Creating English Language Learning iOS Applications from Paper-based Texts – 2013 Progress Update

Gary V. Ireland* ・Joshua Rappeneker**・Maxim Woollerton***

Abstract

This paper is a progress update that describes how a series of English language learning textbooks (The English Course) is being dismantled and recreated for use as applications (known as iApps) for Apple's iOS devices: iPhone, iPad and iPod Touch. The paper explains the continuing development of the prototype modules, the latest aesthetic, pedagogic and the technical changes to the project modules.

Introduction

The English Course is a series of English language textbooks that have been created over the past seven years. The series currently comprises, two listening and speaking books and one writing book. Two reading books and a business English book are being prepared for publication. The overriding principle behind the development of the series was to produce textbooks and complementary materials, (such as audio and visual materials on DVD or CD, and Internet web-based self-access and self-marking text materials), that together formed a marriage (or balance) between using the communicative approach (CA) to language learning in the classroom, and utilizing the best techniques and materials available to incorporate computer assisted language learning (CALL). The authors have always tried to stay up-to-date, not only with developments in the CALL field, but with all technological developments in language learning methodology, and to incorporate this technology as extensively and as imaginatively as possible into the course.

The authors of *The English Course* series of English language learning materials have long believed that the age of printed textbooks is nearing an end. Being committed to utilizing technology in language learning, and already having a large amount of digital-based learning material available, the authors set themselves the challenge of dismantling the existing materials of *The English Course* series and recreating the material in the form iApps.

This project has proved to be a huge challenge and has presented an enormous learning curve

＊教授／CALL
＊＊非常勤講師／CALL
＊＊＊元非常勤講師／CALL

for all of the authors. There are several interconnected reasons for this. Firstly, three forms of media have had to be combined into one. Secondly, the physical size of the viewable area with which the authors interact with the users has been dramatically shrunk. Thirdly, the role of the teacher has been moved from an external and self-determining role to an internal and pre-determined role within the software, the features and attributes of the environment in which learning takes place have changed markedly, and the software language has rapidly evolved during the development of the project. With the continuing development of new operating systems with better capabilities, and with continued research and understanding of the technology and the learning environment, the project has been through numerous changes to date. The project is now entering its third year.

Evaluating and modifying the structure and content of the software

This section of the paper examines the evolution of the structure and content of the software. The changes that have been made to the structure of the software have been informed by paying close attention to principles of good interface and interaction design, based on trying to understand the nature and role of the software. In tandem with this process, there has been a continuous evaluation of the pedagogic function and value of the content. The development process is still incomplete and some changes still need to be made to the content and the functionality in order to improve the usefulness and usability of the software. The areas where there are deficiencies will be described and possible solutions will be discussed.

Defining the nature of the software

Computer software used for educational purposes can be generally categorized as being either ‘tutorial’ in nature or as a ‘tool’ for learning. The tutor or tool distinction was first suggested by Taylor (1980). Levy (1997) expanded upon it by suggesting it be used as the basis of a framework by which educational software can be evaluated pedagogically.

In a tutorial role, software includes the teacher within the machine, guiding and controlling the student through the use of the software (Levy: 182). The tutorial role also features evaluation of the performance of the student by the software. In the tool role, there is no teacher ‘within the machine’ and any function carried out by a teacher is in the form of external instruction or assistance (ibid: 185). If there is no process of evaluation within it, the software is deemed to be a tool (ibid: 180). Tools may be used by students either independent of one another or by working together (ibid: 185). A simple example of software that fits the definition of a tool would be a web browser application, such as *Safari*. The two roles are ‘non-dogmatic’ (ibid: 179) which means that sometimes the definition of software as being a tutor or tool may not be entirely accurate and the lines between the two may become blurred, but this should not be seen as a problem.

Importantly, it should be noted that a piece of software can contain elements within it that can be classified as tutorial as well as other elements that can be identified as tools. This mixture of tutor and tool is how *The English Course software* should be classified. For example, the *Video task* section is tutorial in nature as it contains instructions, scores the user's performance and gives feedback to the user. The same is also true of the *Audio game* section, but in the case of the *Pronunciation practice* section, the categorization is more difficult to make. While there are instructions for this section's use, there is no evaluation carried out by the software. It is the user who is free to interpret the results of use of the software in whatever way he or she decides is appropriate. Levy makes an important point of distinction between the two roles when he states that software operating in a tutor role 'requires methodology governing presentation and interaction' (ibid: 197).

Interaction Design Principles

Kristof and Satran (1995) have presented a clear set of interaction design principles that deal with user orientation to the software and usability issues. They suggest paying attention to the guidelines (below) when designing a piece of interactive software. Many of the suggestions are valid for any piece of interactive software, not just that produced for teaching English as a foreign language (EFL). These points will be examined first. Following that, issues that are specific to EFL software will be examined. Finally, changes to specific sections of the software will be discussed.

