text alignment**.

### 4.1 EPUB generator

Both the EPUB generator and dictionary manager, modules that together create the eBook, were implemented in Python<sup>2</sup>. In order to create and manipulate the HTML content with ease, the EPUB generator makes use of the ElementTree XML API<sup>3</sup>.

Upon executing the EPUB generator, it loads all resources required for creating the eBook that necessitate no editing, such as the audio files, scripts, images and icons. Then, it loads the parallel alignment and forced alignment tables and uses them to make a word count. Then, depending on the user, a list of words considered important in the eBook is either pulled from a list, or generated with the word count.

---

<sup>1</sup><http://www.audacityteam.org/>

<sup>2</sup>Python and IDLE version 2.7.5, Tk version 8.5.2.

<sup>3</sup><https://docs.python.org/2/library/xml.etree.elementtree.html>

Should the latter be selected, two parameters decide which words are added to the important word list: a count threshold, and a word length (in characters) threshold.

Subsequently the EPUB generator script loads the text for both languages and, page by page, merges their content together into a single file, complete with header declarations and user interface elements. This process continues by using the parallel alignment table to create identifiers for sentences or sentence groups. These identifiers will later allow the SMIL emulator to highlight corresponding sentences in both languages as they are read by the audio playback. Finally, the input forced alignment table is read and converted from its word by word representation into groupings delimited by the identifiers. A script element is then added to the page containing a data structure which lists all SMIL timing information for that specific page. The reason behind the decision of including all necessary data in the pages lies within the constraints of the EPUB specification. As the eBook must not have any online support, data cannot be otherwise fetched from a server, and, in due to Javascript limitations, cannot be loaded from a file contained in the eBook either. The same principle is applied later on in the serious games module, to a greater extent.

Once all pages have been processed, the dictionary manager is loaded. Depending on the user's intent, the dictionary manager may at this point be requested to fetch definitions for any words it may not include, present in the eBook, by referring to the previously gathered word count. Regardless of whether this task is performed, the EPUB generator script will then make use of the dictionary manager's database and add definitions to the words. This is done by using HTML elements *aside* and *a*. The anchor elements, *a*, are linked to these aside elements through a reference number. The default behavior on iBooks<sup>4</sup> dictates that all anchor elements will create a small box on the foreground which includes the aside element's text content, so long as both elements have an **epub:type** attribute to identify their purpose. **epub:type="noteref"** indicates the trigger, and **epub:type="footnote"** the footnote's text. Unfortunately, this behavior seems to be particular to iBooks, rendering the word definitions inaccessible on other platforms.

Afterward, the EPUB generator script concludes the task by creating a final HTML page for the serious games, including in it the HTML *body* elements of every text page as data within a script element. As all documents have been either generated or copied over to the target destination at this time, a final document is created, the Package Document, in which all of the EPUB publication information is recorded, as referenced in section 2.3.

Recreating this module following the described implementation would require approximately a four week effort, which includes familiarization with tools such as ElementTree. Beyond creating the script the developer must also split the audio file and its slowed down counterpart to match the segments for each page and format the text as they wish for it to be displayed in the eBook reader application. As mentioned above, it is recommended to automate that process for multiple and larger books. Independent implementations may take longer as the developer determines which features are or not available for the EPUB specification.

---

<sup>4</sup><https://itunesconnect.apple.com/docs/iBooksAssetGuide5.1Revision2.pdf>



## 4.2 Dictionary manager

This module is much simpler in scope than the previous. It consists of a class definition for a dictionary manager and database. The methods defined within the class allow for querying Wiktionary's MediaWiki API for new definitions, and parsing the received response. The parser was implemented specifically for this application as no other parsers seem to exist for Wiktionary, although numerous working modules can be found for Wikipedia in several programming languages. They were not transferable to Wiktionary, however, perhaps for the unique format that it demands.

Parsing proved to be a challenge, as Wiktionary content does not abide by any well defined standard. Nevertheless, some rules exist, with which the implemented parser attempts to interact to extract the intended definition. Figure 4.1 demonstrates a portion of the result from a typical query, which the parser can process with ease.

```
u"={{-pt-}}=\n==Adjetivo==\n{ [...] }\n{{parox\xedtona|mes|tre}},  
{{m}}\n# [[superior]]\n# [[extraordin\xelrio]]\n#* ''Foi um trabalho  
''mestre'' esse que apresentou.'' \n# [[b\xelsico]]; [[fundamental]]\n#*  
''A viga ''mestra'' cedeu e todo o telhado ruiu.'' \n\n==Substantivo==  
\n{ [...] }\n{{parox\xedtona|mes|tre}}, {{m}}\n# [[pessoa]] que  
[[dominar|domina]] um [[of\xedcio]], [[atividade]] ou [[conhecimento]]\n
```

Figure 4.1: Sample from query for the word "mestre".

Emphasis is made on a few key characters and strings, as these are the tokens used to parse the raw data. Firstly, the dictionary manager splits the raw data string into a list of strings, using the newline character (\n) as the delimiter. Then, one by one, it looks for tokens such as ==*Adjetivo*==, which announce that until a similar token is read, all further data is related to the definition of the given word for that particular word class. In the example in figure 4.1, two such tokens exist. In order, they announce definitions for the given word classes of adjective and noun.

One problem that arises in a few select cases, is that occasionally there are repeat tokens, if not multiple repeat tokens, due to poor structuring of the Wiktionary's page for that word. This alone upsets the parser which repeats these tokens as well on the output string.

Upon identifying a token, the parser begins to generate the string with the definition to that word class. Each number sign (#) represents the beginning of a string containing either a word definition or an example which uses the word in context. In order to distinguish these two, the parser analyses the second character in the string. Should it be an asterisk (\*), then it knows to ignore the string and move on to the next one, as it contains an example.

Words enclosed by square brackets ( [ ] ) represent hyperlinks that redirect the user to the Wiktionary page for that word. Oftentimes these brackets include two words, separated by a vertical bar ( | ). In this case, the word preceding the bar is the Wiktionary word the hypertext links to, and the second represents the word actually shown in the page. All brackets are evidently removed, and if a vertical bar is found within, all but the second word is discarded.

Developing a dictionary management module following the above procedures would necessitate a two week effort. A different degree of effort may be expected should the developer opt for an alternative word definition source, as a new API would have to be studied.