General Guidelines

The first issue is user orientation with the software. It is important that the software is designed in a way that helps users get the feel of the software quickly and understand what it is for and how it is organized. Each module of the software begins with the *Contents* screen. Attention has been given to ensuring that each button uses language that describes the content or function of the section that it leads to. From an EFL point of view, the language used for the buttons must be simple enough for basic level learners of English to be able to understand or come to understand quickly. There is scope for improving the design of the *Contents* screen further to give greater clarity of the purpose of the different sections, perhaps by adding pop-up screens that describe the purpose of each section behind 'question mark buttons' adjacent to the buttons that lead to the different sections. Again, from an EFL perspective the choice in the use of the level of the language used needs to be carefully controlled.

The second main issue relates to usability. Kristof and Satran present six ideas for maximizing the usability of a piece of interactive software (Kristof & Satran: 50-51). These are:

1. Remove obstacles [to users interacting with the software]
2. Minimize effort [for example, to navigation]
3. Give feedback [to let users know that something is happening]
4. Be explicit [so that users know what can be done with different objects on the screen]
5. Be flexible [to let users access or exit different features quickly]
6. Be forgiving [of user actions and do not limit the user to getting something right before moving on]

These guidelines have been followed in the design of *The English Course* software for the most part, but there have been some exceptions made, generally for pedagogic reasons. A basic example of this is the use of an English-only interface, rather than 'localising' the software by translating instructions, feedback and other text into various languages. The choice of using English only is a pedagogic and a practical one (at this stage, at least), and relates to the authors' embrace of the philosophy that it is better in the long run to immerse the student in the target language as much as possible. It could easily be argued that 'localising' the software would be better in terms of assisting in rapid orientation and removing obstacles to usability, and this may be done in future. On the other hand, the language used in the development of the software has been English, and until the structure of the software reaches a fairly final form it makes sense to keep the software monolingual.

The other principle that has not been followed in all circumstances is the fifth one: to be flexible. Again the reason is to do with where pedagogic issues conflict with general principles in the design of interactive software. In the role of tutor, the software is designed to follow certain sequences pertaining to access and navigation at certain times, in order to try and promote a style of learning and a way of using the software, that the authors believe is effective and will lead to better results. For example, in the *Video task* section the user may leave the section altogether and may return later, but is prohibited from moving from one question to another without selecting an answer and, as a consequence, the user must answer the questions in a single set sequence. Arguably, this is not the way users would answer the questions if they were writing on paper, as students are often observed answering the questions that they can first and returning to questions they find more difficult later. Discussion between the authors may continue as to if this is the best approach to take based on whether or not users respond positively or negatively to this particular behaviour of the software.

Another general principle that needs further attention is the third one: to give feedback. In certain sections, such as *Pronunciation practice*, what is happening is not always clear to the user. In this section, when any one of the three buttons is pressed at the bottom of the screen, it would be better if that button were illuminated or changed colour, until the operation that the button controls had completed. At present there is no indication that a button has been pressed, that it should be pressed, or that it can be pressed again.

Changes made to the software up to the present time and areas for improvement

Video section

The Video section can be considered to be a tool. The user just watches and listens in whatever way he or she likes. There are currently no instructions for its use. There are English subtitles that the user can switch on or off and the user can pause and resume or rewind and fast-forward at any time. There is no evaluation of the user by the software. This section of the programme has not changed in any meaningful way during the development of the programme, with the exception of the size of the text in the subtitles having been increased to make it easier for the user to read.

Video task section

The *Video task* section is a listening comprehension activity. Originally, the video task was presented as a series of ten statements about the dialogue contained in the *Video* section. The user scrolled through the statements and had to mark the statements true or false by tapping on a single check box next to each statement. Tapping once would mark a check box as true, while tapping twice on a check box would mark it false. There were a number of problems with this design.

The first problem was that the appearance was poor aesthetically as well as practically. As Kristof and Satran have pointed out, the act of scrolling does not appear as good a navigational action as the metaphor of page turning (Kristof & Satran: 57). Moreover, the scroll bar on the right of the screen hovered over the check boxes, making the interface cluttered and confusing. More importantly, a scrolling page led to severe limitations in the space available for text, which in turn forced the original design to use small-sized text, limited the length of the statements used and restricted the question type to true/false or yes/no question types. The fact that there was only one checkbox for each statement also seemed unnecessarily confusing and it was not immediately apparent what the user had to do.

To address these problems, the following changes were implemented. Firstly, the page metaphor replaced the scroll with one question or statement on each page. This brought a number of benefits. Firstly, it allowed for larger text sizes, longer lengths of text and multiple answer buttons or check boxes to enable the option to use any of several different question types. In addition, page numbering was added, since replacing the scroll bar had eliminated that feature's function as a progress indicator to the user. This is regarded as good practice when using the page metaphor for navigation (ibid: 57). Placing one question on each page also resulted in a vastly less cluttered appearance and gives the option of including other features on each page as and when necessary. A feature that was retained was disallowing the user from leaving answers blank before going on to the next question. If the user attempts to do this, a pop-up message instructs the user to select an answer before continuing. The user does have the option of returning to the Contents screen if he/she so desires, but selecting this option resets all hitherto answered questions back to an unanswered state. The aim here is to prevent the user from returning to check the video clip while in the process of answering the questions. When all questions have been answered a feedback page appears and shows the user which statements are true or false. The user can tap on a button to hear the part of the audio from the video clip that contains the content that determines whether a question or statement was true or false. At this stage of development, the feedback page does not indicate which of the user's answers were correct or incorrect and does not give the user a total score. This behaviour will be discussed by the authors and may need to be modified.

Language and vocabulary section

This section contains the text of the dialogue from the *Video* section. Originally the text was presented in small-sized text that resembled the way dialogues appear in traditional textbooks. This has now been changed to 'speech balloons' emanating from the edges of the screen that symbolize a dialogue between two people. The speech balloons are in one of two colours, with one colour being used for one speaker in the dialogue and the other colour for the second speaker. The use of these colourful speech balloons should hopefully be much more interesting for the user to look at and work with.

A new development is that certain vocabulary items that are highlighted in blue can be tapped and, when the user does so, pop-up floating windows containing definitions or explanations appear.

These pop-up floating windows can be hidden again by tapping the pop-up window once more. There are limited instructions and there is no evaluation of what the user does. At present, the page can just be scrolled through and vocabulary items can be checked. There are number of ideas being discussed about ways to expand the activity in this section and make it more interesting to use. One idea might be to incorporate a feature that was used on the DVD that the video and audio material was originally distributed on. On the DVD, the user had the option of turning off the audio for any of the characters in a video clip. The student was then supposed to interact with the remaining character by speaking the lines of the character that had been silenced.

Audio game section

The *Audio game* section in its current form is a tutorial in which students have to reassemble a dialogue that has been cut up line-by-line and jumbled. There are three levels of difficulty: easy, medium and hard. This not only allows students to test their listening skills at different levels, but also to practice with the activity several times in a different mode.

Pronunciation practice section

The *Pronunciation practice* section is the closest thing in the programme to a software tool, although it does contain instructions for how to use the buttons on the screen. The screen also provides a form of feedback on the Audio game section that precedes it, as it presents the lines of dialogue in text form in their correct order. On the other hand, there is no evaluation of the user's performance.

In terms of usability, there is still more work to be done in several areas to improve this section. Firstly, at the moment, the instructions do not explain the point of the activity, which is simply to engage in pronunciation practice, but not to the extent of mimicking the guide audio. *The English Course* uses a variety of speakers from various countries, and many of them are non-native speakers of English. The authors do not want users to be exposed to and copy just one style of pronunciation. The educational philosophy here is that most EFL learners will interact mainly with other non-native rather than native speakers of English. Even when visiting an English-speaking country such as Britain, EFL students will be encounter a society where only a small proportion of the English-speaking population uses the standard model of pronunciation.

Secondly, there is no visual signal for the user to understand what is happening, which is especially important for the *Microphone* button because the user is supposed to speak at this point. There is an audio beep when the *Microphone* button is pressed to begin recording, but this could be quite off-putting because it sounds at the same moment that the user should begin to speak. It might be better to have a visual indication only such as the button highlight/colour change scheme that is detailed below. Moreover, if the beep is retained it needs to be quieter and shorter. As for visual signals for the user, the following scheme (Table 1) is a possibility that the authors may decide to adopt. In this scheme buttons should highlight or change colour, or change shape in order to indicate what is happening, or what should or could happen, as follows (although in determining the actual colours used, consideration will have to be given to users who may be affected by red-green colour blindness):

Play button
- Black = not in use
- Green = available/suggested for use
- Red = in use

Microphone button
- Black = not in use
- Green = available/suggested for use
- Red = in use

Compare button
- Black = not in use
- Green = available/suggested for use
- Red = in use

Table 1 – Colour coding

#	*Play* button	*Microphone* button	*Compare* button
1	Green = available/suggested	Black = not in use	Black = not in use
2	Red = in use	Black = not in use	Black = not in use
3	Black = not in use	Green = available/suggested	Black = not in use
4	Black = not in use	Red = in use	Black = not in use
5	Black = not in use	Black = not in use	Green = available/suggested
6	Black = not in use	Green = available/suggested	Black = not in use
7	Black = not in use	Red = in use	Black = not in use
8	Black = not in use	Black = not in use	Green = available/suggested

Questions and answers section

The *Questions and answers* section is a tutorial activity. Although there are no instructions at the moment, this omission will be dealt with and instructions will be available. The activity involves listening to a question and then listening to three responses. The user has to select the correct response. This section of the programme has a similar layout to and shares the same problem as that in the *Video task* section. Namely, each item in the activity appears on a separate page, but the user cannot advance to the next item without making an answer selection for the current item.