### 4.3 SMIL emulator

This module borrowed its backbone from the now deprecated `rb_smil_emulator`<sup>5</sup> Javascript module created by Alberto Pettarin<sup>6</sup>. `rb_smil_emulator` allowed for mirroring the operation of iBooks's built-in SMIL interpreter. Taking this approach was necessary as each of the eBook's pages must be bound to a single SMIL file, which in turn must be associated to a single audio file. Given there is no support to control the pace of the audio when using SMIL, it was deemed necessary to use two separate tracks, one for the regular audio track, and a slowed down version of the same track. This demands the creation of a separate SMIL table, which in turn rules out the possibility of using iBook's built-in reader. As a result, the `rb_smil_emulator`, which was also meant for single audio file use, was adopted and modified for use in this software system. A simplified overview of the initial `rb_smil_emulator` module is given in figure 4.2

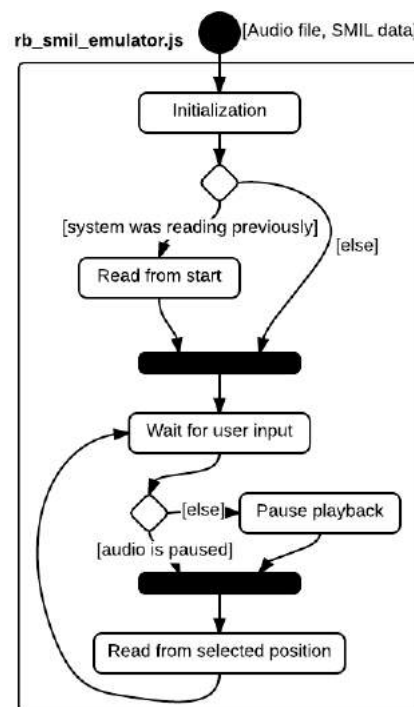


Figure 4.2: Simplified activity diagram of `rb_smil_emulator.js`.

The final implementation consists of a Javascript code able to not only simulate the iBooks SMIL interpreter but also provide extra features otherwise not available. Such features include ways to repeat or rewind the audio playback, set a multiplier to the SMIL data and switch the audio track to reproduce the toggle effect for slowing/speeding up the audio playback, and enabling dictionary mode during which no audio playback is allowed.

<sup>5</sup>[https://github.com/pettarin/rb\\_smil\\_emulator](https://github.com/pettarin/rb_smil_emulator)

<sup>6</sup><http://www.albertopettarin.it/>

One of the goals of this implementation was the ability to control the iBooks overlay. The overlay's default behavior makes it so that it will be toggled with every single touch event processed by the SMIL emulator. This, however, caused problems as the overlay covers a large screen area on the iPad, consequently cutting down on areas that could be used to include user interface elements.

The work around to this problem was not an easy one. Very little to no documentation is available on the iBooks own injected Javascript which controls this behavior. Ultimately changes were made to the whole HTML structure of the pages. Each intractable object in the eBook is now tagged. When the SMIL emulator detects a touch event, it determines its target. Should it be one which triggers the iBooks user interface, then, before any further action is taken, the SMIL emulator prevents all default behavior through use of an event method. In addition, an invisible background area has been defined which encloses each page, to ensure that touch events that do not target any object are still caught by this handler.

One final notable method defined in the SMIL emulator is that of the dictionary mode toggle. Initially the HTML was structured in a way such that each text word was contained within its own span element. Words with definitions however, would be present within an anchor element, as discussed before. Whether or not a word had an important word tag associated with it, each span element included an anchor element. If the word was not on the important word list, the anchor would be devoid of text, but if it indeed was on the list then the span element would be the element with no text instead. Upon request, the SMIL emulator would go over each and every word in the page, and, for any word not tagged, swap the text content of the two elements. This implementation was successful on several platforms, but not on the iPad, where the process was too slow.

To solve this issue, all words were included in anchor elements instead. To remove the hypertext formatting, three anchor classes were defined. One represents permanent word definitions, while the others denote active and inactive word definitions. Similarly to the previous method, upon receiving a dictionary toggle request, the SMIL emulator finds every anchor element and, should they not be associated to a permanent word definition, swaps their active status. Additionally, in order to ensure that inactive word definitions cannot be accessed and are not underlined due to default hypertext formatting, a simple CSS .class selector is bound to the inactive word definitions class as shown in figure 4.3.

```
a.nonperm_note {  
    text-decoration: none;  
    pointer-events: none;  
    cursor: default;  
}
```

Figure 4.3: CSS redefinition for non-permanent word definitions.

One final issue remained, however. The audio playback was being partially cutoff in the iPad platform. Before one sentence could be completely read, audio would be stopped a fraction of a second before the end of a SMIL segment, with the proceeding segment starting playback. This timing issue was exclusive to the iPad, and despite the considerable effort put into identifying the root cause none could be found.

No documentation exists regarding this issue, and despite attempting multiple different implementations the problem persisted.

Ultimately a crude solution was implemented. It should not be interpreted as an actual solution to the problem, but rather a temporary patch. Audio playback in SMIL emulator is controlled by timers. When a play event is processed audio playback starts from the given point and a timer is set to the difference between the end and begin SMIL tags of the segment being played. This patch consists of adding extra time to the timer if the reading system being used is iBooks.

```
if (rb_smil_emulator.current_reading_system == "ibooks") timer_offset = 350;
rb_smil_emulator.timer = window.setTimeout(rb_smil_emulator.on_next_event, (end - begin) * 1000 + timer_offset);
```

Figure 4.4: Workaround to the timing issue.

The offset value of 350 milliseconds was chosen through rudimentary experimentation and is by no means the exact delay value caused by the unknown obstacle.

Recreating a version of the SMIL emulator module utilizing the information within this section and using Alberto Pettarin's `rb_smil_emulator` would allow for a short development cycle of approximately two weeks. The addition of new components or features will invariably increase the overall effort as the impacts of a single addition are often widespread. Any new feature is cause to review the iBooks overlay control module and special attention must be given to the placement of new HTML elements. Extensive testing must be done in the target system, as the performance of the script on the development environment has no direct correlation to the performance on the target system. This, again, is due to the iBooks application's injecting Javascript code to each page of the eBook. For each new feature planned a one week effort may be expected to test and correct unforeseen deficiencies.

## 4.4 Serious games

This final module concerns the three Javascript modules implemented to handle the serious games. One module is responsible for storing and maintaining all game related data such as game scores and number of hints available, as well as generating the HTML content of the game menu. This also includes methods to clear the HTML page of game content when games are concluded. The remaining two modules are responsible for one game each.

The first game, *Fill in the blanks*, uses a version of the SMIL emulator mentioned beforehand. It is heavily modified, includes only the normal pace audio file, and does not register touch events. Instead, the task of moving on to the next sentence of repeating the audio of the active sentence falls upon buttons. Button presses invoke functions of this modified SMIL emulator that reuse the preexisting audio control functions.