The feedback for this activity features a scrolling page in which large clear and colourful icons indicate which items the user has got right and which were incorrect. The user is also able to listen again to the question and the correct answer for each item in this section. The way the feedback works in this section may need some improvement, however. Currently, whether the user has got an item correct or incorrect, the audio available on the feedback is only the correct question and response. It might be better if two of the three responses were available for items that have been answered incorrectly; the correct response and the incorrect one that the user selected previously, in order that the user can listen again. There are other possibilities as well and the audio feedback in this section needs further discussion between the authors.

Dictionary and phrasebook section

The *Dictionary and phrasebook* section was a single scrolling page with vocabulary items and definitions. It also included a subsection called *Language focus*, which contained key expressions related to the target function of the module. The authors decided to delete the list of vocabulary items, but to retain the *Language focus* section. The reason for doing so was that the authors considered that users would probably not want to spend time scrolling through a list of words, because it is highly unusual for readers to spend time looking through printed book dictionaries. Learners usually consult a dictionary when they come across a word in a passage of text that they do not know. Therefore, the authors decided to create the user access to the dictionary within the software directly from the text passage being viewed on the screen. This is also a much faster way for users to be able to check vocabulary items, rather than for them to notice an unknown word, exit the section of the programme they are in to enter the *Dictionary section*, and then scroll to find the word they want.

Language focus section

The *Language focus* section will remain as a separate part of the software, but will be redesigned, using the speech balloon style of presentation used in the *Language and vocabulary* section (and may also be renamed). This should hopefully make the section more visually stimulating for the user to interact with.

Test section

This section of the software is very similar to the way that the *Video task* section used to appear and behave. Although there are no instructions at the moment for the *Test* section, this omission will be dealt with and instructions will be included in the software for this section. At present, the *Test* section consists of a scrolling page, containing sentences that are either grammatically correct or incorrect and appropriate to the situations covered in the module. There is one check box, which when tapped once changes to a ticked state, when tapped again changes to a crossed state, and when tapped a third time is cleared. It is difficult for the user to quickly understand what is being tested and how to interact with the items on the screen. Finally, the user's responses are evaluated. It is clear that this section of the software is still in need of major adjustments if it is to be much easier for users to quickly understand what the section tests and what to do. Proposed changes may be discussed in response to the feedback received through the survey of users that is going to be conducted in Autumn 2013.

Instructions

Instructions pages have been added or will be added for all sections of the software. The wording of existing *instructions* pages will be reviewed and may be revised, if it is thought that the instructions are unclear. One important feature of the *instructions* pages is an option at the bottom of the screen that allows the user to avoid having to see each instructions page, if he or she wishes to skip this particular page. If required, the user is able to recall the instructions for this section from a button at the top of all pages within each section. At present, the two options for the user on the instructions pages are labelled 'Okay' and 'Never show again'. This wording is not particularly accurate or clear. A new proposal is to reword the options as follows:

Next time:
Show instructions　　Skip instructions

Concluding remarks for this section

It will have become apparent through this section of this paper that *The English Course* software is still in mid-development. Much progress has been made to improve the usefulness of the content, the usability of the interface and the ease and speed with which users are able to be orientated to the software. Some work still needs to be done to further improve these key areas in order to create a valuable and effective learning product. The authors decided that another thing needed at this stage in the development process is user feedback in the form of a survey linked to actual use of the software by students and teachers, in order that the software authors are not second-guessing the user's experience. A summary of the design of this survey appears in the conclusion below.

Technical challenges

This section of the paper will focus on the technical challenges faced with the adoption of Apple's interface guidelines for iOS 7.0, the latest iDevice operating system.

iOS 7.0, released on 18 September 2013, is perhaps the most dramatic change to the iDevice mobile operation system since the release of the original iPhone [Business Insider, Tech Crunch]. Initially most obvious changes are visual, although there have been many changes 'under the hood', which require a large amount of rewriting to take advantage of. However, with an adoption rate of over 60%, after just one week of release it is clear that users are enthusiastically embracing the new OS. With this in mind, and the fact that users are much more likely to dismiss an app which doesn't conform to the visual paradigm of the new OS [reference], it is necessary for us to rewrite the app to match users' expectations. At the same time, it provides us with the exciting opportunity to improve the app in using the additions offered by the new OS and its new application programming interfaces (APIs); the interfaces between the high level abstractions the developer works with and the low level access to hardware and the phone's operating system.