Upon loading, the game page is generated by taking the HTML body data of each page in the eBook that was stored into a variable as a string array in the single script element created by the EPUB generator. Whether or not a page is randomly selected, the body content is taken from this array and

used to replace the serious games HTML body content entirely. All segments are then made invisible in preparation for initialization of the game. Furthermore, the UI elements are appended to the document and the number of hints and hints progression stored data is loaded.

The selection of words to be removed is made before the game begins, and these words are selected randomly within the boundaries of three parameters. Firstly, the distance between two words can be no lower than the designer's specified minimum threshold and no greater than a maximum threshold. The final parameter is the minimum word length. Only words greater in character count than this parameter are used in calculating the distances of the previous two parameters. This process uses a cumulative distribution function to ensure that the randomly selected words meet this criteria. All selected words are then stored in a list for use with the hint system and scoring. This process is entirely dynamic, making it widely unlikely that two games generated for the same page are composed of the same set of blanks.

Some alternatives to the word selection process were taken into consideration, such as using the difficult words as the blanks. This could be easily implemented as all difficult words are previously marked as such, and that information is currently passed on to this module. Another solution could use a similar concept, in which a second important word list is loaded by the EPUB generator module and used to mark words in use for this game.

The most apparent drawback of non random approaches such as these is the fact that it reduces the lifetime of the game considerably. Should students notice that the blank patterns remain unchanged during subsequent attempts, they are more likely to revisit the relevant eBook pages to memorize the given words. Additionally, using a second word list rather defeats the purpose of using a word list at all, as words deemed to be important in the eBook absolutely must also be considered important during serious games. The chosen solution also has the advantage of guaranteeing that the game is paced appropriately, assuring that the blanks neither quickly follow one another nor become heavily concentrated in one short section of the game page or are too sparse. These are the chief concerns for player engagement and motivation.

The chosen method is, however, not without its own drawbacks. It is entirely possible that the word selection is so poor that the game becomes either too easy or entirely too difficult to solve. The problem that stems from an easy game is that, although it may briefly motivate players, when they are later presented with a balanced word selection they may experience frustration and confusion as to why the game words have suddenly become, in their context, more obfuscated. Difficult games, on the other hand, may drive players away from the game as they feel their studying was not rewarded.

The number of games for this implementation comprises of the entirety of pages in the eBook, as the book is fairly short. It is unreasonable to mirror this behavior for a larger corpus. In such a situation a select number of pages should be chosen, electing to generate games for pages which have a greater number of important words or important word density.

As for the second game, *Word blitz*, the SMIL emulator is not used in any way. Considering a single sentence is to be read, such a system would be rather cumbersome in relation to the simplicity required of it. In a fashion similar to *Fill in the blanks*, a page is selected at random to be used as the game page for one game, and its visibility set to hidden. Upon acquiring the number of segments in the selected

page, a new random number is used to select which segment will be used for the game, and a word is selected to be the removed. Both these selections have criteria, as in *Fill in the blanks*. Eligible words are those with a character length greater than a given threshold, and a minimum threshold for the segment length, in words, is also used. However, the algorithm responsible for selection of the game word uses a simple discrete uniform distribution, unlike in the previous game, for a single word is to be removed from the segment.

Word selection for this game is similar to that of the previous game. The decision to use this specific approach was one of uniformity. In this scenario using the important words marker becomes much more appealing, as the sentence pool is larger and more unpredictable than in the previous game. Also, given that the results are not tracked and graded as they are in *Fill in the blanks*, the pressure to success a player may feel is extensively lower, and thus they would be less likely to try and exclusively memorize the important words from the eBook.

While developing the second game, the weaknesses of EPUB in regards to the use of Javascript to procedurally generate pages became apparent. After a first implementation, which was entirely successful on the Google Chrome and other web browsers, testing on the iPad revealed that ibooks application would become slow and unresponsive. Naturally that version was discarded for a new software architecture which relied much less on generating HTML than its predecessor in order to create content for the game.

The implementation of the first game necessitated a four-week-long period, but may be longer depending on the chosen method for word selection. The second game, being much simpler in nature, requires a one to two week development effort depending on the performance of the game in the target system.

## Chapter 5

# Results and Evaluation

The development of the system was supported by feedback from a test group comprised of both Portuguese and foreign individuals. This group indicated which features are of their preference and offered further development suggestions. The foreign participants consisted of seven individuals from both the United Kingdom and the United States of America who had either negligible or no prior experience with the Portuguese language. The Portuguese participants' experiences could not be included in the results since they were given the same demonstration eBook, where the target language matched their mother tongue. Their feedback was taken into consideration for matters such as usability, user interface and user experience development. In the survey results section these two groups are considered separately.

It was clear from the initial stage of the project that conceiving a model with which to test the efficacy of the eBooks was a complex task. Having a small test group which would be asked to evaluate the final system in accordance to their personal preferences was the model ultimately adopted. This option was chosen in lieu of a comparison of a test group using the eBooks developed in this project and a control group using a different form of learning aids. The reason for this being that when using small test groups, it is inconceivable that both groups integrate individuals of comparable learning propensity and identical L2 backgrounds. Consequently, it would be impractical to analyze and compare results based on the two test groups. Thus, the evaluation will be one of a subjective nature, revolving on aspects related to the preferences and tastes of each subject.

The survey consists of two phases. Firstly, each individual interacted with the eBooks with little to no guidance. For the purposes of this first phase, they would express their opinions on each of the functionalities, and also whichever thoughts crossed their mind, as was asked of them before the test took place. After this first experience, a short, informal conversation followed. The intent behind this was to scrutinize any point that they might have missed or to inquire further rationalization to a particular point of interest they might have previously alluded to.

Finally, the formal survey would then be presented at the end of the discussion. This survey aims to evaluate the success related to the hypotheses conceived for this learning tool, which are described in the subsequent section.

## 5.1 Survey hypotheses

Ahead of drafting the survey, the following hypotheses were established. These hypotheses represent the ideas upon which the eBooks were designed, and reflect aspects concerning both the design of the dual language eBooks as well as serious games.

- *H1 - Audio support promotes second language comprehension in dual language books.*

This is the main idea behind this work. Every other function expands on this concept and as such it is considered the focal point of the survey.

- *H1.a - The highlighting aids reading along to the audio playback.*

One of the functions which builds upon on H1. It is expected that the offering of a simple visual queue to the reader facilitates reading along to the audio playback.

- *H1.b - The highlighting allows for an easier correspondence of phrases in either language.*

This is yet another intended purpose for the highlighting. The anticipated effect is that of aiding in the analysis of the meaning of the corresponding highlighted sentences. Knowing the corresponding blocks of text is expected to enable a finer interpretation of each individual word.

- *H1.c - Control over audio speed and playback facilitates the acquisition of phonetic characteristics of the L2.*

This encloses all other implemented functions, such as the UI, slower playback speeds, and playback control. One example is that of the replay button, as it allows maintaining the text in plain view while the audio is being repeated, as often as one would like, without having to press the transcript, thus blocking their own view.

- *H2 - Serious games motivate and gratify second language learning.*

This is the goal behind gamification of the self-assessment tools. Gamification, when well designed, should appeal to its target audience and influence their behaviour concerning their tasks, raising levels of engagement, motivation and sense of gratification.

- *H2.a - The hints system drives users to dedicate more time to their studies.*

This acts as the game's currency. It correlates directly with a tangible benefit the students may use at will. It rewards those who carefully study the book beforehand, and eases any possible feeling of stress derived from a situation where one correct answer is all that is holding back the student from a perfect score.

- *H2.b - The scoring system encourages players to commit themselves for better results.*

As briefly touched upon on the previous hypotheses H2.a, the scoring system should drive players who appreciate gaming self-elements to raised levels of engagement in an attempt to pass the books games with a perfect score.



## **5.2 Informal discussion**

The informal discussion can be expressed as the informal data acquired from each person's first reactions to the developed functionalities. This data is invaluable as the aim of this survey is not to test the efficacy of the presented tools per se, but to collect information about each user's expectations and reactions to the manner in which the functionalities were implemented.

Therefore, although the intent was to intrude as little as possible with each user's exploration of the eBook, each discussion was processed differently. While some individuals took to the UI very rapidly, others were halted by a lack of understanding of a few certain features, and needed direction.

Nevertheless, as was requested of them, all individuals would express their impressions towards the features they examined. A short discussion followed as soon as the eBook was fully explored, during which further questioning was conducted in order to fully explore what might have been missed opportunities, reasons as to why certain aspects were not to the individual's liking, and so forth.

It should be noted that all twenty-one participants were shown the same short demonstration eBook. The target language for this eBook is Portuguese, meaning all audio and games are featured in this language. While the text pages are the same for all participants, due to the random nature of the serious games, no two participants were given the exact same games to play. Even those who coincidentally chose the same game pages did not experience the same challenges, owing to the fact that the blanks are generated dynamically.

Analysis of the results is best partitioned into two categories. The first being comments on current functionalities and the second comments on desired implementations. All twenty-one participants in the user study also participated in the informal feedback.

### **5.2.1 Current functionalities**

#### **1. Button feedback and responsiveness**

On this subject, participants were mostly critical of the lack of a toggle or feedback when using certain functions. The one-by-one mode caused more confusion as it was not clear whether or not the mode was active or not, which sixteen participants pointed out. All five participants who attempted to use the eBooks on smaller devices such as the iPad mini or iPhone expressed frustration as it was very difficult to press words and also due to poor responsiveness from the iBooks application.

#### **2. Language learning effectiveness**

A group of eleven individuals, who conveyed more enthusiasm towards this project, underlined the fact that these books are very poor in regards to learning correct grammar, and suggested that a grammatical highlighting function be implemented in the future. It was agreed, however, that it was a good process to acquire new vocabulary, to the point where non Portuguese speakers could correctly identify the meaning of each Portuguese word in a sentence without any assistance.

### **3. Word definitions**

Eight individuals noted that certain word definitions were rather poor, as their Wiktionary definitions were of little value.

### **4. Original audio speed is too fast paced**

With regards to this point, a small group of four would have preferred for the audio playback to be naturally slower. As the eBooks is meant to serve as a tool for learning correct articulation and co-articulation of words, however, this demand would require the speaker to talk in an unnatural way. This would invariably reduce the quality and effectiveness of the eBook as a language learning application.

### **5. Serious games - Fill in the blanks**

When it came to the serious games, most feedback was either positive or neutral, with a few details of note. Sixteen participants found the ability to edit past answers on the first game, *Fill in the blanks*, to be rather confusing, suggesting that the feature should be discarded.

Six participants noted that the games taught very specific things that may not necessarily be what one would prefer to learn. It also rewards those who have memorized the book, rather than testing on how to use the new vocabulary learned from the book.

### **6. Serious games - Word blitz**

On the second game, *Word blitz*, seven users were unsatisfied by the fact answers could only be either absolutely correct or wrong, with no possibility of scoring partial points. This was more predominant a problem as the games were to be solved in Portuguese, and often did the correct solutions include diacritics that foreign users could not write in their devices. These diacritics are present, however, on Apple's iPad. This is true even when the keyboard language is set to English. As such, unless future development aims to expand into other devices, this is not regarded as a fault of the learning system.

## **5.2.2 Desired implementations**

### **1. Single word reader and highlighting**

The most popular request among all feedback received, which was alluded to by eight participants, was for the development of a function similar to the one-by-one function, but, instead of reading whole sentences, it would read the selected word alone. Although the idea has merit, for the eBook used as a demonstration during the user survey this would not have been a simple task, as the narrator often employed natural pronunciation. Four people also suggested a similar idea pertaining to word for word highlighting. This would entail a deeper analysis of the text, and while the idea is worth taking into consideration for future work, one should note that the highlighting for the non-target language would be, at times, jumping erratically from one point of the sentence to the next. This effect becomes more noticeable the further apart the two languages are, as sentence structure may differ widely between the two languages.

## **2. IPA phonetic notation**

Another interesting suggestion given by three participants was the inclusion of the IPA phonetic notation in the dictionary word definitions. The reason for not including these in the eBook was that the reader, ibooks, is unable to display these non-standard characters in the word definition windows.

## **3. Reading pace**

Finally, two individuals noted that, despite the audio speed control function, there ought to be a process that depending on the length of each sentence read, the playback should pause for a short proportional amount of time, giving readers the necessary time to fully catch up to the audio playback.

## **4. Serious games**

As for the serious games, fifteen participants noted that the games should present the player with the correct answer at the end, so that they may learn from it. Moreover, the game could subsequently offer the definition of these words. It was also suggested by five participants that a short game description and explanation of the hints system be given on the main game menu, so as to avoid confusion.

Ideas for some games were given, and included translating a word from the transcript, or asking automatically generated questions about the book.

### **5.2.3 Other applications**

A few individuals pointed out some contexts in which they would like to see this technology used. Examples such as poems and librettos emerged, but the same fundamental idea was behind all suggestions. Most enthusiasts of such forms of arts would appreciate from having the original work, with its original rhyme scheme and prosody, while still having a translation of these transcripts available to them, complete with highlighting and audio functionalities such as the ones in the eBooks generated in this project.