Apple released a mobile human interface guide for its new iOS, which focuses on three aspects of user interface (UI) design: deference, clarity, and depth. The following section of the paper will discuss how the app has been adapted to conform to the new guidelines.

Deference

The UI helps users understand and interact with the content, but never competes with it. [from mobile HIG]

“Take advantage of the whole screen.” [mHIG]

Prior to iOS7.0, content was restricted to the portion of the screen between the status and navigation bar at the top of the screen, and the toolbar (if present) at the bottom of the screen. Apps would typically also inset content from the edges of the screen, leaving large portions of screen space unused. Since the release of iOS 7.0, the status bar is now transparent by default, and navigation bars are translucent, allowing content to show up underneath, giving the user a greater sense of the context of the information being presented. Removing insets from the screen and making sure content could scroll beneath the top bars was fortunately one of the easier tasks in the adaptation to iOS 7.0. [see images]

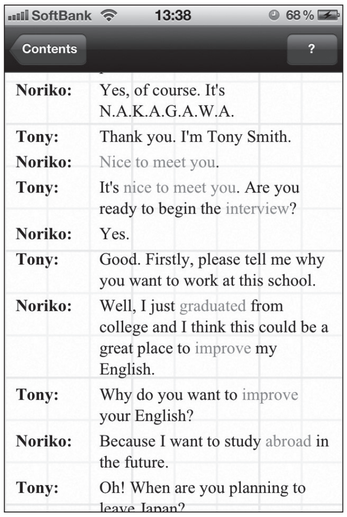
Clarity

Text is legible at every size, icons are precise and lucid, adornments are subtle and appropriate, and a sharpened focus on functionality motivates the design. [mHIG]

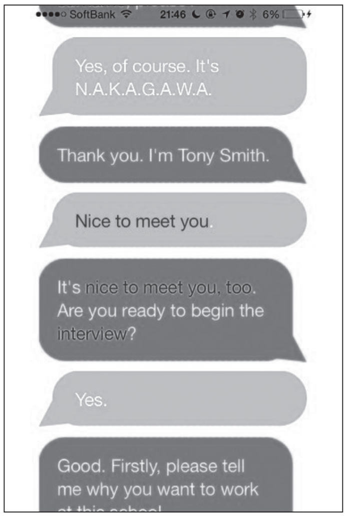
“Use plenty of negative space” [mHIG]

According to Apple’s guidelines, apps which employ the use of negative space, make important content more obvious, and the app appear more focused. As written above this resulted in the removal of most background images and colours, especially in text heavy settings, such as the scripts.

Table 2 – Deference in action - the content on the right stretches to the four edges of the screen



Noriko:	Yes, of course. It's N.A.K.A.G.A.W.A.
Tony:	Thank you. I'm Tony Smith.
Noriko:	Nice to meet you.
Tony:	It's nice to meet you. Are you ready to begin the interview?
Noriko:	Yes.
Tony:	Good. Firstly, please tell me why you want to work at this school.
Noriko:	Well, I just graduated from college and I think this could be a great place to improve my English.
Tony:	Why do you want to improve your English?
Noriko:	Because I want to study abroad in the future.
Tony:	Oh! When are you planning to leave Japan?



Yes, of course. It's N.A.K.A.G.A.W.A.

Thank you. I'm Tony Smith.

Nice to meet you.

It's nice to meet you, too. Are you ready to begin the interview?

Yes.

Good. Firstly, please tell me why you want to work

"Let colour simplify the UI" [mHIG]

The guidelines also suggest the use of a 'key colour' to unify the UI and indicate interactivity. We have decided on two key colours: [blue] to indicate interaction within a certain screen, and [red] to indicate that interaction will take the user to a different screen of content.

For example, a user touching a [blue] word or phrase when looking at a script will cause a definition to pop over the current screen, which can then be dismissed by another touch. Interacting with a [red] word or phrase will take the user to a longer explanation of the phrase on a separate screen. Consistency throughout the interface should make the user more confident in the consequences of interactions, and help the user learn the interface more quickly.

Colour choices are, however, not simply aesthetic. It is strongly encouraged in the guide to avoid the use of red and green to indicate actions with different results, as users with red-green colour blindness will find it difficult to differentiate between the two states. Either different colour combinations can be chosen, or the interactive element can be modified in other ways in addition to its colour. Using shapes, for example a circular icon for 'record', a square icon to indicate 'stop', and a triangular icon for 'play', works even if the user is unable to differentiate colours.