## **5.3 Formal survey**

The user survey focused on three aspects. The participants' demographic and their assessment of the features of the eBook, for which the original book and serious games are taken separately. All propositions are answerable on a scale of one to five, where one represents a complete disagreement and five a strong agreement.

### **5.3.1 General demographic questions**

Firstly comes the demographic related enquiries for all twenty-one participants, which are detailed below in addition to their results.

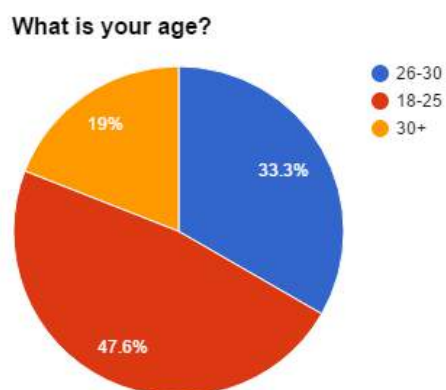


Figure 5.1: Demographic question 1.

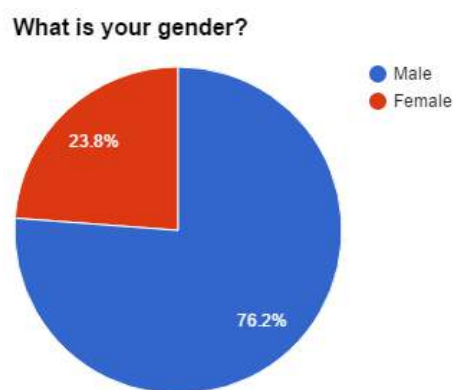


Figure 5.2: Demographic question 2.

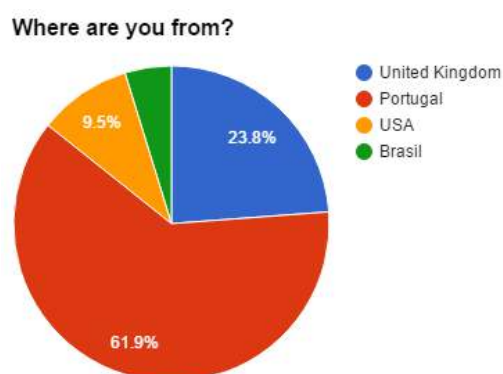


Figure 5.3: Demographic question 3.

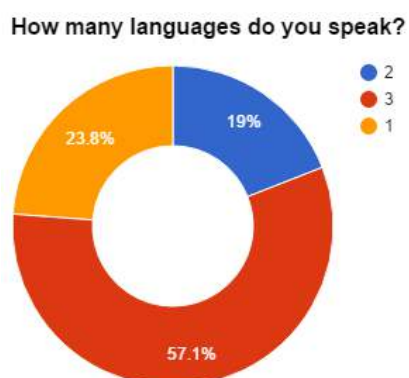


Figure 5.4: Demographic question 4.



Figure 5.5: Demographic question 5.

The question in figure 5.5 refers to language learning outside of the classroom, in which the student used solely materials at their disposal to learn the second language. Such materials could include workbooks and online language learning resources.

A follow up question was given to those who gave a number greater than zero in response to the aforementioned question: *What tools have you used in self-teaching a language?* The responses to this question were YouTube videos, books, Duolingo<sup>1</sup>, Google Translate, video games and television.

<sup>1</sup>Duolingo is an online language learning platform that may be used free of charge. It can be found on the application store for both android and iOS devices, as well as at <https://www.duolingo.com/>

### 5.3.2 Survey results for eBook features

In order to analyze any possible bias to the next group of questions, first a couple of questions were asked to determine the view of each individual on dual language eBooks, with a score of 1 representing *not at all effective* and 5 *very effective*.

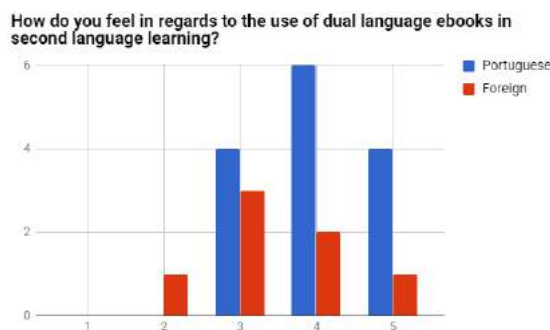


Figure 5.6: eBook preliminary question 1.  
Means: Portuguese 4.00; Foreign 3.43.

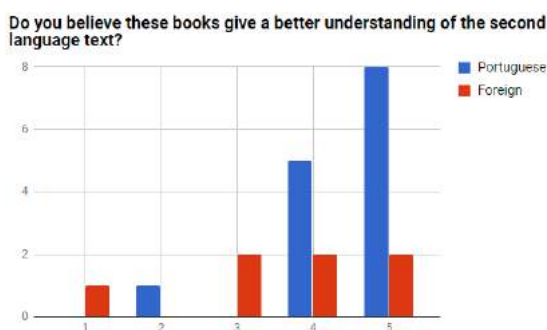


Figure 5.7: eBook preliminary question 2.  
Means: Portuguese 4.43; Foreign 3.57.

Results are slightly favorable, although lower, and close to neutral, when analyzing results from foreign participants. This may be due to cultural differences, as non-Portuguese participants all resided in English speaking countries, where learning a second language, especially at a young age, is not as common a practice as it is in non English speaking countries. Thus, when attempting to self-teach a language, a number of difficulties arise in the form of acquisition of basic grammar and vocabulary, for which dual language books are ill suited. Further analysis of the data revealed it to hold little to no noticeable correlation with the questions to follow.

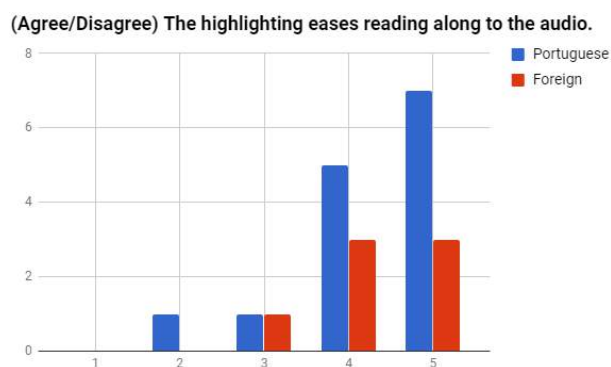


Figure 5.8: eBook question 1.  
Means: Portuguese 4.29; Foreign 4.29.

Results are largely favorable, indicating a general consensus that the implemented functions are well suited to meet the formulated hypotheses. However, a large portion of this positive feedback does not express complete satisfaction, as expressed by the large number of 4 replies. It is straightforward to deduce the implication that these features, although satisfactory, could be improved.

(Agree/Disagree) The highlighting makes matching the second language text to the native language easier.

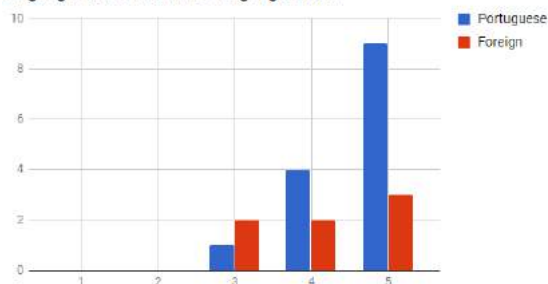


Figure 5.9: eBook question 2.  
Means: Portuguese 4.57; Foreign 4.14.

(Agree/Disagree) The functions controlling audio are helpful in the sense that you are better equipped to understand the correct pronunciation of words.

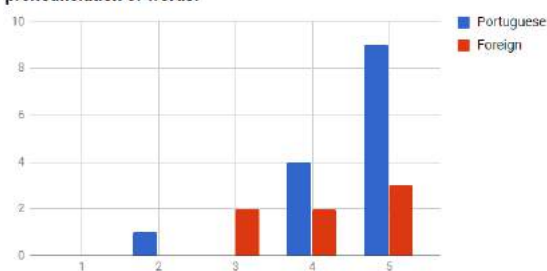


Figure 5.10: eBook question 3.  
Means: Portuguese 4.50; Foreign 4.14.

### 5.3.3 Survey results for serious games

Finally, the survey moves on to the subject of serious games. Again, to scrutinize for existing bias, a couple of preliminary questions lead the section.

Have you ever used gamified approaches to language learning?

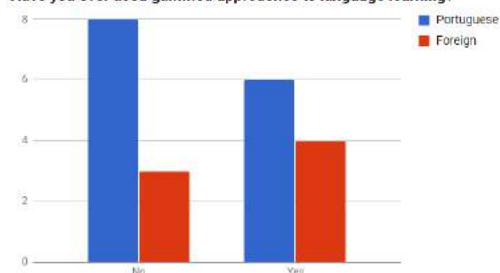


Figure 5.11: Serious games preliminary question 1. "Yes" percentage: Portuguese 42.9%; Foreign 57.1%.

How would you rate these approaches in regards to the motivation and gratification to your learning?

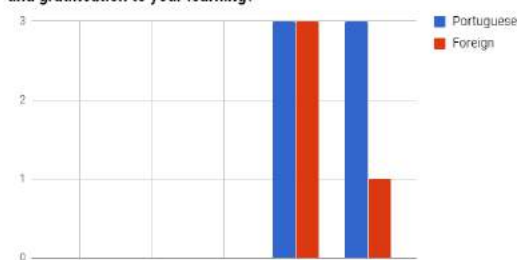


Figure 5.12: Serious games preliminary question 2. Means: Portuguese 4.50; Foreign 4.25.

The second question, for which results are demonstrated in 5.12, was only presented to those who had used gamified approaches to language learning. It is notable that the range of results for this question ranged from positive to very positive, indicating that these participants consider gamification to having been beneficial to them.

(Agree/Disagree) The games included in the book are interesting enough to keep players' attentions.

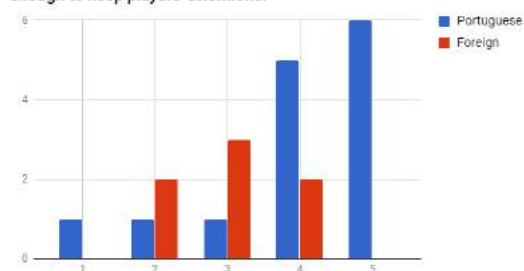


Figure 5.13: Serious games question 1.  
Means: Portuguese 4.00; Foreign 3.00.

(Agree/Disagree) The games are a good way to self-assess your learning.

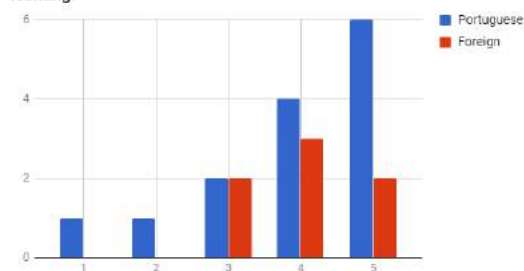


Figure 5.14: Serious games question 2.  
Means: Portuguese 3.93; Foreign 4.00.

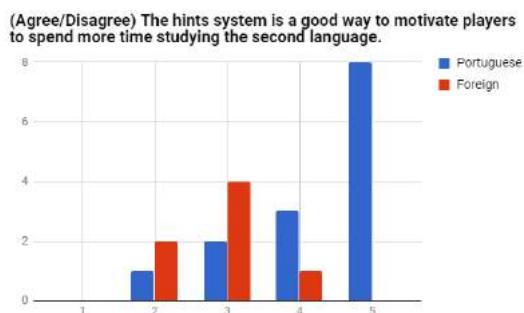


Figure 5.15: Serious games question 3.  
Means: Portuguese 4.29; Foreign 2.86.

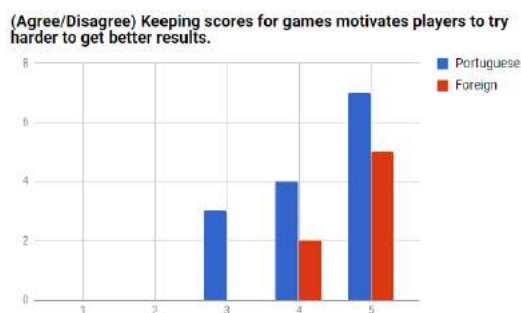


Figure 5.16: Serious games question 4.  
Means: Portuguese 4.29; Foreign 4.71.

Although results were mostly positive, the hints system pales in comparison to the other features, indicating that this feature requires additional development. Of note, a significant 38% of participants gave a score of 3 or less to the first question, regarding their engagement level with the games offered. This is not unexpected as during the informal discussion an ample number of participants offered suggestions for new games and improvements for the current ones.

Most individuals did agree, however, that the games are a good manner of self assessing one's learning. This, again, reinforces that although there is a requirement for such a nature of training exercises, the implementation could be improved.

When filtering the results according to the preliminary questions, results did not seem to alter in any significant manner. However, when filtering with respect to demographics, focusing on individuals who did not speak Portuguese, questions 1 and 3 are altered significantly, with predominantly neutral responses.