"Ensure legibility by using the system fonts" [mHIG]

With iOS 7.0, Apple has introduced a new technology, Dynamic Type, which offers dynamically sized system fonts. In the past, developers would design their interfaces choosing the font size for each UI element, delivering a standardised experience for all users. Developers would also choose the font face and weight to best suit the context. With the introduction of Dynamic Type, developers are able to delegate these decisions to the OS.

The use of Dynamic Type in an application automatically resizes all application text to match the users OS wide preferences. This allows users who have vision difficulties to enlarge text to a comfortable reading size. Similarly, it allows users who wish to display more text per screen to do so. Dynamic Type also affords UI consistency, giving the users clues to the meaning of UI elements based on their font face and weight.

Invoking the new type is very simple, and makes for cleaner, more readable code. In the past the font for heading may have been declared (announced to the compiler) like so:

```
UIFont *oldStyleHeadline =
[UIFont fontWithName:@”HelveticaNeue-CondensedBold” size:17.0f];
```

(The coding examples in this paper use CamelCase, meaning that in compound phrases, each word after the first begins with a capital letter. Thus, oldStyleHeadline should be read ‘old style headline’.)

In this line of code, the developer chooses to use the font family ‘Helvetica Neue’, the styling ‘Condensed Bold’, and a size of 17 points. This allows the developer a lot of flexibility – there are 11 styles of Helvetica Neue to choose from, and many more fonts with similar numbers of styles. However, it does complicate the process of adjusting the UI and keeping track of changes throughout the code.

The new, Dynamic Type, method declares fonts in the following way:

```
UIFont *newStyleHeadline =
[UIFont preferredFontForTextStyle:UIFontTextStyleHeadline];
```

The developer in this example need only select which type of text the element requires. The font family, styling and size are all chosen by the OS, based on the user’s universal preferences.

This allows for significantly less flexibility, as there are only six different text types: headings, subheadings, body text, footnotes, captions, and alternate captions. This has the added advantage of letting the developer focus on semantic content rather than design, in the choice of fonts.

Although Dynamic Type does allow for some simplification of the code, it does not come without some added difficulties. The sizes of system fonts that Dynamic Type offers vary significantly in size [add image], which means the user interface must also dynamically adjust to changes in size. The app must then ‘listen’ for notifications of any font size change and modify the interface accordingly. In some cases, this might mean moving from two columns to one to accommodate a larger font size, in others it may require redrawing the whole screen of content. This means that ‘pixel perfect’ designs (interfaces designed in an image editing program, then replicated pixel-for-pixel in the app) are very difficult without iterating through each possible font size.

Taking difficulties of implementation aside, the introduction of Dynamic Type is a boon for text-heavy applications such as ours. Properly done, it lets the developers focus on the content of the application, and trust that the text will be legible for all users.

“Embrace borderless buttons” [mHIG]

One of the more striking changes to iOS 7.0 was the introduction and widespread use of borderless buttons. Prior to iOS 7.0, buttons were by default surrounded by rounded rectangles. In iOS 7.0, all buttons in the bars at the top and bottom of the screen are now borderless, and it is strongly encouraged that all buttons on screen be the same, although in certain cases it is permitted to add ‘a thin border or tinted background that makes it distinctive’. The previous version of the app used bordered buttons throughout the interface, so removing their borders makes for a significant aesthetic change.

Depth

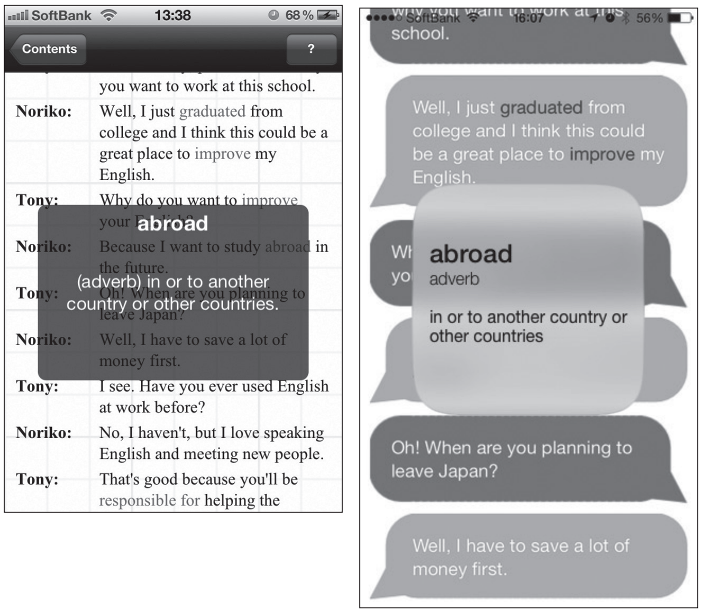
Visual layers and realistic motion impart vitality and heighten users’ delight and understanding.
[mHIG]