A possible interpretation could be that, as hypothesized beforehand, people with backgrounds from the United Kingdom or United States of America may expect games that will cover the basis of the language they are studying. This idea is fundamented by the ideas given by these individuals during the informal discussion segment of the survey, in which participants would ask for word translation and grammatical analysis, or even simple automatically generated questions about the book. Most likely, however, is the bias of the Portuguese participants. Given that the demonstration eBook's target language was their own, this placed the Portuguese and foreign participants in different contexts despite being given the same material to analyze. Only the foreign participant's results should be taken into account for the efficacy of the serious games, since they alone are in the correct language learning context.

Also, in a day and age where foreign media and is widespread, these games might be perceived as overly ineffective in comparison to already existing material.

To summarize both informal discussion and formal survey results, the participants' assessment of the eBook was largely positive, although not excessively so. Some key points that require further revision include:

- Grammatical analysis of the transcript.
- Arrange to collect better word definitions, and their IPA if possible.

- Word for word playback.
- Creating a short description of the games within the main game menu.
- Revealing the correct solutions to the games.
- Improving on the hints system.
- Developing a new game that is different from the existing ones altogether.

Despite these numerous weaknesses, it became evident that this project possesses a vast array of strengths which, if fully developed, will prove to be a valuable resource in the field of second language learning. Still, some of the suggested developments require resources beyond those included in the EPUB format, and thus impossible to implement naturally.



## Chapter 6

# Conclusions and Future Work

### 6.1 Conclusions

This project aimed to fulfill the need for a more substantial form of second language learning through dual language audio books and serious games. In pursuance of this goal, a software application was designed and implemented in a way that would allow for the automated production of these books given solely the transcripts in two languages and a respective audio counterpart.

The hypotheses formulated in the preceding chapter were met to a satisfactory degree by the results of the user study. This is due to the fact that most responses were positive in regards to all points made, with a sole exception regarding the hints system. This component was clearly short in delivering the intended gamification element of currency, and should be improved upon.

Considering the implementation phase and features participants requested for during the user study, the EPUB format fell short of expectations. It was not possible to use SMIL dynamically as Javascript code that altered the HTML pages in any form ran slowly and, despite tailoring a SMIL emulator to suit the eBooks's requirements, timing issues plagued every function call that depended on audio playback.

Considering that the above issues were not at all present when testing on web browsers such as Google Chrome and Internet Explorer, one assumes that the EPUB format is not prepared for dynamic content that is not created through its own built-in functionalities. These processes are, however, very basic in nature and unsuitable for projects that require even the least amount of complexity.

We have to conjecture then that, despite some EPUB readers allowing it, Javascript should not be relied upon to create eBooks in the EPUB format, least of all rely upon it as heavily as this project has.

Regardless of these shortcomings, the overall goal of this project was reached to a satisfactory degree, allowing for it to be interpreted as a successful proof of concept. With further development, this tool would certainly prove to be a robust complement to second language learning and teaching.

## 6.2 Future Work

Future considerations to expand on this project should explore options beyond EPUB. Perhaps an ebook reader could be designed to catalog and display the ebooks, or each book could include a self-contained reader. Having control over the reader application would allow for designs of greater complexity, including features unavailable to EPUB, such as loading auxiliary files to meet data requirements for simpler, easier to maintain and develop software architectures. Also unavailable in EPUB, internet access could allow for recommendations of eBooks given each students' personal library or latest readings.

Current features that could be improved upon include the dictionary, which would benefit from comprehensive word definitions and their IPA phonetic representation, as the current definitions are commonly poor. Furthermore, a grammatical analysis of the text would enable narrowing down the word definitions to those relevant for each context. Moreover, the grammatical analysis would allow for the much requested grammatical highlighting function during the user study.

Further improvements may be made to the playback control and the given interface, as suggested by the user study participants. A function that, depending on the length of each sentence, pauses audio playback for a short proportional amount of time would give readers a longer time to catch up should they fall behind on their reading. Another possible development is one that would allow for playing a single word, to allow a better understanding of the second language. This development would require special attention to relaxed pronunciation. Another benefit of this design when coupled with grammatical analysis would be the possibility of including word for word highlighting during regular audio playback.

The serious games could be improved upon as well, as the current selection was not enough to please most user study participants. New games could take the grammatical analysis of the text into context and prompt players to translate a word in a sentence, e.g. translating the adjective in a sentence, or to even generate simple questions about the book.

As for the current serious games, further development should take the user study's informal discussion into consideration. In regard to *Fill in the blanks*, one should consider disabling incorrect answers, as most users found it confusing that they were able to edit these past answers. Regarding *Word blitz*, a different measure of score could be used, perhaps, to reward answers that are very close to the solution. As several written languages include diacritics that are not present in others, it may be altogether impossible to correctly write them in certain devices.

Although these eBooks are designed specifically for the Apple's iPad device, on smaller devices such as the iPad mini or iPhone utilizing the eBooks is incredible strenuous. As mentioned previously, including zooming and panning procedures could greatly enhance a readers experience in these cases.

A final interesting concept is that of reading along with the audiobook. While this simple notion is not impeded by the current project, there is no feedback or assessment of pronunciation. Should the eBooks indeed be contained in a platform other than ibooks, then it might be possible to record the reader as he narrates the second language text. Given that, the recording software platform could then parse and evaluate the reader's performance, indicating which words require further attention.

# Bibliography

- L. Anderson-Inman. Computer-based solutions for secondary students with learning disabilities: Emerging issues. *Reading & Writing Quarterly*, 15(3):239–249, 1999. DOI: 10.1080/105735699278215.
- E. Balajthy. Text-to-speech software for helping struggling readers. *Reading Online*, 8(4), January 2005. ISSN: 1096-1232.
- I. Borrás and R. C. Lafayette. Effects of multimedia courseware subtitling on the speaking performance of college students of french. *Modern Language Journal*, 78(1):61–75, 1994. <http://dx.doi.org/10.2307/329253>.
- S. Brognaux, S. Roekhaut, T. Drugman, and R. Beaufort. Automatic phone alignment. In H. Isahara and K. Kanzaki, editors, *Advances in Natural Language Processing*, volume 7614 of *Lecture Notes in Computer Science*, pages 300–311. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-33982-0. doi: 10.1007/978-3-642-33983-7\_30. URL [http://dx.doi.org/10.1007/978-3-642-33983-7\\_30](http://dx.doi.org/10.1007/978-3-642-33983-7_30).
- A. Creese, A. Bhatt, N. Bhojani, and P. Martin. Multicultural, heritage and learner identities in complementary schools. *Language and Education*, 20(1):23–43, 2006. ISSN: 0950-0782.
- G. E. Dahl, D. Yu, L. Deng, and A. Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(1):30–42, 2012.
- R. Das, J. Izak, J. Yuan, , and M. Liberman. Forced alignment under adverse conditions. In *Unpublished manuscript*, 2010. [http://www.seas.upenn.edu/cse400/CSE400\\_2009\\_2010/final\\_report/Das\\_Izak.pdf](http://www.seas.upenn.edu/cse400/CSE400_2009_2010/final_report/Das_Izak.pdf).
- G. Davies. Computer assisted language learning. *Subject Centre for Languages, Linguistics and Area Studies Guide to Good Practice*, 2002.
- R. P. Dolan, T. E. Hall, M. Banerjee, E. Chun, and N. Strangman. Applying principles of universal design to test delivery: the effect of computer-based read-aloud on test performance of high school students with learning disabilities. *Journal of Technology, Learning, and Assessment*, 3(7), 2005. Available from <http://www.jtla.org>.
- C. Elbro, I. Rasmussen, and B. Spelling. Teaching reading to disabled readers with language disorders: A controlled evaluation of synthetic speech feedback. *Scandinavian Journal of Psychology*, 37:140–155, 1996. ISSN: 0036-5564.

- J. Elkind. Computer reading machines for poor readers. Technical Report 9801, Lexia Institute, January 1998.
- J. Elkind, M. S. Black, and C. Murray. Computer-based compensation of adult reading disabilities. *Annals of Dyslexia*, 46:159–186, 1996. ISSN: 0736-9387.
- R. Ellis. Principles of instructed language learning. *Asian EFL Journal*, 7(3):9–24, September 2005. ISSN: 1738-1460.
- H. Fujihara and M. Goto. Lyrics-to-audio alignment and its application. In M. Müller, M. Goto, and M. Schedl, editors, *Multimodal Music Processing*, volume 3 of *Dagstuhl Follow-Ups*, pages 23–36. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2012. ISBN 978-3-939897-37-8. doi: <http://dx.doi.org/10.4230/DFU.Vol3.11041.23>. URL <http://drops.dagstuhl.de/opus/volltexte/2012/3464>.
- W. Gale and K. Church. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102, 1993.
- J. Hamari, J. Koivisto, and H. Sarsa. Does gamification work? — a literature review of empirical studies on gamification. *2014 47th Hawaii International Conference on System Science*, 2014. <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6758978>.
- Z. Handley. Is text-to-speech synthesis ready for use in computer-assisted language learning? *Speech communication*, 51(10):906–919, October 2009. DOI: 10.1016/j.specom.2008.12.004.
- P. Hansson. Prosodic phrasing in spontaneous swedish. *Travaux de l’Institut de Linguistique de Lund* 43, pages 9–15, 2003.
- M. Haruno and T. Yamazaki. High-performance bilingual text alignment using statistical and dictionary information. *ACL ’96 Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 131–138, 1996. doi:10.3115/981863.981881.
- L. Hecker, L. Burns, L. Katz, J. Elkind, and K. Elkind. Benefits of assistive reading software for students with attention disorders. *Annals of Dyslexia*, 52, December 2002.
- G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97, 2012.
- W. H.-Y. Huang and D. Soman. Disassembling gamification: the effects of points and meaning on user motivation and performance. Rotman School of Management, Toronto, 2013. <http://inside.rotman.utoronto.ca/behaviouraleconomicsinaction/files/2013/09/GuideGamificationEducationDec2013.pdf>.
- P. Hubbard. Computer assisted language learning: Vol 1 (critical concepts in linguistics). 2009.

- W. L. Johnson. Serious use of a serious game for language learning. *Artificial Intelligence in Education: Building Technology Rich Learning Contexts That Work*, pages 67–74, 2007. ISBN: 978-1-58603-764-2.
- R. Kanters, C. Cucchiarini, and H. Strik. The goodness of pronunciation algorithm : a detailed performance study. In *In SLaTE 2009 - 2009 ISCA Workshop on Speech and Language Technology in Education*, pages 2–5, 2009.
- E. D. Mekler, F. Brühlmann, K. Opwis, and A. N. Tuch. Disassembling gamification: the effects of points and meaning on user motivation and performance. In *CHI EA '13 CHI '13 Extended Abstracts on Human Factors in Computing Systems*, pages 1137–1142, 2013. <http://dl.acm.org/citation.cfm?id=2468559>.
- D. R. Michael and S. L. Chen. *Serious Games: Games That Educate, Train, and Inform*, chapter 2: Serious Games Defined, pages 17–29. Course Technology PTR, 2006. <http://dl.acm.org/citation.cfm?id=1051239>.
- R. Naqvi, A. McKeough, K. Thorne, and C. M. Pfitscher. Dual-language books as an emergent-literacy resource: Culturally and linguistically responsive teaching and learning. *Journal of Early Childhood Research*, 13(4):501–528, 2013a. DOI: 10.1177/1468798412442886.
- R. Naqvi, K. J. Thorne, C. M. Pfitscher, D. W. Nordstokke, and A. McKeough. Reading dual language books: Improving early literacy skills in linguistically diverse classrooms. *Journal of Early Childhood Research*, 11(1):3–15, October 2013b. DOI: 10.1177/1476718X12449453.
- P. Skehan. *A Cognitive Approach to Language Learning*. OUP Oxford, 1998. ISBN: 0194372170.
- R. Sneddon. Magda and albana: Learning to read with dual language books. *Language and Education*, 22(2):137–154, 2008. DOI: 10.2167/le741.0.
- B. H. Sørensen and B. Meyer. Serious games in language learning and teaching – a theoretical perspective. *Proc. of the 3rd International Conference of the Digital Games Research Association*, pages 559–566, 2007.
- J. Tiedemann. Combining clues for word alignment. In *Paper presented at the EACL 2003, Department of Linguistics, Uppsala university, Sweden*, 2003.
- S. H. Ulanoff and S. L. Pucci. Learning words from books: The effects of read-aloud on second language vocabulary acquisition. *Bilingual Research Journal*, 23(4):409–422, 1999. DOI: 10.1080/15235882.1999.10162743.
- J. P. van Santen. Assignment of segmental duration in text-to-speech synthesis. *Computer Speech & Language*, 8(2):95–128, April 1994. doi:10.1006/csla.1994.1005.
- S. Witt and S. Young. Phone-level pronunciation scoring and assessment for interactive language learning. *Speech Communication*, 30:95–108, 2000.

J. Yamagishi. An introduction to hmm-based speech synthesis. October 2006.

H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. W. Black, and K. Tokuda. The hmm-based speech synthesis system (hts) version 2.0. *6th ISCA Workshop on Speech Synthesis*, August 2007.