Visual layers [mHIG]

Using visual layers which appear to float above the main content of the screen helps confirm the relationship between the newly presented content while maintaining separate mental spaces for different types of content. Prior to iOS 7.0, the app used a similar technique, however in iOS 6.0, blurring was a very difficult and computationally expensive operation, which meant that the

pop-over views in the app were slightly difficult to read, depending on the underlying content. New APIs in iOS 7.0 have made blurring background views straightforward and relatively computationally inexpensive, which allows us to employ it throughout the UI.

Table 3 – Blurring gives the user a sense of depth and an understanding of the hierarchy of information within the application, without sacrificing legibility



	you want to work at this school.
Noriko:	Well, I just graduated from college and I think this could be a great place to improve my English.
Tony:	Why do you want to improve your English?
Noriko:	Because I want to study abroad in the future.
Tony:	Oh! When are you planning to leave Japan?
Noriko:	Well, I have to save a lot of money first.
Tony:	I see. Have you ever used English at work before?
Noriko:	No, I haven't, but I love speaking English and meeting new people.
Tony:	That's good because you'll be responsible for helping the

abroad

(adverb) in or to another country or other countries.

school.

Well, I just graduated from college and I think this could be a great place to improve my English.

abroad

adverb

in or to another country or other countries

Oh! When are you planning to leave Japan?

Well, I have to save a lot of money first.

Realistic motion [mHIG]

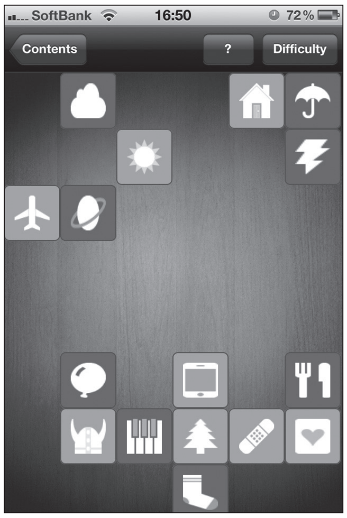
Sprite Kit

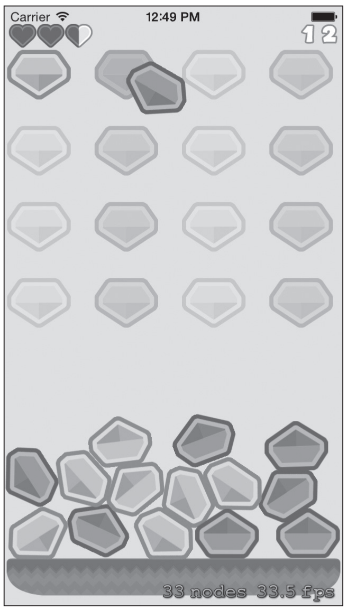
One of the biggest additions to the APIs in iOS was the introduction of ‘Sprite Kit’, Apple’s 2D game API. Sprite Kit gives developers access to a physics simulation, as well as very simple methods of animating and dealing with interactions between ‘sprites’ (2 dimensional images which make up a game screen).

Using the new Sprite Kit, it was possible to add much more game-like mechanics to the Audio Game section. What was previously a series of buttons that could be dragged into the correct order, has been replaced with sprites that interact with each other, as well as being affected by gravity and other physics. Whilst the fundamental aim of the game hasn’t changed, the new activity is much more ‘fun’.

One of the problems with the previous version of the game was that it was simple to 'brute-force' the solution without actually having to understand the conversation. Dropping a square on an incorrect space in the conversation resulted in the square zooming back to its original position, and no other penalty. In the new version of the game, releasing a sprite in an incorrect position results in a sound playing indicating the user has made a mistake, as well as a reduction in score and 'health'. Health is represented by heart sprites displayed at the top left of the screen. A player losing all health will result in the game ending and the need to restart it in order to play again. The number of hearts a player starts with, and the rate at which they are lost is determined by the difficulty. A player can gain points by dragging the sprites to the correct positions, gaining more points for faster correct actions. The introduction of scores and penalties is designed to give the user an achievable but somewhat difficult goal, with the end effect of increasing user motivation. [game design / fun theory references throughout paragraph].

Table 4 – Sprites, as seen on the right, make the experience much more 'game-like', hopefully increasing the user's enjoyment of the experience





Conclusion

There are several conclusions to be drawn from this continued research and development project. The first is that without setting clear deadlines on completion dates, it could easily become an endless exercise in improvement change, which loses sight of its original purpose. In continuing to evaluate the aesthetic, pedagogic and technical qualities of this iApp, the look, feel and successful usability of the end product have certainly been improved. However, consideration now has to be given as to when to stop this improvement process. Given the fact, in particular, that the technical limitations of what can be done are changing every day, work on improving the app could go on forever. There has to come a time when the authors say it is now good enough – and after three years of development – it now seems to be the right time. It is now time to release the module for others to see and judge, and to learn the opinions of people outside of the authors' small group. This leads us to the second conclusion.

The second conclusion is that it is now time to test the application on an audience and attain as much feedback as possible on the actual user experience of the prototype module. With this in mind, a survey will be given to as many English language students (and teachers) as possible over a three-month period. The participants will primarily consist of college and university students in Japan. There are two reasons for this. Firstly, as the authors are teaching in this environment, it is the easiest way to administer the survey and collect data. Secondly, it is felt that this age group and type of learner are likely to be interested in an iApp course. If it proves to be at all possible, the authors are very keen to have the same experimentation and survey completed by students at universities in other countries, to try to understand if this affects the way students feel about and interact with the app. Where possible, the authors will also give the survey to other learners from different learning environments. In addition, it is also intended to obtain feedback and criticism of the app from other language teachers, who will be asked to evaluate the app from an educational viewpoint.

The survey has been designed in 3 sections and aims to gather data on the following three points:

- Demographic data of the people surveyed
- Experience of using technology for educational purposes of the people surveyed
- Overall opinion of the prototype module and specific opinions regarding the different sections of the module

Some of the areas covered by the questions on the survey will include; usability, understandability of the app's purpose, comments about design and functionality of the app and general indications about what want learners want from such an application.

The survey currently comprises 26 questions in total and is available in English and Japanese. The survey will be conducted online where circumstances allow, and in paper form where access to computers is not available.

The reason for the time needed to collect the data is related to the difficulty in giving learners access to the application. As the application is in development stage, it cannot yet be downloaded from Apple Inc's App Store. This means that evaluators can only get the application directly from the authors. In order for the authors to give an evaluator the application, the user's device's Unique Device Identifier (UDID) would have to be obtained, which is a security risk for the evaluator. In addition, the authors would also have to ask each evaluator to download and install the app, which may be time-consuming and problematic.

The only practical method of allowing evaluators to experiment with the app and complete the survey is for the authors to lend phones to evaluators one at a time. It may be possible to have selected teachers assist with this, by doing the same thing in their classes. Obviously, it will take some time to collect and evaluate this feedback. This will form the basis of a future paper, which will be published in 2014.

The third conclusion was related to usability. It was decided that, where possible, the authors should follow the guidelines laid out by Kristof and Satran (above) for maximizing the usability of a piece of interactive software (Kristof & Satran: 50-51). The areas in which we have deferred from these guidelines were, generally speaking, for pedagogic reasons only. As mentioned earlier, these guidelines were as follows:

1. Remove obstacles [to users interacting with the software]
2. Minimize effort [for example, to navigation]
3. Give feedback [to let users know that something is happening]
4. Be explicit [so that users know what can be done with different objects on the screen]
5. Be flexible [to let users access or exit different features quickly]
6. Be forgiving [of user actions and do not limit the user to getting something right before moving on]

The authors believe, that for the learner user, this language-learning prototype module created to-date still has no equal in its complexity or interactivity. The authors also believe that it is now time to test the module on learners through the survey usage and to then continue to create additional modules in the format of the final version of the prototype. It is hoped that with careful selection of materials from *The English Series*, perhaps as many as 40 different modules can be created over a period of time. It is an exciting prospect, and hopefully a step towards self-study through the use of iOS applications that many other materials developers and willing and capable teachers may follow.

References

Kristof, R. & A. Satran (1995) *Interactivity by Design*, (Mountain View, California: Adobe Press)

Levy, M. (1997) *Computer-Assisted Language Learning; Context and Conceptualization,* (Oxford: Clarendon Press)

Taylor, R. P. (1980) (ed.), *The Computer in the School: Tutor, Tool, Tutee*, Teacher's College, Colombia University (New York: Teacher's College Press, New York).

Koster, Raph. (2004) *Theory of Fun for Game Design*. (Pheonix: Paraglyph Press).

https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/index.html

http://serc.carleton.edu/introgeo/games/goodgame.html

http://serc.carleton.edu/resources/1606.html

http://serc.carleton.edu/introgeo/games/howtogbl.html

http://www.businessinsider.com/ios-6-versus-ios-7-apps-2013-9

http://techcrunch.com/2013/09/18/apple-re-invents-its-wheel-with-ios7-takes-developers-along-for-the-ride/

（2013.10.10 受稿，2013.12.11 受理）